FreeBSD 使用手冊

FreeBSD 文件計畫

FreeBSD 使用手冊

by FreeBSD 文件計畫

Published February 1999

歡迎使用FreeBSD！本使用手冊涵蓋範圍包括了*FreeBSD 7.2-RELEASE* 和*FreeBSD 8.0-RELEASE* 的安裝和日常使用。這份使用手冊是很多人的集體創作，而且仍然『持續不斷』的進行中。許多章節仍未完成，已完成的部份也有些需要更新。如果您有興趣協助本計畫的話，請寄e-mail 到FreeBSD documentation project 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc)。在FreeBSD 網站 (http://www.FreeBSD.org/) 可以找到這份文件的最新版本(舊版文件可從http://docs.FreeBSD.org/doc/ 取得)，也可以從FreeBSD FTP 伺服器 (ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/) 或是衆多mirror 站臺 下載不同格式的資料。如果比較偏好實體書面資料，那可以在FreeBSD Mall (http://www.freebsdmall.com/) 購買。此外，也可以在使用手冊 (http://www.FreeBSD.org/search/index.html) 中搜尋資料。

Apple, AirPort, FireWire, Mac, Macintosh, Mac OS, Quicktime, 以及TrueType 是Apple Computer, Inc. 在美國以及其他國家的註冊商標。
Corel 和WordPerfect 是Corel Corporation 和/或其子公司在加拿大、美國和/或其他國家的註冊商標。
Sound Blaster 是Creative Technology Ltd. 在美國和/或其他國家的註冊商標。
CVSup 是John D. Polstra 的註冊商標。
Heidelberg, Helvetica, Palatino, 和Times Roman 是Heidelberger Druckmaschinen AG 在美國以及其他國家的商標或註冊商標。
IBM, AIX, EtherJet, Netfinity, OS/2, PowerPC, PS/2, S/390, 和ThinkPad 是國際商用機器公司在美國和其他國家的註冊商標或商標。
IEEE, POSIX, 和802 是Institute of Electrical and Electronics Engineers, Inc. 在美國的註冊商標。
Intel, Celeron, EtherExpress, i386, i486, Itanium, Pentium, 和Xeon 是Intel Corporation 及其分支機構在美國和其他國家的商標或註冊商標。
Intuit 和Quicken 是Intuit Inc., 或其子公司在美國和其他國家的商標或註冊商標。
Linux 是Linus Torvalds 的註冊商標。
LSI Logic, AcceleRAID, eXtremeRAID, MegaRAID 和Mylex 是LSI Logic Corp 的商標或註冊商標。
M-Systems 和DiskOnChip 是M-Systems Flash Disk Pioneers, Ltd. 的商標或註冊商標。
Macromedia, Flash, 以及Shockwave Macromedia, Inc. 在美國和/或其他國家的商標或註冊商標。
Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media, 和Windows NT 是Microsoft Corporation 在美國和/或其他國家的商標或註冊商標。
Netscape 以及Netscape Navigator 是Netscape Communications Corporation 在美國和其他國家的註冊商標。
GateD 和NextHop 是NextHop 在美國和其他國家的商標或註冊商標。
Motif, OSF/1, 和UNIX 是The Open Group 在美國和其他國家的註冊商標；IT DialTone 和The Open Group 是其商標。
Oracle 是Oracle Corporation 的註冊商標。
PowerQuest 和PartitionMagic 是PowerQuest Corporation 在美國和/或其他國家的註冊商標。
RealNetworks, RealPlayer, 和RealAudio 是RealNetworks, Inc. 的註冊商標。
Red Hat, RPM, 是Red Hat, Inc. 在美國和其他國家的註冊商標。
SAP, R/3, 和mySAP 是SAP AG 在德國以及許多其他國家的商標或註冊商標。
Sun, Sun Microsystems, Java, Java Virtual Machine, JavaServer Pages, JDK, JSP, JVM, Netra, Solaris, StarOffice, Sun Blade, Sun Enterprise, Sun Fire, SunOS, 和Ultra 是Sun Microsystems, Inc. 在美國和其他國家的商標或註冊商標。
Symantec 和Ghost 是Symantec Corporation 在美國和其他國家的註冊商標。
MATLAB 是The MathWorks, Inc. 的註冊商標。
SpeedTouch 是Thomson 的商標。
U.S. Robotics 和Sportster 是U.S. Robotics Corporation 的註冊商標。
VMware 是VMware, Inc. 的商標
Waterloo Maple 和Maple 是Waterloo Maple Inc 的商標或註冊商標
Mathematica 是Wolfram Research, Inc 的註冊商標。
XFree86 是The XFree86 Project, Inc 的商標。.
Ogg Vorbis 和Xiph.Org 是Xiph.Org 的商標。
許多製造商和經銷商使用一些稱為商標的圖案或文字設計來彰顯自己的產品。本文中出現的衆多商標，以及FreeBSD Project 本身廣所人知的商標，後面將以’™’ 或’®’ 符號來標註。

Table of Contents

List of Tables

序

給讀者的話

若您是第一次接觸FreeBSD 的新手，可以在本書第一部分找到FreeBSD 的安裝方法，同時會逐步介紹UNIX® 的基礎概念與一些常用、共通的東西。而閱讀這部分並不難，只需要您有探索的精神和接受新概念。

讀完這些之後，手冊中的第二部分花很長篇幅介紹的各種廣泛主題，相當值得系統管理者去注意。在閱讀這些章節的內容時所需要的背景知識，都註釋在該章的大綱裡面，若不熟的話，可在閱讀前先預習一番。

延伸閱讀方面，可參閱Appendix B。

第三版的主要修訂

您目前看到的這本手冊第三版是FreeBSD 文件計劃的成員歷時兩年完成的心血之作。新版的主要修訂部分，如下：

- Chapter 11, 設定與效能調校(Tuning)，該章節針對新內容作更新，比如：ACPI 電源管理、cron、以及其他更多的kernel tuning 選項說明內容。
- Chapter 14, 系統安全篇，該章節增加了虛擬私人網路(VPN)、檔案系統的存取控制(ACL)，以及安全公告(Security Advisories)的內容。
- Chapter 16, 集權式存取控制(MAC)是本版所增加的章節。本章介紹：什麼是MAC 機制？以及如何運用它來使您的FreeBSD 系統更安全。
- Chapter 18, 儲存設備篇，新增了像是：USB 隨身碟、檔案系統快照(snapshot)、檔案系統配額(quota) 、檔案及網路的備援檔案系統、以及如何對硬碟分割區作加密等詳解。
- Chapter 20Vinum是本版所新增的章節。本章介紹：如何運用Vinum 這種邏輯磁碟(device-independent) ，以及軟體RAID-0, RAID-1 和RAID-5 。
- Chapter 25PPP 及SLIP 一章中增加了故障排除的說明。
- Chapter 26電子郵件一章中新增有關如何使用其它的MTA 程式、SMTP 認証、UUCP、fecthmail、procmail 的運用以及其它進階專題。
- Chapter 27, 網路伺服器篇，是新版中全新的一章。這一章介紹了如何架設Apache HTTP 伺服器、FTPd，以及用於支援Microsoft Windows client 的Samba 伺服器。其中有些段落來自原先的Chapter 29進階網路應用一章。
- Chapter 29進階網路應用一章新增有關在FreeBSD 中使用藍芽設備、安裝無線網路以及使用ATM(Asynchronous Transfer Mode) 網路的介紹。
- 增加了一份詞彙表(Glossary)，用以說明全書中出現的術語。
- 重新美編書中所列的圖表。

第二版的主要修訂

本手冊的第二版是FreeBSD 文件計劃的成員歷時兩年完成的心血之作。第二版包含了如下的主要變動：

- 增加完整的目錄索引。
- 所有的ASCII 圖表均改成圖檔格式的圖表。
- 每個章節均加入概述，以便快速的瀏覽該章節內容摘要、讀者所欲了解的部分。
- 內容架構重新組織成三大部分: “開始使用FreeBSD”, “系統管理”, and “附錄”。
- Chapter 2 (“安裝FreeBSD”) 已經重新改寫，並加入許多說明圖片，讓第一次接觸FreeBSD 的人，直接看圖就很清楚明瞭重點(一圖抵千字的效果)。
- Chapter 3(“UNIX 基礎概念篇”)新增了processes, daemons 和signals 的介紹。
- Chapter 4 (“軟體套件管理篇”)新增了介紹如何管理binary package 套件的資訊。
- Chapter 5 (“X Window 視窗系統篇”) 經過全面改寫，著重於在XFree86™ 4.X 上的流行x11-wm，像是：**KDE** 和**GNOME** 。
- Chapter 12 (“FreeBSD 開機流程篇”)更新相關內容。
- Chapter 18 (“儲存設備篇(Storage)”) 分別以兩個章節“Disks” 與“Backups” 來撰寫。我們認為這樣子會比單一章節來得容易瞭解。還有關於RAID(包含硬體、軟體RAID) 的段落也新增上去了。
- Chapter 24(“Serial 通訊篇”)架構重新改寫，並更新至FreeBSD 4.X/5.X 的內容。
- Chapter 25 (“PPP 及SLIP”)有相當程度的更新。
- Chapter 29(“進階網路應用篇”)加入許多新內容。
- Chapter 26 (“電子郵件篇”)大量新增了設定**sendmail** 的介紹。
- Chapter 10 (“Linux® 相容篇”) 增加許多有關安裝**Oracle®** 及**SAP® R/3®** 的介紹。
- 此外，第二版還新加章節，以介紹下列專題：
  - Chapter 11, 設定與效能調校(Tuning)。
  - Chapter 7, 多媒體影音娛樂(Multimedia)。

本書架構

本書主要分為五大部分，第一部份『開始使用』：介紹FreeBSD 的安裝、基本操作。讀者可根據自己的程度，循序或者跳過一些熟悉的主題來閱讀；第二部分『常用操作』：介紹FreeBSD 常用功能，這部分可以不按順序來讀。每章前面都會有概述，概述會描述本章節涵蓋的內容和讀者應該已知的，這主要是讓讀者可以挑喜歡的章節閱讀；第三部分『系統管理』：介紹FreeBSD 老手所感興趣的各種專題部分；第四部分『網路通訊』：則包括網路和各式Server 專題；而第五部分『附錄』：是各種有關FreeBSD 的資源。

Chapter 1, 簡介篇

向新手介紹FreeBSD。該篇說明了FreeBSD 計劃的歷史、目標和開發模式。

Chapter 2, 安裝篇

介紹安裝程序。其中還有介紹一些進階的安裝主題，包括像是如何透過serial console 來安裝。

Chapter 3, UNIX 基礎概念篇

Covers the basic commands and functionality of the FreeBSD operating system. If you are familiar with Linux or another flavor of UNIX then you can probably skip this chapter.

Chapter 4, 軟體套件管理篇

Covers the installation of third-party software with both FreeBSD's innovative "Ports Collection" and standard binary packages.

Chapter 5, X Window 視窗系統篇

Describes the X Window System in general and using X11 on FreeBSD in particular. Also describes common desktop environments such as **KDE** and **GNOME**.

Chapter 6, Desktop Applications

Lists some common desktop applications, such as web browsers and productivity suites, and describes how to install them on FreeBSD.

Chapter 7, Multimedia

Shows how to set up sound and video playback support for your system. Also describes some sample audio and video applications.

Chapter 8, Configuring the FreeBSD Kernel

Explains why you might need to configure a new kernel and provides detailed instructions for configuring, building, and installing a custom kernel.

Chapter 9, 列印篇

Describes managing printers on FreeBSD, including information about banner pages, printer accounting, and initial setup.

Chapter 10, Linux Binary Compatibility

Describes the Linux compatibility features of FreeBSD. Also provides detailed installation instructions for many popular Linux applications such as **Oracle**, **SAP R/3**, and **Mathematica®**.

Chapter 11, Configuration and Tuning

Describes the parameters available for system administrators to tune a FreeBSD system for optimum performance. Also describes the various configuration files used in FreeBSD and where to find them.

Chapter 12, Booting Process

Describes the FreeBSD boot process and explains how to control this process with configuration options.

Chapter 13, Users and Basic Account Management

Describes the creation and manipulation of user accounts. Also discusses resource limitations that can be set on users and other account management tasks.

Chapter 14, Security

Describes many different tools available to help keep your FreeBSD system secure, including Kerberos, IPsec and OpenSSH.

Chapter 16, Mandatory Access Control

Explains what Mandatory Access Control (MAC) is and how this mechanism can be used to secure a FreeBSD system.

Chapter 18, Storage

Describes how to manage storage media and filesystems with FreeBSD. This includes physical disks, RAID arrays, optical and tape media, memory-backed disks, and network filesystems.

Chapter 19, GEOM

Describes what the GEOM framework in FreeBSD is and how to configure various supported RAID levels.

Chapter 20, Vinum

Describes how to use Vinum, a logical volume manager which provides device-independent logical disks, and software RAID-0, RAID-1 and RAID-5.

Chapter 22, Localization

Describes how to use FreeBSD in languages other than English. Covers both system and application level localization.

Chapter 23, The Cutting Edge

Explains the differences between FreeBSD-STABLE, FreeBSD-CURRENT, and FreeBSD releases. Describes which users would benefit from tracking a development system and outlines that process.

Chapter 24, Serial Communications

Explains how to connect terminals and modems to your FreeBSD system for both dial in and dial out connections.

Chapter 25, PPP and SLIP

Describes how to use PPP, SLIP, or PPP over Ethernet to connect to remote systems with FreeBSD.

Chapter 26, Electronic Mail

Explains the different components of an email server and dives into simple configuration topics for the most popular mail server software: **sendmail**.

Chapter 27, Network Servers

Provides detailed instructions and example configuration files to set up your FreeBSD machine as a network filesystem server, domain name server, network information system server, or time synchronization server.

Chapter 28, Firewalls

Explains the philosophy behind software-based firewalls and provides detailed information about the configuration of the different firewalls available for FreeBSD.

Chapter 29, Advanced Networking

Describes many networking topics, including sharing an Internet connection with other computers on your LAN, advanced routing topics, wireless networking, bluetooth, ATM, IPv6, and much more.

Appendix A, Obtaining FreeBSD

Lists different sources for obtaining FreeBSD media on CDROM or DVD as well as different sites on the Internet that allow you to download and install FreeBSD.

Appendix B, Bibliography

This book touches on many different subjects that may leave you hungry for a more detailed explanation. The bibliography lists many excellent books that are referenced in the text.

Appendix C, Resources on the Internet

Describes the many forums available for FreeBSD users to post questions and engage in technical conversations about FreeBSD.

Appendix D, PGP Keys

Lists the PGP fingerprints of several FreeBSD Developers.

本書的編排體裁

為方便閱讀本書，以下是一些本書所遵循的編排體裁：

文字編排體裁

斜體字(Italic)

斜體字型(Italic) 用於：檔名、目錄、網址(URL)、强調語氣、以及第一次提及的技術詞彙。

定寬字(Monospace)

定寬字(Monospace) 用於：錯誤訊息、指令、環境變數、port 名稱、主機名稱(hostname)、帳號、群組、設備(device)名稱、變數、程式碼等。

粗體字型(Bold)

以**粗體字**表示：應用程式、命令、按鍵。

使用者輸入

鍵盤輸入以**粗體字(Bold)** 表示，以便與一般文字做區隔。組合鍵是指同時按下一些按鍵，我們以'+' 來表示連接，像是：

Ctrl+**Alt**+**Del**

也就是說，一起按**Ctrl** 鍵、**Alt** 鍵，以及**Del** 鍵。

若要逐一按鍵，那麼會以逗號(,)來表示，像是：

Ctrl+**X**, **Ctrl**+**S**

也就是說：先同時按下**Ctrl** 與**X** 鍵，然後放開後再同時按**Ctrl** 與**S** 鍵。

舉個實例

下面例子以E:\> 為開頭的代表MS-DOS® 指令部分。若沒有特殊情況的話，這些指令應該是在Microsoft® Windows® 環境的“命令提示字元(Command Prompt)” 內執行。

```
E:\> tools\fdimage floppies\kern.flp A:
```

例子若是先以# 為開頭再接指令的話，就是指在FreeBSD 中以root 權限來下命令。你可以先以root登入系統並下指令，或是以你自己的帳號登入，並使用su(1) 來取得root 權限。

```
# dd if=kern.flp of=/dev/fd0
```

例子若是先以% 為開頭再接指令的話，就是指在FreeBSD 中以一般帳號來下命令即可。除非有提到其他用法，否則都是預設為C-shell(csh/tcsh) 語法，用來設定環境變數以及下其他指令的意思。

```
% top
```

致謝

您所看到的這本書是經過數百個分散在世界各地的人所努力而來的結果。無論他們只是糾正一些錯誤或提交完整的章節，所有的點滴貢獻都是非常寶貴有用的。

也有一些公司透過提供資金讓作者專注於撰稿、提供出版資金等模式來支持文件的寫作。其中，BSDi (之後併入Wind River Systems (http://www.windriver.com)) 資助FreeBSD 文件計劃成員來專職改善這本書直到2000年3 月第一版的出版。(ISBN 1-57176-241-8) Wind River Systems 同時資助其他作者來對輸出架構做很多改進，以及給文章增加一些附加章節。這項工作結束於2001 年11 月第二版。(ISBN 1-57176-303-1)
在2003-2004 兩年中，FreeBSD Mall (http://www.freebsdmall.com) 把報酬支付給改進這本手冊以使第三版印刷版本能夠出版的志工。

I. 開始使用FreeBSD

這部份是提供給初次使用FreeBSD 的使用者和系統管理者。這些章節包括：

- 介紹FreeBSD 給您。
- 在安裝過程給您指引。
- 教您UNIX 的基礎及原理。
- 展示給您看如何安裝豐富的FreeBSD 的應用軟體
- 向您介紹X，UNIX 的視窗系統以及詳細的桌面環境設定，讓您更有生產力。

我們試著儘可能的讓這段文字的參考連結數目降到最低，讓您在讀使用手冊的這部份時可以不太需要常常前後翻頁。

Chapter 1 簡介

Restructured, reorganized, and parts rewritten by Jim Mock.

1.1 概述

非常感謝您對FreeBSD 感興趣！以下章節涵蓋FreeBSD 計劃的各方面：比如它的歷史、目標、開發模式等等。

讀完這章，您將了解：

- FreeBSD 與其他OS 之間的關係；
- FreeBSD 計劃的歷史源流；
- FreeBSD 計劃的目標；
- FreeBSD open-source 開發模式的基礎概念；
- 當然囉，還有“FreeBSD” 這名字的由來。

1.2 Welcome to FreeBSD!

FreeBSD 是一個從4.4BSD-Lite 衍生出而能在以Intel (x86 and Itanium®), AMD64, Alpha™, Sun UltraSPARC® 為基礎的電腦上執行的作業系統。同時，移植到其他平台的工作也在進行中。對於本計劃歷史的介紹，請看FreeBSD 歷史源流，對於FreeBSD 的最新版本介紹，請看current release 。若打算對於FreeBSD 計劃有所貢獻的話(像是程式碼硬體設備、基金)，請看如何對FreeBSD 有貢獻 (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/contributing/index.html)。

1.2.1 FreeBSD 能做什麼？

FreeBSD 提供給你許多先進功能。這些功能包括：

- 動態優先權調整的『先佔式多工』能夠確保，即使在系統負擔很重的情況下，程式執行平順並且應用程式與使用者公平地共享資源。
- 『多人共用(multi-user)』代表著許多人可以同時使用一個FreeBSD 系統來處理各自的事務。系統的硬體周邊(如印表機及磁帶機)也可以讓所有的使用者適當地分享。也可以針對各別使用者或一群使用者的系統資源，予以設限，以保護系統不致被過度使用。
- 好用的『TCP/IP 網路功能』可支援許多業界標準，比如：SCTP、DHCP、NFS、NIS、PPP、SLIP、IPSec、IPv6 的支援，也就是說FreeBSD 可以容易地跟其他作業系統透過網路共同運作，或是當作企業的伺服器用途，例如提供遠端檔案共享(NFS)及電子郵件(email)等服務，或是讓您的企業連上網際網路(Internet)並提供WWW、FTP、路由(routing)、及防火牆(firewall、security) 等必備服務。
- 『記憶體保護(Memory protection)』能確保程式(或是使用者)不會互相干擾，即使任何程式有不正常的運作，都不會影響其他程式的執行。

- FreeBSD 是『32位元(32-bit)』的作業系統(在Alpha、Itanium、AMD64 及UltraSPARC 上則是『64位元(64-bit)』) 一打從一開始便是這樣設計的。
- 業界標準的『X Window 系統』(X11R7)可以在常見的便宜VGA 顯示卡/螢幕，提供了圖形化的使用者介面(GUI)，並且包括了完整的原始程式碼。
- 能『直接執行』許多其他作業系統(比如：Linux、SCO、SVR4、BSDI 和NetBSD) 的可執行檔。
- 數以萬計的立即可以執行的應用程式，這些都可透過FreeBSD 的『ports』及『packages』軟體管理機制來取得。不再需要費心到網路上到處搜尋所需要的軟體。
- 此外，網路上尚有可非常容易移植的數以萬計應用程式。FreeBSD 的原始程式碼與許多常見的商業版UNIX 系統都相容，所以大部分的程式都只需要很少的修改(或根本不用修改) ，就可以編譯執行。
- 需要時才置換(demand paged) *virtual memory* 及“merged VM/buffer cache” 的設計，這點在系統中有用去大量記憶體的程式執行時，仍然有不錯的效率表現。
- 支援CPU 的對稱多工處理(SMP)：可以支援多CPU 的電腦系統。
- 完全相容的*C*、*C++* 以及*Fortran* 的環境和其他開發工具。以及其他許多可供進階研發的程式語言也收集在ports 和packages。
- 整個系統都有『原始程式碼』，這讓你對作業環境擁有最完全的掌握度。既然能擁有完全開放的系統，何苦被特定封閉軟體所約束，任廠商擺佈呢？
- 廣泛且豐富的『線上文件』。
- *當然囉，還不止如此！*

FreeBSD 系統乃是基於美國加州大學柏克萊分校的電腦系統研究群(Computer Systems Research Group 也就是CSRG) 所發行的4.4BSD-Lite，以及基於BSD 系統開發的優良傳統。除了由CSRG 所提供的高品質的成果，為了提供可處理真正具負荷的工作，FreeBSD 計劃也投入了數千小時以上的細部調整，以能獲得最好的執行效率以及系統的穩定度。正當許多商業上的巨人正努力地希望能提供效能及穩定時，FreeBSD 已經具備這樣的特質-- *就是現在*！

FreeBSD 的運用範圍無限，其實完全限制在你的想像力上。從軟體的開發到工廠自動化，或是人造衛星上面的天線的方位角度的遠端控制；這些功能若可以用商用的Unix 產品來達成，那麼極有可能使用FreeBSD 也能辦到！FreeBSD 也受益於來自於全球各研究中心及大學所開發的數千個高品質的軟體，這些通常只需要花費很少的費用或根本就是免費的。當然也有商業軟體，而且出現的數目是與日俱增。

由於每個人都可以取得FreeBSD 的原始程式碼，這個系統可以被調整而能執行任何原本完全無法想像的功能或計劃，而對於從各廠商取得的作業系統通常沒有辦法這樣地被修改。以下提供一些人們使用FreeBSD 的例子：

- *網路服務：* FreeBSD 內建强勁的網路功能使它成為網路服務(如下例)的理想平台：
  - 檔案伺服器(FTP servers)
  - 全球資訊網伺服器(WWW servers) (標準的或更安全的[SSL] 連線)
  - IPv4 及IPv6 routing
  - 防火牆以及NAT (“IP masquerading”) gateways。
  - 電子郵件伺服器(Electronic Mail servers)
  - 網路新聞伺服器(USENET News) 或是電子佈告欄系統(BBS)
  - 還有更多...

有了FreeBSD，您可以容易地先用便宜的386 PC，再逐步升級您的機器到四個CPU 的Xeon 並使用磁碟陣列(RAID)來滿足您企業運用上的需求。

- *教育：* 若您是資工相關領域的學生，再也沒有比使用FreeBSD 能學到更多作業系統、計算機結構、及網路的方法了。另外如果你想利用電腦來處理一些*其他的* 工作，還有一些如CAD、數學運算以及圖形處理軟體等可以免費地取得使用。
- *研究：*有了完整的原始程式碼，FreeBSD 是研究作業系統及電腦科學的極佳環境。具有免費且自由取得特性的FreeBSD 也使得一個分置兩地的合作計劃，不必擔心版權及系統開放性的問題，而能自在的交流。
- *網路：* 你如果需要router、Name Server (DNS) 或安全的防火牆(Firewall)，FreeBSD 可以輕易的將你沒有用到的386 或486 PC 變身成為絶佳的伺服器，甚至具有過濾封包(packet-filter) 的功能。
- *X 視窗工作站：* FreeBSD 是X 終端機的良策，你可以使用免費的X11 Server。FreeBSD 不但可以充當遠端X 程式終端機，也可以執行本地的X 程式而減輕大型工作站的負荷。如果有一台中央伺服器的話，FreeBSD 甚至可以經由網路開機(不需硬碟，也就是“diskless”)，而變成更便宜且易於管理的工作站。
- *軟體開發：* 基本安裝的FreeBSD 就包含了完整的程式開發工具，如GNU C/C++ 編譯器及除錯器。

你可以經由燒錄CD-ROM、DVD 或是從FTP 站上抓回FreeBSD -- 包括立即可執行的系統以及系統的完整程式碼。詳情請參閱Appendix A 取得FreeBSD。

1.2.2 誰在用FreeBSD？

許多Internet 上的大型網站都是以FreeBSD 作為它的作業系統，例如：

- Yahoo! (http://www.yahoo.com/)
- Apache (http://www.apache.org/)
- Blue Mountain Arts (http://www.bluemountain.com/)
- Pair Networks (http://www.pair.com/)
- Sony Japan (http://www.sony.co.jp/)
- Netcraft (http://www.netcraft.com/)
- Weathernews (http://www.wni.com/)
- Supervalu (http://www.supervalu.com/)
- TELEHOUSE America (http://www.telehouse.com/)
- Sophos Anti-Virus (http://www.sophos.com/)
- JMA Wired (http://www.jmawired.com/)

以及許多其他的網站。

1.3 關於FreeBSD 計劃

接下來講的是FreeBSD 計劃的背景，包含歷史源流的簡介、計劃的目標，以及開發的模式。

1.3.1 FreeBSD 歷史源流的簡介

Contributed by Jordan Hubbard.

FreeBSD 計畫的想法是在1993 年初所形成的，那是源自於維護一組『非官方386BSD 的patchkit(修正工具)』計劃的三個協調維護人Nate Williams，Rod Grimes 和我(Jordan Hubbard)。

我們最初的目標是做出一份386BSD 綜合修正的snapshot 版，以便修正當時一堆patchkit 都不容易解決的問題。有些人可能還記得早期的計劃名稱叫做“386BSD 0.5” 或“386BSD Interim” 就是這個原因。

386BSD 是Bill Jolitz 的作業系統，在當時就已有約一年的分裂討論。當該修正工具(patchkit) 日漸龐雜得令人不舒服，我們無異議地同意要作一些事了，並決定提供一份臨時性的“淨化版(cleanup)” 來幫助Bill。然而，由於Bill Jolitz 忽然決定取消其對該計劃的認可，且沒有明確指出未來的打算，所以該計劃便突然面臨斷炊危機。

不久我們便決定在即使沒有Bill 的支持下，讓該計劃仍然繼續下去，最後我們採用David Greenman 丟銅板決定的名字，也就是『FreeBSD』。在詢問了當時的一些使用者意見之後，就開始決定了最初的目標，當該計劃開始實施一切就要成真時，一切就變得更清楚了。我跟Walnut Creek CD-ROM 討論發行CD-ROM 這樣子不便上網的人就可以用比較簡單的方式取得FreeBSD。Walnut Creek CD-ROM 不只贊成以CD-ROM 來發行FreeBSD 的想法，同時提供了一台機器以及快速的網際網路的頻寬。如果不是Walnut Creek CD-ROM 幾乎是空前的信任這個剛開始還是完全默默無聞的計劃，那麼很可能FreeBSD 不會如此快速的成長到今日這樣的規模。

第一張以CD-ROM (及網路)發行的FreeBSD 1.0 是在1993 年十二月。該版本是基於由U.C. Berkeley 以磁帶方式發行的4.3BSD-Lite (“Net/2”)以及許多來自於386BSD 和自由軟體基金會的軟體。對於第一次發行而言還算成功，我們又接著於1994 年5 月發行了相當成功的FreeBSD 1.1。

然而此後不久，另一個意外的風暴在Novell 和U.C. Berkeley 關於Berkeley Net/2 磁帶之法律地位的訴訟確定之後形成。U.C. Berkeley 接受大部份的Net/2 的程式碼都是『侵佔來的』且是屬於Novell 的財產-- 事實上是當時不久前從AT&T 取得的。Berkeley 得到的是Novell 對於4.4BSD-Lite 的『祝福』，最後當4.4BSD-Lite 終於發行之後，便不再是侵佔行為。而所有現有Net/2 使用者都被强烈建議更換新版本，這包括了FreeBSD。於是，我們被要求於1994 年6 月底前停止散佈基於Net/2 的產品。在此前提之下，本計劃被允許在期限以前作最後一次發行，也就是FreeBSD 1.1.5.1。

FreeBSD 便開始了這宛如『重新發明輪子』的艱鉅工作-- 從全新的且不完整的4.4BSD-Lite 重新整合。這個“Lite” 版本是不完整的，因為Berkeley 的CSRG 已經刪除了大量在建立一個可以開機執行的系統所需要的程式碼(基於若干法律上的要求)，且該版本在Intel 平台的移植是非常不完整的。直到1994 年11 月本計劃才完成了這個轉移，同時在該年12 月底以CD-ROM 以及網路的形式發行了FreeBSD 2.0。雖然該份版本在當時有點匆促粗糙，但仍是富有意義的成功。隨之於1995 年6 月又發行了更容易安裝，更好的FreeBSD 2.0.5。

我們在1996 年8 月發行了FreeBSD 2.1.5，在ISP 和商業團體中非常流行。隨後，2.1-STABLE 分支的另一個版本應運而生，它就是在1997 年2 月發行FreeBSD 2.1.7.1 ，同時也是2.1-STABLE 分支的最後版。之後此分支便進入維護狀態，僅僅提供安全性的加强和其他嚴重錯誤修補的維護(RELENG_2_1_0)。

1996 年11 月FreeBSD 2.2 從開發主軸分支(“-CURRENT”) 出來成為RELENG_2_2 分支。它的第一個完整版(2.2.1)於1997 年4 月發行。2.2 分支的延續版本在97 年夏秋之間發行的，其最後版是在1998 年11 月發行的2.2.8 版。第一個正式的3.0 版本在1998 年10 月發行，亦即宣告2.2 分支的落幕。

1999/01/20 日再度分支，這產生了4.0-CURRENT 以及3.X-STABLE 兩個分支。3.X-STABLE 方面，3.1 發行於1999/02/15，3.2 發行於1999/05/15，3.3 發行於1999/09/16，3.4 發行於1999/12/20，3.5 發行於2000/06/24，接下來幾天後發佈了一些的修補檔(對Kerberos 安全性方面的修正)，就升級至3.5.1 ，這是3.X 分支最後一個發行版本。

在2000/03/13 又有了一個新的分支，也就是4.X-STABLE 。這個分支之後發佈了許多的發行版本：4.0-RELEASE 在2000 年3 月發行，而最後的4.11-RELEASE 則在2005 年1 月發行。4-STABLE 分支的支援會持續到2007/01/31 ，但主要焦點在於安全方面的漏洞、臭蟲及其他嚴重問題的修補。

期待已久的5.0-RELEASE 在2003/01/19 正式發行。這是將近開發三年的巔峰之作，同時也開始加强多顆CPU(SMPng)的支援、kernel thread(KSE) 的支援、檔案系統採用UFS2 以及支援snapshot 等，並支援UltraSPARC 和`ia64` 平台、支援藍芽、32 bit 的PCMCIA 等。之後於2003 年6 月發行了5.1。而-CURRENT 這個發展主軸分支的最後5.X 版本是在2004 年2 月正式發行的5.2.1-RELEASE，在5.X 系列進入-STABLE (RELENG_5分支)之後，-CURRENT 就轉移為6.X 系列。

RELENG_5 分支於2004 年8 月正式開跑，之後是5.3-RELEASE ，它是5-STABLE 分支的第一個發行版本。最後的5.5-RELEASE 是在2006 年5 月發行的，在此之後RELENG_5 分支不再繼續。

RELENG_6 分支於2005 年7 月開跑，而6.X 分支的第一個release(6.0-RELEASE) 是在2005 年11 月出的。最新的7.2-RELEASE 是在May 2009 發行。當然囉，RELENG_6 分支還將有後續的發行版。

RELENG_7 分支於2007 年10 月開跑，最新的8.0-RELEASE 是在Oct 2009 發行。RELENG_7 分支還將有後續的發行版。

目前，長期的開發計畫繼續在8.X-CURRENT (trunk) 分支中進行，而8.X 的CD-ROM (當然，也可以用網路抓) snapshot 版本可以在FreeBSD snapshot server (ftp://current.FreeBSD.org/pub/FreeBSD/snapshots/) 取得。

1.3.2 FreeBSD 計劃的目標

Contributed by Jordan Hubbard.

FreeBSD 計劃的目標在於提供可作任意用途的軟體而不附帶任何限制條文。我們之中許多人對程式碼(以及計畫本身) 都有非常大的投入，因此，當然不介意偶爾有一些資金上的補償，但我們並沒打算堅決地要求得到這類資助。我們認為我們的首要『使命(mission)』是為任何人提供程式碼，不管他們打算用這些程式碼做什麼，因為這樣程式碼將能夠被更廣泛地使用，從而發揮其價值。我認為這是自由軟體最基本的，同時也是我們所倡導的一個目標。

我們程式碼樹中，有若干是以GNU GPL 或者LGPL 發佈的那些程式碼帶有少許的附加限制，還好只是强制性的要求開放程式碼而不是別的。由於使用GPL 的軟體在商業用途上會增加若干複雜性，因此，如果可以選擇的話，我們會比較喜歡使用限制相對更寬鬆的BSD 版權來發佈軟體。

1.3.3 FreeBSD 的開發模式

Contributed by Satoshi Asami.

FreeBSD 的開發是一個非常開放且具彈性的過程，就像從貢獻者名單(http://www.FreeBSD.org/doc/zh_TW.Big5/articles/contributors/article.html) 所看到的，是由全世界成千上萬的貢獻者發展起來的。FreeBSD 的開發基礎架構允許數以百計的開發者透過網際網路協同工作。我們也經常關注著那些對我們的計畫感興趣的新開發者和新的創意，那些有興趣更進一步參與計劃的人只需要在FreeBSD technical discussions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers) 連繫我們。FreeBSD announcements 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce) 對那些希望了解我們進度的人也是相當有用的。

無論是單獨開發者或者封閉式的團隊合作，多瞭解FreeBSD 計劃和它的開發過程會是不錯的：

The SVN and CVS repository

過去數年來FreeBSD 的中央source tree 一直是以CVS (http://ximbiot.com/cvs/wiki/) (Concurrent Versions System) 來維護的，它是個自由軟體，可用來做為版本控制，一裝完FreeBSD 內就有附了。然而在2008 年6 月起，FreeBSD 版本控制系統改用SVN (http://subversion.tigris.org)(Subversion)。這切換動作我們認為是有必要，因為**CVS** 先天的技術限制，導致source tree 以及歷史版本數量不斷快速擴張。因此，主要的repository 目前是採用**SVN** ，而client 端的工具像是**CVSup**、**csup** 都是以舊式的**CVS** 架構為基礎，仍可以繼續正常運作——此乃因**SVN** repository 有backport 回**CVS** 才可以繼續讓client 端相容。目前，就只有中央source tree 是採**SVN** 版本控制方式。而文件、網頁、Ports 這些repository 仍持續使用**CVS** 版本控制方式。而主要的CVS repository (http://www.FreeBSD.org/cgi/cvsweb.cgi) 是位於美國加州Santa Clara 的某台機器上，然後再mirror 到世界上其他的許多機器上。**SVN** tree 內有兩個主分支：-CURRENT 以及-STABLE ，這些都可輕鬆複製到自己機器上。詳情請參閱更新你的source tree 一節。

The committers list

所謂的*committers* 指的是對CVS tree 有*write* 權限，並依不同授權部分，而有不同權限可修改FreeBSD source。(“committer” 這詞源自cvs(1) 中的`commit` 指令，該指令是用來把新的修改提交給CVS repository。) 而提交修改給committer 們檢查的最好方式，就是用send-pr(1) 指令。若提交PR 的流程系統上有壅塞現象的話，也可以改用寄信方式，寄信到FreeBSD committer's 郵遞論壇即可。

The FreeBSD core team

如果把FreeBSD 看成是一家公司的話，*FreeBSD core team* 就相當於『董事會(board of directors)』。core team 的主要職責在於確保此計劃有良好的架構，以朝著正確的方向發展。此外，邀請熱血且負責的軟體開發者加入committers 行列，以在若干成員離去時得以補充新血。目前的core team 是在2008 年7 月committers 候選人中選出來的，每兩年會舉辦一次選舉。

有些core team 成員還負責某些特定範圍，也就是說他們必須盡量確保一些子系統的穩定、效能。關於FreeBSD 開發者們以及各自責任範圍，請參閱貢獻者名單 (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/contributors/article.html)。

> **Note:** core team 大部分成員加入FreeBSD 開發都是志工性質而已，並未從本計劃中獲得任何薪酬，所以不該把“commitment” 誤解為“guaranteed support” 才對。剛前面所講的『董事會』可能是不恰當的類推，或許我們應該說：他們是一群自願放棄原本的優渥生活、個人其他領域成就，而選擇投入FreeBSD 開發的熱血有為者才對！

其他的貢獻者

最後一點，但這點絕非最不重要的，最大的開發者團隊就是持續為我們提供回饋以及錯誤修正的使用者自己。與FreeBSD 非核心開發者互動的主要方式，便是透過訂閱FreeBSD technical discussions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers) 來進行溝通，這方面可參考，請參閱Appendix C 以瞭解各式不同的FreeBSD 郵遞論壇(mailing lists)。

FreeBSD 貢獻者名單 (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/contributors/article.html) 相當長且不斷成長中，只要有貢獻就會被列入其中，要不要立即考慮貢獻FreeBSD 一些回饋呢？

然而，提供原始碼並非為這個計劃做貢獻的唯一方式；還需要大家投入的完整工作列表、說明，請參閱FreeBSD 官網 (http://www.FreeBSD.org/index.html)。

簡單的說，我們的開發模式就像是一組沒有拘束的同心圓。這種集中開發模式是以*給使用者方便*為主，同時讓他們能很容易地共同維護軟體，而不會把潛在的貢獻者排除在外！我們的目標是提供含有大量一致性的應用軟體(ports/packages)，以便讓使用者輕鬆安裝、使用的作業系統——而這開發模式相當符合此一目標。

我們對於那些想要加入FreeBSD 開發者的期待是：請保持如同前人一樣的投入，以確保繼續成功！

1.3.4 最新的FreeBSD 發行版本

FreeBSD 是免費使用且帶有完整原始程式碼的以4.4BSD-Lite 為基礎的系統，可以在Intel i386™, i486™, Pentium®, Pentium Pro, Celeron®, Pentium II, Pentium III, Pentium 4 (或者相容型號), Xeon™, DEC Alpha 和Sun UltraSPARC 為基礎的電腦上執行的作業系統。它主要以加州大學巴爾克利分校的CSRG 研究小組的軟體為基礎，並加入了NetBSD、OpenBSD、386BSD 以及自由軟體基金會的一些東西。

自從1994 年末，我們發佈了FreeBSD 2.0 之後，系統的執行效率、功能、穩定性都有了令人注目的提升。最大的改變就是我們將記憶體與檔案系統的cache 機制結合在一起。這不只使得系統的表現變得更好, 並且使得FreeBSD 系統最少的記憶體需求減少到5 MB。其它的改進包括完整的NIS cilent and server 功能支援，支援transaction TCP、PPP 撥接連線、整合的DHCP 支援、SCSI 子系統的改進、ISDN 的支援，ATM、FDDI 以及乙太網路(Ethernet、包括100 Mbit 和Gigabit) 的支援，提升了最新的Adaptec 控制卡驅動程式的改善，以及數以千計的bug 修正。

除了最基本的系統軟體，FreeBSD 還提供了廣受歡迎的套件軟體管理機制：Ports Collection。到本書付印時，已有超過20,000 個ports，這範疇涵蓋從http(WWW) 伺服器到遊戲、程式語言、編輯器以及您能想到的幾乎所有的東西。完整的Ports Collection 需要約417 MB 的硬碟空間，除了port 基本架構檔案外，都只儲存與該port 軟體的原始碼有『須要變更』的部份。如此一來，我們可以更容易更新這些ports，也大量的減少如舊的1.0 版Ports Collection 對於硬碟空間的需求。要安裝一個port 的話，只需要進入該port 的目錄，輸入`make install`，這樣子系統就會幫你裝好了。您要編譯的每個程式的完整原始程式，都可從FTP 或CD-ROM 中獲得，所以您只需準備足夠的硬碟空間來編譯你要的port 軟體。幾乎每一個port 都有已事先編譯好的“package”以方便安裝，如果不想從編譯port 的人，只要用個簡單指令(`pkg_add`)就可以安裝。有關packages 和ports 的細節，可以參閱Chapter 4。

FreeBSD 主機的`/usr/share/doc` 目錄下找到許多有用的文件，來幫助您安裝、使用FreeBSD。也可以使用下面的網址，以瀏覽器來翻閱本機上安裝的手冊：

FreeBSD 使用手冊

```
/usr/share/doc/handbook/index.html
```

FreeBSD 常見問答集

```
/usr/share/doc/faq/index.html
```

此外，可在下列網址找到最新版(也是更新最頻繁的版本)：http://www.FreeBSD.org/。

Chapter 2 安裝FreeBSD

Restructured, reorganized, and parts rewritten by Jim Mock. The sysinstall walkthrough, screenshots, and general copy by Randy Pratt.

2.1 概述

FreeBSD 提供一個簡單好用的文字介面安裝程式，叫做**sysinstall**。這是FreeBSD 預設使用的安裝程式。協力廠商若有意願的話，也可以改用自己的安裝程式。本章將說明如何使用**sysinstall** 來安裝FreeBSD。

讀完這章，您將了解：

· 如何製作FreeBSD 安裝片

· FreeBSD 對硬碟的使用及配置。

· 如何啓動**sysinstall** 程式。

· 在執行**sysinstall** 時會問的相關問題有哪些、這些問題的意思為何、以及該如何回答。

在開始閱讀這章之前，您需要：

· 閱讀要安裝的FreeBSD 版本所附之硬體支援表，以確定您的硬體有沒有被支援。

> **Note:** 一般來說，此安裝說明是針對i386 (相容的PC 機種) 架構的電腦。如果有其他架構(比如Alpha)的安裝說明，我們會一併列出。雖然本文件會常常更新，但有可能與您安裝版本上所附的說明文件有些許出入。在此，我們建議您把本說明文章當作一般的安裝參考原則就好。

2.2 硬體需求

2.2.1 最低需求

安裝FreeBSD 的硬體方面最低需求，依各FreeBSD 版本與硬體架構差別而有所不同。

關於安裝所需的最低需求，可在FreeBSD 網站的Release Information (http://www.FreeBSD.org/releases/index.html) 找相關的Installation Notes 說明。接下來的章節會有相關說明整理。根據安裝FreeBSD 的方式不同，可能會需要軟碟機或光碟機，或某些情況則是要網路卡。這些部份會在Section 2.3.7 有介紹。

2.2.1.1 FreeBSD/i386 及FreeBSD/pc98 架構

FreeBSD/i386 及FreeBSD/pc98 兩種版本均須486 或更好的處理器，以及至少24 MB 的RAM、至少150 MB 的硬碟空間，才能進行最小安裝。

> **Note:** 對老舊硬體而言，在大部份情況裝更多的RAM 與更大的硬碟空間，會比使用更快的CPU 更有用。

2.2.1.2 FreeBSD/alpha 架構

若要裝FreeBSD/alpha，則需確認該機型是否有支援(請參閱Section 2.2.2) 且必須整顆硬碟皆給FreeBSD 使用。目前無法同時與其他作業系統共存。這硬碟須接到SRM 韌體有支援的SCSI controller 上，或者IDE 硬碟(該機型的SRM 有支援可從IDE 硬碟開機)。

此外，還需該機型的SRM console firmware。有些機型可以選擇AlphaBIOS (or ARC) firmware 或SRM 來用。若沒有的話，則需從硬體廠商的網站去下載新的韌體。

Note: 從FreeBSD 7.0 就不再支援Alpha。FreeBSD 6.*x* 系列則是此架構的最後支援。

2.2.1.3 FreeBSD/amd64 架構

有兩種CPU 能跑FreeBSD/amd64。第一種是包括AMD Athlon™64 、AMD Athlon64-FX、AMD Opteron™ 或更好的CPU。

第二種則是Intel® EM64T 架構的CPU，這些也可以用FreeBSD/amd64。這些CPU 包括了Intel Core™ 2 Duo 、Quad、Extreme 系列以及Intel Xeon 3000、5000、7000 相關系列的CPU。

若主機板晶片組為nVidia nForce3 Pro-150，則*必須* 調整BIOS 設定，將IO APIC 停用才行。若找不到這選項，那可能就是找ACPI 停用。因為Pro-150 晶片組有個bug，目前我們尚無找到堪解之道。

2.2.1.4 FreeBSD/sparc64 架構

若要裝FreeBSD/sparc64，則需確認該機型是否有支援(請參閱Section 2.2.2)。

FreeBSD/sparc64 必須使用整顆硬碟，因為無法同時與其他作業系統共存。

2.2.2 有支援的硬體

FreeBSD 每次release 時都會有附上FreeBSD Hardware Notes 來說明有支援的硬體列表。通常這份文件可在光碟或FTP 的最上層目錄找到，也就是名為`HARDWARE.TXT` 的檔案。此外，在**sysinstall** 的documentation 選項內也可以看到。每次FreeBSD release 時該列表會依各不同架構，而列出相關已知有支援的硬體。在FreeBSD 網站的Release Information (http://www.FreeBSD.org/releases/index.html) 頁可以找到各不同release 版本與各架構上的硬體支援列表。

2.3 安裝前的準備工作

2.3.1 列出您電腦的硬體清單

在安裝FreeBSD 之前，您應該試著將您電腦中的硬體清單列出來。FreeBSD 安裝程式會將這些硬體(硬碟、網路卡、光碟機等等) 以型號及製造廠商列出來。FreeBSD 也會嘗試為這些硬體找出最適當的IRQ 及IO port 的設定。但是因為PC 的硬體種類實在太過複雜，這個步驟不一定保證絕對成功。這時，您就可能需要手動更改有問題的設定值哩。

如果您已裝了其它的作業系統，如：Windows 或Linux，那麼可先由這些系統所提供的工具，來查看這些硬體設定值是怎麼設定的。若真的沒辦法確定某些卡用什麼設定值，那麼可以檢查看看卡上面所標示的東西，說不定它的設定已有標示在卡上。常用的IRQ 號碼為3、5 以及7；而IO 埠的值通常以16 進位表示，例如0x330。

建議您在安裝FreeBSD 之前，把這些資料列印或抄錄下來做成表格，也許會較有用喔，例如：

Table 2-1. 硬體清單(舉例)

硬體名稱	IRQ	IO port(s)	備註
第一顆IDE 硬碟	N/A	N/A	40 GB，Seagate 製造，接在第一條IDE 排線的master
CDROM	N/A	N/A	接在第一條IDE 排線的slave
第二顆硬碟	N/A	N/A	20 GB，IBM 製造，接在第二條IDE 排線的master
第一個IDE controller	14	0x1f0	
網路卡	N/A	N/A	Intel 10/100
數據機	N/A	N/A	3Com® 56K faxmodem，接在COM1
…			

硬體清單完成之後，就需針對你所要裝的FreeBSD 版本之硬體需求，來檢查是否有支援。

2.3.2 備份您的資料

如果要裝的電腦上面存有重要資料，那麼在安裝FreeBSD 前，請確定您已經將這些資料備份了，並且先測試過這些備份檔是否沒有問題。FreeBSD 安裝程式在要寫入任何資料到您的硬碟前，都會先提醒您確認，一旦您確定要寫入，那麼之後就再也沒有反悔的機會囉。

2.3.3 決定要將FreeBSD 安裝到哪裡

如果您想讓FreeBSD 直接使用整顆硬碟，那麼請直接跳到下一節。

然而，如果您想要FreeBSD 跟既有的系統並存，那麼，您必須對硬碟的資料分佈方式有深入的了解，以及其所造成的影響。

2.3.3.1 FreeBSD/i386 架構的硬碟配置模式

PC 上的硬碟可以被細分為許多分散區(chunk)。這些區域叫做*分割區(Partitions)*。由於FreeBSD 內部也有partition，名稱可能很容易造成混淆，因此通常在FreeBSD 這邊會稱呼這些磁碟分散區為disk slices 或簡稱slices。舉例來說，FreeBSD 的`fdisk` 的對象是針對PC 硬碟的slice 而非partition。因為PC 本身先天設計，每個硬碟最多可以有4 個分割區，而這些分割叫做*主要分割區(Primary Partitions)*。為了突破這個限制，以便能使用更多的分割區，就有了新的分割區類型，叫作：*延伸分割區(Extended Partition)*。每個硬碟就只能有一個延伸分割區。然而，在延伸分割區裡面可以建立許多個特殊分割區，叫作*邏輯分割區(Logical Partitions)*。

每種分割區都有其*分割區代號(Partition ID)* 用以區別每種分割區的資料類型。而FreeBSD 分割區代號是`165`。

一般來講，每種作業系統都會有自己獨特的方式來區別分割區。舉例：DOS 及其之後的作業系統，比如Windows 會分配給每個主要分割區及邏輯分割區1 個*磁碟代號(drive letter)*，從`C:` 開始。

FreeBSD 必須安裝在主要分割區。FreeBSD 可以在這個分割區上面存放資料或是您建立的任何檔案。然而，如果您有很多顆硬碟，也可以在這些(或部份)硬碟建立FreeBSD 分割區。安裝FreeBSD 的時候，必須至少要有1 個分割區給FreeBSD 使用，這個分割區可以是尚未使用的分割區，或是現存的分割區(但上面的資料不打算繼續使用)。

如果已經用完了磁碟上所有的分割區，那麼您必須使用其他作業系統所提供的工具(像是DOS or Windows 上的`fdisk`) 來騰出一個分割區給FreeBSD 用。

如果有多餘的分割區，也可以使用它。但使用前，您可能需要先整理一下這些分割區。

FreeBSD最小安裝需要約100 MB 的空間，但是這只是『最小安裝』，幾乎沒剩下多少空間來存放您自己的檔案。較理想的(不含圖形介面)最小安裝是約250 MB，或者是350 MB 左右(包含圖形介面)。還需要安裝其他的套件軟體，那麼將需要更多的硬碟空間。

您可以使用商業軟體像是**PartitionMagic®** 或免費自由工具像是**GParted** 來重新調整分割區空間，來給FreeBSD 用的空間。FreeBSD 光碟、FTP 上面的`tools` 目錄包含兩個免費的工具，也可以達成這個工作，叫作：**FIPS** 及**PResizer**。這些工具的說明文件可以在同個目錄內找到。**FIPS**、**PResizer** 和**PartitionMagic** 可以重新調整在MS-DOS 到Windows ME 所使用的FAT16 及FAT32 分割區大小。目前已知可更改NTFS 分割區的有**PartitionMagic** 及**GParted** 這兩種工具程式。**GParted**在許多Linux distributions 的Live CD 都有提供，像是SystemRescueCD (http://www.sysresccd.org/)。

目前已知Microsoft Vista 分割區的重新調整大小會有問題。在做上述類似動作時，請記得手邊要有Vista 安裝光碟以免萬一。此外，強烈建議先做磁碟維護，以及現有資料備份。

> **Warning:** 不當的使用這些工具，可能會刪除所有硬碟上的資料。在使用這些工具前，請確定您已有先備份好資料。

Example 2-1. 使用現有的分割區

假設您只有一個4 GB 的硬碟，而且已經裝了Windows ，然後將這顆硬碟分成兩個磁碟代號：`C:` 及`D:`，每個大小為2 GB 。`C:` 槽上放了1 GB 的資料，而`D:` 槽上放了0.5 GB 的資料。

這表示硬碟上有兩個分割區，每個磁碟代號槽都是分割區。您可以把所有放在`D:` 的資料，都移動到`C:`，這樣就空出了第二個分割區可以給FreeBSD 使用。

Example 2-2. 縮減現有的分割區

假設您只有一個4 GB 硬碟，而且已經裝了Windows。在安裝Windows 時把4 GB 都給`C:` 槽，並且現在已經用了1.5 GB 空間，而你想將剩下空間的2 GB 給FreeBSD 使用。

如此一來，為了裝FreeBSD，你必須在以下兩種方式二選一：

1. 備份Windows 資料，然後重裝Windows，並在安裝Windows 時給2 GB 的分割空間。
2. 使用上述的工具，像是**PartitionMagic**，來重新調整Windows 所用的分割區大小。

2.3.3.2 Alpha 架構的磁碟配置模式

在Alpha 上，您必須使用一整顆硬碟給FreeBSD，沒有辦法在同顆硬碟上跟其他作業系統共存。依不同型號的Alpha 機器，您的硬碟可以是SCSI 或IDE 硬碟，只要您的機器可以從這些硬碟開機就可以。

按照Digital / Compaq 使用手冊的編排風格，所有SRM 輸入的部分都用大寫表示。注意：SRM 大小寫有別。

要得知您磁碟的名稱以及型號，可以在SRM console 提示下使用`SHOW DEVICE` 命令：

```
>>>SHOW DEVICE
dka0.0.0.4.0               DKA0           TOSHIBA CD-ROM XM-57  3476
dkc0.0.0.1009.0            DKC0                      RZ1BB-BS  0658
dkc100.1.0.1009.0          DKC100                 SEAGATE ST34501W  0015
dva0.0.0.0.1               DVA0
ewa0.0.0.3.0               EWA0              00-00-F8-75-6D-01
pkc0.7.0.1009.0            PKC0                  SCSI Bus ID 7  5.27
pqa0.0.0.4.0               PQA0                       PCI EIDE
pqb0.0.1.4.0               PQB0                       PCI EIDE
```

例子中機器為Digital Personal Workstation 433au，並且顯示出此機器有連接三個磁碟機。第一個是CDROM，叫做`DKA0` ；另外兩個是磁碟機，分別叫做：`DKC0` 及`DKC100`。

磁碟機的名稱中有`DKx` 字樣的是SCSI 硬碟。例如：`DKA100` 表示是SCSI 硬碟，其SCSI ID 為1，位在第一個SCSI 匯流排(A)；而`DKC300` 表示是SCSI 硬碟，其SCSI ID 為3，位於第三個SCSI 匯流排(C)。裝置名稱`PKx` 則為SCSI 控制卡。由上述`SHOW DEVICE` 的結果看來，SCSI 光碟機也被視為是SCSI 硬碟的一種。

若為IDE 硬碟的話，名稱會有`DQx` 字樣，而`PQx` 則表示相對應的IDE 磁碟控制器。

2.3.4 整理你的網路設定資料

如果想透過網路(FTP 站或NFS)安裝FreeBSD，那麼就必須知道您的網路設定。在安裝FreeBSD 的過程中將會提示您輸入這些資訊，以順利完成安裝過程。

2.3.4.1 使用乙太網路(Ethernet)或Cable/DSL 數據機上網

若使用乙太網路，或是要透過Cable/DSL 數據機上網，那麼您必須準備下面的資訊：

1. IP 位址
2. 預設Gateway(閘道) 的IP 位址
3. Hostname(機器名稱)
4. DNS 伺服器的IP 位址
5. Subnet Mask

若不知道這些資訊，您可以詢問系統管理者或是您的ISP 業者。他們可能會說這些資訊會由*DHCP* 自動指派；如果是這樣的話，請記住這一點就可以了。

2.3.4.2 使用數據機上網

若由一般的數據機撥接上網，您仍然可以安裝FreeBSD，只是會需要很長的時間。

您必須知道：

1. 撥接到ISP 的電話號碼。
2. 您的數據機是連到哪個COM 埠。
3. 您撥接到ISP 所用的帳號跟密碼。

2.3.5 查閱FreeBSD 勘誤表(Errata)

雖然我們盡力使得每個FreeBSD 發行版本都很穩定，但是過程中仍然不免有時會發生錯誤。在某些很罕見的情形下，這些錯誤會影響到安裝的過程。當發現這些錯誤且修正後，會將它們列在 FreeBSD 勘誤表(Errata) (http://www.FreeBSD.org/releases/8.0R/errata.html) 中。在您安裝FreeBSD 前，應該先看看勘誤表中有沒有什麼問題會影響到您的安裝。

關於所有發行版本的資訊(包括勘誤表)，可以在FreeBSD 網站 (http://www.FreeBSD.org/index.html) 的發行情報(release information) (http://www.FreeBSD.org/releases/index.html) 找到。

2.3.6 準備好FreeBSD 安裝檔案

FreeBSD 可以透過下面任何一種安裝來源進行安裝：

Local Media

· CDROM 或DVD

· 現有的DOS 分割區

· SCSI 或QIC 磁帶。

· 軟碟磁片

Network

· FTP 站、支援Passvie 模式的FTP 站(若您機器在NAT 內) 、甚至HTTP proxy 都可以。

· NFS 伺服器

· 專用(dedicated)的parallel 或serial 連線

若已經有FreeBSD 的CD 或DVD，但機器不支援從光碟開機的話，那麼請直接進下一節(Section 2.3.7)。

若沒有FreeBSD 安裝片的話，那麼請先看Section 2.13 這裡會介紹如何準備所需要的安裝片，照該節步驟弄好後，就可以繼續下一步Section 2.4。

2.3.7 準備好開機磁片

FreeBSD 安裝流程是要從電腦開機後，進入FreeBSD 安裝畫面——而不是在其他作業系統上執行程式。一般來講，電腦都是用裝在硬碟上的作業系統來開機，也可以用開機磁片來開機；此外，現在大多數電腦都可以從光碟開機。

Tip: 如果您有FreeBSD 的CDROM 或DVD(無論是用買現成的或是自己燒錄的)，且您的電腦可支援由光碟開機，(通常在BIOS 中會有“Boot Order” 或類似選項)，那麼您就可以跳過此小節。因為FreeBSD CDROM 或DVD 都可以用來開機。

請按照下面步驟，以製作開機片：

1. 取得開機片的映像檔(images)

   開機磁片用的映像檔(images)通常會放在光碟片上的`floppies/` 目錄內，另外也可以從像是下面FTP 站的floppies 目錄下
   載：`ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/<arch>/<version>-RELEASE/floppies/` 。
   請將『arch』、『version』替換為打算安裝的電腦架構、OS 版本。例如：想裝的
   是FreeBSD/i386 8.0-RELEASE ，那麼可以
   到 (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.0-RELEASE/floppies/) 下載。

   映像檔(images)的附檔名都是`.flp`。而`floppies/` 目錄內包含一些不同用途的映像檔(images)，這取決於您要裝的FreeBSD 版本、需求、硬體配備為何。通常要4 個映像檔，也就
   是：`boot.flp`、`kern1.flp`、`kern2.flp`、`kern3.flp`。若有疑問的話，請翻閱同一目錄下
   的`README.TXT` 文件檔，以瞭解相關最新注意事項。

   Important: 在使用FTP 下載時，必須使用*binary 模式* 進行傳輸。有些瀏覽器預設是以*text* (或*ASCII*) 模式來傳輸資料，所以這些錯誤傳輸模式下載的映像檔所做成的磁片，會無法使用。

2. 準備開機磁片

   每個映像檔都需要一張磁片，並且請避免使用到壞的磁片。最簡單的檢測方式就是自己先把這些磁片再重新格式化(format) 而不要相信所謂的已格式化的磁片，Windows 內的format 在格式化時，並不會告訴你是否有壞軌，而只會直接將它們標示壞軌而不使用壞軌部分而已。此外，建議採用全新的磁片來製作安裝片比較保險。

   Important: 若在安裝FreeBSD 的過程中發生當機、畫面凍結或是其他怪異的現象，首先要懷疑的就是開機磁片是否壞掉。請用其他的磁片製作映像檔再試試看。

3. 將映像檔(images)寫入到磁片內

   `.flp` 檔並非一般檔案，不能直接把它複製到磁片上。事實上它是包含整張磁片所有內容的映像檔(image)。也就是說，不能純粹複製檔案到磁片上，而必須使用特別的工具程式，來將映像檔直接寫到磁片上。

   若要用MS-DOS/Windows 來作安裝片的話，那麼可以用`fdimage` 工具程式來將映像檔，寫到磁片上。

   若您用的是FreeBSD 光碟的話(假設光碟機代號為`E:` ，那麼請執行類似下面的指令：

   ```
   E:\> tools\fdimage floppies\boot.flp A:
   ```

   請針對每個需要用到的`.flp` 映像檔，重複上述的指令(記得更改相關檔名)，每次的映像檔完成後，都需要換另外一片來裝新的映像檔；請記得：在作好的磁片上註明是使用哪個映像檔作的。若`.flp` 映像檔放在不同地方，請自行修改上述指令。若沒有FreeBSD 光碟的話，可以到FTP 上面的`tools` 目錄(ftp://ftp.FreeBSD.org/pub/FreeBSD/tools/) 下載`fdimage` 使用。

如果要用UNIX 系統(比如其他台FreeBSD 機器) 來製作開機片的話，可以用dd(1) 指令來把映像檔直接寫入到磁片上。在FreeBSD上的話，可以打類似下面的指令：

```
# dd if=boot.flp of=/dev/fd0
```

在FreeBSD 中，`/dev/fd0` 就是指第一台軟碟機(即一般MS-DOS/Windows 上的`A:` 磁碟機)；而`/dev/fd1` 指`B:` 磁碟機，其餘的依此類推。不過其他的UNIX 系統可能會用不同的名稱，這時就要查閱該系統的說明文件了。

現在起，我們可以開始安裝FreeBSD 囉！

2.4 開始安裝

Important: 預設的情況下，安裝過程並不會改變您磁碟機中的任何資料，除非您看到下面的訊息：

```
Last Chance: Are you SURE you want continue the installation?

If you're running this on a disk with data you wish to save then WE
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!

We can take no responsibility for lost disk contents!
```

在看到這最後的警告訊息前，您都可以隨時離開安裝程式而不會變更您的硬碟。如果您發現有任何設定錯誤，這時您可以直接將電源關掉而不會造成任何傷害。

2.4.1 開機啓動流程篇

2.4.1.1 i386™ 平台的開機流程

1. 在一開始，電腦電源開關是關閉的。
2. 打開電腦電源開關。剛開始的時候，它應該會顯示進入系統設定選單或BIOS 要按哪個鍵，常見的有：**F2**, **F10**, **Del** 或**Alt**+**S**。(按鍵請依據實際情況決定) 不論是要按哪個鍵，請按它進入BIOS 設定畫面。有時您的電腦可能會顯示一個圖形畫面，通常做法是按**Esc** 鍵將離開這個圖形畫面，以使您能夠看到必要的設定訊息。
3. 找出可以設定『開機順序(Boot Order)』的選項，通常該選項會列出一些設備讓您選擇，例如：`Floppy`, `CDROM`, `First Hard Disk` 等等。

   如果要用軟碟安裝，請確定floppy disk 要列為開機順序的第一個；若要用光碟安裝，記得CDROM 要列為開機順序的第一個。為了避免不必要的疑惑，請參考機器、主機板說明手冊。

   儲存設定並離開，系統應該會重新啓動。
4. 若要用磁片安裝，請把在Section 2.3.7一節中製作好的`boot.flp` 那張安裝磁片放到第一台軟碟機中。

   如果是從光碟安裝，那麼開機後請將FreeBSD 光碟放入光碟機中。

   如果，開機後如往常一樣而沒有從軟碟或光碟開機，請檢查：

   1. 是不是磁片或光碟太晚放入而錯失開機時間。如果是，請將它們放入，然後重新開機。

2. BIOS 設定不對或忘了儲存設定，請重新檢查BIOS 的設定。
3. 您的電腦BIOS 不支援從光碟開機。

5. 此時，FreeBSD 就開始啓動了。如果是從光碟開機，會見到類似下面的畫面(版本部分省略)：

```
Booting from CD-Rom...
CD Loader 1.2

Building the boot loader arguments
Looking up /BOOT/LOADER... Found
Relocating the loader and the BTX
Starting the BTX loader

BTX loader 1.00 BTX version is 1.01
Console: internal video/keyboard
BIOS CD is cd0
BIOS drive C: is disk0
BIOS drive D: is disk1
BIOS 639kB/261120kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/boot/kernel/kernel text=0x64daa0 data=0xa4e80+0xa9e40 syms=[0x4+0x6cac0+0x4+0x88e9d]
\
```

如果您從軟碟開機，會看到類似下面的畫面(版本部分省略)：

```
Booting from Floppy...
Uncompressing ... done

BTX loader 1.00  BTX version is 1.01
Console: internal video/keyboard
BIOS drive A: is disk0
BIOS drive C: is disk1
BIOS 639kB/261120kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/kernel text=0x277391 data=0x3268c+0x332a8 |

Insert disk labelled "Kernel floppy 1" and press any key...
```

請根據提示將`boot.flp` 磁片取出，並放入`kern1.flp` 這張磁片，然後按**Enter** 鍵。總之，您只需從第一張磁片啓動，然後根據提示，再放入相關磁片即可。

6. 無論從軟碟或光碟開機，接下來會進入FreeBSD boot loader 選單畫面：

Figure 2-1. FreeBSD Boot Loader 選單

您可以等待10 秒，或是按**Enter** 鍵。

2.4.1.2 Alpha 平台的開機流程

1. 在一開始，電腦電源開關是關閉的。
2. 打開電腦電源開關，然後等開機畫面出現。
3. 若要用磁片安裝，請把在Section 2.3.7一節中製作好的`boot.flp` 那張安裝磁片放到第一台軟碟機中。然後，打下列指令來從磁片開機(請把下列軟碟機代號改為你電腦的軟碟機代號)：

   ```
   >>>BOOT DVA0 -FLAGS ” -FILE ”
   ```

   若要用光碟安裝，請把做好的安裝片放入光碟機，然後打下列指令來從光碟開機(請把下列光碟機代號改為你電腦的光碟機代號)：

   ```
   >>>BOOT DKA0 -FLAGS ” -FILE ”
   ```

4. 接著FreeBSD 開機片就會開始了。若是由軟碟開機的話，這時會看到以下訊息：

   ```
   Insert disk labelled "Kernel floppy 1" and press any key...
   ```

   請照指示，拿走`boot.flp` 片，改放`kern1.flp` 片，然後按**Enter**。
5. 無論從軟碟或光碟開機，您都會看到下面這段訊息：

   ```
   Hit [Enter] to boot immediately, or any other key for command prompt.
   Booting [kernel] in 9 seconds... _
   ```

   您可以等待10 秒，或是按**Enter** 鍵。接下來就會進入kernel configuration 選單。

2.4.1.3 Sparc64® 平台的開機流程

大多數的Sparc64® 機器預設會自動從硬碟開機。因此要裝FreeBSD 的話，則需要進入PROM(OpenFirmware)設定由網路或光碟開機才可。

請先重開機，然後等待直到開機訊息出現。這部分可能會隨機器型號不同，而有所差異，但大概會出現像下列這樣：

```
Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
Copyright 1998-2001 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.2, 128 MB memory installed, Serial #51090132.
Ethernet address 0:3:ba:b:92:d4, Host ID: 830b92d4.
```

若您機器此時會先由硬碟開機，那麼需要按**L1**+**A** 或**Stop**+**A** 或者是透過serial console (用法請參閱tip(1) 或cu(1) 內有關~# 的說明) 送出BREAK 指令來進入PROM prompt。大概會像下面：

```
ok          ❶
ok {0}      ❷
```

❶ 這是適用於只有單一CPU 的機器。

❷ 這是適用於SMP 機器，數字部分是指目前在使用中的CPU 編號。

此時請把安裝光碟放入光碟機內，然後在PROM prompt 打`boot cdrom` 即可。

2.4.2 那要怎麼去翻閱偵測硬體的結果呢？

先前在螢幕上所顯示的最後幾百行字，會存在暫存區(buffer) 以便您翻閱。

若要翻閱暫存區，請按**Scroll Lock** 鍵，這會開啓捲動畫面功能。然後就可以使用方向鍵，或是**PageUp**、**PageDown** 鍵來上下翻閱。再按一次**Scroll Lock** 鍵，就會停止畫面捲動。

現在就請試試看，翻閱一下偵測硬體的畫面吧，你應該會看到類似Figure 2-2 的畫面，真正畫面會依你的電腦設備而有所不同。

Figure 2-2. 偵測硬體的例子

```
avail memory = 253050880 (247120K bytes)
Preloaded elf kernel "kernel" at 0xc0817000.
Preloaded mfs_root "/mfsroot" at 0xc0817084.
md0: Preloaded image </mfsroot> 4423680 bytes at 0xc03ddcd4

md1: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1:<VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device 1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <iSA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device 7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
```

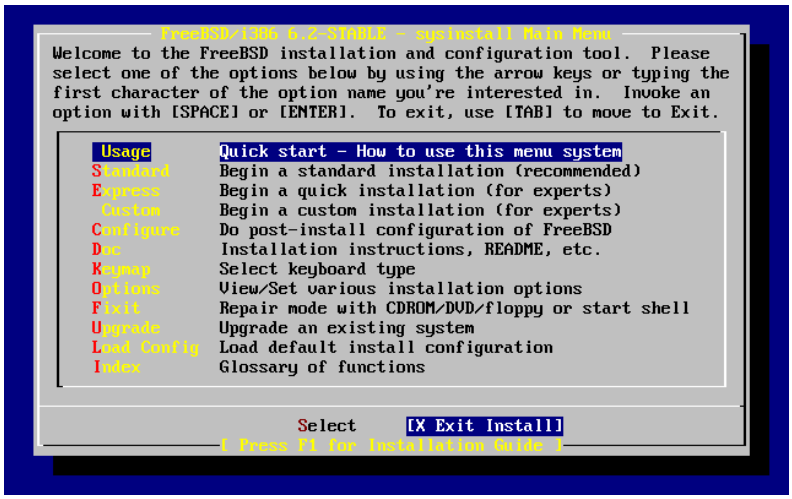
```
uhci0 <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device 7.2 on pci
0
usb0: <VIA 83572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr1
uhub0: 2 ports with 2 removable, self powered
pci0: <unknown card> (vendor=0x1106, dev=0x3040) at 7.3
dc0: <ADMtek AN985 10/100BaseTX> port 0xe800-0xe8ff mem 0xdb000000-0xeb0003ff ir
q 11 at device 8.0 on pci0
dc0: Ethernet address: 00:04:5a:74:6b:b5
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xec00-0xec1f irq 9 at device 10.
0 on pci0
ed0 address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
orm0: <Option ROM> at iomem 0xc0000-0xc7fff on isa0
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq2 on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
atkbdc0: <Keyboard controller (i8042)> at port 0x60,0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq1 on atkbdc0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: model Generic PS/@ mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
pppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
plip0: <PLIP network interface> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master UDMA33
acd0: CD-RW <LITE-ON LTR-1210B> at ata1-slave PIO4
Mounting root from ufs:/dev/md0c
/stand/sysinstall running as init on vty0
```

請仔細檢查每項檢測結果，以確定FreeBSD 有正確偵測到每項硬體。若沒偵測到硬體的話，那畫面就不會列出來了。自訂kernel 可以讓您加上原本預設的`GENERIC` kernel 所不支援的硬體，像是音效卡之類。

而FreeBSD 6.2 起的版本，在偵測硬體後會看到Figure 2-3，請用方向鍵選擇你的國別、地區或群組。然後按**Enter** 鍵就會幫你設定相關國別、鍵盤對應。此外，要離開、重啓**sysinstall** 程式，也很簡單。

Figure 2-3. 選擇國別

Figure 2-4. 離開Sysinstall 程式



在主畫面選擇Exit Install，接下來應該會出現以下訊息：

```
                User Confirmation Requested
     Are you sure you wish to exit? The system will reboot
       (be sure to remove any floppies/CDs/DVDs from the drives).

                        [ Yes ]    No
```

若按下[Yes] 之後，卻忘了把光碟退出來的話，那麼等下重開機後又會再次啓動安裝程式了。

若你是用磁片開機的話，那麼重開機之前，請記得先退出`boot.flp` 片吧。

2.5 介紹Sysinstall

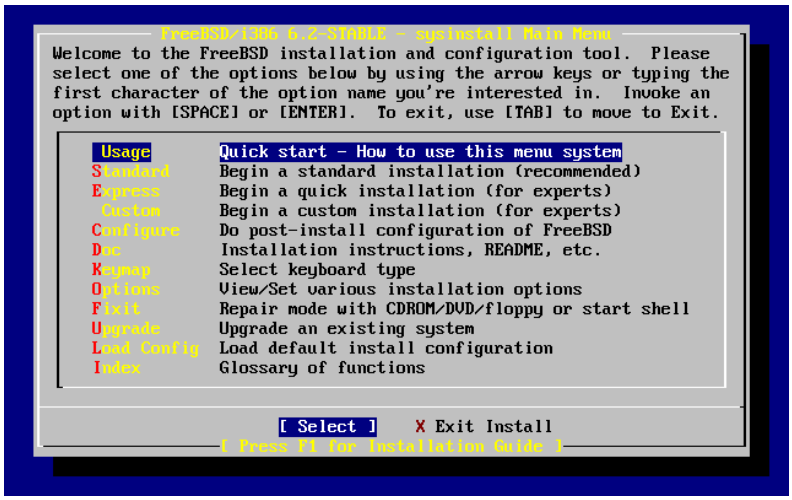
sysinstall 是FreeBSD 計劃所提供的安裝程式。它是以文字模式操作方式為主，分為幾層選單、畫面，以讓您進行安裝。

sysinstall 選單主要由方向鍵、**Enter**、**Tab**、**Space** 以及其他按鍵來進行操作，在**sysinstall** 主畫面的Usage內有這些鍵盤操作上的說明。

要查閱這些說明，請將游標移到Usage，然後選[Select] ，這時你畫面應該會長像Figure 2-5，接著請按**Enter** 鍵。

接下來，會出現安裝的使用說明，閱讀完畢請按**Enter** 以跳回主畫面。

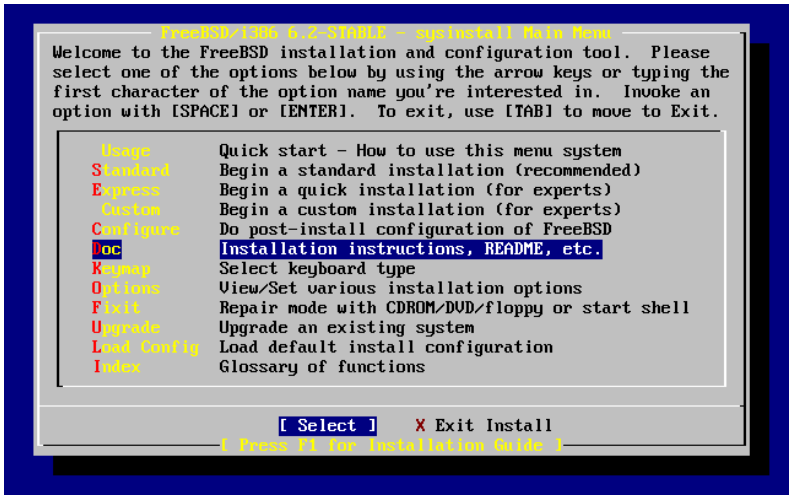
Figure 2-5. 選擇Sysinstall 主畫面的『Usage(快速說明)』



2.5.1 選擇『Documentation(說明文件)』選單

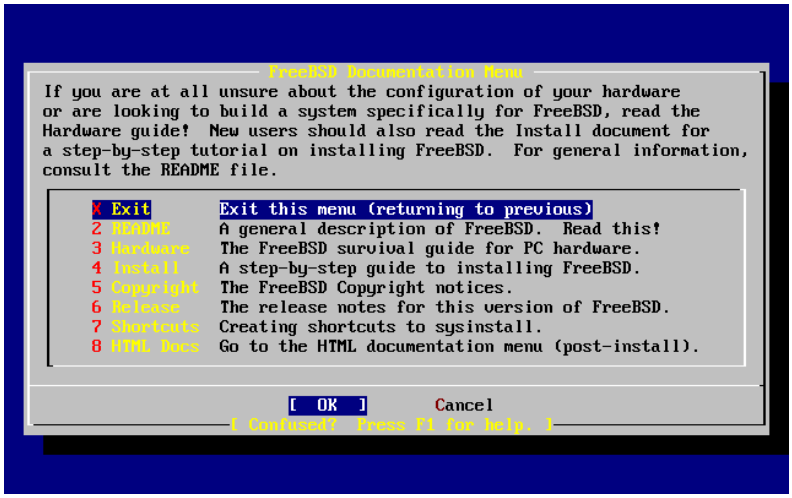
在主畫面用方向鍵選擇Doc，然後按**Enter** 鍵。

Figure 2-6. 選擇『Documentation(說明文件)』選單



這時會出現說明文件的選單。

Figure 2-7. Sysinstall 的說明文件(Documentation)選單



閱讀這些說明文件很重要。

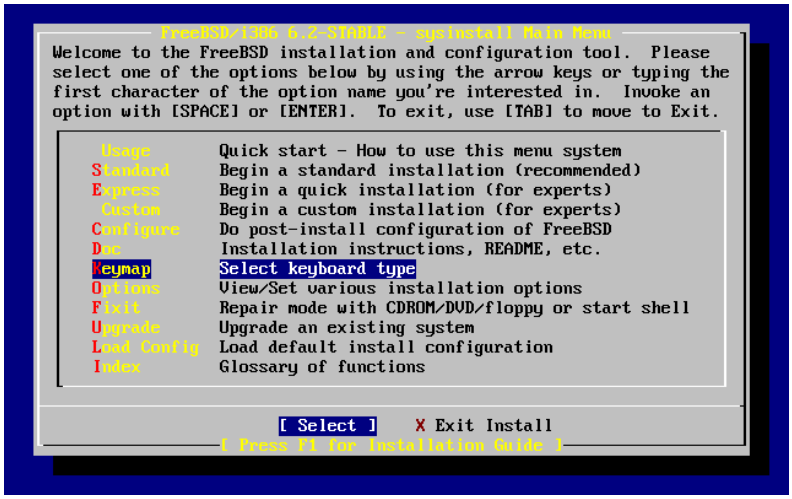
要閱讀文件的話，請用方向鍵選取要閱讀的文件然後按**Enter** 鍵。讀完後，再按一次**Enter** 鍵就會回到說明文件畫面了。

若要回到主畫面，用方向鍵選擇Exit 然後按下**Enter** 鍵即可。

2.5.2 選擇『鍵盤對應』選單

如果要改變鍵盤按鍵的對應模式，請在主選單選取Keymap 然後按**Enter** 鍵即可。一般情況下是不用去改，除非你用的鍵盤不是一般標準或非美式鍵盤。

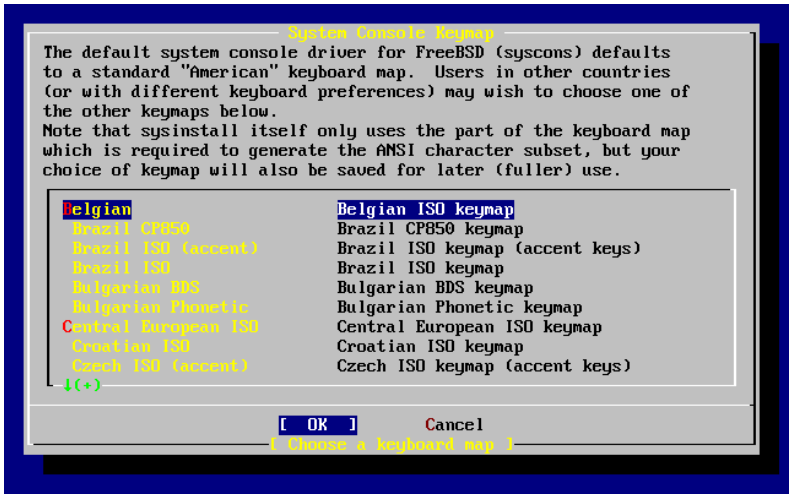
Figure 2-8. Sysinstall 主選單



您可以使用上下鍵移動到您想使用的鍵盤對應方式，然後按下**Space** 鍵以選取它；再按一下**Space** 鍵可以取消選取。當您完成後，請選擇[OK] 然後按**Enter** 鍵即可。

在這個畫面顯示的只是其中一小部分；若只要用預設鍵盤對應方式就好的話，可以用**Tab** 來選[Cancel] 這樣就會返回主畫面。

Figure 2-9. Sysinstall 鍵盤對應選單



2.5.3 安裝選項的設定畫面

請選Options 然後按**Enter** 鍵。

Figure 2-10. Sysinstall 主選單

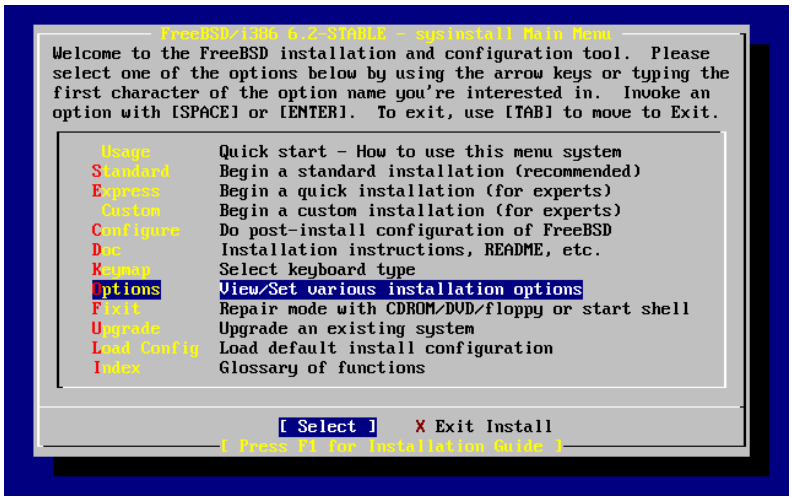
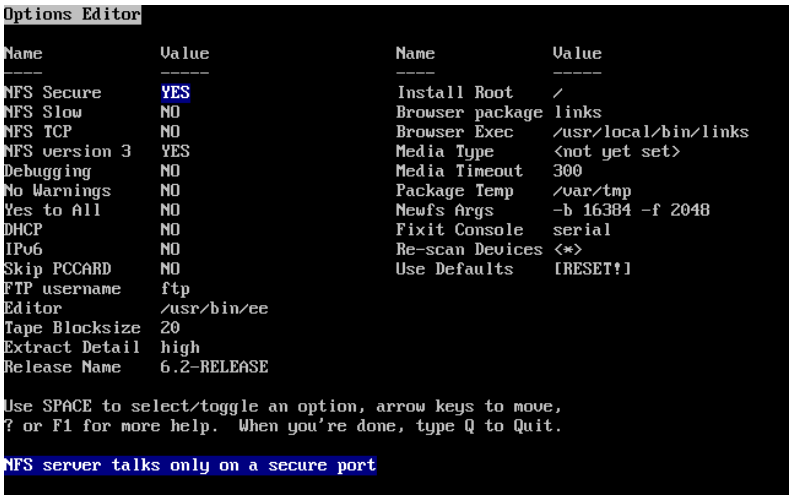


Figure 2-11. Sysinstall 選項設定



通常，使用者大多用預設值就可以了，而不用修改它們。而Release Name 的地方則依你所安裝的版本而有所不同。

而目前所選的的項目，會在畫面下方以藍底白字顯示說明。注意：其中右邊最後的選項是Use Defaults(使用預設值) ，您可以藉由此選項將所有的設定還原為預設值。

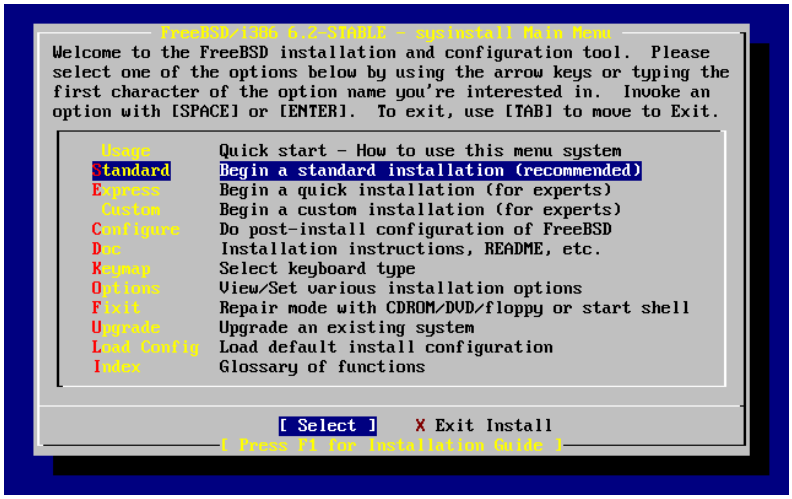
另外，可以按**F1** 鍵來閱讀各選項的說明。

而按**Q** 鍵則是可以回到主畫面。

2.5.4 開始進行標準安裝

Standard(標準)安裝適用於那些初探UNIX 或FreeBSD 的使用者。用方向鍵選擇Standard 然後按**Enter** 鍵即可開始進入標準安裝。

Figure 2-12. 開始進行標準安裝



2.6 硬碟空間的分配

您的第一個任務就是要決定分配給FreeBSD 用的磁碟空間、label，以便**sysinstall** 幫你做相關準備動作。因此，你必須先對FreeBSD 是如何確認磁碟的流程有個概念。

2.6.1 BIOS 磁碟機編號

在安裝、設定FreeBSD 之前，有很重要的一點必須注意，尤其當您有許多顆硬碟的時候。

在PC 架構，當您跑像MS-DOS 或Microsoft Windows 這種跟BIOS 設定相關的作業系統，BIOS 那邊可以調整正常的磁碟機順序，然後這些作業系統會跟著BIOS 做改變。這讓使用者不一定非得要由所謂的“primary master” 硬碟開機。有人發現最簡單、便宜的備份系統方式，就是再去買一顆一模一樣的硬碟，然後定期使用 **Ghost®** 或**XCOPY** 以將資料從第一顆硬碟複製到第二顆硬碟上面去。所以，當第一顆硬碟掛了(可能是病毒或壞軌造成的)，就可以輕鬆透過調整BIOS 中的開機順序，而直接用第二顆硬碟開機。這跟將機殼拆開，把第二顆硬碟跟第一顆對調(要調jumper)有同樣的效果，差別就是：不用拆機殼。

此外，若裝有比較貴的SCSI 卡系統，通常本身也有BIOS 的功能來讓SCSI 設備(最多可到7 個)達到類似改變順序的功能。

習慣上述方式的使用者很可能會感到驚訝，因為在FreeBSD 中並非如此，FreeBSD 不會參考BIOS 設定值，而且也不能偵測“logical BIOS drive mapping” 設定。這會讓人感覺很疑惑，明明就是一樣的硬碟，而且資料也完全從另一顆複製過來，結果卻沒辦法像以前那樣用。

使用FreeBSD 的時候，請將BIOS 中的硬碟開機順序調回原本正常的順序，並且以後不要再改這設定。如果您需要切換硬碟順序的話，那請用硬體方式，直接打開機殼，調jumper 及排線即可。

一段小故事：Bill 及Fred 的安裝歷險

比爾(Bill)打算幫佛列德(Fred)把舊的Wintel 機器灌FreeBSD。他在一顆SCSI 硬碟(ID 是0)裝上FreeBSD。

於是佛列德開始用他新的FreeBSD 系統；但是過了幾天，他發現這顆SCSI 老硬碟發生許多小問題。之後，他就跟比爾說起這件事。

此事經過幾天後，比爾決定是該解決問題的時候了，所以他從後面房間的硬碟“收藏區” 內拿出一個一模一樣的硬碟，並且經過初步surface 掃瞄測試後，顯示這顆硬碟還可堪用，因此，比爾將它的ID 調成4，然後安裝到佛列德的機器，並且將資料從磁碟0 複製到磁碟4。現在新硬碟裝好了，而且看起來好像一切正常；所以，比爾認為現在應該可以開始用它了。於是到SCSI BIOS 中設定SCSI ID 4 為開機碟，用磁碟4 重新開機後，FreeBSD 一切跑得很順利。

佛列德繼續用了幾天後，比爾跟佛列德決定要來玩點新的：試著升級FreeBSD 看看。比爾將ID 0 的硬碟移除(因為有問題)並且又從“收藏區” 中拿了一顆一樣的硬碟來。然後他用佛列德的開機磁片透過FTP 方式將在這顆硬碟上裝了新版的FreeBSD，安裝過程都很順利。

佛列德用了這新版本幾天後，覺得它很適合用在工程部門... 是時候將以前放在舊系統的工作資料複製過來了。因此，佛列德將ID 4 的SCSI 硬碟(裡面有放從舊系統中複製過來的最新資料)先mount 起來，結果竟然發現在ID 4 的硬碟上，他以前的所有資料都不見了！

奇怪，資料到底是跑到哪裡去了呢？

原來，當初比爾將ID 0 硬碟的資料複製到ID 4 硬碟的時候，ID 4 就變成“新的複製品(new clone)” 。而當他調SCSI BIOS 設定ID 4 為開機碟，想讓系統從ID 4 開機，這步驟其實只是他自己搞混了，因為大部分的作業系統可以藉由調BIOS 設定以改變開機順序，但是FreeBSD 卻會把開機順序還原成正常的模式，因此，佛列德的FreeBSD 還是從最初的那顆ID 0 硬碟開機的。事實上，所有的資料都還在故事最初的那顆硬碟上，而不是在他想像中的ID 4 硬碟。

我們很高興在發現這件事時，那些資料都還在，我們把資料從最初的那顆ID 0 硬碟取出來並交還給佛列德(而且比爾也從此瞭解了0 的重要....)。

雖然我們這邊的例子是用SCSI 硬碟，但是相同的觀念也可以套用在IDE 硬碟上。

2.6.2 以FDisk 來建立分割磁區(Slices)

> **Note:** 在這時候您所做的變更都還不會真正寫入硬碟中。如果你發現弄錯了，想要重來一遍的話，可以用選單來離開**sysinstall**，或是按**U** 鍵來Undo(回復) 所有設定。如果你弄亂了而且不知道怎麼離開，你可以直接將電腦電源關掉再重來。

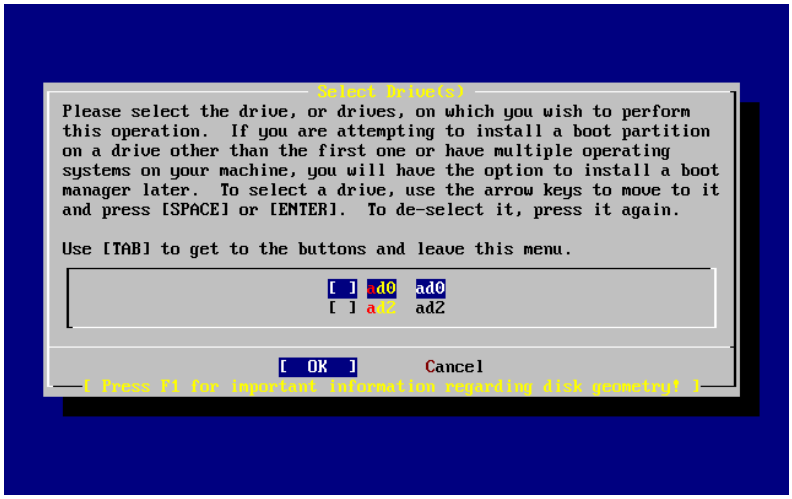
在**sysinstall** 主畫面選擇使用標準安裝後，應該會看到下面的訊息：

```
                              Message
In the next menu, you will need to set up a DOS-style ("fdisk")
partitioning scheme for your hard disk. If you simply wish to devote
all disk space to FreeBSD (overwriting anything else that might be on
the disk(s) selected) then use the (A)ll command to select the default
partitioning scheme followed by a (Q)uit. If you wish to allocate only
free space to FreeBSD, move to a partition marked "unused" and use the
(C)reate command.
                              [  OK  ]
```

```
[ Press enter or space ]
```

這時請依畫面說明，按**Enter** 鍵。然後會看到一個列表，上面會列出所有在偵測硬體時所找到的硬碟。Figure 2-13 範例顯示的是有找到兩個IDE 磁碟機的情形，這兩個磁碟機分別為：`ad0` 與`ad2`。

Figure 2-13. 選擇FDisk 要分割的硬碟



你可能會好奇，為何`ad1` 沒列在這裡。為什麼會不見了呢？

試想，如果您有兩顆IDE 硬碟，一個是primary master，一個是secondary master，這樣會發生什麼事呢？如果FreeBSD 依照找到的順序來為他們命名，比如首先是`ad0` 再來是`ad1` 那麼就不會出現困擾。

但是，現在問題來了。如果您現在想在primary slave 加裝第三顆硬碟，那麼這顆硬碟的名稱就會是`ad1`，之前原本的`ad1` 就會變成`ad2`。這樣會造成什麼問題呢？因為硬體設備的名稱(像是`ad1s1a`)是用來尋找檔案系統的，因此您可能會突然發現，有些檔案系統從此無法正常顯示，必須修改FreeBSD 設定(`/etc/fstab`)才可以正確顯示。

為了解決這個問題，在設定kernel 時可以採用IDE 硬碟所在的位置來命名，而非根據找到的順序。使用這種方式的話，在secondary master 的IDE 硬碟就*永遠*會是`ad2`，即使系統中並沒有`ad0` 或`ad1` 也不受影響。

由於此為FreeBSD kernel 預設設定，也就是為何上述畫面只顯示`ad0` 及`ad2` 之故。畫面上這台機器的兩顆硬碟是分別裝在primary 以及secondary 排線上的master，這兩顆都沒有裝在slave 上。

請選好想安裝FreeBSD 的硬碟，然後按下[OK]。接著就會開始**FDisk**，然後會看到類似Figure 2-14 的畫面。

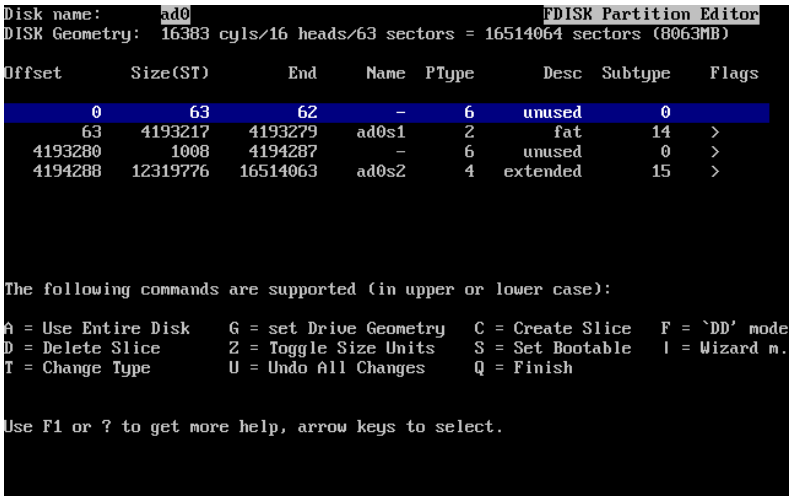
FDisk 的顯示畫面分為三個部分。

第一部份是畫面最上方的前兩行，這裡會顯示目前所選的硬碟資訊，包括它在FreeBSD 的名稱、硬碟geometry、硬碟總容量。

第二部分會顯示目前所選的硬碟上有哪些slice 以及各slice 的起末位置、所佔容量、FreeBSD 名稱、描述說明、子類別(sub-type)。例子中顯示出有2 個小的並且尚未使用的slice，這是受到PC 的硬碟本身架構影響之故。此外，還有一個大的FAT slice(通常是MS-DOS / Windows 中的`C:`)，以及一個延伸磁碟分割區(在MS-DOS / Windows 內的其他磁碟代號)。

第三部分則顯示**FDisk** 可用的指令。

Figure 2-14. (舉例)未編輯前的Fdisk 分割區(Partition)



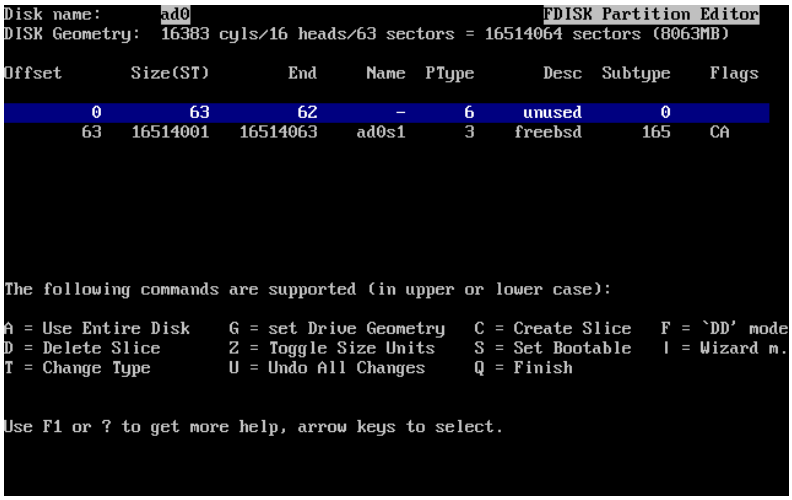
接下來要做的事，跟您要怎麼分割硬碟有關。

若要讓FreeBSD 使用整顆硬碟(稍後的安裝會再要您確認以**sysinstall** 來繼續安裝，就會清除該硬碟內上的資料)，那麼就可以按**A** 鍵(Use Entire Disk)，以刪除所有既存的slice，取而代之的是一個小的並標示為unused(同樣的，這也是PC 硬碟架構所造成)的slice，以及一個大的FreeBSD slice。之後，請用方向鍵把光棒移至該FreeBSD slice，然後按**S** 鍵以便將此slice 標為開機slice。此時的畫面應該類似Figure 2-15。請注意：在Flags 欄位的A 值表示該slice 屬於*active*，也會由此slice 來開機。

若要刪除現有slice 以挪出空間給FreeBSD 使用，可以把光棒移到要刪除的slice 後按**D** 鍵，然後再按**C** 鍵，此時會出現對話框，請輸入要新增的slice 大小為何，輸入合適大小之後按**Enter** 鍵即可。該預設值為可分配空間的最大值，可以是最大的或尚未分配的整顆硬碟大小。

若已建立完畢給FreeBSD 的空間(透過類似**PartitionMagic** 之類的工具)，那麼可以按**C** 鍵以新增slice。同樣也會有對話框出現，來問想要新增的slice 大小為何。

Figure 2-15. Fdisk 採用整顆硬碟作分割區(Partition)



完畢後請按**Q** 鍵。這些更改會暫存給**sysinstall** 使用，但還不會真正寫入到硬碟。

2.6.3 安裝Boot Manager

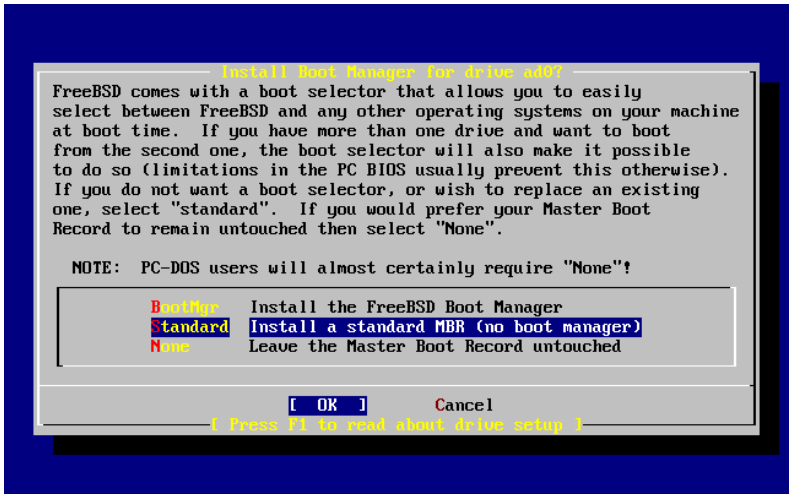
現在可以選擇是否要裝boot manager。一般而言，遇到下列情況才會需要裝boot manager：

- 有一個以上的硬碟，而FreeBSD 並非裝在第一個硬碟上。
- 同一顆硬碟上除了有裝FreeBSD 之外，還有裝其他作業系統，所以需要在開機時選擇要進入哪個作業系統。

若只裝FreeBSD，並且是裝在第一顆硬碟，那麼選Standard 即可。若已經有使用其他的boot manager 可開機進入FreeBSD 那麼請選None 即可。

請依自身需求與情況做抉擇，然後按**Enter** 鍵。

Figure 2-16. Sysinstall 的Boot Manager 選單



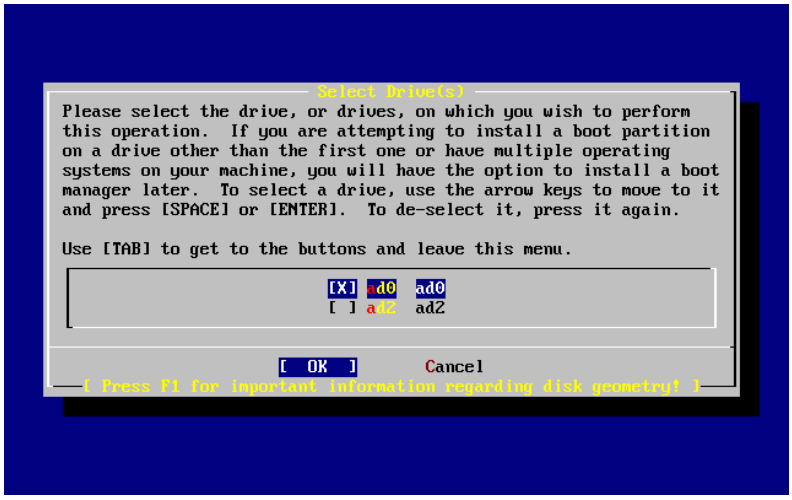
按**F1** 會有不同作業系統共存時，有可能遇到的相關問題說明。

2.6.4 在其他硬碟上建立分割磁區(Slices)

若有一個以上的硬碟，那麼在選完boot manager 之後會再回到選擇硬碟的畫面。若要把FreeBSD 裝在多個硬碟上，那麼可以在此選擇其他硬碟，並重複使用**FDisk** 來建立slice 。

> **Important:** 若第一顆硬碟不是裝FreeBSD 的話，那麼每一顆就要都裝FreeBSD boot manager 才可以。

Figure 2-17. 離開『選擇硬碟』畫面



Tab 鍵可以在最後選擇的硬碟以及[OK]、[Cancel] 之間進行切換。

先按一次**Tab** 會先移到[OK]，然後再按**Enter** 鍵以繼續安裝。

2.6.5 以Disklabel 來建立分割區(Partitions)

現在必須在剛建立好的slice 規劃一些分割區。請注意：每個分割區的代號是從a 到h，此外b、c、d 通常是特殊用途，不該隨意變動。

有些程式可以透過特殊的分割方式而達到更好的效果，尤其是分割區是分散在不同硬碟上的時候。但是，現在是您第一次裝FreeBSD，所以請不要去煩惱該如何分割硬碟才好。最重要的是，裝好FreeBSD 然後學習如何善用之。當對FreeBSD 有一定程度的熟悉之後，可以隨時重裝FreeBSD，並改變分割的方式。

下面例子有四個分割區——其中一個是swap 空間，i 其他三個是檔案系統。

Table 2-2. 第一顆硬碟的分割區(Partition)配置

分割區	檔案系統	大小	介紹
a	/	128 MB	此為根目錄檔案系統(root filesystem)。其他的檔案系統都會掛載在根目錄之下。128 MB 對於此檔案系統來說是相當合理的大小，因為通常這裡並不會放太多資料，而在FreeBSD 裝完後會用到約40 MB 的根目錄空間。剩下的空間是放臨時資料用的，此外也應該要預留一些空間，因為日後的FreeBSD 版本可能會需要更多的/(根目錄) 空間。

分割區	檔案系統	大小	介紹
b	N/A	RAM 的2~3倍	系統的swap 空間是放在b 分割區。如何選擇適合的swap 空間大小可是一門學問。一般來說，swap 空間應該是記憶體(RAM)大小的2 或3 倍。此外，swap 至少需要64 MB，因此若RAM 小於32 MB 的話，請把swap 大小設為64 MB。若有一個以上的硬碟，則可以在每個硬碟都配置swap 空間。FreeBSD 會善用每個硬碟上的swap 空間，如此一來便能有效提高swap 的性能。若您屬這類情況，請先算出總共需要的swap 總大小(比如：128 MB)，然後除以全部的硬碟數量(比如：兩顆硬碟)，這樣算出來的結果就是每個硬碟上所需配置的swap 大小，在這個例子中，則每個硬碟所需之swap 空間為64 MB 。
e	/var	256 MB	/var 目錄會放的檔案有很多種，像是log 檔案以及其他的系統管理檔案。這些檔案大部分都是FreeBSD 每日運作所會讀、寫。把這些檔案另外放到專門的檔案系統(即/var) 則可以最佳化這些檔案的存取，而不致於影響其他目錄的存取。
f	/usr	剩餘的硬碟空間	所有其他檔案通常會存在/usr 及其子目錄內。

若要把FreeBSD 裝在多個硬碟上，那麼必須在您所配置的其他slice 上新增分割區。最簡單的方式，就是在每個硬碟上建立分割區，一個給swap 空間，另一個則是檔案系統。

Table 2-3. 其他硬碟的分割區(Partition)配置

分割區	檔案系統	大小	介紹
b	N/A	請參閱右側的介紹	前面有提過，swap 空間是可以跨各硬碟。即使沒有使用a 分割區，但習慣上還是會把swap 空間設為b 分割區。
e	/disk*n*	剩餘的硬碟空間	剩下的空間是一個大的分割區，最簡單的做法是將之規劃為a 分割區，而不是e 分割區。然而，習慣上a 分割區是保留給根目錄(/)所使用的。當然，您不一定要遵循此習慣，但**sysinstall** 本身會，所以照它既有的方式會讓你安裝更加清爽、潔淨。你可以把這些檔案系統掛載在任何地方，本範例是建議把它們掛載於/disk*n* 目錄，其中的*n* 的數字，則依各硬碟的順序而有所變化。但若您高興，也可以把它們掛載於其他地方。

完成分割區配置之後，就可以用**sysinstall** 來建立之。您會看到如下訊息：

```
                                Message
Now, you need to create BSD partitions inside of the fdisk
partition(s) just created. If you have a reasonable amount of disk
space (200MB or more) and don't have any special requirements, simply
use the (A)uto command to allocate space automatically. If you have
```

```
more specific needs or just don't care for the layout chosen by
(A)uto, press F1 for more information on manual layout.

                                [  OK  ]
                          [ Press enter or space ]
```

請按**Enter** 鍵以進入FreeBSD 分割區編輯器，叫做**Disklabel**。

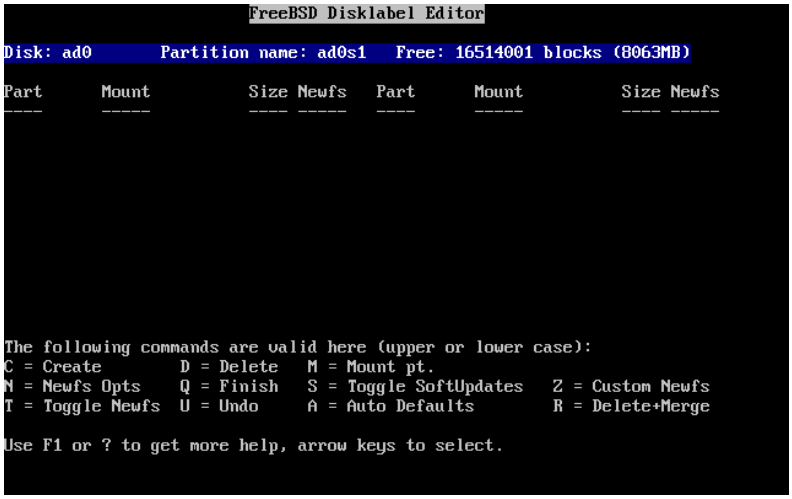
Figure 2-18 顯示第一次執行**Disklabel** 的畫面，這畫面可區分為三個區塊。

前幾行顯示的是正在編輯的硬碟，以及目前正在建立的slice 位於哪個分割區上。(在此處，**Disklabel** 是使用`Partition name`(分割區名稱)，而非slice 名稱)。此畫面也會顯示目前slice 還有多少空間可供使用，換句話說就是尚未指定分割區的多餘空間。

在畫面中間，則顯示已建立的分割區、每個分割區的檔案系統名稱、所佔大小，以及一些參數。

在畫面下方，則顯示**Disklabel** 可用的按鍵。

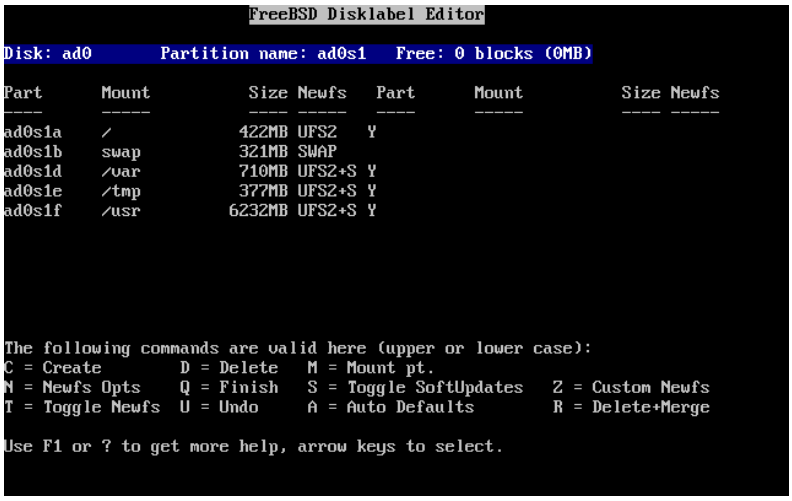
Figure 2-18. Sysinstall 的Disklabel 編輯器



Disklabel 可自動分配分割區，並賦予預設值大小，按**A** 即可自動完成。您會看到類似Figure 2-19 的畫面。不過，由於所用的硬碟大小不一，所以自動分配所設定的大小不一定合用，不要緊，您不一定得使用預設大小才可以。

> **Note:** 預設會給`/tmp` 目錄作為獨立分割區，而非附屬於`/` 之下。如此一來，可避免`/` 會被一堆臨時檔案塞爆。

Figure 2-19. Sysinstall 的Disklabel 編輯器—使用自動分配

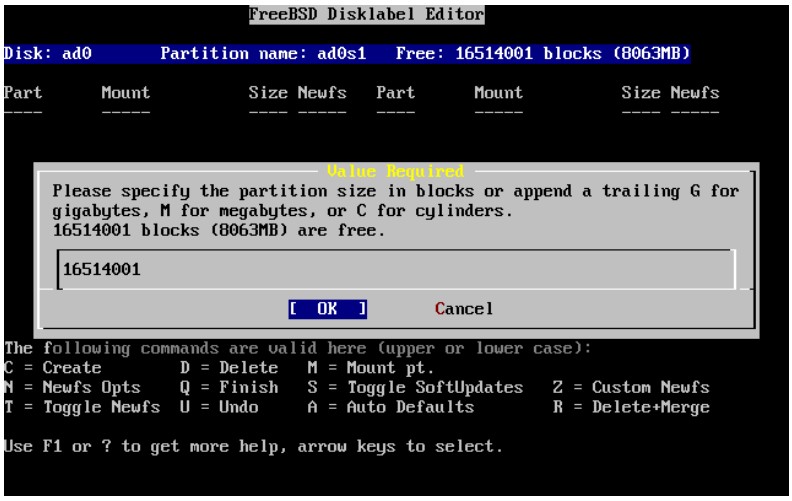


如果您不想用自動分配分割區而希望自行設定，請用方向鍵選擇第一個分割區，並按下**D** 刪除之。重複此動作直到刪除所有分割區。

建立第一個分割區(a，掛載為/ ——根目錄)，請在畫面最上方選擇正確的磁碟分割磁區(slice)並按下**C**。接下來將出現對話框，會要求輸入新的分割區大小(如Figure 2-20 所示) 。這邊可以直接輸入以block 為單位，或者是以M(MB)為單位、或以G(GB)為單位，或者以C(磁柱，cylinders) 為單位。

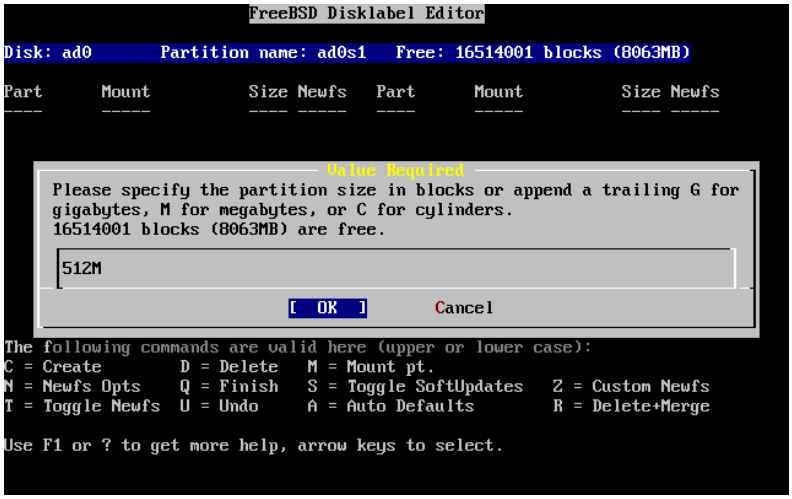
> **Note:** 自FreeBSD 5.X 起，則可使用`Custom Newfs` 選項來用UFS2 (從FreeBSD 5.1 起，此即為預設值)。若是使用`Auto Defaults` 自動預設的情況下，則可以再用`Custom Newfs` 選項，或者在建立檔案系統時指定`-O 2` 參數亦可。若用`Custom Newfs` 選項的話，則別忘了要加上`-U` 來啓用SoftUpdates 功能！

Figure 2-20. 根目錄的空間分配



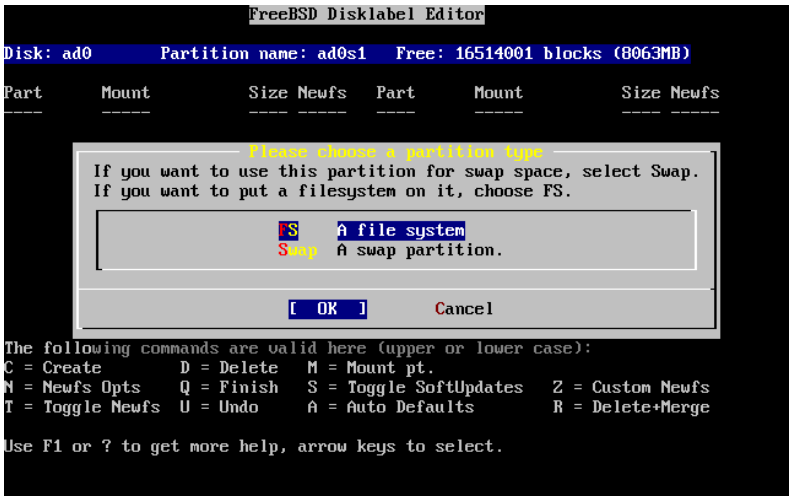
此處預設顯示的大小，會是整個slice 的所有空間。若要採用先前例子所介紹的劃分大小，則按**Backspace** 鍵來消除這些數字，並輸入例子中的**128M**，如Figure 2-21 所示。接著按下[OK]。

Figure 2-21. 修改根目錄的空間分配



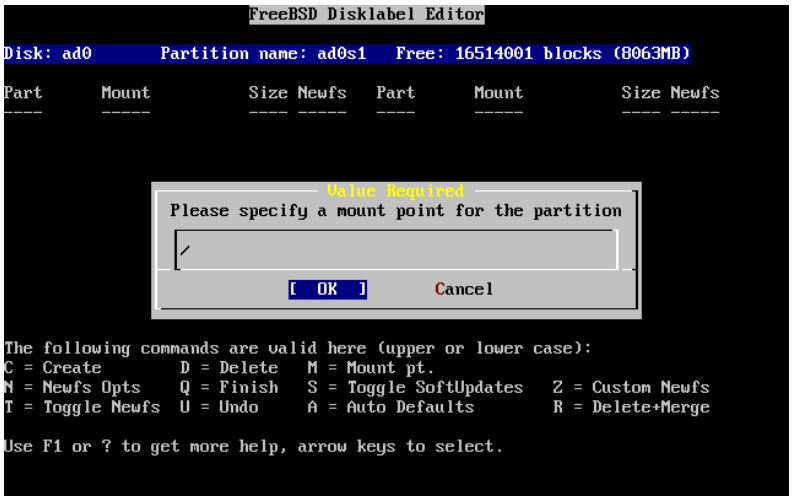
在輸入之後會問所要建立的是檔案系統(file system)或者是swap 空間，如Figure 2-22 所示。第一個選項為檔案系統，所以選擇FS 後按下**Enter**。

Figure 2-22. 選擇分割區的類型



最後，因為要新增的是檔案系統，所以必須告訴**Disklabel** 要將其掛載至何處。如Figure 2-23 所示。根目錄檔案系統的掛載點為**/**，所以請輸入**/** ，然後按下**Enter**。

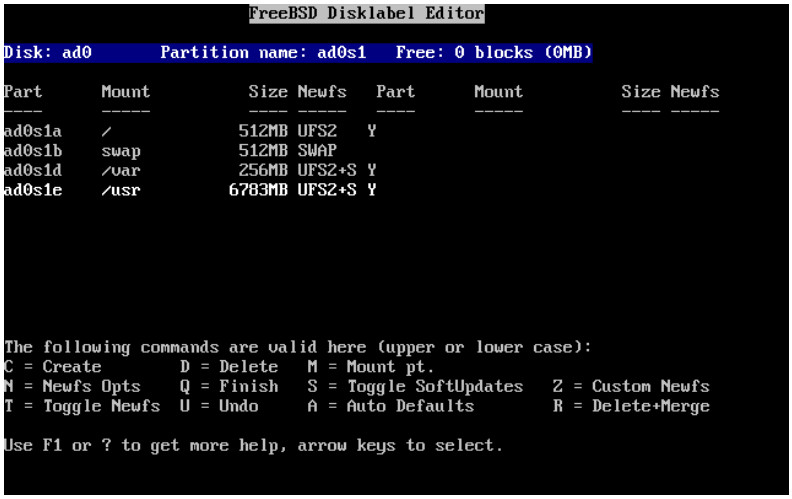
Figure 2-23. 選擇根目錄的掛載點



剛所建立的分割區會顯示在畫面上，可以用上述類似動作來建立其他分割區。然而在建立swap 分割區時，系統並不會問要掛載於哪邊，因為swap 空間是不必額外掛載的。此外在建立最後分割區`/usr` 時，可以直接採用預設大小，也就是該slice 剩餘的所有空間。

最後FreeBSD 上的DiskLabel 編輯器畫面會類似Figure 2-24，實際數字則依安裝選擇而有所不同。請按下**Q** 即可完成分割區規劃。

Figure 2-24. Sysinstall Disklabel 編輯器



2.7 選擇想要安裝的

2.7.1 選擇要安裝的套件集(Distribution Set)

要裝哪些套件，主要取決於該系統的用途為何及磁碟空間而定。預置的套件，從最小安裝到完整安裝都有。若是UNIX 或FreeBSD 新手，通常直接選其中之一即可。而自訂套件比較適合有經驗的人來用。

若要瞭解各套件的選項細節資訊，請按**F1** 鍵。看完之後按**Enter** 即會回到剛才的套件選擇畫面。

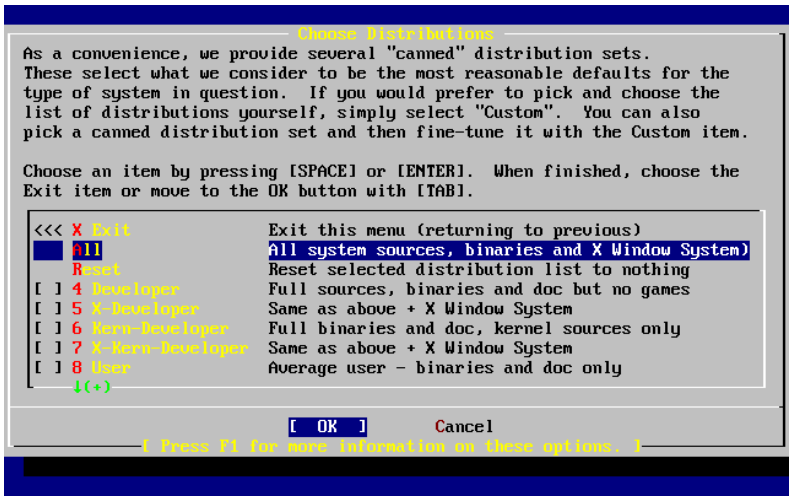
若需要GUI 介面，那必需加選X 開頭的相關套件。至於X server 的設定及要用哪一類的桌面管理，必須在FreeBSD 裝好之後才能進行。X server 設定細節部分請參閱Chapter 5。

預設安裝的X11 版本為**Xorg**。

若需要自訂kernel，那麼需加選有含source code 的選項。至於為何需自訂kernel 及相關細節，請參閱Chapter 8。

很明顯地，全部都裝就不用困擾需要裝什麼了。若硬碟夠大，請以方向鍵選Figure 2-25 圖下的All 選項，並按下**Enter** 即可。若硬碟空間不夠，請依自身需求選擇安裝。當然在安裝完畢後，還是可以依需求再加裝其他套件。

Figure 2-25. 選擇要裝的套件集(Distributions)



2.7.2 安裝Ports Collection

在裝完套件集之後，接著會問是否要裝FreeBSD Ports 套件。Ports 套件可以讓您輕鬆安裝各種常見的軟體，它本身並不含那些軟體的原始碼，而是一個包含如何自動下載、編譯、安裝third-party 軟體的檔案集合。Chapter 4 會介紹如何使用ports。

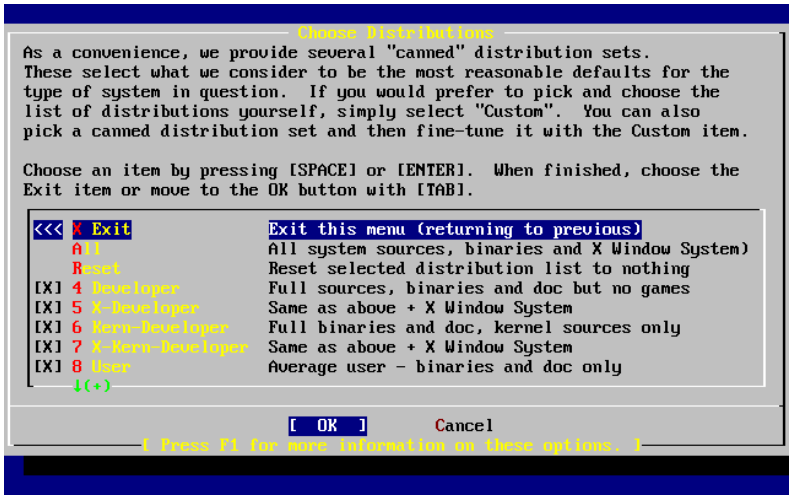
安裝程式並不會檢查是否有足夠空間來放ports tree，所以請先確認有足夠空間。目前FreeBSD 8.0 的FreeBSD Ports Collection 大約需要417 MB 的空間。因此，可以推估更新版的FreeBSD 會需要更多的空間來裝。

```
                    User Confirmation Requested
Would you like to install the FreeBSD ports collection?
```

```
This will give you ready access to over 20,000 ported software packages,
at a cost of around 417 MB of disk space when "clean" and possibly much
more than that if a lot of the distribution tarballs are loaded
(unless you have the extra CDs from a FreeBSD CD/DVD distribution
available and can mount it on /cdrom, in which case this is far less
of a problem).

The Ports Collection is a very valuable resource and well worth having
on your /usr partition, so it is advisable to say Yes to this option.

For more information on the Ports Collection & the latest ports,
visit:
    http://www.FreeBSD.org/ports

                                [ Yes ]     No
```

用方向鍵選[Yes] 就會裝Ports Collection，否則就選[No] 以略過。選好後按**Enter** 繼續，然後會再次回到選擇套件集的畫面。

Figure 2-26. 確認要安裝的套件集



若要勾選的項目都確認沒問題的話，就以方向鍵選Exit 退出並確認[OK] 有選到，然後按I **Enter** 繼續。

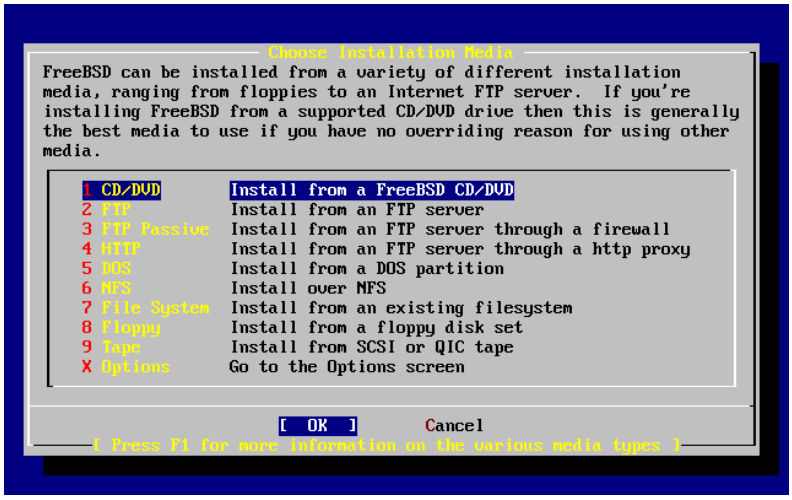
2.8 選擇安裝來源

若要從CDROM 或DVD 安裝，用方向鍵將游標移到Install from a FreeBSD CD/DVD，並確定選[OK] 後按下**Enter** 就會開始裝了。

若是要用其他的方式安裝的話，請選擇適當的安裝來源，然後遵照螢幕指示進行安裝即可。

按**F1** 可以顯示針對此部分(安裝來源)的線上說明。按一下**Enter** 就會回到『選擇安裝來源』的畫面了。

Figure 2-27. 選擇安裝來源



FTP 安裝模式: 使用FTP 安裝的話，有分三種模式：主動式(active)FTP、被動式(passive)FTP 或是透過HTTP proxy server。

主動式FTP：從FTP server 安裝

該選項會透過 “Active” 模式作FTP 傳輸動作。這會無法穿過防火牆，但可用在那些較古早、不支援被動模式的FTP 站。若FTP 連線會卡住(預設為被動模式)，那請改換主動模式看看！

被動式FTP：透過防火牆，從FTP server 安裝

該選項會讓**sysinstall** 全程使用 “Passive(被動式)” 來進行FTP 連線，就可以穿過只允許使用固定TCP port 連入的防火牆。

透過HTTP proxy 的FTP：透過http proxy 來從FTP 站安裝

該選項會讓**sysinstall** 的FTP 連線，先透過HTTP 協定(就像網頁瀏覽器一樣)連到proxy server，而proxy server 會解譯送過來的的請求，然後轉送給FTP server。這可以穿透只允許HTTP 連線但不允許FTP 連線的防火牆。但記得要用之時，必須指定proxy server 的位址。

對proxy FTP server 而言，通常要在登入用的帳號名稱後面，加上 “@” 符號再加上要登入的server 名稱。然後，proxy server 就會 “fakes(偽裝)” 為真的server 樣子。舉個例子，若要到`ftp.FreeBSD.org` 來裝，但中間透過proxy FTP server 也就是`foo.example.com` 並且使用port 1234。

在此情況下，可以到options 選單，將FTP username 設為`ftp@ftp.FreeBSD.org`，密碼則設為自己的email 信箱。安裝來源部分，則使用FTP (或proxy 有支援的話，就用passive FTP)，而URL 則用`ftp://foo.example.com:1234/pub/FreeBSD`。

因為`ftp.FreeBSD.org` 的`/pub/FreeBSD` 會被porxy 到`foo.example.com`，所以就可以從`foo.example.com` *這台*機器(這台會從`ftp.FreeBSD.org` 抓檔回來給您) 安裝。

2.9 開始進行安裝

到此為止，可以開始進行安裝了，這也是您避免更動到硬碟的最後機會。

```
                      User Confirmation Requested
 Last Chance! Are you SURE you want to continue the installation?

 If you’re running this on a disk with data you wish to save then WE
 STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!

 We can take no responsibility for lost disk contents!

                              [ Yes ]    No
```

選擇[Yes] 並按下**Enter**以確認真的要開始安裝

安裝所需時間會依據所選擇安裝的套件集(distribution) 、安裝來源以及電腦速度而有所不同。在安裝的過程中，會有一些訊息顯示目前的安裝進度。

當您看到下面的訊息表示已經安裝完成了：

```
                                 Message

Congratulations! You now have FreeBSD installed on your system.

We will now move on to the final configuration questions.
For any option you do not wish to configure, simply select No.

If you wish to re-enter this utility after the system is up, you may
do so by typing: /usr/sbin/sysinstall.

                                  [ OK ]

                       [  Press enter or space  ]
```

請按**Enter** 鍵來進行相關的後續設定。

如果剛選的是[No] 並按下**Enter** 鍵，那麼會中斷安裝(就不會動到你的原有系統)。接著，會出現以下訊息：

```
                                 Message
Installation complete with some errors.  You may wish to scroll
through the debugging messages on VTY1 with the scroll-lock feature.
You can also choose "No" at the next prompt and go back into the
installation menus to retry whichever operations have failed.

                                  [ OK ]
```

這段訊息乃是因為都沒裝任何東西之故，請按**Enter** 以跳回主畫面。

2.10 後續安裝

安裝系統成功之後，可以在新裝好的FreeBSD 重開機之前，或者是事後再透過`sysinstall` (FreeBSD 5.2 之前版本則是`/stand/sysinstall`) 然後選擇Configure 選項以進行後續設定。

2.10.1 設定網路

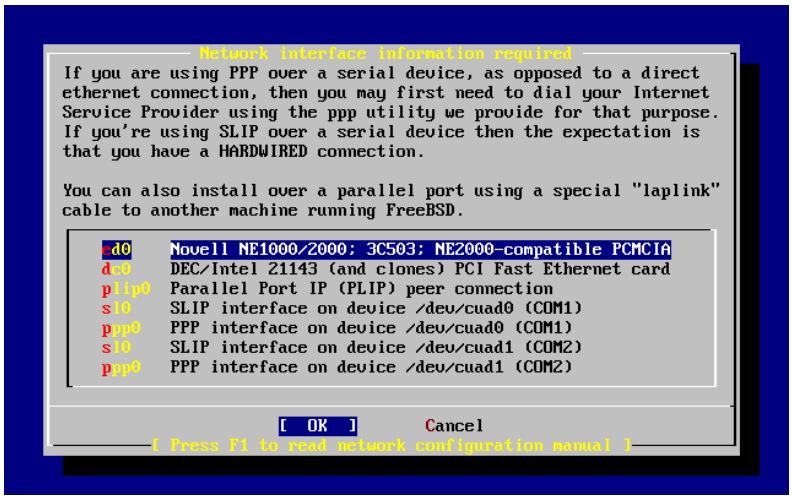
如果您之前有設定用PPP 連線透過FTP 安裝，那麼這個畫面將不會出現；正如上面剛所說的，您可以稍後再做更改。

有關LAN 或把FreeBSD 設定為gateway 或router 請參閱使用手冊中有關網路進階運用 的章節。

```
                    User Confirmation Requested
  Would you like to configure any Ethernet or SLIP/PPP network devices?

                              [ Yes ]   No
```

如果要設定網路卡，請選擇[Yes] 然後按**Enter**。否則請選[No] 以繼續。

Figure 2-28. 選擇網路卡



用方向鍵選擇您要設定的網路卡，然後按**Enter**。

```
                    User Confirmation Requested
        Do you want to try IPv6 configuration of the interface?

                              Yes   [ No ]
```

在私人區域網路的情況，由於目前的Internet 協定(IPv4)還算夠用，所以請選[No] 不設定IPv6，然後按**Enter**。

若是透過RA server 連到既有的IPv6 環境，那麼就選[Yes] 並按**Enter**，之後系統會花幾秒鐘去搜尋RA server。

```
                              User Confirmation Requested
        Do you want to try DHCP configuration of the interface?
```

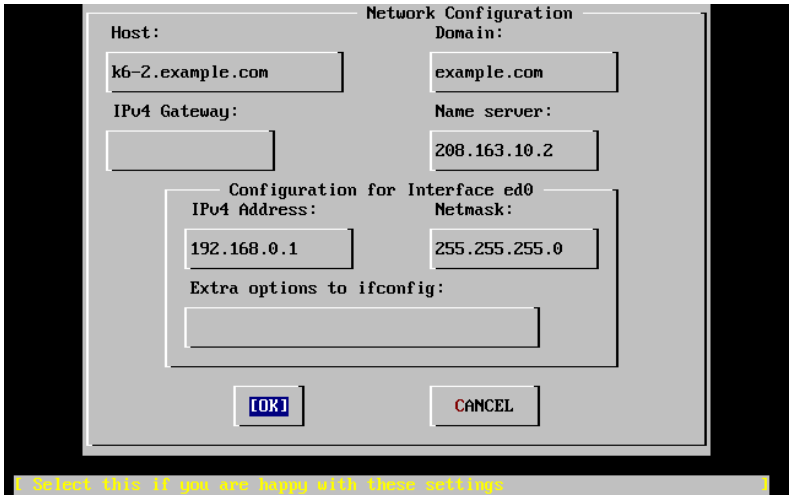
```
Yes   [ No ]
```

接下來，若不需要DHCP (Dynamic Host Configuration Protocol)請選[No] 並按**Enter**。

選擇[Yes] 的話，則會執行**dhclient**，若成功要到IP，則其會自動填上相關的環境設定，細節請參閱Section 27.5。

下面的網路設定圖顯示如何在區域網路(LAN)中，將該機器設定為gateway 的方式：

Figure 2-29. 設定ed0 這張網路卡的網路設定



可用**Tab** 鍵在各欄位間作切換，並填上適合的資料：

Host(機器名稱)

完整的機器名稱，例如本例中的`k6-2.example.com`。

Domain(網域)

機器所屬的網域名稱，例如本例中的`example.com`。

IPv4 Gateway

這裡請輸入Gateway 的IP 位址，其可負責將封包轉遞到遠端網路。只有在該gateway 屬於該網路其中節點之一時，才要輸入。若這機器本身要做為該區域網路的gateway 的話，*請保持本欄為空白*。此外，通常IPv4 Gateway 也會被認為是default gateway 或default route。

Name server(Name server 或DNS server)

該網路所用的DNS server 之IP。本例假設該機器所在的網路沒有DNS，故填上的是該ISP 所提供的DNS server (`208.163.10.2`)。

IPv4 address

The IP address to be used for this interface was `192.168.0.1`

Netmask

The address block being used for this local area network is a Class C block (`192.168.0.0 - 192.168.0.255`). The default netmask is for a Class C network (`255.255.255.0`).

Extra options to ifconfig

Any interface-specific options to `ifconfig` you would like to add. There were none in this case.

Use **Tab** to select [OK] when finished and press **Enter**.

```
                    User Confirmation Requested
        Would you like to Bring Up the ed0 interface right now?

                              [ Yes ]   No
```

Choosing [Yes] and pressing **Enter** will bring the machine up on the network and be ready for use. However, this does not accomplish much during installation, since the machine still needs to be rebooted.

2.10.2 Configure Gateway

```
                     User Confirmation Requested
      Do you want this machine to function as a network gateway?

                               [ Yes ]    No
```

If the machine will be acting as the gateway for a local area network and forwarding packets between other machines then select [Yes] and press **Enter**. If the machine is a node on a network then select [No] and press **Enter** to continue.

2.10.3 Configure Internet Services

```
                      User Confirmation Requested
Do you want to configure inetd and the network services that it provides?

                                Yes   [ No ]
```

If [No] is selected, various services such **telnetd** will not be enabled. This means that remote users will not be able to **telnet** into this machine. Local users will still be able to access remote machines with **telnet**.

These services can be enabled after installation by editing `/etc/inetd.conf` with your favorite text editor. See Section 27.2.1 for more information.

Select [Yes] if you wish to configure these services during install. An additional confirmation will display:

```
                      User Confirmation Requested
The Internet Super Server (inetd) allows a number of simple Internet
services to be enabled, including finger, ftp and telnetd.  Enabling
these services may increase risk of security problems by increasing
the exposure of your system.

With this in mind, do you wish to enable inetd?
```

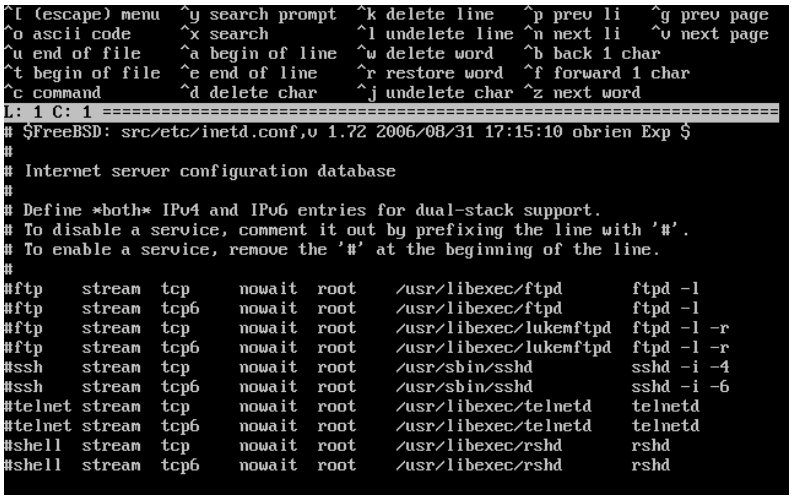
```
                                [ Yes ]   No
```

Select [Yes] to continue.

```
                        User Confirmation Requested
inetd(8) relies on its configuration file, /etc/inetd.conf, to determine
which of its Internet services will be available.  The default FreeBSD
inetd.conf(5) leaves all services disabled by default, so they must be
specifically enabled in the configuration file before they will
function, even once inetd(8) is enabled.  Note that services for
IPv6 must be separately enabled from IPv4 services.

Select [Yes] now to invoke an editor on /etc/inetd.conf, or [No] to
use the current settings.

                                [ Yes ]   No
```

Selecting [Yes] will allow adding services by deleting the # at the beginning of a line.

Figure 2-30. Editing `inetd.conf`



After adding the desired services, pressing **Esc** will display a menu which will allow exiting and saving the changes.

2.10.4 啓用SSH 登入

```
                        User Confirmation Requested
                    Would you like to enable SSH login?
                             Yes       [  No  ]
```

選擇[Yes] 就會啓用sshd(8)，也就是**OpenSSH** 的daemon 程式。這會允許該機器可從遠端安全登入。關於**OpenSSH** 請參閱Section 14.11 部分的說明。

2.10.5 Anonymous FTP

```
                    User Confirmation Requested
 Do you want to have anonymous FTP access to this machine?

                                Yes    [ No ]
```

2.10.5.1 Deny Anonymous FTP

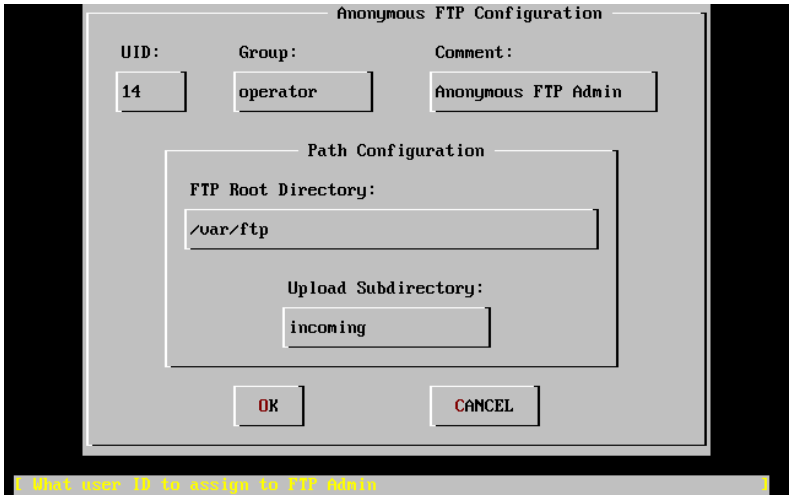
Selecting the default [No] and pressing **Enter** will still allow users who have accounts with passwords to use FTP to access the machine.

2.10.5.2 Allow Anonymous FTP

Anyone can access your machine if you elect to allow anonymous FTP connections. The security implications should be considered before enabling this option. For more information about security see Chapter 14.

To allow anonymous FTP, use the arrow keys to select [Yes] and press **Enter**. The following screen (or similar) will display:

Figure 2-31. Default Anonymous FTP Configuration



Pressing **F1** will display the help:

```
This screen allows you to configure the anonymous FTP user.

The following configuration values are editable:

UID:     The user ID you wish to assign to the anonymous FTP user.
         All files uploaded will be owned by this ID.

Group:   Which group you wish the anonymous FTP user to be in.

Comment: String describing this user in /etc/passwd
```

```
FTP Root Directory:

        Where files available for anonymous FTP will be kept.

Upload subdirectory:

        Where files uploaded by anonymous FTP users will go.
```

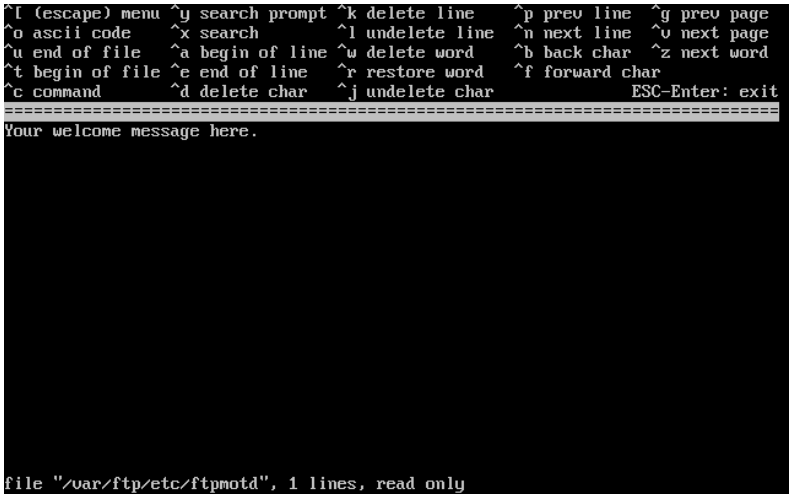
The ftp root directory will be put in `/var` by default. If you do not have enough room there for the anticipated FTP needs, the `/usr` directory could be used by setting the FTP Root Directory to `/usr/ftp`.

When you are satisfied with the values, press **Enter** to continue.

```
                        User Confirmation Requested
          Create a welcome message file for anonymous FTP users?

                                [ Yes ]    No
```

If you select [Yes] and press **Enter**, an editor will automatically start allowing you to edit the message.

Figure 2-32. Edit the FTP Welcome Message



This is a text editor called `ee`. Use the instructions to change the message or change the message later using a text editor of your choice. Note the file name/location at the bottom of the editor screen.

Press **Esc** and a pop-up menu will default to a) leave editor. Press **Enter** to exit and continue. Press **Enter** again to save changes if you made any.

2.10.6 Configure Network File System

Network File System (NFS) allows sharing of files across a network. A machine can be configured as a server, a client, or both. Refer to Section 27.3 for a more information.

2.10.6.1 NFS Server

```
                        User Confirmation Requested
 Do you want to configure this machine as an NFS server?

                                  Yes    [ No ]
```

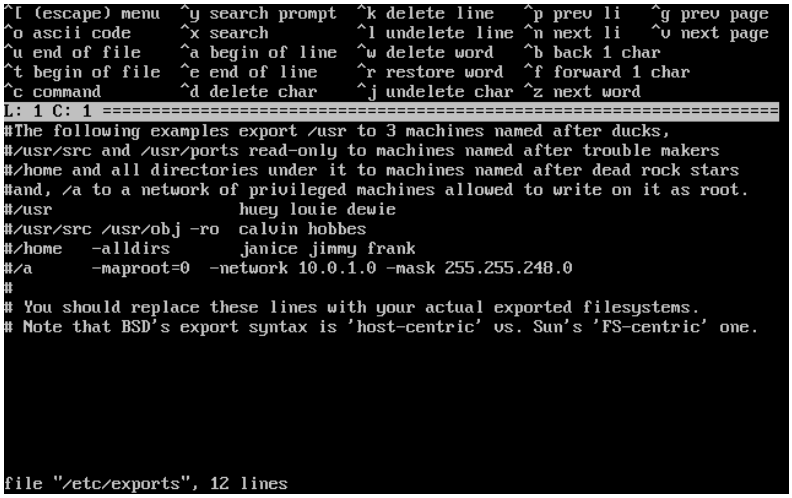
If there is no need for a Network File System server, select [No] and press **Enter**.

If [Yes] is chosen, a message will pop-up indicating that the `exports` file must be created.

```
                                 Message
Operating as an NFS server means that you must first configure an
/etc/exports file to indicate which hosts are allowed certain kinds of
access to your local filesystems.
Press [Enter] now to invoke an editor on /etc/exports
                                 [ OK ]
```

Press **Enter** to continue. A text editor will start allowing the `exports` file to be created and edited.

Figure 2-33. Editing `exports`



Use the instructions to add the actual exported filesystems now or later using a text editor of your choice. Note the file name/location at the bottom of the editor screen.

Press **Esc** and a pop-up menu will default to a) leave editor. Press **Enter** to exit and continue.

2.10.6.2 NFS Client

The NFS client allows your machine to access NFS servers.

```
                        User Confirmation Requested
 Do you want to configure this machine as an NFS client?

                                  Yes   [ No ]
```

With the arrow keys, select [Yes] or [No] as appropriate and press **Enter**.

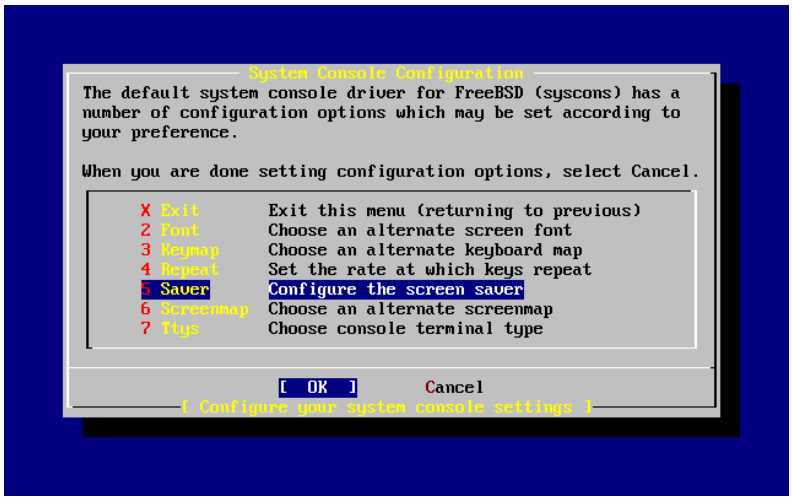
2.10.7 System Console Settings

There are several options available to customize the system console.

```
                    User Confirmation Requested
      Would you like to customize your system console settings?

                                [ Yes ]  No
```

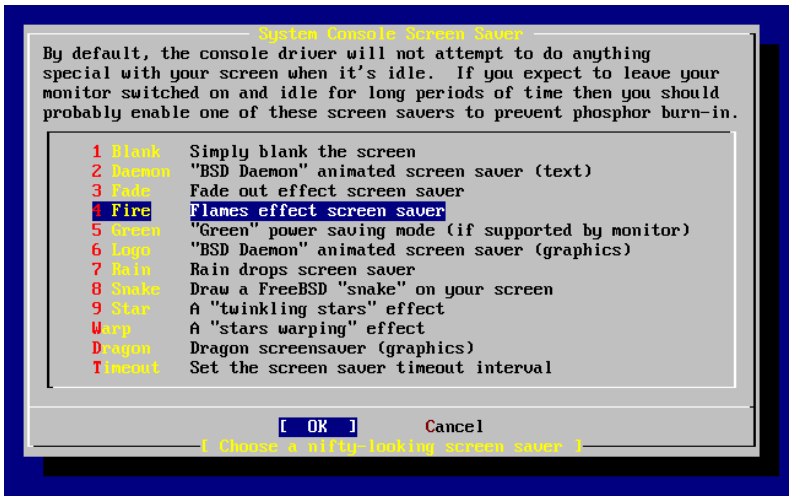
To view and configure the options, select [Yes] and press **Enter**.

Figure 2-34. System Console Configuration Options



A commonly used option is the screen saver. Use the arrow keys to select Saver and then press **Enter**.

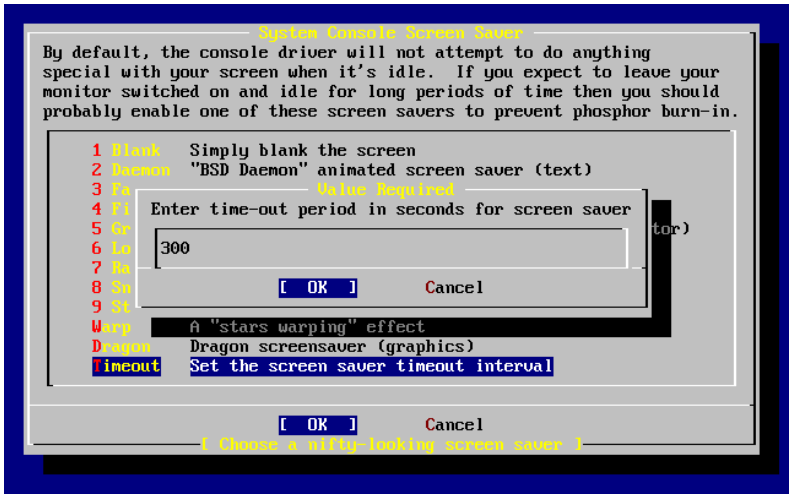
Figure 2-35. Screen Saver Options



Select the desired screen saver using the arrow keys and then press **Enter**. The System Console Configuration menu will redisplay.

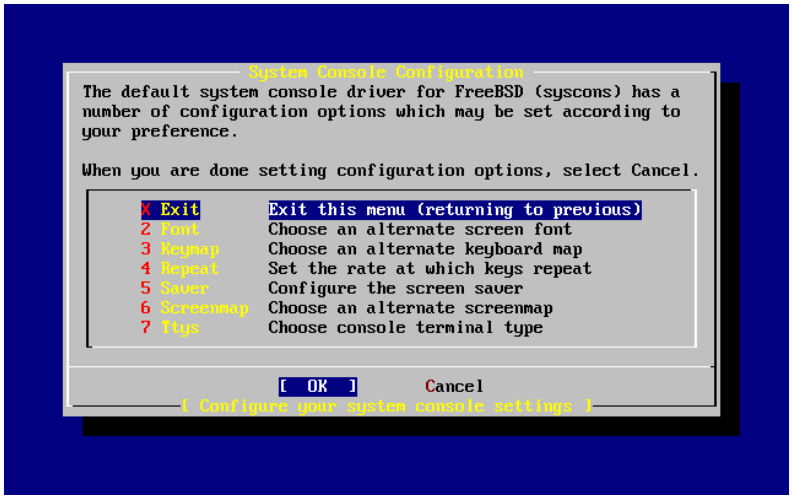
The default time interval is 300 seconds. To change the time interval, select Saver again. At the Screen Saver Options menu, select Timeout using the arrow keys and press **Enter**. A pop-up menu will appear:

Figure 2-36. Screen Saver Timeout



The value can be changed, then select [OK] and press **Enter** to return to the System Console Configuration menu.

Figure 2-37. System Console Configuration Exit



Selecting Exit and pressing **Enter** will continue with the post-installation configurations.

2.10.8 Setting the Time Zone

Setting the time zone for your machine will allow it to automatically correct for any regional time changes and perform other time zone related functions properly.

The example shown is for a machine located in the Eastern time zone of the United States. Your selections will vary according to your geographical location.

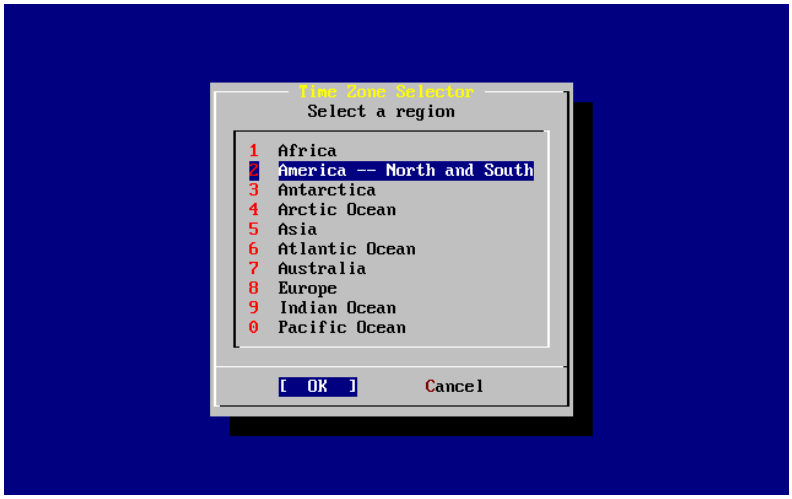
```
                  User Confirmation Requested
        Would you like to set this machine's time zone now?

                          [ Yes ]   No
```

Select [Yes] and press **Enter** to set the time zone.

```
                       User Confirmation Requested
 Is this machine's CMOS clock set to UTC? If it is set to local time
 or you don't know, please choose NO here!

                                Yes   [ No ]
```

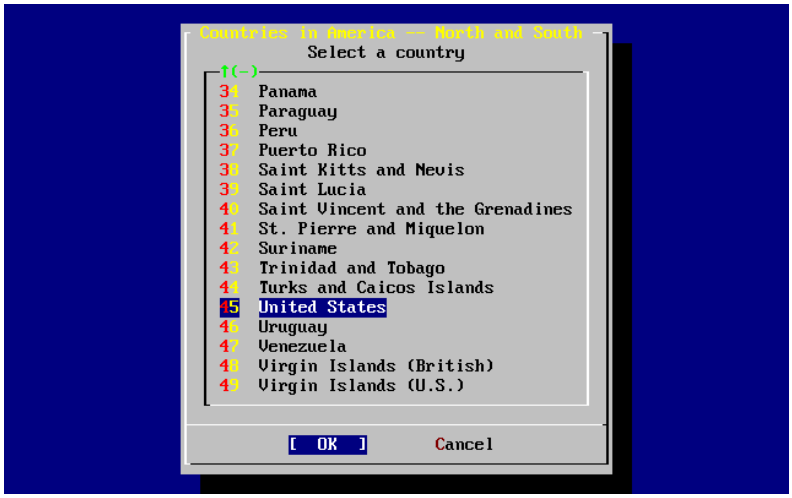
Select [Yes] or [No] according to how the machine's clock is configured and press **Enter**.

Figure 2-38. Select Your Region



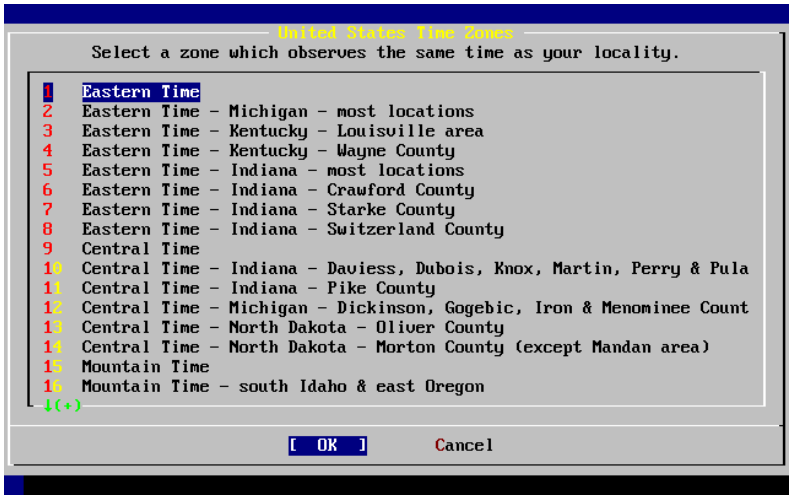
The appropriate region is selected using the arrow keys and then pressing **Enter**.

Figure 2-39. Select Your Country



Select the appropriate country using the arrow keys and press **Enter**.

Figure 2-40. Select Your Time Zone



The appropriate time zone is selected using the arrow keys and pressing **Enter**.

```
                    Confirmation
        Does the abbreviation 'EDT' look reasonable?

                    [ Yes ]   No
```

Confirm the abbreviation for the time zone is correct. If it looks okay, press **Enter** to continue with the post-installation configuration.

2.10.9 Linux Compatibility

```
                  User Confirmation Requested
      Would you like to enable Linux binary compatibility?

                    [ Yes ]   No
```

Selecting [Yes] and pressing **Enter** will allow running Linux software on FreeBSD. The install will add the appropriate packages for Linux compatibility.

If installing by FTP, the machine will need to be connected to the Internet. Sometimes a remote ftp site will not have all the distributions like the Linux binary compatibility. This can be installed later if necessary.

2.10.10 Mouse Settings

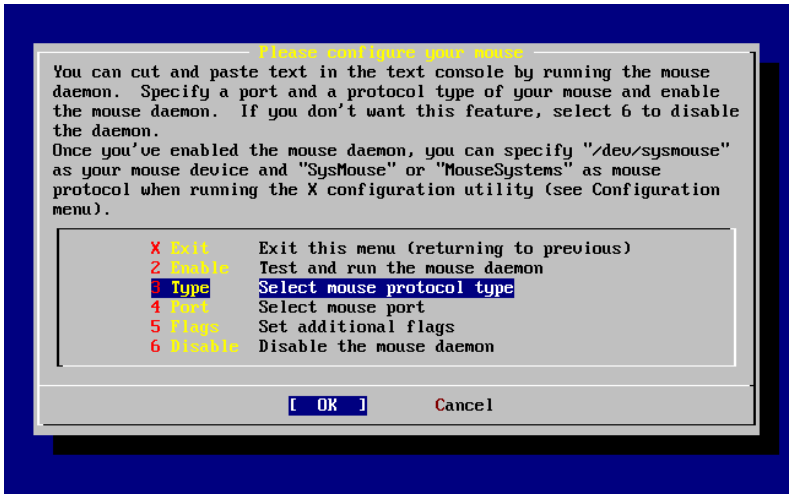
This option will allow you to cut and paste text in the console and user programs with a 3-button mouse. If using a 2-button mouse, refer to manual page, mouse(8), after installation for details on emulating the 3-button style. This example depicts a non-USB mouse configuration (such as a PS/2 or COM port mouse):

```
                  User Confirmation Requested
      Does this system have a PS/2, serial, or bus mouse?
```

```
[ Yes ]    No
```

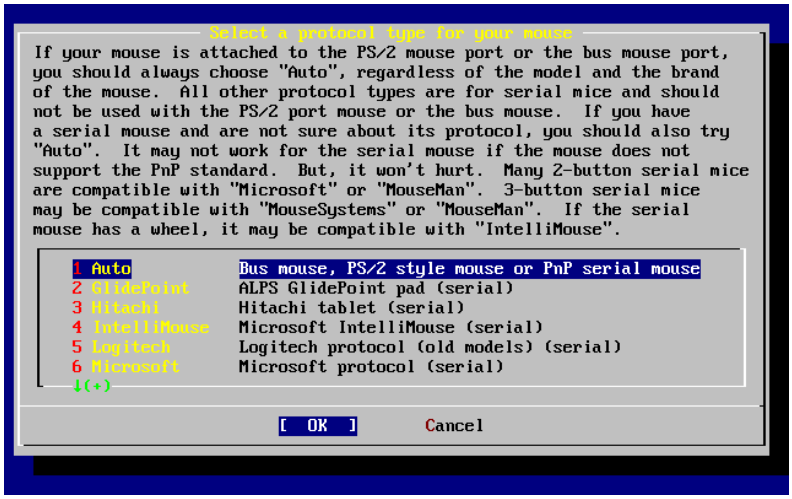
Select [Yes] for a PS/2, serial, or bus mouse, or [No] for a USB mouse and press **Enter**.

Figure 2-41. Select Mouse Protocol Type



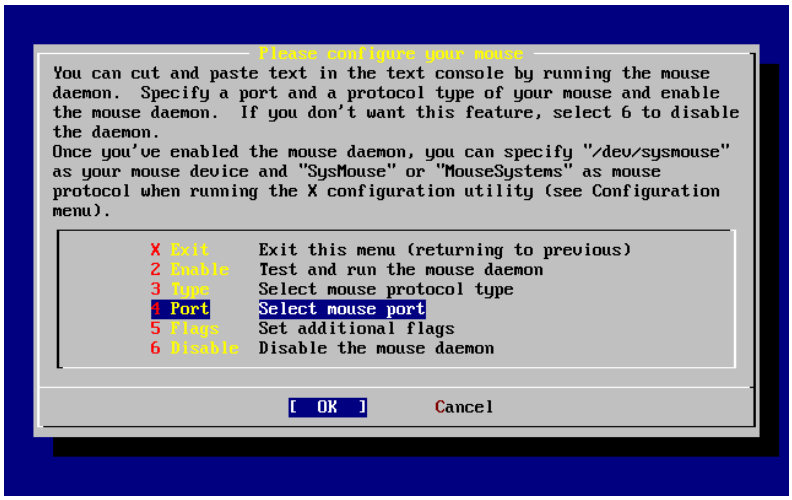
Use the arrow keys to select Type and press **Enter**.

Figure 2-42. Set Mouse Protocol



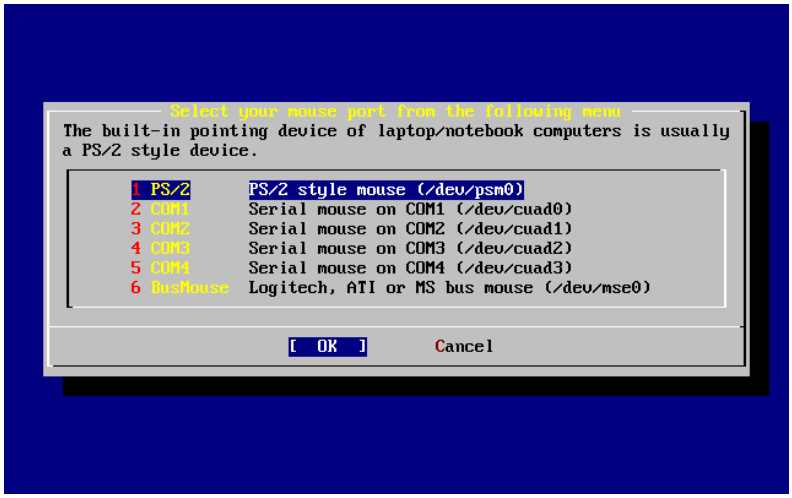
The mouse used in this example is a PS/2 type, so the default Auto was appropriate. To change protocol, use the arrow keys to select another option. Ensure that [OK] is highlighted and press **Enter** to exit this menu.

Figure 2-43. Configure Mouse Port



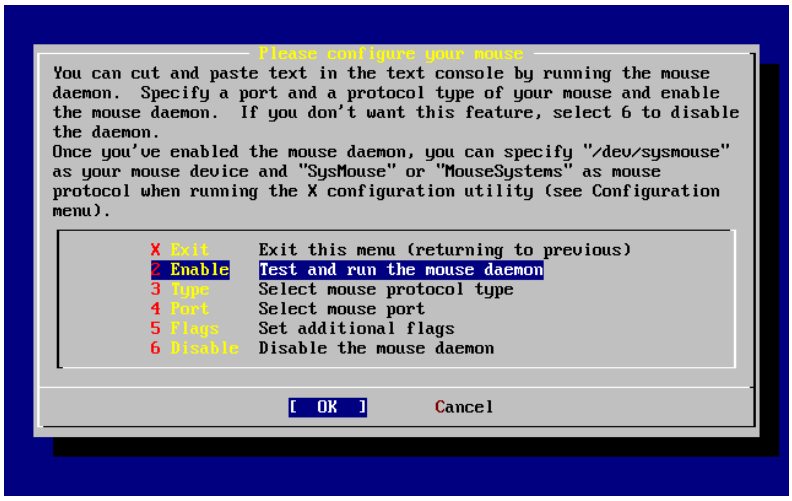
Use the arrow keys to select Port and press **Enter**.

Figure 2-44. Setting the Mouse Port



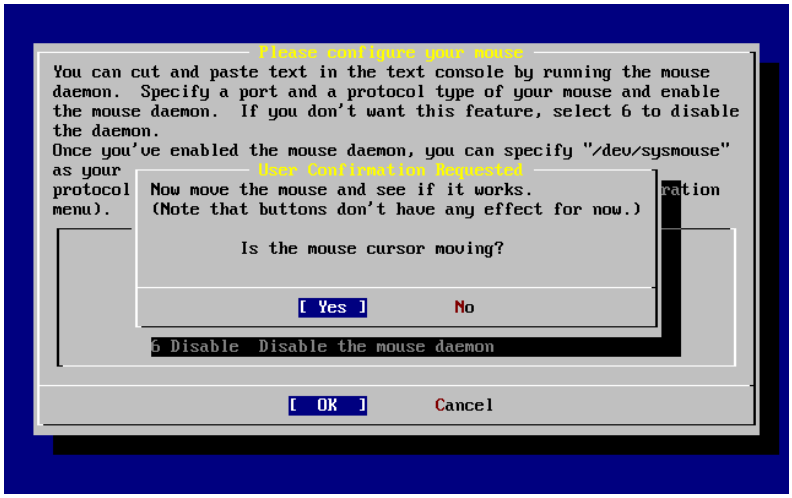
This system had a PS/2 mouse, so the default PS/2 was appropriate. To change the port, use the arrow keys and then press **Enter**.

Figure 2-45. Enable the Mouse Daemon



Last, use the arrow keys to select Enable, and press **Enter** to enable and test the mouse daemon.

Figure 2-46. Test the Mouse Daemon



Move the mouse around the screen and verify the cursor shown responds properly. If it does, select [Yes] and press **Enter**. If not, the mouse has not been configured correctly ——select [No] and try using different configuration options.

Select Exit with the arrow keys and press **Enter** to return to continue with the post-installation configuration.

2.10.11 Install Packages

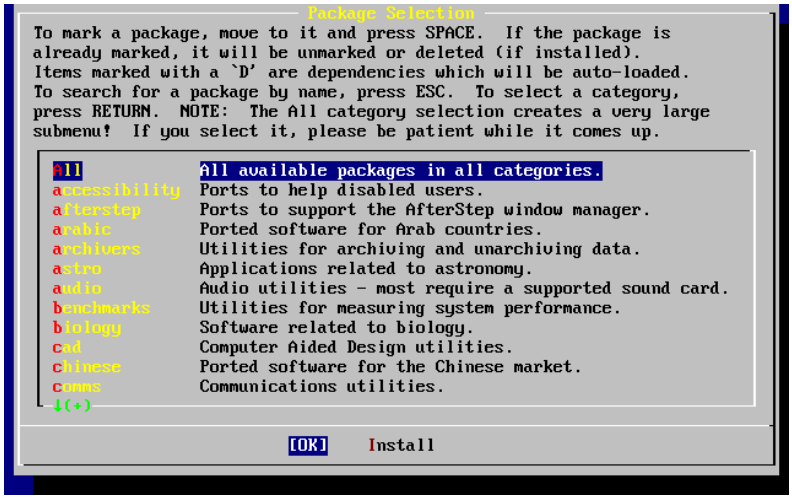
Packages are pre-compiled binaries and are a convenient way to install software.

Installation of one package is shown for purposes of illustration. Additional packages can also be added at this time if desired. After installation `sysinstall` can be used to add additional packages.

```
                    User Confirmation Requested
The FreeBSD package collection is a collection of hundreds of
ready-to-run applications, from text editors to games to WEB servers
and more. Would you like to browse the collection now?

                                [ Yes ]   No
```

Selecting [Yes] and pressing **Enter** will be followed by the Package Selection screens:

Figure 2-47. Select Package Category

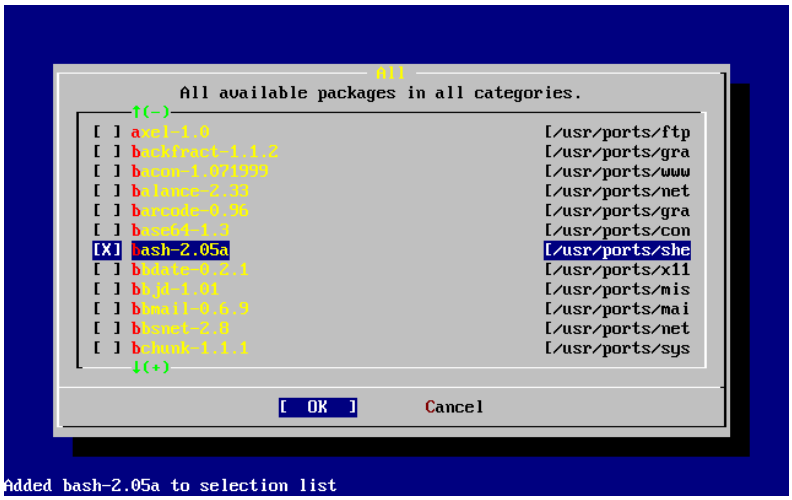


Only packages on the current installation media are available for installation at any given time.

All packages available will be displayed if All is selected or you can select a particular category. Highlight your selection with the arrow keys and press **Enter**.

A menu will display showing all the packages available for the selection made:

Figure 2-48. Select Packages



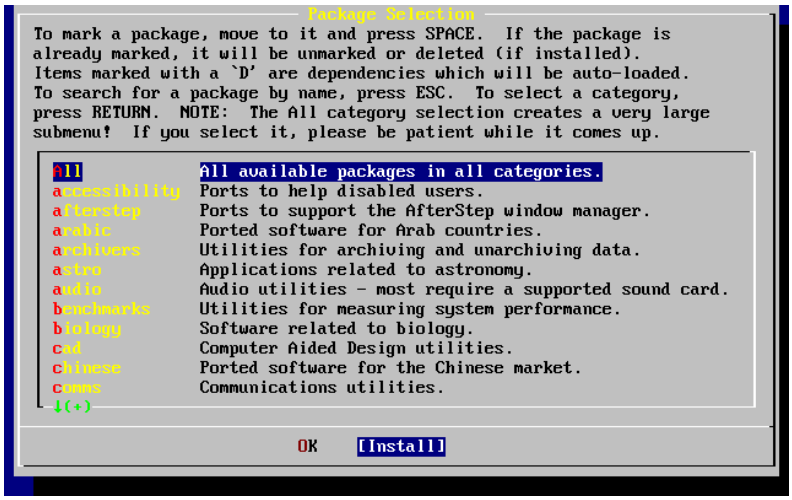
The **bash** shell is shown selected. Select as many as desired by highlighting the package and pressing the **Space** key. A short description of each package will appear in the lower left corner of the screen.

Pressing the **Tab** key will toggle between the last selected package, [OK], and [Cancel].

When you have finished marking the packages for installation, press **Tab** once to toggle to the [OK] and press **Enter** to return to the Package Selection menu.

The left and right arrow keys will also toggle between [OK] and [Cancel]. This method can also be used to select [OK] and press **Enter** to return to the Package Selection menu.

Figure 2-49. Install Packages



Use the **Tab** and arrow keys to select [Install] and press **Enter**. You will then need to confirm that you want to install the packages:

Figure 2-50. Confirm Package Installation

Selecting [OK] and pressing **Enter** will start the package installation. Installing messages will appear until completed. Make note if there are any error messages.

The final configuration continues after packages are installed. If you end up not selecting any packages, and wish to return to the final configuration, select Install anyways.

2.10.12 Add Users/Groups

You should add at least one user during the installation so that you can use the system without being logged in as `root`. The root partition is generally small and running applications as `root` can quickly fill it. A bigger danger is noted below:

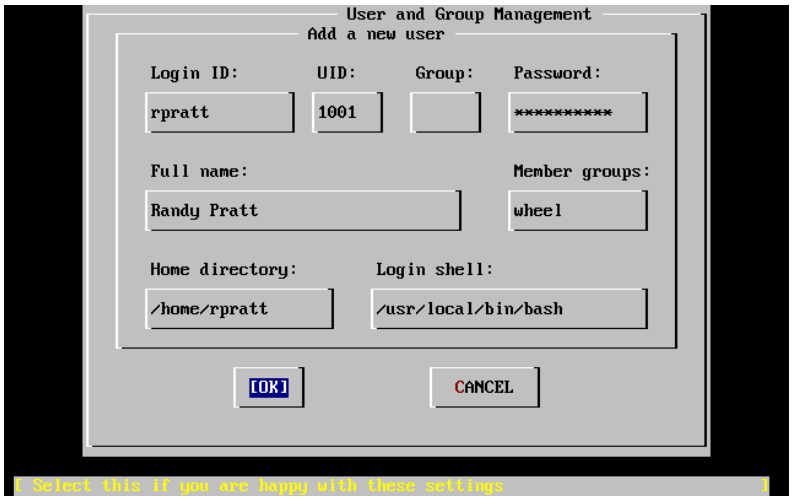
```
                    User Confirmation Requested
Would you like to add any initial user accounts to the system? Adding
at least one account for yourself at this stage is suggested since
working as the "root" user is dangerous (it is easy to do things which
adversely affect the entire system).

                              [ Yes ]   No
```

Select [Yes] and press **Enter** to continue with adding a user.

Figure 2-51. Select User

Select User with the arrow keys and press **Enter**.

Figure 2-52. Add User Information



The following descriptions will appear in the lower part of the screen as the items are selected with **Tab** to assist with entering the required information:

Login ID

The login name of the new user (mandatory).

UID

The numerical ID for this user (leave blank for automatic choice).

Group

The login group name for this user (leave blank for automatic choice).

Password

The password for this user (enter this field with care!).

Full name

The user’s full name (comment).

Member groups

The groups this user belongs to (i.e. gets access rights for).

Home directory

The user’s home directory (leave blank for default).

Login shell

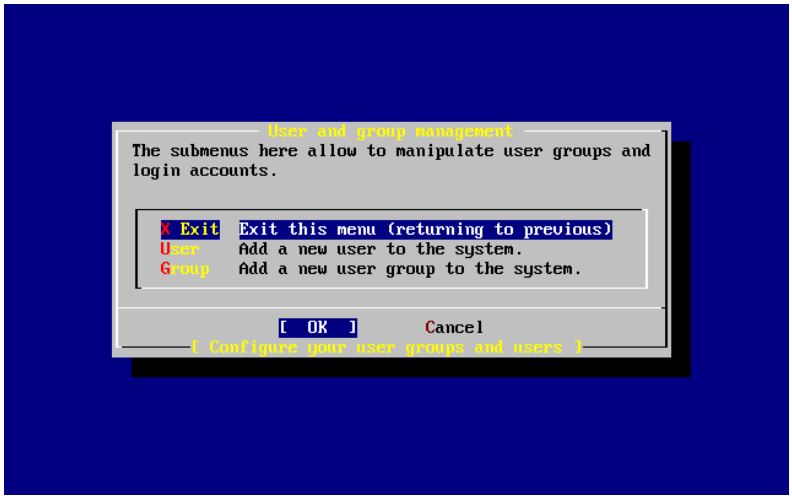
The user’s login shell (leave blank for default, e.g. `/bin/sh`).

The login shell was changed from `/bin/sh` to `/usr/local/bin/bash` to use the **bash** shell that was previously installed as a package. Do not try to use a shell that does not exist or you will not be able to login. The most common shell used in the BSD-world is the C shell, which can be indicated as `/bin/tcsh`.

The user was also added to the `wheel` group to be able to become a superuser with `root` privileges.

When you are satisfied, press [OK] and the User and Group Management menu will redisplay:

Figure 2-53. Exit User and Group Management



Groups can also be added at this time if specific needs are known. Otherwise, this may be accessed through using `sysinstall` (`/stand/sysinstall` in FreeBSD versions older than 5.2) after installation is completed.

When you are finished adding users, select Exit with the arrow keys and press **Enter** to continue the installation.

2.10.13 Set the `root` Password

```
                        Message
 Now you must set the system manager's password.
 This is the password you'll use to log in as "root".

                         [ OK ]

                [ Press enter or space ]
```

Press **Enter** to set the `root` password.

The password will need to be typed in twice correctly. Needless to say, make sure you have a way of finding the password if you forget. Notice that the password you type in is not echoed, nor are asterisks displayed.

```
New password :
Retype new password :
```

The installation will continue after the password is successfully entered.

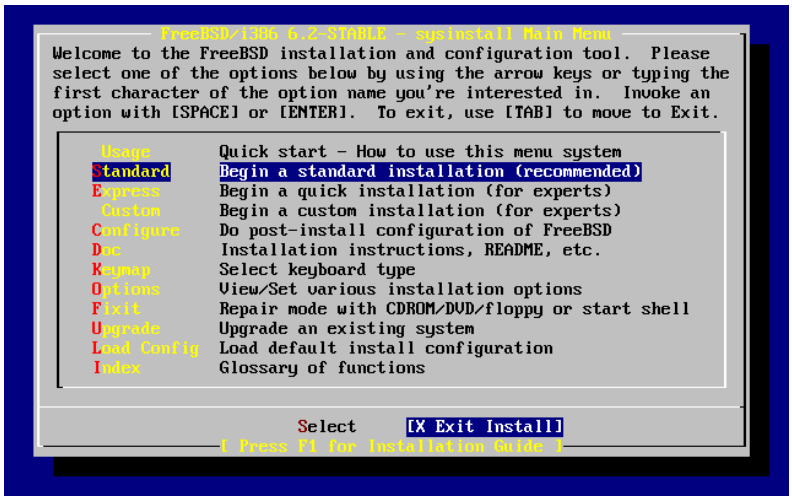
2.10.14 Exiting Install

If you need to configure additional network devices or any other configuration, you can do it at this point or after installation with `sysinstall` (`/stand/sysinstall` in FreeBSD versions older than 5.2).

```
                    User Confirmation Requested
Visit the general configuration menu for a chance to set any last
options?

                                Yes   [ No ]
```

Select [No] with the arrow keys and press **Enter** to return to the Main Installation Menu.

Figure 2-54. Exit Install



Select [X Exit Install] with the arrow keys and press **Enter**. You will be asked to confirm exiting the installation:

```
                    User Confirmation Requested
Are you sure you wish to exit? The system will reboot (be sure to
remove any floppies/CDs/DVDs from the drives).

                               [ Yes ]   No
```

Select [Yes] and remove the floppy if booting from the floppy. The CDROM drive is locked until the machine starts to reboot. The CDROM drive is then unlocked and the disk can be removed from drive (quickly).

The system will reboot so watch for any error messages that may appear, see Section 2.10.16 details.

2.10.15 Configure Additional Network Services

Contributed by Tom Rhodes.

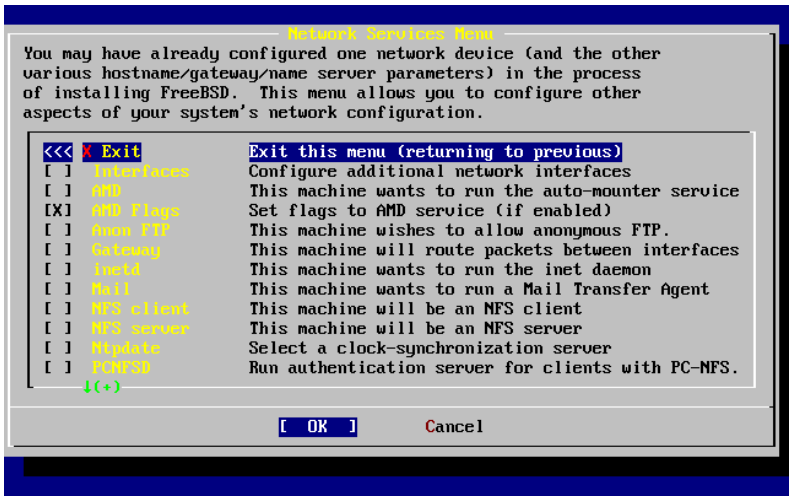
Configuring network services can be a daunting task for new users if they lack previous knowledge in this area. Networking, including the Internet, is critical to all modern operating systems including FreeBSD; as a result, it is

very useful to have some understanding FreeBSD's extensive networking capabilities. Doing this during the installation will ensure users have some understanding of the various services available to them.

Network services are programs that accept input from anywhere on the network. Every effort is made to make sure these programs will not do anything "harmful". Unfortunately, programmers are not perfect and through time there have been cases where bugs in network services have been exploited by attackers to do bad things. It is important that you only enable the network services you know that you need. If in doubt it is best if you do not enable a network service until you find out that you do need it. You can always enable it later by re-running **sysinstall** or by using the features provided by the `/etc/rc.conf` file.

Selecting the Networking option will display a menu similar to the one below:

Figure 2-55. Network Configuration Upper-level



The first option, Interfaces, was previously covered during the Section 2.10.1, thus this option can safely be ignored.

Selecting the AMD option adds support for the BSD automatic mount utility. This is usually used in conjunction with the NFS protocol (see below) for automatically mounting remote file systems. No special configuration is required here.

Next in line is the AMD Flags option. When selected, a menu will pop up for you to enter specific AMD flags. The menu already contains a set of default options:

```
-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map
```

The `-a` option sets the default mount location which is specified here as `/.amd_mnt`. The `-l` option specifies the default `log` file; however, when `syslogd` is used all log activity will be sent to the system log daemon. The `/host` directory is used to mount an exported file system from a remote host, while `/net` directory is used to mount an exported file system from an IP address. The `/etc/amd.map` file defines the default options for AMD exports.

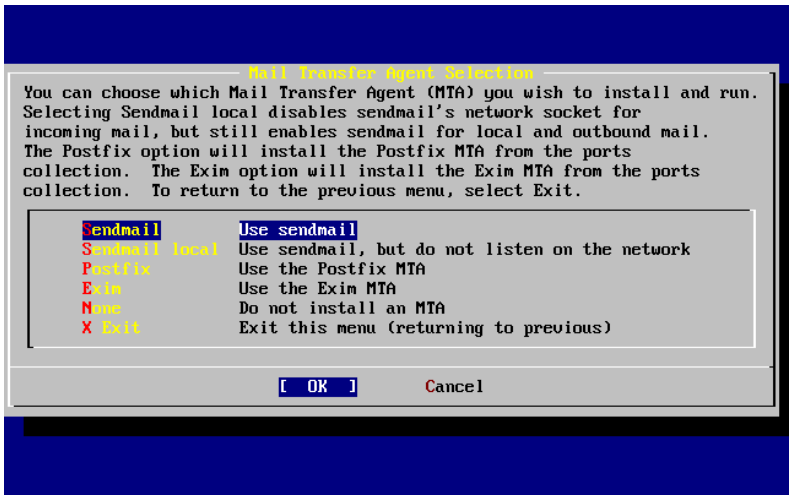
The Anon FTP option permits anonymous FTP connections. Select this option to make this machine an anonymous FTP server. Be aware of the security risks involved with this option. Another menu will be displayed to explain the security risks and configuration in depth.

The Gateway configuration menu will set the machine up to be a gateway as explained previously. This can be used to unset the Gateway option if you accidentally selected it during the installation process.

The Inetd option can be used to configure or completely disable the inetd(8) daemon as discussed above.

The Mail option is used to configure the system's default MTA or Mail Transfer Agent. Selecting this option will bring up the following menu:

Figure 2-56. Select a default MTA



Here you are offered a choice as to which MTA to install and set as the default. An MTA is nothing more than a mail server which delivers email to users on the system or the Internet.

Selecting Sendmail will install the popular **sendmail** server which is the FreeBSD default. The Sendmail local option will set **sendmail** to be the default MTA, but disable its ability to receive incoming email from the Internet. The other options here, Postfix and Exim act similar to Sendmail. They both deliver email; however, some users prefer these alternatives to the **sendmail** MTA.

After selecting an MTA, or choosing not to select an MTA, the network configuration menu will appear with the next option being NFS client.

The NFS client option will configure the system to communicate with a server via NFS. An NFS server makes file systems available to other machines on the network via the NFS protocol. If this is a stand-alone machine, this option can remain unselected. The system may require more configuration later; see Section 27.3 for more information about client and server configuration.

Below that option is the NFS server option, permitting you to set the system up as an NFS server. This adds the required information to start up the RPC remote procedure call services. RPC is used to coordinate connections between hosts and programs.

Next in line is the Ntpdate option, which deals with time synchronization. When selected, a menu like the one below shows up:

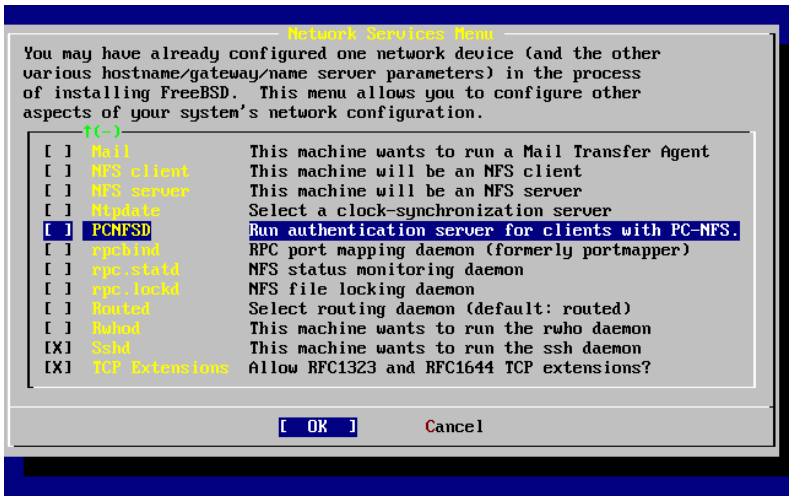
Figure 2-57. Ntpdate Configuration

From this menu, select the server which is the closest to your location. Selecting a close one will make the time synchronization more accurate as a server further from your location may have more connection latency.

The next option is the PCNFSD selection. This option will install the `net/pcnfsd` package from the Ports Collection. This is a useful utility which provides NFS authentication services for systems which are unable to provide their own, such as Microsoft's MS-DOS operating system.

Now you must scroll down a bit to see the other options:

Figure 2-58. Network Configuration Lower-level



The rpcbind(8), rpc.statd(8), and rpc.lockd(8) utilities are all used for Remote Procedure Calls (RPC). The `rpcbind` utility manages communication between NFS servers and clients, and is required for NFS servers to operate correctly. The **rpc.statd** daemon interacts with the **rpc.statd** daemon on other hosts to provide status monitoring. The reported status is usually held in the `/var/db/statd.status` file. The next option listed here is the rpc.lockd option, which, when selected, will provide file locking services. This is usually used with **rpc.statd** to monitor what

hosts are requesting locks and how frequently they request them. While these last two options are marvelous for debugging, they are not required for NFS servers and clients to operate correctly.

As you progress down the list the next item here is Routed, which is the routing daemon. The routed(8) utility manages network routing tables, discovers multicast routers, and supplies a copy of the routing tables to any physically connected host on the network upon request. This is mainly used for machines which act as a gateway for the local network. When selected, a menu will be presented requesting the default location of the utility. The default location is already defined for you and can be selected with the **Enter** key. You will then be presented with yet another menu, this time asking for the flags you wish to pass on to **routed**. The default is `-q` and it should already appear on the screen.

Next in line is the Rwhod option which, when selected, will start the rwhod(8) daemon during system initialization. The `rwhod` utility broadcasts system messages across the network periodically, or collects them when in "consumer" mode. More information can be found in the ruptime(1) and rwho(1) manual pages.

The next to the last option in the list is for the sshd(8) daemon. This is the secure shell server for **OpenSSH** and it is highly recommended over the standard **telnet** and FTP servers. The **sshd** server is used to create a secure connection from one host to another by using encrypted connections.

Finally there is the TCP Extensions option. This enables the TCP Extensions defined in RFC 1323 and RFC 1644. While on many hosts this can speed up connections, it can also cause some connections to be dropped. It is not recommended for servers, but may be beneficial for stand alone machines.

Now that you have configured the network services, you can scroll up to the very top item which is X Exit and continue on to the next configuration item or simply exit **sysinstall** in selecting X Exit twice then [X Exit Install].

2.10.16 FreeBSD 開機流程

2.10.16.1 FreeBSD/i386 的開機流程

If everything went well, you will see messages scroll off the screen and you will arrive at a login prompt. You can view the content of the messages by pressing **Scroll-Lock** and using **PgUp** and **PgDn**. Pressing **Scroll-Lock** again will return to the prompt.

The entire message may not display (buffer limitation) but it can be viewed from the command line after logging in by typing `dmesg` at the prompt.

Login using the username/password you set during installation (`rpratt`, in this example). Avoid logging in as `root` except when necessary.

Typical boot messages (version information omitted):

```
Copyright (c) 1992-2002 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.

Timecounter "i8254"  frequency 1193182 Hz
CPU: AMD-K6(tm) 3D processor (300.68-MHz 586-class CPU)
  Origin = "AuthenticAMD"  Id = 0x580  Stepping = 0
  Features=0x8001bf<FPU,VME,DE,PSE,TSC,MSR,MCE,CX8,MMX>
  AMD Features=0x80000800<SYSCALL,3DNow!>
real memory  = 268435456 (262144K bytes)
config> di sn0
```

```
config> di lnc0
config> di le0
config> di ie0
config> di fe0
config> di cs0
config> di bt0
config> di aic0
config> di aha0
config> di adv0
config> q
avail memory = 256311296 (250304K bytes)
Preloaded elf kernel "kernel" at 0xc0491000.
Preloaded userconfig_script "/boot/kernel.conf" at 0xc049109c.
md0: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1: <VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device 1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device 7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0: <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device 7.2 on pci0
usb0: <VIA 83C572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr 1
uhub0: 2 ports with 2 removable, self powered
chip1: <VIA 82C586B ACPI interface> at device 7.3 on pci0
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xe800-0xe81f irq 9 at
device 10.0 on pci0
ed0: address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq 2 on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
atkbdc0: <keyboard controller (i8042)> at port 0x60-0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq 1 on atkbdc0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: model Generic PS/2 mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
sc0: <System console> at flags 0x1 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
```

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
ppbus0: IEEE1284 device found /NIBBLE
Probing for PnP devices on ppbus0:
plip0: <PLIP network interface> on ppbus0
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
ppi0: <Parallel I/O> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master using UDMA33
ad2: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata1-master using UDMA33
acd0: CDROM <DELTA OTC-H101/ST3 F/W by OIPD> at ata0-slave using PIO4
Mounting root from ufs:/dev/ad0s1a
swapon: adding /dev/ad0s1b as swap device
Automatic boot in progress...
/dev/ad0s1a: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1a: clean, 48752 free (552 frags, 6025 blocks, 0.9% fragmentation)
/dev/ad0s1f: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1f: clean, 128997 free (21 frags, 16122 blocks, 0.0% fragmentation)
/dev/ad0s1g: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1g: clean, 3036299 free (43175 frags, 374073 blocks, 1.3% fragmentation)
/dev/ad0s1e: filesystem CLEAN; SKIPPING CHECKS
/dev/ad0s1e: clean, 128193 free (17 frags, 16022 blocks, 0.0% fragmentation)
Doing initial network setup: hostname.
ed0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        inet6 fe80::5054::5ff::fede:731b%ed0 prefixlen 64 tentative scopeid 0x1
        ether 52:54:05:de:73:1b
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x8
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
Additional routing options: IP gateway=YES TCP keepalive=YES
routing daemons:.
additional daemons: syslogd.
Doing additional network setup:.
Starting final network daemons: creating ssh RSA host key
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
cd:76:89:16:69:0e:d0:6e:f8:66:d0:07:26:3c:7e:2d root@k6-2.example.com
 creating ssh DSA host key
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
f9:a1:a9:47:c4:ad:f9:8d:52:b8:b8:ff:8c:ad:2d:e6 root@k6-2.example.com.
setting ELF ldconfig path: /usr/lib /usr/lib/compat /usr/X11R6/lib
/usr/local/lib
a.out ldconfig path: /usr/lib/aout /usr/lib/compat/aout /usr/X11R6/lib/aout
starting standard daemons: inetd cron sshd usbd sendmail.
Initial rc.i386 initialization:.
```

```
rc.i386 configuring syscons: blank_time screensaver moused.
Additional ABI support: linux.
Local package initialization:.
Additional TCP options:.

FreeBSD/i386 (k6-2.example.com) (ttyv0)

login: rpratt
Password:
```

Generating the RSA and DSA keys may take some time on slower machines. This happens only on the initial boot-up of a new installation. Subsequent boots will be faster.

If the X server has been configured and a Default Desktop chosen, it can be started by typing `startx` at the command line.

2.10.16.2 FreeBSD/alpha 開機流程

Once the install procedure has finished, you will be able to start FreeBSD by typing something like this to the SRM prompt:

```
>>>BOOT DKC0
```

This instructs the firmware to boot the specified disk. To make FreeBSD boot automatically in the future, use these commands:

```
>>> SET BOOT_OSFLAGS A
>>> SET BOOT_FILE ”
>>> SET BOOTDEF_DEV DKC0
>>> SET AUTO_ACTION BOOT
```

The boot messages will be similar (but not identical) to those produced by FreeBSD booting on the i386.

2.10.17 FreeBSD Shutdown

It is important to properly shutdown the operating system. Do not just turn off power. First, become a superuser by typing `su` at the command line and entering the `root` password. This will work only if the user is a member of the `wheel` group. Otherwise, login as `root` and use `shutdown -h now`.

```
The operating system has halted.
Please press any key to reboot.
```

It is safe to turn off the power after the shutdown command has been issued and the message “Please press any key to reboot” appears. If any key is pressed instead of turning off the power switch, the system will reboot.

You could also use the **Ctrl**+**Alt**+**Del** key combination to reboot the system, however this is not recommended during normal operation.

2.11 安裝的疑難雜症解決

The following section covers basic installation troubleshooting, such as common problems people have reported. There are also a few questions and answers for people wishing to dual-boot FreeBSD with MS-DOS or Windows.

2.11.1 What to Do If Something Goes Wrong

Due to various limitations of the PC architecture, it is impossible for probing to be 100% reliable, however, there are a few things you can do if it fails.

Check the Hardware Notes (http://www.FreeBSD.org/releases/index.html) document for your version of FreeBSD to make sure your hardware is supported.

若硬體有在支援清單內，但使用`GENERIC` kernel 仍有問題，那麼就可能需要自訂kernel，以加入有支援的硬體。The kernel on the boot disks is configured assuming that most hardware devices are in their factory default configuration in terms of IRQs, IO addresses, and DMA channels. If your hardware has been reconfigured, you will most likely need to edit the kernel configuration and recompile to tell FreeBSD where to find things.

It is also possible that a probe for a device not present will cause a later probe for another device that is present to fail. In that case, the probes for the conflicting driver(s) should be disabled.

> **Note:** Some installation problems can be avoided or alleviated by updating the firmware on various hardware components, most notably the motherboard. The motherboard firmware may also be referred to as BIOS and most of the motherboard or computer manufactures have a website where the upgrades and upgrade information may be located.
>
> Most manufacturers strongly advise against upgrading the motherboard BIOS unless there is a good reason for doing so, which could possibly be a critical update of sorts. The upgrade process *can* go wrong, causing permanent damage to the BIOS chip.

2.11.2 Using MS-DOS® and Windows® File Systems

At this time, FreeBSD does not support file systems compressed with the **Double Space™** application. Therefore the file system will need to be uncompressed before FreeBSD can access the data. This can be done by running the **Compression Agent** located in the Start> Programs > System Tools menu.

FreeBSD can support MS-DOS based file systems(FAT16 and FAT32). This requires you use the mount_msdosfs(8) command with the required parameters. The utility most common usage is:

```
# mount -t msdosfs /dev/ad0s1 /mnt
```

In this example, the MS-DOS file system is located on the first partition of the primary hard disk. Your situation may be different, check the output from the `dmesg`, and `mount` commands. They should produce enough information to give an idea of the partition layout.

> **Note:** Extended MS-DOS file systems are usually mapped after the FreeBSD partitions. In other words, the slice number may be higher than the ones FreeBSD is using. For instance, the first MS-DOS partition may be `/dev/ad0s1`, the FreeBSD partition may be `/dev/ad0s2`, with the extended MS-DOS partition being located on `/dev/ad0s3`. To some, this can be confusing at first.

NTFS partitions can also be mounted in a similar manner using the mount_ntfs(8) command.

2.11.3 Troubleshooting Questions and Answers

1. My system hangs while probing hardware during boot, or it behaves strangely during install, or the floppy drive isn't probed.

FreeBSD 5.0 and above makes extensive use of the system ACPI service on the i386, amd64 and ia64 platforms to aid in system configuration if it's detected during boot. Unfortunately, some bugs still exist in both the ACPI driver and within system motherboards and BIOS. The use of ACPI can be disabled by setting the `hint.acpi.0.disabled` hint in the third stage boot loader:

```
set hint.acpi.0.disabled="1"
```

This is reset each time the system is booted, so it is necessary to add `hint.acpi.0.disabled="1"` to the file `/boot/loader.conf`. More information about the boot loader can be found in Section 12.1.

2. I go to boot from the hard disk for the first time after installing FreeBSD, the kernel loads and probes my hardware, but stops with messages like:

```
changing root device to ad1s1a panic: cannot mount root
```

What is wrong? What can I do?

What is this `bios_drive:interface(unit,partition)kernel_name` thing that is displayed with the boot help?

There is a longstanding problem in the case where the boot disk is not the first disk in the system. The BIOS uses a different numbering scheme to FreeBSD, and working out which numbers correspond to which is difficult to get right.

In the case where the boot disk is not the first disk in the system, FreeBSD can need some help finding it. There are two common situations here, and in both of these cases, you need to tell FreeBSD where the root filesystem is. You do this by specifying the BIOS disk number, the disk type and the FreeBSD disk number for that type.

The first situation is where you have two IDE disks, each configured as the master on their respective IDE busses, and wish to boot FreeBSD from the second disk. The BIOS sees these as disk 0 and disk 1, while FreeBSD sees them as `ad0` and `ad2`.

FreeBSD is on BIOS disk 1, of type `ad` and the FreeBSD disk number is 2, so you would say:

```
1:ad(2,a)kernel
```

Note that if you have a slave on the primary bus, the above is not necessary (and is effectively wrong).

The second situation involves booting from a SCSI disk when you have one or more IDE disks in the system. In this case, the FreeBSD disk number is lower than the BIOS disk number. If you have two IDE disks as well as the SCSI disk, the SCSI disk is BIOS disk 2, type `da` and FreeBSD disk number 0, so you would say:

```
2:da(0,a)kernel
```

To tell FreeBSD that you want to boot from BIOS disk 2, which is the first SCSI disk in the system. If you only had one IDE disk, you would use '1:' instead.

Once you have determined the correct values to use, you can put the command exactly as you would have typed it in the `/boot.config` file using a standard text editor. Unless instructed otherwise, FreeBSD will use the contents of this file as the default response to the `boot:` prompt.

3. I go to boot from the hard disk for the first time after installing FreeBSD, but the Boot Manager prompt just prints `F?` at the boot menu each time but the boot won't go any further.

The hard disk geometry was set incorrectly in the Partition editor when you installed FreeBSD. Go back into the partition editor and specify the actual geometry of your hard disk. You must reinstall FreeBSD again from the beginning with the correct geometry.

If you are failing entirely in figuring out the correct geometry for your machine, here's a tip: Install a small DOS partition at the beginning of the disk and install FreeBSD after that. The install program will see the DOS partition and try to infer the correct geometry from it, which usually works.

The following tip is no longer recommended, but is left here for reference:

> If you are setting up a truly dedicated FreeBSD server or workstation where you don't care for (future) compatibility with DOS, Linux or another operating system, you've also got the option to use the entire disk ('A' in the partition editor), selecting the non-standard option where FreeBSD occupies the entire disk from the very first to the very last sector. This will leave all geometry considerations aside, but is somewhat limiting unless you're never going to run anything other than FreeBSD on a disk.

4. The system finds my ed(4) network card, but I keep getting device timeout errors.

Your card is probably on a different IRQ from what is specified in the `/boot/device.hints` file. The ed driver does not use the 'soft' configuration by default (values entered using EZSETUP in DOS), but it will use the software configuration if you specify `-1` in the hints for the interface.

Either move the jumper on the card to a hard configuration setting (altering the kernel settings if necessary), or specify the IRQ as `-1` by setting the hint "hint.ed.0.irq="-1"" This will tell the kernel to use the soft configuration.

Another possibility is that your card is at IRQ 9, which is shared by IRQ 2 and frequently a cause of problems (especially when you have a VGA card using IRQ 2!). You should not use IRQ 2 or 9 if at all possible.

2.12 進階安裝指南

Contributed by Valentino Vaschetto.

This section describes how to install FreeBSD in exceptional cases.

2.12.1 Installing FreeBSD on a System without a Monitor or Keyboard

This type of installation is called a "headless install", because the machine that you are trying to install FreeBSD on either does not have a monitor attached to it, or does not even have a VGA output. How is this possible you ask?

Using a serial console. A serial console is basically using another machine to act as the main display and keyboard for a system. To do this, just follow the steps to create installation floppies, explained in Section 2.3.7.

To modify these floppies to boot into a serial console, follow these steps:

1. Enabling the Boot Floppies to Boot into a Serial Console

   If you were to boot into the floppies that you just made, FreeBSD would boot into its normal install mode. We want FreeBSD to boot into a serial console for our install. To do this, you have to mount the `boot.flp` floppy onto your FreeBSD system using the mount(8) command.

   ```
   # mount /dev/fd0 /mnt
   ```

   Now that you have the floppy mounted, you must change into the `/mnt` directory:

   ```
   # cd /mnt
   ```

   Here is where you must set the floppy to boot into a serial console. You have to make a file called `boot.config` containing `/boot/loader -h`. All this does is pass a flag to the bootloader to boot into a serial console.

   ```
   # echo "/boot/loader -h" > boot.config
   ```

   Now that you have your floppy configured correctly, you must unmount the floppy using the umount(8) command:

   ```
   # cd /
   # umount /mnt
   ```

   Now you can remove the floppy from the floppy drive.

2. Connecting Your Null-modem Cable

   You now need to connect a null-modem cable between the two machines. Just connect the cable to the serial ports of the 2 machines. *A normal serial cable will not work here*, you need a null-modem cable because it has some of the wires inside crossed over.

3. Booting Up for the Install

   It is now time to go ahead and start the install. Put the `boot.flp` floppy in the floppy drive of the machine you are doing the headless install on, and power on the machine.

4. Connecting to Your Headless Machine

   Now you have to connect to that machine with cu(1):

   ```
   # cu -l /dev/cuad0
   ```

   在FreeBSD 5.X，請改用`/dev/cuaa0` 而非`/dev/cuad0`。

That's it! You should now be able to control the headless machine through your `cu` session. It will ask you to put in the `kern1.flp`, and then it will come up with a selection of what kind of terminal to use. Select the FreeBSD color console and proceed with your install!

2.13 製作安裝片

Note: 為避免重覆說明，在文中所提到的「FreeBSD 光碟」，在這裡指的是您所購買或自行燒錄的FreeBSD CDROM 或DVD。

There may be some situations in which you need to create your own FreeBSD installation media and/or source. This might be physical media, such as a tape, or a source that **sysinstall** can use to retrieve the files, such as a local FTP site, or an MS-DOS partition.

For example:

- You have many machines connected to your local network, and one FreeBSD disc. You want to create a local FTP site using the contents of the FreeBSD disc, and then have your machines use this local FTP site instead of needing to connect to the Internet.
- You have a FreeBSD disc, and FreeBSD does not recognize your CD/DVD drive, but MS-DOS/Windows does. You want to copy the FreeBSD installation files to a DOS partition on the same computer, and then install FreeBSD using those files.
- The computer you want to install on does not have a CD/DVD drive or a network card, but you can connect a "Laplink-style" serial or parallel cable to a computer that does.
- You want to create a tape that can be used to install FreeBSD.

2.13.1 Creating an Installation CDROM

As part of each release, the FreeBSD project makes available at least two CDROM images ("ISO images") per supported architecture. These images can be written ("burned") to CDs if you have a CD writer, and then used to install FreeBSD. If you have a CD writer, and bandwidth is cheap, then this is the easiest way to install FreeBSD.

1. Download the Correct ISO Images

   The ISO images for each release can be downloaded from `ftp://ftp.FreeBSD.org/pub/FreeBSD/ISO-IMAGES-`*`arch`*`/`*`version`* or the closest mirror. Substitute *`arch`* and *`version`* as appropriate.

   That directory will normally contain the following images:

   Table 2-4. FreeBSD 5.*X* and 6.*X* ISO Image Names and Meanings

檔名	內容
版本`-RELEASE-`*架構*`-bootonly.iso`	Everything you need to boot into a FreeBSD kernel and start the installation interface. The installable files have to be pulled over FTP or some other supported source.
版本`-RELEASE-`*架構*`-disc1.iso`	Everything you need to install FreeBSD and a "live filesystem", which is used in conjunction with the "Repair" facility in **sysinstall**.
版本`-RELEASE-`*架構*`-disc2.iso`	FreeBSD 文件(FreeBSD 6.2 之前的)，以及許多third-party packages。
版本`-RELEASE-`*架構*`-docs.iso`	FreeBSD 文件(FreeBSD 6.2 及之後)。

   You *must* download one of either the bootonly ISO image (if available), or the image of disc one. Do not download both of them, since the disc one image contains everything that the bootonly ISO image contains.

Use the bootonly ISO if Internet access is cheap for you. It will let you install FreeBSD, and you can then install third-party packages by downloading them using the ports/packages system (see Chapter 4) as necessary.

Use the image of disc one if you want to install a FreeBSD release and want a reasonable selection of third-party packages on the disc as well.

The additional disc images are useful, but not essential, especially if you have high-speed access to the Internet.

2. Write the CDs

   You must then write the CD images to disc. If you will be doing this on another FreeBSD system then see Section 18.6 for more information (in particular, Section 18.6.3 and Section 18.6.4).

   If you will be doing this on another platform then you will need to use whatever utilities exist to control your CD writer on that platform. The images provided are in the standard ISO format, which many CD writing applications support.

Note: If you are interested in building a customized release of FreeBSD, please see the Release Engineering Article (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/releng).

2.13.2 Creating a Local FTP Site with a FreeBSD Disc

FreeBSD discs are laid out in the same way as the FTP site. This makes it very easy for you to create a local FTP site that can be used by other machines on your network when installing FreeBSD.

1. On the FreeBSD computer that will host the FTP site, ensure that the CDROM is in the drive, and mounted on `/cdrom`.

   ```
   # mount /cdrom
   ```

2. Create an account for anonymous FTP in `/etc/passwd`. Do this by editing `/etc/passwd` using vipw(8) and adding this line:

   ```
   ftp:*:99:99::0:0:FTP:/cdrom:/nonexistent
   ```

3. Ensure that the FTP service is enabled in `/etc/inetd.conf`.

Anyone with network connectivity to your machine can now chose a media type of FTP and type in **`ftp://`*`your machine`*** after picking “Other” in the FTP sites menu during the install.

Note: If the boot media (floppy disks, usually) for your FTP clients is not precisely the same version as that provided by the local FTP site, then **sysinstall** will not let you complete the installation. If the versions are not similar and you want to override this, you must go into the Options menu and change distribution name to any.

Warning: This approach is OK for a machine that is on your local network, and that is protected by your firewall. Offering up FTP services to other machines over the Internet (and not your local network) exposes your computer to the attention of crackers and other undesirables. We strongly recommend that you follow good security practices if you do this.

2.13.3 建立安裝用的磁片

若您必須從磁片安裝(雖然我們*不*建議這樣做)，不論是因為硬體不支援或是您堅持要用這麼刻苦的方式，您都必須先準備一些磁片以供安裝。

磁片至少得是1.44 MB At a minimum, you will need as many 1.44 MB floppies as it takes to hold all the files in the `base` (base distribution) directory. If you are preparing the floppies from DOS, then they *must* be formatted using the MS-DOS `FORMAT` command. If you are using Windows, use Explorer to format the disks (right-click on the `A:` drive, and select “Format”).

Do *not* trust factory pre-formatted floppies. Format them again yourself, just to be sure. Many problems reported by our users in the past have resulted from the use of improperly formatted media, which is why we are making a point of it now.

If you are creating the floppies on another FreeBSD machine, a format is still not a bad idea, though you do not need to put a DOS filesystem on each floppy. You can use the `disklabel` and `newfs` commands to put a UFS filesystem on them instead, as the following sequence of commands (for a 3.5" 1.44 MB floppy) illustrates:

```
# fdformat -f 1440 fd0.1440
# bsdlabel -w fd0.1440 floppy3
# newfs -t 2 -u 18 -l 1 -i 65536 /dev/fd0
```

Then you can mount and write to them like any other filesystem.

After you have formatted the floppies, you will need to copy the files to them. The distribution files are split into chunks conveniently sized so that five of them will fit on a conventional 1.44 MB floppy. Go through all your floppies, packing as many files as will fit on each one, until you have all of the distributions you want packed up in this fashion. Each distribution should go into a subdirectory on the floppy, e.g.: `a:\base\base.aa`, `a:\base\base.ab`, and so on.

> **Important:** The `base.inf` file also needs to go on the first floppy of the `base` set since it is read by the installation program in order to figure out how many additional pieces to look for when fetching and concatenating the distribution.

Once you come to the Media screen during the install process, select Floppy and you will be prompted for the rest.

2.13.4 從MS-DOS 分割區安裝

若準備要從MS-DOS 分割區進行安裝，請把所有安裝檔都複製到該分割區根目錄內的`freebsd` 目錄。比如：`c:\freebsd`。此目錄結構必須與光碟或FTP 內的目錄結構一致，因此若是要從光碟複製檔案，建議使用DOS 的`xcopy` 指令。例如，要複製FreeBSD 最小安裝所需的檔案：

```
C:\> md c:\freebsd
C:\> xcopy e:\bin c:\freebsd\bin\ /s
C:\> xcopy e:\manpages c:\freebsd\manpages\ /s
```

假設`C:` 槽有多餘空間，可以放FreeBSD 安裝檔；`E:`則是光碟機代號。

若沒有光碟機，可以到ftp.FreeBSD.org (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.0-RELEASE/) 去下載安裝檔。每個安裝套件都有其相對應的目錄；比如*base* 是放在8.0/base/ (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.0-RELEASE/base/) 目錄內。

請將您要安裝的套件(當然空間要夠)放到MS-DOS 分割區的`c:\freebsd` 裡——因為這個`BIN` 安裝套件僅供最精簡安裝而已。

2.13.5 製作安裝用的磁帶

從磁帶上安裝也許是最簡單的方式，比用FTP 或光碟安裝還快。安裝程式假設所有檔案都會壓縮放在磁帶上。在取得所有要裝的安裝檔之後，可以用下列指令把它們壓縮放在磁帶上：

```
# cd /freebsd/distdir
# tar cvf /dev/rwt0 dist1 ... dist2
```

當要安裝時，必須先確認磁帶還有足夠空間，以便讓安裝過程暫存空間(可以自行選擇要放在哪個目錄)，可以容納磁帶安裝時的*全部*檔案。由於磁帶本身並不能隨機存取，因此用磁帶安裝會需要很大的暫存空間。

Note: 在使用安裝磁片開機 *之前*，磁帶一定要先放入磁帶機內，否則在偵測硬體時可能會無法偵測到磁帶機。

2.13.6 Before Installing over a Network

有三種網路安裝方式：Ethernet (標準Ethernet 晶片)、Serial port(SLIP 或PPP)、Parallel port (PLIP (laplink cable))。

透過網路安裝的最快方式，就是使用Ethernet 網路卡！FreeBSD 支援大多數常見的Ethernet 網路卡；所有支援的網路卡(及其所需的設定)都有在各版本的FreeBSD 內的Hardware Note 說明文件內列出。若您所用的是有支援的PCMCIA 網路卡，請務必在開機*之前*，先把該網路卡插上。因為FreeBSD 的安裝過程，目前並不支援PCMCIA 卡的熱插拔。

此外，還需要知道該用的IP 位址以及相對應的netmask 為何，以及機器名稱。若所用的是PPP 連線，而且沒有固定IP，別擔心，因為您的ISP 會自動分配IP 給您。關於這些網路的細部設定，可以洽詢您網路環境的系統管理者。若要能以機器名稱就能連到相對應的機器，而非直接使用IP位址去連，那麼您還需要DNS 以及gateway 的位址(若用的是PPP 連線，gateway 位址就是ISP 所分配給你的IP 位址)。若想透過HTTP proxy 來使用FTP 安裝，那麼必須知道proxy 的網址為何。若您對上述所需資訊不甚了解，那麼請在安裝*之前*，先詢問系統管理者或ISP。

SLIP 的支援相當原始，並且主要受限於電腦之間的實體線路(hard-wired) ，比如筆記型電腦與其他電腦之間的serial 線。之所以得以電腦間以直接線路連結，乃是由於SLIP 安裝目前並不支援撥接功能。PPP 才有提供撥接功能，所以請儘可能優先採用PPP 而非SLIP。

若要透過數據機(modem)來安裝，那PPP 幾乎是您唯一選擇。請先準備好ISP 所提供的相關資料，因為在安裝之初就會用到。

若使用PAP 或CHAP 來連到ISP(換句話說，若在Windows 可以不透過script 就可以連線到ISP)，那麼您僅需在**ppp** 提示符號下輸入`dial` 指令即可撥號。否則，您必須知道如何以該數據機所採用的“AT 指令集”來連到ISP，因為PPP 撥號程式僅提供非常陽春的終端模擬器(terminal emulator)而起。請參閱Handbook 中user-ppp 章節以及FAQ (http://www.FreeBSD.org/doc/zh_TW.Big5/books/faq/ppp.html) 中的相關項目。若有操作上的疑問，可以打`set log local ...` 指令，以便在螢幕上顯示相關記錄。

若可直接以hard-wired 方式連到另外的FreeBSD(2.0-R 及之後) 機器，那麼可以考慮透過“laplink” 平行電纜來安裝。平行埠的傳輸速率比序列埠高很多(最高可達每秒50 kbytes/sec)，所以安裝速度會更快一些。

2.13.6.1 Before Installing via NFS

NFS 安裝方式相當簡便，只需將FreeBSD 安裝檔案都放到某台NFS server 上，然後再指定使用這台NFS 作為安裝來源即可。

若該server 只允許“privileged port”(通常這是Sun 工作站的預設值)，那麼在安裝之前，必須先到Options 選單去指定`NFS Secure` 設定值。

若網路卡的連線品質不佳，那可能需要調整一下`NFS Slow` 設定。

為了讓NFS 安裝能順利完成，NFS 主機必須要可以支援子目錄的掛載(mount)，例如：FreeBSD 8.0 安裝目錄是在：`ziggy:/usr/archive/stuff/FreeBSD`，那麼`ziggy` 必須允許直接掛載在`/usr/archive/stuff/FreeBSD`，而非僅`/usr` 或是`/usr/archive/stuff`。

在FreeBSD 的`/etc/exports` 檔，上述功能是由`-alldirs` 選項所設定。其他的NFS server 可能會有不同的設定方式。若看到“`permission denied`” 錯誤訊息，則表示可能由於沒有啓用這選項所造成的。

Chapter 3 UNIX 基礎概念

Rewritten by Chris Shumway.

3.1 概述

接下來的這一章將涵蓋FreeBSD 作業系統的基本指令及功能。大部份的內容在UNIX-like 作業系統中都是相通的。如果您對這些內容熟悉的話，可以放心的跳過。如果您剛接觸FreeBSD，那您一定要仔細的讀完這章。

讀完這章，您將了解：

- 如何使用FreeBSD 的“virtual consoles”。
- UNIX 檔案權限運作的方式以及FreeBSD 中檔案的flags。
- 預設的FreeBSD 檔案系統配置。
- FreeBSD 的磁碟結構。
- 如何掛載(mount)、卸載(umount)檔案系統
- 什麼是processes、daemons 以及signals 。
- 什麼是shell ，以及如何變更您預設的登入環境。
- 如何使用基本的文字編輯器。
- 什麼是devices 和device nodes 。
- FreeBSD 下使用的binary 格式。
- 如何閱讀manual pages 以獲得更多的資訊。

3.2 Virtual Consoles 和終端機

有很多方法可以操作FreeBSD ，其中一種就是在文字終端機上打字。如此使用FreeBSD 即可輕易的體會到UNIX 作業系統的威力和彈性。這一節描述什麼是“終端機”和“console” ，以及可以如何在FreeBSD 中運用它們。

3.2.1 The Console

如果您沒有將FreeBSD 設定成開機時自動進入圖形化模式，系統會在啓動的script 跑完之後顯示登入的提示符號。您將會看到像是這樣的東西：

```
Additional ABI support:.
Local package initialization:.
Additional TCP options:.

Fri Sep 20 13:01:06 EEST 2002

FreeBSD/i386 (pc3.example.org) (ttyv0)
```

```
login:
```

這個訊息在您的系統上會有些許的不同，但是應該會看到類似的東西。我們感興趣的是最後兩行，最後兩行是：

```
FreeBSD/i386 (pc3.example.org) (ttyv0)
```

這行包含了剛開機完系統的資訊。您看到的是在Intel 或相容處理器的x86 架構上執行的“FreeBSD”的console[1]。這台機器的名字(每台UNIX 機器都有一個名字)是`pc3.example.org`，而您現在看到的是它的系統console——`ttyv0`終端機。

最後的一行應該都會是：

```
login:
```

這是您應該要輸入您的“帳號名稱”的地方。下一小節將告訴您如何登入FreeBSD。

3.2.2 登入FreeBSD

FreeBSD 是一個multiuser、multiprocessing 的系統。這是一個正式的名稱，指的是在單一機器上可以同時被不同人使用，但同時可以執行很多程式的系統。

每一種多使用者系統都需要可以分辨不同“使用者”的方法。在FreeBSD (以及所有的UNIX-like 作業系統)中，所有的使用者在執行程式之前必須先“登入”系統。每個使用者都有一組獨特的帳號名稱(“username”)及密碼(“password”)。FreeBSD 在允許使用者執行程式前將會先問這兩個問題。

在FreeBSD 開機並跑完啓動的script 之後[2]，它將會印出提示字元要求您輸入正確的帳號名稱：

```
login:
```

在這個範例裡，我們假設您的帳號是`john`。在提示字元處輸入`john` 並按下**Enter** 。接著您應該會看到另一個提示字元要您輸入“密碼”：

```
login: john
Password:
```

輸入`john` 的密碼，再按下**Enter**。輸入的密碼*不會顯示在螢幕上*。您不需要為此擔心，這樣做是為了安全上的問題。

如果您輸入了正確的密碼，您應該已經登入FreeBSD。現在就可以嘗試所有可用的指令了。

您應該會看到MOTD (即今日訊息、Messages Of The Day)，後面接著命令提示字元(一個`#`,`$`, 或是`%` 字元)。這就表示您已經成功登入FreeBSD 了。

3.2.3 多重Console

在一個Console 下執行UNIX 當然是沒有問題，然而FreeBSD 是可以同時執行很多程式的。像FreeBSD 這樣可以同時執行一大堆程式的作業系統，只有一個console 可以輸入指令實在是有點浪費。因此“virtual consoles” 就顯得相當好用。

可以設定讓FreeBSD 同時有很多virtual console，用幾個按鍵的組合就可以從一個virtual console 跳到別的virtual console 。每一個console 都有自已不同的輸出頻道，當從某一個virtual console 切換到下一個的時候，FreeBSD 會自動處理鍵盤輸入及螢幕輸出。

FreeBSD 保留了特別的按鍵組合來切換console [3]。您可以用**Alt-F1**、**Alt-F2**、到**Alt-F8** 來切換FreeBSD 的不同console。

當您從一個console 切換到下一個的時候，FreeBSD 會處理螢幕輸出的儲存及回復。這就“好像”有很多“虛擬”的螢幕和鍵盤，可以讓您輸入指令到FreeBSD 執行。在某一個console 上執行的程式並不會因為切到別的console 而停止執行，切換到另一個console 時，它們仍會繼續執行。

3.2.4 `/etc/ttys` 檔

FreeBSD 預設的虛擬console 總共有8 個，但這並非硬性規定，您可輕鬆設定這些虛擬console 的數量增減。有關虛擬console 的編號跟設定都在`/etc/ttys` 這檔案內設定。

可以用`/etc/ttys` 檔案來設定FreeBSD 的虛擬console。檔案內每行非註解文字(該行開頭沒有# 這字)都是設定終端機或虛擬console。FreeBSD 預設有9 個虛擬console 但只啓動8 個，也就是以下以`ttyv` 開頭的那幾行設定。

```
# name  getty                           type    status          comments
#
ttyv0   "/usr/libexec/getty Pc"         cons25  on  secure
# Virtual terminals
ttyv1   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv2   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv3   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv4   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv5   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv6   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv7   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

有關各欄位的設定以及其他選項，請參閱ttys(5) 說明。

3.2.5 Single User 模式的Console

有關“single user 模式” 的介紹在Section 12.6.2 這邊有詳盡介紹。在single user 模式時，能夠使用的console只有一個，並無虛擬console 可用。而single user 模式相關設定值可以在`/etc/ttys` 檔做調整。下面以`console` 開頭的那行，就是了：

```
# name  getty                           type    status          comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown off secure
```

> **Note:** 在`console` 那行前面的註解有提到，可以把那行的`secure` 改為`insecure`，如此一來，即使FreeBSD 進入single user 模式，仍會要求您輸入`root` 的密碼。
>
> *請審慎考慮是否要改為`insecure`*。因為萬一忘記`root` 密碼的話，若要登入single user 模式就有些麻煩了。儘管還有其他方式可以登入，但對不熟FreeBSD 開機程序的人而言，就會相當棘手。

3.2.6 更改console 的顯示畫面

FreeBSD console 預設顯示大小可以調整為1024x768、1280x1024 或其他顯示卡與螢幕有支援的解析度大小。要切換顯示大小，必須要重新編譯kernel 並加入下面這兩項設定：

```
options VESA
options SC_PIXEL_MODE
```

一旦kernel 有加入這兩項並重新編譯完畢，就可以用vidcontrol(1) 來偵測目前所支援的模式有哪些。若要查看支援的模式，可以打：

```
# vidcontrol -i mode
```

該指令會顯示該機器所支援的顯示模式清單。然後可以在root console 內透過vidcontrol(1) 指令，來更改顯示模式：

```
# vidcontrol MODE_279
```

若對新的顯示模式覺得還不錯，可以在/etc/rc.conf 設定之，以讓每次重開機後會自動生效。以上面這情況為例，就是：

```
allscreens_flags="MODE_279"
```

3.3 權限

FreeBSD 源自於BSD UNIX，繼承了幾個重要的UNIX 概念。首先也最明顯，它是一款multi-user 作業系統。它可以同時處理多人多工，負責徹底的分享與管理來自每位使用者對硬碟裝置、週邊設備、記憶體及CPU 時間的要求。

也因為系統能夠支援多使用者，所以系統管理的一切都有權限來決定誰可以讀取、寫入或執行資源。這些權限分別使用三組八進位的數字儲存，一組代表檔案的所有者，一組代表檔案所屬的群組，而最後一組則代表其他所有人。表示這些數字的方式如下:

值	權限	目錄顯示
0	不可讀取, 不可寫入, 不可執行	`---`
1	不可讀取, 不可寫入, 可執行	`--x`
2	不可讀取, 可寫入, 不可執行	`-w-`
3	不可讀取, 可寫入, 可執行	`-wx`
4	可讀取, 不可寫入, 不可執行	`r--`
5	可讀取, 不可寫入, 可執行	`r-x`
6	可讀取, 可寫入, 不可執行	`rw-`
7	可讀取, 可寫入, 可執行	`rwx`

使用ls(1) 指令時，可以加上-l 參數，來檢視詳細的目錄清單。清單中欄位的資訊包含檔案對所有者、群組及其他人的權限。在任一個目錄底下執行ls -l，會顯示如下的結果:

```
% ls -l
total 530
```

```
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 myfile
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel    7680 Sep  5 12:31 email.txt
...
```

在這裡告所您該如何區分`ls -l`第一欄當中的資訊:

```
-rw-r--r--
```

第一個(最左邊) 的字元用來表示這個檔案的類型為何，除標準檔案以外，尚有目錄、特殊字元裝置(Special character device)、Socket 及其他特殊虛擬檔案裝置(Special pseudo-file device)，在此例當中，- 表示該檔案為一個標準的檔案。範例中接下來的三個字元中，`rw-` 代表所有者對檔案擁有的權限。再接下來的三個字元，`r--` 則代表群組對檔案擁有的權限，最後三個字元，`r--` 則代表其他人對檔案擁有的權限。破折號(-) 表示沒有權限，範例中的這個檔案的權限，只允許所有者讀取、寫入檔案，群組以及其他人僅能讀取檔案。根據以上的表格，此種權限的檔案可以使用`644` 來表示，每組數字分別代表檔案的三種權限。

以上是不錯的方式，但系統該如何控制裝置的權限？實際上FreeBSD 對大多的硬碟裝置就如同檔案，程式可以開啓、讀取以及寫入資料如一般檔案。這些特殊裝置檔案(Special device file) 都儲存於`/dev` 目錄中。

目錄也同如檔案，擁有讀取、寫入及執行的權限，但在執行權限上與檔案有明顯的差異。當目錄被標示為可執行時，代表可以使用“cd” (更改目錄) 進入該目錄。也代表能夠存取在此目錄之中的已知檔名的檔案(當然，檔案仍擁有自己的權限)

尤其，要能夠列出目錄內容，必須擁有目錄的讀取權限。而當要刪除已知檔名的檔案時，也必須擁有檔案所在目錄的寫入*以及* 執行的權限。

還有一些權限，但這些權限主要在特殊情況使用，如setuid binaries 及sticky directories。如果您還想知道更多檔案權限的資訊及使用方法，請務必參閱chmod(1) 說明文件。

3.3.1 權限符號

Contributed by Tom Rhodes.

權限符號可稱做符號表示，使用字元的方式來取代使用數值來設定檔案或目錄的權限。符號表示的格式依序為(某人)(動作)(權限)，可使用的符號如下:

項目	字母	意義
(某人)	u	使用者
(某人)	g	群組所有者
(某人)	o	其他
(某人)	a	全部(“world”)
(動作)	+	增加權限
(動作)	-	移除權限
(動作)	=	指定權限
(權限)	r	讀取
(權限)	w	寫入
(權限)	x	執行
(權限)	t	Sticky bit
(權限)	s	Set UID 或GID

如先前同樣使用chmod(1) 指令來設定，但使用的參數為這些字元。例如，您可以使用下列指令禁止其他使用者存取檔案*FILE*:

```
% chmod go= FILE
```

若有兩個以上的符號表示可以使用逗號(,) 區隔。例如，下列指令將會移除群組及其他人對檔案*FILE* 的寫入權限，並使全部人(“world”)對該檔有執行權限。

```
% chmod go-w,a+x FILE
```

3.3.2 FreeBSD 檔案旗標(Flag)

Contributed by Tom Rhodes.

除了前面提到的檔案權限外，FreeBSD 支援使用“檔案旗標”。這些旗標增加了檔案的安全性及管理性，但不包含目錄。

檔案旗標增加了管理性，確保在某些時候root 不會意外將檔案修改或移除。

修改的檔案flag 僅需要使用擁有簡易的介面的chflags(1) 工具。例如，標示系統禁止刪除的旗標於檔案file1，使用下列指令：

```
# chflags sunlink file1
```

若要移除系統禁止刪除的旗標，只需要簡單在sunlink 前加上“no”，例如：

```
# chflags nosunlink file1
```

使用ls(1) 及參數-lo 可檢視檔案目前的旗標：

```
# ls -lo file1
```

輸出的結果如下:

```
-rw-r--r--  1 trhodes  trhodes  sunlnk 0 Mar  1 05:54 file1
```

多數的旗標僅能由root 使用者來標示或移除，而部份旗標可由檔案所有者設定。我們建議系統管理者可閱讀chflags(1) 及chflags(2) 說明以瞭解相關細節。

3.4 目錄結構

認識FreeBSD 的目錄架構，就可對系統有概略的基礎理解。最重要的莫過於整個目錄的根目錄，就是“/”目錄，該目錄會在開機時最先掛載(mount)，裡面會有開機所會用到必備檔案。此外，根目錄還有紀錄其他檔案系統的掛載點相關設定。

「掛載點」就是讓新增的檔案系統，能接到上層的檔案系統(通常就是「根目錄」檔案系統) 的目錄。在Section 3.5 這邊對此有更詳細介紹。標準的掛載點包括了/usr、/var、/tmp、/mnt 以及/cdrom。這些目錄通常會記錄在/etc/fstab 設定檔內。/etc/fstab 是記錄各檔案系統及相關掛載點的表格。大部分在/etc/fstab 有記錄的檔案系統，會在開機時由rc(8) script 來自動掛載，除非它們有設定noauto 選項。其中細節說明可參閱Section 3.6.1。

有關檔案系統架構的完整說明可參閱hier(7)。現在呢，讓我們大致先一窺常見的目錄有哪些吧。

目錄	說明
`/`	檔案系統的根目錄。
`/bin/`	single-user、multi-user 兩種模式皆可使用的基本工具。
`/boot/`	作業系統開機過程會用到的程式、設定檔。
`/boot/defaults/`	預設的開機啓動設定檔，詳情請參閱loader.conf(5)。
`/dev/`	Device nodes，詳情請參閱intro(4)。
`/etc/`	系統設定檔及一些script 檔。
`/etc/defaults/`	預設的系統設定檔，詳情請參閱rc(8)。
`/etc/mail/`	MTA(Mail Transport Agent)的相關設定檔，像是sendmail(8)。
`/etc/namedb/`	`named` 設定檔，詳情請參閱named(8)。
`/etc/periodic/`	每日、每週、每月透過cron(8); 執行的定期排程script，詳情請參閱periodic(8)。
`/etc/ppp/`	`ppp` 設定檔，詳情請參閱ppp(8)。
`/mnt/`	系統管理者慣用充當臨時掛載點的空目錄。
`/proc/`	Process 檔案系統，詳情請參閱procfs(5)及mount_procfs(8)。
`/rescue/`	緊急救援用途的一些statically linked 程式，詳情請參閱rescue(8)。
`/root/`	`root` 帳號的家目錄。
`/sbin/`	供single-user 及multi-user 環境使用的系統程式及管理工具。
`/tmp/`	臨時檔案。一般而言，重開機之後`/tmp` 內的東西會被清除掉。而通常會將memory-based 檔案系統掛載在`/tmp` 上。這些瑣事可透過tmpmfs 相關的rc.conf(5)環境變數來自動完成。(或是在`/etc/fstab` 內做設定，詳情請參閱mdmfs(8)。)
`/usr/`	主要是使用者所安裝的工具程式、應用程式存放處。
`/usr/bin/`	常用工具、開發工具、應用軟體。
`/usr/include/`	標準C include 的相關header 檔案庫。
`/usr/lib/`	函式庫存放處。
`/usr/libdata/`	其他各式工具的資料檔。
`/usr/libexec/`	系統daemons 及系統工具程式(透過其他程式來執行)。

目錄	說明
`/usr/local/`	存放一些自行安裝的執行檔、函式庫等等。同時，也是FreeBSD ports 架構的預設安裝目錄。`/usr/local` 內的目錄架構大致與`/usr` 相同，詳情請參閱hier(7) 說明。但man 目錄例外，它們是直接放在`/usr/local` 底下，而非`/usr/local/share`，而ports 所安裝的說明文件則在`share/doc/`*`port`*。
`/usr/obj/`	在編譯`/usr/src` 目錄時所產生的相關架構object 檔案。
`/usr/ports`	FreeBSD Ports Collection (optional)。
`/usr/sbin/`	系統daemon 及系統工具(直接由使用者執行)。
`/usr/share/`	各架構皆共通的檔案。
`/usr/src/`	BSD 本身的原始碼(或自行新增的)。
`/usr/X11R6/`	X11R6 相關套件的執行檔、函式庫等(optional)。
`/var/`	存放各種用途的log 檔、臨時或暫時存放、列印或郵件的spool 檔案。有時候，memory-based 檔案系統也會掛載在`/var`。這些瑣事可透過varmfs 相關的rc.conf(5) 環境變數來自動完成。(或是在`/etc/fstab` 內做設定，相關細節請參閱mdmfs(8)。)
`/var/log/`	各項系統記錄的log 檔案。
`/var/mail/`	各使用者的mailbox 檔案。
`/var/spool/`	各種印表機、郵件系統的spool 目錄。
`/var/tmp/`	臨時檔案。這些檔案在重開機後通常仍會保留，除非`/var` 是屬於memory-based 檔案系統。
`/var/yp`	記錄NIS maps。

3.5 磁碟組織

FreeBSD 用來尋找檔案的最小單位就是檔案的名稱了。檔案的名稱有大小寫之分，所以說`readme.txt` 和`README.TXT` 是兩個不同的檔案。FreeBSD 並不使用副檔名(`.txt`) 來判別這是一個程式檔、文件檔或是其他類型的檔案。

檔案存在目錄裡面。一個目錄中可能沒有任何檔案，也可能有好幾百個檔案。目錄之中也可以包含其他的目錄；您可以建立階層式的目錄以便資料的管理。

檔案或目錄的對應是藉由給定的檔案或目錄名稱，然後加上正斜線符號(`/`)；之後再視需要加上其他的目錄名稱。如果您有一個目錄`foo` ，裡面有一個目錄叫作`bar`，這個目錄中又包含了一個叫`readme.txt` 的檔案，那麼這個檔案的全名，或者說檔案的*路徑*就是`foo/bar/readme.txt`。

目錄及檔案儲存在檔案系統之中。每個檔案系統都有唯一一個最上層的目錄，叫做*根目錄(root directory)*。然後在這個根目錄下面才能有其他的目錄。

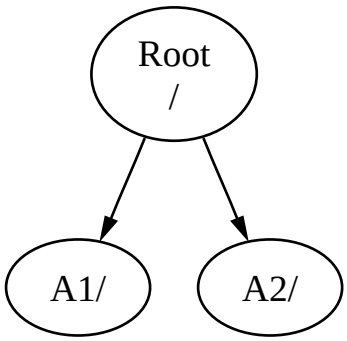
到目前為止大概和其他您用過的的作業系統都差不多。還是有些不一樣的地方就是了，例如MS-DOS 用\ 當檔案和目錄名稱的分隔符號，而Mac OS® 則是用: 符號。

FreeBSD 的路徑中並沒有使用磁碟機代號或其他的磁碟名稱。因此，您不可以使用像`c:/foo/bar/readme.txt` 這樣子的檔案名稱。

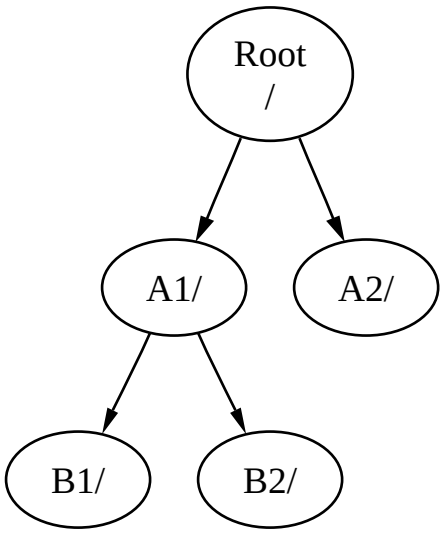
相對的，在FreeBSD 系統中有一個檔案系統被指定為*根檔案系統*。根檔案系統的根目錄由/ 表示。然後其他的檔案系統再*掛載(mount)* 在根檔案系統之下。因此無論您的FreeBSD 系統上有多少顆硬碟，每一個目錄看起來就像在同一個磁碟上。

假設您有三個檔案系統，分別叫作`A`、`B` 及`C`。每個檔案系統都包含兩個目錄，叫做`A1`、`A2` (依此類推得`B1`、`B2` 及`C1`、`C2`)。

稱`A` 為主要的檔案系統；如果您用`ls` 指令查看此目錄的內容，您會看到兩個子目錄：`A1` 及`A2`，如下所示：

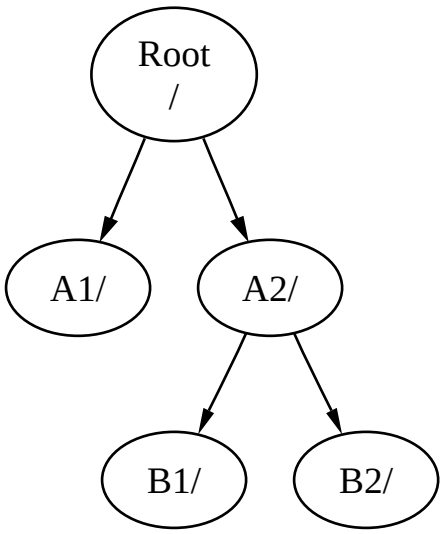


一個檔案系統必須以目錄形式掛載於另一個檔案系統上。因此，假設您將`B` 掛載於`A1` 之上，則`B` 的根目錄就變成了`A1`，而在`B` 之下的任何目錄的路徑也隨之改變:



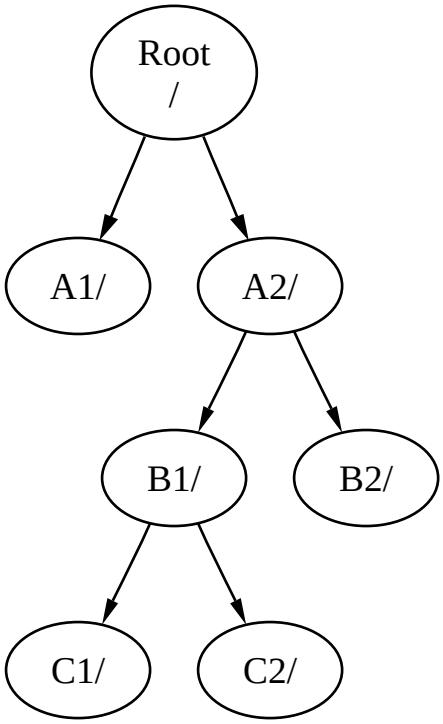
在`B1` 或`B2` 目錄中的任何檔案必須經由路徑`/A1/B1` 或`/A1/B2` 才能達到。所有原來在`/A1` 中的檔案會暫時被隱藏起來，直到`B` 被「*移除(unmounted)*」後才會再顯現出來。

如果`B` 掛載在`A2` 之上，則會變成：

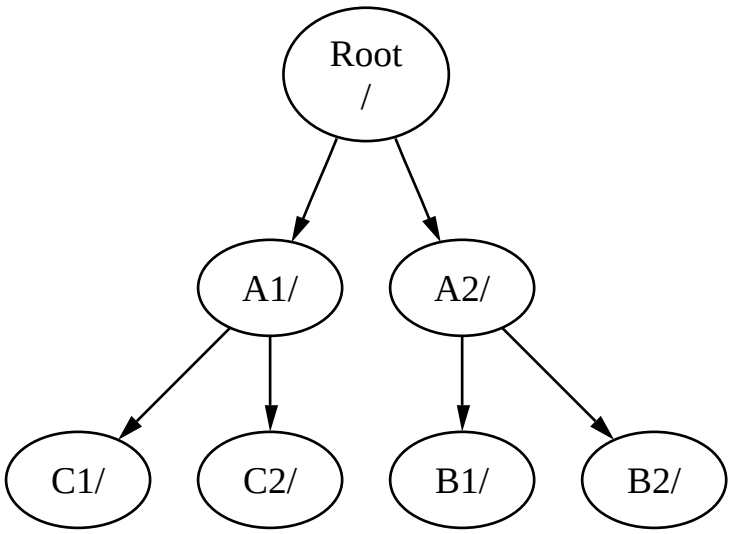


上面的路徑分別為`/A2/B1` 及`/A2/B2`。

檔案系統可以掛在其他檔案系統的目錄之上。延續之前的例子，`C` 檔案系統可以掛在檔案系統`B` 的`B1` 目錄之上，如圖所示：



或者`C` 直接掛載於`A` 的`A1` 目錄之上：



如果您熟悉MS-DOS 的話，這和`join` 指令很類似(雖然不儘相同)。

一般情況下您不需要擔心這些東西。除非您要安裝新的磁碟，不然通常在您安裝FreeBSD 時建立好檔案系統並決定好要掛載在何處之後就不會再做任何更動了。

您完全可以使用單一的一個大的根檔案系統(root file system) 而不建立其他的檔案系統。這樣有好處也有有壞處。

使用多個檔案系統的好處

- 不同的檔案系統在掛上的時候可以有不同的*掛載參數*。舉例來說，為求謹慎您可以將根檔案系統設成唯讀，以避免不小心刪除或修改掉重要的檔案。將使用者可寫入的檔案系統(例如`/home`) 獨立出來也可以讓他們用*nosuid* 的參數掛載，此選項可以讓在這個檔案系統中執行檔的*suid*/*guid* bits 失效，也許可以讓系統更安全。
- FreeBSD 會自動根據您檔案系統的使用方式來做最佳的檔案配置方式。因此，一個有很多小檔案、常常寫入的檔案系統跟只有幾個較大的檔案的檔案系統配置是不一樣的。如果您只有單一一個大的檔案系統，這部分就沒用了。
- FreeBSD 的檔案系統在停電的時候很穩固。然而，在某些重要的時候停電仍然會對檔案系統結構造成損害。分割成許多個檔案系統的話在系統在停電後比較能夠正常啓動，以便您在需要的時候將備份資料回存回來。

使用單一檔案系統的好處

- 檔案系統的大小是固定的。您當初安裝FreeBSD 的時候應該會給定一個大小，可是後來您可能會想把空間加大。如果沒有備份的話是很難達成的；您必須將檔案系統重新建立為您需要的大小，然後將備份回存回來。

> **Important:** FreeBSD 的growfs(8) 指令可以突破此限制直接變更檔案系統的大小。

檔案系統包含在分割區裡面。因為FreeBSD 承襲UNIX 架構，這邊講的分割區和一般提到的分割區(例如MS-DOS 分割區) 不同。每一個分割區由一個代號(字母)表示，從a 到h。每個分割區只能包含一個檔案系統。因此除了說常見到用檔案系統同的掛載點來表示檔案系統外，也可以用包含他的分割區代號來表示。

FreeBSD 也會拿磁碟空間來當*swap space*。Swap space 給FreeBSD 當作*虛擬記憶體*用。這讓您的電腦好像擁有比實際更多的記憶體。當FreeBSD 的記憶體用完的時候，它會把一些目前沒用到的資料移到swap space，然後在用到的時候移回去(同時移出部份沒用到的)。

某些分割區有慣例的使用方式如下：

分割區	慣例
a	通常包含根檔案系統(root file system)
b	通常是swap space
c	通常和整個slice 的大小一樣，給一些會用到整個slice 的工具程式(例如硬碟壞軌檢查工具) 來使用。一般來說您應該不會把檔案系統建立在這個分割區。
d	分割區d 曾經有代表特殊意義，但是已經不再使用。所以現在d 就和其他一般的分割區相同了。

每個包含有檔案系統的分割區是存在所謂的*slice* 裡面。FreeBSD 的slice 就是指平常我們稱為分割區(partition) 的東西。同樣地，會這樣子稱呼也是因為FreeBSD 的UNIX 色彩。而slice 是有編號的，從1 號編到4 號。

slice 號碼跟在裝置名稱後面，先接一個字母s，然後從1 號開始編下去。因此“da0s*1*” 就是指第一個SCSI 硬碟的第一個slice。一個磁碟上只能有四個實體的slice，但是在實體的slice 中您可以塞進適當類型的邏輯slice。這些延伸的slice 編號從5 開始，所以“ad0s*5*” 是第一個IDE 硬碟上的第一個延伸slice。檔案系統在裝置(device) 裡就是在一個slice 之中。

Slices、“dangerously dedicated” 模式的實體磁碟機，以及其他包含*分割區(partition)* 的磁碟都是以字母a 到h 的編號來表示。編號是接在裝置名稱的後面的，因此“da0*a*” 是磁碟機da 上的第一個“dangerously dedicated”模式之分割區。而“ad1s3*e*” 則是第二顆IDE 硬碟上第三個slice 的第五個分割區。

最後，我們就可以把系統上的每個磁碟都區分出來了。一個磁碟的名稱會有一個代碼來表示這個磁碟的類型，接著是一個數字，表示這是哪一個磁碟。這邊跟slice 每個磁碟編號從0 開始不一樣。常見的代碼可以參考Table 3-1。

當要參照一個分割區的時候，FreeBSD 會要您一併輸入包含這個分割區的slice 及磁碟機名稱；當要參照一個slice 的時候，也必須輸入包含這個slice 的磁碟名稱。怎麼做呢？首先先列出磁碟名稱，然後s 加上slice 編號，最後再輸入分割區字母代號。範例可以參考Example 3-1.

Example 3-2 示範了一個基本的磁碟分布模式，相信對您有些幫助。

要安裝FreeBSD，您必須先建置磁碟的slice，接著於slice 中建立要給FreeBSD 用的分割區。最後在這些分割區中建立檔案系統(或swap space) 並決定要將這些檔案系統掛載於哪裡。

Table 3-1. 磁碟機代號

代號	意義
ad	ATAPI(IDE) 磁碟機
da	SCSI 直接存取磁碟機
acd	ATAPI(IDE) 光碟機
cd	SCSI 光碟機

代號	意義
`fd`	軟碟機

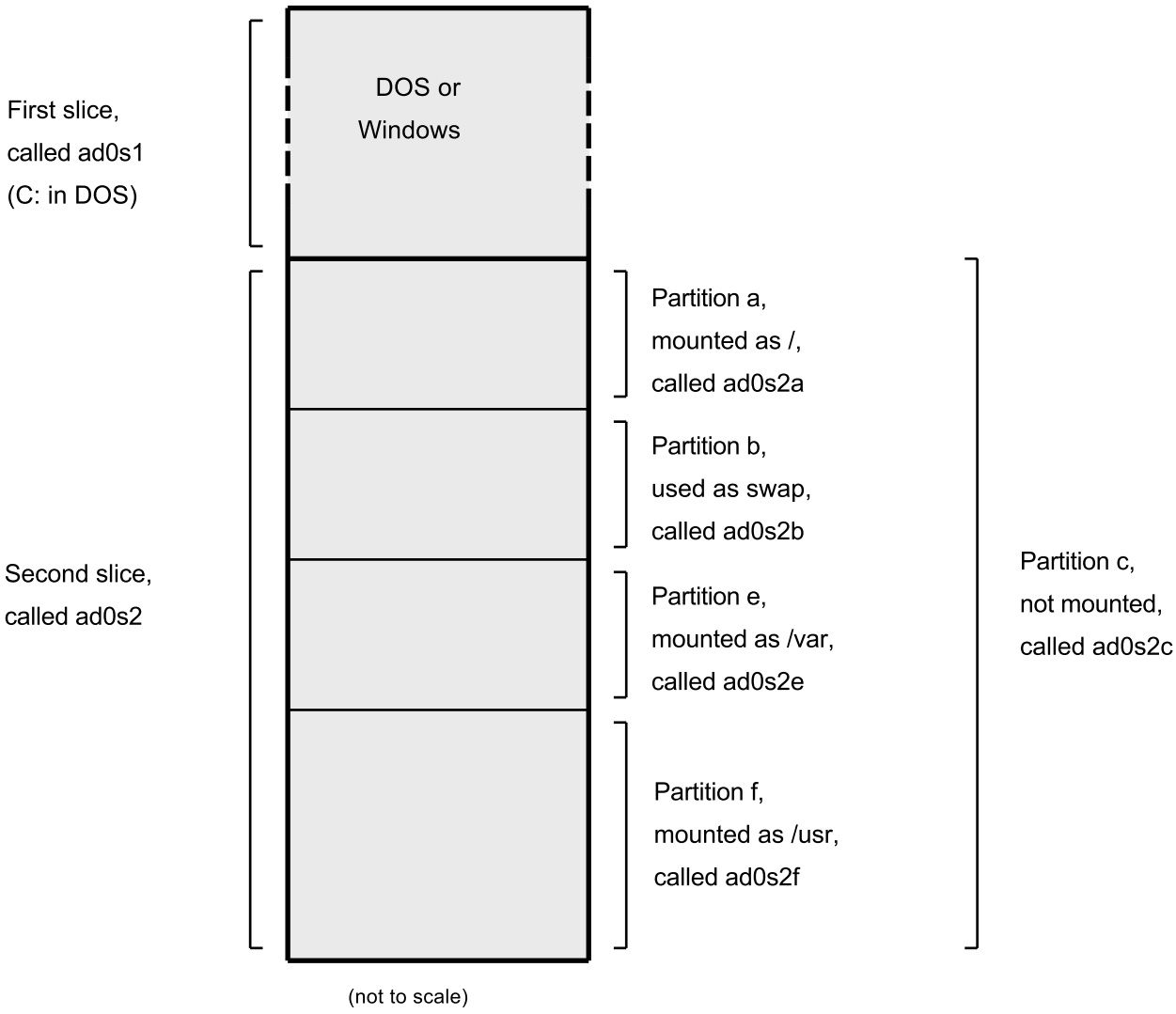
Example 3-1. 磁碟、slice 及分割區命名範例

名稱	意義
`ad0s1a`	第一個IDE 硬碟(`ad0`) 上第一個slice (`s1`)的第一個分割區(`a`) 。
`da1s2e`	第二個SCSI 硬碟(`da1`) 上第二個slice (`s2`) 的第五個分割區(`e`) 。

Example 3-2. 磁碟的概念模型

此圖顯示FreeBSD 中接到系統的第一個IDE 磁碟機內部配置圖。假設這個磁碟的容量是4 GB，並且包含了兩個2 GB 的slice (MS-DOS 的分割區)。第一個slice 是DOS 的`C:` 磁碟機，第二個則安裝了FreeBSD。本範例的FreeBSD 有三個分割區以及一個swap 分割區。

這三個分割區每個都是一個檔案系統。`a` 分割是根(root) 檔案系統；分割`e` 是`/var`；而`f` 分割是`/usr` 目錄結構。



3.6 掛載與卸載檔案系統

檔案系統就像一顆樹。/ 就像是樹根，而/dev，/usr 以及其他在根目錄下的目錄就像是樹枝，而這些樹枝上面又還有分支，像是/usr/local 等。

因為某些原因，我們會將一些目錄分別放在不同的檔案系統上。如/var 包含了可能會滿出來的log/，spool/ 等目錄以及各式各樣的暫存檔。把根檔案系統塞到滿出來顯然不是個好主意，所以我們往往會比較傾向把/var 從/ 中拉出來。

另一個常見到把某些目錄放在不同檔案系統上的理由是：這些檔案在不同的實體或虛擬磁碟機上。像是網路檔案系統 (Network File System) 或是光碟機。

3.6.1 `fstab` 檔

在`/etc/fstab` 裡面有設定的檔案系統會在開機 的過程中自動地被掛載(除非該檔案系統有被加上`noauto` 參數)。

`/etc/fstab` 檔案內容的格式如下：

```
device        /mount-point  fstype      options       dumpfreq      passno
```

`device`

裝置名稱(該裝置必須真的存在)。詳情請參閱Section 18.2.

`mount-point`

檔案系統要掛載到的目錄(該目錄必須真的存在)。

`fstype`

檔案系統類型，這是要傳給mount(8) 的參數。FreeBSD 預設的檔案系統是`ufs`。

`options`

可讀可寫的檔案系統用`rw`，而唯讀的檔案系統則是用`ro`，後面視需要還可以加其他選項。常見的選項如`noauto` 是用在不要於開機過程中自動的掛載的檔案系統。其他選項可參閱mount(8) 說明。

`dumpfreq`

dump(8) 由此項目決定那些檔案系統需要傾印。如果這格空白則以零為預設值。

`passno`

這個項目決定檔案系統檢查的順序。對於要跳過檢查的檔案系統，它們的`passno` 值要設為零。根檔案系統的`passno` 值應設為一(因為需要比所有其他的還要先檢查)，而其他的檔案系統的`passno` 值應該要設得比一大。若有多個檔案系統具有相同的`passno` 值，則fsck(8) 會試著平行地(如果可能的話)檢查這些檔案系統。

更多關於`/etc/fstab` 檔案格式及選項的資訊請參閱fstab(5) 說明文件。

3.6.2 `mount` 指令

mount(8) 指令是拿來掛載檔案系統用的。

基本的操作指令格式如下:

```
# mount device mountpoint
```

在mount(8) 裡面有提到一大堆的選項，不過最常用的就是這些：

掛載選項

`-a`

把`/etc/fstab`裡面所有還沒有被掛載、沒有被標記成“noauto” 而且沒有用`-t`排除的檔案系統掛載起來。

`-d`

執行所有的動作，但是不真的去呼叫掛載的system call。這個選項和`-v`搭配拿來推測mount(8)將要做什麼動作時很好用。

`-f`

強迫掛載不乾淨的檔案系統(危險)，或是用來強制取消寫入權限(把檔案系統的掛載狀態從可存取變成唯讀)。

`-r`

用唯讀的方式掛載檔案系統。這個選項和在`-o`選項中指定`ro` (在FreeBSD 5.2之前的版本是用`rdonly`)參數是一樣的。

`-t` *`fstype`*

用指定的檔案系統型態(fstype)來掛載指定的檔案系統，或是在有`-a`選項時只掛載指定型態的檔案系統。

預設的檔案系統是“ufs”。

`-u`

更新檔案系統的掛載選項。

`-v`

顯示較詳細資訊。

`-w`

以可存取的模式掛載檔案系統。

`-o`選項後面會接著以逗號分隔的參數，例如：

noexec

不允許在這個檔案系統上執行二進位程式碼，這也是一個蠻有用的安全選項。

nosuid

不解析檔案系統上的setuid 或setgid 旗標，這也是一個蠻有用的安全選項。

3.6.3 `umount` 指令

umount(8)指令的參數可以是掛載點(mountpoint)，裝置名稱，以及`-a`或是`-A`等選項。

加上`-f`可以強制卸載，加上`-v`則是會顯示詳細資訊。要注意的是一般來說用`-f`並不是個好主意，強制卸載檔案系統有可能會造成電腦當機，或者損壞檔案系統內的資料。

`-a` 和`-A` 是用來卸載所有已掛載的檔案系統，另外還可以用`-t` 來指定要卸載的是哪些種類的檔案系統。要注意的是`-A` 並不會試圖卸載根檔案系統。

3.7 程序

FreeBSD 是一個多工的作業系統，也就是說在同一時間內可以跑超過一個程式。每一個正在花時間跑的程式就叫做*程序(process)*。您下的每個指令都至少會開啓一個新的程序，而有些系統程序是一直在跑以維持系統正常運作的。

每一個程序都有一個不重覆的數字叫做*process ID* ，或稱為*PID* ，而且就像檔案一樣，每一個程序也有擁有者及群組。擁有者及群組的資訊是用來決定什麼檔案或裝置是這個程序可以開啓的(前面有提到過檔案權限)。大部份的程序都有父程序。父程序是開啓這個程序的程序，例如：您對shell 輸入指令，shell 本身就是一個程序，而您執行的指令也是程序。每一個您用這種方式跑的程序的父程序都是shell。有一個特別的程序叫做init(8) 是個例外。`init` 永遠是第一個程序，所以他的PID 一直都會是1。在FreeBSD 開機的時候`init` 會自動地被kernel 開啓。

要看系統執行中的程序，有兩個相當有用的指令可用：ps(1) 以及top(1)。`ps` 指令是用來列出正在執行之程序，而且可以秀它們的PID、用了多少記憶體、執行的指令名稱及其後之參數是什麼等等。`top` 指令則是顯示所有正在執行的程序，並且數秒鐘更新一次。因此您可以互動式的觀看您的電腦正在做什麼。

在預設的情況下，`ps` 指令只會顯示您所擁有的的程序。例如：

```
% ps
  PID  TT  STAT      TIME COMMAND
  298  p0  Ss     0:01.10 tcsh
 7078  p0  S      2:40.88 xemacs mdoc.xsl (xemacs-21.1.14)
37393  p0  I      0:03.11 xemacs freebsd.dsl (xemacs-21.1.14)
48630  p0  S      2:50.89 /usr/local/lib/netscape-linux/navigator-linux-4.77.bi
48730  p0  IW     0:00.00 (dns helper) (navigator-linux-)
72210  p0  R+     0:00.00 ps
  390  p1  Is     0:01.14 tcsh
 7059  p2  Is+    1:36.18 /usr/local/bin/mutt -y
 6688  p3  IWs    0:00.00 tcsh
10735  p4  IWs    0:00.00 tcsh
20256  p5  IWs    0:00.00 tcsh
  262  v0  IWs    0:00.00 -tcsh (tcsh)
  270  v0  IW+    0:00.00 /bin/sh /usr/X11R6/bin/startx -- -bpp 16
  280  v0  IW+    0:00.00 xinit /home/nik/.xinitrc -- -bpp 16
  284  v0  IW     0:00.00 /bin/sh /home/nik/.xinitrc
  285  v0  S      0:38.45 /usr/X11R6/bin/sawfish
```

在這個範例裡可以看到ps(1) 的輸出分成好幾個欄位。`PID` 就是前面有提到的process ID。PID 的分配是從1開始一直到99999，如果用完的話又會繞回來重頭開始分配(若該PID 已經在用了，則PID 不會重新分配)。`TT` 欄位是指這個程式在哪個tty 上執行，在這裡可以先忽略不管。`STAT` 是程式的狀態，也可以先不要管。`TIME` 是這個程式在CPU 上執行的時間——這通常不是程式總共花的時間，因為當您開始執行程式後，大部份的程式在CPU 上執行前會先花上不少時間等待。最後，`COMMAND` 是執行這個程式的命令列。

ps(1) 有幾個不同的選項組合可以用來變更顯示出來的資訊，其中一個最有用的組合是`auxww`。`a` 可以顯示所有正在跑的程序的指令，不只是您自己的。`u` 則是顯示程序的擁有者名稱以及記憶體使用情況。`x` 可以把daemon 程序顯示出來，而`ww` 可讓ps(1) 顯示出每個程序完整的內容，而不致因過長而被螢幕截掉了。

top(1) 也有類似的輸出。一般的情況看像是這樣：

```
% top
last pid: 72257;  load averages:  0.13,  0.09,  0.03    up 0+13:38:33  22:39:10
47 processes:  1 running, 46 sleeping
CPU states: 12.6% user,  0.0% nice,  7.8% system,  0.0% interrupt, 79.7% idle
Mem: 36M Active, 5256K Inact, 13M Wired, 6312K Cache, 15M Buf, 408K Free
Swap: 256M Total, 38M Used, 217M Free, 15% Inuse

  PID USERNAME PRI NICE  SIZE    RES STATE    TIME   WCPU    CPU COMMAND
72257 nik       28   0  1960K  1044K RUN      0:00 14.86%  1.42% top
 7078 nik        2   0 15280K 10960K select   2:54  0.88%  0.88% xemacs-21.1.14
  281 nik        2   0 18636K  7112K select   5:36  0.73%  0.73% XF86_SVGA
  296 nik        2   0  3240K  1644K select   0:12  0.05%  0.05% xterm
48630 nik        2   0 29816K  9148K select   3:18  0.00%  0.00% navigator-linu
  175 root       2   0   924K   252K select   1:41  0.00%  0.00% syslogd
 7059 nik        2   0  7260K  4644K poll     1:38  0.00%  0.00% mutt
...
```

輸出的資訊分成兩個部份。開頭(前五行) 秀出最近一個程序的PID、系統平均負載(系統有多忙錄的測試)、系統的開機時間(從上次重開算起) 以及現在的時間等。在開頭裡面的其他數字分別是在講有多少程序正在執行(在本例中為47)、有多少記憶體及swap space 被占用了，還有就是系統分別花了多少時間在不同的CPU狀態上。

接下來的部份是由好幾個欄位所構成，和ps(1) 輸出的資訊類似。就如同前例，您可以看到PID、使用者名稱、CPU 花費的時間以及正在執行的指令。top(1) 在預設的情況下還會告訴您程序用掉了多少的記憶體空間。在這邊會分成兩欄，一個是總用量(total size)，另一個是實際用量(resident size)——總用量是指這個應用程式需要的記憶體空間，而實際用量則是指實際上該程式的記憶體使用量。在這個例子裡面您可以看到**Netscape®** 要了幾乎到30 MB 的RAM，但是只有用到9 MB。

top(1) 每隔2 秒鐘會自動更新顯示內容，可用s 選項來改變間隔的時間。

3.8 Daemon、信號及終止程序

當在執行文書編輯器時，您可以很容易地使用它，叫它讀取檔案或是什麼的。可以這樣做是因為編輯器有提供這些功能，還有就是編輯器依附在一個*終端機(Terminal)* 之上。有些程式並不是設計成一直在接收使用者的輸入的，所以它們在一開始執行的時候就從終端機斷開了。例如說，網頁伺服器整天都在回應網頁方面的要求，它通常不需要您輸入任何東西。另外，像是把信從一個站傳送到另一個站的程式，也是這種類型的應用程式。

我們把這種程式稱作*daemon*。Daemon (惡魔、守護神) 是希臘神話中的角色：祂們不屬於善良陣營或邪惡陣營，是守護的小精靈。大致上來說祂們就是在替人類做一些有用的事情，跟今天的網頁伺服器或是郵件伺服器很像。這也就是為何BSD 的吉祥物，長期以來都是一隻穿著帆布鞋拿著三叉耙的快樂小惡魔的原因。

通常來說deamon 程式的名字後面都會加一個字母“d”。**BIND** 是Berkeley Internet Name Domain 的縮寫(但實際上執行的程式名稱是`named`)、Apache 網頁伺服器的程式名稱是`httpd`、印表機服務程式是`lpd`，依此類推。這是習慣用法，並沒有硬性規定，例如**Sendmail** 主要的寄信daemon 是叫做`sendmail` 而不是`maild`，跟您想像的不一樣。

有些時候會需要跟某個daemon 程序溝通，這些溝通是透過所謂的*信號(signal)*來傳遞給該daemon 程序(或是其他執行中的程序)。藉由送出信號，您可以和一個daemon (或是任何一個正在跑的程序) 溝通。信號有很多種——有些有特定的意義，有些則是會由應用程式來解讀。應用程式的說明文件會告訴您該程式是如何解

讀信號的。您只能送信號給您擁有的程序，送kill(1) 或kill(2) 的信號給別人的程序是不被允許的。不過 `root` 不受此限制，他可以送信號給任何人的程序。

FreeBSD 本身在某些情況也會送信號給應用程式。假設有個應用程式寫得很爛，然後企圖要存取它不該碰的記憶體的時候，FreeBSD 會送一個*Segmentation Violation* 信號(`SIGSEGV`) 給這個程序。又如果有一個應用程式用了alarm(3) 的system call 要求系統在過一段時間之後叫他一下，時間到了的時候鬧鐘的信號(`SIGALRM`) 就會被送出了，其他的依此類推。

`SIGTERM` and `SIGKILL` 這兩個信號可以拿來終止程序。用`SIGTERM` 結束程序是比較有禮貌的方式，該程序會*捕捉(catch)* 這個信號而了解到您想要把他關掉。接著下來它會把它自己開的記錄檔通通關掉，然後在關掉程序之前結束掉手邊的工作。在某些情況下程序有可能會裝作沒看見`SIGTERM`，假如它正在做一些不能中斷的工作的話。

`SIGKILL` 就沒有辦法被程序忽略了。這是一個“我管你正在幹嘛，現在就給我停下來”的信號。如果您送了`SIGKILL` 信號給某個程序，FreeBSD 將會把它停掉[4]。

這些是其他您有可能會要用到的信號：`SIGHUP`，`SIGUSR1`，以及`SIGUSR2`。這些是通用的信號，當送出時不同的應用程式會有不同的反應。

假設您更動了您的網頁伺服器的設定檔——您想要叫網頁伺服器去重新讀取設定值。您可以關閉後再重新啓動`httpd`，但是這麼做會造成網頁伺服器暫停服務一段時間，這樣子可能不太好。大部份的daemon 都寫成會去回應`SIGHUP`。當收到這個信號之後，它們會去重新讀取自已的設定檔。因此您可以用送`SIGHUP` 信號來取代關掉重開。又因為沒有標準在規範如何回應這些信號，不同的daemon 可能會有不同的行為，所以有疑問的話請先確認並翻閱deamon 的說明文件。

信號是由kill(1) 指令送出的，如範例所示：

送信號給程序

這個範例將會示範如何送一個信號給inetd(8)。`inetd` 的設定檔是`/etc/inetd.conf`，而`inetd` 會在收到`SIGHUP` 的時候重新讀取這個設定檔。

1. 找出您想要送信號的那個程序的ID。您會用到ps(1) 以及grep(1) 這兩個指令。grep(1) 是用來在輸出中搜尋，找出您指定的字串。這個指令是由一般使用者執行，而inetd(8) 是由`root` 執行，所以在使用ps(1) 時需要加上`ax` 選項。

   ```
   % ps -ax | grep inetd
     198  ??  IWs    0:00.00 inetd -wW
   ```

   因此可知inetd(8) 的PID 為198。在某些情況下`grep inetd` 這個指令本身也會出現在輸出裡。這是因為ps(1) 乃是找所有執行中的程序的方式造成的。

2. 用kill(1) 來送信號。又因為inetd(8) 是由`root` 執行的，您必須用su(1) 切換成`root`先。

   ```
   % su
   Password:
   # /bin/kill -s HUP 198
   ```

   一般情況對大多數UNIX 指令來講，當kill(1) 執行成功時並不會輸出任何訊息。假設您送一個信號給某個不是您所擁有的程序，那麼您就會吃到這個錯誤訊息：“`kill: PID: Operation not permitted`”。而如果您打錯PID 的話，那就會把信號送給錯誤的程序。這樣可能會很糟，不過如果您夠幸運的話，可能剛好就只是把信號送給一個非使用中的PID，那您就只會看到“`kill: PID: No such process`” 而已。

> **為什麼用`/bin/kill`？:** 很多shell 有提供內建的`kill` 指令。也就是說這種shell 會直接送信號，而不是執行`/bin/kill`。這樣是蠻方便的沒錯啦，但是不同的shell 會有不同的語法來指定信號的名稱等。與其嘗試去把它們通通學會，不如就單純的直接用`/bin/kill ...` 吧。

要送其他的信號的話也是非常類似，就視需要把指令中的`TERM` 或`KILL` 替換掉即可。

> **Important:** 隨便抓一個系統中的程序然後把他砍掉並不是個好主意。特別是init(8)，process ID 1，一個非常特別的程序。執行`/bin/kill -s KILL 1` 的結果就是系統立刻關機。因此在您按下**Return** 要執行kill(1)*之前*，請*一定*要記得再次確認您下的參數。

3.9 Shells

在FreeBSD 中，很多日常的工作是在一個叫做shell 的文字介面中完成的。Shell 的主要工作就是從輸入中收到命令並執行它們。許多shell 也有內建一些有助於日常工作的指令，像是檔案管理、檔案比對、命令列編輯、指令巨集以及環境變數等。FreeBSD 有內附了幾個shell，像是`sh`，Bourne Shell，以及`tcsh`，改良版的C-shell。還有許多其他的shell 可以從FreeBSD Ports Collection 中取得，像是`zsh` 以及`bash` 等。

您用哪個shell 呢？其實每個人的喜好都不一樣。如果您是一個C 程式設計師，那對於使用像是`tcsh` 這種C-like 的shell 可能會感到相當愉快。如果你是從Linux 跳過來的，或者您是一個UNIX 新手，那您也許會想要用`bash` 來當作文字介面。每一個shell 都有自己獨特之處，至於這些特點能不能配合您的工作環境？那就是您選擇shell 的重點了。

檔名自動補齊就是常見的shell 功能。首先輸入指令或檔案的前幾個字母，這時通常您只需要按下**Tab** 鍵，接下來shell 就會自動把指令或是檔案名稱剩餘的部份補齊。假設您有兩個檔案分別叫作`foobar` 及`foo.bar`。現在要刪掉`foo.bar`，那麼可以輸入：`rm fo[Tab].[Tab]`

Shell 會印出這個：`rm foo[嗶].bar`。

[嗶] 是console 的響鈴，這嗶的一聲是shell 在告訴我說它沒有辦法完全自動補齊檔名，因為有不只一個檔名符合條件。`foobar` 和`foo.bar` 都是`fo` 開頭的檔名，不過它至少可以補齊到`foo`。如果您接著輸入`.` 然後再按**Tab** 一次，那shell 就能夠替您把剩下的檔名填滿了。

Shell 的另一項特點是使用了環境變數。環境變數是以變數與鍵值（variable/key）的對應關係儲存於shell 的環境空間中，任何由shell 所產生的程序都可以讀取此空間，因此這個空間儲存了許多程序的設定組態。在此附上一份常見環境變數與其涵義的列表：

變數	詳細說明
`USER`	目前登入的使用者名稱。
`PATH`	以冒號（`:`）隔開的目錄列表，用以搜尋執行檔的路徑。
`DISPLAY`	若存在這個環境變數，則代表X11 連結顯示器的網路名稱。
`SHELL`	目前使用的shell。
`TERM`	使用者終端機的名稱，能藉由此變數判斷終端機的能力。

變數	詳細說明
TERMCAP	Database entry of the terminal escape codes to perform various terminal functions.
OSTYPE	作業系統的種類，如：FreeBSD。
MACHTYPE	目前系統所用的CPU 架構。
EDITOR	使用者偏好的文字編輯器。
PAGER	使用者偏好的文字分頁器（text pager）。
MANPATH	以冒號（:）隔開的目錄列表，用以搜尋manual pages 的路徑。

在不同的shell 底下設定環境變數的方式也有所不同。舉例來說，在C-Style 的shell 底下，像是tcsh 和csh，你必須使用setenv 來設定環境變數。但在Bourne shells 底下，像是sh 和bash，你則必須使用export 來設定你所使用的環境變數。再舉個例子來說，若要設定或是修改EDITOR 這個環境變數，在csh 或tcsh 下設定EDITOR 這個環境變數為/usr/local/bin/emacs 的指令是：

```
% setenv EDITOR /usr/local/bin/emacs
```

在Bourne shells 下則是：

```
% export EDITOR="/usr/local/bin/emacs"
```

大多數的shell 都支援使用者在命令列中將$ 字元放在變數之前，以取得環境變數的值。舉例來說，echo $TERM 會顯示出$TERM 的設定值，這是因為shell 取得了$TERM 的設定值，並將其傳給echo 顯示出來。

Shell 中有某些特別的字元是來表示特殊的資料，我們將其稱作meta-characters。其中最常見的是* 字元，他代表了檔名中的任意字元。這些特殊字元可以用在檔名展開（filename globbing）上，舉例來說，輸入echo * 會和輸入ls 得到幾乎相同的結果，這是因為shell 會將所有符合* 字元的檔案傳到命令列上，再由echo 顯示出來。

為了避免shell 轉譯這些特殊字元，我們可以在這些特殊字元前放一個反斜線(\) 字元使他們跳脫(escape) shell 的轉譯。舉例來說，echo $TERM 會印出你目前設定的終端機格式，echo \$TERM 則會直接印出$TERM 這幾個字。

3.9.1 變更你的Shell

變更shell 最簡單的方法就是透過chsh 命令。執行chsh 將會呼叫環境變數中EDITOR 指定的文字編輯器。如果沒有設定，則預設是vi。請依照需求去修改“Shell:” 的值。

你也可以透過chsh 的參數-s，這可以直接設定你的shell 而不需要透過任何文字編輯器。例如，假設想把所用的shell 改為bash，可以透過下列的方式：

```
% chsh -s /usr/local/bin/bash
```

> **Note:** 你所使用的shell *必須* 列於/etc/shells 裡頭。如果是由Ports Collection 來裝shell，那這個步驟已經完成了。但若是手動安裝了一個shell，那麼就必須為新安裝的shell 進行設定。
>
> 舉例來說，若手動安裝了bash 並將它置於/usr/local/bin 底下，你還得：
>
> ```
> # echo "/usr/local/bin/bash" >> /etc/shells
> ```
>
> 然後再重新執行chsh。

3.10 文字編輯器

在FreeBSD 中有許多設定必須透過編輯文字檔完成。因此，若能熟悉文字編輯器是再好不過的。FreeBSD 本身（指base system）就附有幾種文字編輯器，此外，你也可以透過Ports Collection 來安裝其他的文字編輯器。

最簡單易學的文字編輯器叫做**ee**，代表了其全名easy editor。要開始使用**ee**，必須在命令列上輸入`ee filename`，這邊的*`filename`* 代表你想要編輯的檔案名稱。舉例來說，要編輯`/etc/rc.conf`，就要輸入`ee /etc/rc.conf`。而在`ee` 的操作介面下，所有編輯器的功能與操作都會顯示在螢幕的正上方。其中的插入符號(`^`)代表鍵盤上的**Ctrl** 鍵，所以`^e` 就等同於**Ctrl**+**e** 。若要結束**ee**，請按下**Esc** 鍵，接著選擇leave editor 即可。此時如果該檔案有修改過，編輯器會提醒你是否要存檔。

此外，FreeBSD 也內附了幾個好用的文字編輯器，像是base system 的**vi** 及FreeBSD Ports Collection 內的其他編輯器，比如**Emacs** 及**vim** (`editors/emacs` 及`editors/vim`)。這些文字編輯器提供更強的功能，但是也比較難學習。然而若要從事大量文字編輯工作，那麼花點時間來學習這些好用的編輯器，會在日後為您省下更多的時間。

3.11 設備及設備節點

設備(device)主要是指跟硬體比較有關的術語，包括磁碟、印表機、顯示卡和鍵盤。FreeBSD 開機過程當中，大多數硬體通常都能偵測到並顯示出來，也可以查閱`/var/run/dmesg.boot` 內有開機的相關訊息。

舉例來說，`acd0`即為第一台IDE 光碟機的代號，而`kbd0` 則代表鍵盤。

在UNIX 作業系統，大部分的設備都是透過叫做device nodes(設備節點)的特殊檔案來作存取，而這些檔案都位於`/dev` 目錄。

3.11.1 建立設備節點

若要在系統上建立新節點，或者是要編譯某些新硬體的支援軟體，那麼就要先新增設備節點。

3.11.1.1 DEVFS (DEVice File System)

設備檔案系統(或稱為DEVFS) 是指在整體檔案系統namespace 提供kernel 的設備namespace。DEVFS 乃是維護這些檔案系統，而不能新增或修改這些設備節點。

細節請參閱devfs(5) 說明。

3.12 Binary 的格式

若要知道為何FreeBSD 是採用elf(5) 格式，必先瞭解當前UNIX 系統中三種“影響最為重大”的可執行檔相關背景：

- a.out(5)

  最古老、“經典” 的UNIX object 檔格式。It uses a short and compact header with a magic number at the beginning that is often used to characterize the format (see a.out(5) for more details). It contains three loaded segments: .text, .data, and .bss plus a symbol table and a string table.

- COFF

  The SVR3 object format. The header now comprises a section table, so you can have more than just .text, .data, and .bss sections.

- elf(5)

  The successor to COFF, featuring multiple sections and 32-bit or 64-bit possible values. One major drawback: ELF was also designed with the assumption that there would be only one ABI per system architecture. That assumption is actually quite incorrect, and not even in the commercial SYSV world (which has at least three ABIs: SVR4, Solaris, SCO) does it hold true.

  FreeBSD tries to work around this problem somewhat by providing a utility for *branding* a known ELF executable with information about the ABI it is compliant with. See the manual page for brandelf(1) for more information.

FreeBSD comes from the “classic” camp and used the a.out(5) format, a technology tried and proven through many generations of BSD releases, until the beginning of the 3.X branch. Though it was possible to build and run native ELF binaries (and kernels) on a FreeBSD system for some time before that, FreeBSD initially resisted the “push” to switch to ELF as the default format. Why? Well, when the Linux camp made their painful transition to ELF, it was not so much to flee the `a.out` executable format as it was their inflexible jump-table based shared library mechanism, which made the construction of shared libraries very difficult for vendors and developers alike. Since the ELF tools available offered a solution to the shared library problem and were generally seen as “the way forward” anyway, the migration cost was accepted as necessary and the transition made. FreeBSD’s shared library mechanism is based more closely on Sun’s SunOS™ style shared library mechanism and, as such, is very easy to use.

So, why are there so many different formats?

Back in the dim, dark past, there was simple hardware. This simple hardware supported a simple, small system. `a.out` was completely adequate for the job of representing binaries on this simple system (a PDP-11). As people ported UNIX from this simple system, they retained the `a.out` format because it was sufficient for the early ports of UNIX to architectures like the Motorola 68k, VAXen, etc.

Then some bright hardware engineer decided that if he could force software to do some sleazy tricks, then he would be able to shave a few gates off the design and allow his CPU core to run faster. While it was made to work with this new kind of hardware (known these days as RISC), `a.out` was ill-suited for this hardware, so many formats were developed to get to a better performance from this hardware than the limited, simple `a.out` format could offer. Things like COFF, ECOFF, and a few obscure others were invented and their limitations explored before things seemed to settle on ELF.

In addition, program sizes were getting huge and disks (and physical memory) were still relatively small so the concept of a shared library was born. The VM system also became more sophisticated. While each one of these advancements was done using the `a.out` format, its usefulness was stretched more and more with each new feature. In addition, people wanted to dynamically load things at run time, or to junk parts of their program after the init code had run to save in core memory and swap space. Languages became more sophisticated and people wanted code called before main automatically. Lots of hacks were done to the `a.out` format to allow all of these things to happen, and they basically worked for a time. In time, `a.out` was not up to handling all these problems without an ever increasing overhead in code and complexity. While ELF solved many of these problems, it would be painful to switch from the system that basically worked. So ELF had to wait until it was more painful to remain with `a.out` than it was to migrate to ELF.

However, as time passed, the build tools that FreeBSD derived their build tools from (the assembler and loader especially) evolved in two parallel trees. The FreeBSD tree added shared libraries and fixed some bugs. The GNU folks that originally wrote these programs rewrote them and added simpler support for building cross compilers, plugging in different formats at will, and so on. Since many people wanted to build cross compilers targeting FreeBSD, they were out of luck since the older sources that FreeBSD had for **as** and **ld** were not up to the task. The new GNU tools chain (**binutils**) does support cross compiling, ELF, shared libraries, C++ extensions, etc. In addition, many vendors are releasing ELF binaries, and it is a good thing for FreeBSD to run them.

ELF is more expressive than `a.out` and allows more extensibility in the base system. The ELF tools are better maintained, and offer cross compilation support, which is important to many people. ELF may be a little slower than `a.out`, but trying to measure it can be difficult. There are also numerous details that are different between the two in how they map pages, handle init code, etc. None of these are very important, but they are differences. In time support for `a.out` will be moved out of the `GENERIC` kernel, and eventually removed from the kernel once the need to run legacy `a.out` programs is past.

3.13 更多資訊

3.13.1 Manual 線上說明

在使用FreeBSD 時，最詳細的使用說明莫過於man 線上說明。幾乎各程式都會有附上簡短說明，以介紹該程式的基本功能跟相關參數用法。可以透過man 指令來閱讀這些說明，而man 指令的使用相當簡單易懂：

```
% man command
```

command 處就是想要知道的指令。舉個例子，若要知道ls 的詳細用法，就可以打：

```
% man ls
```

而各線上說明因為性質不同，而區分為下列的數字章節：

1. 使用者指令。
2. 系統呼叫(System call) 及錯誤代號。
3. C 語言函式庫。
4. 各設備的驅動程式。
5. 檔案格式。
6. 小遊戲程式及其他娛樂程式。
7. 雜項工具、其他資訊。
8. 系統維護、操作的指令。
9. Kernel 開發用途。

有些情況會有同樣主題但不同章節。舉個例子，系統內會有chmod 指令，但也有chmod() 系統呼叫。在這種情況，man 應該要指定所要查詢的章節：

```
% man 1 chmod
```

如此一來就會查chmod 指令部分。通常在寫文件時會把有參考到某特定章節的man 號碼也一併寫在括號內。所以chmod(1) 就是指chmod 指令，而chmod(2) 則是指系統呼叫的部分。

如果您已經知道命令的名稱，只是不知道要怎樣使用的話，那就比較好辦。但若不知道要用哪個指令時，該怎麼辦呢？這個時候，就可以利用man 的搜尋關鍵字功能，以在各說明的介紹部分搜尋相關字眼。，它的選項是-k：

```
% man -k mail
```

如此一來會看到一堆有“mail” 關鍵字的說明，事實上該功能與apropos 指令是一樣的。

而有時你會看到像是/usr/bin 有許多看起來頗炫的指令，但不知其用途？只要簡單輸入：

```
% cd /usr/bin
% man -f *
```

或者是

```
% cd /usr/bin
% whatis *
```

這兩者的指令效果是一樣的。

3.13.2 GNU Info 檔案

FreeBSD 有許多程式跟工具來自於自由軟體基金會(FSF)。除了man 線上說明之外，這些程式提供了另外一種更具有彈性的hypertext 格式文件，叫做info。可以用info 指令來閱讀，或者若有裝**emacs** 亦可透過**emacs** 的info 模式閱讀。

要用info(1) 指令，只需打：

```
% info
```

按h 會有簡單說明，而若要快速查閱相關操作方式，則請按?。

Notes

1. 這就是i386 的意義。注意即使您不是在Intel 的386 處理器上執行FreeBSD ，一樣是i386。這不是指你的處理器的型號，這裡顯示的是你處理器的“架構”
2. 這些啓動的script 是在開機的時候FreeBSD 會自動執行的程式。他們主要的功能是將所有該執行的東西設定好，並將您設定成背景執行的服務啓動。
3. 在syscons(4)、atkbd(4)、vidcontrol(1)、以及kbdcontrol(1)等manual page 中，對於FreeBSD 的console 及鍵盤驅動程式有詳細的技術說明。我們在這裡不討論細節，有興趣的讀者隨時可以在manual pages 中查到關於運作方式的更詳細且完整的解釋。
4. 不完全正確——還是有少數東西不能被中斷。例如有個程序正在從網路上的別的電腦讀一個檔案，而那部電腦因為某些理由連不到(機器被關掉，或是網路爛掉了)，那這個程序我們就說他是一個“不能中斷的”程序。通常在經過兩分鐘左右之後這個程序會逾時。當發生逾時的時候這個程序就會被結束掉了。

Chapter 4 軟體套件管理篇：Packages 及Ports 機制

4.1 概述

儘管FreeBSD 在base system 已加了很多系統工具。然而，在實務運用上，您可能仍需要安裝額外的軟體。FreeBSD 提供了2 種安裝應用程式的套件管理系統：Ports Collection (以soucre 來編譯、安裝)和package(預先編譯好的binary 檔)。上述的方式，無論要用哪一種，都可以由像是CDROM 等或網路上來安裝想裝的最新版軟體。

讀完這章，您將了解：

- 如何以packages 來安裝軟體。
- 如何以ports 來安裝軟體。
- 已安裝的packages 或ports 要如何移除。
- 如何更改(override) ports collection 所使用的預設值。
- 如何在套件管理系統中，找出想裝的軟體。
- 如何升級已安裝的軟體。

4.2 安裝軟體的各種方式介紹

通常要在UNIX 系統上安裝軟體時，有幾個步驟要作：

1. 先下載該軟體壓縮檔(tarball)，有可能是原始碼或是binary 執行檔。
2. 解開該壓縮檔。(通常是以compress(1) , gzip(1) 或bzip2(1) 壓縮的)
3. 閱讀相關文件檔，以了解如何安裝。(通常檔名是`INSTALL` 或`README` ，或在`doc/` 目錄下的一些文件)
4. 如果所下載的是原始碼，可能要先修改`Makefile` 或是執行`./configure` 之類的script ，接著再編譯該軟體。
5. 最後測試再測試與安裝。

如果一切順利的話，就這麼簡單。如果在安裝非專門設計(移植)給FreeBSD 的軟體時出問題，那可能需要修改一下它的程式碼，才能正常使用。

當然，我們可以在FreeBSD 上使用上述的傳統方式來安裝軟體，但是，我們還有更簡單的選擇。FreeBSD 提供了兩種省事的軟體管理機制：packages 和ports。就在寫這篇文章的時候，已經有超過20,000 個port 軟體可以使用。

所謂的FreeBSD package 就是別人把該應用程式編譯、打包完畢。該package 會包括該應用程式的所有執行檔、設定檔、文件等。而下載到硬碟上的package 都可透過FreeBSD 套件管理指令來進行管理，比如：pkg_add(1)、pkg_delete(1)、pkg_info(1) 等指令。所以，只需簡單打個指令就可輕鬆安裝新的應用程式了。

而FreeBSD port 則是用一些檔案，來自動處理應用程式的安裝流程。

請記住：如果打算自己來編譯的話，需要執行很多操作步驟(下載、解壓、patch、編譯、安裝)。而port 呢，則是涵蓋所有需要完成這些工作的必備步驟，所以只需打一些簡單的指令，那些原始程式碼就會自動下載、解壓、patch、編譯，直至安裝完畢。

事實上，ports 機制還可以用來產生packages，以便他人可以用`pkg_add` 來安裝，或是稍後會介紹到的其他套件管理指令。

而packages 以及ports 它們都是一樣會認*dependencies(軟體相依關係)*。假設：您想安裝某程式，但它有相依另一個已裝的函式庫(library)，而在FreeBSD 的port 以及package 都有這程式以及該函式庫了。所以無論是用`pkg_add` 指令或者port 方式來裝該程式，這兩者(package、port)都會先檢查有沒有裝該函式庫，若沒有就會自動先裝該函式庫了。

這兩種技術都很相似，您可能會好奇為什麼FreeBSD 會弄出這兩種技術來呢。其實，packages 和ports 都有它們各自的長處，使用哪一種完全取決於您自己的喜好。

Package 好處在於：

- 同樣是壓縮過的package 與原始碼tarball 相比，前者通常會比後者小多了。
- package 並不需再進行編譯。對大型應用程式如**Mozilla**、**KDE**、**GNOME** 而言，這點顯得相當重要，尤其是使用速度緩慢的機器。
- 不需要瞭解如何在FreeBSD 上編譯軟體的相關細節過程，即可使用package。

Ports 好處在於：

- 為了讓package 能在大多數系統上順利執行，通常在編譯時會使用比較保守的選項。然而，透過port 安裝的話，則可針對特定環境(比如：Pentium 4 或Athlon CPU) 來調整選項，以符合需求。
- 有些程式在編譯時，會有一些選項可以選擇。舉例來說，**Apache** 可以設定一大堆的編譯選項。若透過port 來安裝的話，會比較彈性多了，可以自己選而不必使用預設的編譯選項。

  在某些情況，同樣的程式但不同編譯選項，則會分成不同的package。比如：**Ghostscript** 會因為是否要裝X11 server，而劃分為`ghostscript` 以及`ghostscript-nox11` 這兩種package。如此的調整對package 算是可成立的，但若該程式有一個以上或兩種不同的編譯選項時，這對package 就沒辦法了。
- 某些軟體的禁止以binary 方式散佈，或者說必須以原始碼方式散佈才可。
- 有些人並不信任binary 套件機制，因為他們覺得至少有原始碼，(理論上)就可以自己檢閱，並尋找是否有潛在的問題。
- 若要對軟體加上自己改過的patch，那麼就必須要先有原始碼才能去上相關patch 修正。
- 有些人喜歡有原始碼在手邊，所以他們無聊時就可以自己閱讀、鑽研、借用(當然要符合原始碼本身的授權規定)原始碼等等。

若想注意port 更新動態的話，可以訂閱FreeBSD ports 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports) 以及FreeBSD ports bugs 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs)。

> **Warning:** 在安裝軟體前，最好先看http://vuxml.freebsd.org/ 內是否有該軟體的安全漏洞通報。
>
> 此外，也可以裝`ports-mgmt/portaudit`，它會自動檢查所有已裝的的軟體是否有已知的安全漏洞，另外，它還會在裝軟體的編譯過程前先行檢查。也可以在裝了某些軟體之後，用`portaudit -F -a` 來作全面強制安檢。

本章接下來將介紹如何在FreeBSD 使用package 及port 來安裝、管理third-party 軟體。

4.3 尋找想裝的軟體

在安裝任何軟體之前，你必須先了解你想要什麼的軟體，以及該軟體叫做什麼名稱。

FreeBSD 上可裝的軟體清單不斷在增加中，不過，我們很慶幸有幾種方式可以來找你想裝的軟體：

- FreeBSD 網站上有更新頻繁的軟體清單，在http://www.FreeBSD.org/ports/ (http://www.FreeBSD.org/ports/index.html)。各ports 皆依其性質而分門別類，既可以透過軟體名稱來搜尋(如果知道名字的話)，也可以在分類中列出所有可用的軟體。
- 由Dan Langille 所維護FreshPorts 網站，網址在http://www.FreshPorts.org/。FreshPorts 會不斷追蹤port tree 中的各種變化，也可以針對某些port 以列入“追蹤名單(watch)” 內，當有任何軟體升級時，就會發email 提醒。
- 如果不知道想裝的軟體名稱，那麼可透過像是FreshMeat (http://www.freshmeat.net/) 這類的網站來找，如果找到了，可以回FreeBSD 網站去看一下這個應用程式是否已經被port 進去了。
- 若知道該port 的正確名稱，但不知道放在哪個分類目錄，可以用whereis(1) 指令來找出來。只要打`whereis` *`file`* 即可，而*`file`* 的地方請改為想裝的軟體名稱。若找到該軟體，就會告訴你，就像下面這樣：

  ```
  # whereis lsof
  lsof: /usr/ports/sysutils/lsof
  ```

  如此一來，就會知道`lsof` (系統工具程式) 是放在`/usr/ports/sysutils/lsof` 目錄。
- 此外，也可以用echo(1) 輕鬆找出該port 是位於porte tree 的何處。舉例來說：

  ```
  # echo /usr/ports/*/*lsof*
  /usr/ports/sysutils/lsof
  ```

  請注意，這也會顯示`/usr/ports/distfiles` 目錄內有符合檔名的檔案。
- 還有另一招，就是用Ports Collection 本身內建的搜尋機制。要用的時候，請先切換到`/usr/ports` 目錄。然後，打`make search name=`*`程式名稱`*，其中*`程式名稱`* 請改為想找的軟體名稱。舉例來說，若要找的是`lsof` 的話，那麼就是：

  ```
  # cd /usr/ports
  # make search name=lsof
  Port:   lsof-4.56.4
  Path:   /usr/ports/sysutils/lsof
  Info:   Lists information about open files (similar to fstat(1))
  Maint:  obrien@FreeBSD.org
  Index:  sysutils
  B-deps:
  R-deps:
  ```

  這些搜尋結果中，要注意的是“Path:” 這行，因為這行會告訴你可以在哪邊找到該port。而搜尋結果的其他部分，因為與port 安裝較無關係，所以這裡就不講了。

  若要更徹底的搜尋，那麼可以改用`make search key=`*`string`*，其中*`string`* 請改為想搜尋的關鍵字。如此一來會找port 名稱、軟體簡介(comments)、軟體敘述檔(descriptions) 以及軟體相依關係(dependencies)裡面是否有符合關鍵字，此外，不清楚軟體名稱的話，也可以拿來找有符合關鍵字主題的port。

  剛講的這兩種方式，搜尋字眼都是case-insensitive(不必區分大小寫)。比如，搜尋“LSOF” 與“lsof” 兩者結果都會是一樣的。

4.4 使用Packages 管理機制

Contributed by Chern Lee.

4.4.1 Package 的安裝方式

可以用pkg_add(1) 從本機上或者透過網路來安裝任一FreeBSD package。

Example 4-1. 手動下載、安裝Package

```
# ftp -a ftp2.FreeBSD.org
Connected to ftp2.FreeBSD.org.
220 ftp2.FreeBSD.org FTP server (Version 6.00LS) ready.
331 Guest login ok, send your email address as password.
230-
230-     This machine is in Vienna, VA, USA, hosted by Verio.
230-         Questions? E-mail freebsd@vienna.verio.net.
230-
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/FreeBSD/ports/packages/sysutils/
250 CWD command successful.
ftp> get lsof-4.56.4.tgz
local: lsof-4.56.4.tgz remote: lsof-4.56.4.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for 'lsof-4.56.4.tgz' (92375 bytes).
100% |***********************************************| 92375       00:00 ETA
226 Transfer complete.
92375 bytes received in 5.60 seconds (16.11 KB/s)
ftp> exit
# pkg_add lsof-4.56.4.tgz
```

若手邊沒有package 來源(像是FreeBSD 光碟)的話，那麼建議使用pkg_add(1) 時，加上`-r` 選項來更輕鬆安裝package。如此一來，就會自動判斷正確的package 格式、以及所搭配的作業系統release 版本，然後會自己從FTP 站抓回、安裝相對應的package。

```
# pkg_add -r lsof
```

上面這例子會自動下載正確的package 並安裝。若想改換用其他FreeBSD Packages Mirror 站，那麼就要設定`PACKAGESITE` 環境變數，如此一來才會取代預設設定。pkg_add(1) 會用fetch(3) 指令來下載檔案，而fetch(3) 本身則會使用相關環境變數的設定，像是：`FTP_PASSIVE_MODE`、`FTP_PROXY` 以及`FTP_PASSWORD`。如果你網路環境處於firewall 後面，或者需要用FTP/HTTP proxy 的話，那麼就需要設定。設定細節請參閱fetch(3)。請注意：上面所說的例子是寫`lsof` 而非`lsof-4.56.4`。當使用遠端抓取功能時，該package 版號就不必加上去了。pkg_add(1) 會自動下載該軟體的最新版回來安裝。

> **Note:** 若用的是FreeBSD-CURRENT 或FreeBSD-STABLE 的話，pkg_add(1) 會自動下載該軟體最新版回來。若用的是屬於-RELEASE 版本，那麼他會抓回屬於該release 上所編譯的package。也可以更改`PACKAGESITE` 環境變數，以改變下載方式。舉例來說，如果是FreeBSD 5.4-RELEASE 的話，那麼pkg_add(1) 預設會從`ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-5.4-release/Latest/` 來抓package。若要

> 強制pkg_add(1) 下載FreeBSD 5-STABLE 所用的package，那麼就把PACKAGESITE 改設為`ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-5-stable/Latest/` 即可。

Package 檔有`.tgz` 以及`.tbz` 兩種格式。這些都可透過ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/ ，或者FreeBSD 光碟內取得。每張4 片裝的FreeBSD 光碟(以及PowerPak 包等等)內都會在`/packages` 目錄內放package。裡面的目錄架構類似`/usr/ports` 的目錄架構。每個分類都各自有專屬目錄，且每份package 都會放在`All` 目錄內。

package 目錄架構與port 的都一致；它們共同構成整個package/port 系統機制。

4.4.2 管理Packages

pkg_info(1) 可用來列出所有已安裝的軟體、軟體簡介。

```
# pkg_info
cvsup-16.1          A general network file distribution system optimized for CV
docbook-1.2         Meta-port for the different versions of the DocBook DTD
...
```

pkg_version(1) 則是列出所有已安裝的軟體版本。它會顯示已裝版本以及目前機器上port tree 的版本差異。

```
# pkg_version
cvsup                       =
docbook                     =
...
```

第二欄的符號表示：已安裝的軟體版本與目前機器上port tree 的版本差異。

符號	代表意義
=	已裝的版本與目前機器上port tree 的版本是同一版的。
<	與目前機器上port tree 版本相比起來，已裝的版本較舊。
>	與目前機器上port tree 版本相比起來，已裝的版本較新。(可能是目前機器上port tree 尚未更新。)
?	已裝的軟體在ports 索引內找無相關資料。(通常可能是，舉例來說：已安裝的該port 已從Ports Collection 中移除或改名了。)
*	該軟體同時有許多版本。

4.4.3 移除已安裝的Package

若要移除已裝的軟體，那麼請多利用pkg_delete(1) 工具，比如：

```
# pkg_delete xchat-1.7.1
```

請注意pkg_delete(1) 須要放上完整的軟體名稱以及版本，若只輸入`xchat` 就不行，必須換成`xchat-1.7.1`

才可。然而，我們可以用pkg_version(1) 輕鬆找出已裝的所有軟體版本，或者以wildcard (萬用字元) 的方式：

```
# pkg_delete xchat\*
```

以上面例子而言，將會移除所有以`xchat` 開頭的軟體。

4.4.4 其他細節部份

所有已裝的package 資訊都會存到`/var/db/pkg` 目錄內，在該目錄下可以找到記載已裝的軟體檔案清單及該軟體簡介的檔案。

4.5 使用Ports 管理機制

下面我們會介紹如何使用Ports Collection 來安裝、移除軟體的基本用法。至於其他可用的`make` 詳細用法與環境設定，可參閱ports(7)。

4.5.1 記得安裝Ports Collection

在安裝任一ports 之前，必須先裝上Ports Collection ——它主要是由`/usr/ports` 內一堆`Makefiles`, patches 以及一些軟體簡介檔所組成的。

在裝FreeBSD 時，若忘了在**sysinstall** 內勾選要裝Ports Collection 的話，沒關係，可以照下列方式來安裝ports collection：

CVSup 方式

使用**CVSup** 是安裝、更新Ports Collection 的快速方法之一。若想更瞭解**CVSup** 用法的話，請參閱使用CVSup。

> **Note: csup** 是以C 語言對**CVSup** 軟體的重寫，在FreeBSD 6.2 及之後版本即有附在系統內。可以直接用系統所附的**csup** 即可跳過步驟一的動作，並將本文相關提到`cvsup` 之處，都改為`csup` 即可。此外，FreeBSD 6.2 之前的版本，則可裝`net/csup` 或者package 來使用**csup**。

第一次跑**CVSup** 之前，請先確認`/usr/ports` 是乾淨的！若你已經裝了Ports Collection ，但又自行加上其他patch 檔，那麼**CVSup** 並不會刪除你自行加上的patch 檔，這樣可能會導致要安裝某些軟體時，發生patch 失敗或編譯失敗。

1. 安裝`net/cvsup-without-gui` package：

   ```
   # pkg_add -r cvsup-without-gui
   ```

   細節用法請參閱安裝CVSup(Section A.5.2)。

2. 執行`cvsup`：

   ```
   # cvsup -L 2 -h cvsup.tw.FreeBSD.org /usr/share/examples/cvsup/ports-supfile
   ```

請把*cvsup.tw.FreeBSD.org* 請改成離你比較近(快)的**CVSup** 主機。這部分可以參閱完整的CVSup mirror 站列表(Section A.5.7)。

> **Note:** 若想改用自己設的ports-supfile，比如說，不想每次都得打指令來指定所使用的**CVSup** 主機。

1. 這種情況下，請以root 權限把/usr/share/examples/cvsup/ports-supfile 複製到其他位置，比如/root 或者自己帳號的家目錄。
2. 修改新的ports-supfile 檔。
3. 把*CHANGE_THIS.FreeBSD.org* 改為離你比較近(快)的**CVSup** 主機。這部分可以參閱完整的CVSup Mirrors (Section A.5.7) 站列表
4. 然後就開始以類似下列指令跑cvsup：

   ```
   # cvsup -L 2 /root/ports-supfile
   ```

3. 執行cvsup(1) 之後，就會開始更新Ports Collection。不過這動作只是『更新』並不是『升級』，不會把已裝的軟體重新編譯、升級。

Portsnap 方式

portsnap(8) 也是更新Ports Collection 的方式之一。FreeBSD 6.0 起開始內建Portsnap 機制，而較舊的系統，則可透過ports-mgmt/portsnap port 來安裝：

```
# pkg_add -r portsnap
```

Portsnap 細節功能，請參閱Portsnap 使用篇。

1. 若/usr/ports 目錄不存在的話，就建立一下吧：

   ```
   # mkdir /usr/ports
   ```

2. 接下來，下載壓縮的Ports Collection 定期更新檔到/var/db/portsnap 目錄。完成下載後，要斷線與否都可以。

   ```
   # portsnap fetch
   ```

3. 若是第一次跑**Portsnap** 的話，則需要先解壓到/usr/ports：

   ```
   # portsnap extract
   ```

   若已有/usr/ports 而且只是想更新而已，那麼就照下面作：

   ```
   # portsnap update
   ```

Sysinstall 方式

這方式要用**sysinstall** 透過安裝來源來裝Ports Collection。請注意：所安裝的Ports Collection 版本只是該release 發佈時的版本而已，而非最新。若能上網(Internet)的話，請使用上述方式之一會比較好。

1. 以root 權限執行sysinstall (在FreeBSD 5.2 之前版本則是/stand/sysinstall)，方式如下：

   ```
   # sysinstall
   ```

2. 請以方向鍵移動選擇項目，選擇Configure，然後按**Enter** 鍵。

3. 選擇Distributions，然後按**Enter** 鍵。
4. 選擇ports，然後按**Space** 鍵。
5. 選Exit，然後按**Enter** 鍵。
6. 選擇要用的安裝來源，比如：CDROM(光碟)、FTP 等方式。
7. 選Exit，然後按**Enter** 鍵。
8. 按下**X** 鍵就可離開**sysinstall** 程式。

4.5.2 Ports 的安裝方式

提到Ports Collection，首先要先說明的是：何謂“skeleton”。簡單來講，port skeleton 就是讓軟體如何在FreeBSD 順利編譯、安裝的最基本檔案組合。每份port skeleton 基本上會有：

- `Makefile` 檔。這個`Makefile` 內容有分許多部分，是用來指定要如何編譯，以及該裝在系統的何處。
- `distinfo` 檔。編譯該軟體所需下載的檔案、checksum(使用md5(1) 及sha256(1) 來檢驗檔案)都會記錄在這檔，以確保所下載的檔案是正確無誤的。
- `files` 目錄。這目錄放的是讓軟體正常編譯、安裝的patch 檔。Patches 檔基本上是一些小檔案，並針對特定檔案來做修改，而且是純文字檔格式，基本上內容通常會像是“Remove line 10(刪除第10 行)”或“Change line 26 to this ...(把第26 行改為...)” 之類的。這些Patches 通常也稱為“diffs” ，因為都是由diff(1) 程式所產生的。

  此外，本目錄也可能會放一些協助編譯該port 的檔案。
- `pkg-descr` 檔，內容是比較詳細的軟體介紹，通常會寫得比較多行。
- `pkg-plist` 檔，該port 會安裝的所有檔案清單。也是告訴系統在移除該port 時，需要刪除哪些檔案。

有些port 還會有其他檔案，像是`pkg-message` 檔。port 系統在一些情況時，會用這些檔案。如果想知道這些檔案的更多細節用途，以及port 一般用法，請參閱FreeBSD Porter’s Handbook (http://www.FreeBSD.org/doc/zh_TW.Big5/books/porters-handbook/index.html)。

port 內寫的是告訴系統如何編譯source code 的相關指令，但並不是真正的source code。而source code 可以從光碟或網路(Internet)來取得，該軟體開發者可能會把source code 以各種格式來發佈。通常是以tar 以及gzip 這兩者工具一起壓縮的檔案，也有可能是以其他工具壓縮，或根本沒壓縮。而軟體的source code 無論是以哪一種壓縮檔型態，我們都稱之為“distfile”。下面將介紹兩種安裝FreeBSD port 的方式。

> **Note:** 要安裝port 的話，請務必切為`root` 身份。

> **Warning:** 在安裝任何port 之前，請務必確認有更新Ports Collection 到最新版，此外請檢閱http://vuxml.freebsd.org/ 來檢查所要裝的port 是否有相關安全漏洞議題需要注意的。
>
> **portaudit** 會在安裝任何port 之前，先自動檢查是否有相關已知的安全漏洞。這個工具在Ports Collection 內有(`ports-mgmt/portaudit`)。在安裝port 之前，可以先跑`portaudit -F` 指令，如此一來就會抓最新的資安漏洞資料庫回來核對。每天的系統定期安檢會自動更新資料庫，並作安全稽核。詳情請參閱portaudit(1) 以及periodic(8) 的線上說明。

Ports Collection 會假設你的網路是可正常連線的。如果沒有的話，那麼需手動把所需的distfile 檔複製到`/usr/ports/distfiles` 才行。

開始操作之前，要先進入打算安裝的port 目錄內：

```
# cd /usr/ports/sysutils/lsof
```

一旦進入`lsof` 目錄後，就可以看到這個port 的skeleton 結構。接下來，就是編譯，也就是“build” 這個port。只需簡單輸入`make` 指令，就可輕鬆完成編譯。完成後，應該可以看到類似下面訊息：

```
# make
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/.
===>  Extracting for lsof-4.57
...
[extraction output snipped]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
===>  Patching for lsof-4.57
===>  Applying FreeBSD patches for lsof-4.57
===>  Configuring for lsof-4.57
...
[configure output snipped]
...
===>  Building for lsof-4.57
...
[compilation output snipped]
...
#
```

請注意：編譯完成後，就會回到提示列(prompt)。接下來就是安裝該port 了，要裝的話，只需在原本的`make` 指令後面再加上一個字即可，那個字就是`install`：

```
# make install
===>  Installing for lsof-4.57
...
[installation output snipped]
...
===>   Generating temporary packing list
===>   Compressing manual pages for lsof-4.57
===>   Registering installation for lsof-4.57
===>  SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
#
```

一旦回到提示列(prompt)，就可以執行剛裝的程式了。另外，因為`lsof` 這程式執行時會有額外權限，所以會出現安全警告。在編譯、安裝port 的時候，請留意任何出現的警告。

此外，建議刪除編譯用的工作目錄(預設是`work`)，這目錄內為在編譯過程中所用到的一些臨時檔案，這些檔案不只佔硬碟空間，而且也可能會在該port 升級新版時，造成不必要的困擾。

```
# make clean
===>  Cleaning for lsof-4.57
```

```
#
```

Note: 用`make install clean` 就可以一口氣完成剛所說`make`、`make install`、`make clean` 這三個步驟了。

Note: 有些shell 會依據`PATH` 環境變數的路徑，把那些路徑的執行檔cache 起來，來加速搜尋執行檔。如果你用的是這類的shell，那麼在裝完port 後需要打`rehash` 指令，才能執行新裝的執行檔，而`rehash` 指令可以在`tcsh` 之類的shell 上使用，若是`sh` 的話，則是`hash -r`。詳情請參閱你所使用的shell 相關文件。

有些由所謂third-party 所發行的DVD-ROM 產品，像是FreeBSD Mall (http://www.freebsdmall.com/) 所發行的FreeBSD Toolkit 會包括distfiles 檔案，這些檔案可用來搭配Ports Collection。把DVD-ROM 掛載在`/cdrom`。若使用其他掛載點的話，要記得設定`CD_MOUNTPTS` 環境變數為相對應的掛載點。如此一來，光碟上若有所需的distfiles 就會自動使用光碟的檔案。

Note: 請注意，有少數port 並不允許透過光碟來發佈檔案。可能的原因有：需先填註冊單才能下載或散佈檔案，或其他原因。如果想安裝在光碟上沒附上的port，就需連上網路才能繼續進行安裝。

ports 系統採用fetch(1) 來下載檔案，它有許多可調整的環境變數，包括：`FTP_PASSIVE_MODE`、`FTP_PROXY`、`FTP_PASSWORD`。如果是處於有防火牆的環境，或者需要使用FTP/HTTP proxy，那麼就需要設定這些變數。使用細節請參閱fetch(3) 說明。

若無法隨時一直上網的話，那麼可以利用`make fetch`。只要在port 的最上層路徑(`/usr/ports`) 打這指令，那麼所有需要用到的檔案都會下載。這指令也可以在下層目錄使用，例如：`/usr/ports/net`。請注意，若該port 有相依的library 或者其他port 的話，那麼它*並不會*跟著一起下載其他所相依的檔案。若想一次下載所有相依的port 所有檔案，那麼指令參數請改用`fetch-recursive` 而非`fetch`。

Note: 可以在某類別或最上層路徑打`make` 指令來編譯所有的port，或者以上述的`make fetch` 指令來下載所有檔案。然而，這樣是相當危險，因為有些port 不能並存。也有另一種情況，有些port 可能會以相同檔名，但是實際上卻是不同內容的檔案。

在某些罕見情況，可能需加上`MASTER_SITES` (檔案的原始下載處)之外的下載點，以下載所需的檔案。可以用下列指令，來更改預設的`MASTER_SITES` 下載點：

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/ fetch
```

上面這例子，是把`MASTER_SITES` 改設`ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/` 為下載點。

Note: 有些port 允許(或要求)您得指定編譯選項，以啓用、停用該軟體中非必須的功能、安全選項以及其他可自訂的選項。具有代表性的包括了`www/mozilla`、`security/gpgme`、`mail/sylpheed-claws`。若有這類選項時，通常在編譯時會出現相關提示訊息。

4.5.2.1 更改(Override)預設的Ports 目錄

有時候，會發現到使用其他目錄作為port、distfiles 目錄可能相當有用(甚至是必須)，可以設定PORTSDIR 及PREFIX 環境變數以修改預設的port 目錄。舉例：

```
# make PORTSDIR=/usr/home/example/ports install
```

以上會在/usr/home/example/ports 內進行編譯，並把所有檔案安裝到/usr/local 內。

```
# make PREFIX=/usr/home/example/local install
```

則會在/usr/ports 目錄內編譯，並把所有檔案安裝到/usr/home/example/local 內。

當然囉，

```
# make PORTSDIR=../ports PREFIX=../local install
```

則會同時包含兩種設定(還有很多變化以致無法在本頁全部都有寫到，但您應該已經有抓到大概概念了吧)。

此外，這些變數也以作為環境變數來設定。請依您所使用的shell 去參閱相關說明，以瞭解如何設定。

4.5.2.2 處理imake

有些port 會使用imake(X Window 系統的一部份) 無法正常運用PREFIX 變數，它們會堅持把檔案都安裝到/usr/X11R6 目錄。同樣地，也有一些Perl port 會忽略PREFIX 並把檔案安裝到Perl 目錄架構內。讓這些ports respect PREFIX 是相當困難，甚至是不可能的事。

4.5.2.3 重新設定Ports 選項

在編譯某些port 時會出現選單畫面(ncurses-based)，可以用來選擇安裝選項。通常裝好該port 之後，便不太會需要重加、移除、更改一些當初安裝的選項。但日後若有需要的話，也有許多方式可以調整這些選項。其中一種方式便是切到該port 目錄，並打make config 即可再次回到選項畫面去作調整。另外還可用make showconfig 以顯示該port 安裝時所用的選項。也可以用make rmconfig 來把所有選項回到初始設定。這些選項跟其他動作都可參閱ports(7) 內的詳細說明。

4.5.3 移除已安裝的Ports

現在您已經知道如何安裝port，而開始想瞭解如何移除。比如裝了一個port 後才意識到裝錯port 了。在此，我們將移除前面例子所裝的那個port (沒仔細注意的話，我們再提醒一下就是lsof)。跟移除package 時相當類似(在Packages section 有介紹)，都是使用pkg_delete(1) 指令：

```
# pkg_delete lsof-4.57
```

4.5.4 升級已安裝的Ports

首先，用pkg_version(1) 指令來列出目前Ports Collection 中提供了那些可升級的port 版本：

```
# pkg_version -v
```

4.5.4.1 /usr/ports/UPDATING

每次更新完Ports Collection 之後，請務必記得在升級port 前，先看看/usr/ports/UPDATING，這裡會寫升級方面的各式問題，比如：檔案格式改變、變更設定檔位置、與舊版不相容的問題等，以及怎麼解決的完整步驟。

若UPDATING 內容與你看到的其他文件有些不同、相衝的話，那麼請以UPDATING 為準。

4.5.4.2 以Portupgrade 來升級已安裝的Ports

portupgrade 可以輕鬆升級已裝的軟體。該工具可從ports-mgmt/portupgrade port 安裝，安裝方式就如同其他port 一樣，用make install clean 指令就可以了：

```
# cd /usr/ports/ports-mgmt/portupgrade
# make install clean
```

首先最好先以pkgdb -F 來掃瞄已裝的ports 資料庫是否有誤，並修正有問題的地方。在每次做升級之前，最好定期做一下pkgdb -F 動作會較為妥當。

跑portupgrade -a 的話，**portupgrade** 會升級系統上所有已裝的過舊ports。若用-i 則在升級每個port 過程當中，會要求確認相關動作是否符合所需。

```
# portupgrade -ai
```

若只想升級某特定程式而非全部，那麼可以用portupgrade *pkgname* 來做指定。若想要**portupgrade** 優先升級某port 所相依的相關套件，則請用-R 參數即可。

```
# portupgrade -R firefox
```

若要用package 而非port 來安裝，則需指定-P 才可以。若有指定這選項，則**portupgrade** 會搜尋PKG_PATH 變數所指定的本機目錄，若找不到則透過網路來下載安裝。若本機跟網路都沒有可用的package 的話，則**portupgrade** 會使用port 方式安裝。若不想如此又變成使用port 方式安裝，則用-PP 即可强制避免使用port 方式安裝。

```
# portupgrade -PP gnome2
```

若只想下載distfiles(或者若指定-P 的話，則是package)而不想編譯或安裝檔案，可以使用-F。詳情請參閱portupgrade(1) 的說明。

4.5.4.3 以Portmanager 來升級已安裝的Ports

Portmanager 也可以用來輕鬆升級已裝的軟體。該工具可從ports-mgmt/portmanager port 安裝：

```
# cd /usr/ports/ports-mgmt/portmanager
# make install clean
```

所有已裝的軟體，都可以輕鬆用類似下列指令來升級：

```
# portmanager -u
```

此外，使用參數可以改為-ui，如此一來**Portmanager** 在升級一些有特殊選項的軟體時，就會詢問該如何升級。**Portmanager** 也可以用來裝新port。與以往常用的make install clean 指令不同之處在於：它會先升級你要裝的port 所相依的所有ports，然後才開始編譯、安裝要裝的port。

```
# portmanager x11/gnome2
```

若要裝的port 之軟體相依關係有問題時，也可以用**Portmanager** 使它們重歸正軌。而**Portmanager** 解決相依問題完畢之後，該port 也會重新編譯，以因應正確的相依關係。

```
# portmanager graphics/gimp -f
```

其餘運用法門，請參閱portmanager(1) 說明。

4.5.5 Ports 與硬碟空間

因為使用Ports Collection 遲早可能會用光硬碟空間，所以在裝完軟體後，記得要以`make clean` 指令來清除臨時的`work` 目錄。此外，可以用下列指令來清除整個Ports Collection 內的臨時目錄：

```
# portsclean -C
```

ports 用久了，您可能會在`distfiles` 目錄內會累積著許多的原始碼檔案。可以手動刪除這些檔案，或者用下列指令來清除所有port 都不使用的舊檔：

```
# portsclean -D
```

或者要清除所有已裝的port 都不再使用的舊檔：

```
# portsclean -DD
```

> **Note:** `portsclean` 這工具乃是**portupgrade** 套件的一部分。

不要忘了移除那些已經安裝，但不再需要用到的ports。有個`ports-mgmt/pkg_cutleaves` port，正是可自動完成這功能的好工具。

4.6 安裝之後，有什麼後續注意事項嗎？

通常，安裝完軟體後，我們可以閱讀所附的一些文件，或需要編輯設定檔，來確保這個軟體能順利運作，或在機器開機的時候啓動(如果是daemon 的話) 等等。

不同的軟體會有不同的設定步驟。不管怎樣，如果裝好了軟體，但是不知道下一步怎麼辦的時候，可以試試看這些小技巧：

- 善用pkg_info(1) ，這指令可以顯示：透過套件管理系統(Packages/Ports)裝了哪些軟體、檔案裝在哪邊。舉例來說，若剛裝了FooPackage (版本1.0.0)，那麼下面這指令：

  ```
  # pkg_info -L foopackage-1.0.0 | less
  ```

  就會顯示這軟體所安裝的檔案清單。請特別注意在`man/` 目錄內是說明檔、`etc/` 目錄內是設定檔、`doc/` 目錄內是完整文件。

  若不確定已裝的套件版本為何，可以用類似下列指令來查：

```
# pkg_info | grep -i foopackage
```

以上將會搜尋所有已裝的套件，列出有符合*foopackage* 的套件名稱。請自行依需求，修改*foopackage* 為想找的套件名稱。

- 一旦確認該程式的線上說明有安裝，就可以用man(1) 來翻閱。同樣地，若該程式有提供的話，也可以參考設定檔樣本，以及其他文件。
- 若該程式有官網的話，還可以透過網站來找文件、常見問答集(FAQ)等。若不知道網址，請用下列指令：

```
# pkg_info foopackage-1.0.0
```

若該程式有官網的話，則會有一行`WWW:` 開頭的出現，這行會列出該程式的官網網址(URL)。

- Port 若須在開機時就會啓動(就像Internet 主機)，通常都會安裝script 到`/usr/local/etc/rc.d` 目錄。您可以檢閱這script 的正確與否，或若有需要，也可以修改、改名。詳情請參閱啓動Services。

4.7 如何處理爛掉(Broken)的Ports？

如果發現某個port 無法順利安裝、運作，有幾種方法可以試試看：

1. 從Problem Report 資料庫 (http://www.FreeBSD.org/support.html#gnats) 中挖寶看看，說不定已經有人送可用的patch 上去囉，那麼或許就可以順利解決問題哩。
2. 向該port 的maintainer 尋求協助：請打`make maintainer` 或翻閱`Makefile` 以查詢maintainer 的email address。記得寄信給maintainer 時，要附註該port 的名稱、版本(或是把`Makefile` 內的`$FreeBSD:` 那一整行附上) 以及相關錯誤訊息。

   > **Note:** 有些port 不是由專門的單一maintainer 負責，而是透過mailing list (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/mailing-list-faq/article.html) 的專題討論。許多(但非全部)的聯絡email 格式通常是`<freebsd-list`名稱`@FreeBSD.org>` 。發問時，請記得把『freebsd-list名稱』改為相關討論的mailing list 名稱。
   >
   > 尤其當port 的maintainer 欄位是`<freebsd-ports@FreeBSD.org>` 時，事實上已經沒人當該port maintainer 了。因此若該port 仍有修正或其他技術支援的話，相關討論都會在freebsd-ports 郵遞論壇上出現。喔，對了，如果有熟悉該軟體者，志願當該port maintainer 的話，我們也都很歡迎您的加入喔。

   若port maintainer 沒有回覆您的信件，則可以用send-pr(1) 來提交問題報告PR。(請參閱Writing FreeBSD Problem Reports (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/problem-reports/article.html))。
3. 試試看修正它吧! Porter's Handbook (http://www.FreeBSD.org/doc/zh_TW.Big5/books/porters-handbook/index.html) 包括了“Ports” 架構的細節部份，這些書中內容有助您修好有問題的port 甚至提交自己的port！
4. 從較近的FTP 站點下載編譯好的package。package collection 的最上游站是在`ftp.FreeBSD.org` 上的packages 目錄 (ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/)內，但請記得先檢查是否已有local mirror (http://mirrorlist.FreeBSD.org/) 站！通常情況下這些package 都可以直接使用，而且應該比自行編譯快一些。用pkg_add(1) 即可順利安裝package 。

Chapter 5 X Window 視窗系統

Updated for X.Org's X11 server by Ken Tom and Marc Fonvieille.

5.1 概述

FreeBSD 使用X11 來提供使用者相當好用的GUI 介面。X11 是X Window 系統，包括**Xorg** 以及**XFree86** 實作的自由軟體版本(以及其他未在本章有介紹的軟體)。FreeBSD 一直到FreeBSD 5.2.1-RELEASE 都仍可在預設的安裝程式內去裝**XFree86** (由The XFree86 Project, Inc 發行的X11 server)。而FreeBSD 5.3-RELEASE 起，預設的X11 改為**Xorg**(由X.Org 基金會所開發的X11 server，並採用與FreeBSD 相當類似的license)。此外，當然也有商業X servers 的FreeBSD 版。

本章主要是介紹X11 (主要著重於**Xorg** 7.4 版部分)的安裝與設定。若欲瞭解**XFree86** 的詳細資料(早期的FreeBSD 內，**XFree86** 乃是預設的X11 套件)，請參閱舊版的FreeBSD Handbook，網址為http://docs.FreeBSD.org/doc/ 。

欲知X11 對於顯示方面硬體的支援情況，請參閱Xorg (http://www.x.org/) 網站。

讀完這章，您將了解：

- X Window 系統的各組成部份，以及它們是如何相互運作。
- 如何安裝、設定X11。
- 如何安裝並使用不同的window managers。
- 如何在X11 上使用TrueType® 字型。
- 如何設定系統以使用圖形登入介面。(**XDM**)

在開始閱讀這章之前，您需要：

- 知道如何運用ports、packages 來安裝軟體。(Chapter 4)

5.2 瞭解X 的世界

第一次接觸X 的人，大概都會有些震撼，尤其是熟悉其他GUI 介面(像是Microsoft Windows 或Mac OS)的使用者。

雖然X 各元件的所有細節及運作方式，並不是必須要知道的。但對它們有些基本概念會更容易上手。

5.2.1 為何叫做X？

X 並非UNIX 上第一套視窗系統，但它卻是最廣為流傳運用。原本的X 研發團隊在研發X 之前有開發另一套視窗系統。那套系統叫做“W”(取“Window” 的第一個字)。而X 則是W 之後的下一個羅馬字母。

X 亦被稱之為“X”、“X Window System”、“X11”，以及其他一些詞彙。使用“X Windows” 這字眼來稱呼X11，可能會讓有些人不爽；這部分細節可參閱X(7) 說明。

5.2.2 X 的Client/Server 架構

X 一開始是設計為網路架構環境，並採用“client-server” 架構。

在X 架構下，“X server” 是在有鍵盤、螢幕、滑鼠的電腦上運作。而server 部份則是負責像是顯示部份的管理、處理來自鍵盤、滑鼠及其他設備(比方像是以繪圖板來輸入、或者是顯示到投影機)的輸入等等，每個X 程式(像是**XTerm**，或**Netscape**)都是“client”。client 會傳訊息到server 上，比如：“Please draw a window at these coordinates”，接著server 會傳回訊息，比如：“The user just clicked on the OK button”。

在家庭或小辦公室環境，通常X server 跟X client 都是在同一台電腦上執行的。然而，也可以在比較爛的桌機上執行X server，並在比較強、比較貴的電腦上跑X 程式(client)來做事情。在這種場景，X client 與server 之間的溝通就需透過網路來進行。

這點可能會讓有些人產生困惑，因為X 術語與他們原本的認知剛好相反。他們原本以為“X server” 是要在最強悍的機器上跑才行，而“X client” 則是在他們桌機上面跑。實際上卻不是這樣。

有點相當重要，請記住X server 是在有接螢幕、鍵盤的機器上運作，而X client 則是顯示這些視窗的程式。

協定(protocol)內並無強制規定client 以及server 兩邊機器都得是同一作業系統，或者得是同型機器才可以。換句話說，也可以在Microsoft Windows 或蘋果電腦(Apple)的Mac OS 上跑X server，而且可以透過許多免費或商業軟體完成這些安裝、設定。

5.2.3 The Window Manager

X 設計哲學與UNIX 設計哲學相當類似，都是“tools, not policy”。也就是說，X 不會試圖強制規定某任務應該要如何完成，而是只提供使用者一些工具，至於如何運用這些工具，則是使用者本身的事了。

X 延續這哲學，它並不規定：螢幕上的視窗該長什麼樣、要如何移動滑鼠指標、該用什麼組合鍵來切換各視窗(比如：在Microsoft Windows 的**Alt**+**Tab**鍵)、各視窗的標題列長相，以及是否該有關閉鈕等等。

事實上，X 把這部分交給所謂的“Window Manager” 來管理。有一堆window manager 程式，像是: **AfterStep** 、**Blackbox**、**ctwm** 、**Enlightenment**、**fvwm**、**Sawfish**、**twm**、**Window Maker** 等等。每一種window manager 都提供不同的使用經驗；有些還可使用“virtual desktops(虛擬桌面)”；有些則可自訂組合鍵來管理桌面；有些會有“Start(開始)”鈕或其他類似設計；有些則是“可更換佈景主題”，可自行安裝新的佈景主題以更換外觀。這些跟其他的window manager 在Ports Collection 內的`x11-wm` 目錄內都有。

此外，**KDE** 及**GNOME** 桌面環境則有其自屬並整合完整的window manager。

每個window manager 也各有其不同的設定機制；有些需手動寫設定檔，而有的則可透過GUI 工具來完成大部分的設定。舉個例子：**Sawfish** 就有以Lisp 語言寫的設定檔。

> **Focus Policy:** window manager 的另一特色就是負責滑鼠指標的“focus policy”。每一種視窗系統都需要選擇作用視窗的方式，以接受鍵盤輸入，以及決定目前哪個視窗是處於使用中的狀態。
>
> 通常較為人熟悉的focus policy 叫做“click-to-focus”，這是Microsoft Windows 所採用的模式，也就是指標在該視窗按一下的話，該視窗就會處於使用中的狀態。
>
> X 並不支援一些特殊的focus policy。換句話說，window manager 會控制哪個視窗在何時是作用中。不同的window manager 有不同的支援方式。但它們都支援click-to-focus，而且大多數都有支援多種方式。
>
> 以下是目前最流行的focus policy：
>
> focus-follows-mouse
>
> 滑鼠移到哪個視窗就是使用該視窗。該視窗不一定位於其他視窗上面，但只要把滑鼠移到該視窗就可以改變作用中的視窗，而不需在它上面點擊。

sloppy-focus

該policy 是針對focus-follows-mouse 的小小延伸。對於focus-follows-mouse 而言，若把游標移到最初的視窗(或桌面)，那所有其他視窗都會處於非作用中，而且所有鍵盤輸入也會失效。若是選用sloppy-focus，則只有在游標移到新視窗時，作用中的視窗才會變成新的，而只離開目前作用中的視窗仍不會改變作用狀態。

click-to-focus

由游標點擊才會決定作用中的視窗。並且該視窗會被“raised(凸顯)” 到所有其他視窗之前，即使游標移到其他視窗，所有的鍵盤輸入仍會由該視窗所接收。

許多window manager 也支援其他policy，與這些相比起來又有些不同，細節部分請參閱該window manager 的文件說明。

5.2.4 Widgets

The X approach of providing tools and not policy extends to the widgets seen on screen in each application.

“Widget” is a term for all the items in the user interface that can be clicked or manipulated in some way; buttons, check boxes, radio buttons, icons, lists, and so on. Microsoft Windows calls these “controls”.

Microsoft Windows and Apple’s Mac OS both have a very rigid widget policy. Application developers are supposed to ensure that their applications share a common look and feel. With X, it was not considered sensible to mandate a particular graphical style, or set of widgets to adhere to.

As a result, do not expect X applications to have a common look and feel. There are several popular widget sets and variations, including the original Athena widget set from MIT, **Motif®** (on which the widget set in Microsoft Windows was modeled, all bevelled edges and three shades of grey), **OpenLook**, and others.

Most newer X applications today will use a modern-looking widget set, either Qt, used by **KDE**, or GTK+, used by the **GNOME** project. In this respect, there is some convergence in look-and-feel of the UNIX desktop, which certainly makes things easier for the novice user.

5.3 安裝X11

Xorg 是FreeBSD 預設的X11 實作。**Xorg** 是由X.Org 基金會所發行之開放源碼軟體X Window 系統實作的X server。**Xorg** 乃是以**XFree86 4.4RC2** 以及X11R6.6 為基礎所產生的。目前FreeBSD Ports Collection 內的**Xorg** 版本為7.4。

從Ports Collection 來安裝**Xorg** 的安裝方式：

```
# cd /usr/ports/x11/xorg
# make install clean
```

Note: 若要編譯完整的**Xorg**，請先確認至少有4 GB 的磁碟空間。

此外X11 也可直接透過package 方式來安裝，可使用pkg_add(1) 來安裝編譯好的X11 套件，記得在透過網路安裝時不要指定版本即可，pkg_add(1) 會自動抓該套件最新版的套件回來。

若要自動透過package 方式來裝**Xorg** ，直接打下面這行即可：

```
# pkg_add -r xorg
```

> **Note:** 上面的例子會裝完整的X11 套件，包括server、client、字型等。此外，還有其他的X11 子套件可透過package 或port 方式來單獨安裝。

本章其餘部分將介紹如何設定X11，以及如何打造高生產力的桌面環境。

5.4 設定X11

Contributed by Christopher Shumway.

5.4.1 在開始之前

在開始設定X11 之前，要先瞭解所要裝的機器資料為何：

- 螢幕規格
- 顯示卡的晶片規格
- 顯示卡的記憶體容量

X11 會依螢幕規格來決定解析度以及更新頻率。這些規格通常可從螢幕所附的文件或廠商網站上取得。最重要的是要知道水平、垂直更新頻率為何。

而顯示卡晶片則決定X11 要用哪一種驅動程式模組。大多數的晶片都可以自動偵測，但最好還是要知道是何種晶片，以免萬一自動偵測失敗。

Video memory on the graphic adapter determines the resolution and color depth which the system can run at. This is important to know so the user knows the limitations of the system.

5.4.2 設定X11

Xorg 自7.3 版起不再需任何設定檔，只要打下列即可：

```
% startx
```

若這指令不行或預設設定無法使用，那麼就需要手動設定X11。設定X11 需要幾個步驟，首先是以系統管理者帳號來建立初始設定檔：

```
# Xorg -configure
```

這會在`/root` 目錄內產生`xorg.conf.new` 設定檔(無論是用su(1) 或直接登入為root，都會改變root 預設的`$HOME` 環境變數)。X11 程式接著會偵測系統的顯示卡相關硬體，並將偵測到硬體訊息寫入設定檔，以便載入正確的驅動程式。

下一步是測試現有的設定檔，以便確認**Xorg** 可以與顯示卡、螢幕相關硬體正確運作：

```
# Xorg -config xorg.conf.new
```

若看得到一堆黑灰夾雜的網格畫面，以及X 形的滑鼠游標，那麼設定檔就是成功的。要退出測試，只要同時按下**Ctrl**+**Alt**+**Backspace** 即可。

Note: 若滑鼠不正確運作，那麼需要先對其作設定。請參閱FreeBSD 安裝一章中的Section 2.10.10 說明。

Next, tune the `xorg.conf.new` configuration file to taste. Open the file in a text editor such as emacs(1) or ee(1). First, add the frequencies for the target system's monitor. These are usually expressed as a horizontal and vertical synchronization rate. These values are added to the `xorg.conf.new` file under the `"Monitor"` section:

```
Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
        HorizSync    30-107
        VertRefresh  48-120
EndSection
```

The `HorizSync` and `VertRefresh` keywords may be missing in the configuration file. If they are, they need to be added, with the correct horizontal synchronization rate placed after the `HorizSync` keyword and the vertical synchronization rate after the `VertRefresh` keyword. In the example above the target monitor's rates were entered.

X allows DPMS (Energy Star) features to be used with capable monitors. The xset(1) program controls the time-outs and can force standby, suspend, or off modes. If you wish to enable DPMS features for your monitor, you must add the following line to the monitor section:

```
        Option       "DPMS"
```

While the `xorg.conf.new` configuration file is still open in an editor, select the default resolution and color depth desired. This is defined in the `"Screen"` section:

```
Section "Screen"
        Identifier "Screen0"
        Device     "Card0"
        Monitor    "Monitor0"
        DefaultDepth 24
        SubSection "Display"
                Viewport  0 0
                Depth     24
                Modes     "1024x768"
        EndSubSection
EndSection
```

The `DefaultDepth` keyword describes the color depth to run at by default. This can be overridden with the `-depth` command line switch to Xorg(1). The `Modes` keyword describes the resolution to run at for the given color depth. Note that only VESA standard modes are supported as defined by the target system's graphics hardware. In the example above, the default color depth is twenty-four bits per pixel. At this color depth, the accepted resolution is 1024 by 768 pixels.

Finally, write the configuration file and test it using the test mode given above.

> **Note:** One of the tools available to assist you during troubleshooting process are the X11 log files, which contain information on each device that the X11 server attaches to. **Xorg** log file names are in the format of `/var/log/Xorg.0.log`. The exact name of the log can vary from `Xorg.0.log` to `Xorg.8.log` and so forth.

If all is well, the configuration file needs to be installed in a common location where Xorg(1) can find it. This is typically `/etc/X11/xorg.conf` or `/usr/local/etc/X11/xorg.conf`.

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

The X11 configuration process is now complete. **Xorg** 目前可透過startx(1) 來啓動之。The X11 server may also be started with the use of xdm(1).

> **Note:** There is also a graphical configuration tool, xorgcfg(1), which comes with the X11 distribution. It allows you to interactively define your configuration by choosing the appropriate drivers and settings. This program can be invoked from the console, by typing the command `xorgcfg -textmode`. For more details, refer to the xorgcfg(1) manual pages.
>
> Alternatively, there is also a tool called xorgconfig(1). This program is a console utility that is less user friendly, but it may work in situations where the other tools do not.

5.4.3 進階設定專欄

5.4.3.1 設定Intel® i810 繪圖晶片組

Configuration with Intel i810 integrated chipsets requires the `agpgart` AGP programming interface for X11 to drive the card. 詳情請參閱agp(4) 説明。

This will allow configuration of the hardware as any other graphics board. Note on systems without the agp(4) driver compiled in the kernel, trying to load the module with kldload(8) will not work. This driver has to be in the kernel at boot time through being compiled in or using `/boot/loader.conf`.

5.4.3.2 為寬螢幕打造更舒適環境

本節假設各位已經有些微進階設定的功力。如果試著使用上述設定工具會有問題的話，請多利用相關log檔(會記錄相關訊息)以便找出解法。找尋解法過程中，可能會需要用到文字編輯器作為輔助。

目前的寬螢幕(WSXGA, WSXGA+, WUXGA, WXGA, WXGA+ 等) 都有支援16:10 及10:9 比例，以及一些可能有問題的比例。以下是一些常見的16:10 螢幕解析度：

- 2560x1600
- 1920x1200
- 1680x1050
- 1440x900
- 1280x800

某方面而言，要增加這些解析度設定也是相當容易的，只要在`Section "Screen"` 內的`Mode` 加上去就好，比如：

```
Section "Screen"
Identifier "Screen0"
Device     "Card0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
 Viewport  0 0
 Depth     24
 Modes     "1680x1050"
EndSubSection
EndSection
```

Xorg 可以透過I2C/DDC 來得知該寬螢幕所支援的解析度等相關資訊，因此就能正確偵測出該螢幕所能支援的頻率、解析度。

若驅動程式並未包括`ModeLine` 訊息的話，那麼就要為**Xorg** 做些設定才行。我們可以透過`/var/log/Xorg.0.log` 檔來取得`ModeLine` 相關設定資料，即可讓螢幕正常顯示。應該可以看到類似下面的訊息：

```
(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 h_blank_end 2240 h_border: 0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 v_blanking: 1089 v_border: 0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 kHz, PixClock max 170 MHz
```

這些訊息被稱為EDID 訊息。可以藉由這些資料，搭配下列的正確順序來產生`ModeLine` 設定：

```
 ModeLine <name> <clock> <4 horiz. timings> <4 vert. timings>
```

所以這個案例`Section "Monitor"` 的`ModeLine` 就會是像下面這樣：

```
Section "Monitor"
Identifier      "Monitor1"
VendorName      "Bigname"
ModelName       "BestModel"
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 1059 1089
Option          "DPMS"
EndSection
```

這樣子就簡單完成了，X 視窗就可以打造為新的寬螢幕環境囉。

5.5 在X11 中使用字型

Contributed by Murray Stokely.

5.5.1 Type1 規格的字型

The default fonts that ship with X11 are less than ideal for typical desktop publishing applications. Large presentation fonts show up jagged and unprofessional looking, and small fonts in **Netscape** are almost completely unintelligible. However, there are several free, high quality Type1 (PostScript®) fonts available which can be readily used with X11. For instance, the URW font collection (`x11-fonts/urwfonts`) includes high quality versions of standard type1 fonts (Times Roman®, Helvetica®, Palatino® and others). The Freefonts collection (`x11-fonts/freefonts`) includes many more fonts, but most of them are intended for use in graphics software such as the **Gimp**, and are not complete enough to serve as screen fonts. In addition, X11 can be configured to use TrueType fonts with a minimum of effort. For more details on this, see the X(7) manual page or the section on TrueType fonts.

To install the above Type1 font collections from the ports collection, run the following commands:

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

And likewise with the freefont or other collections. To have the X server detect these fonts, add an appropriate line to the X server configuration file (`/etc/X11/xorg.conf`), which reads:

```
FontPath "/usr/local/lib/X11/fonts/URW/"
```

Alternatively, at the command line in the X session run:

```
% xset fp+ /usr/local/lib/X11/fonts/URW
% xset fp rehash
```

This will work but will be lost when the X session is closed, unless it is added to the startup file (`~/.xinitrc` for a normal `startx` session, or `~/.xsession` when logging in through a graphical login manager like **XDM**). A third way is to use the new `/usr/local/etc/fonts/local.conf` file: see the section on anti-aliasing.

5.5.2 TrueType® 規格的字型

Xorg has built in support for rendering TrueType fonts. There are two different modules that can enable this functionality. The freetype module is used in this example because it is more consistent with the other font rendering back-ends. To enable the freetype module just add the following line to the `"Module"` section of the `/etc/X11/xorg.conf` file.

```
Load  "freetype"
```

Now make a directory for the TrueType fonts (for example, `/usr/local/lib/X11/fonts/TrueType`) and copy all of the TrueType fonts into this directory. Keep in mind that TrueType fonts cannot be directly taken from a Macintosh®; they must be in UNIX/MS-DOS/Windows format for use by X11. Once the files have been copied into this directory, use **ttmkfdir** to create a `fonts.dir` file, so that the X font renderer knows that these new files have been installed. `ttmkfdir` is available from the FreeBSD Ports Collection as `x11-fonts/ttmkfdir`.

```
# cd /usr/local/lib/X11/fonts/TrueType
```

```
# ttmkfdir -o fonts.dir
```

Now add the TrueType directory to the font path. This is just the same as described above for Type1 fonts, that is, use

```
% xset fp+ /usr/local/lib/X11/fonts/TrueType
% xset fp rehash
```

or add a `FontPath` line to the `xorg.conf` file.

That's it. Now **Netscape**, **Gimp**, **StarOffice**™, and all of the other X applications should now recognize the installed TrueType fonts. Extremely small fonts (as with text in a high resolution display on a web page) and extremely large fonts (within **StarOffice**) will look much better now.

5.5.3 Anti-Aliased 規格的字型

Updated by Joe Marcus Clarke.

Anti-aliasing has been available in X11 since **XFree86** 4.0.2. However, font configuration was cumbersome before the introduction of **XFree86** 4.3.0. Beginning with **XFree86** 4.3.0, all fonts in X11 that are found in `/usr/local/lib/X11/fonts/` and `~/.fonts/` are automatically made available for anti-aliasing to Xft-aware applications. Not all applications are Xft-aware, but many have received Xft support. Examples of Xft-aware applications include Qt 2.3 and higher (the toolkit for the **KDE** desktop), GTK+ 2.0 and higher (the toolkit for the **GNOME** desktop), and **Mozilla** 1.2 and higher.

In order to control which fonts are anti-aliased, or to configure anti-aliasing properties, create (or edit, if it already exists) the file `/usr/local/etc/fonts/local.conf`. Several advanced features of the Xft font system can be tuned using this file; this section describes only some simple possibilities. For more details, please see fonts-conf(5).

This file must be in XML format. Pay careful attention to case, and make sure all tags are properly closed. The file begins with the usual XML header followed by a DOCTYPE definition, and then the <`fontconfig`> tag:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

As previously stated, all fonts in `/usr/local/lib/X11/fonts/` as well as `~/.fonts/` are already made available to Xft-aware applications. If you wish to add another directory outside of these two directory trees, add a line similar to the following to `/usr/local/etc/fonts/local.conf`:

```
<dir>/path/to/my/fonts</dir>
```

After adding new fonts, and especially new font directories, you should run the following command to rebuild the font caches:

```
# fc-cache -f
```

Anti-aliasing makes borders slightly fuzzy, which makes very small text more readable and removes "staircases" from large text, but can cause eyestrain if applied to normal text. To exclude font sizes smaller than 14 point from anti-aliasing, include these lines:

```
<match target="font">
    <test name="size" compare="less">
```

```
            <double>14</double>
        </test>
        <edit name="antialias" mode="assign">
            <bool>false</bool>
        </edit>
    </match>
    <match target="font">
        <test name="pixelsize" compare="less" qual="any">
            <double>14</double>
        </test>
        <edit mode="assign" name="antialias">
            <bool>false</bool>
        </edit>
    </match>
```

Spacing for some monospaced fonts may also be inappropriate with anti-aliasing. This seems to be an issue with **KDE**, in particular. One possible fix for this is to force the spacing for such fonts to be 100. Add the following lines:

```
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>fixed</string>
    </test>
    <edit name="family" mode="assign">
        <string>mono</string>
    </edit>
</match>
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>console</string>
    </test>
    <edit name="family" mode="assign">
        <string>mono</string>
    </edit>
</match>
```

(this aliases the other common names for fixed fonts as `"mono"`), and then add:

```
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>mono</string>
    </test>
    <edit name="spacing" mode="assign">
        <int>100</int>
    </edit>
</match>
```

Certain fonts, such as Helvetica, may have a problem when anti-aliased. Usually this manifests itself as a font that seems cut in half vertically. At worst, it may cause applications such as **Mozilla** to crash. To avoid this, consider adding the following to `local.conf`:

```
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>Helvetica</string>
```

```
    </test>
    <edit name="family" mode="assign">
        <string>sans-serif</string>
    </edit>
</match>
```

Once you have finished editing `local.conf` make sure you end the file with the `</fontconfig>` tag. Not doing this will cause your changes to be ignored.

The default font set that comes with X11 is not very desirable when it comes to anti-aliasing. A much better set of default fonts can be found in the `x11-fonts/bitstream-vera` port. This port will install a `/usr/local/etc/fonts/local.conf` file if one does not exist already. If the file does exist, the port will create a `/usr/local/etc/fonts/local.conf-vera` file. Merge the contents of this file into `/usr/local/etc/fonts/local.conf`, and the Bitstream fonts will automatically replace the default X11 Serif, Sans Serif, and Monospaced fonts.

Finally, users can add their own settings via their personal `.fonts.conf` files. To do this, each user should simply create a `~/.fonts.conf`. This file must also be in XML format.

One last point: with an LCD screen, sub-pixel sampling may be desired. This basically treats the (horizontally separated) red, green and blue components separately to improve the horizontal resolution; the results can be dramatic. To enable this, add the line somewhere in the `local.conf` file:

```
<match target="font">
    <test qual="all" name="rgba">
        <const>unknown</const>
    </test>
    <edit name="rgba" mode="assign">
        <const>rgb</const>
    </edit>
</match>
```

Note: Depending on the sort of display, `rgb` may need to be changed to `bgr`, `vrgb` or `vbgr`: experiment and see which works best.

Anti-aliasing should be enabled the next time the X server is started. However, programs must know how to take advantage of it. At present, the Qt toolkit does, so the entire **KDE** environment can use anti-aliased fonts. GTK+ and **GNOME** can also be made to use anti-aliasing via the "Font" capplet (see Section 5.7.1.3 for details). By default, **Mozilla** 1.2 and greater will automatically use anti-aliasing. To disable this, rebuild **Mozilla** with the `-DWITHOUT_XFT` flag.

5.6 The X Display Manager

Contributed by Seth Kingsley.

5.6.1 Overview

The X Display Manager (**XDM**) is an optional part of the X Window System that is used for login session management. This is useful for several types of situations, including minimal "X Terminals", desktops, and large network display servers. Since the X Window System is network and protocol independent, there are a wide variety of possible configurations for running X clients and servers on different machines connected by a network. **XDM** provides a graphical interface for choosing which display server to connect to, and entering authorization information such as a login and password combination.

Think of **XDM** as providing the same functionality to the user as the getty(8) utility (see Section 24.3.2 for details). That is, it performs system logins to the display being connected to and then runs a session manager on behalf of the user (usually an X window manager). **XDM** then waits for this program to exit, signaling that the user is done and should be logged out of the display. At this point, **XDM** can display the login and display chooser screens for the next user to login.

5.6.2 Using XDM

The **XDM** daemon program is located in `/usr/local/bin/xdm`. This program can be run at any time as `root` and it will start managing the X display on the local machine. If **XDM** is to be run every time the machine boots up, a convenient way to do this is by adding an entry to `/etc/ttys`. For more information about the format and usage of this file, see Section 24.3.2.1. There is a line in the default `/etc/ttys` file for running the **XDM** daemon on a virtual terminal:

```
ttyv8   "/usr/local/bin/xdm -nodaemon"  xterm   off secure
```

By default this entry is disabled; in order to enable it change field 5 from `off` to `on` and restart init(8) using the directions in Section 24.3.2.2. The first field, the name of the terminal this program will manage, is `ttyv8`. This means that **XDM** will start running on the 9th virtual terminal.

5.6.3 Configuring XDM

The **XDM** configuration directory is located in `/usr/local/lib/X11/xdm`. In this directory there are several files used to change the behavior and appearance of **XDM**. Typically these files will be found:

File	Description
`Xaccess`	Client authorization ruleset.
`Xresources`	Default X resource values.
`Xservers`	List of remote and local displays to manage.
`Xsession`	Default session script for logins.
`Xsetup_*`	Script to launch applications before the login interface.
`xdm-config`	Global configuration for all displays running on this machine.
`xdm-errors`	Errors generated by the server program.

File	Description
`xdm-pid`	The process ID of the currently running XDM.

Also in this directory are a few scripts and programs used to set up the desktop when **XDM** is running. The purpose of each of these files will be briefly described. The exact syntax and usage of all of these files is described in xdm(1).

The default configuration is a simple rectangular login window with the hostname of the machine displayed at the top in a large font and "Login:" and "Password:" prompts below. This is a good starting point for changing the look and feel of **XDM** screens.

5.6.3.1 Xaccess

The protocol for connecting to **XDM**-controlled displays is called the X Display Manager Connection Protocol (XDMCP). This file is a ruleset for controlling XDMCP connections from remote machines. It is ignored unless the `xdm-config` is changed to listen for remote connections. By default, it does not allow any clients to connect.

5.6.3.2 Xresources

This is an application-defaults file for the display chooser and login screens. In it, the appearance of the login program can be modified. The format is identical to the app-defaults file described in the X11 documentation.

5.6.3.3 Xservers

This is a list of the remote displays the chooser should provide as choices.

5.6.3.4 Xsession

This is the default session script for **XDM** to run after a user has logged in. Normally each user will have a customized session script in `~/.xsession` that overrides this script.

5.6.3.5 Xsetup_*

These will be run automatically before displaying the chooser or login interfaces. There is a script for each display being used, named `Xsetup_` followed by the local display number (for instance `Xsetup_0`). Typically these scripts will run one or two programs in the background such as `xconsole`.

5.6.3.6 xdm-config

This contains settings in the form of app-defaults that are applicable to every display that this installation manages.

5.6.3.7 xdm-errors

This contains the output of the X servers that **XDM** is trying to run. If a display that **XDM** is trying to start hangs for some reason, this is a good place to look for error messages. These messages are also written to the user's `~/.xsession-errors` file on a per-session basis.

5.6.4 Running a Network Display Server

In order for other clients to connect to the display server, you must edit the access control rules, and enable the connection listener. By default these are set to conservative values. To make **XDM** listen for connections, first comment out a line in the `xdm-config` file:

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:     0
```

and then restart **XDM**. Remember that comments in app-defaults files begin with a "!" character, not the usual "#". More strict access controls may be desired ——look at the example entries in `Xaccess`, and refer to the xdm(1) manual page for further infomation.

5.6.5 Replacements for XDM

Several replacements for the default **XDM** program exist. One of them, **kdm** (bundled with **KDE**) is described later in this chapter. The **kdm** display manager offers many visual improvements and cosmetic frills, as well as the functionality to allow users to choose their window manager of choice at login time.

5.7 桌面環境

Contributed by Valentino Vaschetto.

本章會介紹在FreeBSD 中的X 裡頭，有哪些不同的桌面環境。"桌面環境"範圍很廣，從簡單的window manager 到完整的桌面應用程式，例如**KDE** 或**GNOME**。

5.7.1 GNOME

5.7.1.1 關於GNOME

GNOME is a user-friendly desktop environment that enables users to easily use and configure their computers. **GNOME** includes a panel (for starting applications and displaying status), a desktop (where data and applications can be placed), a set of standard desktop tools and applications, and a set of conventions that make it easy for applications to cooperate and be consistent with each other. Users of other operating systems or environments should feel right at home using the powerful graphics-driven environment that **GNOME** provides. More information regarding **GNOME** on FreeBSD can be found on the FreeBSD GNOME Project (http://www.FreeBSD.org/gnome)'s web site. The web site also contains fairly comprehensive FAQs about installing, configuring, and managing **GNOME**.

5.7.1.2 Installing GNOME

可透過package 或Ports Collection 的方式來輕鬆安裝：

透過網路利用package 安裝**GNOME**：

```
# pkg_add -r gnome2
```

從ports tree 透過原始碼編譯安裝**GNOME**：

```
# cd /usr/ports/x11/gnome2
# make install clean
```

當**GNOME** 安裝完成後，必須告訴X server 啓動**GNOME** 而非原本的window manager。

啓動**GNOME** 最簡單的方法是利用**GDM**(GNOME Display Manager)。**GDM**, which is installed as a part of the **GNOME** desktop (but is disabled by default), can be enabled by adding `gdm_enable="YES"` to `/etc/rc.conf`. Once you have rebooted, **GNOME** will start automatically once you log in ——no further configuration is necessary.

GNOME may also be started from the command-line by properly configuring a file named `.xinitrc`. If a custom `.xinitrc` is already in place, simply replace the line that starts the current window manager with one that starts **/usr/local/bin/gnome-session** instead. If nothing special has been done to the configuration file, then it is enough simply to type:

```
% echo "/usr/local/bin/gnome-session" > ~/.xinitrc
```

Next, type `startx`, and the **GNOME** desktop environment will be started.

> **Note:** If an older display manager, like **XDM**, is being used, this will not work. Instead, create an executable `.xsession` file with the same command in it. To do this, edit the file and replace the existing window manager command with **/usr/local/bin/gnome-session**:

```
% echo "#!/bin/sh" > ~/.xsession
% echo "/usr/local/bin/gnome-session" >> ~/.xsession
% chmod +x ~/.xsession
```

Yet another option is to configure the display manager to allow choosing the window manager at login time; the section on KDE details explains how to do this for **kdm**, the display manager of **KDE**.

5.7.1.3 Anti-aliased Fonts with GNOME

X11 supports anti-aliasing via its "RENDER" extension. GTK+ 2.0 and greater (the toolkit used by **GNOME**) can make use of this functionality. Configuring anti-aliasing is described in Section 5.5.3. So, with up-to-date software, anti-aliasing is possible within the **GNOME** desktop. Just go to Applications→Desktop Preferences→Font, and select either Best shapes, Best contrast, or Subpixel smoothing (LCDs). For a GTK+ application that is not part of the **GNOME** desktop, set the environment variable `GDK_USE_XFT` to `1` before launching the program.

5.7.2 KDE

5.7.2.1 About KDE

KDE is an easy to use contemporary desktop environment. Some of the things that **KDE** brings to the user are:

- A beautiful contemporary desktop
- A desktop exhibiting complete network transparency

- An integrated help system allowing for convenient, consistent access to help on the use of the **KDE** desktop and its applications
- Consistent look and feel of all **KDE** applications
- Standardized menu and toolbars, keybindings, color-schemes, etc.
- Internationalization: **KDE** is available in more than 40 languages
- Centralized, consistent, dialog-driven desktop configuration
- A great number of useful **KDE** applications

KDE comes with a web browser called **Konqueror**, which is a solid competitor to other existing web browsers on UNIX systems. More information on **KDE** can be found on the KDE website (http://www.kde.org/). For FreeBSD specific information and resources on **KDE**, consult the KDE on FreeBSD team (http://freebsd.kde.org/)'s website.

5.7.2.2 安裝KDE

如同**GNOME** 或其他桌面管理軟體一樣，也可以輕鬆透過package 或Ports Collection 來安裝：

To install the **KDE** package from the network, simply type:

```
# pkg_add -r kde
```

pkg_add(1) will automatically fetch the latest version of the application.

To build **KDE** from source, use the ports tree:

```
# cd /usr/ports/x11/kde3
# make install clean
```

After **KDE** has been installed, the X server must be told to launch this application instead of the default window manager. This is accomplished by editing the `.xinitrc` file:

```
% echo "exec startkde" > ~/.xinitrc
```

Now, whenever the X Window System is invoked with `startx`, **KDE** will be the desktop.

If a display manager such as **XDM** is being used, the configuration is slightly different. Edit the `.xsession` file instead. Instructions for **kdm** are described later in this chapter.

5.7.3 More Details on KDE

Now that **KDE** is installed on the system, most things can be discovered through the help pages, or just by pointing and clicking at various menus. Windows or Mac® users will feel quite at home.

The best reference for **KDE** is the on-line documentation. **KDE** comes with its own web browser, **Konqueror**, dozens of useful applications, and extensive documentation. The remainder of this section discusses the technical items that are difficult to learn by random exploration.

5.7.3.1 The KDE Display Manager

An administrator of a multi-user system may wish to have a graphical login screen to welcome users. XDM can be used, as described earlier. However, **KDE** includes an alternative, **kdm**, which is designed to look more attractive and include more login-time options. In particular, users can easily choose (via a menu) which desktop environment (**KDE**, **GNOME**, or something else) to run after logging on.

To enable **kdm**, the `ttyv8` entry in `/etc/ttys` has to be adapted. The line should look as follows:

```
ttyv8 "/usr/local/bin/kdm -nodaemon" xterm on secure
```

5.7.4 XFce

5.7.4.1 About XFce

XFce is a desktop environment based on the GTK+ toolkit used by **GNOME**, but is much more lightweight and meant for those who want a simple, efficient desktop which is nevertheless easy to use and configure. Visually, it looks very much like **CDE**, found on commercial UNIX systems. Some of **XFce**'s features are:

- A simple, easy-to-handle desktop
- Fully configurable via mouse, with drag and drop, etc.
- Main panel similar to **CDE**, with menus, applets and applications launchers
- Integrated window manager, file manager, sound manager, **GNOME** compliance module, and more
- Themeable (since it uses GTK+)
- Fast, light and efficient: ideal for older/slower machines or machines with memory limitations

More information on **XFce** can be found on the XFce website (http://www.xfce.org/).

5.7.4.2 Installing XFce

A binary package for **XFce** exists (at the time of writing). To install, simply type:

```
# pkg_add -r xfce4
```

Alternatively, to build from source, use the ports collection:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

Now, tell the X server to launch **XFce** the next time X is started. Simply type this:

```
% echo "/usr/local/bin/startxfce4" > ~/.xinitrc
```

The next time X is started, **XFce** will be the desktop. As before, if a display manager like **XDM** is being used, create an `.xsession`, as described in the section on GNOME, but with the `/usr/local/bin/startxfce4` command; or, configure the display manager to allow choosing a desktop at login time, as explained in the section on kdm.

II. 一般性工作

既然基礎的部分已經提過了，接下來的這個部分將會討論一些常會用到的FreeBSD 的特色，這些章節包括：

- 介紹給您常見且實用的桌面應用軟體：網頁瀏覽器、生產力工具、文件檢視程式等。
- 介紹給您衆多FreeBSD 上可用的多媒體工具。
- 解釋如何編譯自訂FreeBSD 核心以增加額外系統功能的流程。
- 詳細描述列印系統，包含桌上型印表機及網路印表機的設定。
- 展示給您看如何在您的FreeBSD 系統中執行Linux 應用軟體。

這些章節中有些需要您預先閱讀些相關文件，在各章節開頭的概要內會提及。

Chapter 6 桌面環境應用程式

Contributed by Christophe Juniet.

6.1 概述

在FreeBSD 上面可以執行非常多種類的桌面應用程式，像是網頁瀏覽器和文字處理軟體等。這些程式大都可以透過套件來安裝或是從Ports Collection 中自動編譯安裝。許多新的使用者會希望能在在他們的桌面系統中找到這些程式。這章將會告訴你如何不用費太多功夫去安裝一些熱門的桌面應用程式，不管是從套件或是從Ports Collection 中安裝。

需要注意到的是：當從ports 中安裝程式的時候，它們是從原始碼開始編譯的。依照你編譯的ports 和電腦速度(硬體等級)，有可能會花很長一段時間才能完成。如果從原始碼編譯對你來說會花太多時間的話，大部分的ports 你都能找到事先編譯好的套件來安裝。

因為FreeBSD 具有相容Linux 二進制的特性，許多原先在Linux 上開發的應用程式都能在你的FreeBSD 桌面環境執行。在安裝任何Linux 應用程式之前，强烈建議你先閱讀Chapter 10 Linux 執行相容模式這個章節。而許多用Linux 二進制相容模式的軟體在ports 裡頭通常都會用“linux-” 開頭。當你在搜尋某個特定軟體時，記住這點，並且可以使用whereis(1) 來找。在下列的說明中，都假設你在安裝任何Linux 應用軟體之前，已經事先啓用了Linux 二進制相容模式。

下列目錄是這章中所涵蓋的應用程式：

- 瀏覽器(像是**Mozilla**, **Opera**, **Firefox**, **Konqueror**)
- 辦公軟體(像是**KOffice**, **AbiWord**, **The GIMP**, **OpenOffice.org**)
- 文件瀏覽軟體(像是**Acrobat Reader®**, **gv**, **Xpdf**, **GQview**)
- 財務處理軟體(像是**GnuCash**, **Gnumeric**, **Abacus**)

在閱讀這章之前，你必須

- 知道如何安裝其他的軟體(third-party software) (Chapter 4).
- 知道如何安裝Linux 軟體(Chapter 10).

要知道更多關於多媒體環境的資訊，請先閱讀Chapter 7 多媒體章節。如果你想要設定和使用電子郵件，也請你先看Chapter 26郵件章節。

6.2 瀏覽器

在FreeBSD 中並沒有預先安裝好的特定瀏覽器。但在Ports Collection 之中卻有許多瀏覽器可供你安裝使用。如果你沒有足夠時間去編譯所有的東西(在某些情況下這可能會花上很長的一段時間)，這些都有現成的套件可供直接安裝。

KDE 和**GNOME** 桌面環境都已提供HTML 瀏覽器。請參考Section 5.7 來了解更多有關如何設定這些完整的桌面環境系統資訊。

如果你在尋找輕量化的瀏覽器，你可以從Ports Collection 中找到下面的幾種：`www/dillo`, `www/links`, 或`www/w3m`。

這節介紹這些瀏覽器：

瀏覽器名稱	所需的系統資源	從ports 安裝時間	主要相依的軟體
Mozilla	多	長	**Gtk+**
Opera	少	短	FreeBSD 和Linux 的版本都有。Linux 的版本需要Linux 二進制相容模組以及**linux-openmotif**.
Firefox	中度	長	**Gtk+**
Konqueror	中度	長	**KDE** 函式庫

6.2.1 Mozilla

Mozilla 是相當現代化、穩定且完全移植至FreeBSD 系統上。它也具備有十分符合HTML 標準的顯示引擎，它更提供了郵件及新聞群組的閱讀功能。此外如果你打算要自己寫一些網頁的話，它還提供了HTML 的編輯器。如果是**Netscape** 的使用者，你可能會認出這跟**Communicator** 很像，它們其實同樣是使用相同基礎的瀏覽器。

在速度較慢，像是CPU 速度少於233MHz 或是小於64MB 記憶體的機器上面，完全使用**Mozilla** 會是件極度耗費資源的事。所以在這樣的機器上面，你可能會想要使用**Opera** 這樣輕量級的瀏覽器，而接下來後面會提到。

如果你有什麼原因不能或是不想編譯**Mozilla** 的話，FreeBSD GNOME 團隊已經為你做好了這件事。只要用下面的指令透過網路安裝套件就行了：

```
# pkg_add -r mozilla
```

如果沒有找到套件可以使用，而你也有足夠的時間和磁碟空間來編譯**Mozilla** 並安裝到你的系統中，你可以透過下列步驟來安裝：

```
# cd /usr/ports/www/mozilla
# make install clean
```

Mozilla 需要使用`root` 的權限來執行chrome 註冊來確保正確的初始化。另外，如果你需要抓一些額外的外掛程式像是mouse gestures，你就必須要使用`root` 的權限來安裝，以適當的安裝這些外掛程式。

一旦你完成了**Mozilla** 的安裝，你就再也不需要`root` 的權限了。 你可以直接打下面的指令來啓動**Mozilla**：

```
% mozilla
```

也可以直接打下列指令，直接啓動郵件和新聞閱讀器：

```
% mozilla -mail
```

6.2.2 Firefox

Firefox 是以**Mozilla** 原始碼為基礎的新世代瀏覽器。**Mozilla** 是一堆應用軟體的整合套裝，像是瀏覽器、郵件程式、聊天室軟體等所組成。**Firefox** 則純粹是瀏覽器，這也是為何它能短小精悍之故。

可以打下列指令來安裝：

```
#pkg_add -r firefox
```

也可以透過Ports Collection，以編譯原始碼的方式來安裝：

```
#cd /usr/ports/www/firefox
# make install clean
```

6.2.3 Firefox, Mozilla 的Java™ plugin 程式

Note: 本節以及下一節，均假設您已裝好**Firefox** 或**Mozilla**。

FreeBSD 基金會與Sun Microsystems 有達成授權協議，可以散播Java Runtime Environment(JRE™) 及Java Developement Kit(JDK™) 的FreeBSD 版binary(執行檔)。FreeBSD 版的binary 可以在FreeBSD 基金會 (http://www.freebsdfoundation.org/downloads/java.shtml) 網站下載。

要讓**Firefox** 或**Mozilla** 支援Java™ 的話，首先要先裝`java/javavmwrapper` 這個port。然後再去http://www.freebsdfoundation.org/downloads/java.shtml 下載**Diablo JRE**，並以pkg_add(1) 指令來安裝之。

接著啓動瀏覽器，在網址列輸入`about:plugins` 然後按**Enter** 鍵，就會顯示目前已裝的plugins 清單，這時應該就可以看到**Java** 也有列出來。若仍未看到的話，那就切換為`root` 帳號，打下列指令：

```
# ln -s /usr/local/diablo-jre1.5.0/plugin/i386/ns7/libjavaplugin_oji.so \
  /usr/local/lib/browser_plugins/
```

最後，重啓瀏覽器即可。

6.2.4 Firefox, Mozilla 的Macromedia® Flash™ plugin 程式

Macromedia® Flash™ plugin 程式並沒有FreeBSD 版，然而可以透過軟體層(wrapper)來執行Linux 版的plugin 程式。這個wrapper 同時也支援Adobe® Acrobat® 以及RealPlayer® plugin 等。

接下來去裝`www/linuxpluginwrapper`。linuxpluginwrapper 需要先裝一個很大的`emulators/linux_base`port。然後根據port 所指示的作法，去正確地設定你的`/etc/libmap.conf`！設定的範例檔案位於`/usr/local/share/examples/linuxpluginwrapper/` 的目錄底下。

下一步，則是裝`www/linux-flashplugin7`。裝好後，再啓動瀏覽器，在網址列輸入`about:plugins`，然後按**Enter** 鍵就會顯示目前已裝的plugin 清單。

若Flash plugin 沒出現的話，大多可能是因為漏了做symlink 連結之故。請切為`root` 帳號，打下列指令：

```
# ln -s /usr/local/lib/npapi/linux-flashplugin/libflashplayer.so \
  /usr/local/lib/browser_plugins/
# ln -s /usr/local/lib/npapi/linux-flashplugin/flashplayer.xpt \
  /usr/local/lib/browser_plugins/
```

最後，重啓瀏覽器應該就可看到了。

Note: linuxpluginwrapper 只能在i386 的系統架構下運行。

6.2.5 Opera

Opera 是個具備完整功能、符合標準的瀏覽器。它同時也具備了內建的郵件、新聞閱讀器、IRC、RSS/Atom feeds 閱讀器等。此外**Opera** 更是個輕量級、執行速度又快的瀏覽器。它在ports 中有兩種版本：「原生」的FreeBSD 版本還有在Linux 模擬模式下的版本。

要用**Opera** 的FreeBSD 版本來瀏覽網頁的話，用下面的指令安裝：

```
# pkg_add -r opera
```

有些FTP 站台並沒有全部的套件，但是打下面的指令就能從Ports Collection 中安裝：

```
# cd /usr/ports/www/opera
# make install clean
```

要安裝**Opera** 的Linux 版本的話，請將上面例子中的`opera` 替換成`linux-opera`。有些時候，Linux 的版本是十分有用的，像是只有Linux 版本外掛程式的時候。但在其他方面來說，FreeBSD 和Linux 的版本功能上是一樣的。

6.2.6 Konqueror

Konqueror 是**KDE** 桌面系統的一部分，但是它也可以藉由安裝`x11/kdebase3` 在KDE 環境以外使用。**Konqueror** 不只是個網頁瀏覽器，他同時也是檔案管理器和多媒體瀏覽器。

Konqueror 也有許多的外掛程式，這些外掛程式可以從`misc/konq-plugins` 中安裝。

Konqueror 也支援**Flash** 的外掛程式。如何安裝的說明請參閱：http://freebsd.kde.org/howto.php。

6.3 辦公室軟體

當開始進行辦公，新的使用者通常會去找好用的辦公室軟體或是好上手的文字處理器。目前有些桌面環境像是**KDE**已經提供了辦公軟體組合的套件。FreeBSD 提供了所需的所有辦公軟體，桌面環境也不例外。

這節涵蓋了下列的這些軟體：

軟體名稱	所需系統資源	從Ports 安裝的時間	主要相依套件
KOffice	少	長	**KDE**
AbiWord	少	短	**Gtk+** 或是**GNOME**
The Gimp	少	長	**Gtk+**
OpenOffice.org	多	很久	**JDK 1.4**, **Mozilla**

6.3.1 KOffice

KDE 社群在它的桌面環境裡頭提供了一個可以在**KDE** 外使用的辦公軟體組合。它包含了四種模組：**KWord** 是文字處理器，**KSpread** 是試算表程式，**KPresenter** 是簡報播放程式，另外**Karbon14** 讓你可

以產生圖形化的文件。[1]

在安裝最新版的**KOffice** 之前，請先確定你有最新版本的**KDE**。

若要用套件來安裝**KOffice**，請依照下面的指令：

```
# pkg_add -r koffice
```

如果套件不存在的話，你可以使用ports collection. 例如要安裝**KDE3** 中的**KOffice**，請使用下列指令安裝：

```
# cd /usr/ports/editors/koffice-kde3
# make install clean
```

6.3.2 AbiWord

AbiWord 是一個免費的文字處理軟體，外觀和感覺都近似於**Microsoft Word**。它適合處理文件、信件、報告、備忘錄等等。它也非常快速，包含了許多功能而且非常容易上手。

AbiWord 可以輸入或輸出許多檔案格式，包括一些有專利的格式，例如微軟(Microsoft)公司的`.doc` 格式。

AbiWord 也能用套件安裝，你可以用下列指令來安裝：

```
# pkg_add -r abiword
```

如果找不到套件的話，它也可以從Ports Collection 中編譯安裝。而Ports Collection 應該要保持在最新的狀態。**AbiWord** 可以透過下列方式編譯安裝：

```
# cd /usr/ports/editors/abiword
# make install clean
```

6.3.3 The GIMP

對於影像的編輯及修改來說，**GIMP** 是非常精緻的影像處理軟體。它可以當作簡單的繪圖軟體或是高品質的相片處理軟體。它支援為數眾多的外掛程式及指令稿(script-fu) 介面。**GIMP** 可以讀寫許多檔案格式。它也支援掃描器[2] 和手寫板。

譯註：**GIMP** 在目前是2.x 版，如果你想要安裝1.x 版的話，請用Ports Collection 中的`graphics/gimp1`。另外如果你已經使用習慣Adobe Photoshop，而且不習慣**GIMP** 介面的話，你也可以嘗試安裝`graphics/gimpshop`，它的使用介面十分類似Adobe Photoshop。

你可以使用下面指令安裝套件：

```
# pkg_add -r gimp
```

如果的你的FTP 站台沒有這個套件，你可以使用Ports Collection。在Ports Collection 的graphics (http://www.FreeBSD.org/ports/graphics.html) 目錄下也包含了**The Gimp Manual**(GIMP 使用手冊)。下面示範如何安裝這些程式：

```
# cd /usr/ports/graphics/gimp
# make install clean
# cd /usr/ports/graphics/gimp-manual-pdf
# make install clean
```

譯註：另外在Ports Collection 中也有一些外掛程式可以使用，例如說可以處理數位相機raw 檔案格式的`gimp-ufraw`。

Note: GIMP 使用手冊也有HTML 格式的，你可以在`graphics/gimp-manual-html` 中安裝。

6.3.4 OpenOffice.org

OpenOffice.org 包含了所有完整的辦公軟體組合：文字處理器、試算表、簡報軟體還有繪圖軟體。除了它的使用者介面非常類似其他的辦公軟體，他還能夠輸入和輸出許多熱門的檔案格式。它也包含了不同語言的使用者介面、拼字檢查和字典。

OpenOffice.org 的文字處理器使用XML 檔案格式來增加移植性及彈性。試算表程式支援巨集(macro)功能而且能夠使用外來的資料庫介面。**OpenOffice.org** 已經十分穩定，並且能夠在Windows, Solaris™, Linux, FreeBSD 及Mac OS X 等作業系統上面執行。想知道更多關於**OpenOffice.org** 的資訊可以在OpenOffice.org 網頁 (http://www.openoffice.org/) 上查詢。你也可以在FreeBSD OpenOffice.org 移植團隊 (http://porting.openoffice.org/freebsd/) 的網頁上查詢關於FreeBSD 上OpenOffice 特定的資訊或直接下載已編譯好的套件

要安裝**OpenOffice.org**，請用以下方式來執行：

```
# pkg_add -r openoffice.org
```

Note: 當你在使用FreeBSD -RELEASE 版本的時候，上面的作法應該行得通。要是其他的版本，你應該看一下FreeBSD **OpenOffice.org** 移植團隊的網站，並且用pkg_add(1) 安裝合適的套件。在這個站台都可以下載到穩定的釋出版(release)或開發中的版本。

當已經安裝完之後，你只要鍵入下面的指令就能執行**OpenOffice.org**：

```
% openoffice.org
```

譯註：端看你的版本，有時候需要輸入如openoffice.org-2.0.1 之類的指令，不過你也可以用shell 中的alias 或是用symbolic link 來處理。

Note: 在第一次啓動的時候，OpenOffice 會問到一些問題。而且在你的家目錄底下會自動建立`.openoffice.org2` 的資料夾。

如果無法取得**OpenOffice.org** 的套件，你仍然可以選擇從port 編譯。不過你必須謹記在心：編譯的過程會需要大量的磁碟空間且相當耗時。

```
# cd /usr/ports/editors/openoffice.org-2
# make install clean
```

Note: 如果你想要安裝本地化的版本，把前面的指令代換成下面的：

```
# make LOCALIZED_LANG=你的語言 install clean
```

你必須把*你的語言*換成正確的語言ISO-code [3] 所支援的語言代碼清單可以在port 目錄裡的`files/Makefile.localized` 檔案中找到。

一旦完成了上述步驟，**OpenOffice.org** 可用以下指令啓動：

```
% openoffice.org
```

6.4 文件閱覽器

近年來有些文件格式變得愈來愈流行，基本的系統中也許不會有這些格式所需的標準閱覽器。在這一節，我們來看看怎麼安裝這些軟體。

這張涵蓋了下列的軟體

軟體名稱	所需系統資源	從Ports 安裝時間	主要相依套件
Acrobat Reader	少	短	Linux 二進制相容模組
gv	少	短	**Xaw3d**
Xpdf	少	短	**FreeType**
GQview	少	短	**Gtk+** 或是**GNOME**

6.4.1 Acrobat Reader®

許多文件在散佈的時候都是用PDF 的檔案格式，這個格式是基於“可攜式文件格式(Portable Document Format)”。其中一個推薦的閱覽軟體就是**Acrobat Reader**，它是由Adobe 公司發行給Linux 使用的版本。因為FreeBSD 也可以執行Linux 二進位檔案，所以它也能在FreeBSD 上面執行。

要從Ports Collection 中安裝**Acrobat Reader 7** 只要：

```
# cd /usr/ports/print/acroread7
# make install clean
```

因為授權的限制，所以不提供編譯好的套件。

6.4.2 gv

gv是PostScript 和PDF 的閱覽器。它建構於**ghostview**的基礎上，不過因為使用**Xaw3d** 函式庫，所以外觀看起來比較漂亮。gv 速度快，介面簡潔並且有許多功能，比如說方向性、紙張大小、縮放比例、和反鋸齒(antialias)等。而且幾乎所有的使用都可以從鍵盤或滑鼠來完成。

用套件來安裝**gv**，使用下列指令：

```
# pkg_add -r gv
```

如果你不能取得套件，你可以使用Ports Collection：

```
# cd /usr/ports/print/gv
# make install clean
```

6.4.3 Xpdf

如果你想要一個小型的FreeBSD PDF 閱覽軟體，**Xpdf**是個輕量級而且有效率的閱覽器。它只需要非常少的資源而且十分穩定。它只使用標準的X 字型而不需要**Motif** 或是其他的X 工具組(toolkit)。

用套件來安裝**Xpdf**，使用下列指令：

```
# pkg_add -r xpdf
```

如果套件不存在或是你偏好使用Ports Collection，使用以下指令：

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

一旦完成了安裝，你可以啓動**Xpdf** 並且使用滑鼠右鍵去使用選單。

6.4.4 GQview

GQview 是影像管理軟體。你可以用單鍵來閱覽檔案、啓動額外的編輯器、縮圖預覽等功能。它也有幻燈片播放(slideshow)及一些基本的檔案操作功能。你可用GQview 管理影像集並能輕鬆地找出重複的檔案。**GQview** 能夠使用全螢幕觀看並支援國際化。

如果你想要安裝**GQview**的套件，請使用下列指令：

```
# pkg_add -r gqview
```

如果套件無法取得，或是你比較喜歡使用Ports Collection，只要：

```
# cd /usr/ports/graphics/gqview
# make install clean
```

6.5 財務

如果有任何理由你想要在你的FreeBSD 桌面環境上管理你的個人財務，這裡有一些功能強大、使用簡單的應用程式可供安裝。這些財務管理軟體之中有些是相容於流行的**Quicken®** 或**Excel** 文件。

這節涵蓋了下面這些軟體：

軟體名稱	所需系統資源	從Ports 安裝的時間	主要的相依套件
GnuCash	少	長	**GNOME**
Gnumeric	少	長	**GNOME**
Abacus	少	短	**Tcl/Tk**

6.5.1 GnuCash

GnuCash 是**GNOME** 團隊努力成果中的一部分，而**GNOME** 主要是提供終端使用者(end-users) 親切而强大的桌面應用程式。使用**GnuCash**，你可以持續紀錄你的收入及花費、你的銀行帳戶、或是你的股票證券等。它的特性是介面直覺但功能仍非常專業。

GnuCash 提供了一個智慧的註冊器、帳戶層級系統、許多快速鍵及自動完成(auto-completion)模式。它也能分開單一的報表至數個詳細的部份。**GnuCash** 也能夠輸入及合併**Quicken** QIF 檔案。它也能處理大部分國際的日期及通用貨幣之格式。

要安裝**GnuCash** 到你的系統中，只要做下列步驟：

```
# pkg_add -r gnucash
```

如果不能取得套件，你可以使用Ports Collection:

```
# cd /usr/ports/finance/gnucash
# make install clean
```

6.5.2 Gnumeric

Gnumeric 是**GNOME** 桌面環境中的試算表。它的特點是能夠根據儲存格格式(cell format)及自動補齊的系統，來方便自動地「猜出」使用者的輸入。它也能夠輸入許多熱門的檔案格式，像是**Excel**, **Lotus 1-2-3**, 或是**Quattro Pro**。**Gnumeric** 支援使用`math/guppi` 繪圖軟體來繪圖。它有許多內建的函數而且允許一般的儲存格格式，像是: 數字、貨幣、日期、時間及其他格式等。

要用套件安裝**Gnumeric**，只要打以下指令：

```
# pkg_add -r gnumeric
```

如果套件不存在，你可以做下面的步驟來使用Ports Collection 編譯安裝：

```
# cd /usr/ports/math/gnumeric
# make install clean
```

6.5.3 Abacus

Abacus 是個小巧又使用簡單的試算表。它包含了許多內建的函數，在相關的領域如統計學、財務、數學中很實用。它也可以輸出輸入**Excel** 的檔案格式。另外**Abacus**也能夠輸出PostScript 格式。

從套件安裝**Abacus** 只要做：

```
# pkg_add -r abacus
```

如果套件不能取得的話，你可以使用Ports Collection，並用以下指令：

```
# cd /usr/ports/deskutils/abacus
# make install clean
```

6.6 摘要

雖然FreeBSD 是因為效能及穩定性而在ISP 之間很流行，不過它也可以完全當作桌面環境(desktop)來使用，並不侷限於使用在伺服器上面。目前有數千種應用程式的套件(packages) (http://www.FreeBSD.org/where.html) 或ports (http://www.FreeBSD.org/ports/index.html), 可供使用，你可以根據你的需求打造出一個完美的桌面環境。

下面是這章涵蓋的所有桌面應用軟體之快速回顧表：

軟體名稱	套件名稱	Ports 名稱
Mozilla	`mozilla`	`www/mozilla`
Opera	`opera`	`www/opera`
Firefox	`firefox`	`www/firefox`
KOffice	`koffice-kde3`	`editors/koffice-kde3`
AbiWord	`abiword`	`editors/abiword`
The GIMP	`gimp`	`graphics/gimp`
OpenOffice.org	`openoffice`	`editors/openoffice-1.1`
Acrobat Reader	`acroread`	`print/acroread7`
gv	`gv`	`print/gv`
Xpdf	`xpdf`	`graphics/xpdf`
GQview	`gqview`	`graphics/gqview`
GnuCash	`gnucash`	`finance/gnucash`
Gnumeric	`gnumeric`	`math/gnumeric`
Abacus	`abacus`	`deskutils/abacus`

Notes

1. 譯註：Karbon14 是向量繪圖軟體，以前叫Kontour，更早之前稱為Killustrator。
2. 譯註：你必須透過sane-frontends 或xsane 來掃描
3. 譯註：臺灣正體中文使用者為zh-TW。

Chapter 7 多媒體影音娛樂(Multimedia)

Edited by Ross Lippert.

7.1 概述

FreeBSD 廣泛地支援各種音效卡，讓您可以享受來自電腦上的高傳真音質(Hi-Fi)，此外還包括了錄製和播放MPEG Audio Layer 3 (MP3)、WAV、以及Ogg Vorbis 等許多種格式聲音的能力。同時FreeBSD Ports Collection 也包括了許多的應用程式，讓您可以錄音、編修音效以及控制MIDI 配備。

要是喜歡動手嘗試不同的體驗，FreeBSD 也能播放一般的視訊檔和DVD。編碼、轉換和播放視訊的程式比起處理聲音的程式略少一些。例如，在撰寫這章時，FreeBSD Ports Collection 中還沒有類似`audio/sox` 那樣好用的編碼工具，能夠用來轉換不同的格式。不過，這個領域的軟體研發進展是相當迅速的。

本章將介紹設定音效卡的必要步驟。先前介紹到的X11 (Chapter 5) 安裝和設定裡，已經講到了顯示卡的部份，但要想有更好的播放效果，仍需要一些細部調整。

讀完這章，您將了解：

- 如何設定系統，以正確識別音效卡。
- 如何運用樣本程式，以測試音效卡是否正常運作。
- 如何解決音效卡的設定問題。
- 如何播放、錄製MP3 及其他聲音檔案格式。
- X server 是如何支援顯示卡。
- Ports Collections 內有哪些好用的影像播放、錄製軟體。
- 如何播放DVD 的`.mpg` 及`.avi` 檔
- 如何從CD 和DVD 中擷取(rip)檔案。
- 如何設定電視卡
- 如何設定掃描器

在閱讀這章之前，您應當了解：

- 知道如何設定、安裝新的kernel (Chapter 8)。

> **Warning:** 如果要用mount(8) 指令來mount 音樂光碟的話，通常會發生錯誤，甚至導致*kernel panic*。這是因為音樂光碟是特殊編碼，而非一般的ISO 檔案系統之故。

7.2 設定音效卡

Contributed by Moses Moore. 加强FreeBSD 5.X 的內容：Marc Fonvieille.

7.2.1 設定系統

開始設定之前，必須先知道你的音效卡型號、晶片為何，以及是PCI 或ISA 規格。FreeBSD 有支援許多種的PCI、ISA 音效卡，請檢查支援的音效硬體表Hardware Notes (http://www.FreeBSD.org/releases/8.0R/hardware.html)，以確認你的音效卡是否支援。本文也會提到相對應該卡的驅動程式。

要使用音效卡，必須要載入正確的驅動程式才行。有兩種方式都可以完成這動作，最簡單方式就是以kldload(8) 來輕鬆載入kernel 動態模組(module)，像是下列指令：

```
# kldload snd_emu10k1
```

或者把相關驅動程式加到`/boot/loader.conf` 檔，像是：

```
snd_emu10k1_load="YES"
```

上面例子是給Creative SoundBlaster® Live! 音效卡使用的。其他可用的音效卡驅動程式模組，可參考`/boot/defaults/loader.conf` 範例。若不確定到底該用哪一種驅動程式，那麼可以試試載入`snd_driver` 模組看看：

```
# kldload snd_driver
```

This is a metadriver loading the most common device drivers at once. This speeds up the search for the correct driver. It is also possible to load all sound drivers via the `/boot/loader.conf` facility.

If you wish to find out the driver selected for your soundcard after loading the `snd_driver` metadriver, you may check the `/dev/sndstat` file with the `cat /dev/sndstat` command.

Note: Under FreeBSD 4.X, to load all sound drivers, you have to load the `snd` module instead of `snd_driver`.

A second method is to statically compile in support for your sound card in your kernel. The section below provides the information you need to add support for your hardware in this manner. For more information about recompiling your kernel, please see Chapter 8.

7.2.1.1 Configuring a Custom Kernel with Sound Support

The first thing to do is adding the generic audio driver sound(4) to the kernel, for that you will need to add the following line to the kernel configuration file:

```
device sound
```

Under FreeBSD 4.X, you would use the following line:

```
device pcm
```

Then we have to add the support for our sound card. Therefore, we need to know which driver supports the card. Check the supported audio devices list of the Hardware Notes

(http://www.FreeBSD.org/releases/8.0R/hardware.html), to determine the correct driver for your sound card. For example, a Creative SoundBlaster Live! sound card is supported by the snd_emu10k1(4) driver. To add the support for this card, use the following:

```
device snd_emu10k1
```

Be sure to read the manual page of the driver for the syntax to use. Information regarding the syntax of sound drivers in the kernel configuration can also be found in the `/usr/src/sys/conf/NOTES` file (`/usr/src/sys/i386/conf/LINT` for FreeBSD 4.X).

Non-PnP ISA cards may require you to provide the kernel with information on the sound card settings (IRQ, I/O port, etc). This is done via the `/boot/device.hints` file. At system boot, the loader(8) will read this file and pass the settings to the kernel. For example, an old Creative SoundBlaster 16 ISA non-PnP card will use the snd_sbc(4) driver in conjunction with snd_sb16(4). For this card the following lines have to be added to the kernel configuration file:

```
device snd_sbc
device snd_sb16
```

as well as the following in `/boot/device.hints`:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

In this case, the card uses the `0x220` I/O port and the IRQ `5`.

The syntax used in the `/boot/device.hints` file is covered in the sound driver manual page. On FreeBSD 4.X, these settings are directly written in the kernel configuration file. In the case of our ISA card, we would only use this line:

```
device sbc0 at isa? port 0x220 irq 5 drq 1 flags 0x15
```

The settings shown above are the defaults. In some cases, you may need to change the IRQ or the other settings to match your card. See the snd_sbc(4) manual page for more information.

> **Note:** Under FreeBSD 4.X, some systems with built-in motherboard sound devices may require the following option in the kernel configuration:
>
> ```
> options PNPBIOS
> ```

7.2.2 Testing the Sound Card

After rebooting with the modified kernel, or after loading the required module, the sound card should appear in your system message buffer (dmesg(8)) as something like:

```
pcm0: <Intel ICH3 (82801CA)> port 0xdc80-0xdcbf,0xd800-0xd8ff irq 5 at device 31.5 on pci0
pcm0: [GIANT-LOCKED]
```

```
pcm0: <Cirrus Logic CS4205 AC97 Codec>
```

The status of the sound card may be checked via the `/dev/sndstat` file:

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm)
Installed devices:
pcm0: <Intel ICH3 (82801CA)> at io 0xd800, 0xdc80 irq 5 bufsz 16384
kld snd_ich (1p/2r/0v channels duplex default)
```

The output from your system may vary. If no `pcm` devices show up, go back and review what was done earlier. Go through your kernel configuration file again and make sure the correct device is chosen. Common problems are listed in Section 7.2.2.1.

If all goes well, you should now have a functioning sound card. If your CD-ROM or DVD-ROM drive is properly coupled to your sound card, you can put a CD in the drive and play it with cdcontrol(1):

```
% cdcontrol -f /dev/acd0 play 1
```

Various applications, such as `audio/workman` can provide a friendlier interface. You may want to install an application such as `audio/mpg123` to listen to MP3 audio files. A quick way to test the card is sending data to the `/dev/dsp`, like this:

```
% cat filename > /dev/dsp
```

where *`filename`* can be any file. This command line should produce some noise, confirming the sound card is actually working.

> **Note:** FreeBSD 4.X users need to create the sound card device nodes before being able to use it. If the card showed up in message buffer as `pcm0`, you will have to run the following as `root`:
>
> ```
> # cd /dev
> # sh MAKEDEV snd0
> ```
>
> If the card detection returned `pcm1`, follow the same steps as shown above, replacing `snd0` with `snd1`.
>
> `MAKEDEV` will create a group of device nodes that will be used by the different sound related applications.

Sound card mixer levels can be changed via the mixer(8) command. More details can be found in the mixer(8) manual page.

7.2.2.1 Common Problems

Error	Solution
"`unsupported subdevice XX`"	One or more of the device nodes was not created correctly. Repeat the steps above.
"`sb_dspwr(XX) timed out`"	The I/O port is not set correctly.
"`bad irq XX`"	The IRQ is set incorrectly. Make sure that the set IRQ and the sound IRQ are the same.

Error	Solution
“xxx: gus pcm not attached, out of memory”	There is not enough available memory to use the device.
“xxx: can’t open /dev/dsp!”	Check with `fstat | grep dsp` if another application is holding the device open. Noteworthy troublemakers are **esound** and **KDE**’s sound support.

7.2.3 Utilizing Multiple Sound Sources

Contributed by Munish Chopra.

It is often desirable to have multiple sources of sound that are able to play simultaneously, such as when **esound** or **artsd** do not support sharing of the sound device with a certain application.

FreeBSD lets you do this through *Virtual Sound Channels*, which can be set with the sysctl(8) facility. Virtual channels allow you to multiplex your sound card’s playback channels by mixing sound in the kernel.

To set the number of virtual channels, there are two sysctl knobs which, if you are the `root` user, can be set like this:

```
# sysctl hw.snd.pcm0.vchans=4
# sysctl hw.snd.maxautovchans=4
```

The above example allocates four virtual channels, which is a practical number for everyday use. `hw.snd.pcm0.vchans` is the number of virtual channels `pcm0` has, and is configurable once a device has been attached. `hw.snd.maxautovchans` is the number of virtual channels a new audio device is given when it is attached using kldload(8). Since the `pcm` module can be loaded independently of the hardware drivers, `hw.snd.maxautovchans` can store how many virtual channels any devices which are attached later will be given.

> **Note:** You cannot change the number of virtual channels for a device while it is in use. First close any programs using the device, such as music players or sound daemons.

If you are not using devfs(5), you will have to point your applications at `/dev/dsp0.`*x*, where *x* is 0 to 3 if `hw.snd.pcm.0.vchans` is set to 4 as in the above example. On a system using devfs(5), the above will automatically be allocated transparently to the user.

7.2.4 設定預設(Mixer Channel)的音量大小

Contributed by Josef El-Rayes.

> **Note:** 本功能只有在FreeBSD 5.3-RELEASE 及之後版本才有支援。

The default values for the different mixer channels are hardcoded in the sourcecode of the pcm(4) driver. There are a lot of different applications and daemons that allow you to set values for the mixer they remember and set each time they are started, but this is not a clean solution, we want to have default values at the driver level. This is accomplished by defining the appropriate values in `/boot/device.hints`. E.g.:

```
hint.pcm.0.vol="100"
```

This will set the volume channel to a default value of 100, when the pcm(4) module is loaded.

7.3 MP3 音樂

Contributed by Chern Lee.

MP3 (MPEG Layer 3 Audio) accomplishes near CD-quality sound, leaving no reason to let your FreeBSD workstation fall short of its offerings.

7.3.1 MP3 Players

By far, the most popular X11 MP3 player is **XMMS** (X Multimedia System). **Winamp** skins can be used with **XMMS** since the GUI is almost identical to that of Nullsoft's **Winamp**. **XMMS** also has native plug-in support.

XMMS can be installed from the `multimedia/xmms` port or package.

XMMS' interface is intuitive, with a playlist, graphic equalizer, and more. Those familiar with **Winamp** will find **XMMS** simple to use.

The `audio/mpg123` port is an alternative, command-line MP3 player.

mpg123 can be run by specifying the sound device and the MP3 file on the command line, as shown below:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2 and 3.
Version 0.59r (1999/Jun/15). Written and copyrights by Michael Hipp.
Uses code from various people. See 'README' for more!
THIS SOFTWARE COMES WITH ABSOLUTELY NO WARRANTY! USE AT YOUR OWN RISK!

Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

`/dev/dsp1.0` should be replaced with the `dsp` device entry on your system.

7.3.2 Ripping CD Audio Tracks

Before encoding a CD or CD track to MP3, the audio data on the CD must be ripped onto the hard drive. This is done by copying the raw CDDA (CD Digital Audio) data to WAV files.

The `cdda2wav` tool, which is a part of the `sysutils/cdrtools` suite, is used for ripping audio information from CDs and the information associated with them.

With the audio CD in the drive, the following command can be issued (as `root`) to rip an entire CD into individual (per track) WAV files:

```
# cdda2wav -D 0,1,0 -B
```

cdda2wav will support ATAPI (IDE) CDROM drives. To rip from an IDE drive, specify the device name in place of the SCSI unit numbers. For example, to rip track 7 from an IDE drive:

```
# cdda2wav -D /dev/acd0a -t 7
```

The `-D 0,1,0` indicates the SCSI device `0,1,0`, which corresponds to the output of `cdrecord -scanbus`.

To rip individual tracks, make use of the `-t` option as shown:

```
# cdda2wav -D 0,1,0 -t 7
```

This example rips track seven of the audio CDROM. To rip a range of tracks, for example, track one to seven, specify a range:

```
# cdda2wav -D 0,1,0 -t 1+7
```

The utility dd(1) can also be used to extract audio tracks on ATAPI drives, read Section 18.6.5 for more information on that possibility.

7.3.3 Encoding MP3s

Nowadays, the mp3 encoder of choice is **lame**. **Lame** can be found at `audio/lame` in the ports tree.

Using the ripped WAV files, the following command will convert `audio01.wav` to `audio01.mp3`:

```
# lame -h -b 128 \
--tt "Foo Song Title" \
--ta "FooBar Artist" \
--tl "FooBar Album" \
--ty "2001" \
--tc "Ripped and encoded by Foo" \
--tg "Genre" \
audio01.wav audio01.mp3
```

128 kbits seems to be the standard MP3 bitrate in use. Many enjoy the higher quality 160, or 192. The higher the bitrate, the more disk space the resulting MP3 will consume--but the quality will be higher. The `-h` option turns on the "higher quality but a little slower" mode. The options beginning with `--t` indicate ID3 tags, which usually contain song information, to be embedded within the MP3 file. Additional encoding options can be found by consulting the lame man page.

7.3.4 Decoding MP3s

In order to burn an audio CD from MP3s, they must be converted to a non-compressed WAV format. Both **XMMS** and **mpg123** support the output of MP3 to an uncompressed file format.

Writing to Disk in **XMMS**:

1. Launch **XMMS**.
2. Right-click on the window to bring up the **XMMS** menu.
3. Select `Preference` under `Options`.

4. Change the Output Plugin to “Disk Writer Plugin”.
5. Press `Configure`.
6. Enter (or choose browse) a directory to write the uncompressed files to.
7. Load the MP3 file into **XMMS** as usual, with volume at 100% and EQ settings turned off.
8. Press `Play` ——**XMMS** will appear as if it is playing the MP3, but no music will be heard. It is actually playing the MP3 to a file.
9. Be sure to set the default Output Plugin back to what it was before in order to listen to MP3s again.

Writing to stdout in **mpg123**:

1. Run `mpg123 -s` *`audio01.mp3`* `> audio01.pcm`

XMMS writes a file in the WAV format, while **mpg123** converts the MP3 into raw PCM audio data. Both of these formats can be used with **cdrecord** to create audio CDs. You have to use raw PCM with burncd(8). If you use WAV files, you will notice a small tick sound at the beginning of each track, this sound is the header of the WAV file. You can simply remove the header of a WAV file with the utility **SoX** (it can be installed from the `audio/sox` port or package):

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

Read Section 18.6 for more information on using a CD burner in FreeBSD.

7.4 播放影片

Contributed by Ross Lippert.

Video playback is a very new and rapidly developing application area. Be patient. Not everything is going to work as smoothly as it did with sound.

Before you begin, you should know the model of the video card you have and the chip it uses. While **Xorg** and **XFree86** support a wide variety of video cards, fewer give good playback performance. To obtain a list of extensions supported by the X server using your card use the command xdpyinfo(1) while X11 is running.

It is a good idea to have a short MPEG file which can be treated as a test file for evaluating various players and options. Since some DVD players will look for DVD media in `/dev/dvd` by default, or have this device name hardcoded in them, you might find it useful to make symbolic links to the proper devices:

```
# ln -sf /dev/acd0c /dev/dvd
# ln -sf /dev/racd0c /dev/rdvd
```

On FreeBSD 5.X, which uses devfs(5) there is a slightly different set of recommended links:

```
# ln -sf /dev/acd0 /dev/dvd
# ln -sf /dev/acd0 /dev/rdvd
```

Note that due to the nature of devfs(5), manually created links like these will not persist if you reboot your system. In order to create the symbolic links automatically whenever you boot your system, add the following lines to `/etc/devfs.conf`:

```
link acd0 dvd
link acd0 rdvd
```

Additionally, DVD decryption, which requires invoking special DVD-ROM functions, requires write permission on the DVD devices.

Some of the ports discussed rely on the following kernel options to build correctly. Before attempting to build, add this option to the kernel configuration file, build a new kernel, and reboot:

```
options CPU_ENABLE_SSE
```

> **Note:** On FreeBSD 4.X `options USER_LDT` should be added to the kernel configuration file. This option is not available on FreeBSD 5.X and later version.

To enhance the shared memory X11 interface, it is recommended that the values of some sysctl(8) variables should be increased:

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

7.4.1 Determining Video Capabilities

There are several possible ways to display video under X11. What will really work is largely hardware dependent. Each method described below will have varying quality across different hardware. Secondly, the rendering of video in X11 is a topic receiving a lot of attention lately, and with each version of **Xorg**, or of **XFree86**, there may be significant improvement.

A list of common video interfaces:

1. X11: normal X11 output using shared memory.
2. XVideo: an extension to the X11 interface which supports video in any X11 drawable.
3. SDL: the Simple Directmedia Layer.
4. DGA: the Direct Graphics Access.
5. SVGAlib: low level console graphics layer.

7.4.1.1 XVideo

Xorg and **XFree86 4.X** have an extension called *XVideo* (aka Xvideo, aka Xv, aka xv) which allows video to be directly displayed in drawable objects through a special acceleration. This extension provides very good quality playback even on low-end machines.

To check whether the extension is running, use `xvinfo`:

```
% xvinfo
```

XVideo is supported for your card if the result looks like:

```
X-Video Extension version 2.2
screen #0
```

```
Adaptor #0: "Savage Streams Engine"
  number of ports: 1
  port base: 43
  operations supported: PutImage
  supported visuals:
    depth 16, visualID 0x22
    depth 16, visualID 0x23
  number of attributes: 5
    "XV_COLORKEY" (range 0 to 16777215)
            client settable attribute
            client gettable attribute (current value is 2110)
    "XV_BRIGHTNESS" (range -128 to 127)
            client settable attribute
            client gettable attribute (current value is 0)
    "XV_CONTRAST" (range 0 to 255)
            client settable attribute
            client gettable attribute (current value is 128)
    "XV_SATURATION" (range 0 to 255)
            client settable attribute
            client gettable attribute (current value is 128)
    "XV_HUE" (range -180 to 180)
            client settable attribute
            client gettable attribute (current value is 0)
  maximum XvImage size: 1024 x 1024
  Number of image formats: 7
    id: 0x32595559 (YUY2)
      guid: 59555932-0000-0010-8000-00aa00389b71
      bits per pixel: 16
      number of planes: 1
      type: YUV (packed)
    id: 0x32315659 (YV12)
      guid: 59563132-0000-0010-8000-00aa00389b71
      bits per pixel: 12
      number of planes: 3
      type: YUV (planar)
    id: 0x30323449 (I420)
      guid: 49343230-0000-0010-8000-00aa00389b71
      bits per pixel: 12
      number of planes: 3
      type: YUV (planar)
    id: 0x36315652 (RV16)
      guid: 52563135-0000-0000-0000-000000000000
      bits per pixel: 16
      number of planes: 1
      type: RGB (packed)
      depth: 0
      red, green, blue masks: 0x1f, 0x3e0, 0x7c00
    id: 0x35315652 (RV15)
      guid: 52563136-0000-0000-0000-000000000000
      bits per pixel: 16
      number of planes: 1
      type: RGB (packed)
      depth: 0
```

```
    red, green, blue masks: 0x1f, 0x7e0, 0xf800
  id: 0x31313259 (Y211)
    guid: 59323131-0000-0010-8000-00aa00389b71
    bits per pixel: 6
    number of planes: 3
    type: YUV (packed)
  id: 0x0
    guid: 00000000-0000-0000-0000-000000000000
    bits per pixel: 0
    number of planes: 0
    type: RGB (packed)
    depth: 1
    red, green, blue masks: 0x0, 0x0, 0x0
```

Also note that the formats listed (YUV2, YUV12, etc) are not present with every implementation of XVideo and their absence may hinder some players.

If the result looks like:

```
X-Video Extension version 2.2
screen #0
no adaptors present
```

Then XVideo is probably not supported for your card.

If XVideo is not supported for your card, this only means that it will be more difficult for your display to meet the computational demands of rendering video. Depending on your video card and processor, though, you might still be able to have a satisfying experience. You should probably read about ways of improving performance in the advanced reading Section 7.4.3.

7.4.1.2 Simple Directmedia Layer

The Simple Directmedia Layer, SDL, was intended to be a porting layer between Microsoft Windows, BeOS, and UNIX, allowing cross-platform applications to be developed which made efficient use of sound and graphics. The SDL layer provides a low-level abstraction to the hardware which can sometimes be more efficient than the X11 interface.

The SDL can be found at `devel/sdl12`.

7.4.1.3 Direct Graphics Access

Direct Graphics Access is an X11 extension which allows a program to bypass the X server and directly alter the framebuffer. Because it relies on a low level memory mapping to effect this sharing, programs using it must be run as `root`.

The DGA extension can be tested and benchmarked by dga(1). When `dga` is running, it changes the colors of the display whenever a key is pressed. To quit, use **q**.

7.4.2 Ports and Packages Dealing with Video

This section discusses the software available from the FreeBSD Ports Collection which can be used for video playback. Video playback is a very active area of software development, and the capabilities of various applications are bound to diverge somewhat from the descriptions given here.

Firstly, it is important to know that many of the video applications which run on FreeBSD were developed as Linux applications. Many of these applications are still beta-quality. Some of the problems that you may encounter with video packages on FreeBSD include:

1. An application cannot playback a file which another application produced.
2. An application cannot playback a file which the application itself produced.
3. The same application on two different machines, rebuilt on each machine for that machine, plays back the same file differently.
4. A seemingly trivial filter like rescaling of the image size results in very bad artifacts from a buggy rescaling routine.
5. An application frequently dumps core.
6. Documentation is not installed with the port and can be found either on the web or under the port's `work` directory.

Many of these applications may also exhibit "Linux-isms". That is, there may be issues resulting from the way some standard libraries are implemented in the Linux distributions, or some features of the Linux kernel which have been assumed by the authors of the applications. These issues are not always noticed and worked around by the port maintainers, which can lead to problems like these:

1. The use of `/proc/cpuinfo` to detect processor characteristics.
2. A misuse of threads which causes a program to hang upon completion instead of truly terminating.
3. Software not yet in the FreeBSD Ports Collection which is commonly used in conjunction with the application.

So far, these application developers have been cooperative with port maintainers to minimize the work-arounds needed for port-ing.

7.4.2.1 MPlayer

MPlayer is a recently developed and rapidly developing video player. The goals of the **MPlayer** team are speed and flexibility on Linux and other Unices. The project was started when the team founder got fed up with bad playback performance on then available players. Some would say that the graphical interface has been sacrificed for a streamlined design. However, once you get used to the command line options and the key-stroke controls, it works very well.

7.4.2.1.1 Building MPlayer

MPlayer resides in `multimedia/mplayer`. **MPlayer** performs a variety of hardware checks during the build process, resulting in a binary which will not be portable from one system to another. Therefore, it is important to build it from ports and not to use a binary package. Additionally, a number of options can be specified in the `make` command line, as described in the `Makefile` and at the start of the build:

```
# cd /usr/ports/multimedia/mplayer
```

```
# make
N - O - T - E

Take a careful look into the Makefile in order
to learn how to tune mplayer towards you personal preferences!
For example,
make WITH_GTK1
builds MPlayer with GTK1-GUI support.
If you want to use the GUI, you can either install
/usr/ports/multimedia/mplayer-skins
or download official skin collections from
http://www.mplayerhq.hu/homepage/dload.html
```

The default port options should be sufficient for most users. However, if you need the XviD codec, you have to specify the `WITH_XVID` option in the command line. The default DVD device can also be defined with the `WITH_DVD_DEVICE` option, by default `/dev/acd0` will be used.

As of this writing, the **MPlayer** port will build its HTML documentation and two executables, `mplayer`, and `mencoder`, which is a tool for re-encoding video.

The HTML documentation for **MPlayer** is very informative. If the reader finds the information on video hardware and interfaces in this chapter lacking, the **MPlayer** documentation is a very thorough supplement. You should definitely take the time to read the **MPlayer** documentation if you are looking for information about video support in UNIX.

7.4.2.1.2 Using MPlayer

Any user of **MPlayer** must set up a `.mplayer` subdirectory of her home directory. To create this necessary subdirectory, you can type the following:

```
% cd /usr/ports/multimedia/mplayer
% make install-user
```

The command options for `mplayer` are listed in the manual page. For even more detail there is HTML documentation. In this section, we will describe only a few common uses.

To play a file, such as *`testfile.avi`*, through one of the various video interfaces set the `-vo` option:

```
% mplayer -vo xv testfile.avi

% mplayer -vo sdl testfile.avi

% mplayer -vo x11 testfile.avi

# mplayer -vo dga testfile.avi

# mplayer -vo 'sdl:dga' testfile.avi
```

It is worth trying all of these options, as their relative performance depends on many factors and will vary significantly with hardware.

To play from a DVD, replace the `testfile.avi` with `dvd://`*`N`* `-dvd-device` *`DEVICE`* where *`N`* is the title number to play and *`DEVICE`* is the device node for the DVD-ROM. For example, to play title 3 from `/dev/dvd`:

```
# mplayer -vo xv dvd://3 -dvd-device /dev/dvd
```

> **Note:** The default DVD device can be defined during the build of the **MPlayer** port via the `WITH_DVD_DEVICE` option. By default, this device is `/dev/acd0`. More details can be found in the port `Makefile`.

To stop, pause, advance and so on, consult the keybindings, which are output by running `mplayer -h` or read the manual page.

Additional important options for playback are: `-fs -zoom` which engages the fullscreen mode and `-framedrop` which helps performance.

In order for the mplayer command line to not become too large, the user can create a file `.mplayer/config` and set default options there:

```
vo=xv
fs=yes
zoom=yes
```

Finally, `mplayer` can be used to rip a DVD title into a `.vob` file. To dump out the second title from a DVD, type this:

```
# mplayer -dumpstream -dumpfile out.vob dvd://2 -dvd-device /dev/dvd
```

The output file, `out.vob`, will be MPEG and can be manipulated by the other packages described in this section.

7.4.2.1.3 mencoder

Before using `mencoder` it is a good idea to familiarize yourself with the options from the HTML documentation. There is a manual page, but it is not very useful without the HTML documentation. There are innumerable ways to improve quality, lower bitrate, and change formats, and some of these tricks may make the difference between good or bad performance. Here are a couple of examples to get you going. First a simple copy:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

Improper combinations of command line options can yield output files that are unplayable even by `mplayer`. Thus, if you just want to rip to a file, stick to the `-dumpfile` in `mplayer`.

To convert `input.avi` to the MPEG4 codec with MPEG3 audio encoding (`audio/lame` is required):

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \
  -ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

This has produced output playable by `mplayer` and `xine`.

`input.avi` can be replaced with `dvd://1 -dvd-device /dev/dvd` and run as `root` to re-encode a DVD title directly. Since you are likely to be dissatisfied with your results the first time around, it is recommended you dump the title to a file and work on the file.

7.4.2.2 The xine Video Player

The **xine** video player is a project of wide scope aiming not only at being an all in one video solution, but also in producing a reusable base library and a modular executable which can be extended with plugins. It comes both as a package and as a port, `multimedia/xine`.

The **xine** player is still very rough around the edges, but it is clearly off to a good start. In practice, **xine** requires either a fast CPU with a fast video card, or support for the XVideo extension. The GUI is usable, but a bit clumsy.

As of this writing, there is no input module shipped with **xine** which will play CSS encoded DVD's. There are third party builds which do have modules for this built in them, but none of these are in the FreeBSD Ports Collection.

Compared to **MPlayer**, **xine** does more for the user, but at the same time, takes some of the more fine-grained control away from the user. The **xine** video player performs best on XVideo interfaces.

By default, **xine** player will start up in a graphical user interface. The menus can then be used to open a specific file:

```
% xine
```

Alternatively, it may be invoked to play a file immediately without the GUI with the command:

```
% xine -g -p mymovie.avi
```

7.4.2.3 The transcode Utilities

The software **transcode** is not a player, but a suite of tools for re-encoding video and audio files. With **transcode**, one has the ability to merge video files, repair broken files, using command line tools with `stdin/stdout` stream interfaces.

A great number of options can be specified during the build from the `multimedia/transcode` port, we recommend the following command line to build **transcode**:

```
# make WITH_OPTIMIZED_CFLAGS=yes WITH_LIBA52=yes WITH_LAME=yes WITH_OGG=yes \
WITH_MJPEG=yes -DWITH_XVID=yes
```

The proposed settings should be sufficient for most users.

To illustrate `transcode` capacities, one example to show how to convert a DivX file into a PAL MPEG-1 file (PAL VCD):

```
% transcode -i input.avi -V --export_prof vcd-pal -o output_vcd
% mplex -f 1 -o output_vcd.mpg output_vcd.m1v output_vcd.mpa
```

The resulting MPEG file, `output_vcd.mpg`, is ready to be played with **MPlayer**. You could even burn the file on a CD-R media to create a Video CD, in this case you will need to install and use both `multimedia/vcdimager` and `sysutils/cdrdao` programs.

There is a manual page for `transcode`, but you should also consult the transcode wiki (http://www.transcoding.org/cgi-bin/transcode) for further information and examples.

7.4.3 Further Reading

The various video software packages for FreeBSD are developing rapidly. It is quite possible that in the near future many of the problems discussed here will have been resolved. In the mean time, those who want to get the very most out of FreeBSD's A/V capabilities will have to cobble together knowledge from several FAQs and tutorials and use a few different applications. This section exists to give the reader pointers to such additional information.

The MPlayer documentation (http://www.mplayerhq.hu/DOCS/) is very technically informative. These documents should probably be consulted by anyone wishing to obtain a high level of expertise with UNIX video. The **MPlayer** mailing list is hostile to anyone who has not bothered to read the documentation, so if you plan on making bug reports to them, RTFM.

The xine HOWTO (http://dvd.sourceforge.net/xine-howto/en_GB/html/howto.html) contains a chapter on performance improvement which is general to all players.

Finally, there are some other promising applications which the reader may try:

- Avifile (http://avifile.sourceforge.net/) which is also a port `multimedia/avifile`.
- Ogle (http://www.dtek.chalmers.se/groups/dvd/) which is also a port `multimedia/ogle`.
- Xtheater (http://xtheater.sourceforge.net/)
- `multimedia/dvdauthor`, an open source package for authoring DVD content.

7.5 設定電視卡(TV Cards)

Original contribution by Josef El-Rayes. Enhanced and adapted by Marc Fonvieille.

7.5.1 介紹

電視卡(TV card)可以讓您用電腦來看無線、有線電視節目。許多卡都是透過RCA 或S-video 輸入端子來接收視訊，而且有些卡還可接收FM 廣播的功能。

FreeBSD 可透過bktr(4) 驅動程式，來支援PCI 介面的電視卡，只要這些卡使用的是Brooktree Bt848/849/878/879 或Conexant CN-878/Fusion 878a 視訊擷取晶片。此外，要再確認哪些卡上所附的選台功能是否有支援，可以參考bktr(4) 說明，以查看所支援的硬體清單。

7.5.2 設定相關驅動程式

要用電視卡的話，就要載入bktr(4) 驅動程式，這個可以透過在`/boot/loader.conf` 檔加上下面這一行就可以了：

```
bktr_load="YES"
```

此外，也可以把該kernel module 直接與kernel 編譯在一起，作法就是在你的kernel 設定檔內，加上下面這幾行：

```
device  bktr
device iicbus
device iicbb
```

```
device smbus
```

之所以要加上這些額外的驅動程式，是因為卡的各組成部分都是透過I2C 匯流排而相互連接的。接下來，請重新編譯、安裝新的kernel 。

安裝好新的kernel 之後，要重開機才會生效。開機時，應該會看到類似下面的正確偵測到TV card 訊息：

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

當然，這些訊息可能因您的硬體不同而有所不同。However you should check if the tuner is correctly detected; it is still possible to override some of the detected parameters with sysctl(8) MIBs and kernel configuration file options. For example, if you want to force the tuner to a Philips SECAM tuner, you should add the following line to your kernel configuration file:

```
options OVERRIDE_TUNER=6
```

or you can directly use sysctl(8):

```
# sysctl hw.bt848.tuner=6
```

See the bktr(4) manual page and the `/usr/src/sys/conf/NOTES` file for more details on the available options. (If you are under FreeBSD 4.X, `/usr/src/sys/conf/NOTES` is replaced with `/usr/src/sys/i386/conf/LINT`.)

7.5.3 好用的程式

要用電視卡，可以視需要安裝下列應用程式之一：

- `multimedia/fxtv` provides TV-in-a-window and image/audio/video capture capabilities.
- `multimedia/xawtv` is also a TV application, with the same features as **fxtv**.
- `misc/alevt` decodes and displays Videotext/Teletext.
- `audio/xmradio`, an application to use the FM radio tuner coming with some TV cards.
- `audio/wmtune`, a handy desktop application for radio tuners.

More applications are available in the FreeBSD Ports Collection.

7.5.4 Troubleshooting

If you encounter any problem with your TV card, you should check at first if the video capture chip and the tuner are really supported by the bktr(4) driver and if you used the right configuration options. For more support and various questions about your TV card you may want to contact and use the archives of the freebsd-multimedia (http://lists.FreeBSD.org/mailman/listinfo/freebsd-multimedia) mailing list.

7.6 掃描器

Written by Marc Fonvieille.

7.6.1 介紹

FreeBSD 就像任何現代作業系統一樣，都可以使用掃描器。在FreeBSD 是透過Ports Collection 內的**SANE**(Scanner Access Now Easy) 所提供的API 來操作掃描器。**SANE** 也會使用一些FreeBSD 的驅動程式來控制掃描器硬體。

FreeBSD 同時支援SCSI 和USB 兩種介面的掃描器。在做任何設定之前，請確保**SANE** 有支援您的掃描器。**SANE** 有張支援硬體 (http://sane-project.org/sane-supported-devices.html) 的清單，這裡有介紹掃描器的支援情況和狀態訊息。在uscanner(4) 內也有提供一份USB 掃描器的支援列表。

7.6.2 Kernel 的設定

如同上述所提的SCSI 和USB 界面都有支援。這要取決於您的掃描器界面，而需要不同的設備驅動程式。

7.6.2.1 USB 介面

The `GENERIC` kernel by default includes the device drivers needed to support USB scanners. Should you decide to use a custom kernel, be sure that the following lines are present in your kernel configuration file:

```
device usb
device uhci
device ohci
device uscanner
```

Depending upon the USB chipset on your motherboard, you will only need either `device uhci` or `device ohci`, however having both in the kernel configuration file is harmless.

If you do not want to rebuild your kernel and your kernel is not the `GENERIC` one, you can directly load the uscanner(4) device driver module with the kldload(8) command:

```
# kldload uscanner
```

To load this module at each system startup, add the following line to `/boot/loader.conf`:

```
uscanner_load="YES"
```

After rebooting with the correct kernel, or after loading the required module, plug in your USB scanner. The scanner should appear in your system message buffer (dmesg(8)) as something like:

```
uscanner0: EPSON EPSON Scanner, rev 1.10/3.02, addr 2
```

This shows that our scanner is using the `/dev/uscanner0` device node.

> **Note:** On FreeBSD 4.X, the USB daemon (usbd(8)) must be running to be able to see some USB devices. To enable this, add `usbd_enable="YES"` to your `/etc/rc.conf` file and reboot the machine.

7.6.2.2 SCSI 介面

If your scanner comes with a SCSI interface, it is important to know which SCSI controller board you will use. According to the SCSI chipset used, you will have to tune your kernel configuration file. The `GENERIC` kernel supports the most common SCSI controllers. Be sure to read the `NOTES` file (`LINT` under FreeBSD 4.X) and add the correct line to your kernel configuration file. In addition to the SCSI adapter driver, you need to have the following lines in your kernel configuration file:

```
device scbus
device pass
```

Once your kernel has been properly compiled, you should be able to see the devices in your system message buffer, when booting:

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

If your scanner was not powered-on at system boot, it is still possible to manually force the detection by performing a SCSI bus scan with the camcontrol(8) command:

```
# camcontrol rescan all
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

Then the scanner will appear in the SCSI devices list:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0 (pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0 (pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0 (pass2,cd0)
```

More details about SCSI devices, are available in the scsi(4) and camcontrol(8) manual pages.

7.6.3 設定SANE

The **SANE** system has been splitted in two parts: the backends (`graphics/sane-backends`) and the frontends (`graphics/sane-frontends`). The backends part provides access to the scanner itself. The **SANE**'s supported devices (http://sane-project.org/sane-supported-devices.html) list specifies which backend will support your image scanner. It is mandatory to determine the correct backend for your scanner if you want to be able to use your device. The frontends part provides the graphical scanning interface (**xscanimage**).

The first thing to do is install the `graphics/sane-backends` port or package. Then, use the `sane-find-scanner` command to check the scanner detection by the **SANE** system:

```
# sane-find-scanner -q
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

The output will show the interface type of the scanner and the device node used to attach the scanner to the system. The vendor and the product model may not appear, it is not important.

> **Note:** Some USB scanners require you to load a firmware, this is explained in the backend manual page. You should also read sane-find-scanner(1) and sane(7) manual pages.

Now we have to check if the scanner will be identified by a scanning frontend. By default, the **SANE** backends comes with a command line tool called scanimage(1). This command allows you to list the devices and to perform an image acquisition from the command line. The `-L` option is used to list the scanner device:

```
# scanimage -L
device ‘snapscan:/dev/pass3’ is a AGFA SNAPSCAN 600 flatbed scanner
```

No output or a message saying that no scanners were identified indicates that scanimage(1) is unable to identify the scanner. If this happens, you will need to edit the backend configuration file and define the scanner device used. The `/usr/local/etc/sane.d/` directory contains all backends configuration files. This identification problem does appear with certain USB scanners.

For example, with the USB scanner used in the Section 7.6.2.1, `sane-find-scanner` gives us the following information:

```
# sane-find-scanner -q
found USB scanner (UNKNOWN vendor and product) at device /dev/uscanner0
```

The scanner is correctly detected, it uses the USB interface and is attached to the `/dev/uscanner0` device node. We can now check if the scanner is correctly identified:

```
# scanimage -L

No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
```

Since the scanner is not identified, we will need to edit the `/usr/local/etc/sane.d/epson.conf` file. The scanner model used was the EPSON Perfection® 1650, so we know the scanner will use the `epson` backend. Be sure to read the help comments in the backends configuration files. Line changes are quite simple: comment out all lines that have the wrong interface for your scanner (in our case, we will comment out all lines starting with the word `scsi` as our scanner uses the USB interface), then add at the end of the file a line specifying the interface and the device node used. In this case, we add the following line:

```
usb /dev/uscanner0
```

Please be sure to read the comments provided in the backend configuration file as well as the backend manual page for more details and correct syntax to use. We can now verify if the scanner is identified:

```
# scanimage -L
device ‘epson:/dev/uscanner0’ is a Epson GT-8200 flatbed scanner
```

Our USB scanner has been identified. It is not important if the brand and the model do not match. The key item to be concerned with is the `'epson:/dev/uscanner0'` field, which give us the right backend name and the right device node.

Once the `scanimage -L` command is able to see the scanner, the configuration is complete. The device is now ready to scan.

While scanimage(1) does allow us to perform an image acquisition from the command line, it is preferable to use a graphical user interface to perform image scanning. **SANE** offers a simple but efficient graphical interface: **xscanimage** (`graphics/sane-frontends`).

Xsane (`graphics/xsane`) is another popular graphical scanning frontend. This frontend offers advanced features such as various scanning mode (photocopy, fax, etc.), color correction, batch scans, etc. Both of these applications are useable as a **GIMP** plugin.

7.6.4 Allowing Scanner Access to Other Users

All previous operations have been done with `root` privileges. You may however, need other users to have access to the scanner. The user will need read and write permissions to the device node used by the scanner. As an example, our USB scanner uses the device node `/dev/uscanner0` which is owned by the `operator` group. Adding the user `joe` to the `operator` group will allow him to use the scanner:

```
# pw groupmod operator -m joe
```

For more details read the pw(8) manual page. You also have to set the correct write permissions (0660 or 0664) on the `/dev/uscanner0` device node, by default the `operator` group can only read the device node. This is done by adding the following lines to the `/etc/devfs.rules` file:

```
[system=5]
add path uscanner0 mode 660
```

Then add the following to `/etc/rc.conf` and reboot the machine:

```
devfs_system_ruleset="system"
```

More information regarding these lines can be found in the devfs(8) manual page. Under FreeBSD 4.X, the `operator` group has, by default, read and write permissions to `/dev/uscanner0`.

> **Note:** Of course, for security reasons, you should think twice before adding a user to any group, especially the `operator` group.

Chapter 8 設定FreeBSD Kernel

更新、重排：Jim Mock. 原作為：Jake Hamby.

8.1 概述

kernel 是整個FreeBSD 作業系統的核心。它控制了系統的整體運作，包含和記憶體管理、安全控管、網路、硬碟存取等等。儘管目前FreeBSD 大多可以用動態module 來載入、卸載所需功能，但有時候仍有必要學會重新調配kernel。

讀完這章，您將了解：

- 為何需要重新調配、編譯kernel？
- 要怎麼修改kernel 設定檔？
- 如何以kernel 設定檔來建立、編譯新的kernel 呢？
- 如何安裝新的kernel。
- 如何處理kernel 錯誤無法開機的情形。

本章所舉例的相關指令都是以`root` 權限來進行。

8.2 為何需要重新調配、編譯kernel？

早期的FreeBSD 的kernel 被戲稱為“monolithic” kernel。這意思是說當時的kernel 是個大塊頭程式，且只支援固定的硬體而已。如果您想改變kernel 的設定，那麼必須編譯一個新的並重新開機，才能啓用。

現在的FreeBSD 已快速成長到新型態的管理模式，其重要特色是：kernel 功能可以隨時依據需求，而以動態載入或卸載相關的kernel module。這使得kernel 能夠快速因應新的環境而作調整(有點像是：筆記型電腦上的PCMCIA 卡一樣即插即用)，或是增加其他原本的預設kernel(`GENERIC`)所沒有的功能。這種模式，就叫做modular kernel(核心模組)。

儘管如此，還是有一些功能仍須編譯在kernel 內才行。因為有時候是因為這些功能與kernel 結合的相當複雜緊密，而無法將它們弄成可動態載入的module ；而有時候，則是因為沒有人有空來弄那些kernel module 的實作。

重新調配、編譯kernel 幾乎是每位BSD 使用者所必須經歷的過程。儘管這項工作可能比較耗時，但在FreeBSD 的使用上會有許多好處。跟必須支援大多數各式硬體的`GENERIC` kernel 相比的話，自行調配kernel 不同處在於：可以更『體貼』，只支援『自己硬體』的部分就好。好處在於，譬如：

- 開機速度更快：因為自行調配的kernel 只需要偵測您系統上的硬體，所以讓啓動所花的過程更流暢快速。
- 佔用的記憶體更少：自行調配的kernel 通常會比`GENERIC` 核心使用更少的記憶體，由於kernel 必須一直存放在記憶體內，因此這就顯得更加重要。因此，對於記憶體較小的系統來說，自行調配的kernel 就可發揮更多的作用、揮灑空間。
- 可支援更多硬體：您可在自行調配的kernel 增加一些原本`GENERIC` 核心沒有提供的硬體支援，像是音效卡之類的。

8.3 探測系統硬體

Written by Tom Rhodes.

在進行kernel 設定的探索之旅前，先把該機器各項硬體資訊作點調查會是明智之舉。若FreeBSD 並非主要的作業系統，那麼也可以輕鬆透過目前所使用的作業系統，來查看相關硬體資訊表。舉例來說，Microsoft 的**裝置管理員(Device Manager)** 內通常會有目前有裝的硬體資訊。而**裝置管理員** 是在控制台。

Note: Microsoft Windows 某些版本則是先透過**系統(System)** 再進入**裝置管理員**。

若該機器尚未安裝任何作業系統，那麼就要親自找出相關硬體資訊。其中一種方式是透過dmesg(8) 以及man(1)。FreeBSD 上大多硬體都會有相關的man 說明有支援的規格型號，並且開機的偵測過程中，也會列出有找到的硬體。舉個例子，下面這幾行是說有偵測到滑鼠，並且是以`psm` 驅動程式：

```
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

驅動程式必須要在自訂的kernel 設定檔內加入，或者是用loader.conf(5)。

`dmesg` 有時只顯示系統訊息而沒有開機偵測的部份，遇到這種情況請查閱`/var/run/dmesg.boot` 檔。

另外也可以透過pciconf(8) 來列出更詳細的相關資訊。舉例說明：

```
ath0@pci0:3:0:0:        class=0x020000 card=0x058a1014 chip=0x1014168c rev=0x01 hdr=0x00
    vendor     = 'Atheros Communications Inc.'
    device     = 'AR5212 Atheros AR5212 802.11abg wireless'
    class      = network
    subclass   = ethernet
```

上面顯示是透過`pciconf -lv` 所看到的`ath` 無線網卡驅動程式。可以用`man` *`ath`* 來查看ath(4) 的相關說明。

在使用man(1) 時，加上`-k` 參數也可以提供比較精準的資訊。以上述例子而言，可以改為打：

```
# man -k Atheros
```

就會列出有含上述關鍵字的相關man 說明：

```
ath(4)                   - Atheros IEEE 802.11 wireless network driver
ath_hal(4)               - Atheros Hardware Access Layer (HAL)
```

知己知彼，先瞭解相關硬體環境，才能讓接下來的自訂kernel 打造過程更為順利。

8.4 重新調配、編譯kernel

首先對kernel 相關目錄作快速介紹。這裡所提到的所有目錄都在`/usr/src/sys` 內，也可以用`/sys` 這個symbolic link 來連到這。這裡的許多子目錄分別擺放kernel 的各組成部分，但對打造kernel 影響最重要的目錄是*`arch`*`/conf`，這裡是可以針對需求來修改自訂kernel 相關設定。此外，還有在編譯kernel 過程中會暫時擺放的`compile` 目錄。剛講到的*`arch`* 可以是右列架構之一：`i386`、`alpha`、`amd64`、`ia64`、`powerpc`、`sparc64`、`pc98`(在日本較流行的另一種PC 硬體架構)。在各

特定硬體架構目錄的東西，只搭配相對應的硬體架構而已。而其餘的原始碼則是與硬體架構無關，可以在所有FreeBSD 可裝的平台上共用。整體目錄架構都是有邏輯可循，像是各項有支援的硬體設備、檔案系統，以及相關選項通常都會擺在它們自己的子目錄內。

本章所用到的例子，都是你使用i386 架構的機器。請依實際情況，對相關目錄作調整即可。

> **Note:** 若您系統上*沒裝* `/usr/src/sys` 目錄，也就是說沒裝kernel source code 的話，那麼最簡單安裝方式就是以`root` 權限來執行`sysinstall`，接著請選Configure，然後選Distributions 接著為src 再選base 最後選sys。若不喜歡用**sysinstall** 而且手邊有“正式的” FreeBSD 光碟可以用的話，那麼也可以用以下指令來安裝：
>
> ```
> # mount /cdrom
> # mkdir -p /usr/src/sys
> # ln -s /usr/src/sys /sys
> # cat /cdrom/src/ssys.[a-d]* | tar -xzvf -
> # cat /cdrom/src/sbase.[a-d]* | tar -xzvf -
> ```

接下來，切換到*`arch`*`/conf` 目錄，複製`GENERIC` 設定檔為你想稱呼的新kernel 名稱。例如：

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MYKERNEL
```

通常，命名方式都是大寫。如果你負責維護許多不同硬體架構的FreeBSD 機器的話，那麼照該機器名稱(hostname)來命名會是比較明智。上面例子中之所以命名為`MYKERNEL` 就是因為這緣故。

> **Tip:** 建議不要把改過的kernel 設定檔直接放在`/usr/src`。因為若編譯遇到其他問題時，直接砍掉`/usr/src` 再重練，可能會是比較乾脆的選擇之一。一旦真的砍了之後，你可能幾秒之後才會醒悟到：你同時也砍掉自己改的kernel 設定檔。此外，也不要直接修改`GENERIC`，因為下次你更新source tree時，它會被新版覆蓋，而相關修改也將隨之而逝。
>
> 你也可考慮把kernel 設定檔改放到其他地方，然後再到*`i386`* 目錄內建個指向它的symbolic link。
>
> 舉例：
>
> ```
> # cd /usr/src/sys/i386/conf
> # mkdir /root/kernels
> # cp GENERIC /root/kernels/MYKERNEL
> # ln -s /root/kernels/MYKERNEL
> ```

現在，就開始用自己喜歡的編輯器來修改`MYKERNEL`。若才剛裝好FreeBSD 而已，唯一可用的編輯器很可能是**vi** 了，由於它的用法很多種，礙於篇幅將不詳細介紹，你可在參考書目 內找到相關書籍。不過，FreeBSD 也提供另一個更好用的編輯器，它叫做**ee**，對新手而言，這可能是蠻好的選擇。你可以任意修改檔案內的相關註解以說明相關設定為何，或者其他想改的`GENERIC` 設定內容。

若你有在SunOS 或者其他種BSD 作業系統下進行編譯kernel 的經驗，那麼應該已經很熟悉本篇所介紹的大部分步驟。換句話說，若您之前用的是DOS 這類作業系統，那麼`GENERIC` 設定檔的內容就可能比較難懂些，沒關係，我們將在下面的kernel 設定 會循序漸進地介紹。

> **Note:** 若有從FreeBSD 計劃去更新你的source tree 的話，則切記在進行任何升級之前，務必要察看`/usr/src/UPDATING`。這檔會介紹在更新過程中的重大議題或要注意的事項。由於`/usr/src/UPDATING` 是對應於你機器上目前的FreeBSD source code 版本，因此會提供比本手冊更新的內容。

現在開始來編譯kernel 吧。

編譯Kernel

1. 請切換至/usr/src 目錄：

   ```
   # cd /usr/src
   ```

2. 編譯kernel：

   ```
   # make buildkernel KERNCONF=MYKERNEL
   ```

3. 安裝新kernel：

   ```
   # make installkernel KERNCONF=MYKERNEL
   ```

Note: 要有完整的FreeBSD source tree 才能編譯kernel。

Tip: 預設情況下，在編譯自訂kernel 時，*全部的* kernel modules 也會一起重編。若要快速升級kernel，或是只想重編所需的kernel module，那麼在編譯kernel 前要先改一下/etc/make.conf，比如：

```
MODULES_OVERRIDE = linux acpi sound/sound sound/driver/ds1 ntfs
```

上面該設定值為所希望重編的kernel module 列表。

```
WITHOUT_MODULES = linux acpi sound/sound sound/driver/ds1 ntfs
```

而上面這設定值則為不要編入的kernel module 列表。若想更瞭解其他kernel 編譯的相關變數，請參閱make.conf(5) 說明。

新的kernel 會複製到/boot/kernel 目錄內的/boot/kernel/kernel，而舊的則移至/boot/kernel.old/kernel。現在呢，先關機，然後就會以新kernel 重開機若有問題的話，本章後面會介紹一些疑難雜症來協助你。若新kernel 無法開機的話，請參閱這裡 以恢復系統運作。

Note: 至於開機過程的其他相關檔案、設定，比如loader(8) 及其設定，則放在/boot。Third party 或自訂的kernel modules 則會放在/boot/kernel，不過，應注意要保持kernel module 與kernel 是否有同步，這點很重要，否則會導致不穩或出問題。

8.5 kernel 設定檔解說

Updated for FreeBSD 6.X by Joel Dahl.

kernel 設定檔的內容格式相當簡單。每一行都包括一個關鍵字，以及一個或多個參數。事實上，很多行大多只有一個參數。任何以# 開頭的敘述都將被視為註解而被忽略。接下來將以在GENERIC 所出現的順序一一介紹之。若要看與該平台架構有關的各選項、設備列表，請參閱與GENERIC 檔同目錄的NOTES 檔。而與平台架構差異較無關的通用部份，則可參閱/usr/src/sys/conf/NOTES。

Note: 若為了測試，而需要一份含有所有可用設定的設定檔，那麼請以root 身份下：

```
# cd /usr/src/sys/i386/conf && make LINT
```

下面為GENERIC 設定檔的範例，其中包括說明用的註釋。這例子應該與您機器上的/usr/src/sys/*i386*/conf/GENERIC 相當接近。

```
machine  i386
```

此處是指機器架構，必須為alpha、amd64、i386、ia64、pc98、powerpc、sparc64 其中之一。

```
cpu          I486_CPU
cpu          I586_CPU
cpu          I686_CPU
```

上面設定是指定要用哪一種CPU 型號。也可以同時加上多組CPU 型號(比如說萬一不確定是否要用I586_CPU 或I686_CPU)。然而自訂kernel 的話，建議先確認自己的CPU 型號，然後只用最適合的那組就好了。若不確定CPU 到底是用哪一種，可以查閱/var/run/dmesg.boot 的開機訊息以確定。

```
ident          GENERIC
```

這是設定該kernel 名稱為何，可以隨意命名之，像是取名為MYKERNEL，若是有照先前說明來作大概會取這樣名字。ident 後面的字串會在開機時顯示，因此若要辨認新kernel 與常用kernel 的話，就設定不同組名稱即可(比如在自訂實驗用的kernel)。

```
#To statically compile in device wiring instead of /boot/device.hints
#hints          "GENERIC.hints"         # Default places to look for devices.
```

device.hints(5) 可用來設定各項驅動程式的選項。開機時loader(8) 會檢查預設的/boot/device.hints 設定檔。使用hints 選項，就可以把這些hints 靜態編入kernel 內。如此一來就不必在/boot 內建立device.hints 檔。

```
makeoptions     DEBUG=-g          # Build kernel with gdb(1) debug symbols
```

加上-g 選項的話，FreeBSD 會在編譯過程加上debug 用的資訊，透過這選項會讓gcc(1) 啓用debug 所會用到的相關資訊。

```
options          SCHED_4BSD         # 4BSD scheduler
```

FreeBSD. 傳統所用(並且是預設)的系統CPU scheduler。若您不清楚要如何設定，請保留這設定。

```
options          PREEMPTION         # Enable kernel thread preemption
```

Allows threads that are in the kernel to be preempted by higher priority threads. It helps with interactivity and allows interrupt threads to run sooner rather than waiting.

```
options          INET              # InterNETworking
```

Networking support. Leave this in, even if you do not plan to be connected to a network. Most programs require at least loopback networking (i.e., making network connections within your PC), so this is essentially mandatory.

```
options          INET6             # IPv6 communications protocols
```

This enables the IPv6 communication protocols.

```
options         FFS             # Berkeley Fast Filesystem
```

This is the basic hard drive file system. Leave it in if you boot from the hard disk.

```
options         SOFTUPDATES     # Enable FFS Soft Updates support
```

This option enables Soft Updates in the kernel, this will help speed up write access on the disks. Even when this functionality is provided by the kernel, it must be turned on for specific disks. Review the output from mount(8) to see if Soft Updates is enabled for your system disks. If you do not see the `soft-updates` option then you will need to activate it using the tunefs(8) (for existing file systems) or newfs(8) (for new file systems) commands.

```
options         UFS_ACL         # Support for access control lists
```

This option enables kernel support for access control lists. This relies on the use of extended attributes and UFS2, and the feature is described in detail in Section 14.12. ACLs are enabled by default and should not be disabled in the kernel if they have been used previously on a file system, as this will remove the access control lists, changing the way files are protected in unpredictable ways.

```
options         UFS_DIRHASH     # Improve performance on big directories
```

This option includes functionality to speed up disk operations on large directories, at the expense of using additional memory. You would normally keep this for a large server, or interactive workstation, and remove it if you are using FreeBSD on a smaller system where memory is at a premium and disk access speed is less important, such as a firewall.

```
options         MD_ROOT         # MD is a potential root device
```

This option enables support for a memory backed virtual disk used as a root device.

```
options         NFSCLIENT       # Network Filesystem Client
options         NFSSERVER       # Network Filesystem Server
options         NFS_ROOT        # NFS usable as /, requires NFSCLIENT
```

The network file system. Unless you plan to mount partitions from a UNIX file server over TCP/IP, you can comment these out.

```
options         MSDOSFS         # MSDOS Filesystem
```

The MS-DOS file system. Unless you plan to mount a DOS formatted hard drive partition at boot time, you can safely comment this out. It will be automatically loaded the first time you mount a DOS partition, as described above. Also, the excellent `emulators/mtools` software allows you to access DOS floppies without having to mount and unmount them (and does not require `MSDOSFS` at all).

```
options         CD9660          # ISO 9660 Filesystem
```

The ISO 9660 file system for CDROMs. Comment it out if you do not have a CDROM drive or only mount data CDs occasionally (since it will be dynamically loaded the first time you mount a data CD). Audio CDs do not need this file system.

```
options         PROCFS          # Process filesystem(requires PSEUDOFS)
```

The process file system. This is a “pretend” file system mounted on `/proc` which allows programs like ps(1) to give you more information on what processes are running. Use of `PROCFS` is not required under most circumstances, as most debugging and monitoring tools have been adapted to run without `PROCFS`: installs will not mount this file system by default.

```
options         PSEUDOFS        # Pseudo-filesystem framework
```

6.X kernels making use of `PROCFS` must also include support for `PSEUDOFS`.

```
options         GEOM_GPT        # GUID Partition Tables.
```

This option brings the ability to have a large number of partitions on a single disk.

```
options         COMPAT_43       # Compatible with BSD 4.3 [KEEP THIS!]
```

Compatibility with 4.3BSD. Leave this in; some programs will act strangely if you comment this out.

```
options         COMPAT_FREEBSD4 # Compatible with FreeBSD4
```

This option is required on FreeBSD 5.X i386 and Alpha systems to support applications compiled on older versions of FreeBSD that use older system call interfaces. It is recommended that this option be used on all i386 and Alpha systems that may run older applications; platforms that gained support only in 5.X, such as ia64 and Sparc64, do not require this option.

```
options         COMPAT_FREEBSD5 # 與 FreeBSD5 相容
```

此行是FreeBSD 6.X 及更新的版本若需支援FreeBSD 5.X 系統呼叫才需要設定。

```
options         SCSI_DELAY=5000  # Delay (in ms) before probing SCSI
```

This causes the kernel to pause for 5 seconds before probing each SCSI device in your system. If you only have IDE hard drives, you can ignore this, otherwise you can try to lower this number, to speed up booting. Of course, if you do this and FreeBSD has trouble recognizing your SCSI devices, you will have to raise it again.

```
options         KTRACE          # ktrace(1) support
```

This enables kernel process tracing, which is useful in debugging.

```
options         SYSVSHM         # SYSV-style shared memory
```

This option provides for System V shared memory. The most common use of this is the XSHM extension in X, which many graphics-intensive programs will automatically take advantage of for extra speed. If you use X, you will definitely want to include this.

```
options         SYSVMSG         # SYSV-style message queues
```

Support for System V messages. This option only adds a few hundred bytes to the kernel.

```
options         SYSVSEM         # SYSV-style semaphores
```

Support for System V semaphores. Less commonly used but only adds a few hundred bytes to the kernel.

> **Note:** The `-p` option of the ipcs(1) command will list any processes using each of these System V facilities.

```
options       _KPOSIX_PRIORITY_SCHEDULING # POSIX P1003_1B real-time extensions
```

Real-time extensions added in the 1993 POSIX®. Certain applications in the Ports Collection use these (such as **StarOffice**).

```
options          KBD_INSTALL_CDEV  # install a CDEV entry in /dev
```

This option is required to allow the creation of keyboard device nodes in `/dev`.

```
options          ADAPTIVE_GIANT    # Giant mutex is adaptive.
```

Giant is the name of a mutual exclusion mechanism (a sleep mutex) that protects a large set of kernel resources. Today, this is an unacceptable performance bottleneck which is actively being replaced with locks that protect individual resources. The `ADAPTIVE_GIANT` option causes Giant to be included in the set of mutexes adaptively spun on. That is, when a thread wants to lock the Giant mutex, but it is already locked by a thread on another CPU, the first thread will keep running and wait for the lock to be released. Normally, the thread would instead go back to sleep and wait for its next chance to run. If you are not sure, leave this in.

> **Note:** Note that on FreeBSD 8.0-CURRENT and later versions, all mutexes are adaptive by default, unless explicitly set to non-adaptive by compiling with the `NO_ADAPTIVE_MUTEXES` option. As a result, Giant is adaptive by default now, and the `ADAPTIVE_GIANT` option has been removed from the kernel configuration.

```
device          apic               # I/O APIC
```

The apic device enables the use of the I/O APIC for interrupt delivery. The apic device can be used in both UP and SMP kernels, but is required for SMP kernels. Add `options SMP` to include support for multiple processors.

> **Note:** apic 只限i386 架構才有，其他架構則不必加上這行。

```
device          eisa
```

Include this if you have an EISA motherboard. This enables auto-detection and configuration support for all devices on the EISA bus.

```
device          pci
```

Include this if you have a PCI motherboard. This enables auto-detection of PCI cards and gatewaying from the PCI to ISA bus.

```
# Floppy drives
device          fdc
```

This is the floppy drive controller.

```
# ATA and ATAPI devices
device          ata
```

This driver supports all ATA and ATAPI devices. You only need one `device ata` line for the kernel to detect all PCI ATA/ATAPI devices on modern machines.

```
device          atadisk                 # ATA disk drives
```

This is needed along with `device ata` for ATA disk drives.

```
device          ataraid                 # ATA RAID drives
```

This is needed along with `device ata` for ATA RAID drives.

```
device          atapicd                 # ATAPI CDROM drives
```

This is needed along with `device ata` for ATAPI CDROM drives.

```
device          atapifd                 # ATAPI floppy drives
```

This is needed along with `device ata` for ATAPI floppy drives.

```
device          atapist                 # ATAPI tape drives
```

This is needed along with `device ata` for ATAPI tape drives.

```
options         ATA_STATIC_ID           # Static device numbering
```

This makes the controller number static; without this, the device numbers are dynamically allocated.

```
# SCSI Controllers
device          ahb        # EISA AHA1742 family
device          ahc        # AHA2940 and onboard AIC7xxx devices
options         AHC_REG_PRETTY_PRINT    # Print register bitfields in debug
                                        # output.  Adds ~128k to driver.
device          ahd        # AHA39320/29320 and onboard AIC79xx devices
options         AHD_REG_PRETTY_PRINT    # Print register bitfields in debug
            # output.  Adds ~215k to driver.
device          amd        # AMD 53C974 (Teckram DC-390(T))
device          isp        # Qlogic family
device          ispfw      # Firmware for QLogic HBAs- normally a module
device          mpt        # LSI-Logic MPT-Fusion
#device         ncr        # NCR/Symbios Logic
device          sym        # NCR/Symbios Logic (newer chipsets + those of 'ncr')
device          trm        # Tekram DC395U/UW/F DC315U adapters

device          adv        # Advansys SCSI adapters
device          adw        # Advansys wide SCSI adapters
device          aha        # Adaptec 154x SCSI adapters
device          aic        # Adaptec 15[012]x SCSI adapters, AIC-6[23]60.
device          bt         # Buslogic/Mylex MultiMaster SCSI adapters

device          ncv        # NCR 53C500
device          nsp        # Workbit Ninja SCSI-3
device          stg        # TMC 18C30/18C50
```

SCSI controllers. Comment out any you do not have in your system. If you have an IDE only system, you can remove these altogether. The `*_REG_PRETTY_PRINT` lines are debugging options for their respective drivers.

```
# SCSI peripherals
device          scbus      # SCSI bus (required for SCSI)
device          ch         # SCSI media changers
device          da         # Direct Access (disks)
device          sa         # Sequential Access (tape etc)
device          cd         # CD
device          pass       # Passthrough device (direct SCSI access)
device          ses        # SCSI Environmental Services (and SAF-TE)
```

SCSI peripherals. Again, comment out any you do not have, or if you have only IDE hardware, you can remove them completely.

> **Note:** The USB umass(4) driver and a few other drivers use the SCSI subsystem even though they are not real SCSI devices. Therefore make sure not to remove SCSI support, if any such drivers are included in the kernel configuration.

```
# RAID controllers interfaced to the SCSI subsystem
device          amr        # AMI MegaRAID
device          arcmsr     # Areca SATA II RAID
device          asr        # DPT SmartRAID V, VI and Adaptec SCSI RAID
device          ciss       # Compaq Smart RAID 5*
device          dpt        # DPT Smartcache III, IV - See NOTES for options
device          hptmv      # Highpoint RocketRAID 182x
device          rr232x     # Highpoint RocketRAID 232x
device          iir        # Intel Integrated RAID
device          ips        # IBM (Adaptec) ServeRAID
device          mly        # Mylex AcceleRAID/eXtremeRAID
device          twa        # 3ware 9000 series PATA/SATA RAID

# RAID controllers
device          aac        # Adaptec FSA RAID
device          aacp       # SCSI passthrough for aac (requires CAM)
device          ida        # Compaq Smart RAID
device          mfi        # LSI MegaRAID SAS
device          mlx        # Mylex DAC960 family
device          pst        # Promise Supertrak SX6000
device          twe        # 3ware ATA RAID
```

Supported RAID controllers. If you do not have any of these, you can comment them out or remove them.

```
# atkbdc0 controls both the keyboard and the PS/2 mouse
device          atkbdc     # AT keyboard controller
```

The keyboard controller (`atkbdc`) provides I/O services for the AT keyboard and PS/2 style pointing devices. This controller is required by the keyboard driver (`atkbd`) and the PS/2 pointing device driver (`psm`).

```
device          atkbd      # AT keyboard
```

The atkbd driver, together with atkbdc controller, provides access to the AT 84 keyboard or the AT enhanced keyboard which is connected to the AT keyboard controller.

```
device          psm        # PS/2 mouse
```

Use this device if your mouse plugs into the PS/2 mouse port.

```
device          kbdmux        # keyboard multiplexer
```

多重鍵盤的支援。若不打算同時接多組鍵盤的話，那麼若要移除該行也沒關係。

```
device          vga        # VGA video card driver
```

The video card driver.

```
device          splash     # Splash screen and screen saver support
```

Splash screen at start up! Screen savers require this too.

```
# syscons is the default console driver, resembling an SCO console
device          sc
```

sc is the default console driver and resembles a SCO console. Since most full-screen programs access the console through a terminal database library like termcap, it should not matter whether you use this or vt, the VT220 compatible console driver. When you log in, set your TERM variable to scoansi if full-screen programs have trouble running under this console.

```
# Enable this for the pcvt (VT220 compatible) console driver
#device          vt
#options         XSERVER          # support for X server on a vt console
#options         FAT_CURSOR       # start with block cursor
```

This is a VT220-compatible console driver, backward compatible to VT100/102. It works well on some laptops which have hardware incompatibilities with sc. Also set your TERM variable to vt100 or vt220 when you log in. This driver might also prove useful when connecting to a large number of different machines over the network, where termcap or terminfo entries for the sc device are often not available ——vt100 should be available on virtually any platform.

```
device          agp
```

Include this if you have an AGP card in the system. This will enable support for AGP, and AGP GART for boards which have these features.

```
# Power management support (see NOTES for more options)
#device          apm
```

Advanced Power Management support. Useful for laptops, although in FreeBSD 5.X and above this is disabled in GENERIC by default.

```
# Add suspend/resume support for the i8254.
device          pmtimer
```

Timer device driver for power management events, such as APM and ACPI.

```
# PCCARD (PCMCIA) support
# PCMCIA and cardbus bridge support
device          cbb             # cardbus (yenta) bridge
device          pccard          # PC Card (16-bit) bus
device          cardbus         # CardBus (32-bit) bus
```

PCMCIA support. You want this if you are using a laptop.

```
# Serial (COM) ports
device          sio             # 8250, 16[45]50 based serial ports
```

These are the serial ports referred to as `COM` ports in the MS-DOS/Windows world.

> **Note:** If you have an internal modem on `COM4` and a serial port at `COM2`, you will have to change the IRQ of the modem to 2 (for obscure technical reasons, IRQ2 = IRQ 9) in order to access it from FreeBSD. If you have a multiport serial card, check the manual page for sio(4) for more information on the proper values to add to your `/boot/device.hints`. Some video cards (notably those based on S3 chips) use IO addresses in the form of `0x*2e8`, and since many cheap serial cards do not fully decode the 16-bit IO address space, they clash with these cards making the `COM4` port practically unavailable.
>
> Each serial port is required to have a unique IRQ (unless you are using one of the multiport cards where shared interrupts are supported), so the default IRQs for `COM3` and `COM4` cannot be used.

```
# Parallel port
device          ppc
```

This is the ISA-bus parallel port interface.

```
device          ppbus       # Parallel port bus (required)
```

Provides support for the parallel port bus.

```
device          lpt         # Printer
```

Support for parallel port printers.

> **Note:** All three of the above are required to enable parallel printer support.

```
device          plip        # TCP/IP over parallel
```

This is the driver for the parallel network interface.

```
device          ppi         # Parallel port interface device
```

The general-purpose I/O (“geek port”) + IEEE1284 I/O.

```
#device         vpo         # Requires scbus and da
```

This is for an Iomega Zip drive. It requires `scbus` and `da` support. Best performance is achieved with ports in EPP 1.9 mode.

```
#device         puc
```

Uncomment this device if you have a “dumb” serial or parallel PCI card that is supported by the puc(4) glue driver.

```
# PCI Ethernet NICs.
device          de          # DEC/Intel DC21x4x (“Tulip”)
device          em          # Intel PRO/1000 adapter Gigabit Ethernet Card
device          ixgb        # Intel PRO/10GbE Ethernet Card
device          txp         # 3Com 3cR990 (“Typhoon”)
device          vx          # 3Com 3c590, 3c595 (“Vortex”)
```

Various PCI network card drivers. Comment out or remove any of these not present in your system.

```
# PCI Ethernet NICs that use the common MII bus controller code.
# NOTE: Be sure to keep the 'device miibus' line in order to use these NICs!
device          miibus      # MII bus support
```

MII bus support is required for some PCI 10/100 Ethernet NICs, namely those which use MII-compliant transceivers or implement transceiver control interfaces that operate like an MII. Adding `device miibus` to the kernel config pulls in support for the generic miibus API and all of the PHY drivers, including a generic one for PHYs that are not specifically handled by an individual driver.

```
device          bce         # Broadcom BCM5706/BCM5708 Gigabit Ethernet
device          bfe         # Broadcom BCM440x 10/100 Ethernet
device          bge         # Broadcom BCM570xx Gigabit Ethernet
device          dc          # DEC/Intel 21143 and various workalikes
device          fxp         # Intel EtherExpress PRO/100B (82557, 82558)
device          lge         # Level 1 LXT1001 gigabit ethernet
device          msk         # Marvell/SysKonnect Yukon II Gigabit Ethernet
device          nge         # NatSemi DP83820 gigabit ethernet
device          nve         # nVidia nForce MCP on-board Ethernet Networking
device          pcn         # AMD Am79C97x PCI 10/100 (precedence over 'lnc')
device          re          # RealTek 8139C+/8169/8169S/8110S
device          rl          # RealTek 8129/8139
device          sf          # Adaptec AIC-6915 (“Starfire”)
device          sis         # Silicon Integrated Systems SiS 900/SiS 7016
device          sk          # SysKonnect SK-984x & SK-982x gigabit Ethernet
device          ste         # Sundance ST201 (D-Link DFE-550TX)
device          stge        # Sundance/Tamarack TC9021 gigabit Ethernet
device          ti          # Alteon Networks Tigon I/II gigabit Ethernet
device          tl          # Texas Instruments ThunderLAN
device          tx          # SMC EtherPower II (83c170 “EPIC”)
device          vge         # VIA VT612x gigabit ethernet
device          vr          # VIA Rhine, Rhine II
device          wb          # Winbond W89C840F
device          xl          # 3Com 3c90x (“Boomerang”, “Cyclone”)
```

Drivers that use the MII bus controller code.

```
# ISA Ethernet NICs.  pccard NICs included.
device          cs          # Crystal Semiconductor CS89x0 NIC
# 'device ed' requires 'device miibus'
device          ed          # NE[12]000, SMC Ultra, 3c503, DS8390 cards
```

```
device          ex         # Intel EtherExpress Pro/10 and Pro/10+
device          ep         # Etherlink III based cards
device          fe         # Fujitsu MB8696x based cards
device          ie         # EtherExpress 8/16, 3C507, StarLAN 10 etc.
device          lnc        # NE2100, NE32-VL Lance Ethernet cards
device          sn         # SMC's 9000 series of Ethernet chips
device          xe         # Xircom pccard Ethernet

# ISA devices that use the old ISA shims
#device         le
```

ISA Ethernet drivers. See `/usr/src/sys/`*`i386`*`/conf/NOTES` for details of which cards are supported by which driver.

```
# Wireless NIC cards
device          wlan       # 802.11 support
```

對802.11 標準的支援。若要無線上網，則需加上這行。

```
device          wlan_wep        # 802.11 WEP support
device          wlan_ccmp       # 802.11 CCMP support
device          wlan_tkip       # 802.11 TKIP support
```

對802.11 加密設備的支援。若要安全加密以及802.11i 安全協定，則需加上這行。

```
device          an         # Aironet 4500/4800 802.11 wireless NICs.
device          ath             # Atheros pci/cardbus NIC's
device          ath_hal         # Atheros HAL (Hardware Access Layer)
device          ath_rate_sample # SampleRate tx rate control for ath
device          an         # Aironet 4500/4800 802.11 wireless NICs.
device          awi        # BayStack 660 and others
device          ral        # Ralink Technology RT2500 wireless NICs.
device          wi         # WaveLAN/Intersil/Symbol 802.11 wireless NICs.
#device         wl         # Older non 802.11 Wavelan wireless NIC.
```

Support for various wireless cards.

```
# Pseudo devices
device   loop          # Network loopback
```

This is the generic loopback device for TCP/IP. If you telnet or FTP to `localhost` (a.k.a. `127.0.0.1`) it will come back at you through this device. This is *mandatory*.

```
device   random        # Entropy device
```

Cryptographically secure random number generator.

```
device   ether         # Ethernet support
```

`ether` is only needed if you have an Ethernet card. It includes generic Ethernet protocol code.

```
device   sl            # Kernel SLIP
```

`sl` is for SLIP support. This has been almost entirely supplanted by PPP, which is easier to set up, better suited for modem-to-modem connection, and more powerful.

```
device   ppp           # Kernel PPP
```

This is for kernel PPP support for dial-up connections. There is also a version of PPP implemented as a userland application that uses `tun` and offers more flexibility and features such as demand dialing.

```
device   tun           # Packet tunnel.
```

This is used by the userland PPP software. See the **PPP** section of this book for more information.

```
device   pty           # Pseudo-ttys (telnet etc)
```

This is a "pseudo-terminal" or simulated login port. It is used by incoming `telnet` and `rlogin` sessions, **xterm**, and some other applications such as **Emacs**.

```
device   md            # Memory "disks"
```

Memory disk pseudo-devices.

```
device   gif           # IPv6 and IPv4 tunneling
```

This implements IPv6 over IPv4 tunneling, IPv4 over IPv6 tunneling, IPv4 over IPv4 tunneling, and IPv6 over IPv6 tunneling. The `gif` device is "auto-cloning", and will create device nodes as needed.

```
device   faith         # IPv6-to-IPv4 relaying (translation)
```

This pseudo-device captures packets that are sent to it and diverts them to the IPv4/IPv6 translation daemon.

```
# The 'bpf' device enables the Berkeley Packet Filter.
# Be aware of the administrative consequences of enabling this!
# Note that 'bpf' is required for DHCP.
device   bpf           # Berkeley packet filter
```

This is the Berkeley Packet Filter. This pseudo-device allows network interfaces to be placed in promiscuous mode, capturing every packet on a broadcast network (e.g., an Ethernet). These packets can be captured to disk and or examined with the tcpdump(1) program.

> **Note:** The bpf(4) device is also used by dhclient(8) to obtain the IP address of the default router (gateway) and so on. If you use DHCP, leave this uncommented.

```
# USB support
device          uhci          # UHCI PCI->USB interface
device          ohci          # OHCI PCI->USB interface
device          ehci          # EHCI PCI->USB interface (USB 2.0)
device          usb           # USB Bus (required)
#device         udbp          # USB Double Bulk Pipe devices
device          ugen          # Generic
device          uhid          #  "Human Interface Devices"
device          ukbd          # Keyboard
device          ulpt          # Printer
```

```
device          umass         # Disks/Mass storage - Requires scbus and da
device          ums           # Mouse
device          ural          # Ralink Technology RT2500USB wireless NICs
device          urio          # Diamond Rio 500 MP3 player
device          uscanner      # Scanners
# USB Ethernet, requires mii
device          aue           # ADMtek USB Ethernet
device          axe           # ASIX Electronics USB Ethernet
device          cdce          # Generic USB over Ethernet
device          cue           # CATC USB Ethernet
device          kue           # Kawasaki LSI USB Ethernet
device          rue           # RealTek RTL8150 USB Ethernet
```

Support for various USB devices.

```
# FireWire support
device          firewire      # FireWire bus code
device          sbp           # SCSI over FireWire (Requires scbus and da)
device          fwe           # Ethernet over FireWire (non-standard!)
```

Support for various Firewire devices.

For more information and additional devices supported by FreeBSD, see `/usr/src/sys/i386/conf/NOTES`.

8.5.1 Large Memory Configurations (PAE)

Large memory configuration machines require access to more than the 4 gigabyte limit on User+Kernel Virtual Address (KVA) space. Due to this limitation, Intel added support for 36-bit physical address space access in the Pentium Pro and later line of CPUs.

The Physical Address Extension (PAE) capability of the Intel Pentium Pro and later CPUs allows memory configurations of up to 64 gigabytes. FreeBSD provides support for this capability via the `PAE` kernel configuration option, available in all current release versions of FreeBSD. Due to the limitations of the Intel memory architecture, no distinction is made for memory above or below 4 gigabytes. Memory allocated above 4 gigabytes is simply added to the pool of available memory.

To enable PAE support in the kernel, simply add the following line to your kernel configuration file:

```
options      PAE
```

> **Note:** The PAE support in FreeBSD is only available for Intel IA-32 processors. It should also be noted, that the PAE support in FreeBSD has not received wide testing, and should be considered beta quality compared to other stable features of FreeBSD.

PAE support in FreeBSD has a few limitations:

- A process is not able to access more than 4 gigabytes of VM space.
- KLD modules cannot be loaded into a PAE enabled kernel, due to the differences in the build framework of a module and the kernel.

- Device drivers that do not use the bus_dma(9) interface will cause data corruption in a PAE enabled kernel and are not recommended for use. For this reason, a `PAE` kernel configuration file is provided in FreeBSD which excludes all drivers not known to work in a PAE enabled kernel.
- Some system tunables determine memory resource usage by the amount of available physical memory. Such tunables can unnecessarily over-allocate due to the large memory nature of a PAE system. One such example is the `kern.maxvnodes` sysctl, which controls the maximum number of vnodes allowed in the kernel. It is advised to adjust this and other such tunables to a reasonable value.
- It might be necessary to increase the kernel virtual address (KVA) space or to reduce the amount of specific kernel resource that is heavily used (see above) in order to avoid KVA exhaustion. The `KVA_PAGES` kernel option can be used for increasing the KVA space.

For performance and stability concerns, it is advised to consult the tuning(7) manual page. The pae(4) manual page contains up-to-date information on FreeBSD's PAE support.

8.6 If Something Goes Wrong

There are five categories of trouble that can occur when building a custom kernel. They are:

`config` fails:

If the config(8) command fails when you give it your kernel description, you have probably made a simple error somewhere. Fortunately, config(8) will print the line number that it had trouble with, so that you can quickly locate the line containing the error. For example, if you see:

```
config: line 17: syntax error
```

Make sure the keyword is typed correctly by comparing it to the `GENERIC` kernel or another reference.

`make` fails:

If the `make` command fails, it usually signals an error in your kernel description which is not severe enough for config(8) to catch. Again, look over your configuration, and if you still cannot resolve the problem, send mail to the FreeBSD general questions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) with your kernel configuration, and it should be diagnosed quickly.

The kernel does not boot:

If your new kernel does not boot, or fails to recognize your devices, do not panic! Fortunately, FreeBSD has an excellent mechanism for recovering from incompatible kernels. Simply choose the kernel you want to boot from at the FreeBSD boot loader. You can access this when the system boot menu appears. Select the "Escape to a loader prompt" option, number six. At the prompt, type `unload kernel` and then type `boot /boot/`*`kernel.old`*`/kernel`, or the filename of any other kernel that will boot properly. When reconfiguring a kernel, it is always a good idea to keep a kernel that is known to work on hand.

After booting with a good kernel you can check over your configuration file and try to build it again. One helpful resource is the `/var/log/messages` file which records, among other things, all of the kernel messages from every successful boot. Also, the dmesg(8) command will print the kernel messages from the current boot.

> **Note:** If you are having trouble building a kernel, make sure to keep a `GENERIC`, or some other kernel that is known to work on hand as a different name that will not get erased on the next build. You cannot rely on

`kernel.old` because when installing a new kernel, `kernel.old` is overwritten with the last installed kernel which may be non-functional. Also, as soon as possible, move the working kernel to the proper `/boot/kernel` location or commands such as ps(1) may not work properly. To do this, simply rename the directory containing the good kernel:

```
# mv /boot/kernel /boot/kernel.bad
# mv /boot/kernel.good /boot/kernel
```

The kernel works, but ps(1) does not work any more:

If you have installed a different version of the kernel from the one that the system utilities have been built with, for example, a -CURRENT kernel on a -RELEASE, many system-status commands like ps(1) and vmstat(8) will not work any more. You should recompile and install a world built with the same version of the source tree as your kernel. This is one reason it is not normally a good idea to use a different version of the kernel from the rest of the operating system.

Chapter 9 列印

Contributed by Sean Kelly. Restructured and updated by Jim Mock.

9.1 概述

FreeBSD 可以和各式各樣的印表機搭配列印，從最老的撞針式印表機到最新的雷射印表機都沒問題，讓您的應用程式可以產生出高品質的文件列印輸出。

也可以把FreeBSD 設定成一台網路列印伺服器；這時候的FreeBSD 能接收其他電腦送來的列印工作，包括其他FreeBSD 的電腦、Windows 的電腦以及Mac OS 的電腦。FreeBSD 會確保同時只有一件文件正在列印，而且可以統計哪個使用者及機器印得最多，還有就是印出接下來是誰的文件這類的“標題”頁等。

讀完這章，您將了解：

- 如何設定FreeBSD 的列印多工緩衝處理器。
- 如何安裝列印過濾器以分別處理特殊的列印工作，包括把收到的文件轉換成您的印表機看得懂的列印格式等。
- 了解如何在您列印時順便印出頁首或標題。
- 如何利用別台電腦上的印表機列印。
- 如何利用直接接在網路上的印表機列印。
- 如何控制印表機的權限，包括限制列印工作的檔案大小，以及不允許特定使用者列印等。
- 如何記下印表機的統計資料，以及各帳號的印表機使用量。
- 如何解決列印時遇到的問題。

在開始閱讀這章之前，您需要：

- 要有設定、編譯kernel 的基礎概念(Chapter 8)。

9.2 介紹

要在FreeBSD 上使用印表機，您需要設定好Berkeley 行列式印表機列印緩衝系統，又稱為**LPD** 列印緩衝系統，或者就叫他**LPD** 吧。這是FreeBSD 標準的印表機控制系統，本章會介紹並教您如何設定**LPD**。

如果您已經對**LPD** 或是其他列印緩衝系統很熟悉了，您可以直接跳到基本設定。

LPD 控制著主機上印表機的一切。它負責這些工作：

- 控制本機及網路印表機的使用。
- 讓使用者可以列印文件，送出的文件稱為*工作*。
- 為每台印表機準備一個*佇列*，避免多個使用者同時使用同一台印表機。
- 列印*header pages* (又稱為*banner* or *burst* pages)，方便使用者在出紙閘中找到自己列印的文件。
- 把接在串列埠上的印表機的通訊參數設定好。

· 利用網路傳送列印工作給別台主機上的**LPD**。

· 執行特別的過濾程式將列印工作格式化以配合不同的列印語言或印表機。

· 統計印表機的使用情況。

藉由設定檔(`/etc/printcap`) 以及過濾程式的幫助，您可以讓大多數的印表機配合**LPD** 達成上述全部或部份的功能。

9.2.1 為什麼需要使用多工緩衝處理器

如果您的系統是個人使用，不需要控制存取權限、列印標題頁或者統計使用情況等功能時，您可能會覺得很奇怪為什麼還需要去管這個多工緩衝處理器。當然要直接控制印表機可行的，不過無論如何您還是需要多工緩衝處理器，因為：

· **LPD** 可以在背景(background) 列印，您不需要在那邊等文件送到印表機。

· **LPD** 可以很輕鬆地用過濾器增加日期/ 時間於頁首或是把特別的檔案格式(像是TEX DVI 檔) 轉換成印表機看得懂的的格式，您不需要手動去做這些步驟。

· 許多免費或商業軟體提供的列印功能通常都是和多工緩衝處理器溝通。透過設定緩衝系統，支援您現有或是即將要安裝的其他軟體將變得更容易。

9.3 基礎設定

要用印表機搭配**LPD** 多工緩衝系統，您需要有印表機這個硬體以及**LPD** 這套軟體。本手冊提供了兩階段的設定說明：

· 閱讀簡易印表機設定 來學習如何連接印表機、讓印表機和**LPD** 溝通以及列印純文字文件。

· 閱讀進階印表機設定 來學習如何列印各種特殊格式文件、列印首頁、網路列印、控制印表機權限以及統計使用狀況等。

9.3.1 簡易印表機設定

本章節會告訴您如何設定印表機設備和**LPD** 軟體以使用印表機，基本教學內容：

· 硬體設定 會提示如何將印表機接上電腦的連接埠。

· 軟體設定 會示範如何寫**LPD** 緩衝器設定檔(`/etc/printcap`)。

如果您要把印表機設定接收網路列印資料而不是本機端的話，請參考 印表機及網路資料傳輸介面。

這個章節雖然叫做“簡易印表機設定”，實際上還是有點複雜的。最困難的部份是讓你的印表機和電腦上的**LPD** 緩衝器能夠正常運作。一旦印表機可以正常工作之後，像是印首頁或是做列印統計這些進階的功能就不難做到了。

9.3.1.1 硬體設定

本章節討論各種連接印表機到PC 的方式。這裡會提到不同種類的連接埠和連接線，以及為了讓FreeBSD 能和印表機溝通您可能會需要開啓的核心參數等。

如果您已經把印表機接上電腦，而且在其他作業系統上有成功列印過的話，可以直接跳至軟體設定。

9.3.1.1.1 連接埠和排線

市售個人電腦印表機一般來說不出這三種界面：

- *序列(Serial)* 界面，又稱為RS-232 或COM 埠，用您電腦上的序列埠傳送資料到印表機。序列界面廣泛的為電腦業界所採用，所以排線容易取得，要設定連線並不困難。然而序列介面有時候會需要使用較特別的排線，這時候就有可能需要設定一些較為複雜的通訊參數了。大部份PC 序列埠的傳輸速度最高只到115200 bps，因此想要用序列埠來列印大圖是不切實際的。
- *並列(Parallel)* 界面利用電腦的並列埠將資料送到印表機。並列埠比RS-232 序列埠還快，也是一種電腦業界常用的界面。這種界面的排線非常容易取得，但是較難用手工打造。通常來說並列界面並沒有什麼通訊參數需要指定，所以設定起來超級容易。

  並列埠界面有時候也會被稱為“Centronics” 界面，這是印表機的接頭的名稱。
- USB 界面，也就是通用序列匯流排，傳輸速率比並列界面或是RS-232 序列界面都來得快，而且USB 排線單純又便宜。對列印工作而言，USB 比RS-232 序列埠或是並列埠都來得好，但是在UNIX 系統上的支援度較差。購買同時具有USB 及並列埠兩種界面的印表機可以避免掉這種問題。

一般而言，並列界面只能提供單向傳輸(電腦至印表機)，而要用USB 才能提供雙向。然而在FreeBSD 下，使用較新的並列埠(EPP 和ECP) 以及印表機，再配合使用IEEE-1284 相容排線也可以做到雙向溝通。

電腦和印表機之間藉由並列埠行進雙向溝通的方式有兩種。第一種是使用特製的、能和特定印表機溝通的FreeBSD 印表機驅動程式。這種方式在噴墨印表機上很常見，用來回報墨水存量以及其他狀態資訊等。第二種方法是用PostScript，如果印表機有支援的話。

PostScript jobs are actually programs sent to the printer; they need not produce paper at all and may return results directly to the computer. PostScript also uses two-way communication to tell the computer about problems, such as errors in the PostScript program or paper jams. Your users may be appreciative of such information. Furthermore, the best way to do effective accounting with a PostScript printer requires two-way communication: you ask the printer for its page count (how many pages it has printed in its lifetime), then send the user’s job, then ask again for its page count. Subtract the two values and you know how much paper to charge to the user.

9.3.1.1.2 Parallel Ports

To hook up a printer using a parallel interface, connect the Centronics cable between the printer and the computer. The instructions that came with the printer, the computer, or both should give you complete guidance.

Remember which parallel port you used on the computer. The first parallel port is `ppc0` to FreeBSD; the second is `ppc1`, and so on. The printer device name uses the same scheme: `/dev/lpt0` for the printer on the first parallel ports etc.

9.3.1.1.3 Serial Ports

To hook up a printer using a serial interface, connect the proper serial cable between the printer and the computer. The instructions that came with the printer, the computer, or both should give you complete guidance.

If you are unsure what the “proper serial cable” is, you may wish to try one of the following alternatives:

- A *modem* cable connects each pin of the connector on one end of the cable straight through to its corresponding pin of the connector on the other end. This type of cable is also known as a “DTE-to-DCE” cable.
- A *null-modem* cable connects some pins straight through, swaps others (send data to receive data, for example), and shorts some internally in each connector hood. This type of cable is also known as a “DTE-to-DTE” cable.
- A *serial printer* cable, required for some unusual printers, is like the null-modem cable, but sends some signals to their counterparts instead of being internally shorted.

You should also set up the communications parameters for the printer, usually through front-panel controls or DIP switches on the printer. Choose the highest `bps` (bits per second, sometimes *baud rate*) that both your computer and the printer can support. Choose 7 or 8 data bits; none, even, or odd parity; and 1 or 2 stop bits. Also choose a flow control protocol: either none, or XON/XOFF (also known as “in-band” or “software”) flow control. Remember these settings for the software configuration that follows.

9.3.1.2 Software Setup

This section describes the software setup necessary to print with the **LPD** spooling system in FreeBSD.

Here is an outline of the steps involved:

1. Configure your kernel, if necessary, for the port you are using for the printer; section Kernel Configuration tells you what you need to do.
2. Set the communications mode for the parallel port, if you are using a parallel port; section Setting the Communication Mode for the Parallel Port gives details.
3. Test if the operating system can send data to the printer. Section Checking Printer Communications gives some suggestions on how to do this.
4. Set up **LPD** for the printer by modifying the file `/etc/printcap`. You will find out how to do this later in this chapter.

9.3.1.2.1 Kernel Configuration

The operating system kernel is compiled to work with a specific set of devices. The serial or parallel interface for your printer is a part of that set. Therefore, it might be necessary to add support for an additional serial or parallel port if your kernel is not already configured for one.

To find out if the kernel you are currently using supports a serial interface, type:

```
# grep sioN /var/run/dmesg.boot
```

Where `N` is the number of the serial port, starting from zero. If you see output similar to the following:

```
sio2 at port 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
```

then the kernel supports the port.

To find out if the kernel supports a parallel interface, type:

```
# grep ppcN /var/run/dmesg.boot
```

Where *N* is the number of the parallel port, starting from zero. If you see output similar to the following:

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/8 bytes threshold
```

then the kernel supports the port.

You might have to reconfigure your kernel in order for the operating system to recognize and use the parallel or serial port you are using for the printer.

To add support for a serial port, see the section on kernel configuration. To add support for a parallel port, see that section *and* the section that follows.

9.3.1.3 Setting the Communication Mode for the Parallel Port

When you are using the parallel interface, you can choose whether FreeBSD should use interrupt-driven or polled communication with the printer. The generic printer device driver (lpt(4)) on FreeBSD uses the ppbus(4) system, which controls the port chipset with the ppc(4) driver.

- The *interrupt-driven* method is the default with the GENERIC kernel. With this method, the operating system uses an IRQ line to determine when the printer is ready for data.
- The *polled* method directs the operating system to repeatedly ask the printer if it is ready for more data. When it responds ready, the kernel sends more data.

The interrupt-driven method is usually somewhat faster but uses up a precious IRQ line. Some newer HP printers are claimed not to work correctly in interrupt mode, apparently due to some (not yet exactly understood) timing problem. These printers need polled mode. You should use whichever one works. Some printers will work in both modes, but are painfully slow in interrupt mode.

You can set the communications mode in two ways: by configuring the kernel or by using the lptcontrol(8) program.

To set the communications mode by configuring the kernel:

1. Edit your kernel configuration file. Look for an `ppc0` entry. If you are setting up the second parallel port, use `ppc1` instead. Use `ppc2` for the third port, and so on.
   - If you want interrupt-driven mode, edit the following line:

     `hint.ppc.0.irq="N"`

     in the `/boot/device.hints` file and replace *N* with the right IRQ number. The kernel configuration file must also contain the ppc(4) driver:

     `device ppc`
   - If you want polled mode, remove in your `/boot/device.hints` file, the following line:

     `hint.ppc.0.irq="N"`

     In some cases, this is not enough to put the port in polled mode under FreeBSD. Most of time it comes from acpi(4) driver, this latter is able to probe and attach devices, and therefore, control the access mode to the printer port. You should check your acpi(4) configuration to correct this problem.

2. Save the file. Then configure, build, and install the kernel, then reboot. See kernel configuration for more details.

To set the communications mode with lptcontrol(8):

1. Type:

   ```
   # lptcontrol -i -d /dev/lptN
   ```

   to set interrupt-driven mode for `lptN`.

2. Type:

   ```
   # lptcontrol -p -d /dev/lptN
   ```

   to set polled-mode for `lptN`.

You could put these commands in your `/etc/rc.local` file to set the mode each time your system boots. See lptcontrol(8) for more information.

9.3.1.4 Checking Printer Communications

Before proceeding to configure the spooling system, you should make sure the operating system can successfully send data to your printer. It is a lot easier to debug printer communication and the spooling system separately.

To test the printer, we will send some text to it. For printers that can immediately print characters sent to them, the program lptest(1) is perfect: it generates all 96 printable ASCII characters in 96 lines.

For a PostScript (or other language-based) printer, we will need a more sophisticated test. A small PostScript program, such as the following, will suffice:

```
%!PS
100 100 moveto 300 300 lineto stroke
310 310 moveto /Helvetica findfont 12 scalefont setfont
(Is this thing working?) show
showpage
```

The above PostScript code can be placed into a file and used as shown in the examples appearing in the following sections.

> **Note:** When this document refers to a printer language, it is assuming a language like PostScript, and not Hewlett Packard's PCL. Although PCL has great functionality, you can intermingle plain text with its escape sequences. PostScript cannot directly print plain text, and that is the kind of printer language for which we must make special accommodations.

9.3.1.4.1 Checking a Parallel Printer

This section tells you how to check if FreeBSD can communicate with a printer connected to a parallel port.

To test a printer on a parallel port:

1. Become `root` with su(1).
2. Send data to the printer.

- If the printer can print plain text, then use lptest(1). Type:

  ```
  # lptest > /dev/lptN
  ```

  Where *N* is the number of the parallel port, starting from zero.

- If the printer understands PostScript or other printer language, then send a small program to the printer. Type:

  ```
  # cat > /dev/lptN
  ```

  Then, line by line, type the program *carefully* as you cannot edit a line once you have pressed RETURN or ENTER. When you have finished entering the program, press CONTROL+D, or whatever your end of file key is.

  Alternatively, you can put the program in a file and type:

  ```
  # cat file > /dev/lptN
  ```

  Where *file* is the name of the file containing the program you want to send to the printer.

You should see something print. Do not worry if the text does not look right; we will fix such things later.

9.3.1.4.2 Checking a Serial Printer

This section tells you how to check if FreeBSD can communicate with a printer on a serial port.

To test a printer on a serial port:

1. Become root with su(1).
2. Edit the file /etc/remote. Add the following entry:

   ```
   printer:dv=/dev/port:br#bps-rate:pa=parity
   ```

   Where *port* is the device entry for the serial port (ttyd0, ttyd1, etc.), *bps-rate* is the bits-per-second rate at which the printer communicates, and *parity* is the parity required by the printer (either even, odd, none, or zero).

   Here is a sample entry for a printer connected via a serial line to the third serial port at 19200 bps with no parity:

   ```
   printer:dv=/dev/ttyd2:br#19200:pa=none
   ```

3. Connect to the printer with tip(1). Type:

   ```
   # tip printer
   ```

   If this step does not work, edit the file /etc/remote again and try using /dev/cuaa*N* instead of /dev/ttyd*N*.

4. Send data to the printer.

   - If the printer can print plain text, then use lptest(1). Type:

     ```
     % $lptest
     ```

   - If the printer understands PostScript or other printer language, then send a small program to the printer. Type the program, line by line, *very carefully* as backspacing or other editing keys may be significant to the printer. You may also need to type a special end-of-file key for the printer so it knows it received the whole program. For PostScript printers, press CONTROL+D.

     Alternatively, you can put the program in a file and type:

     ```
     % >file
     ```

Where *file* is the name of the file containing the program. After tip(1) sends the file, press any required end-of-file key.

You should see something print. Do not worry if the text does not look right; we will fix that later.

9.3.1.5 Enabling the Spooler: the /etc/printcap File

At this point, your printer should be hooked up, your kernel configured to communicate with it (if necessary), and you have been able to send some simple data to the printer. Now, we are ready to configure **LPD** to control access to your printer.

You configure **LPD** by editing the file /etc/printcap. The **LPD** spooling system reads this file each time the spooler is used, so updates to the file take immediate effect.

The format of the printcap(5) file is straightforward. Use your favorite text editor to make changes to /etc/printcap. The format is identical to other capability files like /usr/share/misc/termcap and /etc/remote. For complete information about the format, see the cgetent(3).

The simple spooler configuration consists of the following steps:

1. Pick a name (and a few convenient aliases) for the printer, and put them in the /etc/printcap file; see the Naming the Printer section for more information on naming.
2. Turn off header pages (which are on by default) by inserting the sh capability; see the Suppressing Header Pages section for more information.
3. Make a spooling directory, and specify its location with the sd capability; see the Making the Spooling Directory section for more information.
4. Set the /dev entry to use for the printer, and note it in /etc/printcap with the lp capability; see the Identifying the Printer Device for more information. Also, if the printer is on a serial port, set up the communication parameters with the ms# capability which is discussed in the Configuring Spooler Communications Parameters section.
5. Install a plain text input filter; see the Installing the Text Filter section for details.
6. Test the setup by printing something with the lpr(1) command. More details are available in the Trying It Out and Troubleshooting sections.

Note: Language-based printers, such as PostScript printers, cannot directly print plain text. The simple setup outlined above and described in the following sections assumes that if you are installing such a printer you will print only files that the printer can understand.

Users often expect that they can print plain text to any of the printers installed on your system. Programs that interface to **LPD** to do their printing usually make the same assumption. If you are installing such a printer and want to be able to print jobs in the printer language *and* print plain text jobs, you are strongly urged to add an additional step to the simple setup outlined above: install an automatic plain-text-to-PostScript (or other printer language) conversion program. The section entitled Accommodating Plain Text Jobs on PostScript Printers tells how to do this.

9.3.1.5.1 Naming the Printer

The first (easy) step is to pick a name for your printer. It really does not matter whether you choose functional or whimsical names since you can also provide a number of aliases for the printer.

At least one of the printers specified in the `/etc/printcap` should have the alias `lp`. This is the default printer's name. If users do not have the `PRINTER` environment variable nor specify a printer name on the command line of any of the **LPD** commands, then `lp` will be the default printer they get to use.

Also, it is common practice to make the last alias for a printer be a full description of the printer, including make and model.

Once you have picked a name and some common aliases, put them in the `/etc/printcap` file. The name of the printer should start in the leftmost column. Separate each alias with a vertical bar and put a colon after the last alias.

In the following example, we start with a skeletal `/etc/printcap` that defines two printers (a Diablo 630 line printer and a Panasonic KX-P4455 PostScript laser printer):

```
#
#  /etc/printcap for host rose
#
rattan|line|diablo|lp|Diablo 630 Line Printer:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:
```

In this example, the first printer is named `rattan` and has as aliases `line`, `diablo`, `lp`, and `Diablo 630 Line Printer`. Since it has the alias `lp`, it is also the default printer. The second is named `bamboo`, and has as aliases `ps`, `PS`, `S`, `panasonic`, and `Panasonic KX-P4455 PostScript v51.4`.

9.3.1.5.2 Suppressing Header Pages

The **LPD** spooling system will by default print a *header page* for each job. The header page contains the user name who requested the job, the host from which the job came, and the name of the job, in nice large letters. Unfortunately, all this extra text gets in the way of debugging the simple printer setup, so we will suppress header pages.

To suppress header pages, add the `sh` capability to the entry for the printer in `/etc/printcap`. Here is an example `/etc/printcap` with `sh` added:

```
#
#  /etc/printcap for host rose - no header pages anywhere
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:
```

Note how we used the correct format: the first line starts in the leftmost column, and subsequent lines are indented. Every line in an entry except the last ends in a backslash character.

9.3.1.5.3 Making the Spooling Directory

The next step in the simple spooler setup is to make a *spooling directory*, a directory where print jobs reside until they are printed, and where a number of other spooler support files live.

Because of the variable nature of spooling directories, it is customary to put these directories under `/var/spool`. It is not necessary to backup the contents of spooling directories, either. Recreating them is as simple as running mkdir(1).

It is also customary to make the directory with a name that is identical to the name of the printer, as shown below:

```
# mkdir /var/spool/printer-name
```

However, if you have a lot of printers on your network, you might want to put the spooling directories under a single directory that you reserve just for printing with **LPD**. We will do this for our two example printers `rattan` and `bamboo`:

```
# mkdir /var/spool/lpd
# mkdir /var/spool/lpd/rattan
# mkdir /var/spool/lpd/bamboo
```

> **Note:** If you are concerned about the privacy of jobs that users print, you might want to protect the spooling directory so it is not publicly accessible. Spooling directories should be owned and be readable, writable, and searchable by user daemon and group daemon, and no one else. We will do this for our example printers:
>
> ```
> # chown daemon:daemon /var/spool/lpd/rattan
> # chown daemon:daemon /var/spool/lpd/bamboo
> # chmod 770 /var/spool/lpd/rattan
> # chmod 770 /var/spool/lpd/bamboo
> ```

Finally, you need to tell **LPD** about these directories using the `/etc/printcap` file. You specify the pathname of the spooling directory with the `sd` capability:

```
#
#  /etc/printcap for host rose - added spooling directories
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:
```

Note that the name of the printer starts in the first column but all other entries describing the printer should be indented and each line end escaped with a backslash.

If you do not specify a spooling directory with `sd`, the spooling system will use `/var/spool/lpd` as a default.

9.3.1.5.4 Identifying the Printer Device

In the Entries for the Ports section, we identified which entry in the `/dev` directory FreeBSD will use to communicate with the printer. Now, we tell **LPD** that information. When the spooling system has a job to print, it will open the specified device on behalf of the filter program (which is responsible for passing data to the printer).

List the `/dev` entry pathname in the `/etc/printcap` file using the `lp` capability.

In our running example, let us assume that `rattan` is on the first parallel port, and `bamboo` is on a sixth serial port; here are the additions to `/etc/printcap`:

```
#
#  /etc/printcap for host rose - identified what devices to use
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:
```

If you do not specify the `lp` capability for a printer in your `/etc/printcap` file, **LPD** uses `/dev/lp` as a default. `/dev/lp` currently does not exist in FreeBSD.

If the printer you are installing is connected to a parallel port, skip to the section entitled, Installing the Text Filter. Otherwise, be sure to follow the instructions in the next section.

9.3.1.5.5 Configuring Spooler Communication Parameters

For printers on serial ports, **LPD** can set up the bps rate, parity, and other serial communication parameters on behalf of the filter program that sends data to the printer. This is advantageous since:

- It lets you try different communication parameters by simply editing the `/etc/printcap` file; you do not have to recompile the filter program.
- It enables the spooling system to use the same filter program for multiple printers which may have different serial communication settings.

The following `/etc/printcap` capabilities control serial communication parameters of the device listed in the `lp` capability:

`br#`*`bps-rate`*

Sets the communications speed of the device to *`bps-rate`*, where *`bps-rate`* can be 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, or 115200 bits-per-second.

`ms#`*`stty-mode`*

Sets the options for the terminal device after opening the device. stty(1) explains the available options.

When **LPD** opens the device specified by the `lp` capability, it sets the characteristics of the device to those specified with the `ms#` capability. Of particular interest will be the `parenb`, `parodd`, `cs5`, `cs6`, `cs7`, `cs8`, `cstopb`, `crtscts`, and `ixon` modes, which are explained in the stty(1) manual page.

Let us add to our example printer on the sixth serial port. We will set the bps rate to 38400. For the mode, we will set no parity with `-parenb`, 8-bit characters with `cs8`, no modem control with `clocal` and hardware flow control with `crtscts`:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:
```

9.3.1.5.6 Installing the Text Filter

We are now ready to tell **LPD** what text filter to use to send jobs to the printer. A *text filter*, also known as an *input filter*, is a program that **LPD** runs when it has a job to print. When **LPD** runs the text filter for a printer, it sets the filter's standard input to the job to print, and its standard output to the printer device specified with the `lp` capability. The filter is expected to read the job from standard input, perform any necessary translation for the printer, and write the results to standard output, which will get printed. For more information on the text filter, see the Filters section.

For our simple printer setup, the text filter can be a small shell script that just executes `/bin/cat` to send the job to the printer. FreeBSD comes with another filter called `lpf` that handles backspacing and underlining for printers that might not deal with such character streams well. And, of course, you can use any other filter program you want. The filter `lpf` is described in detail in section entitled lpf: a Text Filter.

First, let us make the shell script `/usr/local/libexec/if-simple` be a simple text filter. Put the following text into that file with your favorite text editor:

```
#!/bin/sh
#
# if-simple - Simple text input filter for lpd
# Installed in /usr/local/libexec/if-simple
#
# Simply copies stdin to stdout.  Ignores all filter arguments.

/bin/cat && exit 0
exit 2
```

Make the file executable:

```
# chmod 555 /usr/local/libexec/if-simple
```

And then tell LPD to use it by specifying it with the `if` capability in `/etc/printcap`. We will add it to the two printers we have so far in the example `/etc/printcap`:

```
#
#  /etc/printcap for host rose - added text filter
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\ :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:\
        :if=/usr/local/libexec/if-simple:
```

Note: A copy of the `if-simple` script can be found in the `/usr/share/examples/printing` directory.

9.3.1.5.7 Turn on ***LPD***

lpd(8) is run from `/etc/rc`, controlled by the `lpd_enable` variable. This variable defaults to `NO`. If you have not done so already, add the line:

```
lpd_enable="YES"
```

to `/etc/rc.conf`, and then either restart your machine, or just run lpd(8).

```
# lpd
```

9.3.1.5.8 Trying It Out

You have reached the end of the simple **LPD** setup. Unfortunately, congratulations are not quite yet in order, since we still have to test the setup and correct any problems. To test the setup, try printing something. To print with the **LPD** system, you use the command lpr(1), which submits a job for printing.

You can combine lpr(1) with the lptest(1) program, introduced in section Checking Printer Communications to generate some test text.

*To test the simple **LPD** setup:*

Type:

```
# lptest 20 5 | lpr -Pprinter-name
```

Where *`printer-name`* is a the name of a printer (or an alias) specified in `/etc/printcap`. To test the default printer, type lpr(1) without any `-P` argument. Again, if you are testing a printer that expects PostScript, send a PostScript program in that language instead of using lptest(1). You can do so by putting the program in a file and typing `lpr` *`file`*.

For a PostScript printer, you should get the results of the program. If you are using lptest(1), then your results should look like the following:

```
!"#$%&'()*+,-./01234
"#$%&'()*+,-./012345
#$%&'()*+,-./0123456
$%&'()*+,-./01234567
%&'()*+,-./012345678
```

To further test the printer, try downloading larger programs (for language-based printers) or running lptest(1) with different arguments. For example, `lptest 80 60` will produce 60 lines of 80 characters each.

If the printer did not work, see the Troubleshooting section.

9.4 Advanced Printer Setup

This section describes filters for printing specially formatted files, header pages, printing across networks, and restricting and accounting for printer usage.

9.4.1 Filters

Although **LPD** handles network protocols, queuing, access control, and other aspects of printing, most of the *real* work happens in the *filters*. Filters are programs that communicate with the printer and handle its device dependencies and special requirements. In the simple printer setup, we installed a plain text filter——an extremely simple one that should work with most printers (section Installing the Text Filter).

However, in order to take advantage of format conversion, printer accounting, specific printer quirks, and so on, you should understand how filters work. It will ultimately be the filter's responsibility to handle these aspects. And the bad news is that most of the time *you* have to provide filters yourself. The good news is that many are generally available; when they are not, they are usually easy to write.

Also, FreeBSD comes with one, `/usr/libexec/lpr/lpf`, that works with many printers that can print plain text. (It handles backspacing and tabs in the file, and does accounting, but that is about all it does.) There are also several filters and filter components in the FreeBSD Ports Collection.

Here is what you will find in this section:

- Section How Filters Work, tries to give an overview of a filter's role in the printing process. You should read this section to get an understanding of what is happening "under the hood" when **LPD** uses filters. This knowledge could help you anticipate and debug problems you might encounter as you install more and more filters on each of your printers.
- **LPD** expects every printer to be able to print plain text by default. This presents a problem for PostScript (or other language-based printers) which cannot directly print plain text. Section Accommodating Plain Text Jobs on PostScript Printers tells you what you should do to overcome this problem. You should read this section if you have a PostScript printer.
- PostScript is a popular output format for many programs. Some people even write PostScript code directly. Unfortunately, PostScript printers are expensive. Section Simulating PostScript on Non PostScript Printers tells how you can further modify a printer's text filter to accept and print PostScript data on a *non PostScript* printer. You should read this section if you do not have a PostScript printer.
- Section Conversion Filters tells about a way you can automate the conversion of specific file formats, such as graphic or typesetting data, into formats your printer can understand. After reading this section, you should be able to set up your printers such that users can type `lpr -t` to print troff data, or `lpr -d` to print TEX DVI data, or `lpr -v` to print raster image data, and so forth. I recommend reading this section.
- Section Output Filters tells all about a not often used feature of **LPD**: output filters. Unless you are printing header pages (see Header Pages), you can probably skip that section altogether.
- Section lpf: a Text Filter describes `lpf`, a fairly complete if simple text filter for line printers (and laser printers that act like line printers) that comes with FreeBSD. If you need a quick way to get printer accounting working for plain text, or if you have a printer which emits smoke when it sees backspace characters, you should definitely consider `lpf`.

> **Note:** A copy of the various scripts described below can be found in the `/usr/share/examples/printing` directory.

9.4.1.1 How Filters Work

As mentioned before, a filter is an executable program started by **LPD** to handle the device-dependent part of communicating with the printer.

When **LPD** wants to print a file in a job, it starts a filter program. It sets the filter's standard input to the file to print, its standard output to the printer, and its standard error to the error logging file (specified in the `lf` capability in `/etc/printcap`, or `/dev/console` by default).

Which filter **LPD** starts and the filter's arguments depend on what is listed in the `/etc/printcap` file and what arguments the user specified for the job on the lpr(1) command line. For example, if the user typed `lpr -t`, **LPD** would start the troff filter, listed in the `tf` capability for the destination printer. If the user wanted to print plain text, it would start the `if` filter (this is mostly true: see Output Filters for details).

There are three kinds of filters you can specify in `/etc/printcap`:

- The *text filter*, confusingly called the *input filter* in **LPD** documentation, handles regular text printing. Think of it as the default filter. **LPD** expects every printer to be able to print plain text by default, and it is the text filter's job to make sure backspaces, tabs, or other special characters do not confuse the printer. If you are in an environment where you have to account for printer usage, the text filter must also account for pages printed, usually by counting the number of lines printed and comparing that to the number of lines per page the printer supports. The text filter is started with the following argument list:

`filter-name` [-c] -w*width* -l*length* -i*indent* -n *login* -h *host acct-file*

where

`-c`

appears if the job is submitted with `lpr -l`

`width`

is the value from the `pw` (page width) capability specified in `/etc/printcap`, default 132

`length`

is the value from the `pl` (page length) capability, default 66

`indent`

is the amount of the indentation from `lpr -i`, default 0

`login`

is the account name of the user printing the file

`host`

is the host name from which the job was submitted

acct-file

is the name of the accounting file from the `af` capability.

- A *conversion filter* converts a specific file format into one the printer can render onto paper. For example, ditroff typesetting data cannot be directly printed, but you can install a conversion filter for ditroff files to convert the ditroff data into a form the printer can digest and print. Section Conversion Filters tells all about them. Conversion filters also need to do accounting, if you need printer accounting. Conversion filters are started with the following arguments:

`filter-name` -x*pixel-width* -y*pixel-height* -n *login* -h *host acct-file*

where *pixel-width* is the value from the `px` capability (default 0) and *pixel-height* is the value from the `py` capability (default 0).

- The *output filter* is used only if there is no text filter, or if header pages are enabled. In my experience, output filters are rarely used. Section Output Filters describe them. There are only two arguments to an output filter:

`filter-name` -w*width* -l*length*

which are identical to the text filters `-w` and `-l` arguments.

Filters should also *exit* with the following exit status:

exit 0

If the filter printed the file successfully.

exit 1

If the filter failed to print the file but wants **LPD** to try to print the file again. **LPD** will restart a filter if it exits with this status.

exit 2

If the filter failed to print the file and does not want **LPD** to try again. **LPD** will throw out the file.

The text filter that comes with the FreeBSD release, `/usr/libexec/lpr/lpf`, takes advantage of the page width and length arguments to determine when to send a form feed and how to account for printer usage. It uses the login, host, and accounting file arguments to make the accounting entries.

If you are shopping for filters, see if they are LPD-compatible. If they are, they must support the argument lists described above. If you plan on writing filters for general use, then have them support the same argument lists and exit codes.

9.4.1.2 Accommodating Plain Text Jobs on PostScript® Printers

If you are the only user of your computer and PostScript (or other language-based) printer, and you promise to never send plain text to your printer and to never use features of various programs that will want to send plain text to your printer, then you do not need to worry about this section at all.

But, if you would like to send both PostScript and plain text jobs to the printer, then you are urged to augment your printer setup. To do so, we have the text filter detect if the arriving job is plain text or PostScript. All PostScript jobs must start with `%!` (for other printer languages, see your printer documentation). If those are the first two characters

in the job, we have PostScript, and can pass the rest of the job directly. If those are not the first two characters in the file, then the filter will convert the text into PostScript and print the result.

How do we do this?

If you have got a serial printer, a great way to do it is to install `lprps`. `lprps` is a PostScript printer filter which performs two-way communication with the printer. It updates the printer's status file with verbose information from the printer, so users and administrators can see exactly what the state of the printer is (such as "`toner low`" or "`paper jam`"). But more importantly, it includes a program called `psif` which detects whether the incoming job is plain text and calls `textps` (another program that comes with `lprps`) to convert it to PostScript. It then uses `lprps` to send the job to the printer.

`lprps` is part of the FreeBSD Ports Collection (see The Ports Collection). You can fetch, build and install it yourself, of course. After installing `lprps`, just specify the pathname to the `psif` program that is part of `lprps`. If you installed `lprps` from the Ports Collection, use the following in the serial PostScript printer's entry in `/etc/printcap`:

```
:if=/usr/local/libexec/psif:
```

You should also specify the `rw` capability; that tells **LPD** to open the printer in read-write mode.

If you have a parallel PostScript printer (and therefore cannot use two-way communication with the printer, which `lprps` needs), you can use the following shell script as the text filter:

```
#!/bin/sh
#
#  psif - Print PostScript or plain text on a PostScript printer
#  Script version; NOT the version that comes with lprps
#  Installed in /usr/local/libexec/psif
#

IFS="" read -r first_line
first_two_chars=‘expr "$first_line" : ’\(..\)’‘

if [ "$first_two_chars" = "%!" ]; then
    #
    #  PostScript job, print it.
    #
    echo "$first_line" && cat && printf "\004" && exit 0
    exit 2
else
    #
    #  Plain text, convert it, then print it.
    #
    ( echo "$first_line"; cat ) | /usr/local/bin/textps && printf "\004" && exit 0
    exit 2
fi
```

In the above script, `textps` is a program we installed separately to convert plain text to PostScript. You can use any text-to-PostScript program you wish. The FreeBSD Ports Collection (see The Ports Collection) includes a full featured text-to-PostScript program called `a2ps` that you might want to investigate.

9.4.1.3 Simulating PostScript on Non PostScript Printers

PostScript is the *de facto* standard for high quality typesetting and printing. PostScript is, however, an *expensive* standard. Thankfully, Aladdin Enterprises has a free PostScript work-alike called **Ghostscript** that runs with FreeBSD. Ghostscript can read most PostScript files and can render their pages onto a variety of devices, including many brands of non-PostScript printers. By installing Ghostscript and using a special text filter for your printer, you can make your non PostScript printer act like a real PostScript printer.

Ghostscript is in the FreeBSD Ports Collection, if you would like to install it from there. You can fetch, build, and install it quite easily yourself, as well.

To simulate PostScript, we have the text filter detect if it is printing a PostScript file. If it is not, then the filter will pass the file directly to the printer; otherwise, it will use Ghostscript to first convert the file into a format the printer will understand.

Here is an example: the following script is a text filter for Hewlett Packard DeskJet 500 printers. For other printers, substitute the `-sDEVICE` argument to the `gs` (Ghostscript) command. (Type `gs -h` to get a list of devices the current installation of Ghostscript supports.)

```
#!/bin/sh
#
#  ifhp - Print Ghostscript-simulated PostScript on a DeskJet 500
#  Installed in /usr/local/libexec/ifhp

#
#  Treat LF as CR+LF (to avoid the "staircase effect" on HP/PCL
#  printers):
#
printf "\033&k2G" || exit 2

#
#  Read first two characters of the file
#
IFS="" read -r first_line
first_two_chars=‘expr "$first_line" : ’\(..\)’‘

if [ "$first_two_chars" = "%!" ]; then
    #
    #  It is PostScript; use Ghostscript to scan-convert and print it.
    #
    /usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=djet500 \
      -sOutputFile=- - && exit 0
else
    #
    #  Plain text or HP/PCL, so just print it directly; print a form feed
    #  at the end to eject the last page.
    #
    echo "$first_line" && cat && printf "\033&l0H" &&
exit 0
fi

exit 2
```

Finally, you need to notify **LPD** of the filter via the `if` capability:

```
:if=/usr/local/libexec/ifhp:
```

That is it. You can type `lpr plain.text` and `lpr whatever.ps` and both should print successfully.

9.4.1.4 Conversion Filters

After completing the simple setup described in Simple Printer Setup, the first thing you will probably want to do is install conversion filters for your favorite file formats (besides plain ASCII text).

9.4.1.4.1 Why Install Conversion Filters?

Conversion filters make printing various kinds of files easy. As an example, suppose we do a lot of work with the TEX typesetting system, and we have a PostScript printer. Every time we generate a DVI file from TEX, we cannot print it directly until we convert the DVI file into PostScript. The command sequence goes like this:

```
% dvips seaweed-analysis.dvi
% lpr seaweed-analysis.ps
```

By installing a conversion filter for DVI files, we can skip the hand conversion step each time by having **LPD** do it for us. Now, each time we get a DVI file, we are just one step away from printing it:

```
% lpr -d seaweed-analysis.dvi
```

We got **LPD** to do the DVI file conversion for us by specifying the `-d` option. Section Formatting and Conversion Options lists the conversion options.

For each of the conversion options you want a printer to support, install a *conversion filter* and specify its pathname in `/etc/printcap`. A conversion filter is like the text filter for the simple printer setup (see section Installing the Text Filter) except that instead of printing plain text, the filter converts the file into a format the printer can understand.

9.4.1.4.2 Which Conversion Filters Should I Install?

You should install the conversion filters you expect to use. If you print a lot of DVI data, then a DVI conversion filter is in order. If you have got plenty of troff to print out, then you probably want a troff filter.

The following table summarizes the filters that **LPD** works with, their capability entries for the `/etc/printcap` file, and how to invoke them with the `lpr` command:

File type	`/etc/printcap` capability	`lpr` option
cifplot	`cf`	`-c`
DVI	`df`	`-d`
plot	`gf`	`-g`
ditroff	`nf`	`-n`
FORTRAN text	`rf`	`-f`
troff	`tf`	`-f`
raster	`vf`	`-v`
plain text	`if`	none, `-p`, or `-l`

In our example, using `lpr -d` means the printer needs a `df` capability in its entry in `/etc/printcap`.

Despite what others might contend, formats like FORTRAN text and plot are probably obsolete. At your site, you can give new meanings to these or any of the formatting options just by installing custom filters. For example, suppose you would like to directly print Printerleaf files (files from the Interleaf desktop publishing program), but will never print plot files. You could install a Printerleaf conversion filter under the `gf` capability and then educate your users that `lpr -g` mean "print Printerleaf files."

9.4.1.4.3 Installing Conversion Filters

Since conversion filters are programs you install outside of the base FreeBSD installation, they should probably go under `/usr/local`. The directory `/usr/local/libexec` is a popular location, since they are specialized programs that only **LPD** will run; regular users should not ever need to run them.

To enable a conversion filter, specify its pathname under the appropriate capability for the destination printer in `/etc/printcap`.

In our example, we will add the DVI conversion filter to the entry for the printer named `bamboo`. Here is the example `/etc/printcap` file again, with the new `df` capability for the printer `bamboo`.

```
#
#  /etc/printcap for host rose - added df filter for bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

The DVI filter is a shell script named `/usr/local/libexec/psdf`. Here is that script:

```
#!/bin/sh
#
#  psdf - DVI to PostScript printer filter
#  Installed in /usr/local/libexec/psdf
#
# Invoked by lpd when user runs lpr -d
#
exec /usr/local/bin/dvips -f | /usr/local/libexec/lprps "$@"
```

This script runs `dvips` in filter mode (the `-f` argument) on standard input, which is the job to print. It then starts the PostScript printer filter `lprps` (see section Accommodating Plain Text Jobs on PostScript Printers) with the arguments **LPD** passed to this script. `lprps` will use those arguments to account for the pages printed.

9.4.1.4.4 More Conversion Filter Examples

Since there is no fixed set of steps to install conversion filters, let me instead provide more examples. Use these as guidance to making your own filters. Use them directly, if appropriate.

This example script is a raster (well, GIF file, actually) conversion filter for a Hewlett Packard LaserJet III-Si printer:

```
#!/bin/sh
#
#  hpvf - Convert GIF files into HP/PCL, then print
#  Installed in /usr/local/libexec/hpvf

PATH=/usr/X11R6/bin:$PATH; export PATH
giftopnm | ppmtopgm | pgmtopbm | pbmtolj -resolution 300 \
    && exit 0 \
    || exit 2
```

It works by converting the GIF file into a portable anymap, converting that into a portable graymap, converting that into a portable bitmap, and converting that into LaserJet/PCL-compatible data.

Here is the `/etc/printcap` file with an entry for a printer using the above filter:

```
#
#  /etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:
```

The following script is a conversion filter for troff data from the groff typesetting system for the PostScript printer named `bamboo`:

```
#!/bin/sh
#
#  pstf - Convert groff's troff data into PS, then print.
#  Installed in /usr/local/libexec/pstf
#
exec grops | /usr/local/libexec/lprps "$@"
```

The above script makes use of `lprps` again to handle the communication with the printer. If the printer were on a parallel port, we would use this script instead:

```
#!/bin/sh
#
#  pstf - Convert groff's troff data into PS, then print.
#  Installed in /usr/local/libexec/pstf
#
exec grops
```

That is it. Here is the entry we need to add to `/etc/printcap` to enable the filter:

```
:tf=/usr/local/libexec/pstf:
```

Here is an example that might make old hands at FORTRAN blush. It is a FORTRAN-text filter for any printer that can directly print plain text. We will install it for the printer `teak`:

```
#!/bin/sh
#
# hprf - FORTRAN text filter for LaserJet 3si:
# Installed in /usr/local/libexec/hprf
#

printf "\033&k2G" && fpr && printf "\033&l0H" &&
 exit 0
exit 2
```

And we will add this line to the `/etc/printcap` for the printer `teak` to enable this filter:

```
:rf=/usr/local/libexec/hprf:
```

Here is one final, somewhat complex example. We will add a DVI filter to the LaserJet printer `teak` introduced earlier. First, the easy part: updating `/etc/printcap` with the location of the DVI filter:

```
:df=/usr/local/libexec/hpdf:
```

Now, for the hard part: making the filter. For that, we need a DVI-to-LaserJet/PCL conversion program. The FreeBSD Ports Collection (see The Ports Collection) has one: `dvi2xx` is the name of the package. Installing this package gives us the program we need, `dvilj2p`, which converts DVI into LaserJet IIp, LaserJet III, and LaserJet 2000 compatible codes.

`dvilj2p` makes the filter `hpdf` quite complex since `dvilj2p` cannot read from standard input. It wants to work with a filename. What is worse, the filename has to end in `.dvi` so using `/dev/fd/0` for standard input is problematic. We can get around that problem by linking (symbolically) a temporary file name (one that ends in `.dvi`) to `/dev/fd/0`, thereby forcing `dvilj2p` to read from standard input.

The only other fly in the ointment is the fact that we cannot use `/tmp` for the temporary link. Symbolic links are owned by user and group `bin`. The filter runs as user `daemon`. And the `/tmp` directory has the sticky bit set. The filter can create the link, but it will not be able clean up when done and remove it since the link will belong to a different user.

Instead, the filter will make the symbolic link in the current working directory, which is the spooling directory (specified by the `sd` capability in `/etc/printcap`). This is a perfect place for filters to do their work, especially since there is (sometimes) more free disk space in the spooling directory than under `/tmp`.

Here, finally, is the filter:

```
#!/bin/sh
#
#  hpdf - Print DVI data on HP/PCL printer
#  Installed in /usr/local/libexec/hpdf

PATH=/usr/local/bin:$PATH; export PATH

#
#  Define a function to clean up our temporary files.  These exist
#  in the current directory, which will be the spooling directory
#  for the printer.
```

```
#
cleanup() {
   rm -f hpdf$$.dvi
}

#
#  Define a function to handle fatal errors: print the given message
#  and exit 2.  Exiting with 2 tells LPD to do not try to reprint the
#  job.
#
fatal() {
    echo "$@" 1>&2
    cleanup
    exit 2
}

#
#  If user removes the job, LPD will send SIGINT, so trap SIGINT
#  (and a few other signals) to clean up after ourselves.
#
trap cleanup 1 2 15

#
#  Make sure we are not colliding with any existing files.
#
cleanup

#
#  Link the DVI input file to standard input (the file to print).
#
ln -s /dev/fd/0 hpdf$$.dvi || fatal "Cannot symlink /dev/fd/0"

#
#  Make LF = CR+LF
#
printf "\033&k2G" || fatal "Cannot initialize printer"

#
#  Convert and print.  Return value from dvilj2p does not seem to be
#  reliable, so we ignore it.
#
dvilj2p -M1 -q -e- dfhp$$.dvi

#
#  Clean up and exit
#
cleanup
exit 0
```

9.4.1.4.5 Automated Conversion: an Alternative to Conversion Filters

All these conversion filters accomplish a lot for your printing environment, but at the cost forcing the user to specify (on the lpr(1) command line) which one to use. If your users are not particularly computer literate, having to specify a filter option will become annoying. What is worse, though, is that an incorrectly specified filter option may run a filter on the wrong type of file and cause your printer to spew out hundreds of sheets of paper.

Rather than install conversion filters at all, you might want to try having the text filter (since it is the default filter) detect the type of file it has been asked to print and then automatically run the right conversion filter. Tools such as `file` can be of help here. Of course, it will be hard to determine the differences between *some* file types——and, of course, you can still provide conversion filters just for them.

The FreeBSD Ports Collection has a text filter that performs automatic conversion called `apsfilter`. It can detect plain text, PostScript, and DVI files, run the proper conversions, and print.

9.4.1.5 Output Filters

The **LPD** spooling system supports one other type of filter that we have not yet explored: an output filter. An output filter is intended for printing plain text only, like the text filter, but with many simplifications. If you are using an output filter but no text filter, then:

- **LPD** starts an output filter once for the entire job instead of once for each file in the job.
- **LPD** does not make any provision to identify the start or the end of files within the job for the output filter.
- **LPD** does not pass the user's login or host to the filter, so it is not intended to do accounting. In fact, it gets only two arguments:

  `filter-name -w`*`width`* `-l`*`length`*

  Where *`width`* is from the `pw` capability and *`length`* is from the `pl` capability for the printer in question.

Do not be seduced by an output filter's simplicity. If you would like each file in a job to start on a different page an output filter *will not work*. Use a text filter (also known as an input filter); see section Installing the Text Filter. Furthermore, an output filter is actually *more complex* in that it has to examine the byte stream being sent to it for special flag characters and must send signals to itself on behalf of **LPD**.

However, an output filter is *necessary* if you want header pages and need to send escape sequences or other initialization strings to be able to print the header page. (But it is also *futile* if you want to charge header pages to the requesting user's account, since **LPD** does not give any user or host information to the output filter.)

On a single printer, **LPD** allows both an output filter and text or other filters. In such cases, **LPD** will start the output filter to print the header page (see section Header Pages) only. **LPD** then expects the output filter to *stop itself* by sending two bytes to the filter: ASCII 031 followed by ASCII 001. When an output filter sees these two bytes (031, 001), it should stop by sending `SIGSTOP` to itself. When **LPD**'s done running other filters, it will restart the output filter by sending `SIGCONT` to it.

If there is an output filter but *no* text filter and **LPD** is working on a plain text job, **LPD** uses the output filter to do the job. As stated before, the output filter will print each file of the job in sequence with no intervening form feeds or other paper advancement, and this is probably *not* what you want. In almost all cases, you need a text filter.

The program `lpf`, which we introduced earlier as a text filter, can also run as an output filter. If you need a quick-and-dirty output filter but do not want to write the byte detection and signal sending code, try `lpf`. You can also wrap `lpf` in a shell script to handle any initialization codes the printer might require.

9.4.1.6 `lpf`: a Text Filter

The program `/usr/libexec/lpr/lpf` that comes with FreeBSD binary distribution is a text filter (input filter) that can indent output (job submitted with `lpr -i`), allow literal characters to pass (job submitted with `lpr -l`), adjust the printing position for backspaces and tabs in the job, and account for pages printed. It can also act like an output filter.

`lpf` is suitable for many printing environments. And although it has no capability to send initialization sequences to a printer, it is easy to write a shell script to do the needed initialization and then execute `lpf`.

In order for `lpf` to do page accounting correctly, it needs correct values filled in for the `pw` and `pl` capabilities in the `/etc/printcap` file. It uses these values to determine how much text can fit on a page and how many pages were in a user's job. For more information on printer accounting, see Accounting for Printer Usage.

9.4.2 Header Pages

If you have *lots* of users, all of them using various printers, then you probably want to consider *header pages* as a necessary evil.

Header pages, also known as *banner* or *burst pages* identify to whom jobs belong after they are printed. They are usually printed in large, bold letters, perhaps with decorative borders, so that in a stack of printouts they stand out from the real documents that comprise users' jobs. They enable users to locate their jobs quickly. The obvious drawback to a header page is that it is yet one more sheet that has to be printed for every job, their ephemeral usefulness lasting not more than a few minutes, ultimately finding themselves in a recycling bin or rubbish heap. (Note that header pages go with each job, not each file in a job, so the paper waste might not be that bad.)

The **LPD** system can provide header pages automatically for your printouts *if* your printer can directly print plain text. If you have a PostScript printer, you will need an external program to generate the header page; see Header Pages on PostScript Printers.

9.4.2.1 Enabling Header Pages

In the Simple Printer Setup section, we turned off header pages by specifying `sh` (meaning "suppress header") in the `/etc/printcap` file. To enable header pages for a printer, just remove the `sh` capability.

Sounds too easy, right?

You are right. You *might* have to provide an output filter to send initialization strings to the printer. Here is an example output filter for Hewlett Packard PCL-compatible printers:

```
#!/bin/sh
#
#  hpof - Output filter for Hewlett Packard PCL-compatible printers
#  Installed in /usr/local/libexec/hpof

printf "\033&k2G" || exit 2
exec /usr/libexec/lpr/lpf
```

Specify the path to the output filter in the `of` capability. See the Output Filters section for more information.

Here is an example `/etc/printcap` file for the printer `teak` that we introduced earlier; we enabled header pages and added the above output filter:

```
#
#  /etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:\
        :of=/usr/local/libexec/hpof:
```

Now, when users print jobs to `teak`, they get a header page with each job. If users want to spend time searching for their printouts, they can suppress header pages by submitting the job with `lpr -h`; see the Header Page Options section for more lpr(1) options.

> **Note: LPD** prints a form feed character after the header page. If your printer uses a different character or sequence of characters to eject a page, specify them with the `ff` capability in `/etc/printcap`.

9.4.2.2 Controlling Header Pages

By enabling header pages, **LPD** will produce a *long header*, a full page of large letters identifying the user, host, and job. Here is an example (kelly printed the job named outline from host `rose`):

```
k                   ll       ll
k                    l        l
k                    l        l
k   k     eeee       l        l     y    y
k  k     e    e      l        l     y    y
k k      eeeeee      l        l     y    y
kk k     e           l        l     y    y
k   k    e    e      l        l     y   yy
k    k    eeee      lll      lll     yyy y
                                        y
                                       y    y
                                        yyyy


                              ll
                    t          l        i
                    t          l
 oooo    u    u   ttttt       l       ii     n nnn     eeee
o    o   u    u     t         l        i     nn   n   e    e
o    o   u    u     t         l        i     n    n   eeeeee
o    o   u    u     t         l        i     n    n   e
o    o   u   uu     t  t      l        i     n    n   e    e
 oooo     uuu u      tt      lll      iii    n    n    eeee
```

```
r rrr     oooo     ssss     eeee
rr   r   o    o   s    s   e    e
r        o    o    ss      eeeeee
r        o    o      ss    e
r        o    o   s    s   e    e
r         oooo     ssss     eeee
```

```
Job:  outline
Date: Sun Sep 17 11:04:58 1995
```

LPD appends a form feed after this text so the job starts on a new page (unless you have `sf` (suppress form feeds) in the destination printer's entry in `/etc/printcap`).

If you prefer, **LPD** can make a *short header*; specify `sb` (short banner) in the `/etc/printcap` file. The header page will look like this:

```
rose:kelly  Job: outline  Date: Sun Sep 17 11:07:51 1995
```

Also by default, **LPD** prints the header page first, then the job. To reverse that, specify `hl` (header last) in `/etc/printcap`.

9.4.2.3 Accounting for Header Pages

Using **LPD**'s built-in header pages enforces a particular paradigm when it comes to printer accounting: header pages must be *free of charge*.

Why?

Because the output filter is the only external program that will have control when the header page is printed that could do accounting, and it is not provided with any *user or host* information or an accounting file, so it has no idea whom to charge for printer use. It is also not enough to just "add one page" to the text filter or any of the conversion filters (which do have user and host information) since users can suppress header pages with `lpr -h`. They could still be charged for header pages they did not print. Basically, `lpr -h` will be the preferred option of environmentally-minded users, but you cannot offer any incentive to use it.

It is *still not enough* to have each of the filters generate their own header pages (thereby being able to charge for them). If users wanted the option of suppressing the header pages with `lpr -h`, they will still get them and be charged for them since **LPD** does not pass any knowledge of the `-h` option to any of the filters.

So, what are your options?

You can:

- Accept **LPD**'s paradigm and make header pages free.
- Install an alternative to **LPD**, such as **LPRng**. Section Alternatives to the Standard Spooler tells more about other spooling software you can substitute for **LPD**.
- Write a *smart* output filter. Normally, an output filter is not meant to do anything more than initialize a printer or do some simple character conversion. It is suited for header pages and plain text jobs (when there is no text (input) filter). But, if there is a text filter for the plain text jobs, then **LPD** will start the output filter only for the header pages. And the output filter can parse the header page text that **LPD** generates to determine what user and host to charge for the header page. The only other problem with this method is that the output filter still does not know what accounting file to use (it is not passed the name of the file from the `af` capability), but if you have a well-known accounting file, you can hard-code that into the output filter. To facilitate the parsing step, use the `sh` (short header) capability in `/etc/printcap`. Then again, all that might be too much trouble, and users will certainly appreciate the more generous system administrator who makes header pages free.

9.4.2.4 Header Pages on PostScript Printers

As described above, **LPD** can generate a plain text header page suitable for many printers. Of course, PostScript cannot directly print plain text, so the header page feature of **LPD** is useless——or mostly so.

One obvious way to get header pages is to have every conversion filter and the text filter generate the header page. The filters should use the user and host arguments to generate a suitable header page. The drawback of this method is that users will always get a header page, even if they submit jobs with `lpr -h`.

Let us explore this method. The following script takes three arguments (user login name, host name, and job name) and makes a simple PostScript header page:

```
#!/bin/sh
#
#  make-ps-header - make a PostScript header page on stdout
#  Installed in /usr/local/libexec/make-ps-header
#

#
#  These are PostScript units (72 to the inch).  Modify for A4 or
#  whatever size paper you are using:
#
page_width=612
page_height=792
border=72

#
#  Check arguments
#
if [ $# -ne 3 ]; then
    echo "Usage: ‘basename $0‘ <user> <host> <job>" 1>&2
    exit 1
fi
```

```
#
#  Save these, mostly for readability in the PostScript, below.
#
user=$1
host=$2
job=$3
date=‘date‘

#
#  Send the PostScript code to stdout.
#
exec cat <<EOF
%!PS

%
%  Make sure we do not interfere with user’s job that will follow
%
save

%
%  Make a thick, unpleasant border around the edge of the paper.
%
$border $border moveto
$page_width $border 2 mul sub 0 rlineto
0 $page_height $border 2 mul sub rlineto
currentscreen 3 -1 roll pop 100 3 1 roll setscreen
$border 2 mul $page_width sub 0 rlineto closepath
0.8 setgray 10 setlinewidth stroke 0 setgray

%
%  Display user’s login name, nice and large and prominent
%
/Helvetica-Bold findfont 64 scalefont setfont
$page_width ($user) stringwidth pop sub 2 div $page_height 200 sub moveto
($user) show

%
%  Now show the boring particulars
%
/Helvetica findfont 14 scalefont setfont
/y 200 def
[ (Job:) (Host:) (Date:) ] {
200 y moveto show /y y 18 sub def }
forall

/Helvetica-Bold findfont 14 scalefont setfont
/y 200 def
[ ($job) ($host) ($date) ] {
        270 y moveto show /y y 18 sub def
} forall

%
% That is it
```

```
%
restore
showpage
EOF
```

Now, each of the conversion filters and the text filter can call this script to first generate the header page, and then print the user's job. Here is the DVI conversion filter from earlier in this document, modified to make a header page:

```
#!/bin/sh
#
#  psdf - DVI to PostScript printer filter
#  Installed in /usr/local/libexec/psdf
#
#  Invoked by lpd when user runs lpr -d
#

orig_args="$@"

fail() {
    echo "$@" 1>&2
    exit 2
}

while getopts "x:y:n:h:" option; do
    case $option in
        x|y)  ;; # Ignore
        n)    login=$OPTARG ;;
        h)    host=$OPTARG ;;
        *)    echo "LPD started ‘basename $0‘ wrong." 1>&2
              exit 2
              ;;
    esac
done

[ "$login" ] || fail "No login name"
[ "$host" ] || fail "No host name"

( /usr/local/libexec/make-ps-header $login $host "DVI File"
  /usr/local/bin/dvips -f ) | eval /usr/local/libexec/lprps $orig_args
```

Notice how the filter has to parse the argument list in order to determine the user and host name. The parsing for the other conversion filters is identical. The text filter takes a slightly different set of arguments, though (see section How Filters Work).

As we have mentioned before, the above scheme, though fairly simple, disables the "suppress header page" option (the `-h` option) to `lpr`. If users wanted to save a tree (or a few pennies, if you charge for header pages), they would not be able to do so, since every filter's going to print a header page with every job.

To allow users to shut off header pages on a per-job basis, you will need to use the trick introduced in section Accounting for Header Pages: write an output filter that parses the LPD-generated header page and produces a PostScript version. If the user submits the job with `lpr -h`, then **LPD** will not generate a header page, and neither will your output filter. Otherwise, your output filter will read the text from **LPD** and send the appropriate header page PostScript code to the printer.

If you have a PostScript printer on a serial line, you can make use of `lprps`, which comes with an output filter, `psof`, which does the above. Note that `psof` does not charge for header pages.

9.4.3 Networked Printing

FreeBSD supports networked printing: sending jobs to remote printers. Networked printing generally refers to two different things:

- Accessing a printer attached to a remote host. You install a printer that has a conventional serial or parallel interface on one host. Then, you set up **LPD** to enable access to the printer from other hosts on the network. Section Printers Installed on Remote Hosts tells how to do this.
- Accessing a printer attached directly to a network. The printer has a network interface in addition (or in place of) a more conventional serial or parallel interface. Such a printer might work as follows:
  - It might understand the **LPD** protocol and can even queue jobs from remote hosts. In this case, it acts just like a regular host running **LPD**. Follow the same procedure in section Printers Installed on Remote Hosts to set up such a printer.
  - It might support a data stream network connection. In this case, you "attach" the printer to one host on the network by making that host responsible for spooling jobs and sending them to the printer. Section Printers with Networked Data Stream Interfaces gives some suggestions on installing such printers.

9.4.3.1 Printers Installed on Remote Hosts

The **LPD** spooling system has built-in support for sending jobs to other hosts also running **LPD** (or are compatible with **LPD**). This feature enables you to install a printer on one host and make it accessible from other hosts. It also works with printers that have network interfaces that understand the **LPD** protocol.

To enable this kind of remote printing, first install a printer on one host, the *printer host*, using the simple printer setup described in the Simple Printer Setup section. Do any advanced setup in Advanced Printer Setup that you need. Make sure to test the printer and see if it works with the features of **LPD** you have enabled. Also ensure that the *local host* has authorization to use the **LPD** service in the *remote host* (see Restricting Jobs from Remote Printers).

If you are using a printer with a network interface that is compatible with **LPD**, then the *printer host* in the discussion below is the printer itself, and the *printer name* is the name you configured for the printer. See the documentation that accompanied your printer and/or printer-network interface.

> **Tip:** If you are using a Hewlett Packard Laserjet then the printer name `text` will automatically perform the LF to CRLF conversion for you, so you will not require the `hpif` script.

Then, on the other hosts you want to have access to the printer, make an entry in their `/etc/printcap` files with the following:

1. Name the entry anything you want. For simplicity, though, you probably want to use the same name and aliases as on the printer host.
2. Leave the `lp` capability blank, explicitly (`:lp=:`).

3. Make a spooling directory and specify its location in the `sd` capability. **LPD** will store jobs here before they get sent to the printer host.
4. Place the name of the printer host in the `rm` capability.
5. Place the printer name on the *printer host* in the `rp` capability.

That is it. You do not need to list conversion filters, page dimensions, or anything else in the `/etc/printcap` file.

Here is an example. The host `rose` has two printers, `bamboo` and `rattan`. We will enable users on the host `orchid` to print to those printers. Here is the `/etc/printcap` file for `orchid` (back from section Enabling Header Pages). It already had the entry for the printer `teak`; we have added entries for the two printers on the host `rose`:

```
#
#  /etc/printcap for host orchid - added (remote) printers on rose
#

#
#  teak is local; it is connected directly to orchid:
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:

#
#  rattan is connected to rose; send jobs for rattan to rose:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

#
#  bamboo is connected to rose as well:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:
```

Then, we just need to make spooling directories on `orchid`:

```
# mkdir -p /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chown daemon:daemon /var/spool/lpd/rattan /var/spool/lpd/bamboo
```

Now, users on `orchid` can print to `rattan` and `bamboo`. If, for example, a user on `orchid` typed

```
% lpr -P bamboo -d sushi-review.dvi
```

the **LPD** system on `orchid` would copy the job to the spooling directory `/var/spool/lpd/bamboo` and note that it was a DVI job. As soon as the host `rose` has room in its `bamboo` spooling directory, the two **LPDs** would transfer the file to `rose`. The file would wait in `rose`'s queue until it was finally printed. It would be converted from DVI to PostScript (since `bamboo` is a PostScript printer) on `rose`.

9.4.3.2 Printers with Networked Data Stream Interfaces

Often, when you buy a network interface card for a printer, you can get two versions: one which emulates a spooler (the more expensive version), or one which just lets you send data to it as if you were using a serial or parallel port (the cheaper version). This section tells how to use the cheaper version. For the more expensive one, see the previous section Printers Installed on Remote Hosts.

The format of the `/etc/printcap` file lets you specify what serial or parallel interface to use, and (if you are using a serial interface), what baud rate, whether to use flow control, delays for tabs, conversion of newlines, and more. But there is no way to specify a connection to a printer that is listening on a TCP/IP or other network port.

To send data to a networked printer, you need to develop a communications program that can be called by the text and conversion filters. Here is one such example: the script `netprint` takes all data on standard input and sends it to a network-attached printer. We specify the hostname of the printer as the first argument and the port number to which to connect as the second argument to `netprint`. Note that this supports one-way communication only (FreeBSD to printer); many network printers support two-way communication, and you might want to take advantage of that (to get printer status, perform accounting, etc.).

```
#!/usr/bin/perl
#
#  netprint - Text filter for printer attached to network
#  Installed in /usr/local/libexec/netprint
#
$#ARGV eq 1 || die "Usage: $0 <printer-hostname> <port-number>";

$printer_host = $ARGV[0];
$printer_port = $ARGV[1];

require 'sys/socket.ph';

($ignore, $ignore, $protocol) = getprotobyname('tcp');
($ignore, $ignore, $ignore, $ignore, $address)
    = gethostbyname($printer_host);

$sockaddr = pack('S n a4 x8', &AF_INET, $printer_port, $address);

socket(PRINTER, &PF_INET, &SOCK_STREAM, $protocol)
    || die "Can't create TCP/IP stream socket: $!";
connect(PRINTER, $sockaddr) || die "Can't contact $printer_host: $!";
while (<STDIN>) { print PRINTER; }
exit 0;
```

We can then use this script in various filters. Suppose we had a Diablo 750-N line printer connected to the network. The printer accepts data to print on port number 5100. The host name of the printer is scrivener. Here is the text filter for the printer:

```
#!/bin/sh
#
#  diablo-if-net - Text filter for Diablo printer 'scrivener' listening
#  on port 5100.   Installed in /usr/local/libexec/diablo-if-net
#
exec /usr/libexec/lpr/lpf "$@" | /usr/local/libexec/netprint scrivener 5100
```

9.4.4 Restricting Printer Usage

This section gives information on restricting printer usage. The **LPD** system lets you control who can access a printer, both locally or remotely, whether they can print multiple copies, how large their jobs can be, and how large the printer queues can get.

9.4.4.1 Restricting Multiple Copies

The **LPD** system makes it easy for users to print multiple copies of a file. Users can print jobs with `lpr -#5` (for example) and get five copies of each file in the job. Whether this is a good thing is up to you.

If you feel multiple copies cause unnecessary wear and tear on your printers, you can disable the `-#` option to lpr(1) by adding the `sc` capability to the `/etc/printcap` file. When users submit jobs with the `-#` option, they will see:

```
lpr: multiple copies are not allowed
```

Note that if you have set up access to a printer remotely (see section Printers Installed on Remote Hosts), you need the `sc` capability on the remote `/etc/printcap` files as well, or else users will still be able to submit multiple-copy jobs by using another host.

Here is an example. This is the `/etc/printcap` file for the host `rose`. The printer `rattan` is quite hearty, so we will allow multiple copies, but the laser printer `bamboo` is a bit more delicate, so we will disable multiple copies by adding the `sc` capability:

```
#
#  /etc/printcap for host rose - restrict multiple copies on bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Now, we also need to add the `sc` capability on the host `orchid`'s `/etc/printcap` (and while we are at it, let us disable multiple copies for the printer `teak`):

```
#
#  /etc/printcap for host orchid - no multiple copies for local
#  printer teak or remote printer bamboo
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:sc:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:

rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
```

```
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:sc:
```

By using the `sc` capability, we prevent the use of `lpr -#`, but that still does not prevent users from running lpr(1) multiple times, or from submitting the same file multiple times in one job like this:

```
% lpr forsale.sign forsale.sign forsale.sign forsale.sign forsale.sign
```

There are many ways to prevent this abuse (including ignoring it) which you are free to explore.

9.4.4.2 Restricting Access to Printers

You can control who can print to what printers by using the UNIX group mechanism and the `rg` capability in `/etc/printcap`. Just place the users you want to have access to a printer in a certain group, and then name that group in the `rg` capability.

Users outside the group (including `root`) will be greeted with "`lpr: Not a member of the restricted group`" if they try to print to the controlled printer.

As with the `sc` (suppress multiple copies) capability, you need to specify `rg` on remote hosts that also have access to your printers, if you feel it is appropriate (see section Printers Installed on Remote Hosts).

For example, we will let anyone access the printer `rattan`, but only those in group `artists` can use `bamboo`. Here is the familiar `/etc/printcap` for host `rose`:

```
#
#  /etc/printcap for host rose - restricted group for bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Let us leave the other example `/etc/printcap` file (for the host `orchid`) alone. Of course, anyone on `orchid` can print to `bamboo`. It might be the case that we only allow certain logins on `orchid` anyway, and want them to have access to the printer. Or not.

> **Note:** There can be only one restricted group per printer.

9.4.4.3 Controlling Sizes of Jobs Submitted

If you have many users accessing the printers, you probably need to put an upper limit on the sizes of the files users can submit to print. After all, there is only so much free space on the filesystem that houses the spooling directories, and you also need to make sure there is room for the jobs of other users.

LPD enables you to limit the maximum byte size a file in a job can be with the `mx` capability. The units are in `BUFSIZ` blocks, which are 1024 bytes. If you put a zero for this capability, there will be no limit on file size; however, if no `mx` capability is specified, then a default limit of 1000 blocks will be used.

> **Note:** The limit applies to *files* in a job, and *not* the total job size.

LPD will not refuse a file that is larger than the limit you place on a printer. Instead, it will queue as much of the file up to the limit, which will then get printed. The rest will be discarded. Whether this is correct behavior is up for debate.

Let us add limits to our example printers `rattan` and `bamboo`. Since those artists' PostScript files tend to be large, we will limit them to five megabytes. We will put no limit on the plain text line printer:

```
#
#  /etc/printcap for host rose
#

#
#  No limit on job size:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:mx#0:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

#
#  Limit of five megabytes:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Again, the limits apply to the local users only. If you have set up access to your printers remotely, remote users will not get those limits. You will need to specify the `mx` capability in the remote `/etc/printcap` files as well. See section Printers Installed on Remote Hosts for more information on remote printing.

There is another specialized way to limit job sizes from remote printers; see section Restricting Jobs from Remote Printers.

9.4.4.4 Restricting Jobs from Remote Printers

The **LPD** spooling system provides several ways to restrict print jobs submitted from remote hosts:

Host restrictions

> You can control from which remote hosts a local **LPD** accepts requests with the files `/etc/hosts.equiv` and `/etc/hosts.lpd`. **LPD** checks to see if an incoming request is from a host listed in either one of these files. If not, **LPD** refuses the request.

The format of these files is simple: one host name per line. Note that the file `/etc/hosts.equiv` is also used by the ruserok(3) protocol, and affects programs like rsh(1) and rcp(1), so be careful.

For example, here is the `/etc/hosts.lpd` file on the host `rose`:

```
orchid
violet
madrigal.fishbaum.de
```

This means `rose` will accept requests from the hosts `orchid`, `violet`, and `madrigal.fishbaum.de`. If any other host tries to access `rose`'s **LPD**, the job will be refused.

Size restrictions

You can control how much free space there needs to remain on the filesystem where a spooling directory resides. Make a file called `minfree` in the spooling directory for the local printer. Insert in that file a number representing how many disk blocks (512 bytes) of free space there has to be for a remote job to be accepted.

This lets you insure that remote users will not fill your filesystem. You can also use it to give a certain priority to local users: they will be able to queue jobs long after the free disk space has fallen below the amount specified in the `minfree` file.

For example, let us add a `minfree` file for the printer `bamboo`. We examine `/etc/printcap` to find the spooling directory for this printer; here is `bamboo`'s entry:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:mx#5000:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

The spooling directory is given in the `sd` capability. We will make three megabytes (which is 6144 disk blocks) the amount of free disk space that must exist on the filesystem for **LPD** to accept remote jobs:

```
# echo 6144 > /var/spool/lpd/bamboo/minfree
```

User restrictions

You can control which remote users can print to local printers by specifying the `rs` capability in `/etc/printcap`. When `rs` appears in the entry for a locally-attached printer, **LPD** will accept jobs from remote hosts *if* the user submitting the job also has an account of the same login name on the local host. Otherwise, **LPD** refuses the job.

This capability is particularly useful in an environment where there are (for example) different departments sharing a network, and some users transcend departmental boundaries. By giving them accounts on your systems, they can use your printers from their own departmental systems. If you would rather allow them to use *only* your printers and not your computer resources, you can give them "token" accounts, with no home directory and a useless shell like `/usr/bin/false`.

9.4.5 Accounting for Printer Usage

So, you need to charge for printouts. And why not? Paper and ink cost money. And then there are maintenance costs——printers are loaded with moving parts and tend to break down. You have examined your printers, usage

patterns, and maintenance fees and have come up with a per-page (or per-foot, per-meter, or per-whatever) cost. Now, how do you actually start accounting for printouts?

Well, the bad news is the **LPD** spooling system does not provide much help in this department. Accounting is highly dependent on the kind of printer in use, the formats being printed, and *your* requirements in charging for printer usage.

To implement accounting, you have to modify a printer's text filter (to charge for plain text jobs) and the conversion filters (to charge for other file formats), to count pages or query the printer for pages printed. You cannot get away with using the simple output filter, since it cannot do accounting. See section Filters.

Generally, there are two ways to do accounting:

- *Periodic accounting* is the more common way, possibly because it is easier. Whenever someone prints a job, the filter logs the user, host, and number of pages to an accounting file. Every month, semester, year, or whatever time period you prefer, you collect the accounting files for the various printers, tally up the pages printed by users, and charge for usage. Then you truncate all the logging files, starting with a clean slate for the next period.
- *Timely accounting* is less common, probably because it is more difficult. This method has the filters charge users for printouts as soon as they use the printers. Like disk quotas, the accounting is immediate. You can prevent users from printing when their account goes in the red, and might provide a way for users to check and adjust their "print quotas." But this method requires some database code to track users and their quotas.

The **LPD** spooling system supports both methods easily: since you have to provide the filters (well, most of the time), you also have to provide the accounting code. But there is a bright side: you have enormous flexibility in your accounting methods. For example, you choose whether to use periodic or timely accounting. You choose what information to log: user names, host names, job types, pages printed, square footage of paper used, how long the job took to print, and so forth. And you do so by modifying the filters to save this information.

9.4.5.1 Quick and Dirty Printer Accounting

FreeBSD comes with two programs that can get you set up with simple periodic accounting right away. They are the text filter `lpf`, described in section lpf: a Text Filter, and pac(8), a program to gather and total entries from printer accounting files.

As mentioned in the section on filters (Filters), **LPD** starts the text and the conversion filters with the name of the accounting file to use on the filter command line. The filters can use this argument to know where to write an accounting file entry. The name of this file comes from the `af` capability in `/etc/printcap`, and if not specified as an absolute path, is relative to the spooling directory.

LPD starts `lpf` with page width and length arguments (from the `pw` and `pl` capabilities). `lpf` uses these arguments to determine how much paper will be used. After sending the file to the printer, it then writes an accounting entry in the accounting file. The entries look like this:

```
2.00 rose:andy
3.00 rose:kelly
3.00 orchid:mary
5.00 orchid:mary
2.00 orchid:zhang
```

You should use a separate accounting file for each printer, as `lpf` has no file locking logic built into it, and two `lpf`s might corrupt each other's entries if they were to write to the same file at the same time. An easy way to insure a

separate accounting file for each printer is to use `af=acct` in `/etc/printcap`. Then, each accounting file will be in the spooling directory for a printer, in a file named `acct`.

When you are ready to charge users for printouts, run the pac(8) program. Just change to the spooling directory for the printer you want to collect on and type `pac`. You will get a dollar-centric summary like the following:

```
  Login               pages/feet   runs    price
orchid:kelly               5.00    1   $  0.10
orchid:mary               31.00    3   $  0.62
orchid:zhang               9.00    1   $  0.18
rose:andy                  2.00    1   $  0.04
rose:kelly               177.00  104   $  3.54
rose:mary                 87.00   32   $  1.74
rose:root                 26.00   12   $  0.52

total                    337.00  154   $  6.74
```

These are the arguments pac(8) expects:

`-P`*`printer`*

Which *`printer`* to summarize. This option works only if there is an absolute path in the `af` capability in `/etc/printcap`.

`-c`

Sort the output by cost instead of alphabetically by user name.

`-m`

Ignore host name in the accounting files. With this option, user `smith` on host `alpha` is the same user `smith` on host `gamma`. Without, they are different users.

`-p`*`price`*

Compute charges with *`price`* dollars per page or per foot instead of the price from the `pc` capability in `/etc/printcap`, or two cents (the default). You can specify *`price`* as a floating point number.

`-r`

Reverse the sort order.

`-s`

Make an accounting summary file and truncate the accounting file.

`name` `...`

Print accounting information for the given user *`names`* only.

In the default summary that pac(8) produces, you see the number of pages printed by each user from various hosts. If, at your site, host does not matter (because users can use any host), run `pac -m`, to produce the following summary:

```
  Login               pages/feet   runs    price
andy                       2.00    1   $  0.04
kelly                    182.00  105   $  3.64
mary                     118.00   35   $  2.36
```

```
root                        26.00   12   $  0.52
zhang                        9.00    1   $  0.18

total                      337.00  154   $  6.74
```

To compute the dollar amount due, pac(8) uses the `pc` capability in the `/etc/printcap` file (default of 200, or 2 cents per page). Specify, in hundredths of cents, the price per page or per foot you want to charge for printouts in this capability. You can override this value when you run pac(8) with the `-p` option. The units for the `-p` option are in dollars, though, not hundredths of cents. For example,

```
# pac -p1.50
```

makes each page cost one dollar and fifty cents. You can really rake in the profits by using this option.

Finally, running `pac -s` will save the summary information in a summary accounting file, which is named the same as the printer's accounting file, but with `_sum` appended to the name. It then truncates the accounting file. When you run pac(8) again, it rereads the summary file to get starting totals, then adds information from the regular accounting file.

9.4.5.2 How Can You Count Pages Printed?

In order to perform even remotely accurate accounting, you need to be able to determine how much paper a job uses. This is the essential problem of printer accounting.

For plain text jobs, the problem is not that hard to solve: you count how many lines are in a job and compare it to how many lines per page your printer supports. Do not forget to take into account backspaces in the file which overprint lines, or long logical lines that wrap onto one or more additional physical lines.

The text filter `lpf` (introduced in lpf: a Text Filter) takes into account these things when it does accounting. If you are writing a text filter which needs to do accounting, you might want to examine `lpf`'s source code.

How do you handle other file formats, though?

Well, for DVI-to-LaserJet or DVI-to-PostScript conversion, you can have your filter parse the diagnostic output of `dvilj` or `dvips` and look to see how many pages were converted. You might be able to do similar things with other file formats and conversion programs.

But these methods suffer from the fact that the printer may not actually print all those pages. For example, it could jam, run out of toner, or explode——and the user would still get charged.

So, what can you do?

There is only one *sure* way to do *accurate* accounting. Get a printer that can tell you how much paper it uses, and attach it via a serial line or a network connection. Nearly all PostScript printers support this notion. Other makes and models do as well (networked Imagen laser printers, for example). Modify the filters for these printers to get the page usage after they print each job and have them log accounting information based on that value *only*. There is no line counting nor error-prone file examination required.

Of course, you can always be generous and make all printouts free.

9.5 Using Printers

This section tells you how to use printers you have set up with FreeBSD. Here is an overview of the user-level commands:

lpr(1)

Print jobs

lpq(1)

Check printer queues

lprm(1)

Remove jobs from a printer's queue

There is also an administrative command, lpc(8), described in the section Administering Printers, used to control printers and their queues.

All three of the commands lpr(1), lprm(1), and lpq(1) accept an option `-P` *`printer-name`* to specify on which printer/queue to operate, as listed in the `/etc/printcap` file. This enables you to submit, remove, and check on jobs for various printers. If you do not use the `-P` option, then these commands use the printer specified in the `PRINTER` environment variable. Finally, if you do not have a `PRINTER` environment variable, these commands default to the printer named `lp`.

Hereafter, the terminology *default printer* means the printer named in the `PRINTER` environment variable, or the printer named `lp` when there is no `PRINTER` environment variable.

9.5.1 Printing Jobs

To print files, type:

```
% lpr filename ...
```

This prints each of the listed files to the default printer. If you list no files, lpr(1) reads data to print from standard input. For example, this command prints some important system files:

```
% lpr /etc/host.conf /etc/hosts.equiv
```

To select a specific printer, type:

```
% lpr -P printer-name filename ...
```

This example prints a long listing of the current directory to the printer named `rattan`:

```
% ls -l | lpr -P rattan
```

Because no files were listed for the lpr(1) command, `lpr` read the data to print from standard input, which was the output of the `ls -l` command.

The lpr(1) command can also accept a wide variety of options to control formatting, apply file conversions, generate multiple copies, and so forth. For more information, see the section Printing Options.

9.5.2 Checking Jobs

When you print with lpr(1), the data you wish to print is put together in a package called a "print job", which is sent to the **LPD** spooling system. Each printer has a queue of jobs, and your job waits in that queue along with other jobs from yourself and from other users. The printer prints those jobs in a first-come, first-served order.

To display the queue for the default printer, type lpq(1). For a specific printer, use the `-P` option. For example, the command

```
% lpq -P bamboo
```

shows the queue for the printer named `bamboo`. Here is an example of the output of the `lpq` command:

```
bamboo is ready and printing
Rank   Owner    Job  Files                              Total Size
active kelly    9    /etc/host.conf, /etc/hosts.equiv   88 bytes
2nd    kelly    10   (standard input)                   1635 bytes
3rd    mary     11   ...                                78519 bytes
```

This shows three jobs in the queue for `bamboo`. The first job, submitted by user kelly, got assigned "job number" 9. Every job for a printer gets a unique job number. Most of the time you can ignore the job number, but you will need it if you want to cancel the job; see section Removing Jobs for details.

Job number nine consists of two files; multiple files given on the lpr(1) command line are treated as part of a single job. It is the currently active job (note the word `active` under the "Rank" column), which means the printer should be currently printing that job. The second job consists of data passed as the standard input to the lpr(1) command. The third job came from user `mary`; it is a much larger job. The pathname of the file she is trying to print is too long to fit, so the lpq(1) command just shows three dots.

The very first line of the output from lpq(1) is also useful: it tells what the printer is currently doing (or at least what **LPD** thinks the printer is doing).

The lpq(1) command also support a `-l` option to generate a detailed long listing. Here is an example of `lpq -l`:

```
waiting for bamboo to become ready (offline ?)
kelly: 1st     [job 009rose]
       /etc/host.conf                   73 bytes
       /etc/hosts.equiv                 15 bytes

kelly: 2nd     [job 010rose]
       (standard input)                 1635 bytes

mary: 3rd                               [job 011rose]
      /home/orchid/mary/research/venus/alpha-regio/mapping 78519 bytes
```

9.5.3 Removing Jobs

If you change your mind about printing a job, you can remove the job from the queue with the lprm(1) command. Often, you can even use lprm(1) to remove an active job, but some or all of the job might still get printed.

To remove a job from the default printer, first use lpq(1) to find the job number. Then type:

```
% lprm job-number
```

To remove the job from a specific printer, add the `-P` option. The following command removes job number 10 from the queue for the printer `bamboo`:

```
% lprm -P bamboo 10
```

The lprm(1) command has a few shortcuts:

lprm -

Removes all jobs (for the default printer) belonging to you.

lprm *user*

Removes all jobs (for the default printer) belonging to *user*. The superuser can remove other users' jobs; you can remove only your own jobs.

lprm

With no job number, user name, or `-` appearing on the command line, lprm(1) removes the currently active job on the default printer, if it belongs to you. The superuser can remove any active job.

Just use the `-P` option with the above shortcuts to operate on a specific printer instead of the default. For example, the following command removes all jobs for the current user in the queue for the printer named `rattan`:

```
% lprm -P rattan -
```

Note: If you are working in a networked environment, lprm(1) will let you remove jobs only from the host from which the jobs were submitted, even if the same printer is available from other hosts. The following command sequence demonstrates this:

```
% lpr -P rattan myfile
% rlogin orchid
% lpq -P rattan
Rank   Owner   Job  Files                          Total Size
active seeyan   12 ...                           49123 bytes
2nd    kelly      13   myfile                          12 bytes
% lprm -P rattan 13
rose: Permission denied
% logout
% lprm -P rattan 13
dfA013rose dequeued
cfA013rose dequeued
```

9.5.4 Beyond Plain Text: Printing Options

The lpr(1) command supports a number of options that control formatting text, converting graphic and other file formats, producing multiple copies, handling of the job, and more. This section describes the options.

9.5.4.1 Formatting and Conversion Options

The following lpr(1) options control formatting of the files in the job. Use these options if the job does not contain plain text or if you want plain text formatted through the pr(1) utility.

For example, the following command prints a DVI file (from the TEX typesetting system) named `fish-report.dvi` to the printer named `bamboo`:

```
% lpr -P bamboo -d fish-report.dvi
```

These options apply to every file in the job, so you cannot mix (say) DVI and ditroff files together in a job. Instead, submit the files as separate jobs, using a different conversion option for each job.

> **Note:** All of these options except `-p` and `-T` require conversion filters installed for the destination printer. For example, the `-d` option requires the DVI conversion filter. Section Conversion Filters gives details.

`-c`

Print cifplot files.

`-d`

Print DVI files.

`-f`

Print FORTRAN text files.

`-g`

Print plot data.

`-i` *`number`*

Indent the output by *`number`* columns; if you omit *`number`*, indent by 8 columns. This option works only with certain conversion filters.

> **Note:** Do not put any space between the `-i` and the number.

`-l`

Print literal text data, including control characters.

`-n`

Print ditroff (device independent troff) data.

`-p`

Format plain text with pr(1) before printing. See pr(1) for more information.

`-T title`

Use *`title`* on the pr(1) header instead of the file name. This option has effect only when used with the `-p` option.

`-t`

Print troff data.

`-v`

Print raster data.

Here is an example: this command prints a nicely formatted version of the ls(1) manual page on the default printer:

```
% zcat /usr/share/man/man1/ls.1.gz | troff -t -man | lpr -t
```

The zcat(1) command uncompresses the source of the ls(1) manual page and passes it to the troff(1) command, which formats that source and makes GNU troff output and passes it to lpr(1), which submits the job to the **LPD** spooler. Because we used the `-t` option to lpr(1), the spooler will convert the GNU troff output into a format the default printer can understand when it prints the job.

9.5.4.2 Job Handling Options

The following options to lpr(1) tell **LPD** to handle the job specially:

-# `copies`

Produce a number of *`copies`* of each file in the job instead of just one copy. An administrator may disable this option to reduce printer wear-and-tear and encourage photocopier usage. See section Restricting Multiple Copies.

This example prints three copies of `parser.c` followed by three copies of `parser.h` to the default printer:

```
% lpr -#3 parser.c parser.h
```

-m

Send mail after completing the print job. With this option, the **LPD** system will send mail to your account when it finishes handling your job. In its message, it will tell you if the job completed successfully or if there was an error, and (often) what the error was.

-s

Do not copy the files to the spooling directory, but make symbolic links to them instead.

If you are printing a large job, you probably want to use this option. It saves space in the spooling directory (your job might overflow the free space on the filesystem where the spooling directory resides). It saves time as well since **LPD** will not have to copy each and every byte of your job to the spooling directory.

There is a drawback, though: since **LPD** will refer to the original files directly, you cannot modify or remove them until they have been printed.

> **Note:** If you are printing to a remote printer, **LPD** will eventually have to copy files from the local host to the remote host, so the `-s` option will save space only on the local spooling directory, not the remote. It is still useful, though.

-r

Remove the files in the job after copying them to the spooling directory, or after printing them with the `-s` option. Be careful with this option!

9.5.4.3 Header Page Options

These options to lpr(1) adjust the text that normally appears on a job's header page. If header pages are suppressed for the destination printer, these options have no effect. See section Header Pages for information about setting up header pages.

-C *text*

Replace the hostname on the header page with *text*. The hostname is normally the name of the host from which the job was submitted.

-J *text*

Replace the job name on the header page with *text*. The job name is normally the name of the first file of the job, or `stdin` if you are printing standard input.

-h

Do not print any header page.

> **Note:** At some sites, this option may have no effect due to the way header pages are generated. See Header Pages for details.

9.5.5 Administering Printers

As an administrator for your printers, you have had to install, set up, and test them. Using the lpc(8) command, you can interact with your printers in yet more ways. With lpc(8), you can

- Start and stop the printers
- Enable and disable their queues
- Rearrange the order of the jobs in each queue.

First, a note about terminology: if a printer is *stopped*, it will not print anything in its queue. Users can still submit jobs, which will wait in the queue until the printer is *started* or the queue is cleared.

If a queue is *disabled*, no user (except `root`) can submit jobs for the printer. An *enabled* queue allows jobs to be submitted. A printer can be *started* for a disabled queue, in which case it will continue to print jobs in the queue until the queue is empty.

In general, you have to have `root` privileges to use the lpc(8) command. Ordinary users can use the lpc(8) command to get printer status and to restart a hung printer only.

Here is a summary of the lpc(8) commands. Most of the commands take a *`printer-name`* argument to tell on which printer to operate. You can use `all` for the *`printer-name`* to mean all printers listed in `/etc/printcap`.

`abort` *`printer-name`*

Cancel the current job and stop the printer. Users can still submit jobs if the queue is enabled.

`clean` *`printer-name`*

Remove old files from the printer's spooling directory. Occasionally, the files that make up a job are not properly removed by **LPD**, particularly if there have been errors during printing or a lot of administrative activity. This command finds files that do not belong in the spooling directory and removes them.

`disable` *`printer-name`*

Disable queuing of new jobs. If the printer is running, it will continue to print any jobs remaining in the queue. The superuser (`root`) can always submit jobs, even to a disabled queue.

This command is useful while you are testing a new printer or filter installation: disable the queue and submit jobs as `root`. Other users will not be able to submit jobs until you complete your testing and re-enable the queue with the `enable` command.

`down` *`printer-name message`*

Take a printer down. Equivalent to `disable` followed by `stop`. The *`message`* appears as the printer's status whenever a user checks the printer's queue with lpq(1) or status with `lpc status`.

`enable` *`printer-name`*

Enable the queue for a printer. Users can submit jobs but the printer will not print anything until it is started.

`help` *`command-name`*

Print help on the command *`command-name`*. With no *`command-name`*, print a summary of the commands available.

`restart` *`printer-name`*

Start the printer. Ordinary users can use this command if some extraordinary circumstance hangs **LPD**, but they cannot start a printer stopped with either the `stop` or `down` commands. The `restart` command is equivalent to `abort` followed by `start`.

`start` *`printer-name`*

Start the printer. The printer will print jobs in its queue.

`stop` *`printer-name`*

Stop the printer. The printer will finish the current job and will not print anything else in its queue. Even though the printer is stopped, users can still submit jobs to an enabled queue.

`topq` *`printer-name job-or-username`*

Rearrange the queue for *`printer-name`* by placing the jobs with the listed *`job`* numbers or the jobs belonging to *`username`* at the top of the queue. For this command, you cannot use `all` as the *`printer-name`*.

`up` *`printer-name`*

Bring a printer up; the opposite of the `down` command. Equivalent to `start` followed by `enable`.

lpc(8) accepts the above commands on the command line. If you do not enter any commands, lpc(8) enters an interactive mode, where you can enter commands until you type `exit`, `quit`, or end-of-file.

9.6 Alternatives to the Standard Spooler

If you have been reading straight through this manual, by now you have learned just about everything there is to know about the **LPD** spooling system that comes with FreeBSD. You can probably appreciate many of its shortcomings, which naturally leads to the question: "What other spooling systems are out there (and work with FreeBSD)?"

LPRng

LPRng, which purportedly means "LPR: the Next Generation" is a complete rewrite of PLP. Patrick Powell and Justin Mason (the principal maintainer of PLP) collaborated to make **LPRng**. The main site for **LPRng** is http://www.lprng.org/.

CUPS

CUPS, the Common UNIX Printing System, provides a portable printing layer for UNIX-based operating systems. It has been developed by Easy Software Products to promote a standard printing solution for all UNIX vendors and users.

CUPS uses the Internet Printing Protocol (IPP) as the basis for managing print jobs and queues. The Line Printer Daemon (LPD), Server Message Block (SMB), and AppSocket (a.k.a. JetDirect) protocols are also supported with reduced functionality. CUPS adds network printer browsing and PostScript Printer Description (PPD) based printing options to support real-world printing under UNIX.

The main site for **CUPS** is http://www.cups.org/.

9.7 Troubleshooting

After performing the simple test with lptest(1), you might have gotten one of the following results instead of the correct printout:

It worked, after awhile; or, it did not eject a full sheet.

The printer printed the above, but it sat for awhile and did nothing. In fact, you might have needed to press a PRINT REMAINING or FORM FEED button on the printer to get any results to appear.

If this is the case, the printer was probably waiting to see if there was any more data for your job before it printed anything. To fix this problem, you can have the text filter send a FORM FEED character (or whatever is necessary) to the printer. This is usually sufficient to have the printer immediately print any text remaining in its

internal buffer. It is also useful to make sure each print job ends on a full sheet, so the next job does not start somewhere on the middle of the last page of the previous job.

The following replacement for the shell script `/usr/local/libexec/if-simple` prints a form feed after it sends the job to the printer:

```
#!/bin/sh
#
# if-simple - Simple text input filter for lpd
# Installed in /usr/local/libexec/if-simple
#
# Simply copies stdin to stdout.  Ignores all filter arguments.
# Writes a form feed character (\f) after printing job.

/bin/cat && printf "\f" && exit 0
exit 2
```

It produced the "staircase effect."

You got the following on paper:

```
!"#$%&'()*+,-./01234
               "#$%&'()*+,-./012345
                           #$%&'()*+,-./0123456
```

You have become another victim of the *staircase effect*, caused by conflicting interpretations of what characters should indicate a new line. UNIX style operating systems use a single character: ASCII code 10, the line feed (LF). MS-DOS, OS/2®, and others uses a pair of characters, ASCII code 10 *and* ASCII code 13 (the carriage return or CR). Many printers use the MS-DOS convention for representing new-lines.

When you print with FreeBSD, your text used just the line feed character. The printer, upon seeing a line feed character, advanced the paper one line, but maintained the same horizontal position on the page for the next character to print. That is what the carriage return is for: to move the location of the next character to print to the left edge of the paper.

Here is what FreeBSD wants your printer to do:

Printer received CR	Printer prints CR
Printer received LF	Printer prints CR + LF

Here are some ways to achieve this:

- Use the printer's configuration switches or control panel to alter its interpretation of these characters. Check your printer's manual to find out how to do this.

  Note: If you boot your system into other operating systems besides FreeBSD, you may have to *reconfigure* the printer to use a an interpretation for CR and LF characters that those other operating systems use. You might prefer one of the other solutions, below.

- Have FreeBSD's serial line driver automatically convert LF to CR+LF. Of course, this works with printers on serial ports *only*. To enable this feature, use the `ms#` capability and set the `onlcr` mode in the `/etc/printcap` file for the printer.

- Send an *escape code* to the printer to have it temporarily treat LF characters differently. Consult your printer's manual for escape codes that your printer might support. When you find the proper escape code, modify the text filter to send the code first, then send the print job.

  Here is an example text filter for printers that understand the Hewlett-Packard PCL escape codes. This filter makes the printer treat LF characters as a LF and CR; then it sends the job; then it sends a form feed to eject the last page of the job. It should work with nearly all Hewlett Packard printers.

  ```
  #!/bin/sh
  #
  # hpif - Simple text input filter for lpd for HP-PCL based printers
  # Installed in /usr/local/libexec/hpif
  #
  # Simply copies stdin to stdout.  Ignores all filter arguments.
  # Tells printer to treat LF as CR+LF.  Ejects the page when done.

  printf "\033&k2G" && cat && printf "\033&l0H" && exit 0
  exit 2
  ```

  Here is an example `/etc/printcap` from a host called `orchid`. It has a single printer attached to its first parallel port, a Hewlett Packard LaserJet 3Si named `teak`. It is using the above script as its text filter:

  ```
  #
  #  /etc/printcap for host orchid
  #
  teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
          :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
          :if=/usr/local/libexec/hpif:
  ```

It overprinted each line.

The printer never advanced a line. All of the lines of text were printed on top of each other on one line.

This problem is the "opposite" of the staircase effect, described above, and is much rarer. Somewhere, the LF characters that FreeBSD uses to end a line are being treated as CR characters to return the print location to the left edge of the paper, but not also down a line.

Use the printer's configuration switches or control panel to enforce the following interpretation of LF and CR characters:

Printer receives	Printer prints
CR	CR
LF	CR + LF

The printer lost characters.

While printing, the printer did not print a few characters in each line. The problem might have gotten worse as the printer ran, losing more and more characters.

The problem is that the printer cannot keep up with the speed at which the computer sends data over a serial line (this problem should not occur with printers on parallel ports). There are two ways to overcome the problem:

- If the printer supports XON/XOFF flow control, have FreeBSD use it by specifying the `ixon` mode in the

`ms#` capability.

- If the printer supports carrier flow control, specify the `crtscts` mode in the `ms#` capability. Make sure the cable connecting the printer to the computer is correctly wired for carrier flow control.

It printed garbage.

The printer printed what appeared to be random garbage, but not the desired text.

This is usually another symptom of incorrect communications parameters with a serial printer. Double-check the bps rate in the `br` capability, and the parity setting in the `ms#` capability; make sure the printer is using the same settings as specified in the `/etc/printcap` file.

Nothing happened.

If nothing happened, the problem is probably within FreeBSD and not the hardware. Add the log file (`lf`) capability to the entry for the printer you are debugging in the `/etc/printcap` file. For example, here is the entry for `rattan`, with the `lf` capability:

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:\
        :lf=/var/log/rattan.log
```

Then, try printing again. Check the log file (in our example, `/var/log/rattan.log`) to see any error messages that might appear. Based on the messages you see, try to correct the problem.

If you do not specify a `lf` capability, **LPD** uses `/dev/console` as a default.

Chapter 10 與Linux Binary 的相容方面

Restructured and parts updated by Jim Mock. Originally contributed by Brian N. Handy and Rich Murphey.

10.1 概述

FreeBSD 有提供其他幾種UNIX like 作業系統的binary 相容性，其中包括了Linux。你可能會納悶：為什麼FreeBSD 需要能夠執行Linux 專用執行檔(binary)呢？答案很簡單，許多公司、開發者只會Linux 開發程式，因為這是目前資訊界“最熱門” 的玩意。這逼得許多FreeBSD 使用者不得不去勸說這些人是否提供可直接在FreeBSD 上執行的版本。但問題是，大多數公司並不瞭解會有多少人會用FreeBSD 版，因此他們仍只開發Linux 版。那麼FreeBSD 使用者該怎麼辦呢？答案就是用FreeBSD 所提供的Linux binary 相容。

簡單來講，這種相容性可讓FreeBSD 使用者直接執行約90% 的Linux 程式，而不必做任何修改。這些包括了：**StarOffice**、**Netscape** 的Linux
版、**Adobe Acrobat**、**RealPlayer**、**VMware**™、**Oracle**、**WordPerfect®**、**Doom**、**Quake** 等等。此外，也有人回報說在某些情況下，這些在FreeBSD 上執行的Linux 程式，甚至比原本在Linux 執行得更好。

然而呢，還是有些只限Linux 特定的作業系統功能，在FreeBSD 上並未支援。如果Linux 程式過於濫用只有i386 架構上才能用的功能，比如：虛擬8086 模式，則可能無法在FreeBSD 運作正常。

讀完這章，您將了解：

- 如何啓用Linux 相容模式。
- 如何安裝額外的Linux share libraries。
- 如何在FreeBSD 上安裝Linux 程式。
- FreeBSD 上的Linux 相容模式的實作細節。

在閱讀這章之前，您應當了解：

- 知道如何透過port 機制來安裝軟體(Chapter 4)。

10.2 安裝

預設並不會打開Linux 相容模式，最簡單的啓用方式，就是載入`linux` KLD object (“Kernel LoaDable object”)。載入方式，請切為`root` 權限，然後打下列指令：

```
# kldload linux
```

若要每次開機都啓用的話，請把下列內容加到`/etc/rc.conf` 檔：

```
linux_enable="YES"
```

另外可以用kldstat(8) 指令，來確認有哪些KLD 有載入：

```
% kldstat
Id Refs Address    Size     Name
 1    2 0xc0100000 16bdb8   kernel
 7    1 0xc24db000 d000     linux.ko
```

If for some reason you do not want to or cannot load the KLD, then you may statically link Linux binary compatibility into the kernel by adding `options COMPAT_LINUX` to your kernel configuration file. Then install your new kernel as described in Chapter 8.

10.2.1 Installing Linux Runtime Libraries

This can be done one of two ways, either by using the linux_base port, or by installing them manually.

10.2.1.1 Installing Using the linux_base Port

This is by far the easiest method to use when installing the runtime libraries. It is just like installing any other port from the Ports Collection (/usr/ports/). Simply do the following:

```
# cd /usr/ports/emulators/linux_base-fc4
# make install distclean
```

You should now have working Linux binary compatibility. Some programs may complain about incorrect minor versions of the system libraries. In general, however, this does not seem to be a problem.

> **Note:** There may be multiple versions of the `emulators/linux_base` port available, corresponding to different versions of various Linux distributions. You should install the port most closely resembling the requirements of the Linux applications you would like to install.

10.2.1.2 Installing Libraries Manually

If you do not have the "ports" collection installed, you can install the libraries by hand instead. You will need the Linux shared libraries that the program depends on and the runtime linker. Also, you will need to create a "shadow root" directory, `/compat/linux`, for Linux libraries on your FreeBSD system. Any shared libraries opened by Linux programs run under FreeBSD will look in this tree first. So, if a Linux program loads, for example, `/lib/libc.so`, FreeBSD will first try to open `/compat/linux/lib/libc.so`, and if that does not exist, it will then try `/lib/libc.so`. Shared libraries should be installed in the shadow tree `/compat/linux/lib` rather than the paths that the Linux `ld.so` reports.

Generally, you will need to look for the shared libraries that Linux binaries depend on only the first few times that you install a Linux program on your FreeBSD system. After a while, you will have a sufficient set of Linux shared libraries on your system to be able to run newly imported Linux binaries without any extra work.

10.2.1.3 How to Install Additional Shared Libraries

What if you install the `linux_base` port and your application still complains about missing shared libraries? How do you know which shared libraries Linux binaries need, and where to get them? Basically, there are 2 possibilities (when following these instructions you will need to be `root` on your FreeBSD system).

If you have access to a Linux system, see what shared libraries the application needs, and copy them to your FreeBSD system. Look at the following example:

Let us assume you used FTP to get the Linux binary of **Doom**, and put it on a Linux system you have access to. You then can check which shared libraries it needs by running `ldd linuxdoom`, like so:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

You would need to get all the files from the last column, and put them under `/compat/linux`, with the names in the first column as symbolic links pointing to them. This means you eventually have these files on your FreeBSD system:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

> **Note:** Note that if you already have a Linux shared library with a matching major revision number to the first column of the `ldd` output, you will not need to copy the file named in the last column to your system, the one you already have should work. It is advisable to copy the shared library anyway if it is a newer version, though. You can remove the old one, as long as you make the symbolic link point to the new one. So, if you have these libraries on your system:
>
> ```
> /compat/linux/lib/libc.so.4.6.27
> /compat/linux/lib/libc.so.4 -> libc.so.4.6.27
> ```
>
> and you find a new binary that claims to require a later version according to the output of `ldd`:
>
> ```
> libc.so.4 (DLL Jump 4.5pl26) -> libc.so.4.6.29
> ```
>
> If it is only one or two versions out of date in the in the trailing digit then do not worry about copying `/lib/libc.so.4.6.29` too, because the program should work fine with the slightly older version. However, if you like, you can decide to replace the `libc.so` anyway, and that should leave you with:
>
> ```
> /compat/linux/lib/libc.so.4.6.29
> /compat/linux/lib/libc.so.4 -> libc.so.4.6.29
> ```

> **Note:** The symbolic link mechanism is *only* needed for Linux binaries. The FreeBSD runtime linker takes care of looking for matching major revision numbers itself and you do not need to worry about it.

10.2.2 Installing Linux ELF Binaries

ELF binaries sometimes require an extra step of “branding”. If you attempt to run an unbranded ELF binary, you will get an error message like the following:

```
% ./my-linux-elf-binary
ELF binary type not known
Abort
```

To help the FreeBSD kernel distinguish between a FreeBSD ELF binary from a Linux binary, use the brandelf(1) utility.

```
% brandelf -t Linux my-linux-elf-binary
```

The GNU toolchain now places the appropriate branding information into ELF binaries automatically, so this step should become increasingly unnecessary in the future.

10.2.3 Configuring the Hostname Resolver

If DNS does not work or you get this message:

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

You will need to configure a `/compat/linux/etc/host.conf` file containing:

```
order hosts, bind
multi on
```

The order here specifies that `/etc/hosts` is searched first and DNS is searched second. When `/compat/linux/etc/host.conf` is not installed, Linux applications find FreeBSD's `/etc/host.conf` and complain about the incompatible FreeBSD syntax. You should remove `bind` if you have not configured a name server using the `/etc/resolv.conf` file.

10.3 Installing Mathematica®

Updated for Mathematica 5.X by Boris Hollas.

This document describes the process of installing the Linux version of **Mathematica 5.X** onto a FreeBSD system.

The Linux version of **Mathematica** or **Mathematica for Students** can be ordered directly from Wolfram at http://www.wolfram.com/.

10.3.1 Running the Mathematica Installer

First, you have to tell FreeBSD that **Mathematica**'s Linux binaries use the Linux ABI. The easiest way to do so is to set the default ELF brand to Linux for all unbranded binaries with the command:

```
# sysctl kern.fallback_elf_brand=3
```

This will make FreeBSD assume that unbranded ELF binaries use the Linux ABI and so you should be able to run the installer straight from the CDROM.

Now, copy the file `MathInstaller` to your hard drive:

```
# mount /cdrom
# cp /cdrom/Unix/Installers/Linux/MathInstaller /localdir/
```

and in this file, replace `/bin/sh` in the first line by `/compat/linux/bin/sh`. This makes sure that the installer is executed by the Linux version of sh(1). Next, replace all occurrences of `Linux)` by `FreeBSD)` with a text editor or the script below in the next section. This tells the **Mathematica** installer, who calls `uname -s` to determine the operating system, to treat FreeBSD as a Linux-like operating system. Invoking `MathInstaller` will now install **Mathematica**.

10.3.2 Modifying the Mathematica Executables

The shell scripts that **Mathematica** created during installation have to be modified before you can use them. If you chose `/usr/local/bin` as the directory to place the **Mathematica** executables in, you will find symlinks in this directory to files called `math`, `mathematica`, `Mathematica`, and `MathKernel`. In each of these, replace `Linux)` by `FreeBSD)` with a text editor or the following shell script:

```
#!/bin/sh
cd /usr/local/bin
for i in math mathematica Mathematica MathKernel
  do sed 's/Linux)/FreeBSD)/g' $i > $i.tmp
  sed 's/\/bin\/sh/\/compat\/linux\/bin\/sh/g' $i.tmp > $i
  rm $i.tmp
  chmod a+x $i
done
```

10.3.3 Obtaining Your Mathematica Password

When you start **Mathematica** for the first time, you will be asked for a password. If you have not yet obtained a password from Wolfram, run the program `mathinfo` in the installation directory to obtain your "machine ID". This machine ID is based solely on the MAC address of your first Ethernet card, so you cannot run your copy of **Mathematica** on different machines.

When you register with Wolfram, either by email, phone or fax, you will give them the "machine ID" and they will respond with a corresponding password consisting of groups of numbers.

10.3.4 Running the Mathematica Frontend over a Network

Mathematica uses some special fonts to display characters not present in any of the standard font sets (integrals, sums, Greek letters, etc.). The X protocol requires these fonts to be install *locally*. This means you will have to copy these fonts from the CDROM or from a host with **Mathematica** installed to your local machine. These fonts are normally stored in `/cdrom/Unix/Files/SystemFiles/Fonts` on the CDROM, or `/usr/local/mathematica/SystemFiles/Fonts` on your hard drive. The actual fonts are in the subdirectories `Type1` and `X`. There are several ways to use them, as described below.

The first way is to copy them into one of the existing font directories in `/usr/X11R6/lib/X11/fonts`. This will require editing the `fonts.dir` file, adding the font names to it, and changing the number of fonts on the first line. Alternatively, you should also just be able to run mkfontdir(1) in the directory you have copied them to.

The second way to do this is to copy the directories to `/usr/X11R6/lib/X11/fonts`:

```
# cd /usr/X11R6/lib/X11/fonts
# mkdir X
```

```
# mkdir MathType1
# cd /cdrom/Unix/Files/SystemFiles/Fonts
# cp X/* /usr/X11R6/lib/X11/fonts/X
# cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
# cd /usr/X11R6/lib/X11/fonts/X
# mkfontdir
# cd ../MathType1
# mkfontdir
```

Now add the new font directories to your font path:

```
# xset fp+ /usr/X11R6/lib/X11/fonts/X
# xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
# xset fp rehash
```

If you are using the **Xorg** server, you can have these font directories loaded automatically by adding them to your `xorg.conf` file.

> **Note:** For **XFree86** servers, the configuration file is `XF86Config`.

If you *do not* already have a directory called `/usr/X11R6/lib/X11/fonts/Type1`, you can change the name of the `MathType1` directory in the example above to `Type1`.

10.4 Installing Maple™

Contributed by Aaron Kaplan. Thanks to Robert Getschmann.

Maple™ is a commercial mathematics program similar to **Mathematica**. You must purchase this software from http://www.maplesoft.com/ and then register there for a license file. To install this software on FreeBSD, please follow these simple steps.

1. Execute the `INSTALL` shell script from the product distribution. Choose the "RedHat" option when prompted by the installation program. A typical installation directory might be `/usr/local/maple`.
2. If you have not done so, order a license for **Maple** from Maple Waterloo Software (http://register.maplesoft.com/) and copy it to `/usr/local/maple/license/license.dat`.
3. Install the **FLEXlm** license manager by running the `INSTALL_LIC` install shell script that comes with **Maple**. Specify the primary hostname for your machine for the license server.
4. Patch the `/usr/local/maple/bin/maple.system.type` file with the following:

```
   ----- snip ------------------
*** maple.system.type.orig      Sun Jul  8 16:35:33 2001
--- maple.system.type   Sun Jul  8 16:35:51 2001
***************
*** 72,77 ****
--- 72,78 ----
          # the IBM RS/6000 AIX case
          MAPLE_BIN="bin.IBM_RISC_UNIX"
```

```
          ;;
+     "FreeBSD"|\
      "Linux")
          # the Linux/x86 case
        # We have two Linux implementations, one for Red Hat and
   ----- snip end of patch -----
```

Please note that after the `"FreeBSD"|\` no other whitespace should be present.

This patch instructs **Maple** to recognize "FreeBSD" as a type of Linux system. The `bin/maple` shell script calls the `bin/maple.system.type` shell script which in turn calls `uname -a` to find out the operating system name. Depending on the OS name it will find out which binaries to use.

5. Start the license server.

   The following script, installed as `/usr/local/etc/rc.d/lmgrd.sh` is a convenient way to start up `lmgrd`:

```
   ----- snip ------------

#! /bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin
PATH=${PATH}:/usr/local/maple/bin:/usr/local/maple/FLEXlm/UNIX/LINUX
export PATH

LICENSE_FILE=/usr/local/maple/license/license.dat
LOG=/var/log/lmgrd.log

case "$1" in
start)
 lmgrd -c ${LICENSE_FILE} 2>> ${LOG} 1>&2
 echo -n " lmgrd"
 ;;
stop)
 lmgrd -c ${LICENSE_FILE} -x lmdown 2>> ${LOG} 1>&2
 ;;
*)
 echo "Usage: ‘basename $0‘ {start|stop}" 1>&2
 exit 64
 ;;
esac

exit 0
   ----- snip ------------
```

6. Test-start **Maple**:

```
% cd /usr/local/maple/bin
% ./xmaple
```

   You should be up and running. Make sure to write Maplesoft to let them know you would like a native FreeBSD version!

10.4.1 Common Pitfalls

- The **FLEXlm** license manager can be a difficult tool to work with. Additional documentation on the subject can be found at http://www.globetrotter.com/.
- `lmgrd` is known to be very picky about the license file and to core dump if there are any problems. A correct license file should look like this:

```
# ============================================================
# License File for UNIX Installations ("Pointer File")
# ============================================================
SERVER chillig ANY
#USE_SERVER
VENDOR maplelmg

FEATURE Maple maplelmg 2000.0831 permanent 1 XXXXXXXXXXXX \
         PLATFORMS=i86_r ISSUER="Waterloo Maple Inc." \
         ISSUED=11-may-2000 NOTICE=" Technische Universitat Wien" \
         SN=XXXXXXXXX
```

Note: Serial number and key 'X"ed out. `chillig` is a hostname.

Editing the license file works as long as you do not touch the "FEATURE" line (which is protected by the license key).

10.5 Installing MATLAB®

Contributed by Dan Pelleg.

This document describes the process of installing the Linux version of **MATLAB® version 6.5** onto a FreeBSD system. It works quite well, with the exception of the **Java Virtual Machine**™ (see Section 10.5.3).

The Linux version of **MATLAB** can be ordered directly from The MathWorks at http://www.mathworks.com. Make sure you also get the license file or instructions how to create it. While you are there, let them know you would like a native FreeBSD version of their software.

10.5.1 Installing MATLAB

To install **MATLAB**, do the following:

1. Insert the installation CD and mount it. Become `root`, as recommended by the installation script. To start the installation script type:

```
# /compat/linux/bin/sh /cdrom/install
```

Tip: The installer is graphical. If you get errors about not being able to open a display, type `setenv HOME ~USER`, where *USER* is the user you did a su(1) as.

2. When asked for the **MATLAB** root directory, type: **`/compat/linux/usr/local/matlab`**.

   Tip: For easier typing on the rest of the installation process, type this at your shell prompt: `set MATLAB=/compat/linux/usr/local/matlab`

3. Edit the license file as instructed when obtaining the **MATLAB** license.

   Tip: You can prepare this file in advance using your favorite editor, and copy it to `$MATLAB/license.dat` before the installer asks you to edit it.

4. Complete the installation process.

At this point your **MATLAB** installation is complete. The following steps apply "glue" to connect it to your FreeBSD system.

10.5.2 License Manager Startup

1. Create symlinks for the license manager scripts:

```
# ln -s $MATLAB/etc/lmboot /usr/local/etc/lmboot_TMW
# ln -s $MATLAB/etc/lmdown /usr/local/etc/lmdown_TMW
```

2. Create a startup file at `/usr/local/etc/rc.d/flexlm.sh`. The example below is a modified version of the distributed `$MATLAB/etc/rc.lm.glnx86`. The changes are file locations, and startup of the license manager under Linux emulation.

```
#!/bin/sh
case "$1" in
  start)
        if [ -f /usr/local/etc/lmboot_TMW ]; then
              /compat/linux/bin/sh /usr/local/etc/lmboot_TMW -u username && echo 'MATLAB_lmgrd'
        fi
        ;;
  stop)
 if [ -f /usr/local/etc/lmdown_TMW ]; then
            /compat/linux/bin/sh /usr/local/etc/lmdown_TMW  > /dev/null 2>&1
 fi
        ;;
  *)
 echo "Usage: $0 {start|stop}"
 exit 1
 ;;
esac

exit 0
```

   Important: The file must be made executable:

```
# chmod +x /usr/local/etc/rc.d/flexlm.sh
```

You must also replace *username* above with the name of a valid user on your system (and not `root`).

3. Start the license manager with the command:

```
# /usr/local/etc/rc.d/flexlm.sh start
```

10.5.3 Linking the Java Runtime Environment

Change the **Java** Runtime Environment (JRE) link to one working under FreeBSD:

```
# cd $MATLAB/sys/java/jre/glnx86/
# unlink jre; ln -s ./jre1.1.8 ./jre
```

10.5.4 Creating a MATLAB Startup Script

1. Place the following startup script in `/usr/local/bin/matlab`:

```
#!/bin/sh
/compat/linux/bin/sh /compat/linux/usr/local/matlab/bin/matlab "$@"
```

2. Then type the command `chmod +x /usr/local/bin/matlab`.

Tip: Depending on your version of `emulators/linux_base`, you may run into errors when running this script. To avoid that, edit the file `/compat/linux/usr/local/matlab/bin/matlab`, and change the line that says:

```
if [ ‘expr "$lscmd" : ’.*->.*’‘ -ne 0 ]; then
```

(in version 13.0.1 it is on line 410) to this line:

```
if test -L $newbase; then
```

10.5.5 Creating a MATLAB Shutdown Script

The following is needed to solve a problem with MATLAB not exiting correctly.

1. Create a file `$MATLAB/toolbox/local/finish.m`, and in it put the single line:

```
! $MATLAB/bin/finish.sh
```

Note: The `$MATLAB` is literal.

Tip: In the same directory, you will find the files `finishsav.m` and `finishdlg.m`, which let you save your workspace before quitting. If you use either of them, insert the line above immediately after the `save` command.

2. Create a file `$MATLAB/bin/finish.sh`, which will contain the following:

```
#!/usr/compat/linux/bin/sh
(sleep 5; killall -1 matlab_helper) &
exit 0
```

3. Make the file executable:

```
# chmod +x $MATLAB/bin/finish.sh
```

10.5.6 Using MATLAB

At this point you are ready to type `matlab` and start using it.

10.6 Installing Oracle®

Contributed by Marcel Moolenaar.

10.6.1 Preface

This document describes the process of installing **Oracle 8.0.5** and **Oracle 8.0.5.1 Enterprise Edition** for Linux onto a FreeBSD machine.

10.6.2 Installing the Linux Environment

Make sure you have both `emulators/linux_base` and `devel/linux_devtools` from the Ports Collection installed. If you run into difficulties with these ports, you may have to use the packages or older versions available in the Ports Collection.

If you want to run the intelligent agent, you will also need to install the Red Hat Tcl package: `tcl-8.0.3-20.i386.rpm`. The general command for installing packages with the official **RPM** port (`archivers/rpm`) is:

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/lib/rpm package
```

Installation of the *`package`* should not generate any errors.

10.6.3 Creating the Oracle Environment

Before you can install **Oracle**, you need to set up a proper environment. This document only describes what to do *specially* to run **Oracle** for Linux on FreeBSD, not what has been described in the **Oracle** installation guide.

10.6.3.1 Kernel Tuning

As described in the **Oracle** installation guide, you need to set the maximum size of shared memory. Do not use `SHMMAX` under FreeBSD. `SHMMAX` is merely calculated out of `SHMMAXPGS` and `PGSIZE`. Therefore define `SHMMAXPGS`. All other options can be used as described in the guide. For example:

```
options SHMMAXPGS=10000
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

Set these options to suit your intended use of **Oracle**.

Also, make sure you have the following options in your kernel configuration file:

```
options SYSVSHM #SysV shared memory
options SYSVSEM #SysV semaphores
options SYSVMSG #SysV interprocess communication
```

10.6.3.2 Oracle Account

Create an `oracle` account just as you would create any other account. The `oracle` account is special only that you need to give it a Linux shell. Add `/compat/linux/bin/bash` to `/etc/shells` and set the shell for the `oracle` account to `/compat/linux/bin/bash`.

10.6.3.3 Environment

Besides the normal **Oracle** variables, such as `ORACLE_HOME` and `ORACLE_SID` you must set the following environment variables:

Variable	Value
`LD_LIBRARY_PATH`	`$ORACLE_HOME/lib`
`CLASSPATH`	`$ORACLE_HOME/jdbc/lib/classes111.zip`
`PATH`	`/compat/linux/bin /compat/linux/sbin /compat/linux/usr/bin /compat/linux/usr/sbin /bin /sbin /usr/bin /usr/sbin /usr/local/bin $ORACLE_HOME/bin`

It is advised to set all the environment variables in `.profile`. A complete example is:

```
ORACLE_BASE=/oracle; export ORACLE_BASE
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin
```

```
PATH=$PATH:/compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin
PATH=$PATH:/usr/local/bin:$ORACLE_HOME/bin
export PATH
```

10.6.4 Installing Oracle

Due to a slight inconsistency in the Linux emulator, you need to create a directory named `.oracle` in `/var/tmp` before you start the installer. Let it be owned by the `oracle` user. You should be able to install **Oracle** without any problems. If you have problems, check your **Oracle** distribution and/or configuration first! After you have installed **Oracle**, apply the patches described in the next two subsections.

A frequent problem is that the TCP protocol adapter is not installed right. As a consequence, you cannot start any TCP listeners. The following actions help solve this problem:

```
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk ntcontab.o
# cd $ORACLE_HOME/lib
# ar r libnetwork.a ntcontab.o
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk install
```

Do not forget to run `root.sh` again!

10.6.4.1 Patching root.sh

When installing **Oracle**, some actions, which need to be performed as `root`, are recorded in a shell script called `root.sh`. This script is written in the `orainst` directory. Apply the following patch to `root.sh`, to have it use to proper location of `chown` or alternatively run the script under a Linux native shell.

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998
***************
*** 31,37 ****
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/bin/chown
#
# Define variables to be used in this script
--- 31,37 ----
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
# Define variables to be used in this script
```

When you do not install **Oracle** from CD, you can patch the source for `root.sh`. It is called `rthd.sh` and is located in the `orainst` directory in the source tree.

10.6.4.2 Patching genclntsh

The script `genclntsh` is used to create a single shared client library. It is used when building the demos. Apply the following patch to comment out the definition of `PATH`:

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998
***************
*** 32,38 ****
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
```

10.6.5 Running Oracle

When you have followed the instructions, you should be able to run **Oracle** as if it was run on Linux itself.

10.7 Installing SAP® R/3®

Contributed by Holger Kipp. Original version converted to SGML by Valentino Vaschetto.

Installations of **SAP** Systems using FreeBSD will not be supported by the SAP support team ——they only offer support for certified platforms.

10.7.1 Preface

This document describes a possible way of installing a **SAP R/3 System** with **Oracle Database** for Linux onto a FreeBSD machine, including the installation of FreeBSD and **Oracle**. Two different configurations will be described:

- **SAP R/3 4.6B (IDES)** with **Oracle 8.0.5** on FreeBSD 4.3-STABLE
- **SAP R/3 4.6C** with **Oracle 8.1.7** on FreeBSD 4.5-STABLE

Even though this document tries to describe all important steps in a greater detail, it is not intended as a replacement for the **Oracle** and **SAP R/3** installation guides.

Please see the documentation that comes with the **SAP R/3** Linux edition for **SAP** and **Oracle** specific questions, as well as resources from **Oracle** and **SAP OSS**.

10.7.2 Software

The following CD-ROMs have been used for **SAP** installations:

10.7.2.1 SAP R/3 4.6B, Oracle 8.0.5

Name	Number	Description
KERNEL	51009113	SAP Kernel Oracle / Installation / AIX, Linux, Solaris
RDBMS	51007558	Oracle / RDBMS 8.0.5.X / Linux
EXPORT1	51010208	IDES / DB-Export / Disc 1 of 6
EXPORT2	51010209	IDES / DB-Export / Disc 2 of 6
EXPORT3	51010210	IDES / DB-Export / Disc 3 of 6
EXPORT4	51010211	IDES / DB-Export / Disc 4 of 6
EXPORT5	51010212	IDES / DB-Export / Disc 5 of 6
EXPORT6	51010213	IDES / DB-Export / Disc 6 of 6

Additionally, we used the **Oracle 8 Server** (Pre-production version 8.0.5 for Linux, Kernel Version 2.0.33) CD which is not really necessary, and FreeBSD 4.3-STABLE (it was only a few days past 4.3 RELEASE).

10.7.2.2 SAP R/3 4.6C SR2, Oracle 8.1.7

Name	Number	Description
KERNEL	51014004	SAP Kernel Oracle / SAP Kernel Version 4.6D / DEC, Linux
RDBMS	51012930	Oracle 8.1.7/ RDBMS / Linux
EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 1 of 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 2 of 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 3 of 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 4 of 4
LANG1	51013954	Release 4.6C SR2 / Language / DE, EN, FR / Disc 1 of 3

Depending on the languages you would like to install, additional language CDs might be necessary. Here we are just using DE and EN, so the first language CD is the only one needed. As a little note, the numbers for all four EXPORT CDs are identical. All three language CDs also have the same number (this is different from the 4.6B IDES release CD numbering). At the time of writing this installation is running on FreeBSD 4.5-STABLE (20.03.2002).

10.7.3 SAP Notes

The following notes should be read before installing **SAP R/3** and proved to be useful during installation:

10.7.3.1 SAP R/3 4.6B, Oracle 8.0.5

Number	Title
0171356	SAP Software on Linux: Essential Comments
0201147	INST: 4.6C R/3 Inst. on UNIX - Oracle
0373203	Update / Migration Oracle 8.0.5 --> 8.0.6/8.1.6 LINUX
0072984	Release of Digital UNIX 4.0B for Oracle
0130581	R3SETUP step DIPGNTAB terminates
0144978	Your system has not been installed correctly
0162266	Questions and tips for R3SETUP on Windows NT / W2K

10.7.3.2 SAP R/3 4.6C, Oracle 8.1.7

Number	Title
0015023	Initializing table TCPDB (RSXP0004) (EBCDIC)
0045619	R/3 with several languages or typefaces
0171356	SAP Software on Linux: Essential Comments
0195603	RedHat 6.1 Enterprise version: Known problems
0212876	The new archiving tool SAPCAR
0300900	Linux: Released DELL Hardware
0377187	RedHat 6.2: important remarks
0387074	INST: R/3 4.6C SR2 Installation on UNIX
0387077	INST: R/3 4.6C SR2 Inst. on UNIX - Oracle
0387078	SAP Software on UNIX: OS Dependencies 4.6C SR2

10.7.4 Hardware Requirements

The following equipment is sufficient for the installation of a **SAP R/3 System**. For production use, a more exact sizing is of course needed:

Component	4.6B	4.6C
Processor	2 x 800MHz Pentium III	2 x 800MHz Pentium III
Memory	1GB ECC	2GB ECC
Hard Disk Space	50-60GB (IDES)	50-60GB (IDES)

For use in production, Xeon Processors with large cache, high-speed disk access (SCSI, RAID hardware controller),

USV and ECC-RAM is recommended. The large amount of hard disk space is due to the preconfigured IDES System, which creates 27 GB of database files during installation. This space is also sufficient for initial production systems and application data.

10.7.4.1 SAP R/3 4.6B, Oracle 8.0.5

The following off-the-shelf hardware was used: a dual processor board with 2 800 MHz Pentium III processors, Adaptec® 29160 Ultra160 SCSI adapter (for accessing a 40/80 GB DLT tape drive and CDROM), Mylex® AcceleRAID™ (2 channels, firmware 6.00-1-00 with 32 MB RAM). To the Mylex RAID controller are attached two 17 GB hard disks (mirrored) and four 36 GB hard disks (RAID level 5).

10.7.4.2 SAP R/3 4.6C, Oracle 8.1.7

For this installation a Dell™ PowerEdge™ 2500 was used: a dual processor board with two 1000 MHz Pentium III processors (256 kB Cache), 2 GB PC133 ECC SDRAM, PERC/3 DC PCI RAID Controller with 128 MB, and an EIDE DVD-ROM drive. To the RAID controller are attached two 18 GB hard disks (mirrored) and four 36 GB hard disks (RAID level 5).

10.7.5 Installation of FreeBSD

First you have to install FreeBSD. There are several ways to do this, for more information read the Section 2.13.

10.7.5.1 Disk Layout

To keep it simple, the same disk layout both for the **SAP R/3 46B** and **SAP R/3 46C SR2** installation was used. Only the device names changed, as the installations were on different hardware (`/dev/da` and `/dev/amr` respectively, so if using an AMI MegaRAID®, one will see `/dev/amr0s1a` instead of `/dev/da0s1a`):

File system	Size (1k-blocks)	Size (GB)	Mounted on
`/dev/da0s1a`	1.016.303	1	`/`
`/dev/da0s1b`		6	swap
`/dev/da0s1e`	2.032.623	2	`/var`
`/dev/da0s1f`	8.205.339	8	`/usr`
`/dev/da1s1e`	45.734.361	45	`/compat/linux/oracle`
`/dev/da1s1f`	2.032.623	2	`/compat/linux/sapmnt`
`/dev/da1s1g`	2.032.623	2	`/compat/linux/usr/sap`

Configure and initialize the two logical drives with the Mylex or PERC/3 RAID software beforehand. The software can be started during the BIOS boot phase.

Please note that this disk layout differs slightly from the SAP recommendations, as SAP suggests mounting the **Oracle** subdirectories (and some others) separately ——we decided to just create them as real subdirectories for simplicity.

10.7.5.2 `make world` and a New Kernel

Download the latest -STABLE sources. Rebuild world and your custom kernel after configuring your kernel configuration file. Here you should also include the kernel parameters which are required for both **SAP R/3** and **Oracle**.

10.7.6 Installing the Linux Environment

10.7.6.1 Installing the Linux Base System

First the linux_base port needs to be installed (as `root`):

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

10.7.6.2 Installing Linux Development Environment

The Linux development environment is needed, if you want to install **Oracle** on FreeBSD according to the Section 10.6:

```
# cd /usr/ports/devel/linux_devtools
# make install distclean
```

The Linux development environment has only been installed for the **SAP R/3 46B IDES** installation. It is not needed, if the **Oracle DB** is not relinked on the FreeBSD system. This is the case if you are using the **Oracle** tarball from a Linux system.

10.7.6.3 Installing the Necessary RPMs

To start the `R3SETUP` program, PAM support is needed. During the first **SAP** Installation on FreeBSD 4.3-STABLE we tried to install PAM with all the required packages and finally forced the installation of the PAM package, which worked. For **SAP R/3 4.6C SR2** we directly forced the installation of the PAM RPM, which also works, so it seems the dependent packages are not needed:

```
# rpm -i --ignoreos --nodeps --root /compat/linux --dbpath /var/lib/rpm \
pam-0.68-7.i386.rpm
```

For **Oracle 8.0.5** to run the intelligent agent, we also had to install the RedHat Tcl package `tcl-8.0.5-30.i386.rpm` (otherwise the relinking during **Oracle** installation will not work). There are some other issues regarding relinking of **Oracle**, but that is a **Oracle** Linux issue, not FreeBSD specific.

10.7.6.4 Some Additional Hints

It might also be a good idea to add `linprocfs` to `/etc/fstab`, for more information, see the linprocfs(5) manual page. Another parameter to set is `kern.fallback_elf_brand=3` which is done in the file `/etc/sysctl.conf`.

10.7.7 Creating the SAP R/3 Environment

10.7.7.1 Creating the Necessary File Systems and Mountpoints

For a simple installation, it is sufficient to create the following file systems:

mount point	size in GB
/compat/linux/oracle	45 GB
/compat/linux/sapmnt	2 GB
/compat/linux/usr/sap	2 GB

It is also necessary to created some links. Otherwise the **SAP** Installer will complain, as it is checking the created links:

```
# ln -s /compat/linux/oracle /oracle
# ln -s /compat/linux/sapmnt /sapmnt
# ln -s /compat/linux/usr/sap /usr/sap
```

Possible error message during installation (here with System *PRD* and the **SAP R/3 4.6C SR2** installation):

```
INFO 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:200
    Checking existence of symbolic link /usr/sap/PRD/SYS/exe/dbg to
    /sapmnt/PRD/exe. Creating if it does not exist...

WARNING 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:400
    Link /usr/sap/PRD/SYS/exe/dbg exists but it points to file
    /compat/linux/sapmnt/PRD/exe instead of /sapmnt/PRD/exe. The
    program cannot go on as long as this link exists at this
    location. Move the link to another location.

ERROR 2002-03-19 16:45:36 R3LINKS_IND_IND Ins_SetupLinks:0
    can not setup link '/usr/sap/PRD/SYS/exe/dbg' with content
    '/sapmnt/PRD/exe'
```

10.7.7.2 Creating Users and Directories

SAP R/3 needs two users and three groups. The user names depend on the **SAP** system ID (SID) which consists of three letters. Some of these SIDs are reserved by **SAP** (for example `SAP` and `NIX`. For a complete list please see the **SAP** documentation). For the IDES installation we used `IDS`, for the 4.6C SR2 installation `PRD`, as that system is intended for production use. We have therefore the following groups (group IDs might differ, these are just the values we used with our installation):

group ID	group name	description
100	dba	Data Base Administrator
101	sapsys	SAP System
102	oper	Data Base Operator

For a default **Oracle** installation, only group `dba` is used. As `oper` group, one also uses group `dba` (see **Oracle** and **SAP** documentation for further information).

We also need the following users:

user ID	user name	generic name	group	additional groups	description
1000	idsadm/prdadm	*sid*adm	sapsys	oper	SAP Administrator
1002	oraids/oraprd	ora*sid*	dba	oper	Oracle Administrator

Adding the users with adduser(8) requires the following (please note shell and home directory) entries for “SAP Administrator”:

```
Name: sidadm
Password: ******
Fullname: SAP Administrator SID
Uid: 1000
Gid: 101 (sapsys)
Class:
Groups: sapsys dba
HOME: /home/sidadm
Shell: bash  (/compat/linux/bin/bash)
```

and for “Oracle Administrator”:

```
Name: orasid
Password: ******
Fullname: Oracle Administrator SID
Uid: 1002
Gid: 100 (dba)
Class:
Groups: dba
HOME: /oracle/sid
Shell: bash  (/compat/linux/bin/bash)
```

This should also include group `oper` in case you are using both groups `dba` and `oper`.

10.7.7.3 Creating Directories

These directories are usually created as separate file systems. This depends entirely on your requirements. We choose to create them as simple directories, as they are all located on the same RAID 5 anyway:

First we will set owners and rights of some directories (as user `root`):

```
# chmod 775 /oracle
# chmod 777 /sapmnt
# chown root:dba /oracle
# chown sidadm:sapsys /compat/linux/usr/sap
# chmod 775 /compat/linux/usr/sap
```

Second we will create directories as user ora*sid*. These will all be subdirectories of `/oracle/SID`:

```
# su - orasid
```

```
# cd /oracle/SID
# mkdir mirrlogA mirrlogB origlogA origlogB
# mkdir sapdata1 sapdata2 sapdata3 sapdata4 sapdata5 sapdata6
# mkdir saparch sapreorg
# exit
```

For the **Oracle 8.1.7** installation some additional directories are needed:

```
# su - orasid
# cd /oracle
# mkdir 805_32
# mkdir client stage
# mkdir client/80x_32
# mkdir stage/817_32
# cd /oracle/SID
# mkdir 817_32
```

> **Note:** The directory `client/80x_32` is used with exactly this name. Do not replace the *x* with some number or anything.

In the third step we create directories as user *sid*`adm`:

```
# su - sidadm
# cd /usr/sap
# mkdir SID
# mkdir trans
# exit
```

10.7.7.4 Entries in `/etc/services`

SAP R/3 requires some entries in file `/etc/services`, which will not be set correctly during installation under FreeBSD. Please add the following entries (you need at least those entries corresponding to the instance number — —in this case, `00`. It will do no harm adding all entries from `00` to `99` for `dp`, `gw`, `sp` and `ms`). If you are going to use a **SAProuter** or need to access **SAP** OSS, you also need `99`, as port 3299 is usually used for the **SAProuter** process on the target system:

```
sapdp00    3200/tcp # SAP Dispatcher.      3200 + Instance-Number
sapgw00  3300/tcp # SAP Gateway.         3300 + Instance-Number
sapsp00  3400/tcp #                      3400 + Instance-Number
sapms00  3500/tcp #                      3500 + Instance-Number
sapmsSID 3600/tcp # SAP Message Server.  3600 + Instance-Number
sapgw00s   4800/tcp # SAP Secure Gateway   4800 + Instance-Number
```

10.7.7.5 Necessary Locales

SAP requires at least two locales that are not part of the default RedHat installation. SAP offers the required RPMs as download from their FTP server (which is only accessible if you are a customer with OSS access). See note 0171356 for a list of RPMs you need.

It is also possible to just create appropriate links (for example from *de_DE* and *en_US*), but we would not recommend this for a production system (so far it worked with the IDES system without any problems, though). The following locales are needed:

```
de_DE.ISO-8859-1
en_US.ISO-8859-1
```

Create the links like this:

```
# cd /compat/linux/usr/share/locale
# ln -s de_DE de_DE.ISO-8859-1
# ln -s en_US en_US.ISO-8859-1
```

If they are not present, there will be some problems during the installation. If these are then subsequently ignored (by setting the `STATUS` of the offending steps to `OK` in file `CENTRDB.R3S`), it will be impossible to log onto the **SAP** system without some additional effort.

10.7.7.6 Kernel Tuning

SAP R/3 systems need a lot of resources. We therefore added the following parameters to the kernel configuration file:

```
# Set these for memory pigs (SAP and Oracle):
options MAXDSIZ="(1024*1024*1024)"
options DFLDSIZ="(1024*1024*1024)"
# System V options needed.
options SYSVSHM #SYSV-style shared memory
options SHMMAXPGS=262144 #max amount of shared mem. pages
#options SHMMAXPGS=393216 #use this for the 46C inst.parameters
options SHMMNI=256 #max number of shared memory ident if.
options SHMSEG=100 #max shared mem.segs per process
options SYSVMSG #SYSV-style message queues
options MSGSEG=32767 #max num. of mes.segments in system
options MSGSSZ=32 #size of msg-seg. MUST be power of 2
options MSGMNB=65535 #max char. per message queue
options MSGTQL=2046 #max amount of msgs in system
options SYSVSEM #SYSV-style semaphores
options SEMMNU=256 #number of semaphore UNDO structures
options SEMMNS=1024 #number of semaphores in system
options SEMMNI=520 #number of semaphore identifiers
options SEMUME=100       #number of UNDO keys
```

The minimum values are specified in the documentation that comes from SAP. As there is no description for Linux, see the HP-UX section (32-bit) for further information. As the system for the 4.6C SR2 installation has more main memory, the shared segments can be larger both for **SAP** and **Oracle**, therefore choose a larger number of shared memory pages.

> **Note:** With the default installation of FreeBSD on i386, leave `MAXDSIZ` and `DFLDSIZ` at 1 GB maximum. Otherwise, strange errors like “`ORA-27102: out of memory`” and “`Linux Error: 12: Cannot allocate memory`” might happen.

10.7.8 Installing SAP R/3

10.7.8.1 Preparing SAP CDROMs

There are many CDROMs to mount and unmount during the installation. Assuming you have enough CDROM drives, you can just mount them all. We decided to copy the CDROMs contents to corresponding directories:

`/oracle/SID/sapreorg/cd-name`

where *cd-name* was one of `KERNEL`, `RDBMS`, `EXPORT1`, `EXPORT2`, `EXPORT3`, `EXPORT4`, `EXPORT5` and `EXPORT6` for the 4.6B/IDES installation, and `KERNEL`, `RDBMS`, `DISK1`, `DISK2`, `DISK3`, `DISK4` and `LANG` for the 4.6C SR2 installation. All the filenames on the mounted CDs should be in capital letters, otherwise use the `-g` option for mounting. So use the following commands:

```
# mount_cd9660 -g /dev/cd0a /mnt
# cp -R /mnt/* /oracle/SID/sapreorg/cd-name
# umount /mnt
```

10.7.8.2 Running the Installation Script

First you have to prepare an `install` directory:

```
# cd /oracle/SID/sapreorg
# mkdir install
# cd install
```

Then the installation script is started, which will copy nearly all the relevant files into the `install` directory:

```
# /oracle/SID/sapreorg/KERNEL/UNIX/INSTTOOL.SH
```

The IDES installation (4.6B) comes with a fully customized SAP R/3 demonstration system, so there are six instead of just three EXPORT CDs. At this point the installation template `CENTRDB.R3S` is for installing a standard central instance (**R/3** and database), not the IDES central instance, so one needs to copy the corresponding `CENTRDB.R3S` from the `EXPORT1` directory, otherwise `R3SETUP` will only ask for three EXPORT CDs.

The newer **SAP 4.6C SR2** release comes with four EXPORT CDs. The parameter file that controls the installation steps is `CENTRAL.R3S`. Contrary to earlier releases there are no separate installation templates for a central instance with or without database. **SAP** is using a separate template for database installation. To restart the installation later it is however sufficient to restart with the original file.

During and after installation, **SAP** requires `hostname` to return the computer name only, not the fully qualified domain name. So either set the hostname accordingly, or set an alias with `alias hostname='hostname -s'` for both `ora`*`sid`* and *`sid`*`adm` (and for `root` at least during installation steps performed as `root`). It is also possible to adjust the installed `.profile` and `.login` files of both users that are installed during **SAP** installation.

10.7.8.3 Start `R3SETUP` 4.6B

Make sure `LD_LIBRARY_PATH` is set correctly:

```
# export LD_LIBRARY_PATH=/oracle/IDS/lib:/sapmnt/IDS/exe:/oracle/805_32/lib
```

Start `R3SETUP` as `root` from installation directory:

```
# cd /oracle/IDS/sapreorg/install
# ./R3SETUP -f CENTRDB.R3S
```

The script then asks some questions (defaults in brackets, followed by actual input):

Question	Default	Input
Enter SAP System ID	[C11]	IDS**Enter**
Enter SAP Instance Number	[00]	**Enter**
Enter SAPMOUNT Directory	[/sapmnt]	**Enter**
Enter name of SAP central host	[troubadix.domain.de]	**Enter**
Enter name of SAP db host	[troubadix]	**Enter**
Select character set	[1] (WE8DEC)	**Enter**
Enter Oracle server version (1) Oracle 8.0.5, (2) Oracle 8.0.6, (3) Oracle 8.1.5, (4) Oracle 8.1.6		1**Enter**
Extract Oracle Client archive	[1] (Yes, extract)	**Enter**
Enter path to KERNEL CD	[/sapcd]	/oracle/IDS/sapreorg/KERNEL
Enter path to RDBMS CD	[/sapcd]	/oracle/IDS/sapreorg/RDBMS
Enter path to EXPORT1 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT1
Directory to copy EXPORT1 CD	[/oracle/IDS/sapreorg/CD4_DIR]	**Enter**
Enter path to EXPORT2 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT2
Directory to copy EXPORT2 CD	[/oracle/IDS/sapreorg/CD5_DIR]	**Enter**
Enter path to EXPORT3 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT3
Directory to copy EXPORT3 CD	[/oracle/IDS/sapreorg/CD6_DIR]	**Enter**
Enter path to EXPORT4 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT4
Directory to copy EXPORT4 CD	[/oracle/IDS/sapreorg/CD7_DIR]	**Enter**
Enter path to EXPORT5 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT5
Directory to copy EXPORT5 CD	[/oracle/IDS/sapreorg/CD8_DIR]	**Enter**
Enter path to EXPORT6 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT6
Directory to copy EXPORT6 CD	[/oracle/IDS/sapreorg/CD9_DIR]	**Enter**
Enter amount of RAM for SAP + DB		850**Enter** (in Megabytes)
Service Entry Message Server	[3600]	**Enter**
Enter Group-ID of sapsys	[101]	**Enter**
Enter Group-ID of oper	[102]	**Enter**
Enter Group-ID of dba	[100]	**Enter**
Enter User-ID of *sid*adm	[1000]	**Enter**
Enter User-ID of ora*sid*	[1002]	**Enter**
Number of parallel procs	[2]	**Enter**

If you had not copied the CDs to the different locations, then the **SAP** installer cannot find the CD needed (identified by the `LABEL.ASC` file on the CD) and would then ask you to insert and mount the CD and confirm or enter the

mount path.

The `CENTRDB.R3S` might not be error free. In our case, it requested EXPORT4 CD again but indicated the correct key (6_LOCATION, then 7_LOCATION etc.), so one can just continue with entering the correct values.

Apart from some problems mentioned below, everything should go straight through up to the point where the Oracle database software needs to be installed.

10.7.8.4 Start `R3SETUP` 4.6C SR2

Make sure `LD_LIBRARY_PATH` is set correctly. This is a different value from the 4.6B installation with **Oracle 8.0.5**:

```
# export LD_LIBRARY_PATH=/sapmnt/PRD/exe:/oracle/PRD/817_32/lib
```

Start `R3SETUP` as user `root` from installation directory:

```
# cd /oracle/PRD/sapreorg/install
# ./R3SETUP -f CENTRAL.R3S
```

The script then asks some questions (defaults in brackets, followed by actual input):

Question	Default	Input
Enter SAP System ID	[C11]	PRD**Enter**
Enter SAP Instance Number	[00]	**Enter**
Enter SAPMOUNT Directory	[/sapmnt]	**Enter**
Enter name of SAP central host	[majestix]	**Enter**
Enter Database System ID	[PRD]	PRD**Enter**
Enter name of SAP db host	[majestix]	**Enter**
Select character set	[1] (WE8DEC)	**Enter**
Enter Oracle server version (2) Oracle 8.1.7		2**Enter**
Extract Oracle Client archive	[1] (Yes, extract)	**Enter**
Enter path to KERNEL CD	[/sapcd]	/oracle/PRD/sapreorg/KERNEL
Enter amount of RAM for SAP + DB	2044	1800**Enter** (in Megabytes)
Service Entry Message Server	[3600]	**Enter**
Enter Group-ID of sapsys	[100]	**Enter**
Enter Group-ID of oper	[101]	**Enter**
Enter Group-ID of dba	[102]	**Enter**
Enter User-ID of `oraprd`	[1002]	**Enter**
Enter User-ID of `prdadm`	[1000]	**Enter**
LDAP support		3**Enter** (no support)
Installation step completed	[1] (continue)	**Enter**
Choose installation service	[1] (DB inst,file)	**Enter**

So far, creation of users gives an error during installation in phases OSUSERDBSID_IND_ORA (for creating user `ora`*`sid`*) and OSUSERSIDADM_IND_ORA (creating user *`sid`*`adm`).

Apart from some problems mentioned below, everything should go straight through up to the point where the Oracle database software needs to be installed.

10.7.9 Installing Oracle 8.0.5

Please see the corresponding SAP Notes and Oracle `Readmes` regarding Linux and **Oracle DB** for possible problems. Most if not all problems stem from incompatible libraries.

For more information on installing **Oracle**, refer to the Installing Oracle chapter.

10.7.9.1 Installing the Oracle 8.0.5 with `orainst`

If **Oracle 8.0.5** is to be used, some additional libraries are needed for successfully relinking, as **Oracle 8.0.5** was linked with an old glibc (RedHat 6.0), but RedHat 6.1 already uses a new glibc. So you have to install the following additional packages to ensure that linking will work:

`compat-libs-5.2-2.i386.rpm`

`compat-glibc-5.2-2.0.7.2.i386.rpm`

`compat-egcs-5.2-1.0.3a.1.i386.rpm`

`compat-egcs-c++-5.2-1.0.3a.1.i386.rpm`

`compat-binutils-5.2-2.9.1.0.23.1.i386.rpm`

See the corresponding SAP Notes or Oracle `Readmes` for further information. If this is no option (at the time of installation we did not have enough time to check this), one could use the original binaries, or use the relinked binaries from an original RedHat system.

For compiling the intelligent agent, the RedHat Tcl package must be installed. If you cannot get `tcl-8.0.3-20.i386.rpm`, a newer one like `tcl-8.0.5-30.i386.rpm` for RedHat 6.1 should also do.

Apart from relinking, the installation is straightforward:

```
# su - oraids
# export TERM=xterm
# export ORACLE_TERM=xterm
# export ORACLE_HOME=/oracle/IDS
# cd $ORACLE_HOME/orainst_sap
# ./orainst
```

Confirm all screens with **Enter** until the software is installed, except that one has to deselect the *Oracle On-Line Text Viewer*, as this is not currently available for Linux. **Oracle** then wants to relink with `i386-glibc20-linux-gcc` instead of the available `gcc`, `egcs` or `i386-redhat-linux-gcc .`

Due to time constrains we decided to use the binaries from an **Oracle 8.0.5 PreProduction** release, after the first attempt at getting the version from the RDBMS CD working, failed, and finding and accessing the correct RPMs was a nightmare at that time.

10.7.9.2 Installing the Oracle 8.0.5 Pre-production Release for Linux (Kernel 2.0.33)

This installation is quite easy. Mount the CD, start the installer. It will then ask for the location of the Oracle home directory, and copy all binaries there. We did not delete the remains of our previous RDBMS installation tries, though.

Afterwards, **Oracle** Database could be started with no problems.

10.7.10 Installing the Oracle 8.1.7 Linux Tarball

Take the tarball `oracle81732.tgz` you produced from the installation directory on a Linux system and untar it to `/oracle/SID/817_32/`.

10.7.11 Continue with SAP R/3 Installation

First check the environment settings of users `idsamd` (*sid*adm) and `oraids` (ora*sid*). They should now both have the files `.profile`, `.login` and `.cshrc` which are all using `hostname`. In case the system's hostname is the fully qualified name, you need to change `hostname` to `hostname -s` within all three files.

10.7.11.1 Database Load

Afterwards, `R3SETUP` can either be restarted or continued (depending on whether exit was chosen or not). `R3SETUP` then creates the tablespaces and loads the data (for 46B IDES, from EXPORT1 to EXPORT6, for 46C from DISK1 to DISK4) with `R3load` into the database.

When the database load is finished (might take a few hours), some passwords are requested. For test installations, one can use the well known default passwords (use different ones if security is an issue!):

Question	Input
Enter Password for sapr3	sap**Enter**
Confirum Password for sapr3	sap**Enter**
Enter Password for sys	change_on_install**Enter**
Confirm Password for sys	change_on_install**Enter**
Enter Password for system	manager**Enter**
Confirm Password for system	manager**Enter**

At this point We had a few problems with `dipgntab` during the 4.6B installation.

10.7.11.2 Listener

Start the **Oracle** Listener as user ora*sid* as follows:

```
% umask 0; lsnrctl start
```

Otherwise you might get the error ORA-12546 as the sockets will not have the correct permissions. See SAP Note 072984.

10.7.11.3 Updating MNLS Tables

If you plan to import non-Latin-1 languages into the **SAP** system, you have to update the Multi National Language Support tables. This is described in the SAP OSS Notes 15023 and 45619. Otherwise, you can skip this question during **SAP** installation.

> **Note:** If you do not need MNLS, it is still necessary to check the table TCPDB and initializing it if this has not been done. See SAP note 0015023 and 0045619 for further information.

10.7.12 Post-installation Steps

10.7.12.1 Request SAP R/3 License Key

You have to request your **SAP R/3** License Key. This is needed, as the temporary license that was installed during installation is only valid for four weeks. First get the hardware key. Log on as user `idsadm` and call `saplicense`:

```
# /sapmnt/IDS/exe/saplicense -get
```

Calling `saplicense` without parameters gives a list of options. Upon receiving the license key, it can be installed using:

```
# /sapmnt/IDS/exe/saplicense -install
```

You are then required to enter the following values:

```
SAP SYSTEM ID   = SID, 3 chars
CUSTOMER KEY    = hardware key, 11 chars
INSTALLATION NO = installation, 10 digits
EXPIRATION DATE = yyyymmdd, usually "99991231"
LICENSE KEY     = license key, 24 chars
```

10.7.12.2 Creating Users

Create a user within client 000 (for some tasks required to be done within client 000, but with a user different from users `sap*` and `ddic`). As a user name, We usually choose `wartung` (or `service` in English). Profiles required are `sap_new` and `sap_all`. For additional safety the passwords of default users within all clients should be changed (this includes users `sap*` and `ddic`).

10.7.12.3 Configure Transport System, Profile, Operation Modes, Etc.

Within client 000, user different from `ddic` and `sap*`, do at least the following:

Task	Transaction
Configure Transport System, e.g. as *Stand-Alone Transport Domain Entity*	STMS

Task	Transaction
Create / Edit Profile for System	RZ10
Maintain Operation Modes and Instances	RZ04

These and all the other post-installation steps are thoroughly described in **SAP** installation guides.

10.7.12.4 Edit `init`*`sid`*`.sap` (`initIDS.sap`)

The file `/oracle/IDS/dbs/initIDS.sap` contains the **SAP** backup profile. Here the size of the tape to be used, type of compression and so on need to be defined. To get this running with `sapdba` / `brbackup`, we changed the following values:

```
compress = hardware
archive_function = copy_delete_save
cpio_flags = "-ov --format=newc --block-size=128 --quiet"
cpio_in_flags = "-iuv --block-size=128 --quiet"
tape_size = 38000M
tape_address = /dev/nsa0
tape_address_rew = /dev/sa0
```

Explanations:

`compress`: The tape we use is a HP DLT1 which does hardware compression.

`archive_function`: This defines the default behavior for saving Oracle archive logs: new logfiles are saved to tape, already saved logfiles are saved again and are then deleted. This prevents lots of trouble if you need to recover the database, and one of the archive-tapes has gone bad.

`cpio_flags`: Default is to use `-B` which sets block size to 5120 Bytes. For DLT Tapes, HP recommends at least 32 K block size, so we used `--block-size=128` for 64 K. `--format=newc` is needed because we have inode numbers greater than 65535. The last option `--quiet` is needed as otherwise `brbackup` complains as soon as `cpio` outputs the numbers of blocks saved.

`cpio_in_flags`: Flags needed for loading data back from tape. Format is recognized automatically.

`tape_size`: This usually gives the raw storage capability of the tape. For security reason (we use hardware compression), the value is slightly lower than the actual value.

`tape_address`: The non-rewindable device to be used with `cpio`.

`tape_address_rew`: The rewindable device to be used with `cpio`.

10.7.12.5 Configuration Issues after Installation

The following **SAP** parameters should be tuned after installation (examples for IDES 46B, 1 GB memory):

Name	Value
ztta/roll_extension	250000000
abap/heap_area_dia	300000000
abap/heap_area_nondia	400000000
em/initial_size_MB	256

Name	Value
em/blocksize_kB	1024
ipc/shm_psize_40	70000000

SAP Note 0013026:

Name	Value
ztta/dynpro_area	2500000

SAP Note 0157246:

Name	Value
rdisp/ROLL_MAXFS	16000
rdisp/PG_MAXFS	30000

> **Note:** With the above parameters, on a system with 1 gigabyte of memory, one may find memory consumption similar to:
>
> ```
> Mem: 547M Active, 305M Inact, 109M Wired, 40M Cache, 112M Buf, 3492K Free
> ```

10.7.13 Problems during Installation

10.7.13.1 Restart `R3SETUP` after Fixing a Problem

`R3SETUP` stops if it encounters an error. If you have looked at the corresponding logfiles and fixed the error, you have to start `R3SETUP` again, usually selecting REPEAT as option for the last step `R3SETUP` complained about.

To restart `R3SETUP`, just start it with the corresponding `R3S` file:

```
# ./R3SETUP -f CENTRDB.R3S
```

for 4.6B, or with

```
# ./R3SETUP -f CENTRAL.R3S
```

for 4.6C, no matter whether the error occurred with `CENTRAL.R3S` or `DATABASE.R3S`.

> **Note:** At some stages, `R3SETUP` assumes that both database and **SAP** processes are up and running (as those were steps it already completed). Should errors occur and for example the database could not be started, you have to start both database and **SAP** by hand after you fixed the errors and before starting `R3SETUP` again.
>
> Do not forget to also start the **Oracle** listener again (as `ora`*`sid`* with `umask 0; lsnrctl start`) if it was also stopped (for example due to a necessary reboot of the system).

10.7.13.2 OSUSERSIDADM_IND_ORA during R3SETUP

If R3SETUP complains at this stage, edit the template file R3SETUP used at that time (CENTRDB.R3S (4.6B) or either CENTRAL.R3S or DATABASE.R3S (4.6C)). Locate [OSUSERSIDADM_IND_ORA] or search for the only STATUS=ERROR entry and edit the following values:

```
HOME=/home/sidadm (was empty)
STATUS=OK (had status ERROR)
```

Then you can restart R3SETUP again.

10.7.13.3 OSUSERDBSID_IND_ORA during R3SETUP

Possibly R3SETUP also complains at this stage. The error here is similar to the one in phase OSUSERSIDADM_IND_ORA. Just edit the template file R3SETUP used at that time (CENTRDB.R3S (4.6B) or either CENTRAL.R3S or DATABASE.R3S (4.6C)). Locate [OSUSERDBSID_IND_ORA] or search for the only STATUS=ERROR entry and edit the following value in that section:

```
STATUS=OK
```

Then restart R3SETUP.

10.7.13.4 "oraview.vrf FILE NOT FOUND" during Oracle Installation

You have not deselected *Oracle On-Line Text Viewer* before starting the installation. This is marked for installation even though this option is currently not available for Linux. Deselect this product inside the **Oracle** installation menu and restart installation.

10.7.13.5 "TEXTENV_INVALID" during R3SETUP, RFC or SAPgui Start

If this error is encountered, the correct locale is missing. SAP Note 0171356 lists the necessary RPMs that need be installed (e.g. saplocales-1.0-3, saposcheck-1.0-1 for RedHat 6.1). In case you ignored all the related errors and set the corresponding STATUS from ERROR to OK (in CENTRDB.R3S) every time R3SETUP complained and just restarted R3SETUP, the **SAP** system will not be properly configured and you will then not be able to connect to the system with a **SAPgui**, even though the system can be started. Trying to connect with the old Linux **SAPgui** gave the following messages:

```
Sat May 5 14:23:14 2001
*** ERROR => no valid userarea given [trgmsgo. 0401]
Sat May 5 14:23:22 2001
*** ERROR => ERROR NR 24 occured [trgmsgi. 0410]
*** ERROR => Error when generating text environment. [trgmsgi. 0435]
*** ERROR => function failed [trgmsgi. 0447]
*** ERROR => no socket operation allowed [trxio.c 3363]
Speicherzugriffsfehler
```

This behavior is due to **SAP R/3** being unable to correctly assign a locale and also not being properly configured itself (missing entries in some database tables). To be able to connect to **SAP**, add the following entries to file DEFAULT.PFL (see Note 0043288):

```
abap/set_etct_env_at_new_mode = 0
install/collate/active = 0
rscp/TCP0B = TCP0B
```

Restart the **SAP** system. Now you can connect to the system, even though country-specific language settings might not work as expected. After correcting country settings (and providing the correct locales), these entries can be removed from `DEFAULT.PFL` and the **SAP** system can be restarted.

10.7.13.6 ORA-00001

This error only happened with **Oracle 8.1.7** on FreeBSD. The reason was that the **Oracle** database could not initialize itself properly and crashed, leaving semaphores and shared memory on the system. The next try to start the database then returned ORA-00001.

Find them with `ipcs -a` and remove them with `ipcrm`.

10.7.13.7 ORA-00445 (Background Process PMON Did Not Start)

This error happened with **Oracle 8.1.7**. This error is reported if the database is started with the usual `startsap` script (for example `startsap_majestix_00`) as user `prdadm`.

A possible workaround is to start the database as user `oraprd` instead with `svrmgrl`:

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> startup;
SVRMGR> exit
```

10.7.13.8 ORA-12546 (Start Listener with Correct Permissions)

Start the **Oracle** listener as user `oraids` with the following commands:

```
# umask 0; lsnrctl start
```

Otherwise you might get ORA-12546 as the sockets will not have the correct permissions. See SAP Note 0072984.

10.7.13.9 ORA-27102 (Out of Memory)

This error happened whilst trying to use values for `MAXDSIZ` and `DFLDSIZ` greater than 1 GB (1024x1024x1024). Additionally, we got “`Linux Error 12: Cannot allocate memory`”.

10.7.13.10 [DIPGNTAB_IND_IND] during `R3SETUP`

In general, see SAP Note 0130581 (`R3SETUP` step `DIPGNTAB` terminates). During the IDES-specific installation, for some reason the installation process was not using the proper **SAP** system name “IDS”, but the empty string "" instead. This leads to some minor problems with accessing directories, as the paths are generated dynamically using *`SID`* (in this case IDS). So instead of accessing:

```
/usr/sap/IDS/SYS/...
```

```
/usr/sap/IDS/DVMGS00
```

the following paths were used:

```
/usr/sap//SYS/...
/usr/sap/D00
```

To continue with the installation, we created a link and an additional directory:

```
# pwd
/compat/linux/usr/sap
# ls -l
total 4
drwxr-xr-x 3  idsadm sapsys 512 May 5 11:20 D00
drwxr-x--x 5  idsadm sapsys 512 May 5 11:35 IDS
lrwxr-xr-x 1  root   sapsys 7 May 5 11:35 SYS -> IDS/SYS
drwxrwxr-x 2  idsadm sapsys 512 May 5 13:00 tmp
drwxrwxr-x 11 idsadm sapsys 512 May 4 14:20 trans
```

We also found SAP Notes (0029227 and 0008401) describing this behavior. We did not encounter any of these problems with the **SAP 4.6C** installation.

10.7.13.11 [RFCRSWBOINI_IND_IND] during R3SETUP

During installation of **SAP 4.6C**, this error was just the result of another error happening earlier during installation. In this case, you have to look through the corresponding logfiles and correct the real problem.

If after looking through the logfiles this error is indeed the correct one (check the SAP Notes), you can set `STATUS` of the offending step from `ERROR` to `OK` (file `CENTRDB.R3S`) and restart `R3SETUP`. After installation, you have to execute the report `RSWBOINS` from transaction SE38. See SAP Note 0162266 for additional information about phase `RFCRSWBOINI` and `RFCRADDBDIF`.

10.7.13.12 [RFCRADDBDIF_IND_IND] during R3SETUP

Here the same restrictions apply: make sure by looking through the logfiles, that this error is not caused by some previous problems.

If you can confirm that SAP Note 0162266 applies, just set `STATUS` of the offending step from `ERROR` to `OK` (file `CENTRDB.R3S`) and restart `R3SETUP`. After installation, you have to execute the report `RADDBDIF` from transaction SE38.

10.7.13.13 sigaction sig31: File size limit exceeded

This error occurred during start of **SAP** processes *disp+work*. If starting **SAP** with the `startsap` script, subprocesses are then started which detach and do the dirty work of starting all other **SAP** processes. As a result, the script itself will not notice if something goes wrong.

To check whether the **SAP** processes did start properly, have a look at the process status with `ps ax | grep` *`SID`*, which will give you a list of all **Oracle** and **SAP** processes. If it looks like some processes are missing or if you cannot connect to the **SAP** system, look at the corresponding logfiles which can be found at `/usr/sap/`*`SID`*`/DVEBMGS`*`nr`*`/work/`. The files to look at are `dev_ms` and `dev_disp`.

Signal 31 happens here if the amount of shared memory used by **Oracle** and **SAP** exceed the one defined within the kernel configuration file and could be resolved by using a larger value:

```
# larger value for 46C production systems:
options SHMMAXPGS=393216
# smaller value sufficient for 46B:
#options SHMMAXPGS=262144
```

10.7.13.14 Start of `saposcol` Failed

There are some problems with the program `saposcol` (version 4.6D). The **SAP** system is using `saposcol` to collect data about the system performance. This program is not needed to use the **SAP** system, so this problem can be considered a minor one. The older versions (4.6B) does work, but does not collect all the data (many calls will just return 0, for example for CPU usage).

10.8 Advanced Topics

If you are curious as to how the Linux binary compatibility works, this is the section you want to read. Most of what follows is based heavily on an email written to FreeBSD chat 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat) by Terry Lambert <`tlambert@primenet.com`> (Message ID: <`199906020108.SAA07001@usr09.primenet.com`>).

10.8.1 How Does It Work?

FreeBSD has an abstraction called an “execution class loader”. This is a wedge into the execve(2) system call.

What happens is that FreeBSD has a list of loaders, instead of a single loader with a fallback to the `#!` loader for running any shell interpreters or shell scripts.

Historically, the only loader on the UNIX platform examined the magic number (generally the first 4 or 8 bytes of the file) to see if it was a binary known to the system, and if so, invoked the binary loader.

If it was not the binary type for the system, the execve(2) call returned a failure, and the shell attempted to start executing it as shell commands.

The assumption was a default of “whatever the current shell is”.

Later, a hack was made for sh(1) to examine the first two characters, and if they were `:\n`, then it invoked the csh(1) shell instead (we believe SCO first made this hack).

What FreeBSD does now is go through a list of loaders, with a generic `#!` loader that knows about interpreters as the characters which follow to the next whitespace next to last, followed by a fallback to `/bin/sh`.

For the Linux ABI support, FreeBSD sees the magic number as an ELF binary (it makes no distinction between FreeBSD, Solaris, Linux, or any other OS which has an ELF image type, at this point).

The ELF loader looks for a specialized *brand*, which is a comment section in the ELF image, and which is not present on SVR4/Solaris ELF binaries.

For Linux binaries to function, they must be *branded* as type `Linux` from brandelf(1):

```
# brandelf -t Linux file
```

When this is done, the ELF loader will see the `Linux` brand on the file.

When the ELF loader sees the `Linux` brand, the loader replaces a pointer in the `proc` structure. All system calls are indexed through this pointer (in a traditional UNIX system, this would be the `sysent[]` structure array, containing the system calls). In addition, the process is flagged for special handling of the trap vector for the signal trampoline code, and several other (minor) fix-ups that are handled by the Linux kernel module.

The Linux system call vector contains, among other things, a list of `sysent[]` entries whose addresses reside in the kernel module.

When a system call is called by the Linux binary, the trap code dereferences the system call function pointer off the `proc` structure, and gets the Linux, not the FreeBSD, system call entry points.

In addition, the Linux mode dynamically *reroots* lookups; this is, in effect, what the `union` option to file system mounts (*not* the `unionfs` file system type!) does. First, an attempt is made to lookup the file in the `/compat/linux/`*`original-path`* directory, *then* only if that fails, the lookup is done in the */`*`original-path`* directory. This makes sure that binaries that require other binaries can run (e.g., the Linux toolchain can all run under Linux ABI support). It also means that the Linux binaries can load and execute FreeBSD binaries, if there are no corresponding Linux binaries present, and that you could place a uname(1) command in the `/compat/linux` directory tree to ensure that the Linux binaries could not tell they were not running on Linux.

In effect, there is a Linux kernel in the FreeBSD kernel; the various underlying functions that implement all of the services provided by the kernel are identical to both the FreeBSD system call table entries, and the Linux system call table entries: file system operations, virtual memory operations, signal delivery, System V IPC, etc... The only difference is that FreeBSD binaries get the FreeBSD *glue* functions, and Linux binaries get the Linux *glue* functions (most older OS's only had their own *glue* functions: addresses of functions in a static global `sysent[]` structure array, instead of addresses of functions dereferenced off a dynamically initialized pointer in the `proc` structure of the process making the call).

Which one is the native FreeBSD ABI? It does not matter. Basically the only difference is that (currently; this could easily be changed in a future release, and probably will be after this) the FreeBSD *glue* functions are statically linked into the kernel, and the Linux *glue* functions can be statically linked, or they can be accessed via a kernel module.

Yeah, but is this really emulation? No. It is an ABI implementation, not an emulation. There is no emulator (or simulator, to cut off the next question) involved.

So why is it sometimes called "Linux emulation"? To make it hard to sell FreeBSD! Really, it is because the historical implementation was done at a time when there was really no word other than that to describe what was going on; saying that FreeBSD ran Linux binaries was not true, if you did not compile the code in or load a module, and there needed to be a word to describe what was being loaded——hence "the Linux emulator".

III. 系統管理

FreeBSD 使用手冊剩下的這些章節涵蓋了全方位的FreeBSD 系統管理。每個章節的開頭會先描述在該您讀完該章節後您會學到什麼，也會詳述在您在看這些資料時應該要有的一些背景知識。

這些章節是讓您在需要查資料的時候翻閱用的。您不需要依照特定的順序來讀，也不需要將這些章節全部過讀之後才開始用FreeBSD。

Chapter 11 設定與效能調校(Tuning)

Written by Chern Lee. Based on a tutorial written by Mike Smith. Also based on tuning(7) written by Matt Dillon.

11.1 概述

在FreeBSD 使用過程中，相當重要的環節之一就是系統設定部分。正確的系統設定，可以讓你減輕日後升級的頭痛壓力。本章著重於介紹FreeBSD 的相關重要設定上，包括一些可以調整FreeBSD 效能的參數設定。

讀完這章，您將了解：

- 如何有效運用檔案系統以及swap 分割區。
- `rc.conf` 的設定與`/usr/local/etc/rc.d` 的啓動架構。
- 如何設定、測試網路卡。
- 如何設定virtual hosts。
- 如何設定`/etc` 內的各種設定檔。
- 如何以`sysctl` 來調整FreeBSD 的系統效能。
- 如何調整硬碟效能，以及更改kernel 限制。

在開始閱讀這章之前，您需要：

- 瞭解UNIX 及FreeBSD 相關基本概念(Chapter 3)。
- 要有設定、編譯kernel 的基礎概念(Chapter 8)。

11.2 一開始的規劃

11.2.1 規劃分割區(Partition)

11.2.1.1 Base Partitions

用bsdlabel(8) 或sysinstall(8) 來規劃檔案系統時，請記住：硬碟在傳輸資料方面，(由於結構為碟片因素)外圈會比內圈來得快些。因此，建議把較小、常會存取的分割區儘量放外圈，而較大的分割區像是`/usr` 則應放在較內圈。建議建立分割區的順序，以像是：root, swap, `/var`, `/usr` 這樣順序來建立會較妥。

`/var` 的大小要視機器的用途而定。`/var` 是用來放信箱、log 紀錄檔以及印表機佇列(spools)。信箱以及記錄檔的成長幅度可能無法預估，因為這些成長幅度乃是取決於多少用戶、要放多久等管理原則而定。通常這些使用者並沒有用到1 GB 以上，但請切記：至少要保留一定空間給`/var/tmp` 以便存放packages。

而`/usr` 分割區主要是用來放系統運作時所需的檔案、工具程式等，例如：ports(7) collection(建議安裝)跟source tree(optional)。在安裝FreeBSD 時，這兩者都是可選擇裝與不裝的。不過，這個分割區建議至少要有2 GB 空間以上才夠用。

規劃分割區大小時，記得多保留些成長空間。否則若某個分割區滿了，但另一個分割區卻還剩很多空間，就會相當困窘。

> **Note:** 有些人可能會發現sysinstall(8) 的`Auto-defaults`(自動預設值) 所做的分割區大小，有時候會把`/var` 以及`/` 分割區設太小了。我們建議是：請依使用情況以及需求，來手動調整相關分割區大小。

11.2.1.2 Swap 分割區

根據經驗法則，通常swap 分割區應該設為系統記憶體(RAM)大小的兩倍即可。舉例來說：若機器有128 MB RAM 的話，那麼swap 則應該設為256 MB。記憶體較少的機器，可以透過增加更多swap 空間來提昇效能。我們建議swap 空間不要設低於256 MB，而且該考慮增加記憶體才是良策。當swap 最少為記憶體的兩倍大時，kernel 的VM paging 演算法會把效能調整到最佳狀態。但若是機器記憶體很大，但swap 卻劃分太少的話，會導致VM page 掃瞄的效率過低，此外日後若增加更多記憶體時，也會導致一些異常狀況發生。

在較大型的機器內，通常會有多顆SCSI 磁碟(或多顆IDE 磁碟接在不同IDE 匯流排上)，建議在每顆磁碟上都建立swap(最多到四顆)。而這些swap 應該都大約一樣大小，Kernel 可接受任意大小的swap，但內部資料結構則是最大塊swap 的4 倍。若有保持swap 為同樣大小的話，則可讓kernel 最佳化運用各磁碟之中的swap 空間。即使不太常會用到，分配大的swap 也都還可接受，因為它可在強制重開機之前讓你更容易從當掉的程式中恢復正常。

11.2.1.3 為何要規劃Partition？

有些人覺得把硬碟就直接劃分一個大分割區就好了，但是事實上有些原因會證明為何這是個爛點子，首先，每個分割區都有不同的運作特性，把它們分開的話可以讓檔案系統來調整。比如：`/` 以及`/usr` 分割區大多只是讀取而已，比較少在寫入。而讀寫都很頻繁的則是`/var` 及`/var/tmp`。

By properly partitioning a system, fragmentation introduced in the smaller write heavy partitions will not bleed over into the mostly-read partitions. Keeping the write-loaded partitions closer to the disk's edge, will increase I/O performance in the partitions where it occurs the most. Now while I/O performance in the larger partitions may be needed, shifting them more toward the edge of the disk will not lead to a significant performance improvement over moving `/var` to the edge. Finally, there are safety concerns. A smaller, neater root partition which is mostly read-only has a greater chance of surviving a bad crash.

11.3 最主要的設定檔

The principal location for system configuration information is within `/etc/rc.conf`. This file contains a wide range of configuration information, principally used at system startup to configure the system. Its name directly implies this; it is configuration information for the `rc*` files.

An administrator should make entries in the `rc.conf` file to override the default settings from `/etc/defaults/rc.conf`. The defaults file should not be copied verbatim to `/etc` - it contains default values, not examples. All system-specific changes should be made in the `rc.conf` file itself.

A number of strategies may be applied in clustered applications to separate site-wide configuration from system-specific configuration in order to keep administration overhead down. The recommended approach is to place

site-wide configuration into another file, such as `/etc/rc.conf.site`, and then include this file into `/etc/rc.conf`, which will contain only system-specific information.

As `rc.conf` is read by sh(1) it is trivial to achieve this. For example:

- rc.conf:

```
. /etc/rc.conf.site
hostname="node15.example.com"
network_interfaces="fxp0 lo0"
ifconfig_fxp0="inet 10.1.1.1"
```

- rc.conf.site:

```
defaultrouter="10.1.1.254"
saver="daemon"
blanktime="100"
```

The `rc.conf.site` file can then be distributed to every system using `rsync` or a similar program, while the `rc.conf` file remains unique.

Upgrading the system using sysinstall(8) or `make world` will not overwrite the `rc.conf` file, so system configuration information will not be lost.

11.4 各式應用程式的設定檔

原則上，安裝的軟體都會有其自有的設定檔，也會有自己的格式及語法。因此，將其與系統分開獨立是件非常重要的事情。如此一來，套件管理工具將可以很輕易的找出這些設定檔並管理這些設定檔。

原則上，設定檔會被放置在`/usr/local/etc`。若某軟體的設定檔為數衆多，那將會其下建立一個目錄以供放置

通常，當一個port 或package 被安裝的同時，一些基本的設定範例也會一併被安裝至此。這些範例通常會被用`.default` 做為副檔名。若安裝時沒有自行撰寫的軟體設定檔，那麼將會複製一份`.default` 設定做為預設設定檔

舉個例子，我們來看看`/usr/local/etc/apache`：

```
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf.default
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf.default
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic.default
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types.default
-rw-r--r--  1 root  wheel   7980 May 20  1998 srm.conf
-rw-r--r--  1 root  wheel   7933 May 20  1998 srm.conf.default
```

`srm.conf` 的檔案被修改過了，爾後**Apache** 的更新將不會對這個已修改過的設定檔做任何變動。

11.5 各種Services 的啓動方式

Contributed by Tom Rhodes.

Many users choose to install third party software on FreeBSD from the Ports Collection. In many of these situations it may be necessary to configure the software in a manner which will allow it to be started upon system initialization. Services, such as `mail/postfix` or `www/apache13` are just two of the many software packages which may be started during system initialization. This section explains the procedures available for starting third party software.

In FreeBSD, most included services, such as cron(8), are started through the system start up scripts. These scripts may differ depending on FreeBSD or vendor version; however, the most important aspect to consider is that their start up configuration can be handled through simple startup scripts.

Before the advent of `rc.d`, applications would drop a simple start up script into the `/usr/local/etc/rc.d` directory which would be read by the system initialization scripts. These scripts would then be executed during the latter stages of system start up.

While many individuals have spent hours trying to merge the old configuration style into the new system, the fact remains that some third party utilities still require a script simply dropped into the aforementioned directory. The subtle differences in the scripts depend whether or not `rc.d` is being used. Prior to FreeBSD 5.1 the old configuration style is used and in almost all cases a new style script would do just fine.

While every script must meet some minimal requirements, most of the time these requirements are FreeBSD version agnostic. Each script must have a `.sh` extension appended to the end and every script must be executable by the system. The latter may be achieved by using the `chmod` command and setting the unique permissions of `755`. There should also be, at minimal, an option to `start` the application and an option to `stop` the application.

The simplest start up script would probably look a little bit like this one:

```
#!/bin/sh
echo -n ' utility'

case "$1" in
start)
        /usr/local/bin/utility
        ;;
stop)
        kill -9 `cat /var/run/utility.pid`
        ;;
*)
        echo "Usage: `basename $0` {start|stop}" >&2
        exit 64
        ;;
esac

exit 0
```

This script provides for a `stop` and `start` option for the application hereto referred simply as `utility`.

Could be started manually with:

```
# /usr/local/etc/rc.d/utility.sh start
```

While not all third party software requires the line in `rc.conf`, almost every day a new port will be modified to accept this configuration. Check the final output of the installation for more information on a specific application.

Some third party software will provide start up scripts which permit the application to be used with `rc.d`; although, this will be discussed in the next section.

11.5.1 Extended Application Configuration

Now that FreeBSD includes `rc.d`, configuration of application startup has become easier, and more featureful. Using the key words discussed in the rc.d section, applications may now be set to start after certain other services for example DNS; may permit extra flags to be passed through `rc.conf` in place of hard coded flags in the start up script, etc. A basic script may look similar to the following:

```
#!/bin/sh
#
# PROVIDE: utility
# REQUIRE: DAEMON
# KEYWORD: shutdown

#
# DO NOT CHANGE THESE DEFAULT VALUES HERE
# SET THEM IN THE /etc/rc.conf FILE
#
utility_enable=${utility_enable-"NO"}
utility_flags=${utility_flags-""}
utility_pidfile=${utility_pidfile-"/var/run/utility.pid"}

. /etc/rc.subr

name="utility"
rcvar=‘set_rcvar‘
command="/usr/local/sbin/utility"

load_rc_config $name

pidfile="${utility_pidfile}"

start_cmd="echo \"Starting ${name}.\"; /usr/bin/nice -5 ${command} ${utility_flags} ${command_arg

run_rc_command "$1"
```

This script will ensure that the provided **utility** will be started after the `daemon` service. It also provides a method for setting and tracking the PID, or process ID file.

This application could then have the following line placed in `/etc/rc.conf`:

```
utility_enable="YES"
```

This new method also allows for easier manipulation of the command line arguments, inclusion of the default functions provided in `/etc/rc.subr`, compatibility with the rcorder(8) utility and provides for easier configuration via the `rc.conf` file.

11.5.2 以Services 來啓動各式Services

Other services, such as POP3 server daemons, IMAP, etc. could be started using the inetd(8). This involves installing the service utility from the Ports Collection with a configuration line appended to the `/etc/inetd.conf` file, or uncommenting one of the current configuration lines. Working with **inetd** and its configuration is described in depth in the inetd section.

In some cases, it may be more plausible to use the cron(8) daemon to start system services. This approach has a number of advantages because `cron` runs these processes as the `crontab`'s file owner. This allows regular users to start and maintain some applications.

The `cron` utility provides a unique feature, `@reboot`, which may be used in place of the time specification. This will cause the job to be run when cron(8) is started, normally during system initialization.

11.6 設定cron

Contributed by Tom Rhodes.

FreeBSD 最好用的工具之一就是cron(8)。`cron` 會在背景下運作，並不斷檢查`/etc/crontab` 檔以及`/var/cron/tabs` 目錄，來搜尋是否有新`crontab` 檔案。這些`crontab` 檔會存放一些排程工作的設定，來給`cron` 執行。

`cron` 程式，可同時採用兩種不同類型的設定檔：系統本身的crontab 及使用者本身的crontab。而兩種格式唯一差別在於第六欄的不同；In the system crontab, the sixth field is the name of a user for the command to run as. This gives the system crontab the ability to run commands as any user. In a user crontab, the sixth field is the command to run, and all commands run as the user who created the crontab; this is an important security feature.

> **Note:** User crontabs allow individual users to schedule tasks without the need for `root` privileges. Commands in a user's crontab run with the permissions of the user who owns the crontab.
>
> The `root` user can have a user crontab just like any other user. This one is different from `/etc/crontab` (the system crontab). Because of the system crontab, there is usually no need to create a user crontab for `root`.

Let us take a look at the `/etc/crontab` file (the system crontab):

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD: src/etc/crontab,v 1.32 2002/11/22 16:13:39 tom Exp $
# ❶
#
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ❷
HOME=/var/log
#
#
#minute hour mday month wday who command ❸
#
#
*/5 * * * * root /usr/libexec/atrun ❹
```

❶ Like most FreeBSD configuration files, the # character represents a comment. A comment can be placed in the file as a reminder of what and why a desired action is performed. Comments cannot be on the same line as a command or else they will be interpreted as part of the command; they must be on a new line. Blank lines are ignored.

❷ First, the environment must be defined. The equals (=) character is used to define any environment settings, as with this example where it is used for the `SHELL`, `PATH`, and `HOME` options. If the shell line is omitted, `cron` will use the default, which is `sh`. If the `PATH` variable is omitted, no default will be used and file locations will need to be absolute. If `HOME` is omitted, `cron` will use the invoking users home directory.

❸ This line defines a total of seven fields. Listed here are the values `minute`, `hour`, `mday`, `month`, `wday`, `who`, and `command`. These are almost all self explanatory. `minute` is the time in minutes the command will be run. `hour` is similar to the `minute` option, just in hours. `mday` stands for day of the month. `month` is similar to `hour` and `minute`, as it designates the month. The `wday` option stands for day of the week. All these fields must be numeric values, and follow the twenty-four hour clock. The `who` field is special, and only exists in the `/etc/crontab` file. This field specifies which user the command should be run as. When a user installs his or her `crontab` file, they will not have this option. Finally, the `command` option is listed. This is the last field, so naturally it should designate the command to be executed.

❹ This last line will define the values discussed above. Notice here we have a `*/5` listing, followed by several more `*` characters. These `*` characters mean “first-last”, and can be interpreted as *every* time. So, judging by this line, it is apparent that the `atrun` command is to be invoked by `root` every five minutes regardless of what day or month it is. For more information on the `atrun` command, see the atrun(8) manual page.

Commands can have any number of flags passed to them; however, commands which extend to multiple lines need to be broken with the backslash “\” continuation character.

This is the basic set up for every `crontab` file, although there is one thing different about this one. Field number six, where we specified the username, only exists in the system `/etc/crontab` file. This field should be omitted for individual user `crontab` files.

11.6.1 工作排程(Crontab)的排定與管理

> **Important:** You must not use the procedure described here to edit/install the system crontab. Simply use your favorite editor: the `cron` utility will notice that the file has changed and immediately begin using the updated version. See this FAQ entry (http://www.FreeBSD.org/doc/zh_TW.Big5/books/faq/admin.html#ROOT-NOT-FOUND-CRON-ERRORS) for more information.

To install a freshly written user `crontab`, first use your favorite editor to create a file in the proper format, and then use the `crontab` utility. The most common usage is:

```
% crontab crontab-file
```

In this example, `crontab-file` is the filename of a `crontab` that was previously created.

There is also an option to list installed `crontab` files: just pass the `-l` option to `crontab` and look over the output.

For users who wish to begin their own crontab file from scratch, without the use of a template, the `crontab -e` option is available. This will invoke the selected editor with an empty file. When the file is saved, it will be automatically installed by the `crontab` command.

If you later want to remove your user `crontab` completely, use `crontab` with the `-r` option.

11.7 在FreeBSD 使用rc

Contributed by Tom Rhodes.

從2002 年起，FreeBSD 整合了NetBSD 的`rc.d` 機制來作為系統服務啓動機制。可以到`/etc/rc.d` 目錄下去看，很多檔案都是基本服務，可以用`start`, `stop` 及`restart` 作為使用時的選項。舉個例子，可以用下列指令來重新啓動sshd(8)：

```
# /etc/rc.d/sshd restart
```

其他服務也是類似作法。當然，服務通常只要在rc.conf(5) 內有指定的話，都會在開機時就自動啓動。舉例來說，若要開機時啓動NAT(Network Address Translation) daemon 的話，只要在`/etc/rc.conf` 內加上下列這行即可：

```
natd_enable="YES"
```

若原本寫的是`natd_enable="NO"` 那麼只要把`NO` 改為`YES` 就好了。rc scripts 會在下次重開機時，自動載入相關(有相依)的服務，以下我們會講到這部分。

Since the `rc.d` system is primarily intended to start/stop services at system startup/shutdown time, the standard `start`, `stop` and `restart` options will only perform their action if the appropriate `/etc/rc.conf` variables are set. For instance the above `sshd restart` command will only work if `sshd_enable` is set to `YES` in `/etc/rc.conf`. To `start`, `stop` or `restart` a service regardless of the settings in `/etc/rc.conf`, the commands should be prefixed with “force”. For instance to restart `sshd` regardless of the current `/etc/rc.conf` setting, execute the following command:

```
# /etc/rc.d/sshd forcerestart
```

It is easy to check if a service is enabled in `/etc/rc.conf` by running the appropriate `rc.d` script with the option `rcvar`. Thus, an administrator can check that `sshd` is in fact enabled in `/etc/rc.conf` by running:

```
# /etc/rc.d/sshd rcvar
# sshd
$sshd_enable=YES
```

Note: The second line (`# sshd`) is the output from the `sshd` command, not a `root` console.

若要檢查服務是否有在運作，可以用`status` 選項來查詢。比如：若要確認`sshd` 是否真的有啓動的話，那麼打：

```
# /etc/rc.d/sshd status
sshd is running as pid 433.
```

In some cases it is also possible to `reload` a service. This will attempt to send a signal to an individual service, forcing the service to reload its configuration files. In most cases this means sending the service a `SIGHUP` signal. Support for this feature is not included for every service.

The rc.d system is not only used for network services, it also contributes to most of the system initialization. For instance, consider the bgfsck file. When this script is executed, it will print out the following message:

```
Starting background file system checks in 60 seconds.
```

Therefore this file is used for background file system checks, which are done only during system initialization.

Many system services depend on other services to function properly. For example, NIS and other RPC-based services may fail to start until after the rpcbind (portmapper) service has started. To resolve this issue, information about dependencies and other meta-data is included in the comments at the top of each startup script. The rcorder(8) program is then used to parse these comments during system initialization to determine the order in which system services should be invoked to satisfy the dependencies. The following words may be included at the top of each startup file:

- PROVIDE: Specifies the services this file provides.
- REQUIRE: Lists services which are required for this service. This file will run *after* the specified services.
- BEFORE: Lists services which depend on this service. This file will run *before* the specified services.

By using this method, an administrator can easily control system services without the hassle of “runlevels” like some other UNIX operating systems.

Additional information about the rc.d system can be found in the rc(8) and rc.subr(8) manual pages.

11.8 設定網路卡

Contributed by Marc Fonvieille.

Nowadays we can not think about a computer without thinking about a network connection. Adding and configuring a network card is a common task for any FreeBSD administrator.

11.8.1 選擇正確、可用的驅動程式(Driver)

Before you begin, you should know the model of the card you have, the chip it uses, and whether it is a PCI or ISA card. FreeBSD supports a wide variety of both PCI and ISA cards. Check the Hardware Compatibility List for your release to see if your card is supported.

Once you are sure your card is supported, you need to determine the proper driver for the card. /usr/src/sys/conf/NOTES and /usr/src/sys/*arch*/conf/NOTES will give you the list of network interface drivers with some information about the supported chipsets/cards. If you have doubts about which driver is the correct one, read the manual page of the driver. The manual page will give you more information about the supported hardware and even the possible problems that could occur.

If you own a common card, most of the time you will not have to look very hard for a driver. Drivers for common network cards are present in the GENERIC kernel, so your card should show up during boot, like so:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38
000ff irq 15 at device 11.0 on pci0
dc0: Ethernet address: 00:a0:cc:da:da:da
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
```

```
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30
000ff irq 11 at device 12.0 on pci0
dc1: Ethernet address: 00:a0:cc:da:da:db
miibus1: <MII bus> on dc1
ukphy1: <Generic IEEE 802.3u media interface> on miibus1
ukphy1:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
```

In this example, we see that two cards using the dc(4) driver are present on the system.

If the driver for your NIC is not present in `GENERIC`, you will need to load the proper driver to use your NIC. This may be accomplished in one of two ways:

- The easiest way is to simply load a kernel module for your network card with kldload(8), or automatically at boot time by adding the appropriate line to the file `/boot/loader.conf`. Not all NIC drivers are available as modules; notable examples of devices for which modules do not exist are ISA cards.
- Alternatively, you may statically compile the support for your card into your kernel. Check `/usr/src/sys/conf/NOTES`, `/usr/src/sys/`*`arch`*`/conf/NOTES` and the manual page of the driver to know what to add in your kernel configuration file. For more information about recompiling your kernel, please see Chapter 8. If your card was detected at boot by your kernel (`GENERIC`) you do not have to build a new kernel.

11.8.1.1 Using Windows NDIS Drivers

Unfortunately, there are still many vendors that do not provide schematics for their drivers to the open source community because they regard such information as trade secrets. Consequently, the developers of FreeBSD and other operating systems are left two choices: develop the drivers by a long and pain-staking process of reverse engineering or using the existing driver binaries available for the Microsoft Windows platforms. Most developers, including those involved with FreeBSD, have taken the latter approach.

Thanks to the contributions of Bill Paul (wpaul), as of FreeBSD 5.3-RELEASE there is "native" support for the Network Driver Interface Specification (NDIS). The FreeBSD NDISulator (otherwise known as Project Evil) takes a Windows driver binary and basically tricks it into thinking it is running on Windows. Because the ndis(4) driver is using a Windows binary, it is only usable on i386 and amd64 systems.

> **Note:** The ndis(4) driver is designed to support mainly PCI, CardBus and PCMCIA devices, USB devices are not yet supported.

In order to use the NDISulator, you need three things:

1. Kernel sources
2. Windows XP driver binary (`.SYS` extension)
3. Windows XP driver configuration file (`.INF` extension)

Locate the files for your specific card. Generally, they can be found on the included CDs or at the vendors' websites. In the following examples, we will use `W32DRIVER.SYS` and `W32DRIVER.INF`.

> **Note:** You can not use a Windows/i386 driver with FreeBSD/amd64, you must get a Windows/amd64 driver to make it work properly.

The next step is to compile the driver binary into a loadable kernel module. To accomplish this, as `root`, use ndisgen(8):

```
# ndisgen /path/to/W32DRIVER.INF /path/to/W32DRIVER.SYS
```

The ndisgen(8) utility is interactive and will prompt for any extra information it requires; it will produce a kernel module in the current directory which can be loaded as follows:

```
# kldload ./W32DRIVER.ko
```

In addition to the generated kernel module, you must load the `ndis.ko` and `if_ndis.ko` modules. This should be automatically done when you load any module that depends on ndis(4). If you want to load them manually, use the following commands:

```
# kldload ndis
# kldload if_ndis
```

The first command loads the NDIS miniport driver wrapper, the second loads the actual network interface.

Now, check dmesg(8) to see if there were any errors loading. If all went well, you should get output resembling the following:

```
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1
ndis0: NDIS API version: 5.0
ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5
ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps
ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps
```

From here you can treat the `ndis0` device like any other network interface (e.g., `dc0`).

You can configure the system to load the NDIS modules at boot time in the same way as with any other module. First, copy the generated module, `W32DRIVER.ko`, to the `/boot/modules` directory. Then, add the following line to `/boot/loader.conf`:

```
W32DRIVER_load="YES"
```

11.8.2 設定網路卡

Once the right driver is loaded for the network card, the card needs to be configured. As with many other things, the network card may have been configured at installation time by **sysinstall**.

To display the configuration for the network interfaces on your system, enter the following command:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        ether 00:a0:cc:da:da:da
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
dc1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255
        ether 00:a0:cc:da:da:db
```

```
        media: Ethernet 10baseT/UTP
        status: no carrier
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet 127.0.0.1 netmask 0xff000000
tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
```

> **Note:** Old versions of FreeBSD may require the `-a` option following ifconfig(8), for more details about the correct syntax of ifconfig(8), please refer to the manual page. Note also that entries concerning IPv6 (`inet6` etc.) were omitted in this example.

In this example, the following devices were displayed:

- `dc0`: The first Ethernet interface
- `dc1`: The second Ethernet interface
- `lp0`: The parallel port interface
- `lo0`: The loopback device
- `tun0`: The tunnel device used by **ppp**

FreeBSD uses the driver name followed by the order in which one the card is detected at the kernel boot to name the network card. For example `sis2` would be the third network card on the system using the sis(4) driver.

In this example, the `dc0` device is up and running. The key indicators are:

1. `UP` means that the card is configured and ready.
2. The card has an Internet (`inet`) address (in this case `192.168.1.3`).
3. It has a valid subnet mask (`netmask`; `0xffffff00` is the same as `255.255.255.0`).
4. It has a valid broadcast address (in this case, `192.168.1.255`).
5. The MAC address of the card (`ether`) is `00:a0:cc:da:da:da`
6. The physical media selection is on autoselection mode (`media: Ethernet autoselect (100baseTX <full-duplex>)`). We see that `dc1` was configured to run with `10baseT/UTP` media. For more information on available media types for a driver, please refer to its manual page.
7. The status of the link (`status`) is `active`, i.e. the carrier is detected. For `dc1`, we see `status: no carrier`. This is normal when an Ethernet cable is not plugged into the card.

If the ifconfig(8) output had shown something similar to:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
         ether 00:a0:cc:da:da:da
```

it would indicate the card has not been configured.

To configure your card, you need `root` privileges. The network card configuration can be done from the command line with ifconfig(8) but you would have to do it after each reboot of the system. The file `/etc/rc.conf` is where to add the network card's configuration.

Open /etc/rc.conf in your favorite editor. You need to add a line for each network card present on the system, for example in our case, we added these lines:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

You have to replace dc0, dc1, and so on, with the correct device for your cards, and the addresses with the proper ones. You should read the card driver and ifconfig(8) manual pages for more details about the allowed options and also rc.conf(5) manual page for more information on the syntax of /etc/rc.conf.

If you configured the network during installation, some lines about the network card(s) may be already present. Double check /etc/rc.conf before adding any lines.

You will also have to edit the file /etc/hosts to add the names and the IP addresses of various machines of the LAN, if they are not already there. For more information please refer to hosts(5) and to /usr/share/examples/etc/hosts.

11.8.3 測試與疑難排除

Once you have made the necessary changes in /etc/rc.conf, you should reboot your system. This will allow the change(s) to the interface(s) to be applied, and verify that the system restarts without any configuration errors.

Once the system has been rebooted, you should test the network interfaces.

11.8.3.1 測試乙太網路卡(Ethernet Card)

To verify that an Ethernet card is configured correctly, you have to try two things. First, ping the interface itself, and then ping another machine on the LAN.

First test the local interface:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```

Now we have to ping another machine on the LAN:

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms
```

```
--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

You could also use the machine name instead of `192.168.1.2` if you have set up the `/etc/hosts` file.

11.8.3.2 疑難排除

Troubleshooting hardware and software configurations is always a pain, and a pain which can be alleviated by checking the simple things first. Is your network cable plugged in? Have you properly configured the network services? Did you configure the firewall correctly? Is the card you are using supported by FreeBSD? Always check the hardware notes before sending off a bug report. Update your version of FreeBSD to the latest STABLE version. Check the mailing list archives, or perhaps search the Internet.

If the card works, yet performance is poor, it would be worthwhile to read over the tuning(7) manual page. You can also check the network configuration as incorrect network settings can cause slow connections.

Some users experience one or two "`device timeout`" messages, which is normal for some cards. If they continue, or are bothersome, you may wish to be sure the device is not conflicting with another device. Double check the cable connections. Perhaps you may just need to get another card.

At times, users see a few "`watchdog timeout`" errors. The first thing to do here is to check your network cable. Many cards require a PCI slot which supports Bus Mastering. On some old motherboards, only one PCI slot allows it (usually slot 0). Check the network card and the motherboard documentation to determine if that may be the problem.

"`No route to host`" messages occur if the system is unable to route a packet to the destination host. This can happen if no default route is specified, or if a cable is unplugged. Check the output of `netstat -rn` and make sure there is a valid route to the host you are trying to reach. If there is not, read on to Chapter 29.

"`ping: sendto: Permission denied`" error messages are often caused by a misconfigured firewall. If `ipfw` is enabled in the kernel but no rules have been defined, then the default policy is to deny all traffic, even ping requests! Read on to Chapter 28 for more information.

Sometimes performance of the card is poor, or below average. In these cases it is best to set the media selection mode from `autoselect` to the correct media selection. While this usually works for most hardware, it may not resolve this issue for everyone. Again, check all the network settings, and read over the tuning(7) manual page.

11.9 虛擬主機(Virtual Hosts)

A very common use of FreeBSD is virtual site hosting, where one server appears to the network as many servers. This is achieved by assigning multiple network addresses to a single interface.

A given network interface has one "real" address, and may have any number of "alias" addresses. These aliases are normally added by placing alias entries in `/etc/rc.conf`.

An alias entry for the interface `fxp0` looks like:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

Note that alias entries must start with `alias0` and proceed upwards in order, (for example, `_alias1`, `_alias2`, and so on). The configuration process will stop at the first missing number.

The calculation of alias netmasks is important, but fortunately quite simple. For a given interface, there must be one address which correctly represents the network's netmask. Any other addresses which fall within this network must have a netmask of all 1s (expressed as either `255.255.255.255` or `0xffffffff`).

For example, consider the case where the `fxp0` interface is connected to two networks, the `10.1.1.0` network with a netmask of `255.255.255.0` and the `202.0.75.16` network with a netmask of `255.255.255.240`. We want the system to appear at `10.1.1.1` through `10.1.1.5` and at `202.0.75.17` through `202.0.75.20`. As noted above, only the first address in a given network range (in this case, `10.0.1.1` and `202.0.75.17`) should have a real netmask; all the rest (`10.1.1.2` through `10.1.1.5` and `202.0.75.18` through `202.0.75.20`) must be configured with a netmask of `255.255.255.255`.

The following `/etc/rc.conf` entries configure the adapter correctly for this arrangement:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

11.10 還有哪些主要設定檔呢？

11.10.1 `/etc` Layout

There are a number of directories in which configuration information is kept. These include:

`/etc`	Generic system configuration information; data here is system-specific.
`/etc/defaults`	Default versions of system configuration files.
`/etc/mail`	Extra sendmail(8) configuration, other MTA configuration files.
`/etc/ppp`	Configuration for both user- and kernel-ppp programs.
`/etc/namedb`	Default location for named(8) data. Normally `named.conf` and zone files are stored here.
`/usr/local/etc`	Configuration files for installed applications. May contain per-application subdirectories.
`/usr/local/etc/rc.d`	Start/stop scripts for installed applications.
`/var/db`	Automatically generated system-specific database files, such as the package database, the locate database, and so on

11.10.2 Hostnames

11.10.2.1 /etc/resolv.conf

/etc/resolv.conf dictates how FreeBSD's resolver accesses the Internet Domain Name System (DNS).

The most common entries to resolv.conf are:

nameserver	The IP address of a name server the resolver should query. The servers are queried in the order listed with a maximum of three.
search	Search list for hostname lookup. This is normally determined by the domain of the local hostname.
domain	The local domain name.

A typical resolv.conf:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```

Note: Only one of the search and domain options should be used.

If you are using DHCP, dhclient(8) usually rewrites resolv.conf with information received from the DHCP server.

11.10.2.2 /etc/hosts

/etc/hosts is a simple text database reminiscent of the old Internet. It works in conjunction with DNS and NIS providing name to IP address mappings. Local computers connected via a LAN can be placed in here for simplistic naming purposes instead of setting up a named(8) server. Additionally, /etc/hosts can be used to provide a local record of Internet names, reducing the need to query externally for commonly accessed names.

```
# $FreeBSD$
#
# Host Database
# This file should contain the addresses and aliases
# for local hosts that share this file.
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the resolution order.
#
#
::1                     localhost localhost.my.domain myname.my.domain
127.0.0.1               localhost localhost.my.domain myname.my.domain

#
# Imaginary network.
#10.0.0.2               myname.my.domain myname
#10.0.0.3               myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
```

```
# private nets which will never be connected to the Internet:
#
#       10.0.0.0        -   10.255.255.255
#       172.16.0.0      -   172.31.255.255
#       192.168.0.0     -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers.  PLEASE PLEASE PLEASE do not try
# to invent your own network numbers but instead get one from your
# network provider (if any) or from the Internet Registry (ftp to
# rs.internic.net, directory '/templates').
#
```

`/etc/hosts` takes on the simple format of:

```
[Internet address] [official hostname] [alias1] [alias2] ...
```

For example:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

Consult hosts(5) for more information.

11.10.3 Log File Configuration

11.10.3.1 `syslog.conf`

`syslog.conf` is the configuration file for the syslogd(8) program. It indicates which types of `syslog` messages are logged to particular log files.

```
# $FreeBSD$
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manual page.
*.err;kern.debug;auth.notice;mail.crit          /dev/console
*.notice;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
security.*                                      /var/log/security
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
cron.*                                          /var/log/cron
*.err                                           root
*.notice;news.err                               root
*.alert                                         root
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
#*.*                                            /var/log/all.log
```

```
# uncomment this to enable logging to a remote log host named loghost
#*.*                                            @loghost
# uncomment these if you’re running inn
# news.crit                                      /var/log/news/news.crit
# news.err                                       /var/log/news/news.err
# news.notice                                    /var/log/news/news.notice
!startslip
*.*                                              /var/log/slip.log
!ppp
*.*                                              /var/log/ppp.log
```

Consult the syslog.conf(5) manual page for more information.

11.10.3.2 `newsyslog.conf`

`newsyslog.conf` is the configuration file for newsyslog(8), a program that is normally scheduled to run by cron(8). newsyslog(8) determines when log files require archiving or rearranging. `logfile` is moved to `logfile.0`, `logfile.0` is moved to `logfile.1`, and so on. Alternatively, the log files may be archived in gzip(1) format causing them to be named: `logfile.0.gz`, `logfile.1.gz`, and so on.

`newsyslog.conf` indicates which log files are to be managed, how many are to be kept, and when they are to be touched. Log files can be rearranged and/or archived when they have either reached a certain size, or at a certain periodic time/date.

```
# configuration file for newsyslog
# $FreeBSD$
#
# filename          [owner:group]    mode count size when [ZB] [/pid_file] [sig_num]
/var/log/cron                           600  3     100  *     Z
/var/log/amd.log                        644  7     100  *     Z
/var/log/kerberos.log                   644  7     100  *     Z
/var/log/lpd-errs                       644  7     100  *     Z
/var/log/maillog                        644  7     *    @T00  Z
/var/log/sendmail.st                    644  10    *    168   B
/var/log/messages                       644  5     100  *     Z
/var/log/all.log                        600  7     *    @T00  Z
/var/log/slip.log                       600  3     100  *     Z
/var/log/ppp.log                        600  3     100  *     Z
/var/log/security                       600  10    100  *     Z
/var/log/wtmp                           644  3     *    @01T05 B
/var/log/daily.log                      640  7     *    @T00  Z
/var/log/weekly.log                     640  5     1    $W6D0 Z
/var/log/monthly.log                    640  12    *    $M1D0 Z
/var/log/console.log                    640  5     100  *     Z
```

Consult the newsyslog(8) manual page for more information.

11.10.4 `sysctl.conf`

`sysctl.conf` looks much like `rc.conf`. Values are set in a `variable=value` form. The specified values are set after the system goes into multi-user mode. Not all variables are settable in this mode.

A sample `sysctl.conf` turning off logging of fatal signal exits and letting Linux programs know they are really running under FreeBSD:

```
kern.logsigexit=0       # Do not log fatal signal exits (e.g. sig 11)
compat.linux.osname=FreeBSD
compat.linux.osrelease=4.3-STABLE
```

11.11 Tuning with sysctl

sysctl(8) is an interface that allows you to make changes to a running FreeBSD system. This includes many advanced options of the TCP/IP stack and virtual memory system that can dramatically improve performance for an experienced system administrator. Over five hundred system variables can be read and set using sysctl(8).

At its core, sysctl(8) serves two functions: to read and to modify system settings.

To view all readable variables:

```
% sysctl -a
```

To read a particular variable, for example, `kern.maxproc`:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

To set a particular variable, use the intuitive *`variable=value`* syntax:

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

Settings of sysctl variables are usually either strings, numbers, or booleans (a boolean being `1` for yes or a `0` for no).

If you want to set automatically some variables each time the machine boots, add them to the `/etc/sysctl.conf` file. For more information see the sysctl.conf(5) manual page and the Section 11.10.4.

11.11.1 sysctl(8) Read-only

Contributed by Tom Rhodes.

In some cases it may be desirable to modify read-only sysctl(8) values. While this is sometimes unavoidable, it can only be done on (re)boot.

For instance on some laptop models the cardbus(4) device will not probe memory ranges, and fail with errors which look similar to:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

Cases like the one above usually require the modification of some default sysctl(8) settings which are set read only. To overcome these situations a user can put sysctl(8) “OIDs” in their local `/boot/loader.conf`. Default settings are located in the `/boot/defaults/loader.conf` file.

Fixing the problem mentioned above would require a user to set `hw.pci.allow_unsupported_io_range=1` in the aforementioned file. Now cardbus(4) will work properly.

11.12 Tuning Disks

11.12.1 Sysctl Variables

11.12.1.1 `vfs.vmiodirenable`

The `vfs.vmiodirenable` sysctl variable may be set to either 0 (off) or 1 (on); it is 1 by default. This variable controls how directories are cached by the system. Most directories are small, using just a single fragment (typically 1 K) in the file system and less (typically 512 bytes) in the buffer cache. With this variable turned off (to 0), the buffer cache will only cache a fixed number of directories even if you have a huge amount of memory. When turned on (to 1), this sysctl allows the buffer cache to use the VM Page Cache to cache the directories, making all the memory available for caching directories. However, the minimum in-core memory used to cache a directory is the physical page size (typically 4 K) rather than 512 bytes. We recommend keeping this option on if you are running any services which manipulate large numbers of files. Such services can include web caches, large mail systems, and news systems. Keeping this option on will generally not reduce performance even with the wasted memory but you should experiment to find out.

11.12.1.2 `vfs.write_behind`

The `vfs.write_behind` sysctl variable defaults to `1` (on). This tells the file system to issue media writes as full clusters are collected, which typically occurs when writing large sequential files. The idea is to avoid saturating the buffer cache with dirty buffers when it would not benefit I/O performance. However, this may stall processes and under certain circumstances you may wish to turn it off.

11.12.1.3 `vfs.hirunningspace`

The `vfs.hirunningspace` sysctl variable determines how much outstanding write I/O may be queued to disk controllers system-wide at any given instance. The default is usually sufficient but on machines with lots of disks you may want to bump it up to four or five *megabytes*. Note that setting too high a value (exceeding the buffer cache’s write threshold) can lead to extremely bad clustering performance. Do not set this value arbitrarily high! Higher write values may add latency to reads occurring at the same time.

There are various other buffer-cache and VM page cache related sysctls. We do not recommend modifying these values, the VM system does an extremely good job of automatically tuning itself.

11.12.1.4 vm.swap_idle_enabled

The vm.swap_idle_enabled sysctl variable is useful in large multi-user systems where you have lots of users entering and leaving the system and lots of idle processes. Such systems tend to generate a great deal of continuous pressure on free memory reserves. Turning this feature on and tweaking the swapout hysteresis (in idle seconds) via vm.swap_idle_threshold1 and vm.swap_idle_threshold2 allows you to depress the priority of memory pages associated with idle processes more quickly then the normal pageout algorithm. This gives a helping hand to the pageout daemon. Do not turn this option on unless you need it, because the tradeoff you are making is essentially pre-page memory sooner rather than later; thus eating more swap and disk bandwidth. In a small system this option will have a determinable effect but in a large system that is already doing moderate paging this option allows the VM system to stage whole processes into and out of memory easily.

11.12.1.5 hw.ata.wc

FreeBSD 4.3 flirted with turning off IDE write caching. This reduced write bandwidth to IDE disks but was considered necessary due to serious data consistency issues introduced by hard drive vendors. The problem is that IDE drives lie about when a write completes. With IDE write caching turned on, IDE hard drives not only write data to disk out of order, but will sometimes delay writing some blocks indefinitely when under heavy disk loads. A crash or power failure may cause serious file system corruption. FreeBSD's default was changed to be safe. Unfortunately, the result was such a huge performance loss that we changed write caching back to on by default after the release. You should check the default on your system by observing the hw.ata.wc sysctl variable. If IDE write caching is turned off, you can turn it back on by setting the kernel variable back to 1. This must be done from the boot loader at boot time. Attempting to do it after the kernel boots will have no effect.

For more information, please see ata(4).

11.12.1.6 SCSI_DELAY (kern.cam.scsi_delay)

The SCSI_DELAY kernel config may be used to reduce system boot times. The defaults are fairly high and can be responsible for 15 seconds of delay in the boot process. Reducing it to 5 seconds usually works (especially with modern drives). Newer versions of FreeBSD (5.0 and higher) should use the kern.cam.scsi_delay boot time tunable. The tunable, and kernel config option accept values in terms of *milliseconds* and *not seconds*.

11.12.2 Soft Updates

The tunefs(8) program can be used to fine-tune a file system. This program has many different options, but for now we are only concerned with toggling Soft Updates on and off, which is done by:

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

A filesystem cannot be modified with tunefs(8) while it is mounted. A good time to enable Soft Updates is before any partitions have been mounted, in single-user mode.

Soft Updates drastically improves meta-data performance, mainly file creation and deletion, through the use of a memory cache. We recommend to use Soft Updates on all of your file systems. There are two downsides to Soft Updates that you should be aware of: First, Soft Updates guarantees filesystem consistency in the case of a crash but could very easily be several seconds (even a minute!) behind updating the physical disk. If your system crashes you

may lose more work than otherwise. Secondly, Soft Updates delays the freeing of filesystem blocks. If you have a filesystem (such as the root filesystem) which is almost full, performing a major update, such as `make installworld`, can cause the filesystem to run out of space and the update to fail.

11.12.2.1 More Details about Soft Updates

There are two traditional approaches to writing a file systems meta-data back to disk. (Meta-data updates are updates to non-content data like inodes or directories.)

Historically, the default behavior was to write out meta-data updates synchronously. If a directory had been changed, the system waited until the change was actually written to disk. The file data buffers (file contents) were passed through the buffer cache and backed up to disk later on asynchronously. The advantage of this implementation is that it operates safely. If there is a failure during an update, the meta-data are always in a consistent state. A file is either created completely or not at all. If the data blocks of a file did not find their way out of the buffer cache onto the disk by the time of the crash, fsck(8) is able to recognize this and repair the filesystem by setting the file length to 0. Additionally, the implementation is clear and simple. The disadvantage is that meta-data changes are slow. An `rm -r`, for instance, touches all the files in a directory sequentially, but each directory change (deletion of a file) will be written synchronously to the disk. This includes updates to the directory itself, to the inode table, and possibly to indirect blocks allocated by the file. Similar considerations apply for unrolling large hierarchies (`tar -x`).

The second case is asynchronous meta-data updates. This is the default for Linux/ext2fs and `mount -o async` for *BSD ufs. All meta-data updates are simply being passed through the buffer cache too, that is, they will be intermixed with the updates of the file content data. The advantage of this implementation is there is no need to wait until each meta-data update has been written to disk, so all operations which cause huge amounts of meta-data updates work much faster than in the synchronous case. Also, the implementation is still clear and simple, so there is a low risk for bugs creeping into the code. The disadvantage is that there is no guarantee at all for a consistent state of the filesystem. If there is a failure during an operation that updated large amounts of meta-data (like a power failure, or someone pressing the reset button), the filesystem will be left in an unpredictable state. There is no opportunity to examine the state of the filesystem when the system comes up again; the data blocks of a file could already have been written to the disk while the updates of the inode table or the associated directory were not. It is actually impossible to implement a `fsck` which is able to clean up the resulting chaos (because the necessary information is not available on the disk). If the filesystem has been damaged beyond repair, the only choice is to use newfs(8) on it and restore it from backup.

The usual solution for this problem was to implement *dirty region logging*, which is also referred to as *journaling*, although that term is not used consistently and is occasionally applied to other forms of transaction logging as well. Meta-data updates are still written synchronously, but only into a small region of the disk. Later on they will be moved to their proper location. Because the logging area is a small, contiguous region on the disk, there are no long distances for the disk heads to move, even during heavy operations, so these operations are quicker than synchronous updates. Additionally the complexity of the implementation is fairly limited, so the risk of bugs being present is low. A disadvantage is that all meta-data are written twice (once into the logging region and once to the proper location) so for normal work, a performance “pessimization” might result. On the other hand, in case of a crash, all pending meta-data operations can be quickly either rolled-back or completed from the logging area after the system comes up again, resulting in a fast filesystem startup.

Kirk McKusick, the developer of Berkeley FFS, solved this problem with Soft Updates: all pending meta-data updates are kept in memory and written out to disk in a sorted sequence (“ordered meta-data updates”). This has the effect that, in case of heavy meta-data operations, later updates to an item “catch” the earlier ones if the earlier ones are still in memory and have not already been written to disk. So all operations on, say, a directory are generally performed in memory before the update is written to disk (the data blocks are sorted according to their position so

that they will not be on the disk ahead of their meta-data). If the system crashes, this causes an implicit “log rewind”: all operations which did not find their way to the disk appear as if they had never happened. A consistent filesystem state is maintained that appears to be the one of 30 to 60 seconds earlier. The algorithm used guarantees that all resources in use are marked as such in their appropriate bitmaps: blocks and inodes. After a crash, the only resource allocation error that occurs is that resources are marked as “used” which are actually “free”. fsck(8) recognizes this situation, and frees the resources that are no longer used. It is safe to ignore the dirty state of the filesystem after a crash by forcibly mounting it with `mount -f`. In order to free resources that may be unused, fsck(8) needs to be run at a later time. This is the idea behind the *background fsck*: at system startup time, only a *snapshot* of the filesystem is recorded. The `fsck` can be run later on. All file systems can then be mounted “dirty”, so the system startup proceeds in multiuser mode. Then, background `fsck`s will be scheduled for all file systems where this is required, to free resources that may be unused. (File systems that do not use Soft Updates still need the usual foreground `fsck` though.)

The advantage is that meta-data operations are nearly as fast as asynchronous updates (i.e. faster than with *logging*, which has to write the meta-data twice). The disadvantages are the complexity of the code (implying a higher risk for bugs in an area that is highly sensitive regarding loss of user data), and a higher memory consumption. Additionally there are some idiosyncrasies one has to get used to. After a crash, the state of the filesystem appears to be somewhat “older”. In situations where the standard synchronous approach would have caused some zero-length files to remain after the `fsck`, these files do not exist at all with a Soft Updates filesystem because neither the meta-data nor the file contents have ever been written to disk. Disk space is not released until the updates have been written to disk, which may take place some time after running `rm`. This may cause problems when installing large amounts of data on a filesystem that does not have enough free space to hold all the files twice.

11.13 Tuning Kernel Limits

11.13.1 File/Process Limits

11.13.1.1 `kern.maxfiles`

`kern.maxfiles` can be raised or lowered based upon your system requirements. This variable indicates the maximum number of file descriptors on your system. When the file descriptor table is full, “`file: table is full`” will show up repeatedly in the system message buffer, which can be viewed with the `dmesg` command.

Each open file, socket, or fifo uses one file descriptor. A large-scale production server may easily require many thousands of file descriptors, depending on the kind and number of services running concurrently.

In older FreeBSD releases, `kern.maxfile`’s default value is derived from the `maxusers` option in your dictated by the `maxusers` option in your kernel configuration file. `kern.maxfiles` grows proportionally to the value of `maxusers`. When compiling a custom kernel, it is a good idea to set this kernel configuration option according to the uses of your system. From this number, the kernel is given most of its pre-defined limits. Even though a production machine may not actually have 256 users connected at once, the resources needed may be similar to a high-scale web server.

As of FreeBSD 4.5, `kern.maxusers` is automatically sized at boot based on the amount of memory available in the system, and may be determined at run-time by inspecting the value of the read-only `kern.maxusers` sysctl. Some sites will require larger or smaller values of `kern.maxusers` and may set it as a loader tunable; values of 64, 128, and 256 are not uncommon. We do not recommend going above 256 unless you need a huge number of file

descriptors; many of the tunable values set to their defaults by `kern.maxusers` may be individually overridden at boot-time or run-time in `/boot/loader.conf` (see the loader.conf(5) man page or the `/boot/defaults/loader.conf` file for some hints) or as described elsewhere in this document. Systems older than FreeBSD 4.4 must set this value via the kernel config(8) option `maxusers` instead.

The system will auto-tune `maxusers` for you if you explicitly set it to `0`[1]. When setting this option, you will want to set `maxusers` to at least 4, especially if you are using the X Window System or compiling software. The reason is that the most important table set by `maxusers` is the maximum number of processes, which is set to `20 + 16 * maxusers`, so if you set `maxusers` to 1, then you can only have 36 simultaneous processes, including the 18 or so that the system starts up at boot time and the 15 or so you will probably create when you start the X Window System. Even a simple task like reading a manual page will start up nine processes to filter, decompress, and view it. Setting `maxusers` to 64 will allow you to have up to 1044 simultaneous processes, which should be enough for nearly all uses. If, however, you see the dreaded proc table full error when trying to start another program, or are running a server with a large number of simultaneous users (like `ftp.FreeBSD.org`), you can always increase the number and rebuild.

> **Note:** `maxusers` does *not* limit the number of users which can log into your machine. It simply sets various table sizes to reasonable values considering the maximum number of users you will likely have on your system and how many processes each of them will be running. One keyword which *does* limit the number of simultaneous remote logins and X terminal windows is `pseudo-device pty 16`. With FreeBSD 5.X, you do not have to worry about this number since the pty(4) driver is "auto-cloning"; you simply use the line `device pty` in your configuration file.

11.13.1.2 `kern.ipc.somaxconn`

The `kern.ipc.somaxconn` sysctl variable limits the size of the listen queue for accepting new TCP connections. The default value of `128` is typically too low for robust handling of new connections in a heavily loaded web server environment. For such environments, it is recommended to increase this value to `1024` or higher. The service daemon may itself limit the listen queue size (e.g. sendmail(8), or **Apache**) but will often have a directive in its configuration file to adjust the queue size. Large listen queues also do a better job of avoiding Denial of Service (DoS) attacks.

11.13.2 Network Limits

The `NMBCLUSTERS` kernel configuration option dictates the amount of network Mbufs available to the system. A heavily-trafficked server with a low number of Mbufs will hinder FreeBSD's ability. Each cluster represents approximately 2 K of memory, so a value of 1024 represents 2 megabytes of kernel memory reserved for network buffers. A simple calculation can be done to figure out how many are needed. If you have a web server which maxes out at 1000 simultaneous connections, and each connection eats a 16 K receive and 16 K send buffer, you need approximately 32 MB worth of network buffers to cover the web server. A good rule of thumb is to multiply by 2, so 2x32 MB / 2 KB = 64 MB / 2 kB = 32768. We recommend values between 4096 and 32768 for machines with greater amounts of memory. Under no circumstances should you specify an arbitrarily high value for this parameter as it could lead to a boot time crash. The `-m` option to netstat(1) may be used to observe network cluster use.

`kern.ipc.nmbclusters` loader tunable should be used to tune this at boot time. Only older versions of FreeBSD will require you to use the `NMBCLUSTERS` kernel config(8) option.

For busy servers that make extensive use of the sendfile(2) system call, it may be necessary to increase the number of sendfile(2) buffers via the `NSFBUFS` kernel configuration option or by setting its value in `/boot/loader.conf` (see loader(8) for details). A common indicator that this parameter needs to be adjusted is when processes are seen in the `sfbufa` state. The sysctl variable `kern.ipc.nsfbufs` is a read-only glimpse at the kernel configured variable. This parameter nominally scales with `kern.maxusers`, however it may be necessary to tune accordingly.

> **Important:** Even though a socket has been marked as non-blocking, calling sendfile(2) on the non-blocking socket may result in the sendfile(2) call blocking until enough `struct sf_buf`'s are made available.

11.13.2.1 `net.inet.ip.portrange.*`

The `net.inet.ip.portrange.*` sysctl variables control the port number ranges automatically bound to TCP and UDP sockets. There are three ranges: a low range, a default range, and a high range. Most network programs use the default range which is controlled by the `net.inet.ip.portrange.first` and `net.inet.ip.portrange.last`, which default to 1024 and 5000, respectively. Bound port ranges are used for outgoing connections, and it is possible to run the system out of ports under certain circumstances. This most commonly occurs when you are running a heavily loaded web proxy. The port range is not an issue when running servers which handle mainly incoming connections, such as a normal web server, or has a limited number of outgoing connections, such as a mail relay. For situations where you may run yourself out of ports, it is recommended to increase `net.inet.ip.portrange.last` modestly. A value of `10000`, `20000` or `30000` may be reasonable. You should also consider firewall effects when changing the port range. Some firewalls may block large ranges of ports (usually low-numbered ports) and expect systems to use higher ranges of ports for outgoing connections ——for this reason it is not recommended that `net.inet.ip.portrange.first` be lowered.

11.13.2.2 TCP Bandwidth Delay Product

The TCP Bandwidth Delay Product Limiting is similar to TCP/Vegas in NetBSD. It can be enabled by setting `net.inet.tcp.inflight.enable` sysctl variable to `1`. The system will attempt to calculate the bandwidth delay product for each connection and limit the amount of data queued to the network to just the amount required to maintain optimum throughput.

This feature is useful if you are serving data over modems, Gigabit Ethernet, or even high speed WAN links (or any other link with a high bandwidth delay product), especially if you are also using window scaling or have configured a large send window. If you enable this option, you should also be sure to set `net.inet.tcp.inflight.debug` to `0` (disable debugging), and for production use setting `net.inet.tcp.inflight.min` to at least `6144` may be beneficial. However, note that setting high minimums may effectively disable bandwidth limiting depending on the link. The limiting feature reduces the amount of data built up in intermediate route and switch packet queues as well as reduces the amount of data built up in the local host's interface queue. With fewer packets queued up, interactive connections, especially over slow modems, will also be able to operate with lower *Round Trip Times*. However, note that this feature only effects data transmission (uploading / server side). It has no effect on data reception (downloading).

Adjusting `net.inet.tcp.inflight.stab` is *not* recommended. This parameter defaults to 20, representing 2 maximal packets added to the bandwidth delay product window calculation. The additional window is required to stabilize the algorithm and improve responsiveness to changing conditions, but it can also result in higher ping times over slow links (though still much lower than you would get without the inflight algorithm). In such cases, you may

wish to try reducing this parameter to 15, 10, or 5; and may also have to reduce `net.inet.tcp.inflight.min` (for example, to 3500) to get the desired effect. Reducing these parameters should be done as a last resort only.

11.13.3 Virtual Memory

11.13.3.1 `kern.maxvnodes`

A vnode is the internal representation of a file or directory. So increasing the number of vnodes available to the operating system cuts down on disk I/O. Normally this is handled by the operating system and does not need to be changed. In some cases where disk I/O is a bottleneck and the system is running out of vnodes, this setting will need to be increased. The amount of inactive and free RAM will need to be taken into account.

To see the current number of vnodes in use:

```
# sysctl vfs.numvnodes
vfs.numvnodes: 91349
```

To see the maximum vnodes:

```
# sysctl kern.maxvnodes
kern.maxvnodes: 100000
```

If the current vnode usage is near the maximum, increasing `kern.maxvnodes` by a value of 1,000 is probably a good idea. Keep an eye on the number of `vfs.numvnodes`. If it climbs up to the maximum again, `kern.maxvnodes` will need to be increased further. A shift in your memory usage as reported by top(1) should be visible. More memory should be active.

11.14 Adding Swap Space

No matter how well you plan, sometimes a system does not run as you expect. If you find you need more swap space, it is simple enough to add. You have three ways to increase swap space: adding a new hard drive, enabling swap over NFS, and creating a swap file on an existing partition.

For information on how to encrypt swap space, what options for this task exist and why it should be done, please refer to Section 18.17 of the Handbook.

11.14.1 Swap on a New Hard Drive

The best way to add swap, of course, is to use this as an excuse to add another hard drive. You can always use another hard drive, after all. If you can do this, go reread the discussion of swap space in Section 11.2 of the Handbook for some suggestions on how to best arrange your swap.

11.14.2 Swapping over NFS

Swapping over NFS is only recommended if you do not have a local hard disk to swap to; NFS swapping will be limited by the available network bandwidth and puts an additional burden on the NFS server.

11.14.3 Swapfiles

You can create a file of a specified size to use as a swap file. In our example here we will use a 64MB file called `/usr/swap0`. You can use any name you want, of course.

Example 11-1. Creating a Swapfile on FreeBSD

1. Be certain that your kernel configuration includes the memory disk driver (md(4)). It is default in `GENERIC` kernel.
   ```
   device   md   # Memory "disks"
   ```
2. Create a swapfile (`/usr/swap0`):
   ```
   # dd if=/dev/zero of=/usr/swap0 bs=1024k count=64
   ```
3. Set proper permissions on (`/usr/swap0`):
   ```
   # chmod 0600 /usr/swap0
   ```
4. Enable the swap file in `/etc/rc.conf`:
   ```
   swapfile="/usr/swap0"   # Set to name of swapfile if aux swapfile desired.
   ```
5. Reboot the machine or to enable the swap file immediately, type:
   ```
   # mdconfig -a -t vnode -f /usr/swap0 -u 0 && swapon /dev/md0
   ```

11.15 Power and Resource Management

Written by Hiten Pandya and Tom Rhodes.

It is very important to utilize hardware resources in an efficient manner. Before ACPI was introduced, it was very difficult and inflexible for operating systems to manage the power usage and thermal properties of a system. The hardware was controlled by some sort of BIOS embedded interface, such as *Plug and Play BIOS (PNPBIOS)*, or *Advanced Power Management (APM)* and so on. Power and Resource Management is one of the key components of a modern operating system. For example, you may want an operating system to monitor system limits (and possibly alert you) in case your system temperature increased unexpectedly.

In this section of the FreeBSD Handbook, we will provide comprehensive information about ACPI. References will be provided for further reading at the end.

11.15.1 What Is ACPI?

Advanced Configuration and Power Interface (ACPI) is a standard written by an alliance of vendors to provide a standard interface for hardware resources and power management (hence the name). It is a key element in *Operating System-directed configuration and Power Management*, i.e.: it provides more control and flexibility to the operating

system (OS). Modern systems “stretched” the limits of the current Plug and Play interfaces prior to the introduction of ACPI. ACPI is the direct successor to APM (Advanced Power Management).

11.15.2 Shortcomings of Advanced Power Management (APM)

The *Advanced Power Management (APM)* facility controls the power usage of a system based on its activity. The APM BIOS is supplied by the (system) vendor and it is specific to the hardware platform. An APM driver in the OS mediates access to the *APM Software Interface*, which allows management of power levels.

There are four major problems in APM. Firstly, power management is done by the (vendor-specific) BIOS, and the OS does not have any knowledge of it. One example of this, is when the user sets idle-time values for a hard drive in the APM BIOS, that when exceeded, it (BIOS) would spin down the hard drive, without the consent of the OS. Secondly, the APM logic is embedded in the BIOS, and it operates outside the scope of the OS. This means users can only fix problems in their APM BIOS by flashing a new one into the ROM; which is a very dangerous procedure with the potential to leave the system in an unrecoverable state if it fails. Thirdly, APM is a vendor-specific technology, which means that there is a lot of parity (duplication of efforts) and bugs found in one vendor’s BIOS, may not be solved in others. Last but not the least, the APM BIOS did not have enough room to implement a sophisticated power policy, or one that can adapt very well to the purpose of the machine.

Plug and Play BIOS (PNPBIOS) was unreliable in many situations. PNPBIOS is 16-bit technology, so the OS has to use 16-bit emulation in order to “interface” with PNPBIOS methods.

The FreeBSD APM driver is documented in the apm(4) manual page.

11.15.3 Configuring ACPI

The `acpi.ko` driver is loaded by default at start up by the loader(8) and should *not* be compiled into the kernel. The reasoning behind this is that modules are easier to work with, say if switching to another `acpi.ko` without doing a kernel rebuild. This has the advantage of making testing easier. Another reason is that starting ACPI after a system has been brought up is not too useful, and in some cases can be fatal. In doubt, just disable ACPI all together. This driver should not and can not be unloaded because the system bus uses it for various hardware interactions. ACPI can be disabled with the acpiconf(8) utility. In fact most of the interaction with ACPI can be done via acpiconf(8). Basically this means, if anything about ACPI is in the dmesg(8) output, then most likely it is already running.

> **Note:** ACPI and APM cannot coexist and should be used separately. The last one to load will terminate if the driver notices the other running.

In the simplest form, ACPI can be used to put the system into a sleep mode with acpiconf(8), the `-s` flag, and a `1-5` option. Most users will only need `1`. Option `5` will do a soft-off which is the same action as:

```
# halt -p
```

The other options are available. Check out the acpiconf(8) manual page for more information.

11.16 Using and Debugging FreeBSD ACPI

Written by Nate Lawson. With contributions from Peter Schultz and Tom Rhodes.

ACPI is a fundamentally new way of discovering devices, managing power usage, and providing standardized access to various hardware previously managed by the BIOS. Progress is being made toward ACPI working on all systems, but bugs in some motherboards' *ACPI Machine Language* (AML) bytecode, incompleteness in FreeBSD's kernel subsystems, and bugs in the Intel ACPI-CA interpreter continue to appear.

This document is intended to help you assist the FreeBSD ACPI maintainers in identifying the root cause of problems you observe and debugging and developing a solution. Thanks for reading this and we hope we can solve your system's problems.

11.16.1 Submitting Debugging Information

> **Note:** Before submitting a problem, be sure you are running the latest BIOS version and, if available, embedded controller firmware version.

For those of you that want to submit a problem right away, please send the following information to freebsd-acpi@FreeBSD.org (mailto:freebsd-acpi@FreeBSD.org):

- Description of the buggy behavior, including system type and model and anything that causes the bug to appear. Also, please note as accurately as possible when the bug began occurring if it is new for you.
- The dmesg(8) output after `boot -v`, including any error messages generated by you exercising the bug.
- The dmesg(8) output from `boot -v` with ACPI disabled, if disabling it helps fix the problem.
- Output from `sysctl hw.acpi`. This is also a good way of figuring out what features your system offers.
- URL where your *ACPI Source Language* (ASL) can be found. Do *not* send the ASL directly to the list as it can be very large. Generate a copy of your ASL by running this command:

  ```
  # acpidump -t -d > name-system.asl
  ```

  (Substitute your login name for *`name`* and manufacturer/model for *`system`*. Example: `njl-FooCo6000.asl`)

Most of the developers watch the FreeBSD-CURRENT 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) but please submit problems to freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi) to be sure it is seen. Please be patient, all of us have full-time jobs elsewhere. If your bug is not immediately apparent, we will probably ask you to submit a PR via send-pr(1). When entering a PR, please include the same information as requested above. This will help us track the problem and resolve it. Do not send a PR without emailing freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi) first as we use PRs as reminders of existing problems, not a reporting mechanism. It is likely that your problem has been reported by someone before.

11.16.2 Background

ACPI is present in all modern computers that conform to the ia32 (x86), ia64 (Itanium), and amd64 (AMD) architectures. The full standard has many features including CPU performance management, power planes control, thermal zones, various battery systems, embedded controllers, and bus enumeration. Most systems implement less

than the full standard. For instance, a desktop system usually only implements the bus enumeration parts while a laptop might have cooling and battery management support as well. Laptops also have suspend and resume, with their own associated complexity.

An ACPI-compliant system has various components. The BIOS and chipset vendors provide various fixed tables (e.g., FADT) in memory that specify things like the APIC map (used for SMP), config registers, and simple configuration values. Additionally, a table of bytecode (the *Differentiated System Description Table* DSDT) is provided that specifies a tree-like name space of devices and methods.

The ACPI driver must parse the fixed tables, implement an interpreter for the bytecode, and modify device drivers and the kernel to accept information from the ACPI subsystem. For FreeBSD, Intel has provided an interpreter (ACPI-CA) that is shared with Linux and NetBSD. The path to the ACPI-CA source code is `src/sys/contrib/dev/acpica`. The glue code that allows ACPI-CA to work on FreeBSD is in `src/sys/dev/acpica/Osd`. Finally, drivers that implement various ACPI devices are found in `src/sys/dev/acpica`.

11.16.3 Common Problems

For ACPI to work correctly, all the parts have to work correctly. Here are some common problems, in order of frequency of appearance, and some possible workarounds or fixes.

11.16.3.1 Mouse Issues

In some cases, resuming from a suspend operation will cause the mouse to fail. A known work around is to add `hint.psm.0.flags="0x3000"` to the `/boot/loader.conf` file. If this does not work then please consider sending a bug report as described above.

11.16.3.2 Suspend/Resume

ACPI has three suspend to RAM (STR) states, `S1`-`S3`, and one suspend to disk state (`STD`), called `S4`. `S5` is "soft off" and is the normal state your system is in when plugged in but not powered up. `S4` can actually be implemented two separate ways. `S4`BIOS is a BIOS-assisted suspend to disk. `S4`OS is implemented entirely by the operating system.

Start by checking `sysctl hw.acpi` for the suspend-related items. Here are the results for a Thinkpad:

```
hw.acpi.supported_sleep_state: S3 S4 S5
hw.acpi.s4bios: 0
```

This means that we can use `acpiconf -s` to test `S3`, `S4`OS, and `S5`. If `s4bios` was one (`1`), we would have `S4`BIOS support instead of `S4` OS.

When testing suspend/resume, start with `S1`, if supported. This state is most likely to work since it does not require much driver support. No one has implemented `S2` but if you have it, it is similar to `S1`. The next thing to try is `S3`. This is the deepest STR state and requires a lot of driver support to properly reinitialize your hardware. If you have problems resuming, feel free to email the freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi) list but do not expect the problem to be resolved since there are a lot of drivers/hardware that need more testing and work.

To help isolate the problem, remove as many drivers from your kernel as possible. If it works, you can narrow down which driver is the problem by loading drivers until it fails again. Typically binary drivers like `nvidia.ko`, X11 display drivers, and USB will have the most problems while Ethernet interfaces usually work fine. If you can

properly load/unload the drivers, you can automate this by putting the appropriate commands in `/etc/rc.suspend` and `/etc/rc.resume`. There is a commented-out example for unloading and loading a driver. Try setting `hw.acpi.reset_video` to zero (`0`) if your display is messed up after resume. Try setting longer or shorter values for `hw.acpi.sleep_delay` to see if that helps.

Another thing to try is load a recent Linux distribution with ACPI support and test their suspend/resume support on the same hardware. If it works on Linux, it is likely a FreeBSD driver problem and narrowing down which driver causes the problems will help us fix the problem. Note that the ACPI maintainers do not usually maintain other drivers (e.g sound, ATA, etc.) so any work done on tracking down a driver problem should probably eventually be posted to the freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) list and mailed to the driver maintainer. If you are feeling adventurous, go ahead and start putting some debugging printf(3)s in a problematic driver to track down where in its resume function it hangs.

Finally, try disabling ACPI and enabling APM instead. If suspend/resume works with APM, you may be better off sticking with APM, especially on older hardware (pre-2000). It took vendors a while to get ACPI support correct and older hardware is more likely to have BIOS problems with ACPI.

11.16.3.3 System Hangs (temporary or permanent)

Most system hangs are a result of lost interrupts or an interrupt storm. Chipsets have a lot of problems based on how the BIOS configures interrupts before boot, correctness of the APIC (MADT) table, and routing of the *System Control Interrupt* (SCI).

Interrupt storms can be distinguished from lost interrupts by checking the output of `vmstat -i` and looking at the line that has `acpi0`. If the counter is increasing at more than a couple per second, you have an interrupt storm. If the system appears hung, try breaking to DDB (**CTRL**+**ALT**+**ESC** on console) and type `show interrupts`.

Your best hope when dealing with interrupt problems is to try disabling APIC support with `hint.apic.0.disabled="1"` in `loader.conf`.

11.16.3.4 Panics

Panics are relatively rare for ACPI and are the top priority to be fixed. The first step is to isolate the steps to reproduce the panic (if possible) and get a backtrace. Follow the advice for enabling `options DDB` and setting up a serial console (see Section 24.6.5.3) or setting up a dump(8) partition. You can get a backtrace in DDB with `tr`. If you have to handwrite the backtrace, be sure to at least get the lowest five (5) and top five (5) lines in the trace.

Then, try to isolate the problem by booting with ACPI disabled. If that works, you can isolate the ACPI subsystem by using various values of `debug.acpi.disable`. See the acpi(4) manual page for some examples.

11.16.3.5 System Powers Up After Suspend or Shutdown

First, try setting `hw.acpi.disable_on_poweroff="0"` in loader.conf(5). This keeps ACPI from disabling various events during the shutdown process. Some systems need this value set to `1` (the default) for the same reason. This usually fixes the problem of a system powering up spontaneously after a suspend or poweroff.

11.16.3.6 Other Problems

If you have other problems with ACPI (working with a docking station, devices not detected, etc.), please email a description to the mailing list as well; however, some of these issues may be related to unfinished parts of the ACPI

subsystem so they might take a while to be implemented. Please be patient and prepared to test patches we may send you.

11.16.4 ASL, `acpidump`, and IASL

The most common problem is the BIOS vendors providing incorrect (or outright buggy!) bytecode. This is usually manifested by kernel console messages like this:

```
ACPI-1287: *** Error: Method execution failed [\\_SB_.PCI0.LPC0.FIGD._STA] \\
(Node 0xc3f6d160), AE_NOT_FOUND
```

Often, you can resolve these problems by updating your BIOS to the latest revision. Most console messages are harmless but if you have other problems like battery status not working, they are a good place to start looking for problems in the AML. The bytecode, known as AML, is compiled from a source language called ASL. The AML is found in the table known as the DSDT. To get a copy of your ASL, use acpidump(8). You should use both the `-t` (show contents of the fixed tables) and `-d` (disassemble AML to ASL) options. See the Submitting Debugging Information section for an example syntax.

The simplest first check you can do is to recompile your ASL to check for errors. Warnings can usually be ignored but errors are bugs that will usually prevent ACPI from working correctly. To recompile your ASL, issue the following command:

```
# iasl your.asl
```

11.16.5 Fixing Your ASL

In the long run, our goal is for almost everyone to have ACPI work without any user intervention. At this point, however, we are still developing workarounds for common mistakes made by the BIOS vendors. The Microsoft interpreter (`acpi.sys` and `acpiec.sys`) does not strictly check for adherence to the standard, and thus many BIOS vendors who only test ACPI under Windows never fix their ASL. We hope to continue to identify and document exactly what non-standard behavior is allowed by Microsoft's interpreter and replicate it so FreeBSD can work without forcing users to fix the ASL. As a workaround and to help us identify behavior, you can fix the ASL manually. If this works for you, please send a diff(1) of the old and new ASL so we can possibly work around the buggy behavior in ACPI-CA and thus make your fix unnecessary.

Here is a list of common error messages, their cause, and how to fix them:

11.16.5.1 _OS dependencies

Some AML assumes the world consists of various Windows versions. You can tell FreeBSD to claim it is any OS to see if this fixes problems you may have. An easy way to override this is to set `hw.acpi.osname="Windows 2001"` in `/boot/loader.conf` or other similar strings you find in the ASL.

11.16.5.2 Missing Return statements

Some methods do not explicitly return a value as the standard requires. While ACPI-CA does not handle this, FreeBSD has a workaround that allows it to return the value implicitly. You can also add explicit Return statements where required if you know what value should be returned. To force `iasl` to compile the ASL, use the `-f` flag.

11.16.5.3 Overriding the Default AML

After you customize `your.asl`, you will want to compile it, run:

```
# iasl your.asl
```

You can add the `-f` flag to force creation of the AML, even if there are errors during compilation. Remember that some errors (e.g., missing Return statements) are automatically worked around by the interpreter.

`DSDT.aml` is the default output filename for `iasl`. You can load this instead of your BIOS's buggy copy (which is still present in flash memory) by editing `/boot/loader.conf` as follows:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

Be sure to copy your `DSDT.aml` to the `/boot` directory.

11.16.6 Getting Debugging Output From ACPI

The ACPI driver has a very flexible debugging facility. It allows you to specify a set of subsystems as well as the level of verbosity. The subsystems you wish to debug are specified as "layers" and are broken down into ACPI-CA components (ACPI_ALL_COMPONENTS) and ACPI hardware support (ACPI_ALL_DRIVERS). The verbosity of debugging output is specified as the "level" and ranges from ACPI_LV_ERROR (just report errors) to ACPI_LV_VERBOSE (everything). The "level" is a bitmask so multiple options can be set at once, separated by spaces. In practice, you will want to use a serial console to log the output if it is so long it flushes the console message buffer. A full list of the individual layers and levels is found in the acpi(4) manual page.

Debugging output is not enabled by default. To enable it, add `options ACPI_DEBUG` to your kernel configuration file if ACPI is compiled into the kernel. You can add `ACPI_DEBUG=1` to your `/etc/make.conf` to enable it globally. If it is a module, you can recompile just your `acpi.ko` module as follows:

```
# cd /sys/modules/acpi/acpi
&& make clean &&
make ACPI_DEBUG=1
```

Install `acpi.ko` in `/boot/kernel` and add your desired level and layer to `loader.conf`. This example enables debug messages for all ACPI-CA components and all ACPI hardware drivers (CPU, LID, etc.) It will only output error messages, the least verbose level.

```
debug.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debug.acpi.level="ACPI_LV_ERROR"
```

If the information you want is triggered by a specific event (say, a suspend and then resume), you can leave out changes to `loader.conf` and instead use `sysctl` to specify the layer and level after booting and preparing your system for the specific event. The `sysctl`s are named the same as the tunables in `loader.conf`.

11.16.7 References

More information about ACPI may be found in the following locations:

- The FreeBSD ACPI 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi)
- The ACPI Mailing List Archives http://lists.freebsd.org/pipermail/freebsd-acpi/
- The old ACPI Mailing List Archives http://home.jp.FreeBSD.org/mail-list/acpi-jp/
- The ACPI 2.0 Specification http://acpi.info/spec.htm
- FreeBSD Manual pages: acpi(4), acpi_thermal(4), acpidump(8), iasl(8), acpidb(8)
- DSDT debugging resource (http://www.cpqlinux.com/acpi-howto.html#fix_broken_dsdt). (Uses Compaq as an example but generally useful.)

Notes

1. The auto-tuning algorithm sets `maxusers` equal to the amount of memory in the system, with a minimum of 32, and a maximum of 384.

Chapter 12 FreeBSD 開機流程篇

12.1 概述

The process of starting a computer and loading the operating system is referred to as “the bootstrap process”, or simply “booting”. FreeBSD’s boot process provides a great deal of flexibility in customizing what happens when you start the system, allowing you to select from different operating systems installed on the same computer, or even different versions of the same operating system or installed kernel.

This chapter details the configuration options you can set and how to customize the FreeBSD boot process. This includes everything that happens until the FreeBSD kernel has started, probed for devices, and started init(8). If you are not quite sure when this happens, it occurs when the text color changes from bright white to grey.

讀完這章，您將了解：

- What the components of the FreeBSD bootstrap system are, and how they interact.
- The options you can give to the components in the FreeBSD bootstrap to control the boot process.
- device.hints(5) 的基本概念。

> **x86 Only:** This chapter only describes the boot process for FreeBSD running on Intel x86 systems.

12.2 Booting 問題

Turning on a computer and starting the operating system poses an interesting dilemma. By definition, the computer does not know how to do anything until the operating system is started. This includes running programs from the disk. So if the computer can not run a program from the disk without the operating system, and the operating system programs are on the disk, how is the operating system started?

This problem parallels one in the book *The Adventures of Baron Munchausen*. A character had fallen part way down a manhole, and pulled himself out by grabbing his bootstraps, and lifting. In the early days of computing the term *bootstrap* was applied to the mechanism used to load the operating system, which has become shortened to “booting”.

On x86 hardware the Basic Input/Output System (BIOS) is responsible for loading the operating system. To do this, the BIOS looks on the hard disk for the Master Boot Record (MBR), which must be located on a specific place on the disk. The BIOS has enough knowledge to load and run the MBR, and assumes that the MBR can then carry out the rest of the tasks involved in loading the operating system, possibly with the help of the BIOS.

The code within the MBR is usually referred to as a *boot manager*, especially when it interacts with the user. In this case the boot manager usually has more code in the first *track* of the disk or within some OS’s file system. (A boot manager is sometimes also called a *boot loader*, but FreeBSD uses that term for a later stage of booting.) Popular boot managers include **boot0** (a.k.a. **Boot Easy**, the standard FreeBSD boot manager), **Grub**, **GAG**, and **LILO**. (Only **boot0** fits within the MBR.)

If you have only one operating system installed on your disks then a standard PC MBR will suffice. This MBR searches for the first bootable (a.k.a. active) slice on the disk, and then runs the code on that slice to load the

remainder of the operating system. The MBR installed by fdisk(8), by default, is such an MBR. It is based on `/boot/mbr`.

If you have installed multiple operating systems on your disks then you can install a different boot manager, one that can display a list of different operating systems, and allows you to choose the one to boot from. Two of these are discussed in the next subsection.

The remainder of the FreeBSD bootstrap system is divided into three stages. The first stage is run by the MBR, which knows just enough to get the computer into a specific state and run the second stage. The second stage can do a little bit more, before running the third stage. The third stage finishes the task of loading the operating system. The work is split into these three stages because the PC standards put limits on the size of the programs that can be run at stages one and two. Chaining the tasks together allows FreeBSD to provide a more flexible loader.

The kernel is then started and it begins to probe for devices and initialize them for use. Once the kernel boot process is finished, the kernel passes control to the user process init(8), which then makes sure the disks are in a usable state. init(8) then starts the user-level resource configuration which mounts file systems, sets up network cards to communicate on the network, and generally starts all the processes that usually are run on a FreeBSD system at startup.

12.3 The Boot Manager and Boot Stages

12.3.1 The Boot Manager

The code in the MBR or boot manager is sometimes referred to as *stage zero* of the boot process. This subsection discusses two of the boot managers previously mentioned: **boot0** and **LILO**.

The boot0 Boot Manager: The MBR installed by FreeBSD's installer or boot0cfg(8), by default, is based on `/boot/boot0`. (The **boot0** program is very simple, since the program in the MBR can only be 446 bytes long because of the slice table and 0x55AA identifier at the end of the MBR.) If you have installed **boot0** and multiple operating systems on your hard disks, then you will see a display similar to this one at boot time:

Example 12-1. `boot0` Screenshot

```
F1 DOS
F2 FreeBSD
F3 Linux
F4 ??
F5 Drive 1

Default: F2
```

Other operating systems, in particular Windows, have been known to overwrite an existing MBR with their own. If this happens to you, or you want to replace your existing MBR with the FreeBSD MBR then use the following command:

```
# fdisk -B -b /boot/boot0 device
```

where *`device`* is the device that you boot from, such as `ad0` for the first IDE disk, `ad2` for the first IDE disk on a second IDE controller, `da0` for the first SCSI disk, and so on. Or, if you want a custom configuration of the MBR, use boot0cfg(8).

The LILO Boot Manager: To install this boot manager so it will also boot FreeBSD, first start Linux and add the following to your existing `/etc/lilo.conf` configuration file:

```
other=/dev/hdXY
table=/dev/hdX
loader=/boot/chain.b
label=FreeBSD
```

In the above, specify FreeBSD's primary partition and drive using Linux specifiers, replacing *`X`* with the Linux drive letter and *`Y`* with the Linux primary partition number. If you are using a SCSI drive, you will need to change */dev/hd* to read something similar to */dev/sd*. The `loader=/boot/chain.b` line can be omitted if you have both operating systems on the same drive. Now run `/sbin/lilo -v` to commit your new changes to the system; this should be verified by checking its screen messages.

12.3.2 Stage One, `/boot/boot1`, and Stage Two, `/boot/boot2`

Conceptually the first and second stages are part of the same program, on the same area of the disk. Because of space constraints they have been split into two, but you would always install them together. They are copied from the combined file `/boot/boot` by the installer or **disklabel** (see below).

They are located outside file systems, in the first track of the boot slice, starting with the first sector. This is where boot0, or any other boot manager, expects to find a program to run which will continue the boot process. The number of sectors used is easily determined from the size of `/boot/boot`.

`boot1` is very simple, since it can only be 512 bytes in size, and knows just enough about the FreeBSD *disklabel*, which stores information about the slice, to find and execute `boot2`.

`boot2` is slightly more sophisticated, and understands the FreeBSD file system enough to find files on it, and can provide a simple interface to choose the kernel or loader to run.

Since the loader is much more sophisticated, and provides a nice easy-to-use boot configuration, `boot2` usually runs it, but previously it was tasked to run the kernel directly.

Example 12-2. `boot2` Screenshot

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/kernel
boot:
```

If you ever need to replace the installed `boot1` and `boot2` use disklabel(8):

```
# disklabel -B diskslice
```

where *`diskslice`* is the disk and slice you boot from, such as `ad0s1` for the first slice on the first IDE disk.

> **Dangerously Dedicated Mode:** If you use just the disk name, such as `ad0`, in the disklabel(8) command you will create a dangerously dedicated disk, without slices. This is almost certainly not what you want to do, so make sure you double check the disklabel(8) command before you press **Return**.

12.3.3 Stage Three, `/boot/loader`

The loader is the final stage of the three-stage bootstrap, and is located on the file system, usually as `/boot/loader`.

The loader is intended as a user-friendly method for configuration, using an easy-to-use built-in command set, backed up by a more powerful interpreter, with a more complex command set.

12.3.3.1 Loader Program Flow

During initialization, the loader will probe for a console and for disks, and figure out what disk it is booting from. It will set variables accordingly, and an interpreter is started where user commands can be passed from a script or interactively.

The loader will then read `/boot/loader.rc`, which by default reads in `/boot/defaults/loader.conf` which sets reasonable defaults for variables and reads `/boot/loader.conf` for local changes to those variables. `loader.rc` then acts on these variables, loading whichever modules and kernel are selected.

Finally, by default, the loader issues a 10 second wait for key presses, and boots the kernel if it is not interrupted. If interrupted, the user is presented with a prompt which understands the easy-to-use command set, where the user may adjust variables, unload all modules, load modules, and then finally boot or reboot.

12.3.3.2 Loader Built-In Commands

These are the most commonly used loader commands. For a complete discussion of all available commands, please see loader(8).

autoboot *`seconds`*

> Proceeds to boot the kernel if not interrupted within the time span given, in seconds. It displays a countdown, and the default time span is 10 seconds.

boot [*`-options`*] [*`kernelname`*]

> Immediately proceeds to boot the kernel, with the given options, if any, and with the kernel name given, if it is.

boot-conf

> Goes through the same automatic configuration of modules based on variables as what happens at boot. This only makes sense if you use `unload` first, and change some variables, most commonly `kernel`.

help [*`topic`*]

> Shows help messages read from `/boot/loader.help`. If the topic given is `index`, then the list of available topics is given.

include *`filename ...`*

> Processes the file with the given filename. The file is read in, and interpreted line by line. An error immediately stops the include command.

load [`-t` *`type`*] *`filename`*

> Loads the kernel, kernel module, or file of the type given, with the filename given. Any arguments after filename are passed to the file.

ls [-l] [*path*]

Displays a listing of files in the given path, or the root directory, if the path is not specified. If -l is specified, file sizes will be shown too.

lsdev [-v]

Lists all of the devices from which it may be possible to load modules. If -v is specified, more details are printed.

lsmod [-v]

Displays loaded modules. If -v is specified, more details are shown.

more *filename*

Displays the files specified, with a pause at each LINES displayed.

reboot

Immediately reboots the system.

set *variable*
set *variable*=*value*

Sets the loader's environment variables.

unload

Removes all loaded modules.

12.3.3.3 Loader Examples

Here are some practical examples of loader usage:

- To simply boot your usual kernel, but in single-user mode:

  ```
  boot -s
  ```

- To unload your usual kernel and modules, and then load just your old (or another) kernel:

  ```
  unload
  load kernel.old
  ```

  You can use kernel.GENERIC to refer to the generic kernel that comes on the install disk, or kernel.old to refer to your previously installed kernel (when you have upgraded or configured your own kernel, for example).

  > **Note:** Use the following to load your usual modules with another kernel:
  >
  > ```
  > unload
  > set kernel="kernel.old"
  > boot-conf
  > ```

- To load a kernel configuration script (an automated script which does the things you would normally do in the kernel boot-time configurator):

```
load -t userconfig_script /boot/kernel.conf
```

12.4 Kernel Interaction During Boot

Once the kernel is loaded by either loader (as usual) or boot2 (bypassing the loader), it examines its boot flags, if any, and adjusts its behavior as necessary.

12.4.1 Kernel Boot Flags

Here are the more common boot flags:

`-a`

during kernel initialization, ask for the device to mount as the root file system.

`-C`

boot from CDROM.

`-c`

run UserConfig, the boot-time kernel configurator

`-s`

boot into single-user mode

`-v`

be more verbose during kernel startup

Note: There are other boot flags, read boot(8) for more information on them.

12.5 Device Hints

Contributed by Tom Rhodes.

Note: This is a FreeBSD 5.0 and later feature which does not exist in earlier versions.

During initial system startup, the boot loader(8) will read the device.hints(5) file. This file stores kernel boot information known as variables, sometimes referred to as “device hints”. These “device hints” are used by device drivers for device configuration.

Device hints may also be specified at the Stage 3 boot loader prompt. Variables can be added using `set`, removed with `unset`, and viewed with the `show` commands. Variables set in the `/boot/device.hints` file can be

overridden here also. Device hints entered at the boot loader are not permanent and will be forgotten on the next reboot.

Once the system is booted, the kenv(1) command can be used to dump all of the variables.

The syntax for the `/boot/device.hints` file is one variable per line, using the standard hash “#” as comment markers. Lines are constructed as follows:

```
hint.driver.unit.keyword="value"
```

The syntax for the Stage 3 boot loader is:

```
set hint.driver.unit.keyword=value
```

`driver` is the device driver name, `unit` is the device driver unit number, and `keyword` is the hint keyword. The keyword may consist of the following options:

- `at`: specifies the bus which the device is attached to.
- `port`: specifies the start address of the I/O to be used.
- `irq`: specifies the interrupt request number to be used.
- `drq`: specifies the DMA channel number.
- `maddr`: specifies the physical memory address occupied by the device.
- `flags`: sets various flag bits for the device.
- `disabled`: if set to `1` the device is disabled.

Device drivers may accept (or require) more hints not listed here, viewing their manual page is recommended. For more information, consult the device.hints(5), kenv(1), loader.conf(5), and loader(8) manual pages.

12.6 Init: Process Control Initialization

Once the kernel has finished booting, it passes control to the user process init(8), which is located at `/sbin/init`, or the program path specified in the `init_path` variable in `loader`.

12.6.1 Automatic Reboot Sequence

The automatic reboot sequence makes sure that the file systems available on the system are consistent. If they are not, and fsck(8) cannot fix the inconsistencies, init(8) drops the system into single-user mode for the system administrator to take care of the problems directly.

12.6.2 Single-User Mode

This mode can be reached through the automatic reboot sequence, or by the user booting with the `-s` option or setting the `boot_single` variable in `loader`.

It can also be reached by calling shutdown(8) without the reboot (`-r`) or halt (`-h`) options, from multi-user mode.

If the system `console` is set to `insecure` in `/etc/ttys`, then the system prompts for the `root` password before initiating single-user mode.

Example 12-3. An Insecure Console in `/etc/ttys`

```
# name  getty                           type    status          comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown off insecure
```

> **Note:** An `insecure` console means that you consider your physical security to the console to be insecure, and want to make sure only someone who knows the `root` password may use single-user mode, and it does not mean that you want to run your console insecurely. Thus, if you want security, choose `insecure`, not `secure`.

12.6.3 Multi-User Mode

If init(8) finds your file systems to be in order, or once the user has finished in single-user mode, the system enters multi-user mode, in which it starts the resource configuration of the system.

12.6.3.1 Resource Configuration (rc)

The resource configuration system reads in configuration defaults from `/etc/defaults/rc.conf`, and system-specific details from `/etc/rc.conf`, and then proceeds to mount the system file systems mentioned in `/etc/fstab`, start up networking services, start up miscellaneous system daemons, and finally runs the startup scripts of locally installed packages.

The rc(8) manual page is a good reference to the resource configuration system, as is examining the scripts themselves.

12.7 Shutdown Sequence

Upon controlled shutdown, via shutdown(8), init(8) will attempt to run the script `/etc/rc.shutdown`, and then proceed to send all processes the `TERM` signal, and subsequently the `KILL` signal to any that do not terminate timely.

To power down a FreeBSD machine on architectures and systems that support power management, simply use the command `shutdown -p now` to turn the power off immediately. To just reboot a FreeBSD system, just use `shutdown -r now`. You need to be `root` or a member of `operator` group to run shutdown(8). The halt(8) and reboot(8) commands can also be used, please refer to their manual pages and to shutdown(8)'s one for more information.

> **Note:** Power management requires acpi(4) support in the kernel or loaded as module for FreeBSD 5.X and apm(4) support for FreeBSD 4.X.

Chapter 13 使用者與基本帳號管理

Contributed by Neil Blakey-Milner.

13.1 概述

FreeBSD 允許多個使用者同時使用電腦。當然，這並不是很多人同時坐在同一台電腦前[1]，而是其他使用者可以透過網路來使用同一台電腦以完成他們的工作。要使用系統的話，那麼每個人都得有一個帳號。

讀完這章，您將了解：

- 在FreeBSD 系統上不同帳號之間的區別。
- 如何增加帳號。
- 如何刪除帳號。
- 如何更改帳號的基本資料，像是帳號全名，或是使用的shell 種類。
- 如何針對帳號、群組來設限，比如：允許存取記憶體或CPU 資源多寡等。
- 如何運用群組，來更容易地管理帳號。

在開始閱讀這章之前，您需要：

- 瞭解UNIX 及FreeBSD (Chapter 3)的基礎概念。

13.2 介紹

系統的所有存取是經由帳號來進行，而所有的程式process 是由使用者來進行，所以使用者及帳號的管理，乃是FreeBSD 系統上不可或缺的重點。

所有於FreeBSD 系統中的帳號皆包含下列相關資訊用來辨識身份。

使用者名稱

使用者名稱要輸入在`login:` 提示出現後。使用者名稱必須是獨一無二，不能有重複的使用者名稱。至於如何建立有效使用者名稱的規則，請參閱passwd(5) 說明，通常使用者名稱是以八個以內的小寫字母所組成。

密碼

每個帳號都可擁有一組密碼。密碼也可以不設，如此就不需密碼即可登入系統，但通常這並非妙策，每個帳號都應設定一組密碼。

使用者代號(User ID, UID)

UID 是系統用來辨識使用者的數字，通常範圍是從0 到65535 [2]。FreeBSD 內部是使用UID 來辨識使用者——FreeBSD 在執行任何指定使用者的指令之前，都會先把使用者名稱轉換為UID。也就是說，比如可以有數個不同的使用者名稱，但是都使用同一個UID，對FreeBSD 來說，這些帳號都只代表同一使用者。不過，實際上需要這樣做的可能性不大。

群組代號(Group ID, GID)

GID 是系統用來辨識使用者所屬群組的數字，通常範圍是從0 到65535[2] 。用群組來控制資源存取，可有效減少一些設定檔的大小。此外，使用者還可以同時屬於多個不同的群組。

登入分類(Login classes)

登入分類是群組的延伸機制，提供了不同的使用者更彈性的。

密碼變更期限

FreeBSD 預設並不要求使用者週期性的更改密碼。您可以强制某些或全部的使用者在指定的期間過後必須更改密碼。

帳號期限

FreeBSD 的帳號沒有預設的期限，如果您已知道帳號的使用期限，例如，學校中提供學生使用的帳號，可在建立帳號時指定帳號的期限。當帳號過期後會無法登入系統，但該帳號的目錄及檔案則會保留。

使用者全名

FreeBSD 的帳號使用使用者名稱用來辨識，但使用者名稱並不一定代表真實使用者的姓名。為帳號所需的相關資訊。

家目錄

家目錄為使用者登入系統時的所在目錄的完整路徑。通常會將所有使用者的家目錄放置於`/home/`*使用者名稱* 或`/usr/home/`*使用者名稱*。使用者可以將其個人的資料放置於其家目錄之中，並可以在此目錄底下建立新的目錄

使用者Shell

Shell 提供預設的環境讓使用者與系統互動。Shell 擁有數種不同的種類, 可供進階使用者依使用習慣選擇。

帳號主要分為下列三類: 系統管理者帳號, 系統帳號，及使用者帳號。系統管理者帳號的帳號通常為`root`，擁有最大的權限來管理系統。系統帳號用來執行伺服器服務。最後，使用者帳號供真正的使用者使用，可登入、讀信等等。

13.3 系統管理者帳號

The superuser account, usually called `root`, comes preconfigured to facilitate system administration, and should not be used for day-to-day tasks like sending and receiving mail, general exploration of the system, or programming.

This is because the superuser, unlike normal user accounts, can operate without limits, and misuse of the superuser account may result in spectacular disasters. User accounts are unable to destroy the system by mistake, so it is generally best to use normal user accounts whenever possible, unless you especially need the extra privilege.

You should always double and triple-check commands you issue as the superuser, since an extra space or missing character can mean irreparable data loss.

So, the first thing you should do after reading this chapter is to create an unprivileged user account for yourself for general usage if you have not already. This applies equally whether you are running a multi-user or single-user

machine. Later in this chapter, we discuss how to create additional accounts, and how to change between the normal user and superuser.

13.4 系統帳號

System users are those used to run services such as DNS, mail, web servers, and so forth. The reason for this is security; if all services ran as the superuser, they could act without restriction.

Examples of system users are `daemon`, `operator`, `bind` (for the Domain Name Service), `news`, and `www`.

`nobody` is the generic unprivileged system user. However, it is important to keep in mind that the more services that use `nobody`, the more files and processes that user will become associated with, and hence the more privileged that user becomes.

13.5 使用者帳號

User accounts are the primary means of access for real people to the system, and these accounts insulate the user and the environment, preventing the users from damaging the system or other users, and allowing users to customize their environment without affecting others.

Every person accessing your system should have a unique user account. This allows you to find out who is doing what, prevent people from clobbering each others' settings or reading each others' mail, and so forth.

Each user can set up their own environment to accommodate their use of the system, by using alternate shells, editors, key bindings, and language.

13.6 更改帳號

在UNIX 的環境之中提供了各式不同的指令管理使用者帳號，以下為較常使用的指令摘要及更詳細的使用範例。

指令	摘要
adduser(8)	新增使用者。
rmuser(8)	移除使用者。
chpass(1)	更改使用者資料。
passwd(1)	更改使用者密碼。
pw(8)	修改使用者的各種資料。

13.6.1 `adduser`

adduser(8) 是一支新增使用者的簡單程式。它會建立資料於系統的`passwd` 與`group` 檔案之中。同時也會建立使用者的家目錄，從 `/usr/share/skel` 複製預設的組態檔(“dotfiles”)，並可以選擇性的郵件通知新使用者歡迎訊息。

Example 13-1. 在FreeBSD 內新增使用者

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
#
```

Note: 您輸入的密碼並不會回應到螢幕，所以不會以星號顯示。請確定您所輸入的密碼無誤。

13.6.2 `rmuser`

您可以使用rmuser(8) 來將使用者從系統之中完全移除rmuser(8) 會執行以下動作:

1. 移除該使用者的crontab(1) 資料(如果存在)。
2. 移除所有屬於該使用者的at(1) 工作。
3. 中止所有該使用者擁有的程序。
4. 移除系統本機密碼檔中該使用者的資料。
5. 移除該使用者的家目錄(如果為該使用者所有)。
6. 移除`/var/mail` 中屬於該使用者的郵件。
7. 移除暫存空間(如: `/tmp`) 中所有屬於該使用者的檔案。

8. 最後，在`/etc/group` 檔內移除該使用者帳號。

> **Note:** 若該群組已無成員，或者是群組名稱與該使用者名稱相同時，則群組將會被移除；此操作會與adduser(8) 所建立的帳號群組相對應。

rmuser(8) 無法移除系統管理者帳號帳號，因為這即代表嚴重的破壞行為。

為了確認您的操作，預設採互動模式。

Example 13-2. `rmuser` 帳號移除

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user’s home directory (/home/jru)? y
Updating password file, updating databases, done.
Updating group file: trusted (removing group jru -- personal group is empty) done.
Removing user’s incoming mail file /var/mail/jru: done.
Removing files belonging to jru from /tmp: done.
Removing files belonging to jru from /var/tmp: done.
Removing files belonging to jru from /var/tmp/vi.recover: done.
#
```

13.6.3 chpass

chpass(1) 可更改使用者資料如: 密碼、Shell及個人資訊。

僅系統管理者即系統管理者帳號可利用chpass(1) 更改其他使用者的資訊及密碼

除了指定使用者名稱，當不加參數時，chpass(1) 會將使用者資訊顯示於編輯器當中。並於使用者離開編輯器時更新使用者資訊。

> **Note:** 若您並非系統管理者帳號，在離開編輯器前會詢問您的密碼。

Example 13-3. 系統管理者帳號chpass

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
```

```
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

一般使用者僅可更改自己的少部份資訊。

Example 13-4. 一般使用者chpass

```
#Changing user database information for jru.
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

> **Note:** chfn(1) 與chsh(1) 即為chpass(1)，也同ypchpass(1)、ypchfn(1) 與ypchsh(1)。NIS 支援是自動的，所以無需在指令前加上`yp`。若這會困擾您，請不必擔心，Chapter 27 將函蓋NIS 的部份的說明。

13.6.4 passwd

passwd(1) 是更改密碼常用的方式，除了超級管理者可更改其他使用者的密碼外使用者僅能更改自己的密碼。

> **Note:** 為了避免意外或未經同意的修改，在更新密碼前需輸入原密碼。

Example 13-5. 更改您的密碼

```
% passwd
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

Example 13-6. 以系統管理者帳號去更改其他使用者的密碼

```
# passwd jru
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

Note: chpass(1)、yppasswd(1) 即為passwd(1)，皆支援NIS。

13.6.5 pw

pw(8) 用來建立、移除、修改及查詢使用者及群組。其功能即為系統使用者及群組檔案的前端。pw(8) 擁有大量的指令參數較適合使用於shell script 中，對新手來說會此指令較其他指令複雜許多。

13.7 使用者資源限制

若您擁有許多使用者，接下會想到該如何限制使用的資源。FreeBSD 提供管理者許多方法來限制系統的資源給每個人使用。這些限制分為兩個部份: 磁碟限額，以及其他資源限制。

磁碟限額可以限制使用者的磁碟用量，它提供了一種方法可以快速的檢查並計算用量而不需每次重新計算，關於磁碟限額將於Section 18.15 會討論。

其他資源限制包含了CPU、記憶體、以及其他每個使用者可使用的資源做限制，這些限制可使用Login class來定義並於在本章討論。

Login class 定義於`/etc/login.conf`。明確語意不會在本節說明但詳細的描述會在login.conf(5) 文件中。每使用者預設被分配到一個Login class 中(預設為`default`)，而每個Login class 都有其資源的限制(Login capabilitiy)。Login capabilitiy 以*名稱*=*值* 成對，*名稱* 代表資源的種類，而*值* 為任意的字串，為對應名稱的參數。設定Login class 及Login capability 相當簡單，並同樣在login.conf(5) 中詳細說明。

Note: 系統不會直接讀取`/etc/login.conf` 的組態而是讀取提供查詢較快的`/etc/login.conf.db` 資料庫檔。要從`/etc/login.conf` 產生`/etc/login.conf.db` 需要執行以下指令:

```
# cap_mkdb /etc/login.conf
```

資源限制於一般的Login capability 有兩點不同。第一，每種限制分為軟性限制及硬性限制。軟性限制可由使用者或應用程式調整，但不能高於硬性限制。後者限制可被使用者降低，但無法再提高。第二，多數資源限制是針對每個使用者的個別行程限制，而不是使用者的所有行程。注意，這些差異是由指定的限制程式托管，並非實作於Login capability 的架構(例如，這些不是*真正* 登入容量的特例)。

另外，為了避免麻煩，以下為幾個常用的資源限制(剩下及其他的Login capability 可在login.conf(5) 中找到說明)。

`coredumpsize`

The limit on the size of a core file generated by a program is, for obvious reasons, subordinate to other limits on disk usage (e.g., `filesize`, or disk quotas). Nevertheless, it is often used as a less-severe method of controlling disk space consumption: since users do not generate core files themselves, and often do not delete them, setting this may save them from running out of disk space should a large program (e.g., **emacs**) crash.

`cputime`

This is the maximum amount of CPU time a user's process may consume. Offending processes will be killed by the kernel.

> **Note:** This is a limit on CPU *time* consumed, not percentage of the CPU as displayed in some fields by top(1) and ps(1). A limit on the latter is, at the time of this writing, not possible, and would be rather useless: a compiler——probably a legitimate task——can easily use almost 100% of a CPU for some time.

`filesize`

This is the maximum size of a file the user may possess. Unlike disk quotas, this limit is enforced on individual files, not the set of all files a user owns.

`maxproc`

This is the maximum number of processes a user may be running. This includes foreground and background processes alike. For obvious reasons, this may not be larger than the system limit specified by the `kern.maxproc` sysctl(8). Also note that setting this too small may hinder a user's productivity: it is often useful to be logged in multiple times or execute pipelines. Some tasks, such as compiling a large program, also spawn multiple processes (e.g., make(1), cc(1), and other intermediate preprocessors).

`memorylocked`

This is the maximum amount a memory a process may have requested to be locked into main memory (e.g., see mlock(2)). Some system-critical programs, such as amd(8), lock into main memory such that in the event of being swapped out, they do not contribute to a system's trashing in time of trouble.

`memoryuse`

This is the maximum amount of memory a process may consume at any given time. It includes both core memory and swap usage. This is not a catch-all limit for restricting memory consumption, but it is a good start.

`openfiles`

This is the maximum amount of files a process may have open. In FreeBSD, files are also used to represent sockets and IPC channels; thus, be careful not to set this too low. The system-wide limit for this is defined by the `kern.maxfiles` sysctl(8).

`sbsize`

This is the limit on the amount of network memory, and thus mbufs, a user may consume. This originated as a response to an old DoS attack by creating a lot of sockets, but can be generally used to limit network communications.

`stacksize`

This is the maximum size a process' stack may grow to. This alone is not sufficient to limit the amount of memory a program may use; consequently, it should be used in conjunction with other limits.

There are a few other things to remember when setting resource limits. Following are some general tips, suggestions, and miscellaneous comments.

- Processes started at system startup by `/etc/rc` are assigned to the `daemon` login class.
- Although the `/etc/login.conf` that comes with the system is a good source of reasonable values for most limits, only you, the administrator, can know what is appropriate for your system. Setting a limit too high may open your system up to abuse, while setting it too low may put a strain on productivity.
- Users of the X Window System (X11) should probably be granted more resources than other users. X11 by itself takes a lot of resources, but it also encourages users to run more programs simultaneously.
- Remember that many limits apply to individual processes, not the user as a whole. For example, setting `openfiles` to 50 means that each process the user runs may open up to 50 files. Thus, the gross amount of files a user may open is the value of `openfiles` multiplied by the value of `maxproc`. This also applies to memory consumption.

For further information on resource limits and login classes and capabilities in general, please consult the relevant manual pages: cap_mkdb(1), getrlimit(2), login.conf(5).

13.8 群組

A group is simply a list of users. Groups are identified by their group name and GID (Group ID). In FreeBSD (and most other UNIX like systems), the two factors the kernel uses to decide whether a process is allowed to do something is its user ID and list of groups it belongs to. Unlike a user ID, a process has a list of groups associated with it. You may hear some things refer to the "group ID" of a user or process; most of the time, this just means the first group in the list.

The group name to group ID map is in `/etc/group`. This is a plain text file with four colon-delimited fields. The first field is the group name, the second is the encrypted password, the third the group ID, and the fourth the comma-delimited list of members. It can safely be edited by hand (assuming, of course, that you do not make any syntax errors!). For a more complete description of the syntax, see the group(5) manual page.

If you do not want to edit `/etc/group` manually, you can use the pw(8) command to add and edit groups. For example, to add a group called `teamtwo` and then confirm that it exists you can use:

Example 13-7. Adding a Group Using pw(8)

```
# pw groupadd teamtwo
# pw groupshow teamtwo
teamtwo:*:1100:
```

The number `1100` above is the group ID of the group `teamtwo`. Right now, `teamtwo` has no members, and is thus rather useless. Let's change that by inviting `jru` to the `teamtwo` group.

Example 13-8. Adding Somebody to a Group Using pw(8)

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
teamtwo:*:1100:jru
```

The argument to the `-M` option is a comma-delimited list of users who are members of the group. From the preceding sections, we know that the password file also contains a group for each user. The latter (the user) is automatically added to the group list by the system; the user will not show up as a member when using the `groupshow` command to pw(8), but will show up when the information is queried via id(1) or similar tool. In other words, pw(8) only manipulates the `/etc/group` file; it will never attempt to read additionally data from `/etc/passwd`.

Example 13-9. Using id(1) to Determine Group Membership

```
% id jru
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

As you can see, `jru` is a member of the groups `jru` and `teamtwo`.

For more information about pw(8), see its manual page, and for more information on the format of `/etc/group`, consult the group(5) manual page.

Notes

1. Well..除非您連接multiple terminals，這種情況我們會在Chapter 24 講到。
2. UID/GID 最大可使用至4294967295，但這樣的ID 可能會對已假設範圍的軟體造成嚴重問題。

Chapter 14 系統安全

Much of this chapter has been taken from the security(7) manual page by Matthew Dillon.

14.1 概述

這一章將對系統安全的基本概念進行介紹，除此之外，還將介紹一些好的習慣，以及FreeBSD 下的一些更深入的話題。這章的許多內容對於一般的系統和Internet 安全也適用。如今，Internet 已經不再像以前那樣是個人人都願意與您作好鄰居的『友善場所』。必須讓系統更安全，才能去保護您的資料、智慧財產、寶貴時間以及其他很多東西，而不至於被入侵者或心存惡意的人所竊取。

FreeBSD 提供了一系列工具和相關機制，來確保系統和網路的完整、安全。

讀完這章，您將了解：

- FreeBSD 系統的基本安全概念。
- FreeBSD 中許多可用的加密機制，例如DES 及MD5。
- 如何建立一次性(one-time)密碼驗證機制。
- 如何設定TCP Wrappers 以便與`inetd` 配合使用。
- 如何在FreeBSD 5.0. 之前的版本上設定**KerberosIV**。
- 如何在FreeBSD 5.0 (含之後版本)上設定**Kerberos5**。
- 如何設定IPsec 以及在FreeBSD/Windows 上建立VPN 網路。
- 如何設定、運用**OpenSSH**，以及FreeBSD 的SSH 實作方式(implementation)
- 了解檔案系統的ACLs 機制為何，以及如何運用。
- 如何使用**Portaudit** 工具來檢驗(audit) 從Ports Collection 安裝的軟體安全性。
- 如何善用FreeBSD 安全公告(Security Advisories)，並採取相應措施。
- 瞭解Process Accounting 機制及如何在FreeBSD 上啓動。

在開始閱讀這章之前，您需要：

- 瞭解FreeBSD 及Internet 的基本概念。

本書中其他章節，也有介紹安全方面的其他話題。例如：在Chapter 16 有談到Mandatory Access Control，Internet Firewalls 則在Chapter 28。

14.2 介紹

安全，對系統管理者而言，是至始至終最基本的要求。由於所有的BSD UNIX multi-user 系統都提供了與生俱來的基本安全，所以建立、維護額外的安全機制，以確保使用者的『可靠』，可能也就是系統管理員最需要慎思的艱巨任務了。機器的安全性取決於您所建立的安全措施，而許多安全方面的考量，則會與人們使用電腦時的便利相矛盾。一般來說，UNIX 系統可同時執行許多數目的程式process ，並且其中許多process 也同時以Server 端來運作。——這意味著，外部實體機器能夠與它們互相連接，並產生互動。現

在的一般桌機，已經能夠達到以前小型主機甚至大型主機的性能，而隨著這些電腦的網路連接和在更大範圍內互相連接，安全也成為了一個日益嚴峻的課題。

安全最好的方式，是能夠透過像『洋蔥』那樣的層層防護模式。簡單講，應該儘可能的建立多層次安全防護，並小心地監視各類針對系統的入侵疑點。You do not want to overbuild your security or you will interfere with the detection side, and detection is one of the single most important aspects of any security mechanism. For example, it makes little sense to set the `schg` flag (see chflags(1)) on every system binary because while this may temporarily protect the binaries, it prevents an attacker who has broken in from making an easily detectable change that may result in your security mechanisms not detecting the attacker at all.

System security also pertains to dealing with various forms of attack, including attacks that attempt to crash, or otherwise make a system unusable, but do not attempt to compromise the `root` account (“break root”). Security concerns can be split up into several categories:

1. 服務阻斷攻擊(DoS)
2. 竊取其他使用者的帳號。
3. 透過各式Server 上所提供的Service 來竊取root 帳號。
4. 透過使用者帳號竊取root 帳號。
5. 開後門。

A denial of service attack is an action that deprives the machine of needed resources. Typically, DoS attacks are brute-force mechanisms that attempt to crash or otherwise make a machine unusable by overwhelming its servers or network stack. Some DoS attacks try to take advantage of bugs in the networking stack to crash a machine with a single packet. The latter can only be fixed by applying a bug fix to the kernel. Attacks on servers can often be fixed by properly specifying options to limit the load the servers incur on the system under adverse conditions. Brute-force network attacks are harder to deal with. A spoofed-packet attack, for example, is nearly impossible to stop, short of cutting your system off from the Internet. It may not be able to take your machine down, but it can saturate your Internet connection.

A user account compromise is even more common than a DoS attack. Many sysadmins still run standard **telnetd**, **rlogind**, **rshd**, and **ftpd** servers on their machines. These servers, by default, do not operate over encrypted connections. The result is that if you have any moderate-sized user base, one or more of your users logging into your system from a remote location (which is the most common and convenient way to login to a system) will have his or her password sniffed. The attentive system admin will analyze his remote access logs looking for suspicious source addresses even for successful logins.

One must always assume that once an attacker has access to a user account, the attacker can break `root`. However, the reality is that in a well secured and maintained system, access to a user account does not necessarily give the attacker access to `root`. The distinction is important because without access to `root` the attacker cannot generally hide his tracks and may, at best, be able to do nothing more than mess with the user’s files, or crash the machine. User account compromises are very common because users tend not to take the precautions that sysadmins take.

System administrators must keep in mind that there are potentially many ways to break `root` on a machine. The attacker may know the `root` password, the attacker may find a bug in a root-run server and be able to break `root` over a network connection to that server, or the attacker may know of a bug in a suid-root program that allows the attacker to break `root` once he has broken into a user’s account. If an attacker has found a way to break `root` on a machine, the attacker may not have a need to install a backdoor. Many of the `root` holes found and closed to date involve a considerable amount of work by the attacker to cleanup after himself, so most attackers install backdoors. A backdoor provides the attacker with a way to easily regain `root` access to the system, but it also gives the smart system administrator a convenient way to detect the intrusion. Making it impossible for an attacker to install a

backdoor may actually be detrimental to your security, because it will not close off the hole the attacker found to break in the first place.

Security remedies should always be implemented with a multi-layered “onion peel” approach and can be categorized as follows:

1. Securing `root` and staff accounts.
2. Securing `root`–run servers and suid/sgid binaries.
3. Securing user accounts.
4. Securing the password file.
5. Securing the kernel core, raw devices, and file systems.
6. Quick detection of inappropriate changes made to the system.
7. Paranoia.

The next section of this chapter will cover the above bullet items in greater depth.

14.3 FreeBSD 的系統安全

> **Command vs. Protocol:** Throughout this document, we will use **bold** text to refer to an application, and a `monospaced` font to refer to specific commands. Protocols will use a normal font. This typographical distinction is useful for instances such as ssh, since it is a protocol as well as command.

The sections that follow will cover the methods of securing your FreeBSD system that were mentioned in the last section of this chapter.

14.3.1 Securing the `root` Account and Staff Accounts

First off, do not bother securing staff accounts if you have not secured the `root` account. Most systems have a password assigned to the `root` account. The first thing you do is assume that the password is *always* compromised. This does not mean that you should remove the password. The password is almost always necessary for console access to the machine. What it does mean is that you should not make it possible to use the password outside of the console or possibly even with the su(1) command. For example, make sure that your ptys are specified as being insecure in the `/etc/ttys` file so that direct `root` logins via `telnet` or `rlogin` are disallowed. If using other login services such as **sshd**, make sure that direct `root` logins are disabled there as well. You can do this by editing your `/etc/ssh/sshd_config` file, and making sure that `PermitRootLogin` is set to `NO`. Consider every access method ——services such as FTP often fall through the cracks. Direct `root` logins should only be allowed via the system console.

Of course, as a sysadmin you have to be able to get to `root`, so we open up a few holes. But we make sure these holes require additional password verification to operate. One way to make `root` accessible is to add appropriate staff accounts to the `wheel` group (in `/etc/group`). The staff members placed in the `wheel` group are allowed to `su` to `root`. You should never give staff members native `wheel` access by putting them in the `wheel` group in their password entry. Staff accounts should be placed in a `staff` group, and then added to the `wheel` group via the `/etc/group` file. Only those staff members who actually need to have `root` access should be placed in the `wheel`

group. It is also possible, when using an authentication method such as Kerberos, to use Kerberos' `.k5login` file in the `root` account to allow a ksu(1) to `root` without having to place anyone at all in the `wheel` group. This may be the better solution since the `wheel` mechanism still allows an intruder to break `root` if the intruder has gotten hold of your password file and can break into a staff account. While having the `wheel` mechanism is better than having nothing at all, it is not necessarily the safest option.

An indirect way to secure staff accounts, and ultimately `root` access is to use an alternative login access method and do what is known as "starring" out the encrypted password for the staff accounts. Using the vipw(8) command, one can replace each instance of an encrypted password with a single "`*`" character. This command will update the `/etc/master.passwd` file and user/password database to disable password-authenticated logins.

A staff account entry such as:

```
foobar:R9DT/Fa1/LV9U:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Should be changed to this:

```
foobar:*:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

This change will prevent normal logins from occurring, since the encrypted password will never match "`*`". With this done, staff members must use another mechanism to authenticate themselves such as kerberos(1) or ssh(1) using a public/private key pair. When using something like Kerberos, one generally must secure the machines which run the Kerberos servers and your desktop workstation. When using a public/private key pair with ssh, one must generally secure the machine used to login *from* (typically one's workstation). An additional layer of protection can be added to the key pair by password protecting the key pair when creating it with ssh-keygen(1). Being able to "star" out the passwords for staff accounts also guarantees that staff members can only login through secure access methods that you have set up. This forces all staff members to use secure, encrypted connections for all of their sessions, which closes an important hole used by many intruders: sniffing the network from an unrelated, less secure machine.

The more indirect security mechanisms also assume that you are logging in from a more restrictive server to a less restrictive server. For example, if your main box is running all sorts of servers, your workstation should not be running any. In order for your workstation to be reasonably secure you should run as few servers as possible, up to and including no servers at all, and you should run a password-protected screen blanker. Of course, given physical access to a workstation an attacker can break any sort of security you put on it. This is definitely a problem that you should consider, but you should also consider the fact that the vast majority of break-ins occur remotely, over a network, from people who do not have physical access to your workstation or servers.

Using something like Kerberos also gives you the ability to disable or change the password for a staff account in one place, and have it immediately affect all the machines on which the staff member may have an account. If a staff member's account gets compromised, the ability to instantly change his password on all machines should not be underrated. With discrete passwords, changing a password on N machines can be a mess. You can also impose re-passwording restrictions with Kerberos: not only can a Kerberos ticket be made to timeout after a while, but the Kerberos system can require that the user choose a new password after a certain period of time (say, once a month).

14.3.2 Securing Root-run Servers and SUID/SGID Binaries

The prudent sysadmin only runs the servers he needs to, no more, no less. Be aware that third party servers are often the most bug-prone. For example, running an old version of **imapd** or **popper** is like giving a universal `root` ticket out to the entire world. Never run a server that you have not checked out carefully. Many servers do not need to be run as `root`. For example, the **ntalk**, **comsat**, and **finger** daemons can be run in special user *sandboxes*. A sandbox is not perfect, unless you go through a large amount of trouble, but the onion approach to security still stands: If

someone is able to break in through a server running in a sandbox, they still have to break out of the sandbox. The more layers the attacker must break through, the lower the likelihood of his success. Root holes have historically been found in virtually every server ever run as `root`, including basic system servers. If you are running a machine through which people only login via **sshd** and never login via **telnetd** or **rshd** or **rlogind**, then turn off those services!

FreeBSD now defaults to running **ntalkd**, **comsat**, and **finger** in a sandbox. Another program which may be a candidate for running in a sandbox is named(8). `/etc/defaults/rc.conf` includes the arguments necessary to run **named** in a sandbox in a commented-out form. Depending on whether you are installing a new system or upgrading an existing system, the special user accounts used by these sandboxes may not be installed. The prudent sysadmin would research and implement sandboxes for servers whenever possible.

There are a number of other servers that typically do not run in sandboxes: **sendmail**, **popper**, **imapd**, **ftpd**, and others. There are alternatives to some of these, but installing them may require more work than you are willing to perform (the convenience factor strikes again). You may have to run these servers as `root` and rely on other mechanisms to detect break-ins that might occur through them.

The other big potential `root` holes in a system are the suid-root and sgid binaries installed on the system. Most of these binaries, such as **rlogin**, reside in `/bin`, `/sbin`, `/usr/bin`, or `/usr/sbin`. While nothing is 100% safe, the system-default suid and sgid binaries can be considered reasonably safe. Still, `root` holes are occasionally found in these binaries. A `root` hole was found in `Xlib` in 1998 that made **xterm** (which is typically suid) vulnerable. It is better to be safe than sorry and the prudent sysadmin will restrict suid binaries, that only staff should run, to a special group that only staff can access, and get rid of (`chmod 000`) any suid binaries that nobody uses. A server with no display generally does not need an **xterm** binary. Sgid binaries can be almost as dangerous. If an intruder can break an sgid-kmem binary, the intruder might be able to read `/dev/kmem` and thus read the encrypted password file, potentially compromising any passworded account. Alternatively an intruder who breaks group `kmem` can monitor keystrokes sent through ptys, including ptys used by users who login through secure methods. An intruder that breaks the `tty` group can write to almost any user's tty. If a user is running a terminal program or emulator with a keyboard-simulation feature, the intruder can potentially generate a data stream that causes the user's terminal to echo a command, which is then run as that user.

14.3.3 Securing User Accounts

User accounts are usually the most difficult to secure. While you can impose Draconian access restrictions on your staff and "star" out their passwords, you may not be able to do so with any general user accounts you might have. If you do have sufficient control, then you may win out and be able to secure the user accounts properly. If not, you simply have to be more vigilant in your monitoring of those accounts. Use of ssh and Kerberos for user accounts is more problematic, due to the extra administration and technical support required, but still a very good solution compared to a crypted password file.

14.3.4 Securing the Password File

The only sure fire way is to `*` out as many passwords as you can and use ssh or Kerberos for access to those accounts. Even though the encrypted password file (`/etc/spwd.db`) can only be read by `root`, it may be possible for an intruder to obtain read access to that file even if the attacker cannot obtain root-write access.

Your security scripts should always check for and report changes to the password file (see the Checking file integrity section below).

14.3.5 Securing the Kernel Core, Raw Devices, and File systems

If an attacker breaks `root` he can do just about anything, but there are certain conveniences. For example, most modern kernels have a packet sniffing device driver built in. Under FreeBSD it is called the `bpf` device. An intruder will commonly attempt to run a packet sniffer on a compromised machine. You do not need to give the intruder the capability and most systems do not have the need for the `bpf` device compiled in.

But even if you turn off the `bpf` device, you still have `/dev/mem` and `/dev/kmem` to worry about. For that matter, the intruder can still write to raw disk devices. Also, there is another kernel feature called the module loader, kldload(8). An enterprising intruder can use a KLD module to install his own `bpf` device, or other sniffing device, on a running kernel. To avoid these problems you have to run the kernel at a higher secure level, at least securelevel 1. The securelevel can be set with a `sysctl` on the `kern.securelevel` variable. Once you have set the securelevel to 1, write access to raw devices will be denied and special `chflags` flags, such as `schg`, will be enforced. You must also ensure that the `schg` flag is set on critical startup binaries, directories, and script files ——everything that gets run up to the point where the securelevel is set. This might be overdoing it, and upgrading the system is much more difficult when you operate at a higher secure level. You may compromise and run the system at a higher secure level but not set the `schg` flag for every system file and directory under the sun. Another possibility is to simply mount `/` and `/usr` read-only. It should be noted that being too Draconian in what you attempt to protect may prevent the all-important detection of an intrusion.

14.3.6 Checking File Integrity: Binaries, Configuration Files, Etc.

When it comes right down to it, you can only protect your core system configuration and control files so much before the convenience factor rears its ugly head. For example, using `chflags` to set the `schg` bit on most of the files in `/` and `/usr` is probably counterproductive, because while it may protect the files, it also closes a detection window. The last layer of your security onion is perhaps the most important ——detection. The rest of your security is pretty much useless (or, worse, presents you with a false sense of safety) if you cannot detect potential incursions. Half the job of the onion is to slow down the attacker, rather than stop him, in order to give the detection side of the equation a chance to catch him in the act.

The best way to detect an incursion is to look for modified, missing, or unexpected files. The best way to look for modified files is from another (often centralized) limited-access system. Writing your security scripts on the extra-secure limited-access system makes them mostly invisible to potential attackers, and this is important. In order to take maximum advantage you generally have to give the limited-access box significant access to the other machines in the business, usually either by doing a read-only NFS export of the other machines to the limited-access box, or by setting up ssh key-pairs to allow the limited-access box to ssh to the other machines. Except for its network traffic, NFS is the least visible method ——allowing you to monitor the file systems on each client box virtually undetected. If your limited-access server is connected to the client boxes through a switch, the NFS method is often the better choice. If your limited-access server is connected to the client boxes through a hub, or through several layers of routing, the NFS method may be too insecure (network-wise) and using ssh may be the better choice even with the audit-trail tracks that ssh lays.

Once you give a limited-access box, at least read access to the client systems it is supposed to monitor, you must write scripts to do the actual monitoring. Given an NFS mount, you can write scripts out of simple system utilities such as find(1) and md5(1). It is best to physically md5 the client-box files at least once a day, and to test control files such as those found in `/etc` and `/usr/local/etc` even more often. When mismatches are found, relative to the base md5 information the limited-access machine knows is valid, it should scream at a sysadmin to go check it out. A good security script will also check for inappropriate suid binaries and for new or deleted files on system partitions such as `/` and `/usr`.

When using ssh rather than NFS, writing the security script is much more difficult. You essentially have to `scp` the scripts to the client box in order to run them, making them visible, and for safety you also need to `scp` the binaries (such as find) that those scripts use. The **ssh** client on the client box may already be compromised. All in all, using ssh may be necessary when running over insecure links, but it is also a lot harder to deal with.

A good security script will also check for changes to user and staff members access configuration files: `.rhosts`, `.shosts`, `.ssh/authorized_keys` and so forth... files that might fall outside the purview of the `MD5` check.

If you have a huge amount of user disk space, it may take too long to run through every file on those partitions. In this case, setting mount flags to disallow suid binaries and devices on those partitions is a good idea. The `nodev` and `nosuid` options (see mount(8)) are what you want to look into. You should probably scan them anyway, at least once a week, since the object of this layer is to detect a break-in whether or not the break-in is effective.

Process accounting (see accton(8)) is a relatively low-overhead feature of the operating system which might help as a post-break-in evaluation mechanism. It is especially useful in tracking down how an intruder has actually broken into a system, assuming the file is still intact after the break-in occurs.

Finally, security scripts should process the log files, and the logs themselves should be generated in as secure a manner as possible ——remote syslog can be very useful. An intruder tries to cover his tracks, and log files are critical to the sysadmin trying to track down the time and method of the initial break-in. One way to keep a permanent record of the log files is to run the system console to a serial port and collect the information on a continuing basis through a secure machine monitoring the consoles.

14.3.7 Paranoia

A little paranoia never hurts. As a rule, a sysadmin can add any number of security features, as long as they do not affect convenience, and can add security features that *do* affect convenience with some added thought. Even more importantly, a security administrator should mix it up a bit ——if you use recommendations such as those given by this document verbatim, you give away your methodologies to the prospective attacker who also has access to this document.

14.3.8 DoS(Denial of Service)服務阻斷攻擊

這一節將介紹服務阻斷攻擊。DoS 攻擊通常是以封包的方式進行攻擊，儘管幾乎沒有任何辦法來阻止大量的偽造封包耗盡網路資源，但通常可以透過一些方式來降低這類攻擊的損害，使它們無法擊垮伺服器。

1. Limiting server forks.
2. Limiting springboard attacks (ICMP response 攻擊，ping broadcast等等)
3. Kernel Route Cache.

A common DoS attack is against a forking server that attempts to cause the server to eat processes, file descriptors, and memory, until the machine dies. **inetd** (see inetd(8)) has several options to limit this sort of attack. It should be noted that while it is possible to prevent a machine from going down, it is not generally possible to prevent a service from being disrupted by the attack. Read the **inetd** manual page carefully and pay specific attention to the `-c`, `-C`, and `-R` options. Note that spoofed-IP attacks will circumvent the `-C` option to **inetd**, so typically a combination of options must be used. Some standalone servers have self-fork-limitation parameters.

Sendmail has its `-OMaxDaemonChildren` option, which tends to work much better than trying to use sendmail's load limiting options due to the load lag. You should specify a `MaxDaemonChildren` parameter, when you start

sendmail, high enough to handle your expected load, but not so high that the computer cannot handle that number of **sendmails** without falling on its face. It is also prudent to run sendmail in queued mode (`-ODeliveryMode=queued`) and to run the daemon (`sendmail -bd`) separate from the queue-runs (`sendmail -q15m`). If you still want real-time delivery you can run the queue at a much lower interval, such as `-q1m`, but be sure to specify a reasonable `MaxDaemonChildren` option for *that* sendmail to prevent cascade failures.

Syslogd can be attacked directly and it is strongly recommended that you use the `-s` option whenever possible, and the `-a` option otherwise.

You should also be fairly careful with connect-back services such as **TCP Wrapper**'s reverse-identd, which can be attacked directly. You generally do not want to use the reverse-ident feature of **TCP Wrapper** for this reason.

It is a very good idea to protect internal services from external access by firewalling them off at your border routers. The idea here is to prevent saturation attacks from outside your LAN, not so much to protect internal services from network-based `root` compromise. Always configure an exclusive firewall, i.e., "firewall everything *except* ports A, B, C, D, and M-Z". This way you can firewall off all of your low ports except for certain specific services such as **named** (if you are primary for a zone), **ntalkd**, **sendmail**, and other Internet-accessible services. If you try to configure the firewall the other way ——as an inclusive or permissive firewall, there is a good chance that you will forget to "close" a couple of services, or that you will add a new internal service and forget to update the firewall. You can still open up the high-numbered port range on the firewall, to allow permissive-like operation, without compromising your low ports. Also take note that FreeBSD allows you to control the range of port numbers used for dynamic binding, via the various `net.inet.ip.portrange` `sysctl`'s (`sysctl -a | fgrep portrange`), which can also ease the complexity of your firewall's configuration. For example, you might use a normal first/last range of 4000 to 5000, and a hiport range of 49152 to 65535, then block off everything under 4000 in your firewall (except for certain specific Internet-accessible ports, of course).

Another common DoS attack is called a springboard attack ——to attack a server in a manner that causes the server to generate responses which overloads the server, the local network, or some other machine. The most common attack of this nature is the *ICMP ping broadcast attack*. The attacker spoofs ping packets sent to your LAN's broadcast address with the source IP address set to the actual machine they wish to attack. If your border routers are not configured to stomp on ping's to broadcast addresses, your LAN winds up generating sufficient responses to the spoofed source address to saturate the victim, especially when the attacker uses the same trick on several dozen broadcast addresses over several dozen different networks at once. Broadcast attacks of over a hundred and twenty megabits have been measured. A second common springboard attack is against the ICMP error reporting system. By constructing packets that generate ICMP error responses, an attacker can saturate a server's incoming network and cause the server to saturate its outgoing network with ICMP responses. This type of attack can also crash the server by running it out of mbuf's, especially if the server cannot drain the ICMP responses it generates fast enough. FreeBSD 4.X kernels have a kernel compile option called `ICMP_BANDLIM` which limits the effectiveness of these sorts of attacks. Later kernels use the **sysctl** variable `net.inet.icmp.icmplim`. The last major class of springboard attacks is related to certain internal **inetd** services such as the udp echo service. An attacker simply spoofs a UDP packet with the source address being server A's echo port, and the destination address being server B's echo port, where server A and B are both on your LAN. The two servers then bounce this one packet back and forth between each other. The attacker can overload both servers and their LANs simply by injecting a few packets in this manner. Similar problems exist with the internal **chargen** port. A competent sysadmin will turn off all of these inetd-internal test services.

Spoofed packet attacks may also be used to overload the kernel route cache. Refer to the `net.inet.ip.rtexpire`, `rtminexpire`, and `rtmaxcache` `sysctl` parameters. A spoofed packet attack that uses a random source IP will cause the kernel to generate a temporary cached route in the route table, viewable with `netstat -rna | fgrep W3`. These routes typically timeout in 1600 seconds or so. If the kernel detects that the cached route table has gotten too big it will dynamically reduce the `rtexpire` but will never decrease it to less than `rtminexpire`. There are two

problems:

1. The kernel does not react quickly enough when a lightly loaded server is suddenly attacked.
2. The `rtminexpire` is not low enough for the kernel to survive a sustained attack.

If your servers are connected to the Internet via a T3 or better, it may be prudent to manually override both `rtexpire` and `rtminexpire` via sysctl(8). Never set either parameter to zero (unless you want to crash the machine). Setting both parameters to 2 seconds should be sufficient to protect the route table from attack.

14.3.9 Access Issues with Kerberos and SSH

There are a few issues with both Kerberos and ssh that need to be addressed if you intend to use them. Kerberos V is an excellent authentication protocol, but there are bugs in the kerberized **telnet** and **rlogin** applications that make them unsuitable for dealing with binary streams. Also, by default Kerberos does not encrypt a session unless you use the `-x` option. **ssh** encrypts everything by default.

ssh works quite well in every respect except that it forwards encryption keys by default. What this means is that if you have a secure workstation holding keys that give you access to the rest of the system, and you ssh to an insecure machine, your keys are usable. The actual keys themselves are not exposed, but ssh installs a forwarding port for the duration of your login, and if an attacker has broken `root` on the insecure machine he can utilize that port to use your keys to gain access to any other machine that your keys unlock.

We recommend that you use ssh in combination with Kerberos whenever possible for staff logins. **ssh** can be compiled with Kerberos support. This reduces your reliance on potentially exposed ssh keys while at the same time protecting passwords via Kerberos. ssh keys should only be used for automated tasks from secure machines (something that Kerberos is unsuited to do). We also recommend that you either turn off key-forwarding in the ssh configuration, or that you make use of the `from=IP/DOMAIN` option that ssh allows in its `authorized_keys` file to make the key only usable to entities logging in from specific machines.

14.4 DES, MD5, and Crypt

Parts rewritten and updated by Bill Swingle.

Every user on a UNIX system has a password associated with their account. It seems obvious that these passwords need to be known only to the user and the actual operating system. In order to keep these passwords secret, they are encrypted with what is known as a “one-way hash”, that is, they can only be easily encrypted but not decrypted. In other words, what we told you a moment ago was obvious is not even true: the operating system itself does not *really* know the password. It only knows the *encrypted* form of the password. The only way to get the “plain-text” password is by a brute force search of the space of possible passwords.

Unfortunately the only secure way to encrypt passwords when UNIX came into being was based on DES, the Data Encryption Standard. This was not such a problem for users resident in the US, but since the source code for DES could not be exported outside the US, FreeBSD had to find a way to both comply with US law and retain compatibility with all the other UNIX variants that still used DES.

The solution was to divide up the encryption libraries so that US users could install the DES libraries and use DES but international users still had an encryption method that could be exported abroad. This is how FreeBSD came to

use MD5 as its default encryption method. MD5 is believed to be more secure than DES, so installing DES is offered primarily for compatibility reasons.

14.4.1 Recognizing Your Crypt Mechanism

Before FreeBSD 4.4 `libcrypt.a` was a symbolic link pointing to the library which was used for encryption. FreeBSD 4.4 changed `libcrypt.a` to provide a configurable password authentication hash library. Currently the library supports DES, MD5 and Blowfish hash functions. By default FreeBSD uses MD5 to encrypt passwords.

It is pretty easy to identify which encryption method FreeBSD is set up to use. Examining the encrypted passwords in the `/etc/master.passwd` file is one way. Passwords encrypted with the MD5 hash are longer than those encrypted with the DES hash and also begin with the characters `$1$`. Passwords starting with `$2a$` are encrypted with the Blowfish hash function. DES password strings do not have any particular identifying characteristics, but they are shorter than MD5 passwords, and are coded in a 64-character alphabet which does not include the `$` character, so a relatively short string which does not begin with a dollar sign is very likely a DES password.

The password format used for new passwords is controlled by the `passwd_format` login capability in `/etc/login.conf`, which takes values of `des`, `md5` or `blf`. See the login.conf(5) manual page for more information about login capabilities.

14.5 One-time Passwords

S/Key is a one-time password scheme based on a one-way hash function. FreeBSD uses the MD4 hash for compatibility but other systems have used MD5 and DES-MAC. S/Key has been part of the FreeBSD base system since version 1.1.5 and is also used on a growing number of other operating systems. S/Key is a registered trademark of Bell Communications Research, Inc.

From version 5.0 of FreeBSD, S/Key has been replaced with the functionally equivalent OPIE (One-time Passwords In Everything). OPIE uses the MD5 hash by default.

There are three different sorts of passwords which we will discuss below. The first is your usual UNIX style or Kerberos password; we will call this a “UNIX password”. The second sort is the one-time password which is generated by the S/Key `key` program or the OPIE opiekey(1) program and accepted by the `keyinit` or opiepasswd(1) programs and the login prompt; we will call this a “one-time password”. The final sort of password is the secret password which you give to the `key`/`opiekey` programs (and sometimes the `keyinit`/`opiepasswd` programs) which it uses to generate one-time passwords; we will call it a “secret password” or just unqualified “password”.

The secret password does not have anything to do with your UNIX password; they can be the same but this is not recommended. S/Key and OPIE secret passwords are not limited to 8 characters like old UNIX passwords[1], they can be as long as you like. Passwords of six or seven word long phrases are fairly common. For the most part, the S/Key or OPIE system operates completely independently of the UNIX password system.

Besides the password, there are two other pieces of data that are important to S/Key and OPIE. One is what is known as the “seed” or “key”, consisting of two letters and five digits. The other is what is called the “iteration count”, a number between 1 and 100. S/Key creates the one-time password by concatenating the seed and the secret password, then applying the MD4/MD5 hash as many times as specified by the iteration count and turning the result into six short English words. These six English words are your one-time password. The authentication system (primarily PAM) keeps track of the last one-time password used, and the user is authenticated if the hash of the user-provided password is equal to the previous password. Because a one-way hash is used it is impossible to

generate future one-time passwords if a successfully used password is captured; the iteration count is decremented after each successful login to keep the user and the login program in sync. When the iteration count gets down to 1, S/Key and OPIE must be reinitialized.

There are three programs involved in each system which we will discuss below. The `key` and `opiekey` programs accept an iteration count, a seed, and a secret password, and generate a one-time password or a consecutive list of one-time passwords. The `keyinit` and `opiepasswd` programs are used to initialize S/Key and OPIE respectively, and to change passwords, iteration counts, or seeds; they take either a secret passphrase, or an iteration count, seed, and one-time password. The `keyinfo` and `opieinfo` programs examine the relevant credentials files (`/etc/skeykeys` or `/etc/opiekeys`) and print out the invoking user's current iteration count and seed.

There are four different sorts of operations we will cover. The first is using `keyinit` or `opiepasswd` over a secure connection to set up one-time-passwords for the first time, or to change your password or seed. The second operation is using `keyinit` or `opiepasswd` over an insecure connection, in conjunction with `key` or `opiekey` over a secure connection, to do the same. The third is using `key`/`opiekey` to log in over an insecure connection. The fourth is using `key` or `opiekey` to generate a number of keys which can be written down or printed out to carry with you when going to some location without secure connections to anywhere.

14.5.1 Secure Connection Initialization

To initialize S/Key for the first time, change your password, or change your seed while logged in over a secure connection (e.g. on the console of a machine or via **ssh**), use the `keyinit` command without any parameters while logged in as yourself:

```
% keyinit
Adding unfurl:
Reminder - Only use this method if you are directly connected.
If you are using telnet or rlogin exit with no password and use keyinit -s.
Enter secret password:
Again secret password:

ID unfurl s/key is 99 to17757
DEFY CLUB PRO NASH LACE SOFT
```

For OPIE, `opiepasswd` is used instead:

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:
ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

At the `Enter new secret pass phrase:` or `Enter secret password:` prompts, you should enter a password or phrase. Remember, this is not the password that you will use to login with, this is used to generate your one-time login keys. The "ID" line gives the parameters of your particular instance: your login name, the iteration count, and seed. When logging in the system will remember these parameters and present them back to you so you do not

have to remember them. The last line gives the particular one-time password which corresponds to those parameters and your secret password; if you were to re-login immediately, this one-time password is the one you would use.

14.5.2 Insecure Connection Initialization

To initialize or change your secret password over an insecure connection, you will need to already have a secure connection to some place where you can run `key` or `opiekey`; this might be in the form of a desk accessory on a Macintosh, or a shell prompt on a machine you trust. You will also need to make up an iteration count (100 is probably a good value), and you may make up your own seed or use a randomly-generated one. Over on the insecure connection (to the machine you are initializing), use the `keyinit -s` command:

```
% keyinit -s
Updating unfurl:
Old key: to17758
Reminder you need the 6 English words from the key command.
Enter sequence count from 1 to 9999: 100
Enter new key [default to17759]:
s/key 100 to 17759
s/key access password:
s/key access password:CURE MIKE BANE HIM RACY GORE
```

For OPIE, you need to use `opiepasswd`:

```
% opiepasswd

Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
        otp-md5 498 to4268 ext
        Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
        otp-md5 499 to4269
        Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

To accept the default seed (which the `keyinit` program confusingly calls a `key`), press **Return**. Then before entering an access password, move over to your secure connection or S/Key desk accessory, and give it the same parameters:

```
% key 100 to17759
Reminder - Do not use this program while logged in via telnet or rlogin.
Enter secret password: <secret password>
CURE MIKE BANE HIM RACY GORE
```

Or for OPIE:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don’t use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
```

```
GAME GAG WELT OUT DOWN CHAT
```

Now switch back over to the insecure connection, and copy the one-time password generated over to the relevant program.

14.5.3 Generating a Single One-time Password

Once you have initialized S/Key or OPIE, when you login you will be presented with a prompt like this:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
s/key 97 fw13894
Password:
```

Or for OPIE:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
otp-md5 498 gr4269 ext
Password:
```

As a side note, the S/Key and OPIE prompts have a useful feature (not shown here): if you press **Return** at the password prompt, the prompter will turn echo on, so you can see what you are typing. This can be extremely useful if you are attempting to type in a password by hand, such as from a printout.

At this point you need to generate your one-time password to answer this login prompt. This must be done on a trusted system that you can run `key` or `opiekey` on. (There are versions of these for DOS, Windows and Mac OS as well.) They need both the iteration count and the seed as command line options. You can cut-and-paste these right from the login prompt on the machine that you are logging in to.

On the trusted system:

```
% key 97 fw13894
Reminder - Do not use this program while logged in via telnet or rlogin.
Enter secret password:
WELD LIP ACTS ENDS ME HAAG
```

For OPIE:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
```

```
Reminder: Don’t use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Now that you have your one-time password you can continue logging in:

```
login: <username>
s/key 97 fw13894
Password: <return to enable echo>
s/key 97 fw13894
Password [echo on]: WELD LIP ACTS ENDS ME HAAG
Last login: Tue Mar 21 11:56:41 from 10.0.0.2 ...
```

14.5.4 Generating Multiple One-time Passwords

Sometimes you have to go places where you do not have access to a trusted machine or secure connection. In this case, it is possible to use the `key` and `opiekey` commands to generate a number of one-time passwords beforehand to be printed out and taken with you. For example:

```
% key -n 5 30 zz99999
Reminder - Do not use this program while logged in via telnet or rlogin.
Enter secret password: <secret password>
26: SODA RUDE LEA LIND BUDD SILT
27: JILT SPY DUTY GLOW COWL ROT
28: THEM OW COLA RUNT BONG SCOT
29: COT MASH BARR BRIM NAN FLAG
30: CAN KNEE CAST NAME FOLK BILK
```

Or for OPIE:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Don’t use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

The `-n 5` requests five keys in sequence, the `30` specifies what the last iteration number should be. Note that these are printed out in *reverse* order of eventual use. If you are really paranoid, you might want to write the results down by hand; otherwise you can cut-and-paste into `lpr`. Note that each line shows both the iteration count and the one-time password; you may still find it handy to scratch off passwords as you use them.

14.5.5 Restricting Use of UNIX® Passwords

S/Key can place restrictions on the use of UNIX passwords based on the host name, user name, terminal port, or IP address of a login session. These restrictions can be found in the configuration file `/etc/skey.access`. The

skey.access(5) manual page has more information on the complete format of the file and also details some security cautions to be aware of before depending on this file for security.

If there is no `/etc/skey.access` file (this is the default on FreeBSD 4.X systems), then all users will be allowed to use UNIX passwords. If the file exists, however, then all users will be required to use S/Key unless explicitly permitted to do otherwise by configuration statements in the `skey.access` file. In all cases, UNIX passwords are permitted on the console.

Here is a sample `skey.access` configuration file which illustrates the three most common sorts of configuration statements:

```
permit internet 192.168.0.0 255.255.0.0
permit user fnord
permit port ttyd0
```

The first line (`permit internet`) allows users whose IP source address (which is vulnerable to spoofing) matches the specified value and mask, to use UNIX passwords. This should not be considered a security mechanism, but rather, a means to remind authorized users that they are using an insecure network and need to use S/Key for authentication.

The second line (`permit user`) allows the specified username, in this case `fnord`, to use UNIX passwords at any time. Generally speaking, this should only be used for people who are either unable to use the `key` program, like those with dumb terminals, or those who are ineducable.

The third line (`permit port`) allows all users logging in on the specified terminal line to use UNIX passwords; this would be used for dial-ups.

OPIE can restrict the use of UNIX passwords based on the IP address of a login session just like S/Key does. The relevant file is `/etc/opieaccess`, which is present by default on FreeBSD 5.0 and newer systems. Please check opieaccess(5) for more information on this file and which security considerations you should be aware of when using it.

Here is a sample `opieaccess` file:

```
permit 192.168.0.0 255.255.0.0
```

This line allows users whose IP source address (which is vulnerable to spoofing) matches the specified value and mask, to use UNIX passwords at any time.

If no rules in `opieaccess` are matched, the default is to deny non-OPIE logins.

14.6 TCP Wrappers

Written by: Tom Rhodes.

每個熟inetd(8) 的人幾乎都會聽過TCP Wrappers 這個東西，但很少人能完全瞭解它在網路環境上的好用在哪。大多數的人都會裝防火牆來保護網路，雖然，防火牆用途非常廣泛，但並非萬能。例如：若打算回傳一段文字給連線來源者等之類的。而TCP 軟體卻可以做到這點，還有其他更多事情。在以下段落內，我們將繼續介紹TCP Wrappers 提供的功能，以及一些實際運用的例子。

TCP Wrappers 可以讓`inetd` 所管理的每個server daemon ，都會在TCP Wrappers 的掌握之下。透過TCP Wrappers 這種方式可以支援連線紀錄(logging) 、回傳一段文字給連線來源者、可以讓daemon 只接受內部連

線等等。雖然其中部份功能用防火牆也可以做到，但TCP Wrappers 不只是增加了一層保護，還提供了防火牆所辦不到的事情。

然而，由TCP Wrappers 所提供的這些額外安全功能，不應該視為優秀防火牆的替代方案。應該結合TCP Wrappers 及防火牆、其他加强安全措施來一併運用才對，這樣才可以為系統提供多層安全防護。

由於這些設定是主要針對`inetd` 所提供的，所以我們建議您先參閱inetd 設定 一節。

> **Note:** 雖然inetd(8) 所啓動的程式並非全部都是真正的『daemons』，但一般來講，我們都還是會稱呼為『daemons』，下面我們仍將使用這字眼來表達。

14.6.1 Initial Configuration

若要在FreeBSD 中使用TCP Wrappers 的話，只要確定`inetd` 在啓動時，有在`/etc/rc.conf` 加上`-Ww` 的參數即可，這個設定在系統預設就有了。當然還需要適當修改`/etc/hosts.allow` 設定檔，但syslogd(8) 仍會在系統log 檔內，紀錄相關資料下來。

> **Note:** FreeBSD 的TCP Wrappers 實作方式與其他作業系統上的TCP Wrappers 不太一樣，目前FreeBSD 已經廢棄不用`/etc/hosts.deny` ，而一律改用`/etc/hosts.allow`。

最簡單的設定方式是，每個對daemon 的連線都由`/etc/hosts.allow` 來決定是否允許或拒絕。The default configuration in FreeBSD is to allow a connection to every daemon started with `inetd`. Changing this will be discussed only after the basic configuration is covered.

Basic configuration usually takes the form of `daemon : address : action`. Where `daemon` is the daemon name which `inetd` started. The `address` can be a valid hostname, an IP address or an IPv6 address enclosed in brackets ([ ]). The action field can be either allow or deny to grant or deny access appropriately. Keep in mind that configuration works off a first rule match semantic, meaning that the configuration file is scanned in ascending order for a matching rule. When a match is found the rule is applied and the search process will halt.

Several other options exist but they will be explained in a later section. A simple configuration line may easily be constructed from that information alone. For example, to allow POP3 connections via the `mail/qpopper` daemon, the following lines should be appended to `hosts.allow`:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

加上上面這行之後，必須重新啓動`inetd`。重新啓動的方式可以用kill(1) 指令，或打『`/etc/rc.d/inetd restart`』來完成。

14.6.2 Advanced Configuration

TCP Wrappers has advanced options too; they will allow for more control over the way connections are handled. In some cases it may be a good idea to return a comment to certain hosts or daemon connections. In other cases, perhaps a log file should be recorded or an email sent to the administrator. Other situations may require the use of a service for local connections only. This is all possible through the use of configuration options known as `wildcards`, expansion characters and external command execution. The next two sections are written to cover these situations.

14.6.2.1 External Commands

Suppose that a situation occurs where a connection should be denied yet a reason should be sent to the individual who attempted to establish that connection. How could it be done? That action can be made possible by using the `twist` option. When a connection attempt is made, `twist` will be called to execute a shell command or script. An example already exists in the `hosts.allow` file:

```
# The rest of the daemons are protected.
ALL : ALL \
        : severity auth.info \
        : twist /bin/echo "You are not welcome to use %d from %h."
```

This example shows that the message, "You are not allowed to use `daemon` from `hostname`." will be returned for any daemon not previously configured in the access file. This is extremely useful for sending a reply back to the connection initiator right after the established connection is dropped. Note that any message returned *must* be wrapped in quote " characters; there are no exceptions to this rule.

> **Warning:** It may be possible to launch a denial of service attack on the server if an attacker, or group of attackers could flood these daemons with connection requests.

Another possibility is to use the `spawn` option in these cases. Like `twist`, the `spawn` implicitly denies the connection and may be used to run external shell commands or scripts. Unlike `twist`, `spawn` will not send a reply back to the individual who established the connection. For an example, consider the following configuration line:

```
# We do not allow connections from example.com:
ALL : .example.com \
 : spawn (/bin/echo %a from %h attempted to access %d >> \
   /var/log/connections.log) \
 : deny
```

This will deny all connection attempts from the `*.example.com` domain; simultaneously logging the hostname, IP address and the daemon which they attempted to access in the `/var/log/connections.log` file.

Aside from the already explained substitution characters above, e.g. %a, a few others exist. See the hosts_access(5) manual page for the complete list.

14.6.2.2 Wildcard Options

Thus far the `ALL` example has been used continuously throughout the examples. Other options exist which could extend the functionality a bit further. For instance, `ALL` may be used to match every instance of either a daemon, domain or an IP address. Another wildcard available is `PARANOID` which may be used to match any host which provides an IP address that may be forged. In other words, `paranoid` may be used to define an action to be taken whenever a connection is made from an IP address that differs from its hostname. The following example may shed some more light on this discussion:

```
# Block possibly spoofed requests to sendmail:
sendmail : PARANOID : deny
```

In that example all connection requests to `sendmail` which have an IP address that varies from its hostname will be denied.

Caution: Using the `PARANOID` may severely cripple servers if the client or server has a broken DNS setup. Administrator discretion is advised.

To learn more about wildcards and their associated functionality, see the hosts_access(5) manual page.

Before any of the specific configuration lines above will work, the first configuration line should be commented out in `hosts.allow`. This was noted at the beginning of this section.

14.7 KerberosIV

Contributed by Mark Murray. Based on a contribution by Mark Dapoz.

Kerberos is a network add-on system/protocol that allows users to authenticate themselves through the services of a secure server. Services such as remote login, remote copy, secure inter-system file copying and other high-risk tasks are made considerably safer and more controllable.

The following instructions can be used as a guide on how to set up Kerberos as distributed for FreeBSD. However, you should refer to the relevant manual pages for a complete description.

14.7.1 Installing KerberosIV

Kerberos is an optional component of FreeBSD. The easiest way to install this software is by selecting the `krb4` or `krb5` distribution in **sysinstall** during the initial installation of FreeBSD. This will install the "eBones" (KerberosIV) or "Heimdal" (Kerberos5) implementation of Kerberos. These implementations are included because they are developed outside the USA/Canada and were thus available to system owners outside those countries during the era of restrictive export controls on cryptographic code from the USA.

Alternatively, the MIT implementation of Kerberos is available from the Ports Collection as `security/krb5`.

14.7.2 Creating the Initial Database

This is done on the Kerberos server only. First make sure that you do not have any old Kerberos databases around. You should change to the directory `/etc/kerberosIV` and check that only the following files are present:

```
# cd /etc/kerberosIV
# ls
README  krb.conf        krb.realms
```

If any additional files (such as `principal.*` or `master_key`) exist, then use the `kdb_destroy` command to destroy the old Kerberos database, or if Kerberos is not running, simply delete the extra files.

You should now edit the `krb.conf` and `krb.realms` files to define your Kerberos realm. In this case the realm will be `EXAMPLE.COM` and the server is `grunt.example.com`. We edit or create the `krb.conf` file:

```
# cat krb.conf
EXAMPLE.COM
EXAMPLE.COM grunt.example.com admin server
CS.BERKELEY.EDU okeeffe.berkeley.edu
```

```
ATHENA.MIT.EDU kerberos.mit.edu
ATHENA.MIT.EDU kerberos-1.mit.edu
ATHENA.MIT.EDU kerberos-2.mit.edu
ATHENA.MIT.EDU kerberos-3.mit.edu
LCS.MIT.EDU kerberos.lcs.mit.edu
TELECOM.MIT.EDU bitsy.mit.edu
ARC.NASA.GOV trident.arc.nasa.gov
```

In this case, the other realms do not need to be there. They are here as an example of how a machine may be made aware of multiple realms. You may wish to not include them for simplicity.

The first line names the realm in which this system works. The other lines contain realm/host entries. The first item on a line is a realm, and the second is a host in that realm that is acting as a "key distribution center". The words `admin server` following a host's name means that host also provides an administrative database server. For further explanation of these terms, please consult the Kerberos manual pages.

Now we have to add `grunt.example.com` to the `EXAMPLE.COM` realm and also add an entry to put all hosts in the `.example.com` domain in the `EXAMPLE.COM` realm. The `krb.realms` file would be updated as follows:

```
# cat krb.realms
grunt.example.com EXAMPLE.COM
.example.com EXAMPLE.COM
.berkeley.edu CS.BERKELEY.EDU
.MIT.EDU ATHENA.MIT.EDU
.mit.edu ATHENA.MIT.EDU
```

Again, the other realms do not need to be there. They are here as an example of how a machine may be made aware of multiple realms. You may wish to remove them to simplify things.

The first line puts the *specific* system into the named realm. The rest of the lines show how to default systems of a particular subdomain to a named realm.

Now we are ready to create the database. This only needs to run on the Kerberos server (or Key Distribution Center). Issue the `kdb_init` command to do this:

```
# kdb_init
Realm name [default  ATHENA.MIT.EDU ]: EXAMPLE.COM
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.

Enter Kerberos master key:
```

Now we have to save the key so that servers on the local machine can pick it up. Use the `kstash` command to do this:

```
# kstash

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
```

This saves the encrypted master password in `/etc/kerberosIV/master_key`.

14.7.3 Making It All Run

Two principals need to be added to the database for *each* system that will be secured with Kerberos. Their names are `kpasswd` and `rcmd`. These two principals are made for each system, with the instance being the name of the individual system.

These daemons, **kpasswd** and **rcmd** allow other systems to change Kerberos passwords and run commands like rcp(1), rlogin(1) and rsh(1).

Now let us add these entries:

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: passwd
Instance: grunt

<Not found>, Create [y] ? y

Principal: passwd, Instance: grunt, kdc_key_ver: 1
New Password:                    <---- enter RANDOM here
Verifying password

New Password: <---- enter RANDOM here

Random password [y] ? y

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name: rcmd
Instance: grunt

<Not found>, Create [y] ?

Principal: rcmd, Instance: grunt, kdc_key_ver: 1
New Password:  <---- enter RANDOM here
Verifying password

New Password:           <---- enter RANDOM here

Random password [y] ?

Principal's new key version = 1
```

```
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name:          <---- null entry here will cause an exit
```

14.7.4 Creating the Server File

We now have to extract all the instances which define the services on each machine. For this we use the `ext_srvtab` command. This will create a file which must be copied or moved *by secure means* to each Kerberos client's `/etc/kerberosIV` directory. This file must be present on each server and client, and is crucial to the operation of Kerberos.

```
# ext_srvtab grunt
Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
Generating 'grunt-new-srvtab'....
```

Now, this command only generates a temporary file which must be renamed to `srvtab` so that all the servers can pick it up. Use the mv(1) command to move it into place on the original system:

```
# mv grunt-new-srvtab srvtab
```

If the file is for a client system, and the network is not deemed safe, then copy the *`client`*`-new-srvtab` to removable media and transport it by secure physical means. Be sure to rename it to `srvtab` in the client's `/etc/kerberosIV` directory, and make sure it is mode 600:

```
# mv grumble-new-srvtab srvtab
# chmod 600 srvtab
```

14.7.5 Populating the Database

We now have to add some user entries into the database. First let us create an entry for the user `jane`. Use the `kdb_edit` command to do this:

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: jane
```

```
Instance:

<Not found>, Create [y] ? y

Principal: jane, Instance: , kdc_key_ver: 1
New Password:                <---- enter a secure password here
Verifying password

New Password:                <---- re-enter the password here
Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name:      <---- null entry here will cause an exit
```

14.7.6 Testing It All Out

First we have to start the Kerberos daemons. Note that if you have correctly edited your `/etc/rc.conf` then this will happen automatically when you reboot. This is only necessary on the Kerberos server. Kerberos clients will automatically get what they need from the `/etc/kerberosIV` directory.

```
# kerberos &
Kerberos server starting
Sleep forever on error
Log file is /var/log/kerberos.log
Current Kerberos master key version is 1.

Master key entered. BEWARE!

Current Kerberos master key version is 1
Local realm: EXAMPLE.COM
# kadmind -n &
KADM Server KADM0.0A initializing
Please do not use 'kill -9' to kill this job, use a
regular kill instead

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
```

Now we can try using the `kinit` command to get a ticket for the ID `jane` that we created above:

```
% kinit jane
MIT Project Athena (grunt.example.com)
Kerberos Initialization for "jane"
Password:
```

Try listing the tokens using `klist` to see if we really have them:

```
% klist
Ticket file:    /tmp/tkt245
```

```
Principal:      jane@EXAMPLE.COM

  Issued           Expires          Principal
Apr 30 11:23:22  Apr 30 19:23:22  krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

Now try changing the password using passwd(1) to check if the **kpasswd** daemon can get authorization to the Kerberos database:

```
% passwd
realm EXAMPLE.COM
Old password for jane:
New Password for jane:
Verifying password
New Password for jane:
Password changed.
```

14.7.7 Adding su Privileges

Kerberos allows us to give *each* user who needs `root` privileges their own *separate* su(1) password. We could now add an ID which is authorized to su(1) to `root`. This is controlled by having an instance of `root` associated with a principal. Using `kdb_edit` we can create the entry `jane.root` in the Kerberos database:

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: jane
Instance: root

<Not found>, Create [y] ? y

Principal: jane, Instance: root, kdc_key_ver: 1
New Password:                    <---- enter a SECURE password here
Verifying password

New Password:        <---- re-enter the password here

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ? 12 <--- Keep this short!
Attributes [ 0 ] ?
Edit O.K.
Principal name:           <---- null entry here will cause an exit
```

Now try getting tokens for it to make sure it works:

```
# kinit jane.root
MIT Project Athena (grunt.example.com)
Kerberos Initialization for "jane.root"
Password:
```

Now we need to add the user to root's .klogin file:

```
# cat /root/.klogin
jane.root@EXAMPLE.COM
```

Now try doing the su(1):

```
% su
Password:
```

and take a look at what tokens we have:

```
# klist
Ticket file: /tmp/tkt_root_245
Principal:      jane.root@EXAMPLE.COM

  Issued           Expires          Principal
May  2 20:43:12  May  3 04:43:12  krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

14.7.8 Using Other Commands

In an earlier example, we created a principal called jane with an instance root. This was based on a user with the same name as the principal, and this is a Kerberos default; that a <principal>.<instance> of the form <username>.root will allow that <username> to su(1) to root if the necessary entries are in the .klogin file in root's home directory:

```
# cat /root/.klogin
jane.root@EXAMPLE.COM
```

Likewise, if a user has in their own home directory lines of the form:

```
% cat ~/.klogin
jane@EXAMPLE.COM
jack@EXAMPLE.COM
```

This allows anyone in the EXAMPLE.COM realm who has authenticated themselves as jane or jack (via kinit, see above) to access to jane's account or files on this system (grunt) via rlogin(1), rsh(1) or rcp(1).

For example, jane now logs into another system using Kerberos:

```
% kinit
MIT Project Athena (grunt.example.com)
Password:
% rlogin grunt
Last login: Mon May  1 21:14:47 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
```

```
        The Regents of the University of California.   All rights reserved.

FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

Or `jack` logs into `jane`'s account on the same machine (`jane` having set up the `.klogin` file as above, and the person in charge of Kerberos having set up principal *jack* with a null instance):

```
% kinit
% rlogin grunt -l jane
MIT Project Athena (grunt.example.com)
Password:
Last login: Mon May  1 21:16:55 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
        The Regents of the University of California.   All rights reserved.
FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

14.8 Kerberos5

Contributed by Tillman Hodgson. Based on a contribution by Mark Murray.

Every FreeBSD release beyond FreeBSD-5.1 includes support only for **Kerberos5**. Hence **Kerberos5** is the only version included, and its configuration is similar in many aspects to that of **KerberosIV**. The following information only applies to **Kerberos5** in post FreeBSD-5.0 releases. Users who wish to use the **KerberosIV** package may install the `security/krb4` port.

Kerberos is a network add-on system/protocol that allows users to authenticate themselves through the services of a secure server. Services such as remote login, remote copy, secure inter-system file copying and other high-risk tasks are made considerably safer and more controllable.

Kerberos can be described as an identity-verifying proxy system. It can also be described as a trusted third-party authentication system. **Kerberos** provides only one function ——the secure authentication of users on the network. It does not provide authorization functions (what users are allowed to do) or auditing functions (what those users did). After a client and server have used **Kerberos** to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business.

Therefore it is highly recommended that **Kerberos** be used with other security methods which provide authorization and audit services.

The following instructions can be used as a guide on how to set up **Kerberos** as distributed for FreeBSD. However, you should refer to the relevant manual pages for a complete description.

For purposes of demonstrating a **Kerberos** installation, the various name spaces will be handled as follows:

- The DNS domain (“zone”) will be example.org.
- The **Kerberos** realm will be EXAMPLE.ORG.

> **Note:** Please use real domain names when setting up **Kerberos** even if you intend to run it internally. This avoids DNS problems and assures inter-operation with other **Kerberos** realms.

14.8.1 History

Kerberos was created by MIT as a solution to network security problems. The **Kerberos** protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection.

Kerberos is both the name of a network authentication protocol and an adjective to describe programs that implement the program (**Kerberos** telnet, for example). The current version of the protocol is version 5, described in RFC 1510.

Several free implementations of this protocol are available, covering a wide range of operating systems. The Massachusetts Institute of Technology (MIT), where **Kerberos** was originally developed, continues to develop their **Kerberos** package. It is commonly used in the US as a cryptography product, as such it has historically been affected by US export regulations. The MIT **Kerberos** is available as a port (`security/krb5`). Heimdal **Kerberos** is another version 5 implementation, and was explicitly developed outside of the US to avoid export regulations (and is thus often included in non-commercial UNIX variants). The Heimdal **Kerberos** distribution is available as a port (`security/heimdal`), and a minimal installation of it is included in the base FreeBSD install.

In order to reach the widest audience, these instructions assume the use of the Heimdal distribution included in FreeBSD.

14.8.2 Setting up a Heimdal KDC

The Key Distribution Center (KDC) is the centralized authentication service that **Kerberos** provides ——it is the computer that issues **Kerberos** tickets. The KDC is considered “trusted” by all other computers in the **Kerberos** realm, and thus has heightened security concerns.

Note that while running the **Kerberos** server requires very few computing resources, a dedicated machine acting only as a KDC is recommended for security reasons.

To begin setting up a KDC, ensure that your `/etc/rc.conf` file contains the correct settings to act as a KDC (you may need to adjust paths to reflect your own system):

```
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"
kerberos_stash="YES"
```

> **Note:** The `kerberos_stash` is only available in FreeBSD 4.X.

Next we will set up your **Kerberos** config file, `/etc/krb5.conf`:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
[realms]
    EXAMPLE.ORG = {
        kdc = kerberos.example.org
        admin_server = kerberos.example.org
    }
[domain_realm]
    .example.org = EXAMPLE.ORG
```

Note that this `/etc/krb5.conf` file implies that your KDC will have the fully-qualified hostname of `kerberos.example.org`. You will need to add a CNAME (alias) entry to your zone file to accomplish this if your KDC has a different hostname.

> **Note:** For large networks with a properly configured BIND DNS server, the above example could be trimmed to:
>
> ```
> [libdefaults]
>       default_realm = EXAMPLE.ORG
> ```
>
> With the following lines being appended to the `example.org` zonefile:
>
> ```
> _kerberos._udp      IN  SRV     01 00 88 kerberos.example.org.
> _kerberos._tcp      IN  SRV     01 00 88 kerberos.example.org.
> _kpasswd._udp       IN  SRV     01 00 464 kerberos.example.org.
> _kerberos-adm._tcp  IN  SRV     01 00 749 kerberos.example.org.
> _kerberos           IN  TXT     EXAMPLE.ORG
> ```

> **Note:** For clients to be able to find the **Kerberos** services, you *must* have either a fully configured `/etc/krb5.conf` or a miminally configured `/etc/krb5.conf` *and* a properly configured DNS server.

Next we will create the **Kerberos** database. This database contains the keys of all principals encrypted with a master password. You are not required to remember this password, it will be stored in a file (`/var/heimdal/m-key`). To create the master key, run `kstash` and enter a password.

Once the master key has been created, you can initialize the database using the `kadmin` program with the `-l` option (standing for "local"). This option instructs `kadmin` to modify the database files directly rather than going through the `kadmind` network service. This handles the chicken-and-egg problem of trying to connect to the database before it is created. Once you have the `kadmin` prompt, use the `init` command to create your realms initial database.

Lastly, while still in `kadmin`, create your first principal using the `add` command. Stick to the defaults options for the principal for now, you can always change them later with the `modify` command. Note that you can use the `?` command at any prompt to see the available options.

A sample database creation session is shown below:

```
# kstash
Master key: xxxxxxxx
Verifying password - Master key: xxxxxxxx

# kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
Password: xxxxxxxx
Verifying password - Password: xxxxxxxx
```

Now it is time to start up the KDC services. Run `/etc/rc.d/kerberos start` and `/etc/rc.d/kadmind start` to bring up the services. Note that you will not have any kerberized daemons running at this point but you should be able to confirm the that the KDC is functioning by obtaining and listing a ticket for the principal (user) that you just created from the command-line of the KDC itself:

```
% k5init tillman
tillman@EXAMPLE.ORG's Password:

% k5list
Credentials cache: FILE:/tmp/krb5cc_500
 Principal: tillman@EXAMPLE.ORG

  Issued           Expires          Principal
Aug 27 15:37:58  Aug 28 01:37:58  krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
```

14.8.3 Kerberos enabling a server with Heimdal services

First, we need a copy of the **Kerberos** configuration file, `/etc/krb5.conf`. To do so, simply copy it over to the client computer from the KDC in a secure fashion (using network utilities, such as scp(1), or physically via a floppy disk).

Next you need a `/etc/krb5.keytab` file. This is the major difference between a server providing **Kerberos** enabled daemons and a workstation ——the server must have a `keytab` file. This file contains the servers host key, which allows it and the KDC to verify each others identity. It must be transmitted to the server in a secure fashion, as the security of the server can be broken if the key is made public. This explicitly means that transferring it via a clear text channel, such as FTP, is a very bad idea.

Typically, you transfer to the `keytab` to the server using the `kadmin` program. This is handy because you also need to create the host principal (the KDC end of the `krb5.keytab`) using `kadmin`.

Note that you must have already obtained a ticket and that this ticket must be allowed to use the `kadmin` interface in the `kadmind.acl`. See the section titled "Remote administration" in the Heimdal info pages (`info heimdal`) for details on designing access control lists. If you do not want to enable remote `kadmin` access, you can simply securely connect to the KDC (via local console, ssh(1) or **Kerberos** telnet(1)) and perform administration locally using `kadmin -l`.

After installing the `/etc/krb5.conf` file, you can use `kadmin` from the **Kerberos** server. The `add --random-key` command will let you add the servers host principal, and the `ext` command will allow you to extract the servers host principal to its own keytab. For example:

```
# kadmin
kadmin> add --random-key host/myserver.example.org
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
kadmin> ext host/myserver.example.org
kadmin> exit
```

Note that the `ext` command (short for "extract") stores the extracted key in `/etc/krb5.keytab` by default.

If you do not have `kadmind` running on the KDC (possibly for security reasons) and thus do not have access to `kadmin` remotely, you can add the host principal (`host/myserver.EXAMPLE.ORG`) directly on the KDC and then extract it to a temporary file (to avoid over-writing the `/etc/krb5.keytab` on the KDC) using something like this:

```
# kadmin
kadmin> ext --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

You can then securely copy the keytab to the server computer (using `scp` or a floppy, for example). Be sure to specify a non-default keytab name to avoid over-writing the keytab on the KDC.

At this point your server can communicate with the KDC (due to its `krb5.conf` file) and it can prove its own identity (due to the `krb5.keytab` file). It is now ready for you to enable some **Kerberos** services. For this example we will enable the `telnet` service by putting a line like this into your `/etc/inetd.conf` and then restarting the inetd(8) service with `/etc/rc.d/inetd restart`:

```
telnet    stream  tcp     nowait  root    /usr/libexec/telnetd  telnetd -a user
```

The critical bit is that the `-a` (for authentication) type is set to user. Consult the telnetd(8) manual page for more details.

14.8.4 Kerberos enabling a client with Heimdal

Setting up a client computer is almost trivially easy. As far as **Kerberos** configuration goes, you only need the **Kerberos** configuration file, located at `/etc/krb5.conf`. Simply securely copy it over to the client computer from the KDC.

Test your client computer by attempting to use `kinit`, `klist`, and `kdestroy` from the client to obtain, show, and then delete a ticket for the principal you created above. You should also be able to use **Kerberos** applications to connect to **Kerberos** enabled servers, though if that does not work and obtaining a ticket does the problem is likely with the server and not with the client or the KDC.

When testing an application like `telnet`, try using a packet sniffer (such as tcpdump(1)) to confirm that your password is not sent in the clear. Try using `telnet` with the `-x` option, which encrypts the entire data stream (similar to `ssh`).

The core **Kerberos** client applications (traditionally named `kinit`, `klist`, `kdestroy`, and `kpasswd`) are installed in the base FreeBSD install. Note that FreeBSD versions prior to 5.0 renamed them to `k5init`, `k5list`, `k5destroy`, `k5passwd`, and `k5stash` (though it is typically only used once).

Various non-core **Kerberos** client applications are also installed by default. This is where the “minimal” nature of the base Heimdal installation is felt: `telnet` is the only **Kerberos** enabled service.

The Heimdal port adds some of the missing client applications: **Kerberos** enabled versions of `ftp`, `rsh`, `rcp`, `rlogin`, and a few other less common programs. The MIT port also contains a full suite of **Kerberos** client applications.

14.8.5 User configuration files: `.k5login` and `.k5users`

Users within a realm typically have their **Kerberos** principal (such as `tillman@EXAMPLE.ORG`) mapped to a local user account (such as a local account named `tillman`). Client applications such as `telnet` usually do not require a user name or a principal.

Occasionally, however, you want to grant access to a local user account to someone who does not have a matching **Kerberos** principal. For example, `tillman@EXAMPLE.ORG` may need access to the local user account `webdevelopers`. Other principals may also need access to that local account.

The `.k5login` and `.k5users` files, placed in a users home directory, can be used similar to a powerful combination of `.hosts` and `.rhosts`, solving this problem. For example, if a `.k5login` with the following contents:

```
tillman@example.org
jdoe@example.org
```

Were to be placed into the home directory of the local user `webdevelopers` then both principals listed would have access to that account without requiring a shared password.

Reading the manual pages for these commands is recommended. Note that the `ksu` manual page covers `.k5users`.

14.8.6 Kerberos Tips, Tricks, and Troubleshooting

- When using either the Heimdal or MIT **Kerberos** ports ensure that your `PATH` environment variable lists the **Kerberos** versions of the client applications before the system versions.
- Do all the computers in your realm have synchronized time settings? If not, authentication may fail. Section 27.11 describes how to synchronize clocks using NTP.
- MIT and Heimdal inter-operate nicely. Except for `kadmin`, the protocol for which is not standardized.
- If you change your hostname, you also need to change your `host/` principal and update your keytab. This also applies to special keytab entries like the `www/` principal used for Apache's `www/mod_auth_kerb`.
- All hosts in your realm must be resolvable (both forwards and reverse) in DNS (or `/etc/hosts` as a minimum). CNAMEs will work, but the A and PTR records must be correct and in place. The error message is not very intuitive: "`Kerberos5 refuses authentication because Read req failed: Key table entry not found`".
- Some operating systems that may being acting as clients to your KDC do not set the permissions for `ksu` to be setuid `root`. This means that `ksu` does not work, which is a good security idea but annoying. This is not a KDC error.
- With MIT **Kerberos**, if you want to allow a principal to have a ticket life longer than the default ten hours, you must use `modify_principal` in `kadmin` to change the maxlife of both the principal in question and the `krbtgt` principal. Then the principal can use the `-l` option with `kinit` to request a ticket with a longer lifetime.
-

> **Note:** If you run a packet sniffer on your KDC to add in troubleshooting and then run `kinit` from a workstation, you will notice that your TGT is sent immediately upon running `kinit` ——even before you type your password! The explanation is that the **Kerberos** server freely transmits a TGT (Ticket Granting Ticket) to any unauthorized request; however, every TGT is encrypted in a key derived from the user's password. Therefore, when a user types their password it is not being sent to the KDC, it is being used to decrypt the TGT that `kinit` already obtained. If the decryption process results in a valid ticket with a valid time stamp, the user has valid **Kerberos** credentials. These credentials include a session key for establishing secure communications with the **Kerberos** server in the future, as well as the actual ticket-granting ticket, which is actually encrypted with the **Kerberos** server's own key. This second layer of encryption is unknown to the user, but it is what allows the **Kerberos** server to verify the authenticity of each TGT.

- If you want to use long ticket lifetimes (a week, for example) and you are using **OpenSSH** to connect to the machine where your ticket is stored, make sure that **Kerberos** `TicketCleanup` is set to `no` in your `sshd_config` or else your tickets will be deleted when you log out.
- Remember that host principals can have a longer ticket lifetime as well. If your user principal has a lifetime of a week but the host you are connecting to has a lifetime of nine hours, you will have an expired host principal in your cache and the ticket cache will not work as expected.
- When setting up a `krb5.dict` file to prevent specific bad passwords from being used (the manual page for `kadmind` covers this briefly), remember that it only applies to principals that have a password policy assigned to them. The `krb5.dict` files format is simple: one string per line. Creating a symbolic link to `/usr/share/dict/words` might be useful.

14.8.7 Differences with the MIT port

The major difference between the MIT and Heimdal installs relates to the `kadmin` program which has a different (but equivalent) set of commands and uses a different protocol. This has a large implications if your KDC is MIT as you will not be able to use the Heimdal `kadmin` program to administer your KDC remotely (or vice versa, for that matter).

The client applications may also take slightly different command line options to accomplish the same tasks. Following the instructions on the MIT **Kerberos** web site (http://web.mit.edu/Kerberos/www/) is recommended. Be careful of path issues: the MIT port installs into `/usr/local/` by default, and the "normal" system applications may be run instead of MIT if your `PATH` environment variable lists the system directories first.

> **Note:** With the MIT `security/krb5` port that is provided by FreeBSD, be sure to read the `/usr/local/share/doc/krb5/README.FreeBSD` file installed by the port if you want to understand why logins via `telnetd` and `klogind` behave somewhat oddly. Most importantly, correcting the "incorrect permissions on cache file" behavior requires that the `login.krb5` binary be used for authentication so that it can properly change ownership for the forwarded credentials.

14.8.8 Mitigating limitations found in Kerberos

14.8.8.1 Kerberos is an all-or-nothing approach

Every service enabled on the network must be modified to work with **Kerberos** (or be otherwise secured against network attacks) or else the users credentials could be stolen and re-used. An example of this would be **Kerberos** enabling all remote shells (via `rsh` and `telnet`, for example) but not converting the POP3 mail server which sends passwords in plain text.

14.8.8.2 Kerberos is intended for single-user workstations

In a multi-user environment, **Kerberos** is less secure. This is because it stores the tickets in the `/tmp` directory, which is readable by all users. If a user is sharing a computer with several other people simultaneously (i.e. multi-user), it is possible that the user's tickets can be stolen (copied) by another user.

This can be overcome with the `-c` filename command-line option or (preferably) the `KRB5CCNAME` environment variable, but this is rarely done. In principal, storing the ticket in the users home directory and using simple file permissions can mitigate this problem.

14.8.8.3 The KDC is a single point of failure

By design, the KDC must be as secure as the master password database is contained on it. The KDC should have absolutely no other services running on it and should be physically secured. The danger is high because **Kerberos** stores all passwords encrypted with the same key (the "master" key), which in turn is stored as a file on the KDC.

As a side note, a compromised master key is not quite as bad as one might normally fear. The master key is only used to encrypt the **Kerberos** database and as a seed for the random number generator. As long as access to your KDC is secure, an attacker cannot do much with the master key.

Additionally, if the KDC is unavailable (perhaps due to a denial of service attack or network problems) the network services are unusable as authentication can not be performed, a recipe for a denial-of-service attack. This can alleviated with multiple KDCs (a single master and one or more slaves) and with careful implementation of secondary or fall-back authentication (PAM is excellent for this).

14.8.8.4 Kerberos Shortcomings

Kerberos allows users, hosts and services to authenticate between themselves. It does not have a mechanism to authenticate the KDC to the users, hosts or services. This means that a trojanned `kinit` (for example) could record all user names and passwords. Something like `security/tripwire` or other file system integrity checking tools can alleviate this.

14.8.9 Resources and further information

- The **Kerberos** FAQ (http://www.faqs.org/faqs/Kerberos-faq/general/preamble.html)
- Designing an Authentication System: a Dialog in Four Scenes (http://web.mit.edu/Kerberos/www/dialogue.html)
- RFC 1510, The **Kerberos** Network Authentication Service (V5) (http://www.ietf.org/rfc/rfc1510.txt?number=1510)
- MIT **Kerberos** home page (http://web.mit.edu/Kerberos/www/)
- Heimdal **Kerberos** home page (http://www.pdc.kth.se/heimdal/)

14.9 OpenSSL

Written by: Tom Rhodes.

One feature that many users overlook is the **OpenSSL** toolkit included in FreeBSD. **OpenSSL** provides an encryption transport layer on top of the normal communications layer; thus allowing it to be intertwined with many network applications and services.

Some uses of **OpenSSL** may include encrypted authentication of mail clients, web based transactions such as credit card payments and more. Many ports such as `www/apache13-ssl`, and `mail/sylpheed-claws` will offer compilation support for building with **OpenSSL**.

> **Note:** In most cases the Ports Collection will attempt to build the `security/openssl` port unless the `WITH_OPENSSL_BASE` make variable is explicitly set to "yes".

The version of **OpenSSL** included in FreeBSD supports Secure Sockets Layer v2/v3 (SSLv2/SSLv3), Transport Layer Security v1 (TLSv1) network security protocols and can be used as a general cryptographic library.

> **Note:** While **OpenSSL** supports the IDEA algorithm, it is disabled by default due to United States patents. To use it, the license should be reviewed and, if the restrictions are acceptable, the `MAKE_IDEA` variable must be set in `make.conf`.

One of the most common uses of **OpenSSL** is to provide certificates for use with software applications. These certificates ensure that the credentials of the company or individual are valid and not fraudulent. If the certificate in question has not been verified by one of the several "Certificate Authorities", or CAs, a warning is usually produced. A Certificate Authority is a company, such as VeriSign (http://www.verisign.com), which will sign certificates in order to validate credentials of individuals or companies. This process has a cost associated with it and is definitely not a requirement for using certificates; however, it can put some of the more paranoid users at ease.

14.9.1 Generating Certificates

To generate a certificate, the following command is available:

```
# openssl req -new -nodes -out req.pem -keyout cert.pem
Generating a 1024 bit RSA private key
..............++++++
.....................................++++++
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
```

```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (eg, YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SOME PASSWORD
An optional company name []:Another Name
```

Notice the response directly after the "Common Name" prompt shows a domain name. This prompt requires a server name to be entered for verification purposes; placing anything but a domain name would yield a useless certificate. Other options, for instance expire time, alternate encryption algorithms, etc. are available. A complete list may be obtained by viewing the openssl(1) manual page.

Two files should now exist in the directory in which the aforementioned command was issued. The certificate request, `req.pem`, may be sent to a certificate authority who will validate the credentials that you entered, sign the request and return the certificate to you. The second file created will be named `cert.pem` and is the private key for the certificate and should be protected at all costs; if this falls in the hands of others it can be used to impersonate you (or your server).

In cases where a signature from a CA is not required, a self signed certificate can be created. First, generate the RSA key:

```
# openssl dsaparam -rand -genkey -out myRSA.key 1024
```

Next, generate the CA key:

```
# openssl gendsa -des3 -out myca.key myRSA.key
```

Use this key to create the certificate:

```
# openssl req -new -x509 -days 365 -key myca.key -out new.crt
```

Two new files should appear in the directory: a certificate authority signature file, `myca.key` and the certificate itself, `new.crt`. These should be placed in a directory, preferably under `/etc`, which is readable only by `root`. Permissions of 0700 should be fine for this and they can be set with the `chmod` utility.

14.9.2 Using Certificates, an Example

So what can these files do? A good use would be to encrypt connections to the **Sendmail** MTA. This would dissolve the use of clear text authentication for users who send mail via the local MTA.

> **Note:** This is not the best use in the world as some MUAs will present the user with an error if they have not installed the certificate locally. Refer to the documentation included with the software for more information on certificate installation.

The following lines should be placed inside the local `.mc` file:

```
dnl SSL Options
define('confCACERT_PATH','/etc/certs')dnl
```

```
define('confCACERT','/etc/certs/new.crt')dnl
define('confSERVER_CERT','/etc/certs/new.crt')dnl
define('confSERVER_KEY','/etc/certs/myca.key')dnl
define('confTLS_SRV_OPTIONS', 'V')dnl
```

Where `/etc/certs/` is the directory to be used for storing the certificate and key files locally. The last few requirements are a rebuild of the local `.cf` file. This is easily achieved by typing `make` *`install`* within the `/etc/mail` directory. Follow that up with `make` *`restart`* which should start the **Sendmail** daemon.

If all went well there will be no error messages in the `/var/log/maillog` file and **Sendmail** will show up in the process list.

For a simple test, simply connect to the mail server using the telnet(1) utility:

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is '^]'.
220 example.com ESMTP Sendmail 8.12.10/8.12.10; Tue, 31 Aug 2004 03:41:22 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

If the "STARTTLS" line appears in the output then everything is working correctly.

14.10 VPN over IPsec

Written by Nik Clayton.

Creating a VPN between two networks, separated by the Internet, using FreeBSD gateways.

14.10.1 Understanding IPsec

Written by Hiten M. Pandya.

This section will guide you through the process of setting up IPsec, and to use it in an environment which consists of FreeBSD and **Microsoft Windows 2000/XP** machines, to make them communicate securely. In order to set up IPsec, it is necessary that you are familiar with the concepts of building a custom kernel (see Chapter 8).

IPsec is a protocol which sits on top of the Internet Protocol (IP) layer. It allows two or more hosts to communicate in a secure manner (hence the name). The FreeBSD IPsec "network stack" is based on the KAME (http://www.kame.net/) implementation, which has support for both protocol families, IPv4 and IPv6.

> **Note:** FreeBSD 5.X contains a "hardware accelerated" IPsec stack, known as "Fast IPsec", that was obtained from OpenBSD. It employs cryptographic hardware (whenever possible) via the crypto(4) subsystem to optimize the performance of IPsec. This subsystem is new, and does not support all the features that are available in the KAME version of IPsec. However, in order to enable hardware-accelerated IPsec, the following kernel option has to be added to your kernel configuration file:
>
> ```
> options   FAST_IPSEC  # new IPsec (cannot define w/ IPSEC)
> ```
>
> Note, that it is not currently possible to use the "Fast IPsec" subsystem in lue with the KAME implementation of IPsec. Consult the fast_ipsec(4) manual page for more information.

IPsec consists of two sub-protocols:

- *Encapsulated Security Payload (ESP)*, protects the IP packet data from third party interference, by encrypting the contents using symmetric cryptography algorithms (like Blowfish, 3DES).
- *Authentication Header (AH)*, protects the IP packet header from third party interference and spoofing, by computing a cryptographic checksum and hashing the IP packet header fields with a secure hashing function. This is then followed by an additional header that contains the hash, to allow the information in the packet to be authenticated.

ESP and AH can either be used together or separately, depending on the environment.

IPsec can either be used to directly encrypt the traffic between two hosts (known as *Transport Mode*); or to build "virtual tunnels" between two subnets, which could be used for secure communication between two corporate networks (known as *Tunnel Mode*). The latter is more commonly known as a *Virtual Private Network (VPN)*. The ipsec(4) manual page should be consulted for detailed information on the IPsec subsystem in FreeBSD.

To add IPsec support to your kernel, add the following options to your kernel configuration file:

```
options   IPSEC        #IP security
options   IPSEC_ESP    #IP security (crypto; define w/ IPSEC)
```

If IPsec debugging support is desired, the following kernel option should also be added:

```
options   IPSEC_DEBUG  #debug for IP security
```

14.10.2 The Problem

There is no standard for what constitutes a VPN. VPNs can be implemented using a number of different technologies, each of which have their own strengths and weaknesses. This section presents a scenario, and the strategies used for implementing a VPN for this scenario.

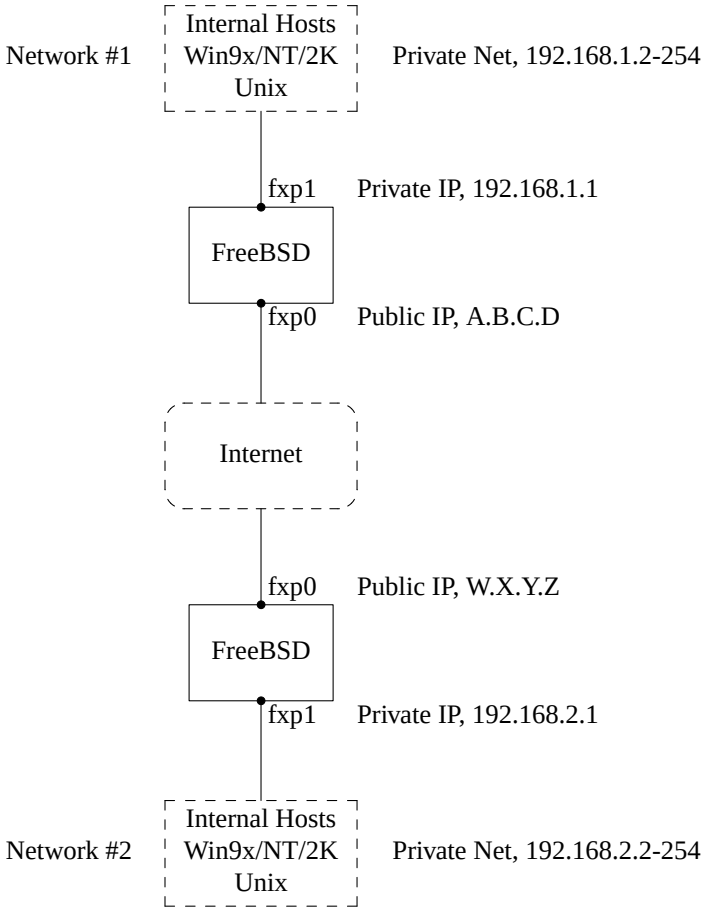
14.10.3 The Scenario: Two networks, connected to the Internet, to behave as one

The premise is as follows:

- You have at least two sites
- Both sites are using IP internally
- Both sites are connected to the Internet, through a gateway that is running FreeBSD.
- The gateway on each network has at least one public IP address.
- The internal addresses of the two networks can be public or private IP addresses, it does not matter. You can be running NAT on the gateway machine if necessary.
- The internal IP addresses of the two networks *do not collide*. While I expect it is theoretically possible to use a combination of VPN technology and NAT to get this to work, I expect it to be a configuration nightmare.

If you find that you are trying to connect two networks, both of which, internally, use the same private IP address range (e.g. both of them use `192.168.1.x`), then one of the networks will have to be renumbered.

The network topology might look something like this:



Notice the two public IP addresses. I will use the letters to refer to them in the rest of this article. Anywhere you see those letters in this article, replace them with your own public IP addresses. Note also that internally, the two gateway machines have .1 IP addresses, and that the two networks have different private IP addresses (`192.168.1.x`

and `192.168.2.x` respectively). All the machines on the private networks have been configured to use the `.1` machine as their default gateway.

The intention is that, from a network point of view, each network should view the machines on the other network as though they were directly attached the same router -- albeit a slightly slow router with an occasional tendency to drop packets.

This means that (for example), machine `192.168.1.20` should be able to run

```
ping 192.168.2.34
```

and have it work, transparently. Windows machines should be able to see the machines on the other network, browse file shares, and so on, in exactly the same way that they can browse machines on the local network.

And the whole thing has to be secure. This means that traffic between the two networks has to be encrypted.

Creating a VPN between these two networks is a multi-step process. The stages are as follows:

1. Create a "virtual" network link between the two networks, across the Internet. Test it, using tools like ping(8), to make sure it works.
2. Apply security policies to ensure that traffic between the two networks is transparently encrypted and decrypted as necessary. Test this, using tools like tcpdump(1), to ensure that traffic is encrypted.
3. Configure additional software on the FreeBSD gateways, to allow Windows machines to see one another across the VPN.

14.10.3.1 Step 1: Creating and testing a "virtual" network link

Suppose that you were logged in to the gateway machine on network #1 (with public IP address `A.B.C.D`, private IP address `192.168.1.1`), and you ran `ping 192.168.2.1`, which is the private address of the machine with IP address `W.X.Y.Z`. What needs to happen in order for this to work?

1. The gateway machine needs to know how to reach `192.168.2.1`. In other words, it needs to have a route to `192.168.2.1`.
2. Private IP addresses, such as those in the `192.168.x` range are not supposed to appear on the Internet at large. Instead, each packet you send to `192.168.2.1` will need to be wrapped up inside another packet. This packet will need to appear to be from `A.B.C.D`, and it will have to be sent to `W.X.Y.Z`. This process is called *encapsulation*.
3. Once this packet arrives at `W.X.Y.Z` it will need to "unencapsulated", and delivered to `192.168.2.1`.

You can think of this as requiring a "tunnel" between the two networks. The two "tunnel mouths" are the IP addresses `A.B.C.D` and `W.X.Y.Z`, and the tunnel must be told the addresses of the private IP addresses that will be allowed to pass through it. The tunnel is used to transfer traffic with private IP addresses across the public Internet.

This tunnel is created by using the generic interface, or `gif` devices on FreeBSD. As you can imagine, the `gif` interface on each gateway host must be configured with four IP addresses; two for the public IP addresses, and two for the private IP addresses.

Support for the gif device must be compiled in to the FreeBSD kernel on both machines. You can do this by adding the line:

```
device gif
```

to the kernel configuration files on both machines, and then compile, install, and reboot as normal.

Configuring the tunnel is a two step process. First the tunnel must be told what the outside (or public) IP addresses are, using gifconfig(8). Then the private IP addresses must be configured using ifconfig(8).

Note: In FreeBSD 5.X, the functionality provided by the gifconfig(8) utility has been merged into ifconfig(8).

On the gateway machine on network #1 you would run the following two commands to configure the tunnel.

```
gifconfig gif0 A.B.C.D W.X.Y.Z
ifconfig gif0 inet 192.168.1.1 192.168.2.1 netmask 0xffffffff
```

On the other gateway machine you run the same commands, but with the order of the IP addresses reversed.

```
gifconfig gif0 W.X.Y.Z A.B.C.D
ifconfig gif0 inet 192.168.2.1 192.168.1.1 netmask 0xffffffff
```

You can then run:

```
gifconfig gif0
```

to see the configuration. For example, on the network #1 gateway, you would see this:

```
# gifconfig gif0
gif0: flags=8011<UP,POINTTOPOINT,MULTICAST> mtu 1280
inet 192.168.1.1 --> 192.168.2.1 netmask 0xffffffff
physical address inet A.B.C.D --> W.X.Y.Z
```

As you can see, a tunnel has been created between the physical addresses `A.B.C.D` and `W.X.Y.Z`, and the traffic allowed through the tunnel is that between `192.168.1.1` and `192.168.2.1`.

This will also have added an entry to the routing table on both machines, which you can examine with the command `netstat -rn`. This output is from the gateway host on network #1.

```
# netstat -rn
Routing tables

Internet:
Destination      Gateway       Flags    Refs    Use    Netif  Expire
...
192.168.2.1      192.168.1.1   UH        0        0    gif0
...
```

As the “Flags” value indicates, this is a host route, which means that each gateway knows how to reach the other gateway, but they do not know how to reach the rest of their respective networks. That problem will be fixed shortly.

It is likely that you are running a firewall on both machines. This will need to be circumvented for your VPN traffic. You might want to allow all traffic between both networks, or you might want to include firewall rules that protect both ends of the VPN from one another.

It greatly simplifies testing if you configure the firewall to allow all traffic through the VPN. You can always tighten things up later. If you are using ipfw(8) on the gateway machines then a command like

```
ipfw add 1 allow ip from any to any via gif0
```

will allow all traffic between the two end points of the VPN, without affecting your other firewall rules. Obviously you will need to run this command on both gateway hosts.

This is sufficient to allow each gateway machine to ping the other. On `192.168.1.1`, you should be able to run

```
ping 192.168.2.1
```

and get a response, and you should be able to do the same thing on the other gateway machine.

However, you will not be able to reach internal machines on either network yet. This is because of the routing -- although the gateway machines know how to reach one another, they do not know how to reach the network behind each one.

To solve this problem you must add a static route on each gateway machine. The command to do this on the first gateway would be:

```
route add 192.168.2.0 192.168.2.1 netmask 0xffffff00
```

This says "In order to reach the hosts on the network `192.168.2.0`, send the packets to the host `192.168.2.1`". You will need to run a similar command on the other gateway, but with the `192.168.1.x` addresses instead.

IP traffic from hosts on one network will now be able to reach hosts on the other network.

That has now created two thirds of a VPN between the two networks, in as much as it is "virtual" and it is a "network". It is not private yet. You can test this using ping(8) and tcpdump(1). Log in to the gateway host and run

```
tcpdump dst host 192.168.2.1
```

In another log in session on the same host run

```
ping 192.168.2.1
```

You will see output that looks something like this:

```
16:10:24.018080 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:24.018109 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:25.018814 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:25.018847 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:26.028896 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:26.029112 192.168.1.1 > 192.168.2.1: icmp: echo reply
```

As you can see, the ICMP messages are going back and forth unencrypted. If you had used the `-s` parameter to tcpdump(1) to grab more bytes of data from the packets you would see more information.

Obviously this is unacceptable. The next section will discuss securing the link between the two networks so that it all traffic is automatically encrypted.

Summary:

- Configure both kernels with “pseudo-device gif”.
- Edit `/etc/rc.conf` on gateway host #1 and add the following lines (replacing IP addresses as necessary).

```
gifconfig_gif0="A.B.C.D W.X.Y.Z"
ifconfig_gif0="inet 192.168.1.1 192.168.2.1 netmask 0xffffffff"
static_routes="vpn"
route_vpn="192.168.2.0 192.168.2.1 netmask 0xffffff00"
```

- Edit your firewall script (`/etc/rc.firewall`, or similar) on both hosts, and add

```
ipfw add 1 allow ip from any to any via gif0
```

- Make similar changes to `/etc/rc.conf` on gateway host #2, reversing the order of IP addresses.

14.10.3.2 Step 2: Securing the link

To secure the link we will be using IPsec. IPsec provides a mechanism for two hosts to agree on an encryption key, and to then use this key in order to encrypt data between the two hosts.

The are two areas of configuration to be considered here.

1. There must be a mechanism for two hosts to agree on the encryption mechanism to use. Once two hosts have agreed on this mechanism there is said to be a “security association” between them.
2. There must be a mechanism for specifying which traffic should be encrypted. Obviously, you do not want to encrypt all your outgoing traffic -- you only want to encrypt the traffic that is part of the VPN. The rules that you put in place to determine what traffic will be encrypted are called “security policies”.

Security associations and security policies are both maintained by the kernel, and can be modified by userland programs. However, before you can do this you must configure the kernel to support IPsec and the Encapsulated Security Payload (ESP) protocol. This is done by configuring a kernel with:

```
options IPSEC
options IPSEC_ESP
```

and recompiling, reinstalling, and rebooting. As before you will need to do this to the kernels on both of the gateway hosts.

You have two choices when it comes to setting up security associations. You can configure them by hand between two hosts, which entails choosing the encryption algorithm, encryption keys, and so forth, or you can use daemons that implement the Internet Key Exchange protocol (IKE) to do this for you.

I recommend the latter. Apart from anything else, it is easier to set up.

Editing and displaying security policies is carried out using setkey(8). By analogy, `setkey` is to the kernel’s security policy tables as route(8) is to the kernel’s routing tables. `setkey` can also display the current security associations, and to continue the analogy further, is akin to `netstat -r` in that respect.

There are a number of choices for daemons to manage security associations with FreeBSD. This article will describe how to use one of these, racoon ——which is available from `security/ipsec-tools` in the FreeBSD Ports collection.

The **racoon** software must be run on both gateway hosts. On each host it is configured with the IP address of the other end of the VPN, and a secret key (which you choose, and must be the same on both gateways).

The two daemons then contact one another, confirm that they are who they say they are (by using the secret key that you configured). The daemons then generate a new secret key, and use this to encrypt the traffic over the VPN. They periodically change this secret, so that even if an attacker were to crack one of the keys (which is as theoretically close to unfeasible as it gets) it will not do them much good -- by the time they have cracked the key the two daemons have chosen another one.

The configuration file for racoon is stored in `${PREFIX}/etc/racoon`. You should find a configuration file there, which should not need to be changed too much. The other component of racoon's configuration, which you will need to change, is the "pre-shared key".

The default racoon configuration expects to find this in the file `${PREFIX}/etc/racoon/psk.txt`. It is important to note that the pre-shared key is *not* the key that will be used to encrypt your traffic across the VPN link, it is simply a token that allows the key management daemons to trust one another.

`psk.txt` contains a line for each remote site you are dealing with. In this example, where there are two sites, each `psk.txt` file will contain one line (because each end of the VPN is only dealing with one other end).

On gateway host #1 this line should look like this:

```
W.X.Y.Z            secret
```

That is, the *public* IP address of the remote end, whitespace, and a text string that provides the secret. Obviously, you should not use "secret" as your key -- the normal rules for choosing a password apply.

On gateway host #2 the line would look like this

```
A.B.C.D            secret
```

That is, the public IP address of the remote end, and the same secret key. `psk.txt` must be mode `0600` (i.e., only read/write to `root`) before racoon will run.

You must run racoon on both gateway machines. You will also need to add some firewall rules to allow the IKE traffic, which is carried over UDP to the ISAKMP (Internet Security Association Key Management Protocol) port. Again, this should be fairly early in your firewall ruleset.

```
ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp
ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp
```

Once racoon is running you can try pinging one gateway host from the other. The connection is still not encrypted, but racoon will then set up the security associations between the two hosts -- this might take a moment, and you may see this as a short delay before the ping commands start responding.

Once the security association has been set up you can view it using setkey(8). Run

```
setkey -D
```

on either host to view the security association information.

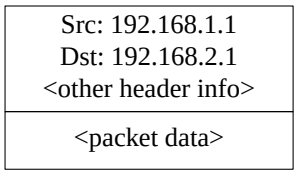
That's one half of the problem. They other half is setting your security policies.

To create a sensible security policy, let's review what's been set up so far. This discussions hold for both ends of the link.

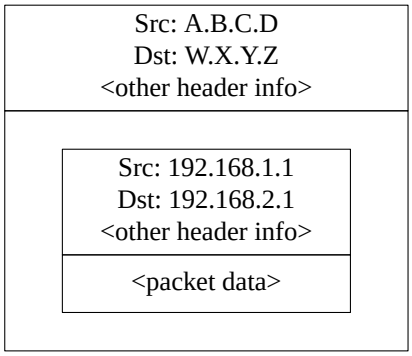
Each IP packet that you send out has a header that contains data about the packet. The header includes the IP addresses of both the source and destination. As we already know, private IP addresses, such as the `192.168.x.y`

range are not supposed to appear on the public Internet. Instead, they must first be encapsulated inside another packet. This packet must have the public source and destination IP addresses substituted for the private addresses.

So if your outgoing packet started looking like this:



Then it will be encapsulated inside another packet, looking something like this:



This encapsulation is carried out by the `gif` device. As you can see, the packet now has real IP addresses on the outside, and our original packet has been wrapped up as data inside the packet that will be put out on the Internet.

Obviously, we want all traffic between the VPNs to be encrypted. You might try putting this in to words, as:

"If a packet leaves from `A.B.C.D`, and it is destined for `W.X.Y.Z`, then encrypt it, using the necessary security associations."

"If a packet arrives from `W.X.Y.Z`, and it is destined for `A.B.C.D`, then decrypt it, using the necessary security associations."

That's close, but not quite right. If you did this, all traffic to and from `W.X.Y.Z`, even traffic that was not part of the VPN, would be encrypted. That's not quite what you want. The correct policy is as follows

"If a packet leaves from `A.B.C.D`, and that packet is encapsulating another packet, and it is destined for `W.X.Y.Z`, then encrypt it, using the necessary security associations."

"If a packet arrives from `W.X.Y.Z`, and that packet is encapsulating another packet, and it is destined for `A.B.C.D`, then decrypt it, using the necessary security associations."

A subtle change, but a necessary one.

Security policies are also set using setkey(8). setkey(8) features a configuration language for defining the policy. You can either enter configuration instructions via stdin, or you can use the `-f` option to specify a filename that contains configuration instructions.

The configuration on gateway host #1 (which has the public IP address `A.B.C.D`) to force all outbound traffic to `W.X.Y.Z` to be encrypted is:

```
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

Put these commands in a file (e.g. `/etc/ipsec.conf`) and then run

```
# setkey -f /etc/ipsec.conf
```

`spdadd` tells setkey(8) that we want to add a rule to the secure policy database. The rest of this line specifies which packets will match this policy. `A.B.C.D/32` and `W.X.Y.Z/32` are the IP addresses and netmasks that identify the network or hosts that this policy will apply to. In this case, we want it to apply to traffic between these two hosts. `ipencap` tells the kernel that this policy should only apply to packets that encapsulate other packets. `-P out` says that this policy applies to outgoing packets, and `ipsec` says that the packet will be secured.

The second line specifies how this packet will be encrypted. `esp` is the protocol that will be used, while `tunnel` indicates that the packet will be further encapsulated in an IPsec packet. The repeated use of `A.B.C.D` and `W.X.Y.Z` is used to select the security association to use, and the final `require` mandates that packets must be encrypted if they match this rule.

This rule only matches outgoing packets. You will need a similar rule to match incoming packets.

```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec esp/tunnel/W.X.Y.Z-A.B.C.D/require;
```

Note the `in` instead of `out` in this case, and the necessary reversal of the IP addresses.

The other gateway host (which has the public IP address `W.X.Y.Z`) will need similar rules.

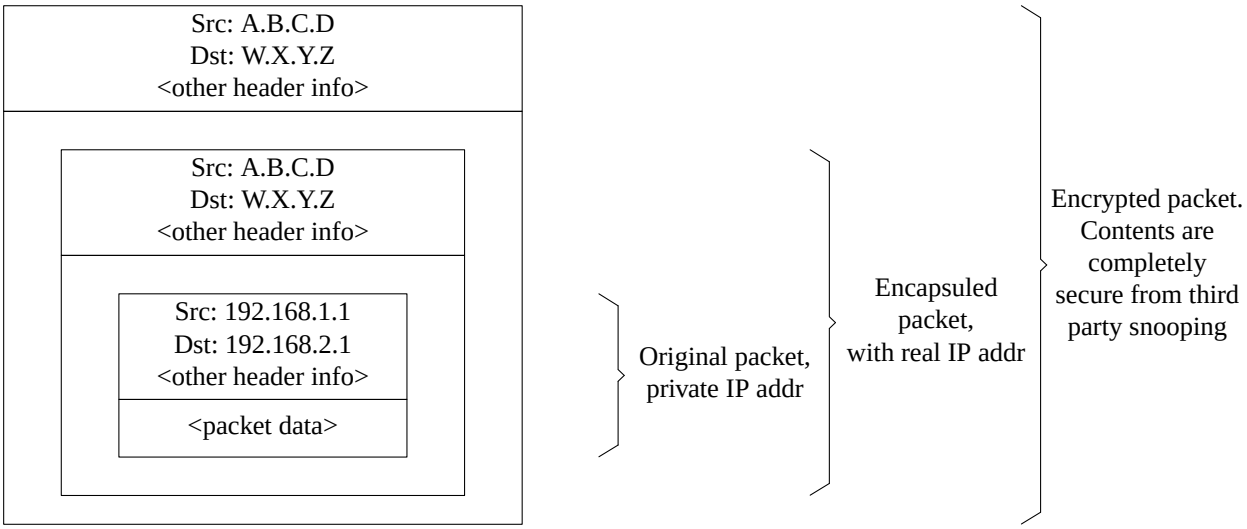
```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec esp/tunnel/W.X.Y.Z-A.B.C.D/require;
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

Finally, you need to add firewall rules to allow ESP and IPENCAP packets back and forth. These rules will need to be added to both hosts.

```
ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z
ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D
ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z
ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D
```

Because the rules are symmetric you can use the same rules on each gateway host.

Outgoing packets will now look something like this:



When they are received by the far end of the VPN they will first be decrypted (using the security associations that have been negotiated by racoon). Then they will enter the `gif` interface, which will unwrap the second layer, until you are left with the innermost packet, which can then travel in to the inner network.

You can check the security using the same ping(8) test from earlier. First, log in to the `A.B.C.D` gateway machine, and run:

```
tcpdump dst host 192.168.2.1
```

In another log in session on the same host run

```
ping 192.168.2.1
```

This time you should see output like the following:

```
XXX tcpdump output
```

Now, as you can see, tcpdump(1) shows the ESP packets. If you try to examine them with the `-s` option you will see (apparently) gibberish, because of the encryption.

Congratulations. You have just set up a VPN between two remote sites.

Summary

- Configure both kernels with:

  ```
  options IPSEC
  options IPSEC_ESP
  ```

- Install `security/ipsec-tools`. Edit `${PREFIX}/etc/racoon/psk.txt` on both gateway hosts, adding an entry for the remote host's IP address and a secret key that they both know. Make sure this file is mode 0600.
- Add the following lines to `/etc/rc.conf` on each host:

  ```
  ipsec_enable="YES"
  ipsec_file="/etc/ipsec.conf"
  ```

- Create an `/etc/ipsec.conf` on each host that contains the necessary spdadd lines. On gateway host #1 this would be:

```
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec
  esp/tunnel/A.B.C.D-W.X.Y.Z/require;
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec
  esp/tunnel/W.X.Y.Z-A.B.C.D/require;
```

  On gateway host #2 this would be:

```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec
  esp/tunnel/W.X.Y.Z-A.B.C.D/require;
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec
  esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

- Add firewall rules to allow IKE, ESP, and IPENCAP traffic to both hosts:

```
ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp
ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp
ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z
ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D
ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z
ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D
```

The previous two steps should suffice to get the VPN up and running. Machines on each network will be able to refer to one another using IP addresses, and all traffic across the link will be automatically and securely encrypted.

14.11 OpenSSH

Contributed by Chern Lee.

OpenSSH is a set of network connectivity tools used to access remote machines securely. It can be used as a direct replacement for `rlogin`, `rsh`, `rcp`, and `telnet`. Additionally, TCP/IP connections can be tunneled/forwarded securely through SSH. **OpenSSH** encrypts all traffic to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks.

OpenSSH is maintained by the OpenBSD project, and is based upon SSH v1.2.12 with all the recent bug fixes and updates. It is compatible with both SSH protocols 1 and 2. **OpenSSH** has been in the base system since FreeBSD 4.0.

14.11.1 Advantages of Using OpenSSH

Normally, when using telnet(1) or rlogin(1), data is sent over the network in an clear, un-encrypted form. Network sniffers anywhere in between the client and server can steal your user/password information or data transferred in your session. **OpenSSH** offers a variety of authentication and encryption methods to prevent this from happening.

14.11.2 Enabling sshd

The **sshd** daemon is enabled by default on FreeBSD 4.X. In FreeBSD 5.X and later enabling **sshd** is an option presented during a `Standard` install of FreeBSD. To see if **sshd** is enabled, check the `rc.conf` file for:

```
sshd_enable="YES"
```

This will load sshd(8), the daemon program for **OpenSSH**, the next time your system initializes. Alternatively, you can simply run directly the **sshd** daemon by typing `sshd` on the command line.

14.11.3 SSH Client

The ssh(1) utility works similarly to rlogin(1).

```
# ssh user@example.com
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'example.com' added to the list of known hosts.
user@example.com's password: *******
```

The login will continue just as it would have if a session was created using `rlogin` or `telnet`. SSH utilizes a key fingerprint system for verifying the authenticity of the server when the client connects. The user is prompted to enter `yes` only when connecting for the first time. Future attempts to login are all verified against the saved fingerprint key. The SSH client will alert you if the saved fingerprint differs from the received fingerprint on future login attempts. The fingerprints are saved in `~/.ssh/known_hosts`, or `~/.ssh/known_hosts2` for SSH v2 fingerprints.

By default, recent versions of the **OpenSSH** servers only accept SSH v2 connections. The client will use version 2 if possible and will fall back to version 1. The client can also be forced to use one or the other by passing it the `-1` or `-2` for version 1 or version 2, respectively. The version 1 compatability is maintained in the client for backwards compatability with older versions.

14.11.4 Secure Copy

The scp(1) command works similarly to rcp(1); it copies a file to or from a remote machine, except in a secure fashion.

```
#  scp user@example.com:/COPYRIGHT COPYRIGHT
user@example.com's password: *******
COPYRIGHT            100% |*****************************|  4735
00:00
#
```

Since the fingerprint was already saved for this host in the previous example, it is verified when using scp(1) here.

The arguments passed to scp(1) are similar to cp(1), with the file or files in the first argument, and the destination in the second. Since the file is fetched over the network, through SSH, one or more of the file arguments takes on the form `user@host:<path_to_remote_file>`.

14.11.5 Configuration

The system-wide configuration files for both the **OpenSSH** daemon and client reside within the `/etc/ssh` directory.

`ssh_config` configures the client settings, while `sshd_config` configures the daemon.

Additionally, the `sshd_program` (`/usr/sbin/sshd` by default), and `sshd_flags` `rc.conf` options can provide more levels of configuration.

14.11.6 ssh-keygen

Instead of using passwords, ssh-keygen(1) can be used to generate DSA or RSA keys to authenticate a user:

```
% ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_dsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_dsa.
Your public key has been saved in /home/user/.ssh/id_dsa.pub.
The key fingerprint is:
bb:48:db:f2:93:57:80:b6:aa:bc:f5:d5:ba:8f:79:17 user@host.example.com
```

ssh-keygen(1) will create a public and private key pair for use in authentication. The private key is stored in `~/.ssh/id_dsa` or `~/.ssh/id_rsa`, whereas the public key is stored in `~/.ssh/id_dsa.pub` or `~/.ssh/id_rsa.pub`, respectively for DSA and RSA key types. The public key must be placed in `~/.ssh/authorized_keys` of the remote machine in order for the setup to work. Similarly, RSA version 1 public keys should be placed in `~/.ssh/authorized_keys`.

This will allow connection to the remote machine based upon SSH keys instead of passwords.

If a passphrase is used in ssh-keygen(1), the user will be prompted for a password each time in order to use the private key. ssh-agent(1) can alleviate the strain of repeatedly entering long passphrases, and is explored in the Section 14.11.7 section below.

> **Warning:** The various options and files can be different according to the **OpenSSH** version you have on your system; to avoid problems you should consult the ssh-keygen(1) manual page.

14.11.7 ssh-agent and ssh-add

The ssh-agent(1) and ssh-add(1) utilities provide methods for **SSH** keys to be loaded into memory for use, without needing to type the passphrase each time.

The ssh-agent(1) utility will handle the authentication using the private key(s) that are loaded into it. ssh-agent(1) should be used to launch another application. At the most basic level, it could spawn a shell or at a more advanced level, a window manager.

To use ssh-agent(1) in a shell, first it will need to be spawned with a shell as an argument. Secondly, the identity needs to be added by running ssh-add(1) and providing it the passphrase for the private key. Once these steps have been completed the user will be able to ssh(1) to any host that has the corresponding public key installed. For example:

```
% ssh-agent csh
% ssh-add
Enter passphrase for /home/user/.ssh/id_dsa:
Identity added: /home/user/.ssh/id_dsa (/home/user/.ssh/id_dsa)
%
```

To use ssh-agent(1) in X11, a call to ssh-agent(1) will need to be placed in `~/.xinitrc`. This will provide the ssh-agent(1) services to all programs launched in X11. An example `~/.xinitrc` file might look like this:

```
exec ssh-agent startxfce4
```

This would launch ssh-agent(1), which would in turn launch **XFCE**, every time X11 starts. Then once that is done and X11 has been restarted so that the changes can take effect, simply run ssh-add(1) to load all of your SSH keys.

14.11.8 SSH Tunneling

OpenSSH has the ability to create a tunnel to encapsulate another protocol in an encrypted session.

The following command tells ssh(1) to create a tunnel for **telnet**:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

The `ssh` command is used with the following options:

`-2`

Forces `ssh` to use version 2 of the protocol. (Do not use if you are working with older SSH servers)

`-N`

Indicates no command, or tunnel only. If omitted, `ssh` would initiate a normal session.

`-f`

Forces `ssh` to run in the background.

`-L`

Indicates a local tunnel in *`localport:remotehost:remoteport`* fashion.

`user@foo.example.com`

The remote SSH server.

An SSH tunnel works by creating a listen socket on `localhost` on the specified port. It then forwards any connection received on the local host/port via the SSH connection to the specified remote host and port.

In the example, port *`5023`* on `localhost` is being forwarded to port *`23`* on `localhost` of the remote machine. Since *`23`* is **telnet**, this would create a secure **telnet** session through an SSH tunnel.

This can be used to wrap any number of insecure TCP protocols such as SMTP, POP3, FTP, etc.

Example 14-1. Using SSH to Create a Secure Tunnel for SMTP

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

```
220 mailserver.example.com ESMTP
```

This can be used in conjunction with an ssh-keygen(1) and additional user accounts to create a more seamless/hassle-free SSH tunneling environment. Keys can be used in place of typing a password, and the tunnels can be run as a separate user.

14.11.8.1 Practical SSH Tunneling Examples

14.11.8.1.1 Secure Access of a POP3 Server

At work, there is an SSH server that accepts connections from the outside. On the same office network resides a mail server running a POP3 server. The network, or network path between your home and office may or may not be completely trustable. Because of this, you need to check your e-mail in a secure manner. The solution is to create an SSH connection to your office's SSH server, and tunnel through to the mail server.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password: ******
```

When the tunnel is up and running, you can point your mail client to send POP3 requests to `localhost` port 2110. A connection here will be forwarded securely across the tunnel to `mail.example.com`.

14.11.8.1.2 Bypassing a Draconian Firewall

Some network administrators impose extremely draconian firewall rules, filtering not only incoming connections, but outgoing connections. You may be only given access to contact remote machines on ports 22 and 80 for SSH and web surfing.

You may wish to access another (perhaps non-work related) service, such as an Ogg Vorbis server to stream music. If this Ogg Vorbis server is streaming on some other port than 22 or 80, you will not be able to access it.

The solution is to create an SSH connection to a machine outside of your network's firewall, and use it to tunnel to the Ogg Vorbis server.

```
% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-system.example.org
user@unfirewalled-system.example.org's password: *******
```

Your streaming client can now be pointed to `localhost` port 8888, which will be forwarded over to `music.example.com` port 8000, successfully evading the firewall.

14.11.9 The `AllowUsers` Users Option

It is often a good idea to limit which users can log in and from where. The `AllowUsers` option is a good way to accomplish this. For example, to only allow the `root` user to log in from `192.168.1.32`, something like this would be appropriate in the `/etc/ssh/sshd_config` file:

```
AllowUsers root@192.168.1.32
```

To allow the user `admin` to log in from anywhere, just list the username by itself:

```
AllowUsers admin
```

Multiple users should be listed on the same line, like so:

```
AllowUsers root@192.168.1.32 admin
```

> **Note:** It is important that you list each user that needs to log in to this machine; otherwise they will be locked out.

After making changes to `/etc/ssh/sshd_config` you must tell sshd(8) to reload its config files, by running:

```
# /etc/rc.d/sshd reload
```

14.11.10 Further Reading

OpenSSH (http://www.openssh.com/)

ssh(1) scp(1) ssh-keygen(1) ssh-agent(1) ssh-add(1) ssh_config(5)

sshd(8) sftp-server(8) sshd_config(5)

14.12 File System Access Control Lists

Contributed by Tom Rhodes.

In conjunction with file system enhancements like snapshots, FreeBSD 5.0 and later offers the security of File System Access Control Lists (ACLs).

Access Control Lists extend the standard UNIX permission model in a highly compatible (POSIX.1e) way. This feature permits an administrator to make use of and take advantage of a more sophisticated security model.

To enable ACL support for UFS file systems, the following:

```
options UFS_ACL
```

must be compiled into the kernel. If this option has not been compiled in, a warning message will be displayed when attempting to mount a file system supporting ACLs. This option is included in the `GENERIC` kernel. ACLs rely on extended attributes being enabled on the file system. Extended attributes are natively supported in the next generation UNIX file system, UFS2.

> **Note:** A higher level of administrative overhead is required to configure extended attributes on UFS1 than on UFS2. The performance of extended attributes on UFS2 is also substantially higher. As a result, UFS2 is generally recommended in preference to UFS1 for use with access control lists.

ACLs are enabled by the mount-time administrative flag, `acls`, which may be added to `/etc/fstab`. The mount-time flag can also be automatically set in a persistent manner using tunefs(8) to modify a superblock ACLs flag in the file system header. In general, it is preferred to use the superblock flag for several reasons:

- The mount-time ACLs flag cannot be changed by a remount (mount(8) `-u`), only by means of a complete umount(8) and fresh mount(8). This means that ACLs cannot be enabled on the root file system after boot. It also means that you cannot change the disposition of a file system once it is in use.
- Setting the superblock flag will cause the file system to always be mounted with ACLs enabled even if there is not an `fstab` entry or if the devices re-order. This prevents accidental mounting of the file system without ACLs enabled, which can result in ACLs being improperly enforced, and hence security problems.

> **Note:** We may change the ACLs behavior to allow the flag to be enabled without a complete fresh mount(8), but we consider it desirable to discourage accidental mounting without ACLs enabled, because you can shoot your feet quite nastily if you enable ACLs, then disable them, then re-enable them without flushing the extended attributes. In general, once you have enabled ACLs on a file system, they should not be disabled, as the resulting file protections may not be compatible with those intended by the users of the system, and re-enabling ACLs may re-attach the previous ACLs to files that have since had their permissions changed, resulting in other unpredictable behavior.

File systems with ACLs enabled will show a + (plus) sign in their permission settings when viewed. For example:

```
drwx------  2 robert  robert  512 Dec 27 11:54 private
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

Here we see that the `directory1`, `directory2`, and `directory3` directories are all taking advantage of ACLs. The `public_html` directory is not.

14.12.1 Making Use of ACLs

The file system ACLs can be viewed by the getfacl(1) utility. For instance, to view the ACL settings on the `test` file, one would use the command:

```
% getfacl test
 #file:test
 #owner:1001
 #group:1001
 user::rw-
 group::r--
 other::r--
```

To change the ACL settings on this file, invoke the setfacl(1) utility. Observe:

```
% setfacl -k test
```

The `-k` flag will remove all of the currently defined ACLs from a file or file system. The more preferable method would be to use `-b` as it leaves the basic fields required for ACLs to work.

```
% setfacl -m u:trhodes:rwx,group:web:r--,o::--- test
```

In the aforementioned command, the `-m` option was used to modify the default ACL entries. Since there were no pre-defined entries, as they were removed by the previous command, this will restore the default options and assign the options listed. Take care to notice that if you add a user or group which does not exist on the system, an "`Invalid argument`" error will be printed to `stdout`.

14.13 Monitoring Third Party Security Issues

Contributed by Tom Rhodes.

In recent years, the security world has made many improvements to how vulnerability assessment is handled. The threat of system intrusion increases as third party utilities are installed and configured for virtually any operating system available today.

Vulnerability assessment is a key factor in security, and while FreeBSD releases advisories for the base system, doing so for every third party utility is beyond the FreeBSD Project's capability. There is a way to mitigate third party vulnerabilities and warn administrators of known security issues. A FreeBSD add on utility known as **Portaudit** exists solely for this purpose.

The `security/portaudit` port polls a database, updated and maintained by the FreeBSD Security Team and ports developers, for known security issues.

To begin using **Portaudit**, one must install it from the Ports Collection:

```
# cd /usr/ports/security/portaudit && make install clean
```

During the install process, the configuration files for periodic(8) will be updated, permitting **Portaudit** output in the daily security runs. Ensure the daily security run emails, which are sent to `root`'s email account, are being read. No more configuration will be required here.

After installation, an administrator can update the database and view known vulnerabilities in installed packages by invoking the following command:

```
# portaudit -Fda
```

> **Note:** The database will automatically be updated during the periodic(8) run; thus, the previous command is completely optional. It is only required for the following examples.

To audit the third party utilities installed as part of the Ports Collection at anytime, an administrator need only run the following command:

```
# portaudit -a
```

Portaudit will produce something like this for vulnerable packages:

```
Affected package: cups-base-1.1.22.0_1
Type of problem: cups-base -- HPGL buffer overflow vulnerability.
Reference: <http://www.FreeBSD.org/ports/portaudit/40a3bca2-6809-11d9-a9e7-0001020eed82.html>
```

```
1 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.
```

By pointing a web browser to the URL shown, an administrator may obtain more information about the vulnerability in question. This will include versions affected, by FreeBSD Port version, along with other web sites which may contain security advisories.

In short, **Portaudit** is a powerful utility and extremely useful when coupled with the **Portupgrade** port.

14.14 FreeBSD Security Advisories

Contributed by Tom Rhodes.

Like many production quality operating systems, FreeBSD publishes “Security Advisories”. These advisories are usually mailed to the security lists and noted in the Errata only after the appropriate releases have been patched. This section will work to explain what an advisory is, how to understand it, and what measures to take in order to patch a system.

14.14.1 What does an advisory look like?

The FreeBSD security advisories look similar to the one below, taken from the freebsd-security-notifications (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications) mailing list.

```
=============================================================================
FreeBSD-SA-XX:XX.UTIL                                       Security Advisory
                                                            The FreeBSD Project

Topic:          denial of service due to some problem❶

Category:       core❷
Module:         sys❸
Announced:      2003-09-23❹
Credits:        Person@EMAIL-ADDRESS❺
Affects:        All releases of FreeBSD❻
                FreeBSD 4-STABLE prior to the correction date
Corrected:      2003-09-23 16:42:59 UTC (RELENG_4, 4.9-PRERELEASE)
                2003-09-23 20:08:42 UTC (RELENG_5_1, 5.1-RELEASE-p6)
                2003-09-23 20:07:06 UTC (RELENG_5_0, 5.0-RELEASE-p15)
                2003-09-23 16:44:58 UTC (RELENG_4_8, 4.8-RELEASE-p8)
                2003-09-23 16:47:34 UTC (RELENG_4_7, 4.7-RELEASE-p18)
                2003-09-23 16:49:46 UTC (RELENG_4_6, 4.6-RELEASE-p21)
                2003-09-23 16:51:24 UTC (RELENG_4_5, 4.5-RELEASE-p33)
                2003-09-23 16:52:45 UTC (RELENG_4_4, 4.4-RELEASE-p43)
                2003-09-23 16:54:39 UTC (RELENG_4_3, 4.3-RELEASE-p39)❼
FreeBSD only:   NO❽

For general information regarding FreeBSD Security Advisories,
including descriptions of the fields above, security branches, and the
following sections, please visit
http://www.FreeBSD.org/security/.
```

```
I.   Background❾

II.  Problem Description(10)

III. Impact(11)

IV.  Workaround(12)

V.   Solution(13)

VI.  Correction details(14)

VII. References(15)
```

❶ The `Topic` field indicates exactly what the problem is. It is basically an introduction to the current security advisory and notes the utility with the vulnerability.

❷ The `Category` refers to the affected part of the system which may be one of `core`, `contrib`, or `ports`. The `core` category means that the vulnerability affects a core component of the FreeBSD operating system. The `contrib` category means that the vulnerability affects software contributed to the FreeBSD Project, such as **sendmail**. Finally the `ports` category indicates that the vulnerability affects add on software available as part of the Ports Collection.

❸ The `Module` field refers to the component location, for instance `sys`. In this example, we see that the module, `sys`, is affected; therefore, this vulnerability affects a component used within the kernel.

❹ The `Announced` field reflects the date said security advisory was published, or announced to the world. This means that the security team has verified that the problem does exist and that a patch has been committed to the FreeBSD source code repository.

❺ The `Credits` field gives credit to the individual or organization who noticed the vulnerability and reported it.

❻ The `Affects` field explains which releases of FreeBSD are affected by this vulnerability. For the kernel, a quick look over the output from `ident` on the affected files will help in determining the revision. For ports, the version number is listed after the port name in `/var/db/pkg`. If the system does not sync with the FreeBSD CVS repository and rebuild daily, chances are that it is affected.

❼ The `Corrected` field indicates the date, time, time offset, and release that was corrected.

❽ The `FreeBSD only` field indicates whether this vulnerability affects just FreeBSD, or if it affects other operating systems as well.

❾ The `Background` field gives information on exactly what the affected utility is. Most of the time this is why the utility exists in FreeBSD, what it is used for, and a bit of information on how the utility came to be.

(10) The `Problem Description` field explains the security hole in depth. This can include information on flawed code, or even how the utility could be maliciously used to open a security hole.

(11) The `Impact` field describes what type of impact the problem could have on a system. For example, this could be anything from a denial of service attack, to extra privileges available to users, or even giving the attacker superuser access.

(12) The `Workaround` field offers a feasible workaround to system administrators who may be incapable of upgrading the system. This may be due to time constraints, network availability, or a slew of other reasons. Regardless, security should not be taken lightly, and an affected system should either be patched or the security hole workaround should be implemented.

(13) The `Solution` field offers instructions on patching the affected system. This is a step by step tested and verified method for getting a system patched and working securely.

(14) The `Correction Details` field displays the CVS branch or release name with the periods changed to underscore characters. It also shows the revision number of the affected files within each branch.

(15) The `References` field usually offers sources of other information. This can included web URLs, books, mailing lists, and newsgroups.

14.15 Process Accounting

Contributed by Tom Rhodes.

Process accounting is a security method in which an administrator may keep track of system resources used, their allocation among users, provide for system monitoring, and minimally track a user's commands.

This indeed has its own positive and negative points. One of the positives is that an intrusion may be narrowed down to the point of entry. A negative is the amount of logs generated by process accounting, and the disk space they may require. This section will walk an administrator through the basics of process accounting.

14.15.1 Enable and Utilizing Process Accounting

Before making use of process accounting, it must be enabled. To do this, execute the following commands:

```
# touch /var/account/acct

# accton /var/account/acct

# echo 'accounting_enable="YES"' >> /etc/rc.conf
```

Once enabled, accounting will begin to track CPU stats, commands, etc. All accounting logs are in a non-human readable format and may be viewed using the sa(8) utility. If issued without any options, `sa` will print information relating to the number of per user calls, the total elapsed time in minutes, total CPU and user time in minutes, average number of I/O operations, etc.

To view information about commands being issued, one would use the lastcomm(1) utility. The `lastcomm` may be used to print out commands issued by users on specific ttys(5), for example:

```
# lastcomm ls
 trhodes ttyp1
```

Would print out all known usage of the `ls` by `trhodes` on the ttyp1 terminal.

Many other useful options exist and are explained in the lastcomm(1), acct(5) and sa(8) manual pages.

Notes

1. Under FreeBSD the standard login password may be up to 128 characters in length.

Chapter 15 Jails

Contributed by Matteo Riondato.

15.1 概述

本章將介紹FreeBSD jail 為何，以及如何運用之法。Jails 有時也常被認為是*chroot 環境* 的加强型替代品之一，它對系統管理者而言是非常好用的工具，此外，它的一些基本用法對進階使用者而言，也是相當有用。

讀完這章，您將了解：

- jail 是什麼，以及它在FreeBSD 上可以發揮的作用。
- 如何編譯、啓動、停止jail。
- jail 管理的基本概念：包括從jail 內部或主機本身。

其他有用的jail 相關資源還有：

- jail(8) 線上說明。這是有關`jail` 的完整說明——FreeBSD 內的啓動、停止、控制FreeBSD jail 相關管理工具。
- 郵遞論壇(mailing lists)及舊信檔案館(archives)。FreeBSD list server (http://lists.FreeBSD.org/mailman/listinfo) 所提供的FreeBSD general questions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) 及其他郵遞論壇的舊信，已有包括一堆jail 的有用資料。通常，搜尋舊信或者在freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) 上發問，也相當有效。

15.2 Jail 相關術語

為協助更容易理解FreeBSD 系統的jail 相關部分，以及它們與FreeBSD 其他部分的相互作用關係，以下列出本章將使用的術語：

chroot(2) (指令)

FreeBSD 的system call 之一，其作用為改變process 及其衍生process 所能運用的根目錄(`/` dir)。

chroot(2) (環境)

指在“chroot” 中運行的process 環境。這包括了類似檔案系統的可見部分、可用的UID、GID、網路卡及其他IPC 機制等資源。

jail(8) (command)

允許程式在jail 環境下執行的系統管理工具。

host (系統、process、帳號等等)

jail 環境的控制系統。host 系統可以使用全部可用的硬體資源，並能控制jail 環境內外的process。host 系統與jail 最大的差別在於：在host 系統中的superuser processes 並不像在jail 環境那樣處處受到一堆限制。

hosted (系統、process、帳號等等)

可用資源受到FreeBSD jail 限制的process、帳號、或其他設備資源。

15.3 背景故事

由於系統管理是困難又繁瑣的工作，因此人們開發許多好用工具，以讓管理工作更加簡單輕鬆。這些改善通常是讓系統能夠以更簡單的方式安裝、設定、維護，而有些改善目標則是系統安全的正確設定，使其能真正發揮原本用途，而非陷入安全風險之中。

FreeBSD 系統所提供的一種用於强化安全的工具就是*jail* 。Jail 是由Poul-Henning Kamp <phk@FreeBSD.org> 於FreeBSD 4.X 開始導入，而在FreeBSD 5.X 受到許多重大改良而集大成，成為强大而靈活的子系統，目前仍在持續開發、以提高其可用性、效能與安全。

15.3.1 何為Jail

BSD-like 作業系統自4.2BSD 起即提供chroot(2)。chroot(8) 可用來變更一組process 的根目錄位置，藉此建立與實體系統中相隔離的安全環境。處於chrooted 環境的process 會無法不能存取世外的檔案或資源。由於此因素，故即使攻擊者攻破某個處於chroot 環境的service，也不能攻破整個系統。chroot(8) 對於那些不太需要彈性或複雜又高級的簡單應用而言相當好用。另外，在引入chroot 概念的過程中，曾經發現許多可脫逃chroot 環境的方式，儘管這些問題在較新版本的FreeBSD kernel 均已修正，但很明顯地chroot(2) 絶非用於强化安全的理想解決方案。因此，勢必得實作新的子系統來解決這些問題。

這就是為何要開發*jail* 的最主要原因。

Jail 在各種方式分進合擊，改進傳統chroot(2) 環境的概念。在傳統的chroot(2) 環境中，只限制process 對於檔案系統的存取部分，而系統資源的其他部分(例如系統帳號、執行中的process、網路子系統)則是由chroot process 與host 系統的其他process 一起共享。Jail 以『虛擬化』來擴展這模型，不單只有檔案系統的存取，還延伸到系統帳號、FreeBSD kernel 的網路子系統及其他系統資源的虛擬化。關於這些jail 環境存取的細微調控，請參閱Section 15.5。

jail 具有下列四項特色：

- 目錄子樹(directory subtree) ——也就是進入jail 的起點。一旦進入jail 之後，process 就不再被允許跳到subtree 以外。&傳統會影響到man.chroot.2; 最初設計的安全問題，就不會再影響FreeBSD jail。
- 主機名稱(hostname) ——用於jail 的hostname。由於jail 主要用於網路服務，因此若各jail 皆有名稱，對於系統管理工作的簡化會相當有效。
- IP address ——是用來給jail 使用，並且在jail 生命週期內都無法變更。通常jail 的IP address 是現有網卡的alias address，但這並不是必須的。
- 指令(Command) ——準備在jail 內執行的完整路徑。這指令是相對於jail 環境的根目錄，視jail 環境的類型不同，而有所差異。

除了上述之外，jail 也可擁有自己的帳號及`root` 帳號。當然，這裡的`root` 權力會受制於jail 環境內。並且從host 系統的角度來看，jail 的`root` 並非無所不能的帳號。此外jail 的`root` 並不能執行其對於jail(8) 環境以外的一些關鍵性操作。關於`root` 的能力與限制，將於稍後的Section 15.5 介紹之。

15.4 建立和控制Jail

有些系統管理者把jail 分為下列兩種：“complete(完全)” jail ——通常包括完整的FreeBSD 系統；另一種則為“service(服務)” jail ——專門只跑某單一可能要用特殊權限的程式或service。這只是一種概念上的區分，並不影響如何建立jail 的過程。至於如何建立jail 在jail(8) 內有更詳細的說明：

```
# setenv D /here/is/the/jail
# mkdir -p $D ❶
# cd /usr/src
# make world DESTDIR=$D ❷
# cd etc/ [1]
# make distribution DESTDIR=$D ❸
# mount -t devfs $D/dev ❹
```

❶ 首先就是先為jail 找個家。該路徑是在host 系統中的jail 實體位置。習慣是放在`/usr/jail/`*`jailname`*，*`jailname`* 請替換為該jail 的hostname 以便辨別。通常`/usr` 會有足夠空間來存放jail 檔案系統，對於“complete” jail 而言，它通常包括了FreeBSD 預設安裝base system 所有檔案的拷貝檔。

❷ 該指令將會在jail 目錄中安裝所需的binary、library、manual 說明等。這些是以傳統的FreeBSD 方式完成——即首先先編譯所有檔案，接著再裝到目的地。

❸ 使用`distribution` 這個**make** target 來裝所有會用到的設定檔。簡單來說該動作就是把`/usr/src/etc/` 複製到jail 環境內的`/etc`，也就是`$D/etc/`。

❹ 對於jail 環境而言，devfs(8) 檔案系統的掛載並非必須，但另一方面，幾乎所有應用程式都會需要存取至少一個設備(device)，這主要取決於該程式目的而定。控制jail 所能存取的設備非常重要，因為不正確的設定，會讓攻擊者對jail 有機可趁。至於如何透過devfs(8) 來控制的規則，可以參閱devfs(8) 及devfs.conf(5) 說明。

裝好jail 之後，就可以用jail(8) 工具。jail(8) 需要四項必填參數，這些參數在Section 15.3.1 有介紹過。除了這四個參數之外，還可以指定其他參數，像是以特定帳號在jail 中執行process。*`command`* 參數取決於jail 類型而定；對於*virtual system(虛擬系統)* ，那麼就選擇`/etc/rc`，因為它會完成真正FreeBSD 系統啓動所需的操作。對於*service(服務)* jail 而言，執行的指令取決於將在jail 內執行的service 或應用程式而定。

Jail 通常要在系統開機時啓動，因此FreeBSD 的`rc` 機制提供一些便利的方式來簡化這些工作：

1. 開機時要啓動的jail 清單要加到rc.conf(5) 設定檔：

   ```
   jail_enable="YES"   # 若設為 NO 則表示不自動啓動 jail
   jail_list="www"     # 若有許多 jail 則請以空白隔開來寫
   ```

2. 對於每一筆在`jail_list` 所列出的jail，也要在rc.conf(5) 做出相對應的設定：

   ```
   jail_www_rootdir="/usr/jail/www"     # jail 的根目錄
   jail_www_hostname="www.example.org"  # jail 的 hostname
   jail_www_ip="192.168.0.10"           # jail 的 IP address
   jail_www_devfs_enable="YES"          # 在 jail 內 mount devfs
   jail_www_devfs_ruleset="www_ruleset" # jail 內所用的 devfs 規則表
   ```

   在rc.conf(5) 所預設的jail 啓動設定會跑`/etc/rc` 內的jail script，也就是說會假設jail 是完整的虛擬系統。若要用service jail 類型，則要另外指定啓動指令，方法是設定對應的`jail_`*`jailname`*`_exec_start` 設定。

Note: 若欲知道所有可用的選項清單，請參閱rc.conf(5) 說明。

也可以透過手動執行`/etc/rc.d/jail` script 來啓動或停止`rc.conf` 所設定的jail：

```
# /etc/rc.d/jail start www
# /etc/rc.d/jail stop www
```

目前尚無任何方法來很乾淨地關閉jail(8)。此乃因為正常用來關閉系統的指令，目前尚不能在jail 中使用。目前關閉jail 最佳的方式，是在jail 內執行下列指令，或者jail 外面透過jexec(8) 執行下列指令：

```
# sh /etc/rc.shutdown
```

詳情請參閱jail(8) 說明。

15.5 微調與管理

可以為jail 設定許多不同選項，並讓FreeBSD 的host 系統與jail 以各種不同方式組合搭配，以符合更多的應用用途。本節要介紹的是：

- 用以微調jail 行為與安全限制的選項。
- 可透過FreeBSD Ports Collection 安裝的高階jail 管理程式，搭配這些程式可以達到一些jail-based 解決方案。

15.5.1 FreeBSD 所提供的jail tuning 工具

對於jail 設定的微調，基本上都是透過設定sysctl(8) 變數來完成。系統提供一組sysctl 的特殊子樹，全部相關的選項都在該子樹內，也就是FreeBSD kernel 中的`security.jail.*` 子樹。下面則是與jail 相關的主要sysctl 設定及預設值，這些名稱都相當容易理解，如欲更進一步的資訊，請參閱jail(8) 與sysctl(8) 說明：

- `security.jail.set_hostname_allowed: 1`
- `security.jail.socket_unixiproute_only: 1`
- `security.jail.sysvipc_allowed: 0`
- `security.jail.enforce_statfs: 2`
- `security.jail.allow_raw_sockets: 0`
- `security.jail.chflags_allowed: 0`
- `security.jail.jailed: 0`

系統管理者可在*host system* 透過修改這些設定值來增加、取消Jail 內`root` 帳號的預設限制。請注意：有些限制是不能取消，在jail(8) 環境的`root` 不能掛載或卸載檔案系統。此外亦不能載入、卸載devfs(8) 規則、設定防火牆規則，或執行其他需修改kernel 資料的管理作業，例如設定kernel 的`securelevel` 值。

FreeBSD base system 內附一些基本工具，可用來查閱目前使用中的jail、並接上(attach) jail 以執行管理指令。jls(8) 及jexec(8) 均屬於FreeBSD base system 之一，可用來執行一些簡單工作：

- 列出有在使用的jail 及其相對應的jail identifier (JID)、IP address、hostname、路徑。
- 接上(Attach)正在運作中的jail，並在其中執行指令以進行管理工作。這點在當`root` 想乾淨關閉jail 時相當有用，jexec(8) 也可用在jail 中啓動shell 以便對其進行管理，比如：

```
# jexec 1 tcsh
```

15.5.2 FreeBSD Ports Collection 所提供的高階管理工具

在諸多third-party 所提供的jail 管理工具當中，`sysutils/jailutils` 是最完整也最好用的。該套件是由一系列jail(8) 管理小工具所組成的。詳情請參閱其網站介紹。

15.6 Jail 的應用

15.6.1 Service Jails

Contributed by Daniel Gerzo.

本節主要以Simon L. Nielsen <`simon@FreeBSD.org`> 寫的http://simon.nitro.dk/service-jails.html 為主，加上Ken Tom <`locals@gmail.com`> 所更新的文章。本節介紹如何設定FreeBSD 以jail(8) 功能來增加額外的安全層面。這部分假設您系統跑的是RELENG_6_0 或更新的版本，並且對本章先前部分均能理解。

15.6.1.1 Design

Jail 的主要問題之一在於如何對其進行更新、升級和管理。由於每個jail 都是從頭重新編譯，對於單一jail 而言，升級也許還不是很嚴重的問題，因為更新、升級並不會太麻煩。但對於一堆jail 而言，升級不僅會耗費太多時間，並相當枯燥乏味。

> **Warning:** 這些設定的前提是您對FreeBSD 使用、功能運用上有相當的經驗，若下面的設定對您來說太過複雜，建議您該考慮用較簡易的系統，像是`sysutils/ezjail`，其提供更簡單的FreeBSD jail 管理方式。

基本的想法是在不同的jail 中儘量以安全的方式來共用資源——採用唯讀的mount_nullfs(8) 掛載，來讓升級更簡單，並把各個service 放到不同的jail 的作法會更加可行。此外，其也提供對於如何增加、刪除、升級jail 的簡便方式。

> **Note:** service 常見的例子包括：HTTP server、DNS server、SMTP server 等等。

本節介紹的設定目的在於：

- 建立簡易且容易理解的jail 架構。也就是說*不必*為每個jail 都執行完整的installworld 。
- 讓jail 的新增、移除更簡單。
- 讓jail 的更新、升級更輕鬆。
- 可以跑自行打造的FreeBSD 分支。

· 對安全有更偏執狂的追求，儘可能降低被攻陷的可能。

· 儘量節省空間與inode。

如同先前所提到的，這設計主要是靠把唯讀的主要模版(也就是大家所熟知的**nullfs**)掛載到每個jail，並且讓每個jail 有個可讀、寫的設備，這設備可以是獨立實體硬碟、、分割區、或以vnode 為後端的md(4) 設備。在本例當中，我們採用可讀寫的**nullfs** 掛載。

下面的表則介紹檔案系統的配置：

· 每個jail 都會掛載到`/home/j` 底下的其中一個目錄。

· `/home/j/mroot` 則是每個jail 共用的模版，並對於所有jail 而言都是唯讀。

· 每個jail 在`/home/j` 底下都有一個相對應的空目錄。

· 每個jail 都會有`/s` 目錄，該目錄會連到系統的可讀寫部分。

· 每個jail 都會在`/home/j/skel` 目錄建立自屬的可讀寫空間。

· 每個jailspace (各jail 可讀寫的部分) 都建在`/home/js`>。

> **Note:** 這邊假設所有jail 都放在`/home` 分割區。當然，也可以依自身需求更改，但接下來的例子中，也要記得修改相對應的地方。

15.6.1.2 建立模版

本節將逐步介紹如何建立jail 要用的唯讀主模版。

建議先把FreeBSD 系統升級到最新的-RELEASE 分支，至於如何做請參閱Handbook 的相關章節 (http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/makeworld.html)。當更新完成之後，就要進行buildworld 程序，此外還要裝`sysutils/cpdup` 套件。我們將用portsnap(8) 來下載FreeBSD Ports Collection，在Handbook 中對Portsnap 章節 (http://www.FreeBSD.org/doc/zh_TW.Big5/books/handbook/portsnap.html) 中有相關介紹，初學者可以看看。

1. 首先，先建立唯讀的目錄結構給jail 放FreeBSD binary，接著到FreeBSD source tree 目錄，並安裝jail 模版：

```
# mkdir -p /home/j/mroot
# cd /usr/src
# make installworld DESTDIR=/home/j/mroot
```

2. 接著跟FreeBSD source tree 一樣，也把FreeBSD Ports Collection 放一份供jail 使用，以備**mergemaster** ：

```
# cd /home/j/mroot
# mkdir usr/ports
# portsnap -p /home/j/mroot/usr/ports fetch extract
# cpdup /usr/src /home/j/mroot/usr/src
```

3. 建立可讀寫部分的骨架：

```
# mkdir /home/j/skel /home/j/skel/home /home/j/skel/usr-X11R6 /home/j/skel/distfiles
# mv etc /home/j/skel
# mv usr/local /home/j/skel/usr-local
# mv tmp /home/j/skel
```

```
# mv var /home/j/skel
# mv root /home/j/skel
```

4. 用**mergemaster** 來裝漏掉的設定檔。接下來刪除**mergemaster** 所建立的多餘目錄：

```
# mergemaster -t /home/j/skel/var/tmp/temproot -D /home/j/skel -i
# cd /home/j/skel
# rm -R bin boot lib libexec mnt proc rescue sbin sys usr dev
```

5. 現在把可讀寫的檔案系統以symlink 方式連到唯讀的檔案系統。請確認symbolic link 是否有正確連到s/目錄，若目錄建立方式不對，或指向位置不對，可能會導致安裝失敗。

```
# cd /home/j/mroot
# mkdir s
# ln -s s/etc etc
# ln -s s/home home
# ln -s s/root root
# ln -s ../s/usr-local usr/local
# ln -s ../s/usr-X11R6 usr/X11R6
# ln -s ../../s/distfiles usr/ports/distfiles
# ln -s s/tmp tmp
# ln -s s/var var
```

6. 最後則是新增/home/j/skel/etc/make.conf ，並填入以下內容：

```
WRKDIRPREFIX?=  /s/portbuild
```

要設定WRKDIRPREFIX 才可以讓各jail 得以順利編譯FreeBSD ports。請記住ports 目錄是屬唯讀檔案系統。而搭配自訂的WRKDIRPREFIX 才可以讓各jail 在可讀寫空間進行編譯。

15.6.1.3 建立Jail

現在已經有完整的FreeBSD jail 模版，可以在/etc/rc.conf 內做相關設定。下面這例子則示範如何建立3個jail：“NS”、“MAIL”、“WWW”。

1. 在/etc/fstab 加上下列設定，以便讓系統自動掛載各jail 所需的唯讀模版與讀寫空間：

```
/home/j/mroot   /home/j/ns     nullfs  ro  0   0
/home/j/mroot   /home/j/mail   nullfs  ro  0   0
/home/j/mroot   /home/j/www    nullfs  ro  0   0
/home/js/ns     /home/j/ns/s   nullfs  rw  0   0
/home/js/mail   /home/j/mail/s nullfs  rw  0   0
/home/js/www    /home/j/www/s  nullfs  rw  0   0
```

> **Note:** 分割區的pass number 標示為0 就不會在開機時做fsck(8) 檢查；而分割區的dump number 標示為0 則不會被dump(8) 所備份。我們並不希望**fsck** 檢查**nullfs** 的掛載，或者讓**dump** 備份jail 內唯讀的nullfs 掛載。這也就是為何上述fstab 每行設定後面都有兩欄為“0 0”。

2. 在/etc/rc.conf 內設定jail：

```
jail_enable="YES"
jail_set_hostname_allow="NO"
jail_list="ns mail www"
jail_ns_hostname="ns.example.org"
```

```
jail_ns_ip="192.168.3.17"
jail_ns_rootdir="/usr/home/j/ns"
jail_ns_devfs_enable="YES"
jail_mail_hostname="mail.example.org"
jail_mail_ip="192.168.3.18"
jail_mail_rootdir="/usr/home/j/mail"
jail_mail_devfs_enable="YES"
jail_www_hostname="www.example.org"
jail_www_ip="62.123.43.14"
jail_www_rootdir="/usr/home/j/www"
jail_www_devfs_enable="YES"
```

> **Warning:** 之所以要把`jail_name_rootdir` 從`/home` 改為`/usr/home` 的原因在於FreeBSD 預設安裝的`/home` 目錄其實只是指向`/usr/home` 的symbolic link。而`jail_name_rootdir` 變數須為*實體目錄* 而非symbolic link，否則jail 會拒絕啓動。可以用realpath(1) 來決定該變數。詳情請參閱FreeBSD-SA-07:01.jail 安全通告。

3. 替每個jail 建立必須的唯讀檔案系統掛載點：

   ```
   # mkdir /home/j/ns /home/j/mail /home/j/www
   ```

4. 為每個jail 安裝可讀寫的模版。請注意這時要用`sysutils/cpdup` ，它能確保每個目錄都有正確複製。

   ```
   # mkdir /home/js
   # cpdup /home/j/skel /home/js/ns
   # cpdup /home/j/skel /home/js/mail
   # cpdup /home/j/skel /home/js/www
   ```

5. 如此一來就已完成jail 環境建立，可以準備好要用了。請先為各jail 掛載所須的檔案系統，再用`/etc/rc.d/jail` script 來啓動：

   ```
   # mount -a
   # /etc/rc.d/jail start
   ```

現在jail 應該就會啓動了。若要檢查是否有正常啓動，可以用jls(8) 指令來看，該指令的執行結果應該類似下面：

```
# jls
   JID  IP Address      Hostname                      Path
     3  192.168.3.17    ns.example.org                /home/j/ns
     2  192.168.3.18    mail.example.org              /home/j/mail
     1  62.123.43.14    www.example.org               /home/j/www
```

此時就可以登入各jail 並新增帳號與設定相關service 要用的daemon 。上面的`JID` 欄代表正在運作中的jail 編號。可用下列指令以在`JID` 編號3 的jail 執行管理工作：

```
# jexec 3 tcsh
```

15.6.1.4 升級

有時由於安全問題或者jail 內要用新功能，而需要把FreeBSD 系統升級到更新。這種安裝設計方式讓既有的jail 升級變得更加容易。jail 也可以把service 停機時間(downtime)降到最低，因為jail 只需在最後關鍵才需要重開。此外，萬一新版有問題的話，它也提供輕鬆回溯到舊版的功能。

1. 首先是照一般方式來升級host system，再新增臨時的唯讀模版`/home/j/mroot2`：

```
# mkdir /home/j/mroot2
# cd /usr/src
# make installworld DESTDIR=/home/j/mroot2
# cd /home/j/mroot2
# cpdup /usr/src usr/src
# mkdir s
```

同樣地，在執行`installworld` 時會建立一些用不著的目錄，請把這些砍掉：

```
# chflags -R 0 var
# rm -R etc var root usr/local tmp
```

2. 重新建立到主系統的可讀寫空間symlink：

```
# ln -s s/etc etc
# ln -s s/root root
# ln -s s/home home
# ln -s ../s/usr-local usr/local
# ln -s ../s/usr-X11R6 usr/X11R6
# ln -s s/tmp tmp
# ln -s s/var var
```

3. 現在可以關閉jail：

```
# /etc/rc.d/jail stop
```

4. 卸載原先的檔案系統：

```
# umount /home/j/ns/s
# umount /home/j/ns
# umount /home/j/mail/s
# umount /home/j/mail
# umount /home/j/www/s
# umount /home/j/www
```

> **Note:** 可讀寫空間(`/s`) 是掛載在唯讀檔案系統底下，故要先卸載。

5. 把舊的唯讀系統搬走，換成新的。如此一來，可同時保留先前系統的備份，以備萬一升級後有問題可回復。這邊的命名方式採新唯讀檔案系統的建立時間，此外原先FreeBSD Ports Collection 直接搬到新的檔案系統，以節省硬碟空間與inode ：

```
# cd /home/j
# mv mroot mroot.20060601
# mv mroot2 mroot
# mv mroot.20060601/usr/ports mroot/usr
```

6. 現在新的唯讀模版準備好了，只剩下重新掛載以及啓動jail：

```
# mount -a
# /etc/rc.d/jail start
```

最後以jls(8) 來檢查jail 是否均正常啓動。別忘了要在各jail 內執行mergemaster，還有相關設定檔以及rc.d scripts 均要更新。

Notes

1. FreeBSD 6.0(含) 之後就不需這步驟。

Chapter 16 Mandatory Access Control

Written by Tom Rhodes.

16.1 Synopsis

FreeBSD 5.X introduced new security extensions from the TrustedBSD project based on the POSIX.1e draft. Two of the most significant new security mechanisms are file system Access Control Lists (ACLs) and Mandatory Access Control (MAC) facilities. Mandatory Access Control allows new access control modules to be loaded, implementing new security policies. Some provide protections of a narrow subset of the system, hardening a particular service, while others provide comprehensive labeled security across all subjects and objects. The mandatory part of the definition comes from the fact that the enforcement of the controls is done by administrators and the system, and is not left up to the discretion of users as is done with discretionary access control (DAC, the standard file and System V IPC permissions on FreeBSD).

This chapter will focus on the Mandatory Access Control Framework (MAC Framework), and a set of pluggable security policy modules enabling various security mechanisms.

After reading this chapter, you will know:

- What MAC security policy modules are currently included in FreeBSD and their associated mechanisms.
- What MAC security policy modules implement as well as the difference between a labeled and non-labeled policy.
- How to efficiently configure a system to use the MAC framework.
- How to configure the different security policy modules included with the MAC framework.
- How to implement a more secure environment using the MAC framework and the examples shown.
- How to test the MAC configuration to ensure the framework has been properly implemented.

Before reading this chapter, you should:

- Understand UNIX and FreeBSD basics (Chapter 3).
- Be familiar with the basics of kernel configuration/compilation (Chapter 8).
- Have some familiarity with security and how it pertains to FreeBSD (Chapter 14).

> **Warning:** The improper use of the information in this chapter may cause loss of system access, aggravation of users, or inability to access the features provided by X11. More importantly, MAC should not be relied upon to completely secure a system. The MAC framework only augments existing security policy; without sound security practices and regular security checks, the system will never be completely secure.
>
> It should also be noted that the examples contained within this chapter are just that, examples. It is not recommended that these particular settings be rolled out on a production system. Implementing the various security policy modules takes a good deal of thought. One who does not fully understand exactly how everything works may find him or herself going back through the entire system and reconfiguring many files or directories.

16.1.1 What Will Not Be Covered

This chapter covers a broad range of security issues relating to the MAC framework; however, the development of new MAC security policy modules will not be covered. A number of security policy modules included with the MAC framework have specific characteristics which are provided for both testing and new module development. These include the mac_test(4), mac_stub(4) and mac_none(4). For more information on these security policy modules and the various mechanisms they provide, please review the manual pages.

16.2 Key Terms in this Chapter

Before reading this chapter, a few key terms must be explained. This will hopefully clear up any confusion that may occur and avoid the abrupt introduction of new terms and information.

- *compartment*: A compartment is a set of programs and data to be partitioned or separated, where users are given explicit access to specific components of a system. Also, a compartment represents a grouping, such as a work group, department, project, or topic. Using compartments, it is possible to implement a need-to-know security policy.
- *integrity*: Integrity, as a key concept, is the level of trust which can be placed on data. As the integrity of the data is elevated, so does the ability to trust that data.
- *label*: A label is a security attribute which can be applied to files, directories, or other items in the system. It could be considered a confidentiality stamp; when a label is placed on a file it describes the security properties for that specific file and will only permit access by files, users, resources, etc. with a similar security setting. The meaning and interpretation of label values depends on the policy configuration: while some policies might treat a label as representing the integrity or secrecy of an object, other policies might use labels to hold rules for access.
- *level*: The increased or decreased setting of a security attribute. As the level increases, its security is considered to elevate as well.
- *multilabel*: The `multilabel` property is a file system option which can be set in single user mode using the tunefs(8) utility, during the boot operation using the fstab(5) file, or during the creation of a new file system. This option will permit an administrator to apply different MAC labels on different objects. This option only applies to security policy modules which support labeling.
- *object*: An object or system object is an entity through which information flows under the direction of a *subject*. This includes directories, files, fields, screens, keyboards, memory, magnetic storage, printers or any other data storage/moving device. Basically, an object is a data container or a system resource; access to an *object* effectively means access to the data.
- *policy*: A collection of rules which defines how objectives are to be achieved. A *policy* usually documents how certain items are to be handled. This chapter will consider the term *policy* in this context as a *security policy*; i.e. a collection of rules which will control the flow of data and information and define whom will have access to that data and information.
- *sensitivity*: Usually used when discussing MLS. A sensitivity level is a term used to describe how important or secret the data should be. As the sensitivity level increases, so does the importance of the secrecy, or confidentiality of the data.
- *single label*: A single label is when the entire file system uses one label to enforce access control over the flow of data. When a file system has this set, which is any time when the `multilabel` option is not set, all files will

conform to the same label setting.

- *subject*: a subject is any active entity that causes information to flow between *objects*; e.g. a user, user processor, system process, etc. On FreeBSD, this is almost always a thread acting in a process on behalf of a user.

16.3 Explanation of MAC

With all of these new terms in mind, consider how the MAC framework augments the security of the system as a whole. The various security policy modules provided by the MAC framework could be used to protect the network and file systems, block users from accessing certain ports and sockets, and more. Perhaps the best use of the policy modules is to blend them together, by loading several security policy modules at a time for a multi-layered security environment. In a multi-layered security environment, multiple policy modules are in effect to keep security in check. This is different to a hardening policy, which typically hardens elements of a system that is used only for specific purposes. The only downside is administrative overhead in cases of multiple file system labels, setting network access control user by user, etc.

These downsides are minimal when compared to the lasting effect of the framework; for instance, the ability to pick and choose which policies are required for a specific configuration keeps performance overhead down. The reduction of support for unneeded policies can increase the overall performance of the system as well as offer flexibility of choice. A good implementation would consider the overall security requirements and effectively implement the various security policy modules offered by the framework.

Thus a system utilizing MAC features should at least guarantee that a user will not be permitted to change security attributes at will; all user utilities, programs and scripts must work within the constraints of the access rules provided by the selected security policy modules; and that total control of the MAC access rules are in the hands of the system administrator.

It is the sole duty of the system administrator to carefully select the correct security policy modules. Some environments may need to limit access control over the network; in these cases, the mac_portacl(4), mac_ifoff(4) and even mac_biba(4) policy modules might make good starting points. In other cases, strict confidentiality of file system objects might be required. Policy modules such as mac_bsdextended(4) and mac_mls(4) exist for this purpose.

Policy decisions could be made based on network configuration. Perhaps only certain users should be permitted access to facilities provided by ssh(1) to access the network or the Internet. The mac_portacl(4) would be the policy module of choice for these situations. But what should be done in the case of file systems? Should all access to certain directories be severed from other groups or specific users? Or should we limit user or utility access to specific files by setting certain objects as classified?

In the file system case, access to objects might be considered confidential to some users, but not to others. For an example, a large development team might be broken off into smaller groups of individuals. Developers in project A might not be permitted to access objects written by developers in project B. Yet they might need to access objects created by developers in project C; that is quite a situation indeed. Using the different security policy modules provided by the MAC framework; users could be divided into these groups and then given access to the appropriate areas without fear of information leakage.

Thus, each security policy module has a unique way of dealing with the overall security of a system. Module selection should be based on a well thought out security policy. In many cases, the overall policy may need to be revised and reimplemented on the system. Understanding the different security policy modules offered by the MAC framework will help administrators choose the best policies for their situations.

The default FreeBSD kernel does not include the option for the MAC framework; thus the following kernel option must be added before trying any of the examples or information in this chapter:

```
options MAC
```

And the kernel will require a rebuild and a reinstall.

> **Caution:** While the various manual pages for MAC policy modules state that they may be built into the kernel, it is possible to lock the system out of the network and more. Implementing MAC is much like implementing a firewall, care must be taken to prevent being completely locked out of the system. The ability to revert back to a previous configuration should be considered while the implementation of MAC remotely should be done with extreme caution.

16.4 Understanding MAC Labels

A MAC label is a security attribute which may be applied to subjects and objects throughout the system.

When setting a label, the user must be able to comprehend what it is, exactly, that is being done. The attributes available on an object depend on the policy module loaded, and that policy modules interpret their attributes in different ways. If improperly configured due to lack of comprehension, or the inability to understand the implications, the result will be the unexpected and perhaps, undesired, behavior of the system.

The security label on an object is used as a part of a security access control decision by a policy. With some policies, the label by itself contains all information necessary to make a decision; in other models, the labels may be processed as part of a larger rule set, etc.

For instance, setting the label of `biba/low` on a file will represent a label maintained by the Biba security policy module, with a value of “low”.

A few policy modules which support the labeling feature in FreeBSD offer three specific predefined labels. These are the low, high, and equal labels. Although they enforce access control in a different manner with each policy module, you can be sure that the low label will be the lowest setting, the equal label will set the subject or object to be disabled or unaffected, and the high label will enforce the highest setting available in the Biba and MLS policy modules.

Within single label file system environments, only one label may be used on objects. This will enforce one set of access permissions across the entire system and in many environments may be all that is required. There are a few cases where multiple labels may be set on objects or subjects in the file system. For those cases, the `multilabel` option may be passed to tunefs(8).

In the case of Biba and MLS, a numeric label may be set to indicate the precise level of hierarchical control. This numeric level is used to partition or sort information into different groups of say, classification only permitting access to that group or a higher group level.

In most cases the administrator will only be setting up a single label to use throughout the file system.

Hey wait, this is similar to DAC! I thought MAC gave control strictly to the administrator. That statement still holds true, to some extent as `root` is the one in control and who configures the policies so that users are placed in the appropriate categories/access levels. Alas, many policy modules can restrict the `root` user as well. Basic control over objects will then be released to the group, but `root` may revoke or modify the settings at any time. This is the hierarchal/clearance model covered by policies such as Biba and MLS.

16.4.1 Label Configuration

Virtually all aspects of label policy module configuration will be performed using the base system utilities. These commands provide a simple interface for object or subject configuration or the manipulation and verification of the configuration.

All configuration may be done by use of the setfmac(8) and setpmac(8) utilities. The `setfmac` command is used to set MAC labels on system objects while the `setpmac` command is used to set the labels on system subjects. Observe:

```
# setfmac biba/high test
```

If no errors occurred with the command above, a prompt will be returned. The only time these commands are not quiescent is when an error occurred; similarly to the chmod(1) and chown(8) commands. In some cases this error may be a "`Permission denied`" and is usually obtained when the label is being set or modified on an object which is restricted.[1] The system administrator may use the following commands to overcome this:

```
# setfmac biba/high test
"Permission denied"
# setpmac biba/low setfmac biba/high test
# getfmac test
test: biba/high
```

As we see above, `setpmac` can be used to override the policy module's settings by assigning a different label to the invoked process. The `getpmac` utility is usually used with currently running processes, such as **sendmail**: although it takes a process ID in place of a command the logic is extremely similar. If users attempt to manipulate a file not in their access, subject to the rules of the loaded policy modules, the "`Operation not permitted`" error will be displayed by the `mac_set_link` function.

16.4.1.1 Common Label Types

For the mac_biba(4), mac_mls(4) and mac_lomac(4) policy modules, the ability to assign simple labels is provided. These take the form of high, equal and low, what follows is a brief description of what these labels provide:

- The `low` label is considered the lowest label setting an object or subject may have. Setting this on objects or subjects will block their access to objects or subjects marked high.
- The `equal` label should only be placed on objects considered to be exempt from the policy.
- The `high` label grants an object or subject the highest possible setting.

With respect to each policy module, each of those settings will instate a different information flow directive. Reading the proper manual pages will further explain the traits of these generic label configurations.

16.4.1.1.1 Advanced Label Configuration

Numeric grade numbers used for `comparison:compartment+compartment`; thus the following:

```
biba/10:2+3+6(5:2+3-20:2+3+4+5+6)
```

May be interpreted as:

"Biba Policy Label" / "Grade 10" : "Compartments 2, 3 and 6" : ("grade 5 ...")

In this example, the first grade would be considered the "effective grade" with "effective compartments", the second grade is the low grade and the last one is the high grade. In most configurations these settings will not be used; indeed, they offered for more advanced configurations.

When applied to system objects, they will only have a current grade/compartments as opposed to system subjects as they reflect the range of available rights in the system, and network interfaces, where they are used for access control.

The grade and compartments in a subject and object pair are used to construct a relationship referred to as "dominance", in which a subject dominates an object, the object dominates the subject, neither dominates the other, or both dominate each other. The "both dominate" case occurs when the two labels are equal. Due to the information flow nature of Biba, you have rights to a set of compartments, "need to know", that might correspond to projects, but objects also have a set of compartments. Users may have to subset their rights using `su` or `setpmac` in order to access objects in a compartment from which they are not restricted.

16.4.1.2 Users and Label Settings

Users themselves are required to have labels so that their files and processes may properly interact with the security policy defined on the system. This is configured through the `login.conf` file by use of login classes. Every policy module that uses labels will implement the user class setting.

An example entry containing every policy module setting is displayed below:

```
default:\
 :copyright=/etc/COPYRIGHT:\
 :welcome=/etc/motd:\
 :setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
 :path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
 :manpath=/usr/share/man /usr/local/man:\
 :nologin=/usr/sbin/nologin:\
 :cputime=1h30m:\
 :datasize=8M:\
 :vmemoryuse=100M:\
 :stacksize=2M:\
 :memorylocked=4M:\
 :memoryuse=8M:\
 :filesize=8M:\
 :coredumpsize=8M:\
 :openfiles=24:\
 :maxproc=32:\
 :priority=0:\
 :requirehome:\
 :passwordtime=91d:\
 :umask=022:\
 :ignoretime@:\
 :label=partition/13,mls/5,biba/10(5-15),lomac10[2]:
```

The `label` option is used to set the user class default label which will be enforced by MAC. Users will never be permitted to modify this value, thus it can be considered not optional in the user case. In a real configuration, however, the administrator will never wish to enable every policy module. It is recommended that the rest of this chapter be reviewed before any of this configuration is implemented.

Note: Users may change their label after the initial login; however, this change is subject constraints of the policy. The example above tells the Biba policy that a process's minimum integrity is 5, its maximum is 15, but the default effective label is 10. The process will run at 10 until it chooses to change label, perhaps due to the user using the setpmac command, which will be constrained by Biba to the range set at login.

In all cases, after a change to `login.conf`, the login class capability database must be rebuilt using `cap_mkdb` and this will be reflected throughout every forthcoming example or discussion.

It is useful to note that many sites may have a particularly large number of users requiring several different user classes. In depth planning is required as this may get extremely difficult to manage.

Future versions of FreeBSD will include a new way to deal with mapping users to labels; however, this will not be available until some time after FreeBSD 5.3.

16.4.1.3 Network Interfaces and Label Settings

Labels may also be set on network interfaces to help control the flow of data across the network. In all cases they function in the same way the policies function with respect to objects. Users at high settings in `biba`, for example, will not be permitted to access network interfaces with a label of low.

The `maclabel` may be passed to `ifconfig` when setting the MAC label on network interfaces. For example:

```
# ifconfig bge0 maclabel biba/equal
```

will set the MAC label of `biba/equal` on the bge(4) interface. When using a setting similar to `biba/high(low-high)` the entire label should be quoted; otherwise an error will be returned.

Each policy module which supports labeling has a tunable which may be used to disable the MAC label on network interfaces. Setting the label to `equal` will have a similar effect. Review the output from `sysctl`, the policy manual pages, or even the information found later in this chapter for those tunables.

16.4.2 Singlelabel or Multilabel?

By default the system will use the `singlelabel` option. But what does this mean to the administrator? There are several differences which, in their own right, offer pros and cons to the flexibility in the systems security model.

The `singlelabel` only permits for one label, for instance `biba/high` to be used for each subject or object. It provides for lower administration overhead but decreases the flexibility of policies which support labeling. Many administrators may want to use the `multilabel` option in their security policy.

The `multilabel` option will permit each subject or object to have its own independent MAC label in place of the standard `singlelabel` option which will allow only one label throughout the partition. The `multilabel` and `single` label options are only required for the policies which implement the labeling feature, including the Biba, Lomac, MLS and SEBSD policies.

In many cases, the `multilabel` may not need to be set at all. Consider the following situation and security model:

- FreeBSD web-server using the MAC framework and a mix of the various policies.
- This machine only requires one label, `biba/high`, for everything in the system. Here the file system would not require the `multilabel` option as a single label will always be in effect.

- But, this machine will be a web server and should have the web server run at `biba/low` to prevent write up capabilities. The Biba policy and how it works will be discussed later, so if the previous comment was difficult to interpret just continue reading and return. The server could use a separate partition set at `biba/low` for most if not all of its runtime state. Much is lacking from this example, for instance the restrictions on data, configuration and user settings; however, this is just a quick example to prove the aforementioned point.

If any of the non-labeling policies are to be used, then the `multilabel` option would never be required. These include the `seeotheruids`, `portacl` and `partition` policies.

It should also be noted that using `multilabel` with a partition and establishing a security model based on `multilabel` functionality could open the doors for higher administrative overhead as everything in the file system would have a label. This includes directories, files, and even device nodes.

The following command will set `multilabel` on the file systems to have multiple labels. This may only be done in single user mode:

```
# tunefs -l enable /
```

This is not a requirement for the swap file system.

> **Note:** Some users have experienced problems with setting the `multilabel` flag on the root partition. If this is the case, please review the Section 16.16 of this chapter.

16.4.3 Controlling MAC with Tunables

Without any modules loaded, there are still some parts of MAC which may be configured using the `sysctl` interface. These tunables are described below and in all cases the number one (1) means enabled while the number zero (0) means disabled:

- `security.mac.enforce_fs` defaults to one (1) and enforces MAC file system policies on the file systems.
- `security.mac.enforce_kld` defaults to one (1) and enforces MAC kernel linking policies on the dynamic kernel linker (see kld(4)).
- `security.mac.enforce_network` defaults to one (1) and enforces MAC network policies.
- `security.mac.enforce_pipe` defaults to one (1) and enforces MAC policies on pipes.
- `security.mac.enforce_process` defaults to one (1) and enforces MAC policies on processes which utilize inter-process communication.
- `security.mac.enforce_socket` defaults to one (1) and enforces MAC policies on sockets (see the socket(2) manual page).
- `security.mac.enforce_system` defaults to one (1) and enforces MAC policies on system activities such as accounting and rebooting.
- `security.mac.enforce_vm` defaults to one (1) and enforces MAC policies on the virtual memory system.

> **Note:** Every policy or MAC option supports tunables. These usually hang off of the `security.mac.<policyname>` tree. To view all of the tunables from MAC use the following command:

```
# sysctl -da | grep mac
```

This should be interpreted as all of the basic MAC policies are enforced by default. If the modules were built into the kernel the system would be extremely locked down and most likely unable to communicate with the local network or connect to the Internet, etc. This is why building the modules into the kernel is not completely recommended. Not because it limits the ability to disable features on the fly with `sysctl`, but it permits the administrator to instantly switch the policies of a system without the requirement of rebuilding and reinstalling a new system.

16.5 Module Configuration

Every module included with the MAC framework may be either compiled into the kernel as noted above or loaded as a run-time kernel module. The recommended method is to add the module name to the `/boot/loader.conf` file so that it will load during the initial boot operation.

The following sections will discuss the various MAC modules and cover their features. Implementing them into a specific environment will also be a consideration of this chapter. Some modules support the use of labeling, which is controlling access by enforcing a label such as "this is allowed and this is not". A label configuration file may control how files may be accessed, network communication can be exchanged, and more. The previous section showed how the `multilabel` flag could be set on file systems to enable per-file or per-partition access control.

A single label configuration would enforce only one label across the system, that is why the `tunefs` option is called `multilabel`.

16.5.1 The MAC seeotheruids Module

Module name: `mac_seeotheruids.ko`

Kernel configuration line: `options MAC_SEEOTHERUIDS`

Boot option: `mac_seeotheruids_load="YES"`

The mac_seeotheruids(4) module mimics and extends the `security.bsd.see_other_uids` and `security.bsd.see_other_gids` `sysctl` tunables. This option does not require any labels to be set before configuration and can operate transparently with the other modules.

After loading the module, the following `sysctl` tunables may be used to control the features:

- `security.mac.seeotheruids.enabled` will enable the module's features and use the default settings. These default settings will deny users the ability to view processes and sockets owned by other users.
- `security.mac.seeotheruids.specificgid_enabled` will allow a certain group to be exempt from this policy. To exempt specific groups from this policy, use the `security.mac.seeotheruids.specificgid=`*`XXX`* `sysctl` tunable. In the above example, the *`XXX`* should be replaced with the numeric group ID to be exempted.
- `security.mac.seeotheruids.primarygroup_enabled` is used to exempt specific primary groups from this policy. When using this tunable, the `security.mac.seeotheruids.specificgid_enabled` may not be set.

16.6 The MAC bsdextended Module

Module name: `mac_bsdextended.ko`

Kernel configuration line: `options MAC_BSDEXTENDED`

Boot option: `mac_bsdextended_load="YES"`

The mac_bsdextended(4) module enforces the file system firewall. This module's policy provides an extension to the standard file system permissions model, permitting an administrator to create a firewall-like ruleset to protect files, utilities, and directories in the file system hierarchy.

The policy may be created using a utility, ugidfw(8), that has a syntax similar to that of ipfw(8). More tools can be written by using the functions in the libugidfw(3) library.

Extreme caution should be taken when working with this module; incorrect use could block access to certain parts of the file system.

16.6.1 Examples

After the mac_bsdextended(4) module has been loaded, the following command may be used to list the current rule configuration:

```
# ugidfw list
0 slots, 0 rules
```

As expected, there are no rules defined. This means that everything is still completely accessible. To create a rule which will block all access by users but leave `root` unaffected, simply run the following command:

```
# ugidfw add subject not uid root new object not uid root mode n
```

> **Note:** In releases prior to FreeBSD 5.3, the *add* parameter did not exist. In those cases the *set* should be used instead. See below for a command example.

This is a very bad idea as it will block all users from issuing even the most simple commands, such as `ls`. A more patriotic list of rules might be:

```
# ugidfw set 2 subject uid user1 object uid user2 mode n
# ugidfw set 3 subject uid user1 object gid user2 mode n
```

This will block any and all access, including directory listings, to *user2*'s home directory from the username `user1`.

In place of `user1`, the `not uid` *user2* could be passed. This will enforce the same access restrictions above for all users in place of just one user.

> **Note:** The `root` user will be unaffected by these changes.

This should give a general idea of how the mac_bsdextended(4) module may be used to help fortify a file system. For more information, see the mac_bsdextended(4) and the ugidfw(8) manual pages.

16.7 The MAC ifoff Module

Module name: `mac_ifoff.ko`

Kernel configuration line: `options MAC_IFOFF`

Boot option: `mac_ifoff_load="YES"`

The mac_ifoff(4) module exists solely to disable network interfaces on the fly and keep network interfaces from being brought up during the initial system boot. It does not require any labels to be set up on the system, nor does it have a dependency on other MAC modules.

Most of the control is done through the `sysctl` tunables listed below.

- `security.mac.ifoff.lo_enabled` will enable/disable all traffic on the loopback (lo(4)) interface.
- `security.mac.ifoff.bpfrecv_enabled` will enable/disable all traffic on the Berkeley Packet Filter interface (bpf(4))
- `security.mac.ifoff.other_enabled` will enable/disable traffic on all other interfaces.

One of the most common uses of mac_ifoff(4) is network monitoring in an environment where network traffic should not be permitted during the boot sequence. Another suggested use would be to write a script which uses `security/aide` to automatically block network traffic if it finds new or altered files in protected directories.

16.8 The MAC portacl Module

Module name: `mac_portacl.ko`

Kernel configuration line: `MAC_PORTACL`

Boot option: `mac_portacl_load="YES"`

The mac_portacl(4) module is used to limit binding to local TCP and UDP ports using a variety of `sysctl` variables. In essence mac_portacl(4) makes it possible to allow non-`root` users to bind to specified privileged ports, i.e. ports fewer than 1024.

Once loaded, this module will enable the MAC policy on all sockets. The following tunables are available:

- `security.mac.portacl.enabled` will enable/disable the policy completely.[2]
- `security.mac.portacl.port_high` will set the highest port number that mac_portacl(4) will enable protection for.
- `security.mac.portacl.suser_exempt` will, when set to a non-zero value, exempt the `root` user from this policy.
- `security.mac.portacl.rules` will specify the actual mac_portacl policy; see below.

The actual `mac_portacl` policy, as specified in the `security.mac.portacl.rules` sysctl, is a text string of the form: `rule[,rule,...]` with as many rules as needed. Each rule is of the form: `idtype:id:protocol:port`. The *`idtype`* parameter can be `uid` or `gid` and used to interpret the *`id`* parameter as either a user id or group id, respectively. The *`protocol`* parameter is used to determine if the rule should apply to TCP or UDP by setting the parameter to `tcp` or `udp`. The final *`port`* parameter is the port number to allow the specified user or group to bind to.

> **Note:** Since the ruleset is interpreted directly by the kernel only numeric values can be used for the user ID, group ID, and port parameters. I.e. user, group, and port service names cannot be used.

By default, on UNIX-like systems, ports fewer than 1024 can only be used by/bound to privileged processes, i.e. those run as `root`. For mac_portacl(4) to allow non-privileged processes to bind to ports below 1024 this standard UNIX restriction has to be disabled. This can be accomplished by setting the sysctl(8) variables `net.inet.ip.portrange.reservedlow` and `net.inet.ip.portrange.reservedhigh` to zero.

See the examples below or review the mac_portacl(4) manual page for further information.

16.8.1 Examples

The following examples should illuminate the above discussion a little better:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0 net.inet.ip.portrange.reservedhigh=0
```

First we set mac_portacl(4) to cover the standard privileged ports and disable the normal UNIX bind restrictions.

```
# sysctl security.mac.portacl.suser_exempt=1
```

The `root` user should not be crippled by this policy, thus set the `security.mac.portacl.suser_exempt` to a non-zero value. The mac_portacl(4) module has now been set up to behave the same way UNIX-like systems behave by default.

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

Allow the user with UID 80 (normally the `www` user) to bind to port 80. This can be used to allow the `www` user to run a web server without ever having `root` privilege.

```
# sysctl security.mac.portacl.rules=uid:1001:tcp:110,uid:1001:tcp:995
```

Permit the user with the UID of 1001 to bind to the TCP ports 110 (“pop3”) and 995 (“pop3s”). This will permit this user to start a server that accepts connections on ports 110 and 995.

16.9 MAC Policies with Labeling Features

The next few sections will discuss MAC policies which use labels.

From here on this chapter will focus on the features of mac_biba(4), mac_lomac(4), mac_partition(4), and mac_mls(4).

> **Note:** This is an example configuration only and should not be considered for a production implementation. The goal is to document and show the syntax as well as examples for implementation and testing.

For these policies to work correctly several preparations must be made.

16.9.1 Preparation for Labeling Policies

The following changes are required in the `login.conf` file:

- An `insecure` class, or another class of similar type, must be added. The login class of `insecure` is not required and just used as an example here; different configurations may use another class name.
- The `insecure` class should have the following settings and definitions. Several of these can be altered but the line which defines the default label is a requirement and must remain.

```
insecure:\
 :copyright=/etc/COPYRIGHT:\
 :welcome=/etc/motd:\
 :setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
 :path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
 :manpath=/usr/share/man /usr/local/man:\
 :nologin=/usr/sbin/nologin:\
 :cputime=1h30m:\
 :datasize=8M:\
 :vmemoryuse=100M:\
 :stacksize=2M:\
 :memorylocked=4M:\
 :memoryuse=8M:\
 :filesize=8M:\
 :coredumpsize=8M:\
 :openfiles=24:\
 :maxproc=32:\
 :priority=0:\
 :requirehome:\
 :passwordtime=91d:\
 :umask=022:\
 :ignoretime@:\
 :label=partition/13,mls/5,biba/low:
```

  The cap_mkdb(1) command needs to be ran on login.conf(5) before any of the users can be switched over to the new class.

  The `root` username should also be placed into a login class; otherwise, almost every command executed by `root` will require the use of `setpmac`.

  > **Warning:** Rebuilding the `login.conf` database may cause some errors later with the daemon class. Simply uncommenting the daemon account and rebuilding the database should alleviate these issues.

- Ensure that all partitions on which MAC labeling will be implemented support the `multilabel`. We must do this because many of the examples here contain different labels for testing purposes. Review the output from the `mount` command as a precautionary measure.
- Switch any users who will have the higher security mechanisms enforced over to the new user class. A quick run of pw(8) or vipw(8) should do the trick.

16.10 The MAC partition Module

Module name: `mac_partition.ko`

Kernel configuration line: `options MAC_PARTITION`

Boot option: `mac_partition_load="YES"`

The mac_partition(4) policy will drop processes into specific "partitions" based on their MAC label. Think of it as a special type of jail(8), though that is hardly a worthy comparison.

This is one module that should be added to the loader.conf(5) file so that it loads and enables the policy during the boot process.

Most configuration for this policy is done using the setpmac(8) utility which will be explained below. The following `sysctl` tunable is available for this policy:

· `security.mac.partition.enabled` will enable the enforcement of MAC process partitions.

When this policy is enabled, users will only be permitted to see their processes but will not be permitted to work with certain utilities. For instance, a user in the `insecure` class above will not be permitted to access the `top` command as well as many other commands that must spawn a process.

To set or drop utilities into a partition label, use the `setpmac` utility:

```
# setpmac partition/13 top
```

This will add the `top` command to the label set on users in the `insecure` class. Note that all processes spawned by users in the `insecure` class will stay in the `partition/13` label.

16.10.1 Examples

The following command will show you the partition label and the process list:

```
# ps Zax
```

This next command will allow the viewing of another user's process partition label and that user's currently running processes:

```
# ps -ZU trhodes
```

> **Note:** Users can see processes in `root`'s label unless the mac_seeotheruids(4) policy is loaded.

A really crafty implementation could have all of the services disabled in `/etc/rc.conf` and started by a script that starts them with the proper labeling set.

> **Note:** The following policies support integer settings in place of the three default labels offered. These options, including their limitations, are further explained in the module manual pages.

16.11 The MAC Multi-Level Security Module

Module name: `mac_mls.ko`

Kernel configuration line: `options MAC_MLS`

Boot option: `mac_mls_load="YES"`

The mac_mls(4) policy controls access between subjects and objects in the system by enforcing a strict information flow policy.

In MLS environments, a "clearance" level is set in each subject or objects label, along with compartments. Since these clearance or sensibility levels can reach numbers greater than six thousand; it would be a daunting task for any system administrator to thoroughly configure each subject or object. Thankfully, three "instant" labels are already included in this policy.

These labels are `mls/low`, `mls/equal` and `mls/high`. Since these labels are described in depth in the manual page, they will only get a brief description here:

- The `mls/low` label contains a low configuration which permits it to be dominated by all other objects. Anything labeled with `mls/low` will have a low clearance level and not be permitted to access information of a higher level. In addition, this label will prevent objects of a higher clearance level from writing or passing information on to them.
- The `mls/equal` label should be placed on objects considered to be exempt from the policy.
- The `mls/high` label is the highest level of clearance possible. Objects assigned this label will hold dominance over all other objects in the system; however, they will not permit the leaking of information to objects of a lower class.

MLS provides for:

- A hierarchical security level with a set of non hierarchical categories;
- Fixed rules: no read up, no write down (a subject can have read access to objects on its own level or below, but not above. Similarly, a subject can have write access to objects on its own level or above but not beneath.);
- Secrecy (preventing inappropriate disclosure of data);
- Basis for the design of systems that concurrently handle data at multiple sensitivity levels (without leaking information between secret and confidential).

The following `sysctl` tunables are available for the configuration of special services and interfaces:

- `security.mac.mls.enabled` is used to enable/disable the MLS policy.
- `security.mac.mls.ptys_equal` will label all pty(4) devices as `mls/equal` during creation.
- `security.mac.mls.revocation_enabled` is used to revoke access to objects after their label changes to a label of a lower grade.
- `security.mac.mls.max_compartments` is used to set the maximum number of compartment levels with objects; basically the maximum compartment number allowed on a system.

To manipulate the MLS labels, the setfmac(8) command has been provided. To assign a label to an object, issue the following command:

```
# setfmac mls/5 test
```

To get the MLS label for the file `test` issue the following command:

```
# getfmac test
```

This is a summary of the MLS policy's features. Another approach is to create a master policy file in `/etc` which specifies the MLS policy information and to feed that file into the `setfmac` command. This method will be explained after all policies are covered.

Observations: an object with lower clearance is unable to observe higher clearance processes. A basic policy would be to enforce `mls/high` on everything not to be read, even if it needs to be written. Enforce `mls/low` on everything not to be written, even if it needs to be read. And finally enforce `mls/equal` on the rest. All users marked `insecure` should be set at `mls/low`.

16.12 The MAC Biba Module

Module name: `mac_biba.ko`

Kernel configuration line: `options MAC_BIBA`

Boot option: `mac_biba_load="YES"`

The mac_biba(4) module loads the MAC Biba policy. This policy works much like that of the MLS policy with the exception that the rules for information flow are slightly reversed. This is said to prevent the downward flow of sensitive information whereas the MLS policy prevents the upward flow of sensitive information; thus, much of this section can apply to both policies.

In Biba environments, an "integrity" label is set on each subject or object. These labels are made up of hierarchal grades, and non-hierarchal components. As an object's or subject's grade ascends, so does its integrity.

Supported labels are `biba/low`, `biba/equal`, and `biba/high`; as explained below:

- The `biba/low` label is considered the lowest integrity an object or subject may have. Setting this on objects or subjects will block their write access to objects or subjects marked high. They still have read access though.
- The `biba/equal` label should only be placed on objects considered to be exempt from the policy.
- The `biba/high` label will permit writing to objects set at a lower label, but not permit reading that object. It is recommended that this label be placed on objects that affect the integrity of the entire system.

Biba provides for:

- Hierarchical integrity level with a set of non hierarchical integrity categories;
- Fixed rules: no write up, no read down (opposite of MLS). A subject can have write access to objects on its own level or below, but not above. Similarly, a subject can have read access to objects on its own level or above, but not below;
- Integrity (preventing inappropriate modification of data);
- Integrity levels (instead of MLS sensitivity levels).

The following `sysctl` tunables can be used to manipulate the Biba policy.

- `security.mac.biba.enabled` may be used to enable/disable enforcement of the Biba policy on the target machine.

- `security.mac.biba.ptys_equal` may be used to disable the Biba policy on pty(4) devices.
- `security.mac.biba.revocation_enabled` will force the revocation of access to objects if the label is changed to dominate the subject.

To access the Biba policy setting on system objects, use the `setfmac` and `getfmac` commands:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

Observations: a lower integrity subject is unable to write to a higher integrity subject; a higher integrity subject cannot observe or read a lower integrity object.

16.13 The MAC LOMAC Module

Module name: `mac_lomac.ko`

Kernel configuration line: `options MAC_LOMAC`

Boot option: `mac_lomac_load="YES"`

Unlike the MAC Biba policy, the mac_lomac(4) policy permits access to lower integrity objects only after decreasing the integrity level to not disrupt any integrity rules.

The MAC version of the Low-watermark integrity policy, not to be confused with the older lomac(4) implementation, works almost identically to Biba, but with the exception of using floating labels to support subject demotion via an auxiliary grade compartment. This secondary compartment takes the form of `[auxgrade]`. When assigning a lomac policy with an auxiliary grade, it should look a little bit like: `lomac/10[2]` where the number two (2) is the auxiliary grade.

The MAC LOMAC policy relies on the ubiquitous labeling of all system objects with integrity labels, permitting subjects to read from low integrity objects and then downgrading the label on the subject to prevent future writes to high integrity objects. This is the `[auxgrade]` option discussed above, thus the policy may provide for greater compatibility and require less initial configuration than Biba.

16.13.1 Examples

Like the Biba and MLS policies; the `setfmac` and `setpmac` utilities may be used to place labels on system objects:

```
# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]
```

Notice the auxiliary grade here is `low`, this is a feature provided only by the MAC LOMAC policy.

16.14 Implementing a Secure Environment with MAC

The following demonstration will implement a secure environment using various MAC modules with properly configured policies. This is only a test and should not be considered the complete answer to everyone’s security woes. Just implementing a policy and ignoring it never works and could be disastrous in a production environment.

Before beginning this process, the `multilabel` option must be set on each file system as stated at the beginning of this chapter. Not doing so will result in errors.

16.14.1 Create an insecure User Class

Begin the procedure by adding the following user class to the `/etc/login.conf` file:

```
insecure:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
:manpath=/usr/share/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=partition/13,mls/5:
```

And adding the following line to the default user class:

```
:label=mls/equal,biba/equal,partition/15:
```

Once this is completed, the following command must be issued to rebuild the database:

```
# cap_mkdb /etc/login.conf
```

16.14.2 Boot with the Correct Modules

Add the following lines to `/boot/loader.conf` so the required modules will load during system initialization:

```
mac_biba_load="YES"
mac_mls_load="YES"
mac_seeotheruids_load="YES"
mac_partition_load="YES"
```

16.14.3 Set All Users to Insecure

All user accounts that are not `root` or system users will now require a login class. The login class is required otherwise users will be refused access to common commands such as vi(1). The following `sh` script should do the trick:

```
# for x in ‘awk -F: ’($3 >= 1001) && ($3 != 65534) { print $1 }’ \
 /etc/passwd‘; do pw usermod $x -L insecure; done;
```

The `cap_mkdb` command will need to be run on `/etc/master.passwd` after this change.

16.14.4 Complete the Configuration

A contexts file should now be created; the following example was taken from Robert Watson’s example policy and should be placed in `/etc/policy.contexts`.

```
# This is the default BIBA/MLS policy for this system.

.*                              biba/high,mls/high
/sbin/dhclient                  biba/high(low),mls/high(low)
/dev(/.*)?                      biba/equal,mls/equal
# This is not an exhaustive list of all "privileged" devices.
/dev/mdctl                      biba/high,mls/high
/dev/pci                        biba/high,mls/high
/dev/k?mem                      biba/high,mls/high
/dev/io                         biba/high,mls/high
/dev/agp.*                      biba/high,mls/high
(/var)?/tmp(/.*)?               biba/equal,mls/equal
/tmp/\.X11-unix                 biba/high(equal),mls/high(equal)
/tmp/\.X11-unix/.*              biba/equal,mls/equal
/proc(/.*)?                     biba/equal,mls/equal
/mnt.*                          biba/low,mls/low
(/usr)?/home                    biba/high(low),mls/high(low)
(/usr)?/home/.*                 biba/low,mls/low
/var/mail(/.*)?                 biba/low,mls/low
/var/spool/mqueue(/.*)?         biba/low,mls/low
(/mnt)?/cdrom(/.*)?             biba/high,mls/high
(/usr)?/home/(ftp|samba)(/.*)?  biba/high,mls/high
/var/log/sendmail\.st           biba/low,mls/low
/var/run/utmp                   biba/equal,mls/equal
/var/log/(lastlog|wtmp)         biba/equal,mls/equal
```

This policy will enforce security by setting restrictions on both the downward and upward flow of information with regards to the directories and utilities listed on the left.

This can now be read into our system by issuing the following command:

```
# setfsmac -ef /etc/policy.contexts /
# setfsmac -ef /etc/policy.contexts /usr
```

Note: The above file system layout may be different depending on environment.

The /etc/mac.conf file requires the following modifications in the main section:

```
default_labels file ?biba,?mls
default_labels ifnet ?biba,?mls
default_labels process ?biba,?mls,?partition
default_labels socket ?biba,?mls
```

16.14.5 Testing the Configuration

Add a user with the adduser command and place that user in the insecure class for these tests.

The examples below will show a mix of root and regular user tests; use the prompt to distinguish between the two.

16.14.5.1 Basic Labeling Tests

```
% getpmac
biba/15(15-15),mls/15(15-15),partition/15
# setpmac partition/15,mls/equal top
```

Note: The top process will be killed before we start another top process.

16.14.5.2 MAC Seeotheruids Tests

```
% ps Zax
biba/15(15-15),mls/15(15-15),partition/15  1096 #C:  S      0:00.03 -su (bash)
biba/15(15-15),mls/15(15-15),partition/15  1101 #C:  R+     0:00.01 ps Zax
```

We should not be permitted to see any processes owned by other users.

16.14.5.3 MAC Partition Test

Disable the MAC seeotheruids policy for the rest of these tests:

```
# sysctl security.mac.seeotheruids.enabled=0
% ps Zax
LABEL                                                PID  TT  STAT      TIME COMMAND
  biba/equal(low-high),mls/equal(low-high),partition/15  1122 #C:  S+     0:00.02 top
  biba/15(15-15),mls/15(15-15),partition/15              1096 #C:  S      0:00.05 -su (bash)
  biba/15(15-15),mls/15(15-15),partition/15              1123 #C:  R+     0:00.01 ps Zax
```

All users should be permitted to see every process in their partition.

16.14.5.4 Testing Biba and MLS Labels

```
# setpmac partition/15,mls/equal,biba/high\(high-high\) top
% ps Zax
LABEL                                                  PID  TT  STAT    TIME   COMMAND
  biba/high(high-high),mls/equal(low-high),partition/15   1251 #C:  S+     0:00.02 top
  biba/15(15-15),mls/15(15-15),partition/15               1096 #C:  S      0:00.06 -su (bash)
  biba/15(15-15),mls/15(15-15),partition/15               1157 #C:  R+     0:00.00 ps Zax
```

The Biba policy allows us to read higher-labeled objects.

```
# setpmac partition/15,mls/equal,biba/low top
% ps Zax
LABEL                                          PID  TT  STAT      TIME COMMAND
  biba/15(15-15),mls/15(15-15),partition/15  1096 #C:  S      0:00.07 -su (bash)
  biba/15(15-15),mls/15(15-15),partition/15  1226 #C:  R+     0:00.01 ps Zax
```

The Biba policy does not allow lower-labeled objects to be read; however, MLS does.

```
% ifconfig bge0 | grep maclabel
maclabel biba/low(low-low),mls/low(low-low)
% ping -c 1 192.0.34.166
PING 192.0.34.166 (192.0.34.166): 56 data bytes
ping: sendto: Permission denied
```

Users are unable to ping `example.com`, or any domain for that matter.

To prevent this error from occurring, run the following command:

```
# sysctl security.mac.biba.trust_all_interfaces=1
```

This sets the default interface label to insecure mode, so the default Biba policy label will not be enforced.

```
# ifconfig bge0 maclabel biba/equal\(low-high\),mls/equal\(low-high\)
% ping -c 1 192.0.34.166
PING 192.0.34.166 (192.0.34.166): 56 data bytes
64 bytes from 192.0.34.166: icmp_seq=0 ttl=50 time=204.455 ms
--- 192.0.34.166 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 204.455/204.455/204.455/0.000 ms
```

By setting a more correct label, we can issue `ping` requests.

Now to create a few files for some read and write testing procedures:

```
# touch test1 test2 test3 test4 test5
# getfmac test1
test1: biba/equal,mls/equal
# setfmac biba/low test1 test2; setfmac biba/high test4 test5; \
  setfmac mls/low test1 test3; setfmac mls/high test2 test4
# setfmac mls/equal,biba/equal test3 && getfmac test?
test1: biba/low,mls/low
test2: biba/low,mls/high
test3: biba/equal,mls/equal
test4: biba/high,mls/high
```

```
test5: biba/high,mls/equal
# chown testuser:testuser test?
```

All of these files should now be owned by our `testuser` user. And now for some read tests:

```
% ls
test1   test2   test3   test4   test5
% ls test?
ls: test1: Permission denied
ls: test2: Permission denied
ls: test4: Permission denied
test3   test5
```

We should not be permitted to observe pairs; e.g.: `(biba/low,mls/low)`, `(biba/low,mls/high)` and `(biba/high,mls/high)`. And of course, read access should be denied. Now for some write tests:

```
% for i in ‘echo test*‘; do echo 1 > $i; done
-su: test1: Permission denied
-su: test4: Permission denied
-su: test5: Permission denied
```

Like with the read tests, write access should not be permitted to write pairs; e.g.: `(biba/low,mls/high)` and `(biba/equal,mls/equal)`.

```
% cat test?
cat: test1: Permission denied
cat: test2: Permission denied
1
cat: test4: Permission denied
```

And now as `root`:

```
# cat test2
1
```

16.15 Another Example: Using MAC to Constrain a Web Server

A separate location for the web data which users must be capable of accessing will be appointed. This will permit `biba/high` processes access rights to the web data.

Begin by creating a directory to store the web data in:

```
# mkdir /usr/home/cvs
```

Now initialize it with `cvs`:

```
# cvs -d /usr/home/cvs init
```

The first goal is to enable the `biba` policy, thus the `mac_biba_enable="YES"` should be placed in `/boot/loader.conf`. This assumes that support for MAC has been enabled in the kernel.

From this point on everything in the system should be set at `biba/high` by default.

The following modification must be made to the `login.conf` file, under the default user class:

```
:ignoretime@:\
 :umask=022:\
 :label=biba/high:
```

Every user should now be placed in the default class; a command such as:

```
# for x in ‘awk -F: ’($3 >= 1001) && ($3 != 65534) { print $1 }’ \
 /etc/passwd‘; do pw usermod $x -L default; done;
```

will accomplish this task in a few moments.

Now create another class, web, a copy of default, with the label setting of `biba/low`.

Create a user who will be used to work with the main web data stored in a **cvs** repository. This user must be placed in our new login class, `web`.

Since the default is `biba/high` everywhere, the repository will be the same. The web data must also be the same for users to have read/write access to it; however, since our web server will be serving data that `biba/high` users must access, we will need to downgrade the data as a whole.

The perfect tools for this are sh(1) and cron(8) and are already provided in FreeBSD. The following script should do everything we want:

```
PATH=/bin:/usr/bin:/usr/local/bin; export PATH;
CVSROOT=/home/repo; export CVSROOT;
cd /home/web;
cvs -qR checkout -P htdocs;
exit;
```

> **Note:** In many cases the `cvs` Id tags must be placed into the web site data files.

This script may now be placed into `web`’s home directory and the following crontab(1) entry added:

```
# Check out the web data as biba/low every twelve hours:
0       */12       *       *       *       web    /home/web/checkout.sh
```

This will check out the HTML sources every twelve hours on the machine.

The default startup method for the web server must also be modified to start the process as `biba/low`. This can be done by making the following modification to the `/usr/local/etc/rc.d/apache.sh` script:

```
command="setpmac biba/low /usr/local/sbin/httpd"
```

The **Apache** configuration must be altered to work with the `biba/low` policy. In this case the software must be configured to append to the log files in a directory set at `biba/low` or else “`access denied`” errors will be returned.

> **Note:** Following this example requires that the `docroot` directive be set to `/home/web/htdocs`; otherwise, **Apache** will fail when trying to locate the directory to serve documents from.

Other configuration variables must be altered as well, including the PID file, `Scoreboardfile`, `DocumentRoot`, log file locations, or any other variable which requires write access. When using `biba`, all write access will be denied to the server in areas *not* set at `biba/low`.

16.16 Troubleshooting the MAC Framework

During the development stage, a few users reported problems with normal configuration. Some of these problems are listed below:

16.16.1 The `multilabel` option cannot be enabled on `/`

The `multilabel` flag does not stay enabled on my root (`/`) partition!

It seems that one out of every fifty users has this problem, indeed, we had this problem during our initial configuration. Further observation of this so called "bug" has lead me to believe that it is a result of either incorrect documentation or misinterpretation of the documentation. Regardless of why it happened, the following steps may be taken to resolve it:

1. Edit `/etc/fstab` and set the root partition at `ro` for read-only.
2. Reboot into single user mode.
3. Run `tunefs -l enable` on `/`.
4. Reboot the system into normal mode.
5. Run `mount -urw /` and change the `ro` back to `rw` in `/etc/fstab` and reboot the system again.
6. Double-check the output from the `mount` to ensure that `multilabel` has been properly set on the root file system.

16.16.2 Cannot start a X11 server after MAC

After establishing a secure environment with MAC, I am no longer able to start X!

This could be caused by the MAC `partition` policy or by a mislabeling in one of the MAC labeling policies. To debug, try the following:

1. Check the error message; if the user is in the `insecure` class, the `partition` policy may be the culprit. Try setting the user's class back to the `default` class and rebuild the database with the `cap_mkdb` command. If this does not alleviate the problem, go to step two.
2. Double-check the label policies. Ensure that the policies are set correctly for the user in question, the X11 application, and the `/dev` entries.
3. If neither of these resolve the problem, send the error message and a description of your environment to the TrustedBSD discussion lists located at the TrustedBSD (http://www.TrustedBSD.org) website or to the FreeBSD general questions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) mailing list.

16.16.3 Error: _secure_path(3) cannot stat .login_conf

When I attempt to switch from the root to another user in the system, the error message "_secure_path: unable to state .login_conf".

This message is usually shown when the user has a higher label setting then that of the user whom they are attempting to become. For instance a user on the system, joe, has a default label of biba/low. The root user, who has a label of biba/high, cannot view joe's home directory. This will happen regardless if root has used the su command to become joe, or not. In this scenario, the Biba integrity model will not permit root to view objects set at a lower integrity level.

16.16.4 The root username is broken!

In normal or even single user mode, the root is not recognized. The whoami command returns 0 (zero) and su returns "who are you?". What could be going on?

This can happen if a labeling policy has been disabled, either by a sysctl(8) or the policy module was unloaded. If the policy is being disabled or has been temporarily disabled, then the login capabilities database needs to be reconfigured with the label option being removed. Double check the login.conf file to ensure that all label options have been removed and rebuild the database with the cap_mkdb command.

Notes

1. Other conditions may produce different failures. For instance, the file may not be owned by the user attempting to relabel the object, the object may not exist or may be read only. A mandatory policy will not allow the process to relabel the file, maybe because of a property of the file, a property of the process, or a property of the proposed new label value. For example: a user running at low integrity tries to change the label of a high integrity file. Or perhaps a user running at low integrity tries to change the label of a low integrity file to a high integrity label.
2. Due to a bug the security.mac.portacl.enabled sysctl variable will not work on FreeBSD 5.2.1 or previous releases.

Chapter 17 Security Event Auditing

Written by Tom Rhodes.

17.1 Synopsis

The FreeBSD 7-CURRENT development branch includes support for Event Auditing based on the POSIX.1e draft and Sun's published BSM API and file format. Event auditing permits the selective logging of security-relevant system events for the purposes of post-mortem analysis, system monitoring, and intrusion detection. After some settling time in FreeBSD 7-CURRENT, this support will be merged to FreeBSD 6-STABLE and appear in subsequent releases.

> **Warning:** The audit facility in FreeBSD is considered experimental, and production deployment should occur only after careful consideration of the risks of deploying experimental software.

This chapter will focus mainly on the installation and configuration of Event Auditing. Explanation of audit policies, and an example configuration will be provided for the convenience of the reader.

After reading this chapter, you will know:

- What Event Auditing is and how it works.
- How to configure Event Auditing on FreeBSD for users and processes.

Before reading this chapter, you should:

- Understand UNIX and FreeBSD basics (Chapter 3).
- Be familiar with the basics of kernel configuration/compilation (Chapter 8).
- Have some familiarity with security and how it pertains to FreeBSD (Chapter 14).

> **Warning:** Event auditing can generate a great deal of log file data, exceeding gigabytes a week in some configurations. An administrator should read this chapter in its entirety to avoid possible self-inflicted DoS attacks due to improper configuration.

The implementation of Event Auditing in FreeBSD is similar to that of the Sun™ Basic Security Module, or BSM library. Thus, the configuration is almost completely interchangeable with Solaris and Mac OS X/Darwin operating systems.

17.2 Key Terms - Words to Know

Before reading this chapter, a few key terms must be explained. This is intended to clear up any confusion that may occur and to avoid the abrupt introduction of new terms and information.

- *event*: An auditable event is an event that can be logged using the audit subsystem. The administrator can configure which events will be audited. Examples of security-relevant events include the creation of a file, the building of a network connection, or the logging in of a user. Events are either "attributable", meaning that they can be traced back to a user authentication, or "non-attributable". Examples of non-attributable events are any events that occur before authentication has succeeded in the login process, such as failed authentication attempts.
- *class*: Events may be assigned to one or more classes, usually based on the general category of the events, such as "file creation", "file access", or "network". Login and logout events are assigned to the `lo` class. The use of classes allows the administrator to specify high level auditing rules without having to specify whether each individual auditable operation will be logged.
- *record*: A record is a log entry describing a security event. Records typically have a record event type, information on the subject (user) associated with the event, time information, information on any objects, such as files, and information on whether the event corresponded to a successful operation.
- *trail*: An audit trail, or log file, consists of a series of audit records describing security events. Typically, trails are in roughly chronological order with respect to the time events completed. Only authorized processes are allowed to commit records to the audit trail.
- *prefix*: A prefix is considered to be the configuration element used to toggle auditing for success and failed events.

17.3 Installing Audit Support

Support for Event Auditing is installed with the normal `installworld` process. An administrator may confirm this by viewing the contents of `/etc/security`. Files beginning with the word *audit* should be present. For example, `audit_event`.

In-kernel support for the framework must also exist. This may be done by adding the following lines to the local kernel configuration file:

```
options AUDIT
```

Rebuild and reinstall the kernel via the normal process explained in Chapter 8.

Once completed, enable the audit daemon by adding the following line to rc.conf(5):

```
auditd_enable="YES"
```

Functionality not provided by the default may be added here with the `auditd_flags` option.

17.4 Audit Configuration

All configuration files for security audit are found in `/etc/security`. The following files must be present before the audit daemon is started:

- `audit_class` - Contains the definitions of the audit classes.
- `audit_control` - Controls aspects of the audit subsystem, such as default audit classes, minimum disk space to leave on the audit log volume, etc.
- `audit_event` - Defines the kernel audit events. These map, mostly, to system calls.

- `audit_user` - The events to audit for individual users. Users not appearing here will be subject to the default configuration in the control configuration file.
- `audit_warn` - A shell script used by auditd to generate warning messages in exceptional situations, such as when space for audit records is running low.

17.4.1 Audit File Syntax

The configuration file syntax is rather arcane, albeit easy to work with. One thing an administrator must be leery about is overriding system defaults. This could create potential openings for audit data to not be collected properly.

The audit subsystem will accept both the short name and long name with regards to configuration syntax. A syntax map has been included below.

The following list contains all supported audit classes:

- `all` - `all` - All audit flags set.
- `ad` - `administrative` - Administrative actions performed on the system as a whole.
- `ap` - `application` - Application defined action.
- `cl` - `file_close` - Audit calls to the `close` system call.
- `ex` - `exec` - Audit program or utility execution.
- `fa` - `file_attr_acc` - Audit the access of object attributes such as stat(1), pathconf(2) and similar events.
- `fc` - `file_creation` - Audit events where a file is created as a result.
- `fd` - `file_deletion` - Audit events where file deletion occurs.
- `fm` - `file_attr_mod` - Audit events where file attribute modification occurs, such as chown(8), chflags(1), flock(2), etc.
- `fr` - `file_read` - Audit events in which data is read, files are opened for reading, etc.
- `fw` - `file_write` - Audit events in which data is written, files are written or modified, etc.
- `io` - `ioctl` - Audit use of the ioctl(2) system call.
- `ip` - `ipc` - Audit various forms of Inter-Process Communication, including POSIX pipes and System V IPC operations.
- `lo` - `login_logout` - Audit login(1) and logout(1) events occurring on the system.
- `na` - `non_attrib` - Audit non-attributable events.
- `no` - `no_class` - Null class used to disable event auditing.
- `nt` - `network` - Audit events related to network actions, such as connect(2) and accept(2).
- `ot` - `other` - Audit miscellaneous events.
- `pc` - `process` - Audit process operations, such as exec(3) and exit(3).

Following is a list of all supported audit prefixes:

- `none` - Audit both the success or failure of an event. For example, just listing a class will result in the auditing of both success and failure.

- `+` - Audit successful events only.
- `-` - Audit failed events only.

> **Warning:** Using the `all` class with either the positive or negative prefix can generate a large amount of data at an extremely rapid rate.

Extra prefixes used to modify the default configuration values:

- `^-` - Disable auditing of failed events.
- `^+` - Enable auditing of successful events.
- `^` - Disable auditing of both successful and failed events.

17.4.2 Configuration Files

In most cases, administrators will need to modify only two files when configuring the audit system: `audit_control` and `audit_user`. The first controls system-wide audit paramaters and defaults for both attributable and non-attributable events. The second may be used to tune the level and nature of auditing for individual users.

17.4.2.1 The `audit_control` File

The `audit_control` file contains some basic defaults that the administrator may wish to modify. Perhaps even set some new ones. Viewing the contents of this file, we see the following:

```
dir:/var/audit
flags:lo
minfree:20
naflags:lo
```

The `dir` option is used to set the default directory where audit logs are stored. Audit is frequently configured so that audit logs are stored on a dedicated file system, so as to prevent interference between the audit subsystem and other subsystems when file systems become full.

The `flags` option is used to set the system-wide defaults. The current setting, `lo` configures the auditing of all login(1) and logout(1) actions. A more complex example, `lo,ad,-all,^-fa,^-fc,^-cl` audits all system login(1) and logout(1) actions, all administrator actions, all failed events in the system, and finally disables auditing of failed attempts for `fa`, `fc`, and `cl`. Even though the `-all` turned on the auditing of all failed attempts, the `^-` prefix will override that for the latter options.

Notice that the previous paragraph shows the file is read from left to right. As such, values further on the right side may override a previous value specified to its left.

The `minfree` option defines the minimum percentage of free space for audit file systems. This relates to the file system where audit logs are stored. For example, if the `dir` specifies `/var/audit` and `minfree` is set to twenty (20), warning messages will be generated when the `/var` file system grows to eighty (80) percent full.

The `naflags` option specifies audit classes to be audited for non-attributed events ——that is, events for which there is no authenticated user.

17.4.2.2 The `audit_user` File

The `audit_user` file permits the administrator to determine which classes of audit events should be logged for which system users.

The following is the defaults currently placed in the `audit_user` file:

```
root:lo:no
audit:fc:no
```

Notice how the default is to audit all cases of `login`/`logout` and disable auditing of all other actions for `root`. This configuration also audits all file creation and disables all other auditing for the `audit` user. While event auditing does not require a special user exist, some configurations, specifically environments making use of MAC, may require it.

17.5 Event Audit Administration

Events written by the kernel audit subsystem cannot be altered or read in plain text. Data is stored and accessed in a method similar to that of ktrace(1) and kdump(1), that is, they may only be viewed by dumping them using the `praudit` command; audit trails may be reduced using the `auditreduce` command, which selects records from an audit trail based on properties of interest, such as the user, time of the event, and type of operation.

For example, the `praudit` utility will dump the entire contents of a specified audit log in plain text. To dump an audit log in its entirety, use:

```
# praudit /var/audit/AUDITFILE
```

Where *`AUDITFILE`* is the audit log of viewing choice. Since audit logs may contain enormous amounts of data, an administrator may prefer to select records for specific users. This is made possible with the following command, where `trhodes` is the user of choice:

```
# auditreduce -e trhodes /var/audit/AUDITFILE | praudit
```

This will select all audit records produced by the user `trhodes` stored in the *`AUDITFILE`* file.

There are several other options available for reading audit records, see the aforementioned command's manual pages for a more in depth explanation.

17.5.1 Rotating Audit Log Files

Due to log reliability requirements, audit trails are written to only by the kernel, and managed only by `auditd`. Administrators should not attempt to use newsyslog.conf(5) or other tools to directly rotate audit logs. Instead, the `audit` management tool should be used to shut down auditing, reconfigure the audit system, and perform log rotation. The following command causes the audit daemon to create a new audit log and signal the kernel to switch to using the new log. The old log will be terminated and renamed, at which point it may then be manipulated by the administrator.

```
# audit -n
```

Warning: If the `auditd` daemon is not currently running, the previous command will fail and an error message will be produced.

Adding the following line to `/etc/crontab` will force the rotation every twelve hours from cron(8):

```
*     */12       *       *       *       root    /usr/sbin/audit -n
```

The change will take effect once you have saved the new `/etc/crontab`.

17.5.2 Delegating Audit Review Rights

By default, only the root user has the right to read system audit logs. However, that right may be delegated to members of the `audit` group, as the audit directory and audit trail files are assigned to that group, and made group-readable. As the ability to track audit log contents provides significant insight into the behavior of users and processes, it is recommended that the delegation of audit review rights be performed with caution.

Chapter 18 儲存設備篇

18.1 概述

本章涵蓋如何在FreeBSD 下使用碟片裝置[1] 包含memory-backed disk (用記憶體作為磁碟使用)、跨網路使用的磁碟、標準SCSI/IDE 磁碟、USB 介面的設備等。

閱讀本章後，您裝學會：

- FreeBSD 如何描述資料在磁碟上的劃分情形(partition 和slices)。
- 如何在系統上加入磁碟
- 如何設定FreeBSD 來使用USB 裝置。
- 如何設定虛擬檔案系統(virtual file systems), 例如memory disks (用記憶體作為磁碟使用)。
- 如何用quota 來限制磁碟空間的使用。
- 如何對磁碟加密以應付攻擊。
- 如何在FreeBSD 下建立、燒錄CD 和DVD。
- 各種不同的備份設備。
- 如何使用FreeBSD 提供的備份工具。
- 如何備份到軟碟。
- 什麼是snapshots ，且如何有效率地使用之。

在閱讀之前，您應該：

- 知道如何設定、安裝新的FreeBSD kernel。(Chapter 8).

18.2 裝置名稱

下面是FreeBSD 支援的儲存媒體列表，及它們對應的裝置名稱。

Table 18-1. 命名規則

裝置類型	裝置名稱
IDE 磁碟機	`ad`
IDE 光碟機	`acd`
SCSI 磁碟機和USB 碟	`da`
SCSI 光碟機	`cd`
非標準規格光碟機	Mitsumi 光碟機用`mcd`，Sony 光碟機用`scd`。
軟碟機	`fd`
SCSI 碟帶機	`sa`
IDE 碟帶機	`ast`
Flash 磁碟機	DiskOnChip® Flash 磁碟機用`fla`

裝置類型	裝置名稱
RAID 磁碟機	Adaptec AdvancedRAID 用`aacd`，Mylex 用`mlxd` 和`mlyd`，AMI MegaRAID 用`amrd`，Compaq Smart RAID 用`idad`，3ware® RAID 用`twed`。

18.3 新增磁碟

Originally contributed by David O'Brien.

假設我們想新增SCSI 磁碟到一臺原先只有一顆磁碟的機器上，首先將電腦關機，依製造商的指示將磁碟裝上去，詳細的操作方式請參考製造商的說明文件。

安裝好磁碟後，用`root` 登入系統，看一下`/var/run/dmesg.boot` 以確認系統是否抓到新磁碟。繼續剛才的範例，新增的磁碟會是`da1`，假設我們想將它掛載到`/1` 這個位置(如果您新增的是IDE 磁碟的話，請用`ad1`)。

FreeBSD 為了在IBM-PC 相容電腦上執行，必須配合PC BIOS partition，因此和傳統的BSD partition 有很大的不同。在PC 裡磁碟最多可以有四筆BIOS partition 資訊(亦即最多可分割成四個partition)。如果這個磁碟打算全部讓FreeBSD 使用，可選擇*dedicated* 模式，不然的話FreeBSD 必須置身於其中一個PC BIOS partition 中。在FreeBSD 裡，PC BIOS partition 稱為*slice*，這是為了不要和傳統的BSD partition 搞混了。[2] 不論是完全由FreeBSD 使用的磁碟，還是安裝了其它作業系統的磁碟，您都可以使用slice。這樣的好處是，其它非FreeBSD 作業系統的`fdisk` 工具可以順利操作。

如果使用slice，這個新增的磁碟會是`/dev/da1s1e`。可以這樣來解讀它：SCSI 磁碟、unit number 1(第二個SCSI 磁碟)、slice 1(第一個PC BIOS partition)、及`e` BSD partition。在dedicated 模式的話，新磁碟則是`/dev/da1e`。

因為bsdlabel(8) 是用32-bit 整數來儲存sector(磁區) 數，因此限制一個磁碟最大只能有2^32-1 個sector，亦即2TB 的空間。而fdisk(8) 的格式容許起始sector 編號不超過2^32-1，長度也不超過2^32-1，因此partition 最大空間是2TB，而磁碟最大是4TB。sunlabel(8) 則限制partition 最大是2TB，磁碟最多可有8 個partition，因此最大是16TB。如果要使用更大的磁碟，請使用gpt(8)。

18.3.1 使用sysinstall(8)

1. 操作**Sysinstall**

   透過`sysinstall` 的選單介面，可以輕易為磁碟分割BIOS partition(slice) 和BSD patition。必須以root 身份使用`sysinstall`，要嘛用root 登入，要嘛用`su` 切換到root。執行`sysinstall` 後，選`Configure`，在`FreeBSD Configuration Menu` 裡移到`Fdisk` 選項。

2. **fdisk** Partition 編輯器

   在**fdisk** 裡，按下**`A`** 表示整個磁碟都給FreeBSD 使用。接著會提示您『是否要相容其它的作業系統』，回答`YES`。按**`W`** 會將這些改變立即寫入磁碟，再按**`q`** 可以離開FDISK 編輯器。接下來會問您要將“Master Boot Record” 安裝於何處，由於現在是新增磁碟，表示作業系統已經裝在別的磁碟上了，所以可以選`None` 就行了。

3. Disk Label Editor(磁碟Label 編輯器)

   接著請關閉**sysinstall**，再重開一次。照著上一節的指示，不過這次改選`Label` 進入`Disk Label Editor`，在此您可以編輯傳統的BSD partition。一個磁碟(或著一個slice) 最多可切分成8 個BSD

partition，依序用a-h 來表示。有些字母有特別的意義，a partition 表示這是root partition(根分割區，/)，因此只有安裝系統的磁碟(例如用來開機的磁碟) 有a partition。b partition 表示這是swap partitions(交換分割區)，每個磁碟上都可以有swap。c partition 用來表示整個磁碟(如果使用dedicated mode 的話)或整個slice。其它的字母則用來表示普通的BSD partition 。

sysinstall 的Label editor(磁碟Label 編輯器) 偏好用e 來表示非root、也非swap 的分割區[3] 在Label editor 裡，按**c** 可以新增一個檔案系統(BSD label)，它會問您這是一個FS(file system ，檔案系統) 或是swap(交換分割區)，選擇FS 接著輸入要掛載的位置(例如/mnt)。如果系統安裝完後才新增磁碟，**sysinstall** 不會幫您把這筆掛載資料加入/etc/fstab，所以掛載的位置不太重要。

當您準備好將新的label 寫入磁碟、建立檔案系統，按**w** 即可。如果出現在什麼錯誤，**sysinstall** 可能無法幫您掛載這個新分割區。結束Label Editor、結束**sysinstall** 就行了。

4. 完成

   最後要做的是編輯/etc/fstab，加入您新增的分割區資訊。

18.3.2 使用命令列工具

18.3.2.1 使用Slices(BIOS partitions)

這種模式能讓您的磁碟分割區與其它作業系統的fdisk 工具和平共處，因此我們建議您使用slice 模式。如果您一定要使用dedicated 模式，您得有個好理由！[4]

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# fdisk -BI da1 # 初始您的磁碟。
# bsdlabel -B -w -r da1s1 auto # 建立 bsdlabel。
# bsdlabel -e da1s1 # 編輯 bsdlabel 以新增 label。
# mkdir -p /1
# newfs /dev/da1s1e # 如果您新增了多個 label，對每個 label 重覆這個步驟。
# mount /dev/da1s1e /1 # 掛載這些新 label。
# vi /etc/fstab # 在 /etc/fstab 加入適當的資訊。
```

如果您新增的是IDE 磁碟，將da 改成ad 即可[5]。

18.3.2.2 Dedicated

如果您不打算將新磁碟用於其它的作業系統，您可以使用dedicated 模式。注意：Microsoft 的作業系統認不得這個模式，不過也不會去破壞它；然而IBM 的OS/2 就沒那麼好心了，它會去調整所有它不認得的分割區[6]。

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# bsdlabel -Brw da1 auto
# bsdlabel -e da1     # 建立 'e' partition。
# newfs -d0 /dev/da1e
# mkdir -p /1
# vi /etc/fstab     # 新增一筆 /dev/da1e 的資訊。
# mount /1
```

另一種方法：

```
# dd if=/dev/zero of=/dev/da1 count=2
# bsdlabel /dev/da1 | bsdlabel -BrR da1 /dev/stdin
# newfs /dev/da1e
# mkdir -p /1
# vi /etc/fstab      # 新增一筆 /dev/da1e 的資訊。
# mount /1
```

18.4 RAID

18.4.1 軟體RAID

18.4.1.1 連接式磁碟裝置驅動程式(CCD, Concatenated Disk Driver) 設定

Original work by Christopher Shumway. Revised by Jim Brown.

對大容量儲存設備而言，最關鍵的要素乃是速度、可靠性及價格。然而這三者往往難以兼顧：快速可靠的設備通常很貴；而降低成本通常也犧牲了速度或可靠性。

接下來要介紹的系統，價格是最重要的考量，接下來是速度，最後才是可靠性。順序如此是因為資料傳輸的速度最終取決於網路，而儘管可靠性十分重要，卻有簡單的取代方案：將資料完整備份於CD-R 中。

選擇大容量儲存設備方案時，首先要定義您的需求。如果您重視速度或可靠性甚於價格，接下來的介紹恐非您所需。

18.4.1.1.1 安裝硬體

除了系統磁碟外，下面介紹的CCD 磁碟陣列將使用到三顆30GB、5400 RPM 的Western Digital IDE 磁碟，以提供約90GB 的儲存空間。最理想的情況是每個磁碟由獨立使用的排線連接獨立使用的IDE 控制器，不過為了降低成本，利用jumper 設定磁碟，使每個IDE 控制器可連接一個主磁碟加一個副磁碟，如此可不必加裝額外的IDE 控制器。

開機後，BIOS 應該設定成自重偵測磁碟。更重要的是FreeBSD 應該要偵測到它們：

```
ad0: 19574MB <WDC WD205BA> [39770/16/63] at ata0-master UDMA33
ad1: 29333MB <WDC WD307AA> [59598/16/63] at ata0-slave UDMA33
ad2: 29333MB <WDC WD307AA> [59598/16/63] at ata1-master UDMA33
ad3: 29333MB <WDC WD307AA> [59598/16/63] at ata1-slave UDMA33
```

> **Note:** 如果FreeBSD 沒有偵測到所有磁碟，請確認jumper 都設定正確。許多IDE 磁碟可以設定成“Cable Select” (根據排線位置決定)，這*並非* master(主磁碟) 或slave(副磁碟)。請參閱磁碟的說明文件以正確設定jumper 。

接下來，考慮如何將它們變成檔案系統的一部份。您可以參考vinum(8)(Chapter 20) 及ccd(4)。在此我們選擇ccd(4)。

18.4.1.1.2 設定CCD

ccd(4) 可以將多個磁碟接起來成為一個大磁碟。要使用ccd(4)，您的kernel 需要支援ccd(4)。將這行加入到kernel 設定檔，並重編、重安裝kernel：

```
device   ccd
```

也可以載入kernel 動態模組來支援ccd(4)。

使用ccd(4) 請先用bsdlabel(8) 來初始磁碟：

```
bsdlabel -r -w ad1 auto
bsdlabel -r -w ad2 auto
bsdlabel -r -w ad3 auto
```

上述指令會建立`ad1c`，`ad2c` 和`ad3c`，這些bsdlabel 都使用了整個磁碟。

下一步是修改label type，同樣用bsdlabel(8) 來處理：

```
bsdlabel -e ad1
bsdlabel -e ad2
bsdlabel -e ad3
```

這個指令會打開一個編輯器(預設是vi(1)，可以用`EDITOR` 環境變數來指定其它編輯器)，並將目前磁碟的label 資訊顯示在該編輯器裡。

一個還未變動過的磁碟label 資訊看起來會像這樣：

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    0 - 59597)
```

在此我們要新增一個e partition 給ccd(4) 使用。通常複製c partition 那一行，再把`fstype` 那一行改成**`4.2BSD`** 就可以了。改完之後看起來應該會像這樣：

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    0 - 59597)
  e: 60074784        0    4.2BSD        0     0     0   # (Cyl.    0 - 59597)
```

18.4.1.1.3 建立檔案系統

現在所有的磁碟都已經建好bsdlabel 了，可以開始建立ccd(4)。用ccdconfig(8) 來建立ccd(4)，參考下面的指令：

```
ccdconfig ccd0❶ 32❷ 0❸ /dev/ad1e❹ /dev/ad2e /dev/ad3e
```

每個參數的作用如下：

❶ 第一個參數是要設定的裝置名稱，在這個例子裡是`/dev/ccd0c`。其中`/dev/` 可有可無。

❷ 「interleave」的大小。所謂interleave 是指一排磁碟區塊(disk block)的大小，通常以512 bytes 為單位，所以interleave 設為32 即為16,384 bytes。

❸ ccdconfig(8) 設定模式的參數。如果您打算啓用磁碟鏡設(drive mirroring)，您可以在此指定參數。這個例子沒有使用鏡設，所以設成0。

❹ ccdconfig(8) 最後的參數是要加入到陣列的所有磁碟。請使用完整的路徑。

執行ccdconfig(8) 之後，ccd(4) 已設定完成可供建立檔案系統。請參考newfs(8) 或輸入：

```
newfs /dev/ccd0c
```

18.4.1.1.4 讓一切自動完成

通常您會希望每次開機時都能自動掛上(mount) ccd(4)。用下面的指令將您目前的設定寫入`/etc/ccd.conf`：

```
ccdconfig -g > /etc/ccd.conf
```

如果`/etc/ccd.conf` 存在，每次開機時`/etc/rc` 都會執行`ccdconfig -C` 。如此便可自動設定ccd(4) 以便之後掛上(mount)檔案系統。

> **Note:** 如果您開機時選擇進入單人模式(single mode)，在掛上(mount(8)) ccd(4) 的檔案系統之前您得先執行設定的指令：
>
> ```
> ccdconfig -C
> ```

要在每次開機時自動掛上(mount) ccd(4)，請在`/etc/fstab` 加入ccd(4)：

```
/dev/ccd0c              /media       ufs     rw      2       2
```

18.4.1.2 Vinum 容量管理系統

Vinum 容量管理系統(以下簡稱Vinum) 可視為一種虛擬磁碟。它將區塊裝置(block device) 的介面與對應資料的方式切割開來，比起原本slice 劃分的磁碟，Vinum 可增加了彈性、效能和穩定度[7] vinum(8) 實作了RAID-0、RAID-1 和RAID-5 等模組，它們都可以單獨使用，也可以互相搭配使用。

請見Chapter 20 以參考更多關於vinum(8) 的資訊。

18.4.2 硬體RAID

FreeBSD 也支援許多硬體RAID 控制器。這些控制器自行掌控一個小型的RAID 系統，因此不需要特定軟體來管理。

透過控制器上的BIOS 幾乎能控制所有的操作。接下來將簡單介紹如何設定Promise IDE RAID 控制卡。首先確認控制卡已安裝，接著開機。它應該會提示一些資訊[8]。依指示進入控制卡的設定畫面，從這裡您可以將全部的硬體結合成一個大磁碟。完成之後，FreeBSD 將只會看到這個大磁碟。當然您也可以使用其它的RAID 模式。

18.4.3 重建(rebuild) ATA RAID1 陣列

FreeBSD 允許您熱插拔磁碟陣列裡壞掉的磁碟，當然在重開機前就得先發現。

也許您會在/var/log/messages(或dmesg(8) 的輸出) 看到類似下面的訊息：

```
ad6 on monster1 suffered a hard error.
ad6: READ command timeout tag=0 serv=0 - resetting
ad6: trying fallback to PIO mode
ata3: resetting devices .. done
ad6: hard error reading fsbn 1116119 of 0-7 (ad6 bn 1116119; cn 1107 tn 4 sn 11)\\
status=59 error=40
ar0: WARNING - mirror lost
```

請用atacontrol(8) 來得到更多資訊：

```
# atacontrol list
ATA channel 0:
 Master:      no device present
 Slave:   acd0 <HL-DT-ST CD-ROM GCR-8520B/1.00> ATA/ATAPI rev 0

ATA channel 1:
 Master:      no device present
 Slave:       no device present

ATA channel 2:
 Master:  ad4 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
 Slave:       no device present

ATA channel 3:
 Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
 Slave:       no device present

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: DEGRADED
```

1. 首先您得將損壞磁碟所在的ata channel 卸載(detach)，如此才能安全地移除：

   ```
   # atacontrol detach ata3
   ```

2. 用好的磁碟換下損壞的。

3. 重新載入(re-attach) ata channel：

   ```
   # atacontrol attach ata3
   Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
   Slave:   no device present
   ```

4. 將新的磁碟加入原本的磁碟陣列成為備援(spare) 磁碟：

   ```
   # atacontrol addspare ar0 ad6
   ```

5. 重建磁碟陣列：

   ```
   # atacontrol rebuild ar0
   ```

6. 可以用下面指定來確認重建的進度：

```
# dmesg | tail -10
[output removed]
ad6: removed from configuration
ad6: deleted from ar0 disk1
ad6: inserted into ar0 disk1 as spare

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: REBUILDING 0% completed
```

7. 等重建完就完成了。

18.5 USB 儲存裝置

Contributed by Marc Fonvieille.

在現在，有許多外部儲存裝置採用USB(Universal Serial Bus) 介面，例如硬碟、USB 拇指碟、CD-R 燒錄機等。FreeBSD 提供對這些裝置的支援。

18.5.1 設定

USB mass 儲存裝置驅動程式(umass(4))提供USB 儲存裝置的支援。但如果是用`GENERIC` kernel，就不需要做任何設定變動。若是自訂kernel，請確認kernel 設定檔含有下面這幾行：

```
device scbus
device da
device pass
device uhci
device ohci
device usb
device umass
```

umass(4) 驅動程式透過SCSI 子系統存取USB 儲存裝置，您的USB 裝置會被系統辨識成SCSI 裝置。依照您主機板上USB 晶片型號，您只需要`device uhci` 或`device ohci` 其中一個。然而，將兩者都編進kernel 也無妨。只要別忘了在修改kernel 設定後重新編譯及安裝新的kernel 就行了。

> **Note:** 如果您的USB 裝置是CD-R 或DVD 燒錄機，則SCSI 光碟機驅動程式cd(4) 必須寫入kernel 設定檔，像這樣：
>
> ```
> device cd
> ```
>
> 因為燒錄機會被當成SCSI 裝置，所以atapicam(4) 驅動程式不需要編入kernel。

USB 2.0 控制器的支援由FreeBSD; 提供，然而必須在kernel 設定檔增加下面這行以提供USB 2.0 支援：

```
device ehci
```

注意，如果您需要USB 1.x 支援，您仍然需要將uhci(4) 及ohci(4) 驅動程式編入kernel。

18.5.2 測試設定

The configuration is ready to be tested: plug in your USB device, and in the system message buffer (dmesg(8)), the drive should appear as something like:

```
umass0: USB Solid state disk, rev 1.10/1.00, addr 2
GEOM: create disk da0 dp=0xc2d74850
da0 at umass-sim0 bus 0 target 0 lun 0
da0: <Generic Traveling Disk 1.11> Removable Direct Access SCSI-2 device
da0: 1.000MB/s transfers
da0: 126MB (258048 512 byte sectors: 64H 32S/T 126C)
```

Of course, the brand, the device node (`da0`) and other details can differ according to your configuration.

Since the USB device is seen as a SCSI one, the `camcontrol` command can be used to list the USB storage devices attached to the system:

```
# camcontrol devlist
<Generic Traveling Disk 1.11>      at scbus0 target 0 lun 0 (da0,pass0)
```

If the drive comes with a file system, you should be able to mount it. The Section 18.3 will help you to format and create partitions on the USB drive if needed.

If you unplug the device (the disk must be unmounted before), you should see, in the system message buffer, something like the following:

```
umass0: at uhub0 port 1 (addr 2) disconnected
(da0:umass-sim0:0:0:0): lost device
(da0:umass-sim0:0:0:0): removing device entry
GEOM: destroy disk da0 dp=0xc2d74850
umass0: detached
```

18.5.3 Further Reading

Beside the Adding Disks and Mounting and Unmounting File Systems sections, reading various manual pages may be also useful: umass(4), camcontrol(8), and usbdevs(8).

18.6 Creating and Using Optical Media (CDs)

Contributed by Mike Meyer.

18.6.1 Introduction

CDs have a number of features that differentiate them from conventional disks. Initially, they were not writable by the user. They are designed so that they can be read continuously without delays to move the head between tracks. They are also much easier to transport between systems than similarly sized media were at the time.

CDs do have tracks, but this refers to a section of data to be read continuously and not a physical property of the disk. To produce a CD on FreeBSD, you prepare the data files that are going to make up the tracks on the CD, then write the tracks to the CD.

The ISO 9660 file system was designed to deal with these differences. It unfortunately codifies file system limits that were common then. Fortunately, it provides an extension mechanism that allows properly written CDs to exceed those limits while still working with systems that do not support those extensions.

The `sysutils/cdrtools` port includes mkisofs(8), a program that you can use to produce a data file containing an ISO 9660 file system. It has options that support various extensions, and is described below.

Which tool to use to burn the CD depends on whether your CD burner is ATAPI or something else. ATAPI CD burners use the `burncd` program that is part of the base system. SCSI and USB CD burners should use `cdrecord` from the `sysutils/cdrtools` port.

`burncd` has a limited number of supported drives. To find out if a drive is supported, see the CD-R/RW supported drives (http://www.freebsd.dk/ata/) list.

> **Note:** If you run FreeBSD 5.X, FreeBSD 4.8-RELEASE version or higher, it will be possible to use `cdrecord` and other tools for SCSI drives on an ATAPI hardware with the ATAPI/CAM module.

If you want a CD burning software with a graphical user interface, you should have a look to **X-CD-Roast** or **K3b**. These tools are available as packages or from the `sysutils/xcdroast` and `sysutils/k3b` ports. **X-CD-Roast** and **K3b** require the ATAPI/CAM module with ATAPI hardware.

18.6.2 mkisofs

The mkisofs(8) program, which is part of the `sysutils/cdrtools` port, produces an ISO 9660 file system that is an image of a directory tree in the UNIX file system name space. The simplest usage is:

```
# mkisofs -o imagefile.iso /path/to/tree
```

This command will create an *`imagefile.iso`* containing an ISO 9660 file system that is a copy of the tree at */path/to/tree*. In the process, it will map the file names to names that fit the limitations of the standard ISO 9660 file system, and will exclude files that have names uncharacteristic of ISO file systems.

A number of options are available to overcome those restrictions. In particular, `-R` enables the Rock Ridge extensions common to UNIX systems, `-J` enables Joliet extensions used by Microsoft systems, and `-hfs` can be used to create HFS file systems used by Mac OS.

For CDs that are going to be used only on FreeBSD systems, `-U` can be used to disable all filename restrictions. When used with `-R`, it produces a file system image that is identical to the FreeBSD tree you started from, though it may violate the ISO 9660 standard in a number of ways.

The last option of general use is `-b`. This is used to specify the location of the boot image for use in producing an "El Torito" bootable CD. This option takes an argument which is the path to a boot image from the top of the tree being written to the CD. By default, mkisofs(8) creates an ISO image in the so-called "floppy disk emulation" mode, and thus expects the boot image to be exactly 1200, 1440 or 2880 KB in size. Some boot loaders, like the one used by the FreeBSD distribution disks, do not use emulation mode; in this case, the `-no-emul-boot` option should be used. So, if `/tmp/myboot` holds a bootable FreeBSD system with the boot image in `/tmp/myboot/boot/cdboot`, you could produce the image of an ISO 9660 file system in `/tmp/bootable.iso` like so:

```
# mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

Having done that, if you have md configured in your kernel, you can mount the file system with:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

At which point you can verify that /mnt and /tmp/myboot are identical.

There are many other options you can use with mkisofs(8) to fine-tune its behavior. In particular: modifications to an ISO 9660 layout and the creation of Joliet and HFS discs. See the mkisofs(8) manual page for details.

18.6.3 burncd

If you have an ATAPI CD burner, you can use the burncd command to burn an ISO image onto a CD. burncd is part of the base system, installed as /usr/sbin/burncd. Usage is very simple, as it has few options:

```
# burncd -f cddevice data imagefile.iso fixate
```

Will burn a copy of *imagefile.iso* on *cddevice*. The default device is /dev/acd0. See burncd(8) for options to set the write speed, eject the CD after burning, and write audio data.

18.6.4 cdrecord

If you do not have an ATAPI CD burner, you will have to use cdrecord to burn your CDs. cdrecord is not part of the base system; you must install it from either the port at sysutils/cdrtools or the appropriate package. Changes to the base system can cause binary versions of this program to fail, possibly resulting in a “coaster”. You should therefore either upgrade the port when you upgrade your system, or if you are tracking -STABLE, upgrade the port when a new version becomes available.

While cdrecord has many options, basic usage is even simpler than burncd. Burning an ISO 9660 image is done with:

```
# cdrecord dev=device imagefile.iso
```

The tricky part of using cdrecord is finding the dev to use. To find the proper setting, use the -scanbus flag of cdrecord, which might produce results like this:

```
# cdrecord -scanbus
Cdrecord-Clone 2.01 (i386-unknown-freebsd7.0) Copyright (C) 1995-2004 Jörg Schilling
Using libscg version 'schily-0.1'
scsibus0:
        0,0,0     0) 'SEAGATE ' 'ST39236LW       ' '0004' Disk
        0,1,0     1) 'SEAGATE ' 'ST39173W        ' '5958' Disk
        0,2,0     2) *
        0,3,0     3) 'iomega  ' 'jaz 1GB         ' 'J.86' Removable Disk
        0,4,0     4) 'NEC     ' 'CD-ROM DRIVE:466' '1.26' Removable CD-ROM
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) *
        1,1,0   101) *
```

```
1,2,0   102) *
1,3,0   103) *
1,4,0   104) *
1,5,0   105) 'YAMAHA  ' 'CRW4260         ' '1.0q' Removable CD-ROM
1,6,0   106) 'ARTEC   ' 'AM12S           ' '1.06' Scanner
1,7,0   107) *
```

This lists the appropriate `dev` value for the devices on the list. Locate your CD burner, and use the three numbers separated by commas as the value for `dev`. In this case, the CRW device is 1,5,0, so the appropriate input would be `dev=1,5,0`. There are easier ways to specify this value; see cdrecord(1) for details. That is also the place to look for information on writing audio tracks, controlling the speed, and other things.

18.6.5 Duplicating Audio CDs

You can duplicate an audio CD by extracting the audio data from the CD to a series of files, and then writing these files to a blank CD. The process is slightly different for ATAPI and SCSI drives.

SCSI Drives

1. Use `cdda2wav` to extract the audio.

   ```
   % cdda2wav -v255 -D2,0 -B -Owav
   ```

2. Use `cdrecord` to write the `.wav` files.

   ```
   % cdrecord -v dev=2,0 -dao -useinfo  *.wav
   ```

   Make sure that *2,0* is set appropriately, as described in Section 18.6.4.

ATAPI Drives

1. The ATAPI CD driver makes each track available as `/dev/acd`*d*`t`*nn*, where *d* is the drive number, and *nn* is the track number written with two decimal digits, prefixed with zero as needed. So the first track on the first disk is `/dev/acd0t01`, the second is `/dev/acd0t02`, the third is `/dev/acd0t03`, and so on.

   Make sure the appropriate files exist in `/dev`. If the entries are missing, force the system to retaste the media:

   ```
   # dd if=/dev/acd0 of=/dev/null count=1
   ```

2. Extract each track using dd(1). You must also use a specific block size when extracting the files.

   ```
   # dd if=/dev/acd0t01 of=track1.cdr bs=2352
   # dd if=/dev/acd0t02 of=track2.cdr bs=2352
   ...
   ```

3. Burn the extracted files to disk using `burncd`. You must specify that these are audio files, and that `burncd` should fixate the disk when finished.

   ```
   # burncd -f /dev/acd0 audio track1.cdr track2.cdr ... fixate
   ```

18.6.6 Duplicating Data CDs

You can copy a data CD to a image file that is functionally equivalent to the image file created with mkisofs(8), and you can use it to duplicate any data CD. The example given here assumes that your CDROM device is `acd0`. Substitute your correct CDROM device.

```
# dd if=/dev/acd0 of=file.iso bs=2048
```

Now that you have an image, you can burn it to CD as described above.

18.6.7 Using Data CDs

Now that you have created a standard data CDROM, you probably want to mount it and read the data on it. By default, mount(8) assumes that a file system is of type `ufs`. If you try something like:

```
# mount /dev/cd0 /mnt
```

you will get a complaint about "`Incorrect super block`", and no mount. The CDROM is not a `UFS` file system, so attempts to mount it as such will fail. You just need to tell mount(8) that the file system is of type `ISO9660`, and everything will work. You do this by specifying the `-t cd9660` option mount(8). For example, if you want to mount the CDROM device, `/dev/cd0`, under `/mnt`, you would execute:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Note that your device name (`/dev/cd0` in this example) could be different, depending on the interface your CDROM uses. Also, the `-t cd9660` option just executes mount_cd9660(8). The above example could be shortened to:

```
# mount_cd9660 /dev/cd0 /mnt
```

You can generally use data CDROMs from any vendor in this way. Disks with certain ISO 9660 extensions might behave oddly, however. For example, Joliet disks store all filenames in two-byte Unicode characters. The FreeBSD kernel does not speak Unicode (yet!), so non-English characters show up as question marks. (The FreeBSD CD9660 driver includes hooks to load an appropriate Unicode conversion table on the fly. Modules for some of the common encodings are available via the `sysutils/cd9660_unicode` port.)

Occasionally, you might get "`Device not configured`" when trying to mount a CDROM. This usually means that the CDROM drive thinks that there is no disk in the tray, or that the drive is not visible on the bus. It can take a couple of seconds for a CDROM drive to realize that it has been fed, so be patient.

Sometimes, a SCSI CDROM may be missed because it did not have enough time to answer the bus reset. If you have a SCSI CDROM please add the following option to your kernel configuration and rebuild your kernel.

```
options SCSI_DELAY=15000
```

This tells your SCSI bus to pause 15 seconds during boot, to give your CDROM drive every possible chance to answer the bus reset.

18.6.8 Burning Raw Data CDs

You can choose to burn a file directly to CD, without creating an ISO 9660 file system. Some people do this for backup purposes. This runs more quickly than burning a standard CD:

```
# burncd -f /dev/acd1 -s 12 data archive.tar.gz fixate
```

In order to retrieve the data burned to such a CD, you must read data from the raw device node:

```
# tar xzvf /dev/acd1
```

You cannot mount this disk as you would a normal CDROM. Such a CDROM cannot be read under any operating system except FreeBSD. If you want to be able to mount the CD, or share data with another operating system, you must use mkisofs(8) as described above.

18.6.9 Using the ATAPI/CAM Driver

Contributed by Marc Fonvieille.

This driver allows ATAPI devices (CD-ROM, CD-RW, DVD drives etc...) to be accessed through the SCSI subsystem, and so allows the use of applications like `sysutils/cdrdao` or cdrecord(1).

To use this driver, you will need to add the following line to your kernel configuration file:

```
device atapicam
```

You also need the following lines in your kernel configuration file:

```
device ata
device scbus
device cd
device pass
```

which should already be present.

Then rebuild, install your new kernel, and reboot your machine. During the boot process, your burner should show up, like so:

```
acd0: CD-RW <MATSHITA CD-RW/DVD-ROM UJDA740> at ata1-master PIO4
cd0 at ata1 bus 0 target 0 lun 0
cd0: <MATSHITA CDRW/DVD UJDA740 1.00> Removable CD-ROM SCSI-0 device
cd0: 16.000MB/s transfers
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

The drive could now be accessed via the `/dev/cd0` device name, for example to mount a CD-ROM on `/mnt`, just type the following:

```
# mount -t cd9660 /dev/cd0 /mnt
```

As `root`, you can run the following command to get the SCSI address of the burner:

```
# camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00>   at scbus1 target 0 lun 0 (pass0,cd0)
```

So `1,0,0` will be the SCSI address to use with cdrecord(1) and other SCSI application.

For more information about ATAPI/CAM and SCSI system, refer to the atapicam(4) and cam(4) manual pages.

18.7 Creating and Using Optical Media (DVDs)

Contributed by Marc Fonvieille. With inputs from Andy Polyakov.

18.7.1 Introduction

Compared to the CD, the DVD is the next generation of optical media storage technology. The DVD can hold more data than any CD and is nowadays the standard for video publishing.

Five physical recordable formats can be defined for what we will call a recordable DVD:

- DVD-R: This was the first DVD recordable format available. The DVD-R standard is defined by the DVD Forum (http://www.dvdforum.com/forum.shtml). This format is write once.
- DVD-RW: This is the rewriteable version of the DVD-R standard. A DVD-RW can be rewritten about 1000 times.
- DVD-RAM: This is also a rewriteable format supported by the DVD Forum. A DVD-RAM can be seen as a removable hard drive. However, this media is not compatible with most DVD-ROM drives and DVD-Video players; only a few DVD writers support the DVD-RAM format.
- DVD+RW: This is a rewriteable format defined by the DVD+RW Alliance (http://www.dvdrw.com/). A DVD+RW can be rewritten about 1000 times.
- DVD+R: This format is the write once variation of the DVD+RW format.

A single layer recordable DVD can hold up to 4,700,000,000 bytes which is actually 4.38 GB or 4485 MB (1 kilobyte is 1024 bytes).

> **Note:** A distinction must be made between the physical media and the application. For example, a DVD-Video is a specific file layout that can be written on any recordable DVD physical media: DVD-R, DVD+R, DVD-RW etc. Before choosing the type of media, you must be sure that both the burner and the DVD-Video player (a standalone player or a DVD-ROM drive on a computer) are compatible with the media under consideration.

18.7.2 Configuration

The program growisofs(1) will be used to perform DVD recording. This command is part of the **dvd+rw-tools** utilities (`sysutils/dvd+rw-tools`). The **dvd+rw-tools** support all DVD media types.

These tools use the SCSI subsystem to access to the devices, therefore the ATAPI/CAM support must be added to your kernel. If your burner uses the USB interface this addition is useless, and you should read the Section 18.5 for more details on USB devices configuration.

You also have to enable DMA access for ATAPI devices, this can be done in adding the following line to the `/boot/loader.conf` file:

```
hw.ata.atapi_dma="1"
```

Before attempting to use the **dvd+rw-tools** you should consult the dvd+rw-tools' hardware compatibility notes (http://fy.chalmers.se/~appro/linux/DVD+RW/hcn.html) for any information related to your DVD burner.

Note: If you want a graphical user interface, you should have a look to **K3b** (`sysutils/k3b`) which provides a user friendly interface to growisofs(1) and many others burning tools.

18.7.3 Burning Data DVDs

The growisofs(1) command is a frontend to mkisofs, it will invoke mkisofs(8) to create the file system layout and will perform the write on the DVD. This means you do not need to create an image of the data before the burning process.

To burn onto a DVD+R or a DVD-R the data from the `/path/to/data` directory, use the following command:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /path/to/data
```

The options `-J -R` are passed to mkisofs(8) for the file system creation (in this case: an ISO 9660 file system with Joliet and Rock Ridge extensions), consult the mkisofs(8) manual page for more details.

The option `-Z` is used for the initial session recording in any case: multiple sessions or not. The DVD device, */dev/cd0*, must be changed according to your configuration. The `-dvd-compat` parameter will close the disk, the recording will be unappendable. In return this should provide better media compatibility with DVD-ROM drives.

It is also possible to burn a pre-mastered image, for example to burn the image *imagefile.iso*, we will run:

```
# growisofs -dvd-compat -Z /dev/cd0=imagefile.iso
```

The write speed should be detected and automatically set according to the media and the drive being used. If you want to force the write speed, use the `-speed=` parameter. For more information, read the growisofs(1) manual page.

18.7.4 Burning a DVD-Video

A DVD-Video is a specific file layout based on ISO 9660 and the micro-UDF (M-UDF) specifications. The DVD-Video also presents a specific data structure hierarchy, it is the reason why you need a particular program such as `multimedia/dvdauthor` to author the DVD.

If you already have an image of the DVD-Video file system, just burn it in the same way as for any image, see the previous section for an example. If you have made the DVD authoring and the result is in, for example, the directory `/path/to/video`, the following command should be used to burn the DVD-Video:

```
# growisofs -Z /dev/cd0 -dvd-video /path/to/video
```

The `-dvd-video` option will be passed down to mkisofs(8) and will instruct it to create a DVD-Video file system layout. Beside this, the `-dvd-video` option implies `-dvd-compat` growisofs(1) option.

18.7.5 Using a DVD+RW

Unlike CD-RW, a virgin DVD+RW needs to be formatted before first use. The growisofs(1) program will take care of it automatically whenever appropriate, which is the *recommended* way. However you can use the `dvd+rw-format` command to format the DVD+RW:

```
# dvd+rw-format /dev/cd0
```

You need to perform this operation just once, keep in mind that only virgin DVD+RW medias need to be formatted. Then you can burn the DVD+RW in the way seen in previous sections.

If you want to burn new data (burn a totally new file system not append some data) onto a DVD+RW, you do not need to blank it, you just have to write over the previous recording (in performing a new initial session), like this:

```
# growisofs -Z /dev/cd0 -J -R /path/to/newdata
```

DVD+RW format offers the possibility to easily append data to a previous recording. The operation consists in merging a new session to the existing one, it is not multisession writing, growisofs(1) will *grow* the ISO 9660 file system present on the media.

For example, if we want to append data to our previous DVD+RW, we have to use the following:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

The same mkisofs(8) options we used to burn the initial session should be used during next writes.

> **Note:** You may want to use the `-dvd-compat` option if you want better media compatibility with DVD-ROM drives. In the DVD+RW case, this will not prevent you from adding data.

If for any reason you really want to blank the media, do the following:

```
# growisofs -Z /dev/cd0=/dev/zero
```

18.7.6 Using a DVD-RW

A DVD-RW accepts two disc formats: the incremental sequential one and the restricted overwrite. By default DVD-RW discs are in sequential format.

A virgin DVD-RW can be directly written without the need of a formatting operation, however a non-virgin DVD-RW in sequential format needs to be blanked before to be able to write a new initial session.

To blank a DVD-RW in sequential mode, run:

```
# dvd+rw-format -blank=full /dev/cd0
```

> **Note:** A full blanking (`-blank=full`) will take about one hour on a 1x media. A fast blanking can be performed using the `-blank` option if the DVD-RW will be recorded in Disk-At-Once (DAO) mode. To burn the DVD-RW in DAO mode, use the command:
>
> ```
> # growisofs -use-the-force-luke=dao -Z /dev/cd0=imagefile.iso
> ```
>
> The `-use-the-force-luke=dao` option should not be required since growisofs(1) attempts to detect minimally (fast blanked) media and engage DAO write.
>
> In fact one should use restricted overwrite mode with any DVD-RW, this format is more flexible than the default incremental sequential one.

To write data on a sequential DVD-RW, use the same instructions as for the other DVD formats:

```
# growisofs -Z /dev/cd0 -J -R /path/to/data
```

If you want to append some data to your previous recording, you will have to use the growisofs(1) `-M` option. However, if you perform data addition on a DVD-RW in incremental sequential mode, a new session will be created on the disc and the result will be a multi-session disc.

A DVD-RW in restricted overwrite format does not need to be blanked before a new initial session, you just have to overwrite the disc with the `-Z` option, this is similar to the DVD+RW case. It is also possible to grow an existing ISO 9660 file system written on the disc in a same way as for a DVD+RW with the `-M` option. The result will be a one-session DVD.

To put a DVD-RW in the restricted overwrite format, the following command must be used:

```
# dvd+rw-format /dev/cd0
```

To change back to the sequential format use:

```
# dvd+rw-format -blank=full /dev/cd0
```

18.7.7 Multisession

Very few DVD-ROM drives support multisession DVDs, they will most of time, hopefully, only read the first session. DVD+R, DVD-R and DVD-RW in sequential format can accept multiple sessions, the notion of multiple sessions does not exist for the DVD+RW and the DVD-RW restricted overwrite formats.

Using the following command after an initial (non-closed) session on a DVD+R, DVD-R, or DVD-RW in sequential format, will add a new session to the disc:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Using this command line with a DVD+RW or a DVD-RW in restricted overwrite mode, will append data in merging the new session to the existing one. The result will be a single-session disc. This is the way used to add data after an initial write on these medias.

> **Note:** Some space on the media is used between each session for end and start of sessions. Therefore, one should add sessions with large amount of data to optimize media space. The number of sessions is limited to 154 for a DVD+R, about 2000 for a DVD-R, and 127 for a DVD+R Double Layer.

18.7.8 For More Information

To obtain more information about a DVD, the `dvd+rw-mediainfo /dev/cd0` command can be ran with the disc in the drive.

More information about the **dvd+rw-tools** can be found in the growisofs(1) manual page, on the dvd+rw-tools web site (http://fy.chalmers.se/~appro/linux/DVD+RW/) and in the cdwrite mailing list (http://lists.debian.org/cdwrite/) archives.

> **Note:** The `dvd+rw-mediainfo` output of the resulting recording or the media with issues is mandatory for any problem report. Without this output, it will be quite impossible to help you.

18.8 Creating and Using Floppy Disks

Original work by Julio Merino. Rewritten by Martin Karlsson.

Storing data on floppy disks is sometimes useful, for example when one does not have any other removable storage media or when one needs to transfer small amounts of data to another computer.

This section will explain how to use floppy disks in FreeBSD. It will primarily cover formatting and usage of 3.5inch DOS floppies, but the concepts are similar for other floppy disk formats.

18.8.1 Formatting Floppies

18.8.1.1 The Device

Floppy disks are accessed through entries in `/dev`, just like other devices. To access the raw floppy disk, simply use `/dev/fd`*`N`*.

18.8.1.2 Formatting

A floppy disk needs to be low-level formated before it can be used. This is usually done by the vendor, but formatting is a good way to check media integrity. Although it is possible to force larger (or smaller) disk sizes, 1440kB is what most floppy disks are designed for.

To low-level format the floppy disk you need to use fdformat(1). This utility expects the device name as an argument.

Make note of any error messages, as these can help determine if the disk is good or bad.

18.8.1.2.1 Formatting Floppy Disks

Use the `/dev/fd`*`N`* devices to format the floppy. Insert a new 3.5inch floppy disk in your drive and issue:

```
# /usr/sbin/fdformat -f 1440 /dev/fd0
```

18.8.2 The Disk Label

After low-level formatting the disk, you will need to place a disk label on it. This disk label will be destroyed later, but it is needed by the system to determine the size of the disk and its geometry later.

The new disk label will take over the whole disk, and will contain all the proper information about the geometry of the floppy. The geometry values for the disk label are listed in `/etc/disktab`.

You can run now bsdlabel(8) like so:

```
# /sbin/bsdlabel -B -r -w /dev/fd0 fd1440
```

Note: Since FreeBSD 5.1-RELEASE, the bsdlabel(8) utility replaces the old bsdlabel(8) program. With bsdlabel(8) a number of obsolete options and parameters have been retired; in the example above the option `-r` should be removed. For more information, please refer to the bsdlabel(8) manual page.

18.8.3 The File System

Now the floppy is ready to be high-level formated. This will place a new file system on it, which will let FreeBSD read and write to the disk. After creating the new file system, the disk label is destroyed, so if you want to reformat the disk, you will have to recreate the disk label.

The floppy's file system can be either UFS or FAT. FAT is generally a better choice for floppies.

To put a new file system on the floppy, issue:

```
# /sbin/newfs_msdos /dev/fd0
```

The disk is now ready for use.

18.8.4 Using the Floppy

To use the floppy, mount it with mount_msdos(8). One can also use `emulators/mtools` from the ports collection.

18.9 Creating and Using Data Tapes

The major tape media are the 4mm, 8mm, QIC, mini-cartridge and DLT.

18.9.1 4mm (DDS: Digital Data Storage)

4mm tapes are replacing QIC as the workstation backup media of choice. This trend accelerated greatly when Conner purchased Archive, a leading manufacturer of QIC drives, and then stopped production of QIC drives. 4mm drives are small and quiet but do not have the reputation for reliability that is enjoyed by 8mm drives. The cartridges are less expensive and smaller (3 x 2 x 0.5 inches, 76 x 51 x 12 mm) than 8mm cartridges. 4mm, like 8mm, has comparatively short head life for the same reason, both use helical scan.

Data throughput on these drives starts ~150 kB/s, peaking at ~500 kB/s. Data capacity starts at 1.3 GB and ends at 2.0 GB. Hardware compression, available with most of these drives, approximately doubles the capacity. Multi-drive tape library units can have 6 drives in a single cabinet with automatic tape changing. Library capacities reach 240 GB.

The DDS-3 standard now supports tape capacities up to 12 GB (or 24 GB compressed).

4mm drives, like 8mm drives, use helical-scan. All the benefits and drawbacks of helical-scan apply to both 4mm and 8mm drives.

Tapes should be retired from use after 2,000 passes or 100 full backups.

18.9.2 8mm (Exabyte)

8mm tapes are the most common SCSI tape drives; they are the best choice of exchanging tapes. Nearly every site has an Exabyte 2 GB 8mm tape drive. 8mm drives are reliable, convenient and quiet. Cartridges are inexpensive and small (4.8 x 3.3 x 0.6 inches; 122 x 84 x 15 mm). One downside of 8mm tape is relatively short head and tape life due to the high rate of relative motion of the tape across the heads.

Data throughput ranges from ~250 kB/s to ~500 kB/s. Data sizes start at 300 MB and go up to 7 GB. Hardware compression, available with most of these drives, approximately doubles the capacity. These drives are available as single units or multi-drive tape libraries with 6 drives and 120 tapes in a single cabinet. Tapes are changed automatically by the unit. Library capacities reach 840+ GB.

The Exabyte “Mammoth” model supports 12 GB on one tape (24 GB with compression) and costs approximately twice as much as conventional tape drives.

Data is recorded onto the tape using helical-scan, the heads are positioned at an angle to the media (approximately 6 degrees). The tape wraps around 270 degrees of the spool that holds the heads. The spool spins while the tape slides over the spool. The result is a high density of data and closely packed tracks that angle across the tape from one edge to the other.

18.9.3 QIC

QIC-150 tapes and drives are, perhaps, the most common tape drive and media around. QIC tape drives are the least expensive “serious” backup drives. The downside is the cost of media. QIC tapes are expensive compared to 8mm or 4mm tapes, up to 5 times the price per GB data storage. But, if your needs can be satisfied with a half-dozen tapes, QIC may be the correct choice. QIC is the *most* common tape drive. Every site has a QIC drive of some density or another. Therein lies the rub, QIC has a large number of densities on physically similar (sometimes identical) tapes. QIC drives are not quiet. These drives audibly seek before they begin to record data and are clearly audible whenever reading, writing or seeking. QIC tapes measure (6 x 4 x 0.7 inches; 152 x 102 x 17 mm).

Data throughput ranges from ~150 kB/s to ~500 kB/s. Data capacity ranges from 40 MB to 15 GB. Hardware compression is available on many of the newer QIC drives. QIC drives are less frequently installed; they are being supplanted by DAT drives.

Data is recorded onto the tape in tracks. The tracks run along the long axis of the tape media from one end to the other. The number of tracks, and therefore the width of a track, varies with the tape’s capacity. Most if not all newer drives provide backward-compatibility at least for reading (but often also for writing). QIC has a good reputation regarding the safety of the data (the mechanics are simpler and more robust than for helical scan drives).

Tapes should be retired from use after 5,000 backups.

18.9.4 DLT

DLT has the fastest data transfer rate of all the drive types listed here. The 1/2" (12.5mm) tape is contained in a single spool cartridge (4 x 4 x 1 inches; 100 x 100 x 25 mm). The cartridge has a swinging gate along one entire side of the cartridge. The drive mechanism opens this gate to extract the tape leader. The tape leader has an oval hole in it which the drive uses to “hook” the tape. The take-up spool is located inside the tape drive. All the other tape cartridges listed here (9 track tapes are the only exception) have both the supply and take-up spools located inside the tape cartridge itself.

Data throughput is approximately 1.5 MB/s, three times the throughput of 4mm, 8mm, or QIC tape drives. Data capacities range from 10 GB to 20 GB for a single drive. Drives are available in both multi-tape changers and multi-tape, multi-drive tape libraries containing from 5 to 900 tapes over 1 to 20 drives, providing from 50 GB to 9 TB of storage.

With compression, DLT Type IV format supports up to 70 GB capacity.

Data is recorded onto the tape in tracks parallel to the direction of travel (just like QIC tapes). Two tracks are written at once. Read/write head lifetimes are relatively long; once the tape stops moving, there is no relative motion between the heads and the tape.

18.9.5 AIT

AIT is a new format from Sony, and can hold up to 50 GB (with compression) per tape. The tapes contain memory chips which retain an index of the tape's contents. This index can be rapidly read by the tape drive to determine the position of files on the tape, instead of the several minutes that would be required for other tapes. Software such as **SAMS:Alexandria** can operate forty or more AIT tape libraries, communicating directly with the tape's memory chip to display the contents on screen, determine what files were backed up to which tape, locate the correct tape, load it, and restore the data from the tape.

Libraries like this cost in the region of $20,000, pricing them a little out of the hobbyist market.

18.9.6 Using a New Tape for the First Time

The first time that you try to read or write a new, completely blank tape, the operation will fail. The console messages should be similar to:

```
sa0(ncr1:4:0): NOT READY asc:4,1
sa0(ncr1:4:0):  Logical unit is in process of becoming ready
```

The tape does not contain an Identifier Block (block number 0). All QIC tape drives since the adoption of QIC-525 standard write an Identifier Block to the tape. There are two solutions:

- `mt fsf 1` causes the tape drive to write an Identifier Block to the tape.
- Use the front panel button to eject the tape.

  Re-insert the tape and `dump` data to the tape.

  `dump` will report "`DUMP: End of tape detected`" and the console will show: "`HARDWARE FAILURE info:280 asc:80,96`".

  rewind the tape using: `mt rewind`.

  Subsequent tape operations are successful.

18.10 Backups to Floppies

18.10.1 Can I Use Floppies for Backing Up My Data?

Floppy disks are not really a suitable media for making backups as:

- The media is unreliable, especially over long periods of time.
- Backing up and restoring is very slow.
- They have a very limited capacity (the days of backing up an entire hard disk onto a dozen or so floppies has long since passed).

However, if you have no other method of backing up your data then floppy disks are better than no backup at all.

If you do have to use floppy disks then ensure that you use good quality ones. Floppies that have been lying around the office for a couple of years are a bad choice. Ideally use new ones from a reputable manufacturer.

18.10.2 So How Do I Backup My Data to Floppies?

The best way to backup to floppy disk is to use tar(1) with the `-M` (multi volume) option, which allows backups to span multiple floppies.

To backup all the files in the current directory and sub-directory use this (as `root`):

```
# tar Mcvf /dev/fd0 *
```

When the first floppy is full tar(1) will prompt you to insert the next volume (because tar(1) is media independent it refers to volumes; in this context it means floppy disk).

```
Prepare volume #2 for /dev/fd0 and hit return:
```

This is repeated (with the volume number incrementing) until all the specified files have been archived.

18.10.3 Can I Compress My Backups?

Unfortunately, tar(1) will not allow the `-z` option to be used for multi-volume archives. You could, of course, gzip(1) all the files, tar(1) them to the floppies, then gunzip(1) the files again!

18.10.4 How Do I Restore My Backups?

To restore the entire archive use:

```
# tar Mxvf /dev/fd0
```

There are two ways that you can use to restore only specific files. First, you can start with the first floppy and use:

```
# tar Mxvf /dev/fd0 filename
```

The utility tar(1) will prompt you to insert subsequent floppies until it finds the required file.

Alternatively, if you know which floppy the file is on then you can simply insert that floppy and use the same command as above. Note that if the first file on the floppy is a continuation from the previous one then tar(1) will warn you that it cannot restore it, even if you have not asked it to!

18.11 Backup Strategies

Original work by Lowell Gilbert.

The first requirement in devising a backup plan is to make sure that all of the following problems are covered:

- Disk failure
- Accidental file deletion
- Random file corruption
- Complete machine destruction (e.g. fire), including destruction of any on-site backups.

It is perfectly possible that some systems will be best served by having each of these problems covered by a completely different technique. Except for strictly personal systems with very low-value data, it is unlikely that one technique would cover all of them.

Some of the techniques in the toolbox are:

- Archives of the whole system, backed up onto permanent media offsite. This actually provides protection against all of the possible problems listed above, but is slow and inconvenient to restore from. You can keep copies of the backups onsite and/or online, but there will still be inconveniences in restoring files, especially for non-privileged users.
- Filesystem snapshots. This is really only helpful in the accidental file deletion scenario, but it can be *very* helpful in that case, and is quick and easy to deal with.
- Copies of whole filesystems and/or disks (e.g. periodic rsync of the whole machine). This is generally most useful in networks with unique requirements. For general protection against disk failure, it is usually inferior to RAID. For restoring accidentally deleted files, it can be comparable to UFS snapshots, but that depends on your preferences.
- RAID. Minimizes or avoids downtime when a disk fails. At the expense of having to deal with disk failures more often (because you have more disks), albeit at a much lower urgency.
- Checking fingerprints of files. The mtree(8) utility is very useful for this. Although it is not a backup technique, it helps guarantee that you will notice when you need to resort to your backups. This is particularly important for offline backups, and should be checked periodically.

It is quite easy to come up with even more techniques, many of them variations on the ones listed above. Specialized requirements will usually lead to specialized techniques (for example, backing up a live database usually requires a method particular to the database software as an intermediate step). The important thing is to know what dangers you want to protect against, and how you will handle each.

18.12 Backup Basics

The three major backup programs are dump(8), tar(1), and cpio(1).

18.12.1 Dump and Restore

The traditional UNIX backup programs are `dump` and `restore`. They operate on the drive as a collection of disk blocks, below the abstractions of files, links and directories that are created by the file systems. `dump` backs up an entire file system on a device. It is unable to backup only part of a file system or a directory tree that spans more than one file system. `dump` does not write files and directories to tape, but rather writes the raw data blocks that comprise files and directories.

> **Note:** If you use `dump` on your root directory, you would not back up `/home`, `/usr` or many other directories since these are typically mount points for other file systems or symbolic links into those file systems.

`dump` has quirks that remain from its early days in Version 6 of AT&T UNIX (circa 1975). The default parameters are suitable for 9-track tapes (6250 bpi), not the high-density media available today (up to 62,182 ftpi). These defaults must be overridden on the command line to utilize the capacity of current tape drives.

It is also possible to backup data across the network to a tape drive attached to another computer with `rdump` and `rrestore`. Both programs rely upon rcmd(3) and ruserok(3) to access the remote tape drive. Therefore, the user performing the backup must be listed in the `.rhosts` file on the remote computer. The arguments to `rdump` and `rrestore` must be suitable to use on the remote computer. When `rdump`ing from a FreeBSD computer to an Exabyte tape drive connected to a Sun called `komodo`, use:

```
# /sbin/rdump 0dsbfu 54000 13000 126 komodo:/dev/nsa8 /dev/da0a 2>&1
```

Beware: there are security implications to allowing `.rhosts` authentication. Evaluate your situation carefully.

It is also possible to use `dump` and `restore` in a more secure fashion over `ssh`.

Example 18-1. Using `dump` over `ssh`

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh -c blowfish \
          targetuser@targetmachine.example.com dd of=/mybigfiles/dump-usr-l0.gz
```

Or using `dump`'s built-in method, setting the environment variable `RSH`:

Example 18-2. Using `dump` over `ssh` with `RSH` set

```
# RSH=/usr/bin/ssh /sbin/dump -0uan -f targetuser@targetmachine.example.com:/dev/sa0 /usr
```

18.12.2 `tar`

tar(1) also dates back to Version 6 of AT&T UNIX (circa 1975). `tar` operates in cooperation with the file system; it writes files and directories to tape. `tar` does not support the full range of options that are available from cpio(1), but it does not require the unusual command pipeline that `cpio` uses.

On FreeBSD 5.3 and later, both GNU `tar` and the default `bsdtar` are available. The GNU version can be invoked with `gtar`. It supports remote devices using the same syntax as `rdump`. To `tar` to an Exabyte tape drive connected to a Sun called `komodo`, use:

```
# /usr/bin/gtar cf komodo:/dev/nsa8 . 2>&1
```

The same could be accomplished with `bsdtar` by using a pipeline and `rsh` to send the data to a remote tape drive.

```
# tar cf - . | rsh hostname dd of=tape-device obs=20b
```

If you are worried about the security of backing up over a network you should use the `ssh` command instead of `rsh`.

18.12.3 cpio

cpio(1) is the original UNIX file interchange tape program for magnetic media. `cpio` has options (among many others) to perform byte-swapping, write a number of different archive formats, and pipe the data to other programs. This last feature makes `cpio` an excellent choice for installation media. `cpio` does not know how to walk the directory tree and a list of files must be provided through `stdin`.

`cpio` does not support backups across the network. You can use a pipeline and `rsh` to send the data to a remote tape drive.

```
# for f in directory_list; do
find $f >> backup.list
done
# cpio -v -o --format=newc < backup.list | ssh user@host "cat > backup_device"
```

Where *`directory_list`* is the list of directories you want to back up, *`user@host`* is the user/hostname combination that will be performing the backups, and *`backup_device`* is where the backups should be written to (e.g., `/dev/nsa0`).

18.12.4 pax

pax(1) is IEEE/POSIX's answer to `tar` and `cpio`. Over the years the various versions of `tar` and `cpio` have gotten slightly incompatible. So rather than fight it out to fully standardize them, POSIX created a new archive utility. `pax` attempts to read and write many of the various `cpio` and `tar` formats, plus new formats of its own. Its command set more resembles `cpio` than `tar`.

18.12.5 Amanda

Amanda (Advanced Maryland Network Disk Archiver) is a client/server backup system, rather than a single program. An **Amanda** server will backup to a single tape drive any number of computers that have **Amanda** clients and a network connection to the **Amanda** server. A common problem at sites with a number of large disks is that the length of time required to backup to data directly to tape exceeds the amount of time available for the task. **Amanda** solves this problem. **Amanda** can use a "holding disk" to backup several file systems at the same time. **Amanda** creates "archive sets": a group of tapes used over a period of time to create full backups of all the file systems listed in **Amanda**'s configuration file. The "archive set" also contains nightly incremental (or differential) backups of all the file systems. Restoring a damaged file system requires the most recent full backup and the incremental backups.

The configuration file provides fine control of backups and the network traffic that **Amanda** generates. **Amanda** will use any of the above backup programs to write the data to tape. **Amanda** is available as either a port or a package, it is not installed by default.

18.12.6 Do Nothing

"Do nothing" is not a computer program, but it is the most widely used backup strategy. There are no initial costs. There is no backup schedule to follow. Just say no. If something happens to your data, grin and bear it!

If your time and your data is worth little to nothing, then "Do nothing" is the most suitable backup program for your computer. But beware, UNIX is a useful tool, you may find that within six months you have a collection of files that are valuable to you.

"Do nothing" is the correct backup method for `/usr/obj` and other directory trees that can be exactly recreated by your computer. An example is the files that comprise the HTML or PostScript version of this Handbook. These document formats have been created from SGML input files. Creating backups of the HTML or PostScript files is not necessary. The SGML files are backed up regularly.

18.12.7 Which Backup Program Is Best?

dump(8) *Period*. Elizabeth D. Zwicky torture tested all the backup programs discussed here. The clear choice for preserving all your data and all the peculiarities of UNIX file systems is `dump`. Elizabeth created file systems containing a large variety of unusual conditions (and some not so unusual ones) and tested each program by doing a backup and restore of those file systems. The peculiarities included: files with holes, files with holes and a block of nulls, files with funny characters in their names, unreadable and unwritable files, devices, files that change size during the backup, files that are created/deleted during the backup and more. She presented the results at LISA V in Oct. 1991. See torture-testing Backup and Archive Programs (http://berdmann.dyndns.org/zwicky/testdump.doc.html).

18.12.8 Emergency Restore Procedure

18.12.8.1 Before the Disaster

There are only four steps that you need to perform in preparation for any disaster that may occur.

First, print the bsdlabel from each of your disks (e.g. `bsdlabel da0 | lpr`), your file system table (`/etc/fstab`) and all boot messages, two copies of each.

Second, determine that the boot and fix-it floppies (`boot.flp` and `fixit.flp`) have all your devices. The easiest way to check is to reboot your machine with the boot floppy in the floppy drive and check the boot messages. If all your devices are listed and functional, skip on to step three.

Otherwise, you have to create two custom bootable floppies which have a kernel that can mount all of your disks and access your tape drive. These floppies must contain: `fdisk`, `bsdlabel`, `newfs`, `mount`, and whichever backup program you use. These programs must be statically linked. If you use `dump`, the floppy must contain `restore`.

Third, create backup tapes regularly. Any changes that you make after your last backup may be irretrievably lost. Write-protect the backup tapes.

Fourth, test the floppies (either `boot.flp` and `fixit.flp` or the two custom bootable floppies you made in step two.) and backup tapes. Make notes of the procedure. Store these notes with the bootable floppy, the printouts and

the backup tapes. You will be so distraught when restoring that the notes may prevent you from destroying your backup tapes (How? In place of `tar xvf /dev/sa0`, you might accidentally type `tar cvf /dev/sa0` and over-write your backup tape).

For an added measure of security, make bootable floppies and two backup tapes each time. Store one of each at a remote location. A remote location is NOT the basement of the same office building. A number of firms in the World Trade Center learned this lesson the hard way. A remote location should be physically separated from your computers and disk drives by a significant distance.

Example 18-3. A Script for Creating a Bootable Floppy

```
#!/bin/sh
#
# create a restore floppy
#
# format the floppy
#
PATH=/bin:/sbin:/usr/sbin:/usr/bin

fdformat -q fd0
if [ $? -ne 0 ]
then
  echo "Bad floppy, please use a new one"
  exit 1
fi

# place boot blocks on the floppy
#
bsdlabel -w -B /dev/fd0c fd1440

#
# newfs the one and only partition
#
newfs -t 2 -u 18 -l 1 -c 40 -i 5120 -m 5 -o space /dev/fd0a

#
# mount the new floppy
#
mount /dev/fd0a /mnt

#
# create required directories
#
mkdir /mnt/dev
mkdir /mnt/bin
mkdir /mnt/sbin
mkdir /mnt/etc
mkdir /mnt/root
mkdir /mnt/mnt   # for the root partition
mkdir /mnt/tmp
mkdir /mnt/var

#
```

```
# populate the directories
#
if [ ! -x /sys/compile/MINI/kernel ]
then
  cat &lt;&lt; EOM
The MINI kernel does not exist, please create one.
Here is an example config file:
#
# MINI -- A kernel to get FreeBSD onto a disk.
#
machine         "i386"
cpu             "I486_CPU"
ident           MINI
maxusers        5

options         INET                    # needed for _tcp _icmpstat _ipstat
                                        #            _udpstat _tcpstat _udb
options         FFS                     #Berkeley Fast File System
options         FAT_CURSOR              #block cursor in syscons or pccons
options         SCSI_DELAY=15           #Be pessimistic about Joe SCSI device
options         NCONS=2                 #1 virtual consoles
options         USERCONFIG              #Allow user configuration with -c XXX

config          kernel root on da0 swap on da0 and da1 dumps on da0

device          isa0
device          pci0

device          fdc0 at isa? port "IO_FD1" bio irq 6 drq 2 vector fdintr
device          fd0 at fdc0 drive 0

device          ncr0

device          scbus0

device          sc0 at isa? port "IO_KBD" tty irq 1 vector scintr
device          npx0 at isa? port "IO_NPX" irq 13 vector npxintr

device          da0
device          da1
device          da2

device          sa0

pseudo-device   loop            # required by INET
pseudo-device   gzip            # Exec gzipped a.out’s
EOM
  exit 1
fi

cp -f /sys/compile/MINI/kernel /mnt

gzip -c -best /sbin/init &gt; /mnt/sbin/init
```

```
gzip -c -best /sbin/fsck &gt; /mnt/sbin/fsck
gzip -c -best /sbin/mount &gt; /mnt/sbin/mount
gzip -c -best /sbin/halt &gt; /mnt/sbin/halt
gzip -c -best /sbin/restore &gt; /mnt/sbin/restore

gzip -c -best /bin/sh &gt; /mnt/bin/sh
gzip -c -best /bin/sync &gt; /mnt/bin/sync

cp /root/.profile /mnt/root

cp -f /dev/MAKEDEV /mnt/dev
chmod 755 /mnt/dev/MAKEDEV

chmod 500 /mnt/sbin/init
chmod 555 /mnt/sbin/fsck /mnt/sbin/mount /mnt/sbin/halt
chmod 555 /mnt/bin/sh /mnt/bin/sync
chmod 6555 /mnt/sbin/restore

#
# create the devices nodes
#
cd /mnt/dev
./MAKEDEV std
./MAKEDEV da0
./MAKEDEV da1
./MAKEDEV da2
./MAKEDEV sa0
./MAKEDEV pty0
cd /

#
# create minimum file system table
#
cat &gt; /mnt/etc/fstab &lt;&lt;EOM
/dev/fd0a    /    ufs    rw  1  1
EOM

#
# create minimum passwd file
#
cat &gt; /mnt/etc/passwd &lt;&lt;EOM
root:*:0:0:Charlie &:/root:/bin/sh
EOM

cat &gt; /mnt/etc/master.passwd &lt;&lt;EOM
root::0:0::0:0:Charlie &:/root:/bin/sh
EOM

chmod 600 /mnt/etc/master.passwd
chmod 644 /mnt/etc/passwd
/usr/sbin/pwd_mkdb -d/mnt/etc /mnt/etc/master.passwd

#
```

```
# umount the floppy and inform the user
#
/sbin/umount /mnt
echo "The floppy has been unmounted and is now ready."
```

18.12.8.2 After the Disaster

The key question is: did your hardware survive? You have been doing regular backups so there is no need to worry about the software.

If the hardware has been damaged, the parts should be replaced before attempting to use the computer.

If your hardware is okay, check your floppies. If you are using a custom boot floppy, boot single-user (type `-s` at the `boot:` prompt). Skip the following paragraph.

If you are using the `boot.flp` and `fixit.flp` floppies, keep reading. Insert the `boot.flp` floppy in the first floppy drive and boot the computer. The original install menu will be displayed on the screen. Select the `Fixit--Repair mode with CDROM or floppy.` option. Insert the `fixit.flp` when prompted. `restore` and the other programs that you need are located in `/mnt2/rescue` (`/mnt2/stand` for FreeBSD versions older than 5.2).

Recover each file system separately.

Try to `mount` (e.g. `mount /dev/da0a /mnt`) the root partition of your first disk. If the bsdlabel was damaged, use `bsdlabel` to re-partition and label the disk to match the label that you printed and saved. Use `newfs` to re-create the file systems. Re-mount the root partition of the floppy read-write (`mount -u -o rw /mnt`). Use your backup program and backup tapes to recover the data for this file system (e.g. `restore vrf /dev/sa0`). Unmount the file system (e.g. `umount /mnt`). Repeat for each file system that was damaged.

Once your system is running, backup your data onto new tapes. Whatever caused the crash or data loss may strike again. Another hour spent now may save you from further distress later.

18.13 Network, Memory, and File-Backed File Systems

Reorganized and enhanced by Marc Fonvieille.

Aside from the disks you physically insert into your computer: floppies, CDs, hard drives, and so forth; other forms of disks are understood by FreeBSD - the *virtual disks*.

These include network file systems such as the Network File System and Coda, memory-based file systems and file-backed file systems.

According to the FreeBSD version you run, you will have to use different tools for creation and use of file-backed and memory-based file systems.

Note: Use devfs(5) to allocate device nodes transparently for the user.

18.13.1 File-Backed File System

The utility mdconfig(8) is used to configure and enable memory disks, md(4), under FreeBSD. To use mdconfig(8), you have to load md(4) module or to add the support in your kernel configuration file:

```
device md
```

The mdconfig(8) command supports three kinds of memory backed virtual disks: memory disks allocated with malloc(9), memory disks using a file or swap space as backing. One possible use is the mounting of floppy or CD images kept in files.

To mount an existing file system image:

Example 18-4. Using `mdconfig` to Mount an Existing File System Image

```
# mdconfig -a -t vnode -f diskimage -u 0
# mount /dev/md0 /mnt
```

To create a new file system image with mdconfig(8):

Example 18-5. Creating a New File-Backed Disk with `mdconfig`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdconfig -a -t vnode -f newimage -u 0
# bsdlabel -w md0 auto
# newfs md0a
/dev/md0a: 5.0MB (10224 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 1.25MB, 80 blks, 192 inodes.
super-block backups (for fsck -b #) at:
 160, 2720, 5280, 7840
# mount /dev/md0a /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0a       4710    4  4330     0%    /mnt
```

If you do not specify the unit number with the `-u` option, mdconfig(8) will use the md(4) automatic allocation to select an unused device. The name of the allocated unit will be output on stdout like `md4`. For more details about mdconfig(8), please refer to the manual page.

The utility mdconfig(8) is very useful, however it asks many command lines to create a file-backed file system. FreeBSD also comes with a tool called mdmfs(8), this program configures a md(4) disk using mdconfig(8), puts a UFS file system on it using newfs(8), and mounts it using mount(8). For example, if you want to create and mount the same file system image as above, simply type the following:

Example 18-6. Configure and Mount a File-Backed Disk with `mdmfs`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdmfs -F newimage -s 5m md0 /mnt
# df /mnt
```

```
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0        4718    4  4338     0%    /mnt
```

If you use the option `md` without unit number, mdmfs(8) will use md(4) auto-unit feature to automatically select an unused device. For more details about mdmfs(8), please refer to the manual page.

18.13.2 Memory-Based File System

For a memory-based file system the "swap backing" should normally be used. Using swap backing does not mean that the memory disk will be swapped out to disk by default, but merely that the memory disk will be allocated from a memory pool which can be swapped out to disk if needed. It is also possible to create memory-based disk which are malloc(9) backed, but using malloc backed memory disks, especially large ones, can result in a system panic if the kernel runs out of memory.

Example 18-7. Creating a New Memory-Based Disk with `mdconfig`

```
# mdconfig -a -t malloc -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
 using 4 cylinder groups of 1.27MB, 81 blks, 256 inodes.
 with soft updates
super-block backups (for fsck -b #) at:
 32, 2624, 5216, 7808
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1        4846    2  4458     0%    /mnt
```

Example 18-8. Creating a New Memory-Based Disk with `mdmfs`

```
# mdmfs -M -s 5m md2 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md2        4846    2  4458     0%    /mnt
```

Instead of using a malloc(9) backed file system, it is possible to use swap, for that just replace `malloc` with `swap` in the command line of mdconfig(8). The mdmfs(8) utility by default (without `-M`) creates a swap-based disk. For more details, please refer to mdconfig(8) and mdmfs(8) manual pages.

18.13.3 Detaching a Memory Disk from the System

When a memory-based or file-based file system is not used, you should release all resources to the system. The first thing to do is to unmount the file system, then use mdconfig(8) to detach the disk from the system and release the resources.

For example to detach and free all resources used by `/dev/md4`:

```
# mdconfig -d -u 4
```

It is possible to list information about configured md(4) devices in using the command `mdconfig -l`.

18.14 File System Snapshots

Contributed by Tom Rhodes.

FreeBSD offers a feature in conjunction with Soft Updates: File system snapshots.

Snapshots allow a user to create images of specified file systems, and treat them as a file. Snapshot files must be created in the file system that the action is performed on, and a user may create no more than 20 snapshots per file system. Active snapshots are recorded in the superblock so they are persistent across unmount and remount operations along with system reboots. When a snapshot is no longer required, it can be removed with the standard rm(1) command. Snapshots may be removed in any order, however all the used space may not be acquired because another snapshot will possibly claim some of the released blocks.

The un-alterable `snapshot` file flag is set by mksnap_ffs(8) after initial creation of a snapshot file. The unlink(1) command makes an exception for snapshot files since it allows them to be removed.

Snapshots are created with the mount(8) command. To place a snapshot of `/var` in the file `/var/snapshot/snap` use the following command:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

Alternatively, you can use mksnap_ffs(8) to create a snapshot:

```
# mksnap_ffs /var /var/snapshot/snap
```

One can find snapshot files on a file system (e.g. `/var`) by using the find(1) command:

```
# find /var -flags snapshot
```

Once a snapshot has been created, it has several uses:

- Some administrators will use a snapshot file for backup purposes, because the snapshot can be transfered to CDs or tape.
- File integrity, fsck(8) may be ran on the snapshot. Assuming that the file system was clean when it was mounted, you should always get a clean (and unchanging) result. This is essentially what the background fsck(8) process does.
- Run the dump(8) utility on the snapshot. A dump will be returned that is consistent with the file system and the timestamp of the snapshot. dump(8) can also take a snapshot, create a dump image and then remove the snapshot in one command using the `-L` flag.
- mount(8) the snapshot as a frozen image of the file system. To mount(8) the snapshot `/var/snapshot/snap` run:

  ```
  # mdconfig -a -t vnode -f /var/snapshot/snap -u 4
  # mount -r /dev/md4 /mnt
  ```

You can now walk the hierarchy of your frozen `/var` file system mounted at `/mnt`. Everything will initially be in the same state it was during the snapshot creation time. The only exception is that any earlier snapshots will appear as zero length files. When the use of a snapshot has delimited, it can be unmounted with:

```
# umount /mnt
# mdconfig -d -u 4
```

For more information about `softupdates` and file system snapshots, including technical papers, you can visit Marshall Kirk McKusick's website at http://www.mckusick.com/.

18.15 磁碟空間配額(Quota)

磁碟配額(Quota)屬於作業系統上的選用功能，可以用來限制使用者或群組的可用空間大小，或者檔案的總數多寡。這功能通常用在多人共用的系統環境上，因為要限制各使用者或各群組所能運用的系統資源。如此一來，就可避免磁碟空間被某使用者或某群組全部耗盡。

18.15.1 啓用磁碟配額

在用磁碟配額之前，請先確認kernel 已經有作相關設定，也就是kernel 設定檔要有下面這行：

```
options QUOTA
```

預設的GENERIC kernel 並不會加上這項，所以若要啓用就必需加上，並重新編譯、安裝kernel。kernel 設定部分可參閱Chapter 8 的説明。

接著就是在/etc/rc.conf 設定啓動磁碟配額。請加上下列這行：

```
enable_quotas="YES"
```

為了能更完善的控管磁碟配額的啓動，還有一個設定可以用。通常開機時，quotacheck(8) 程式會檢查各檔案系統上的配額。quotacheck(8) 可以確保配額資料庫的資料與實際檔案系統的資料有符合。但這功能也會在開機時，會對啓動時間造成相當明顯的影響。若想跳過這步驟，則可以在/etc/rc.conf 加上：

```
check_quotas="NO"
```

最後，要記得改/etc/fstab 來啓用以檔案系統為對象的磁碟配額功能。也可以啓用針對使用者或群組，或者兩者皆有之的磁碟配額。

若要啓用針對使用者的配額，可以在/etc/fstab 內要設定的檔案系統加上userquota 選項。比如：

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

同理若要啓用針對群組的配額，則把剛剛的userquota 換成groupquota 即可。而若要兩者同時啓用，那麼則是：

```
/dev/da1s2g    /home    ufs rw,userquota,groupquota 1 2
```

針對使用者以及群組的磁碟配額設定檔，預設分別會放在該檔案系統根目錄的quota.user 以及quota.group 。細節部分請參閱fstab(5)。雖然fstab(5) 提到可以為配額設定檔指定其他地方，但並不建議如此作，因為各種磁碟配額管理工具並不見得對這些預設值能隨之彈性變化。

接下來就可以用新kernel 來重開機。/etc/rc 會自動執行相關指令以對/etc/fstab 有設定配額管理的部分，作初始設定。所以並不需要逐一手動產生相關空的配額設定檔。

正常操作過程中，並不需要手動執行quotacheck(8)、quotaon(8) 、quotaoff(8) 這些指令。不過，若要更熟悉相關操作方式的話，或許可以閱讀相關的manual 線上説明。

18.15.2 設定配額限制

一旦開始啓用配額管理之後，請記得確認是否有真的啓用。可以打下列指令來作簡單檢查：

```
# quota -v
```

應該可以看到有關各檔案系統的配額限量，以及現在使用量的摘要訊息。

現在可以開始用edquota(8) 來設定各磁碟配額的限制。

有幾種選項可以用來限制使用者或群組所能運用的磁碟空間，以及所能建立的檔案數量多寡。可以依磁碟空間(block 配額)或檔案數量(inode 配額)，或者搭配兩者一起設定。而每種限制還可以細分為兩類：hard(硬性)上限、soft(彈性)上限。

硬性上限是不能超過的。一旦使用者達到硬性上限時，就無法在該檔案系統上繼續使用更多的使用空間了。舉例來說，若有位使用者的硬性上限為500 KB，而目前用了490 KB，那麼他就只能再多用10 KB 而已，若要新增的檔案有11 KB 就會失敗。

然而，彈性上限則可允許一定時間內的超額使用，這段期間稱為grace period(寬限期)，預設值是一週。若使用者持續超額使用並超出grace period 而逾期，則彈性上限就會轉為硬性上限，而不允許該使用者繼續新增空間。直到該使用者的空間已經清到低於彈性上限之後，才會重設grace period。

下面則是使用edquota(8) 的例子。在執行edquota(8) 時，會進入設定磁碟配額上限的編輯器內，至於是哪一種編輯器則視您的`EDITOR` 環境變數而定，若沒設定`EDITOR` 的話，則會用**vi** 編輯器。

```
# edquota -u test

Quotas for user test:
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
        inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
        inodes in use: 0, limits (soft = 50, hard = 60)
```

一般來說，每個啓動了磁碟配額的檔案系統都會有兩行設定。第一行是block 上限，而另一行則是inode 上限。若要更改磁碟配額上限，只需要修改後面的數值即可。舉例來說，要增加這位使用者的block 上限部分：把彈性上限50 調為500，硬性上限則由75 調為600 ，只需修改下面這行：

```
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
```

改為下列：

```
/usr: kbytes in use: 65, limits (soft = 500, hard = 600)
```

然後存檔離開後，新的配額設定就會立即生效。

有時候會想一次改大範圍UID 的帳號設定，這時可以用edquota(8) 的`-p` 參數功能來完成。首先，把某個帳號調為想要的相關配額，然後可以用`edquota -p protouser startuid-enduid` 之類的方式來改。舉例來說，假設`test` 這帳號已經設定好相關配額，然後要改的對象為UID 從10,000 到19,999 的帳號，那麼就可以下列指令來設定同樣的配額：

```
# edquota -p test 10000-19999
```

細節說明請參閱edquota(8)。

18.15.3 檢查磁碟配額設定、磁碟使用量

可以用quota(1) 或repquota(8) 來檢查磁碟配額設定，以及磁碟使用量。quota(1) 可用來檢查單一使用者或群組的磁碟配額、磁碟使用量。不過一般帳號只能查自己的以及自己群組的磁碟配額、磁碟使用量，只有系統管理者帳號才能察看所有使用者、群組的配額設定與使用量。而repquota(8) 則可以看到所有已啓動磁碟配額的檔案系統設定、磁碟使用量摘要。

下面例子則是在兩個有配額設定的檔案系統上，打`quota -v`的顯示結果：

```
Disk quotas for user test (uid 1002):
     Filesystem  usage    quota   limit   grace   files   quota   limit   grace
           /usr      65*     50      75   5days       7      50      60
       /usr/var       0      50      75               0      50      60
```

在上面這例中，該使用者在`/usr`的彈性配額是50 KB，實際上已經超額多用15 KB，而grace period 還有5 天就逾期。請注意這個星號`*`是表示目前該使用者已經超越其配額的彈性上限了。

一般來說，若使用者並沒有用到某個檔案系統，那麼就算該檔案有啓用磁碟配額，在quota(1) 也不會顯示出來。而`-v`參數則可以把這些檔案系統都全部列出來，比如上例中的`/usr/var`。

18.15.4 透過NFS 使用磁碟配額

NFS server 端可以强制以quota subsystem(配額子系統)來用磁碟配額。而NFS client 端則可以透過rpc.rquotad(8) daemon 來讓quota(1) 指令抓到相關配額資料，也就可以讓client 端的使用者察看其配額的統計資料。

若要啓用`rpc.rquotad`，可以在`/etc/inetd.conf`加上下列類似設定：

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

然後重啓`inetd`即可：

```
# kill -HUP ‘cat /var/run/inetd.pid‘
```

18.16 Encrypting Disk Partitions

Contributed by Lucky Green.

FreeBSD offers excellent online protections against unauthorized data access. File permissions and Mandatory Access Control (MAC) (see Chapter 16) help prevent unauthorized third-parties from accessing data while the operating system is active and the computer is powered up. However, the permissions enforced by the operating system are irrelevant if an attacker has physical access to a computer and can simply move the computer’s hard drive to another system to copy and analyze the sensitive data.

Regardless of how an attacker may have come into possession of a hard drive or powered-down computer, both **GEOM Based Disk Encryption (gbde)** and `geli` cryptographic subsystems in FreeBSD are able to protect the data on the computer’s file systems against even highly-motivated attackers with significant resources. Unlike cumbersome encryption methods that encrypt only individual files, `gbde` and `geli` transparently encrypt entire file systems. No cleartext ever touches the hard drive’s platter.

18.16.1 Disk Encryption with gbde

1. Become `root`

   Configuring **gbde** requires super-user privileges.

```
% su -
Password:
```

2. Add gbde(4) Support to the Kernel Configuration File

   Add the following line to the kernel configuration file:

   ```
   options GEOM_BDE
   ```

   Rebuild the kernel as described in Chapter 8.

   Reboot into the new kernel.

18.16.1.1 Preparing the Encrypted Hard Drive

The following example assumes that you are adding a new hard drive to your system that will hold a single encrypted partition. This partition will be mounted as `/private`. **gbde** can also be used to encrypt `/home` and `/var/mail`, but this requires more complex instructions which exceed the scope of this introduction.

1. Add the New Hard Drive

   Install the new drive to the system as explained in Section 18.3. For the purposes of this example, a new hard drive partition has been added as `/dev/ad4s1c`. The `/dev/ad0s1*` devices represent existing standard FreeBSD partitions on the example system.

   ```
   # ls /dev/ad*
   /dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
   /dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
   /dev/ad0s1a     /dev/ad0s1d     /dev/ad4
   ```

2. Create a Directory to Hold gbde Lock Files

   ```
   # mkdir /etc/gbde
   ```

   The **gbde** lock file contains information that **gbde** requires to access encrypted partitions. Without access to the lock file, **gbde** will not be able to decrypt the data contained in the encrypted partition without significant manual intervention which is not supported by the software. Each encrypted partition uses a separate lock file.

3. Initialize the gbde Partition

   A **gbde** partition must be initialized before it can be used. This initialization needs to be performed only once:

   ```
   # gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c
   ```

   gbde(8) will open your editor, permitting you to set various configuration options in a template. For use with UFS1 or UFS2, set the sector_size to 2048:

   ```
   $FreeBSD: src/sbin/gbde/template.txt,v 1.1 2002/10/20 11:16:13 phk Exp $
   #
   # Sector size is the smallest unit of data which can be read or written.
   # Making it too small decreases performance and decreases available space.
   # Making it too large may prevent filesystems from working.  512 is the
   # minimum and always safe.  For UFS, use the fragment size
   #
   sector_size     =       2048
   [...]
   ```

gbde(8) will ask you twice to type the passphrase that should be used to secure the data. The passphrase must be the same both times. **gbde**'s ability to protect your data depends entirely on the quality of the passphrase that you choose. [9]

The `gbde init` command creates a lock file for your **gbde** partition that in this example is stored as `/etc/gbde/ad4s1c`.

> **Caution: gbde** lock files *must* be backed up together with the contents of any encrypted partitions. While deleting a lock file alone cannot prevent a determined attacker from decrypting a **gbde** partition, without the lock file, the legitimate owner will be unable to access the data on the encrypted partition without a significant amount of work that is totally unsupported by gbde(8) and its designer.

4. Attach the Encrypted Partition to the Kernel

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
```

You will be asked to provide the passphrase that you selected during the initialization of the encrypted partition. The new encrypted device will show up in `/dev` as `/dev/device_name.bde`:

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. Create a File System on the Encrypted Device

Once the encrypted device has been attached to the kernel, you can create a file system on the device. To create a file system on the encrypted device, use newfs(8). Since it is much faster to initialize a new UFS2 file system than it is to initialize the old UFS1 file system, using newfs(8) with the `-O2` option is recommended.

```
# newfs -U -O2 /dev/ad4s1c.bde
```

> **Note:** The newfs(8) command must be performed on an attached **gbde** partition which is identified by a `*.bde` extension to the device name.

6. Mount the Encrypted Partition

Create a mount point for the encrypted file system.

```
# mkdir /private
```

Mount the encrypted file system.

```
# mount /dev/ad4s1c.bde /private
```

7. Verify That the Encrypted File System is Available

The encrypted file system should now be visible to df(1) and be available for use.

```
% df -H
Filesystem        Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a      1037M    72M   883M     8%    /
/devfs            1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f       8.1G    55K   7.5G     0%    /home
/dev/ad0s1e      1037M   1.1M   953M     0%    /tmp
/dev/ad0s1d       6.1G   1.9G   3.7G    35%    /usr
```

```
/dev/ad4s1c.bde   150G   4.1K   138G     0%    /private
```

18.16.1.2 Mounting Existing Encrypted File Systems

After each boot, any encrypted file systems must be re-attached to the kernel, checked for errors, and mounted, before the file systems can be used. The required commands must be executed as user `root`.

1. Attach the gbde Partition to the Kernel

   ```
   # gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
   ```

   You will be asked to provide the passphrase that you selected during initialization of the encrypted **gbde** partition.

2. Check the File System for Errors

   Since encrypted file systems cannot yet be listed in `/etc/fstab` for automatic mounting, the file systems must be checked for errors by running fsck(8) manually before mounting.

   ```
   # fsck -p -t ffs /dev/ad4s1c.bde
   ```

3. Mount the Encrypted File System

   ```
   # mount /dev/ad4s1c.bde /private
   ```

   The encrypted file system is now available for use.

18.16.1.2.1 Automatically Mounting Encrypted Partitions

It is possible to create a script to automatically attach, check, and mount an encrypted partition, but for security reasons the script should not contain the gbde(8) password. Instead, it is recommended that such scripts be run manually while providing the password via the console or ssh(1).

As of FreeBSD 5.2-RELEASE, there is a new `rc.d` script provided. Arguments for this script can be passed via rc.conf(5), for example:

```
gbde_autoattach_all="YES"
gbde_devices="ad4s1c"
```

This will require that the **gbde** passphrase be entered at boot time. After typing the correct passphrase, the **gbde** encrypted partition will be mounted automatically. This can be very useful when using **gbde** on notebooks.

18.16.1.3 Cryptographic Protections Employed by gbde

gbde(8) encrypts the sector payload using 128-bit AES in CBC mode. Each sector on the disk is encrypted with a different AES key. For more information on **gbde**'s cryptographic design, including how the sector keys are derived from the user-supplied passphrase, see gbde(4).

18.16.1.4 Compatibility Issues

sysinstall(8) is incompatible with **gbde**-encrypted devices. All `*.bde` devices must be detached from the kernel before starting sysinstall(8) or it will crash during its initial probing for devices. To detach the encrypted device used in our example, use the following command:

```
# gbde detach /dev/ad4s1c
```

Also note that, as vinum(4) does not use the geom(4) subsystem, you cannot use **gbde** with **vinum** volumes.

18.16.2 Disk Encryption with `geli`

Contributed by Daniel Gerzo.

A new cryptographic GEOM class is available as of FreeBSD 6.0 - `geli`. It is currently being developed by Pawel Jakub Dawidek <`pjd@FreeBSD.org`>. `Geli` is different to `gbde`; it offers different features and uses a different scheme for doing cryptographic work.

The most important features of geli(8) are:

- Utilizes the crypto(9) framework ——when cryptographic hardware is available, `geli` will use it automatically.
- Supports multiple cryptographic algorithms (currently AES, Blowfish, and 3DES).
- Allows the root partition to be encrypted. The passphrase used to access the encrypted root partition will be requested during the system boot.
- Allows the use of two independent keys (e.g. a “key” and a “company key”).
- `geli` is fast - performs simple sector-to-sector encryption.
- Allows backup and restore of Master Keys. When a user has to destroy his keys, it will be possible to get access to the data again by restoring keys from the backup.
- Allows to attach a disk with a random, one-time key ——useful for swap partitions and temporary file systems.

More `geli` features can be found in the geli(8) manual page.

The next steps will describe how to enable support for `geli` in the FreeBSD kernel and will explain how to create a new `geli` encryption provider. At the end it will be demonstrated how to create an encrypted swap partition using features provided by `geli`.

In order to use `geli`, you must be running FreeBSD 6.0-RELEASE or later. Super-user privileges will be required since modifications to the kernel are necessary.

1. Adding `geli` Support to the Kernel Configuration File

   Add the following lines to the kernel configuration file:

   ```
   options GEOM_ELI
   device crypto
   ```

   Rebuild the kernel as described in Chapter 8.

   Alternatively, the `geli` module can be loaded at boot time. Add the following line to the `/boot/loader.conf`:

   ```
   geom_eli_load="YES"
   ```

geli(8) should now be supported by the kernel.

2. Generating the Master Key

   The following example will describe how to generate a key file, which will be used as part of the Master Key for the encrypted provider mounted under `/private`. The key file will provide some random data used to encrypt the Master Key. The Master Key will be protected by a passphrase as well. Provider's sector size will be 4kB big. Furthermore, the discussion will describe how to attach the `geli` provider, create a file system on it, how to mount it, how to work with it, and finally how to detach it.

   It is recommended to use a bigger sector size (like 4kB) for better performance.

   The Master Key will be protected with a passphrase and the data source for key file will be `/dev/random`. The sector size of `/dev/da2.eli`, which we call provider, will be 4kB.

```
# dd if=/dev/random of=/root/da2.key bs=64 count=1
# geli init -s 4096 -K /root/da2.key /dev/da2
Enter new passphrase:
Reenter new passphrase:
```

   It is not mandatory that both a passphrase and a key file are used; either method of securing the Master Key can be used in isolation.

   If key file is given as "-", standard input will be used. This example shows how more than one key file can be used.

```
# cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2
```

3. Attaching the Provider with the generated Key

```
# geli attach -k /root/da2.key /dev/da2
Enter passphrase:
```

   The new plaintext device will be named `/dev/da2.eli`.

```
# ls /dev/da2*
/dev/da2  /dev/da2.eli
```

4. Creating the new File System

```
# dd if=/dev/random of=/dev/da2.eli bs=1m
# newfs /dev/da2.eli
# mount /dev/da2.eli /private
```

   The encrypted file system should be visible to df(1) and be available for use now.

```
# df -H
Filesystem     Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a    248M    89M   139M    38%    /
/devfs         1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f    7.7G   2.3G   4.9G    32%    /usr
/dev/ad0s1d    989M   1.5M   909M     0%    /tmp
/dev/ad0s1e    3.9G   1.3G   2.3G    35%    /var
/dev/da2.eli   150G   4.1K   138G     0%    /private
```

5. Unmounting and Detaching the Provider

   Once the work on the encrypted partition is done, and the `/private` partition is no longer needed, it is prudent to consider unmounting and detaching the `geli` encrypted partition from the kernel.

```
# umount /private
# geli detach da2.eli
```

More information about the use of geli(8) can be found in the manual page.

18.16.2.1 Encrypting a Swap Partition

The following example demonstrates how to create a `geli` encrypted swap partition.

```
# dd if=/dev/random of=/dev/ad0s1b bs=1m
# geli onetime -d -a 3des ad0s1b
# swapon /dev/ad0s1b.eli
```

18.16.2.2 Using the `geli rc.d` Script

`geli` comes with a `rc.d` script which can be used to simplify the usage of `geli`. An example of configuring `geli` through rc.conf(5) follows:

```
geli_devices="da2"
geli_da2_flags="-p -k /root/da2.key"
```

This will configure `/dev/da2` as a `geli` provider of which the Master Key file is located in `/root/da2.key`, and `geli` will not use a passphrase when attaching the provider (note that this can only be used if -P was given during the `geli` init phase). The system will detach the `geli` provider from the kernel before the system shuts down.

More information about configuring `rc.d` is provided in the rc.d section of the Handbook.

18.17 Encrypting Swap Space

Written by Christian Brüffer.

Swap encryption in FreeBSD is easy to configure and has been available since FreeBSD 5.3-RELEASE. Depending on which version of FreeBSD is being used, different options are available and configuration can vary slightly. From FreeBSD 6.0-RELEASE onwards, the gbde(8) or geli(8) encryption systems can be used for swap encryption. With earlier versions, only gbde(8) is available. Both systems use the `encswap` rc.d script.

The previous section, Encrypting Disk Partitions, includes a short discussion on the different encryption systems.

18.17.1 Why should Swap be Encrypted?

Like the encryption of disk partitions, encryption of swap space is done to protect sensitive information. Imagine an application that e.g. deals with passwords. As long as these passwords stay in physical memory, all is well. However, if the operating system starts swapping out memory pages to free space for other applications, the passwords may be written to the disk platters unencrypted and easy to retrieve for an adversary. Encrypting swap space can be a solution for this scenario.

18.17.2 Preparation

Note: For the remainder of this section, `ad0s1b` will be the swap partition.

Up to this point the swap has been unencrypted. It is possible that there are already passwords or other sensitive data on the disk platters in cleartext. To rectify this, the data on the swap partition should be overwritten with random garbage:

```
# dd if=/dev/random of=/dev/ad0s1b bs=1m
```

18.17.3 Swap Encryption with gbde(8)

If FreeBSD 6.0-RELEASE or newer is being used, the `.bde` suffix should be added to the device in the respective `/etc/fstab` swap line:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1b.bde         none            swap    sw              0       0
```

For systems prior to FreeBSD 6.0-RELEASE, the following line in `/etc/rc.conf` is also needed:

```
gbde_swap_enable="YES"
```

18.17.4 Swap Encryption with geli(8)

Alternatively, the procedure for using geli(8) for swap encryption is similar to that of using gbde(8). The `.eli` suffix should be added to the device in the respective `/etc/fstab` swap line:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1b.eli         none            swap    sw              0       0
```

geli(8) uses the AES algorithm with a key length of 256 bit by default.

Optionally, these defaults can be altered using the `geli_swap_flags` option in `/etc/rc.conf`. The following line tells the `encswap` rc.d script to create geli(8) swap partitions using the Blowfish algorithm with a key length of 128 bit, a sectorsize of 4 kilobytes and the “detach on last close” option set:

```
geli_swap_flags="-a blowfish -l 128 -s 4096 -d"
```

Please refer to the description of the `onetime` command in the geli(8) manual page for a list of possible options.

18.17.5 Verifying that it Works

Once the system has been rebooted, proper operation of the encrypted swap can be verified using the `swapinfo` command.

If gbde(8) is being used:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.bde    542720        0   542720     0%
```

If geli(8) is being used:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.eli    542720        0   542720     0%
```

Notes

1. 譯註：雖然有些設備沒有『碟片』，例如USB 隨身碟，不過在此仍把Disk 譯為『碟片裝置』。此外，為方便起見，後文所有的Disk 都譯為『磁碟』。
2. 譯註：基於相同的理由，現在BSD partition 常稱為BSD label，或簡稱label。
3. 譯註：老實說我看不懂這句指的是什麼？原文是**sysinstall** Label editor favors the e partition for non-root, non-swap partitions.
4. 譯註：如果您自始至終都不打算將這個磁碟用於FreeBSD 之外的作業系統，那可以算是個好理由。不過就算如此，用slice 模式也沒什麼壞處就是了:-)。
5. 譯註：da 是direct access (disk) 的縮寫；ad 是ata disk 的縮寫。
6. 譯註：我對這句的意思沒什麼信心，原文是IBM's OS/2 however, will "appropriate" any partition it finds which it does not understand.
7. 譯註：原文這裡是用「和」，但要視實際使用方式而定。例如用RAID-0 就不會增加穩定度:)。
8. 譯註：例如按F1 可以進入控制卡BIOS 之類的資訊。
9. For tips on how to select a secure passphrase that is easy to remember, see the Diceware Passphrase (http://world.std.com/~reinhold/diceware.html) website.

Chapter 19 GEOM: Modular Disk Transformation Framework

Written by Tom Rhodes.

19.1 概述

本章涵蓋如何在FreeBSD 的GEOM 架構下使用磁碟，包含用來設定幾種常用的RAID 的控制工具。本章不會深入探討GEOM 如何處理底層的I/O，這類資訊請參考geom(4) 及相關的SEE ALSO 部份。本章也非RAID 設定指南，在這裡只會討論目前GEOM 支援的RAID 模式。

讀完這章，您將了解：

- 透過GEOM 可支援哪些模式的RAID。
- 如何使用基本工具來配置、操作、維護不同模式的RAID。
- 如何透過GEOM 來完成鏡射(mirror)、分散連結(stripe)、加密(encrypt) 、遠端連接磁碟等。
- 當GEOM 架構下的磁碟發生問題，如何排除。

在開始閱讀這章之前，您需要：

- 了解FreeBSD 如何看待磁碟(Chapter 18) 。
- 知道如何設定、安裝新的FreeBSD 核心(Chapter 8) 。

19.2 GEOM 導論

GEOM 透過privoder(即`/dev/` 下的特殊裝置檔案) 來操控classes(如Master Boot Records、BSD labels 等) 。GEOM 支援多種軟體RAID 配置，透過GEOM 存取時，作業系統和應用程式不會意識到GEOM 存在。

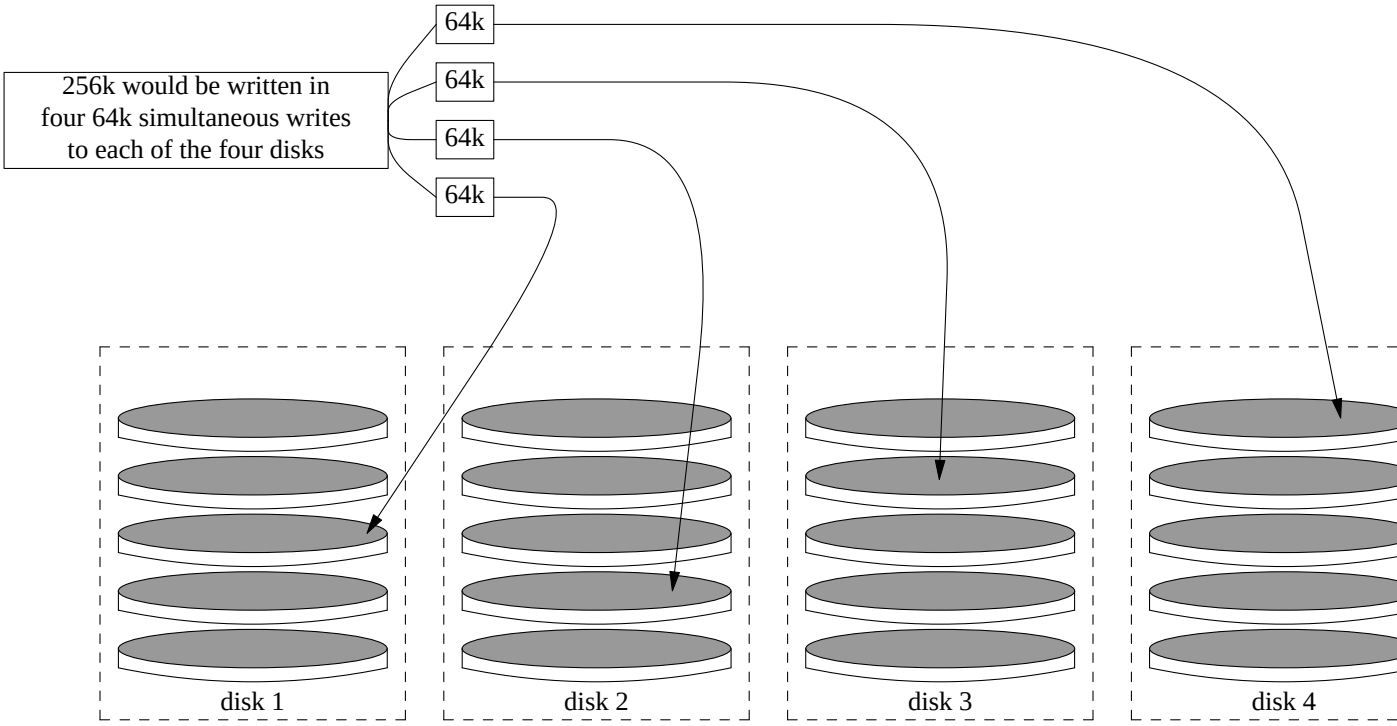
19.3 RAID0 - 分散連結(striping)

Written by Tom Rhodes and Murray Stokely.

分散連結(striping) 可用來連結多個磁碟成為一大塊空間。很多時候硬體控制器可以完成這件事，不過GEOM 也提供了軟體版本的RAID0，也就是分散連結(striping)。

在RAID0 裡，資料會被切分成很多塊，再分散寫入全部的磁碟。例如要寫入256k 的資料到單一磁碟，在四個磁碟的RAID0 中可同時寫入64k 到四個磁碟裡，因此可大幅提升I/O 效能。如果使用更多的磁碟控制器，I/O 效能可再提升。

由於讀或寫時會同步交錯對許多磁碟進行I/O 處理，因此RAID0 的每個磁碟必須大小一樣。



用未格式化的ATA 磁碟來建立分散連結(striping)

1. 載入`geom_stripe` kernel module：

   ```
   # kldload geom_stripe.ko
   ```

2. 確定掛載點(mount point)存在。如果想用分散連結(striping)的空間做為根目錄(root partition，即`/`)，則先用個暫時的掛載點，如`/mnt`：

   ```
   # mkdir /mnt
   ```

3. 確認要用來分散連結(striping)的裝置名稱，接著建立新的分散連結(striping)。例如下面的指令會分散連結(striping)兩個未使用、尚未分割區的ATA 磁碟(`/dev/ad2` 和`/dev/ad3`) ：

   ```
   #   gstripe label -v st0 /dev/ad2 /dev/ad3
   ```

   ```
   # gstripe label -v st0 /dev/ad2 /dev/ad3
   ```

4. 用下面的指令來建立分割區表(partition table)：

   ```
   # bsdlabel -wB /dev/stripe/st0
   ```

5. 除了先前建立的`st0` ，這個步驟還會在`/dev/stripe` 下新增兩個裝置：`st0a` 和`st0c`。利用`newfs` 指令可以在`st0a` 建立檔案系統：

   ```
   # newfs -U /dev/stripe/st0a
   ```

   螢幕上會有一堆數字傾瀉而過，幾秒鐘後就會完成。此時空間已建立，可用來掛載使用了。

下面指令可用來手動掛載分散連結(striping)空間：

```
# mount /dev/stripe/st0a /mnt
```

如果要在開機時自動掛載，在`/etc/fstab` 加入這塊空間的資訊：

```
# echo "/dev/stripe/st0a /mnt ufs rw 2 2" \
    >> /etc/fstab
```

而geom kernel module 必須在系統初始化時自動載入，因此在/boot/lodaer.conf 加入一行：

```
# echo 'geom_stripe_load="YES"' >> /boot/loader.conf
```

19.4 RAID1 - 鏡射(Mirroring)

許多企業或個人用戶用鏡射(mirroring) 來不中斷系統進行備份。鏡射簡單來說就是在B 磁碟上重覆一份A 磁碟的資料，或者C+D 磁碟重覆A+B 磁碟的資料。不論設定如何，最重要的是所有磁碟或分割區(partition) 上的資料都會被複製，之後可在不中斷服務的情況下復原、備份資料，使儲存的資料更安全。

開始之前，請先確定系統上有兩個容量相同的磁碟，後面的範例假設這兩顆磁碟是direct access(da(4)) SCSI 磁碟。

首先我們假設FreeBSD 安裝在第一個磁碟上，且只有兩個分割區(partition)。其中一個是交換分割區(swap partition，大小為RAM 的兩倍)，而剩下的全用於根目錄(即/，root file system)。當然要在不同掛載點(mount point) 切出更多分割區(partition) 也可以，不過難度會大幅提升，因為必須手動操作bsdlabel(8) 和fdisk(8) 工具。

重開機並等到系統完全初始化完畢，用root 登入。

建立/dev/mirror/gm 裝置並以/dev/da1 連結：

```
# gmirror label -vnb round-robin gm0 /dev/da1
```

這時系統應該會回應：

```
Metadata value stored on /dev/da1.
Done.
```

初始化GEOM，這動作會自動載入/boot/kernel/geom_mirror.ko kernel module：

```
# gmirror load
```

> **Note:** 這動作應該會在/dev/mirror 下建立gm0 裝置結點(device node)。

在這個新建的gm0 裝置上安置一般的fdisk label 和開機磁區：

```
# fdisk -vBI /dev/mirror/gm0
```

接著安置bsdlabel 資訊：

```
# bsdlabel -wB /dev/mirror/gm0s1
```

> **Note:** 如果存在多個slice 和分割區(partition)，記得修改上兩指令的參數，且另一個磁碟上的slice 和分割區(partition) 大小必須相同。

用newfs(8) 工具在gm0s1a 裝置結點建立預設的檔案系統：

```
# newfs -U /dev/mirror/gm0s1a
```

系統會印出許多資訊和一大堆數字，這是正常的。確認是否有認何錯誤，接著就可以將這個裝置掛載到/mnt 掛載點(mount mount)：

```
# mount /dev/mirror/gm0s1a /mnt
```

接著將原本開機磁碟的資料搬移到新的檔案系統(/mnt)。範例是用dump(8) 和restore(8) ，不過用dd(1) 也可以。

```
# dump -L -0 -f- / |(cd /mnt && restore -r -v -f-)
```

執行上述指令時，只要將恰當的檔案系統掛在正確的位置，應該就能成功。

接著編輯/mnt/etc/fstab 檔將swap file 那行移除或註解起來。[1]請參考下面範例，並根據新磁碟修改其它的檔案系統資訊：

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
#/dev/da0s2b             none            swap    sw              0       0
/dev/mirror/gm0s1a       /               ufs     rw              1       1
```

在目前的根目錄及新的根目錄建立boot.conf 檔案，這個檔案可以『幫助』系統BIOS 開機：

```
# echo "1:da(1,a)/boot/loader" > /boot.config

# echo "1:da(1,a)/boot/loader" > /mnt/boot.config
```

> **Note:** 在兩個根目錄上都新增檔案是為了安全起見，如果因為某些原因新的根目錄無法開機，至少還可用原本的根目錄。

接著在/boot/loader.conf 新增兩行：

```
# echo 'geom_mirror_load="YES"' >> /mnt/boot/loader.conf
```

這會指示loader(8) 在開機時載入geom_mirror.ko kernel module。

重開機：

```
# shutdown -r now
```

如果一切順利，系統應該會從gm0s1a 裝置開機，接下來出現login 提示畫面。如果出錯了，請參閱下面Troubleshooting 那一節。現在可以將da0 磁碟加入gm0 裝置：

```
# gmirror configure -a gm0
# gmirror insert gm0 /dev/da0
```

其中-a 旗標告訴gmirror(8) 使用「自動同步(automatic synchronization)」，例如自動同步寫入磁碟的動作。manual 說明了如何重建、取代磁碟等，不過manual 裡的範例是用data 而不是gm0。

19.4.1 Troubleshooting

19.4.1.1 系統無法開機

如果開機提示類似這樣：

```
ffs_mountroot: can't find rootvp
Root mount failed: 6
mountroot>
```

請用機器面板上的Power 按鈕或reset 按鈕來重開機，並在開機選單選(6)，這樣子，系統就會進入loader(8)交談模式。這時候，請照下面指令來手動載入所需的kernel module ，也就是`geom_mirror.ko`：

```
OK? load geom_mirror.ko
OK? boot
```

如果這樣成功了的話，表示因為某些原因無法自動載入kernel module。請將：

```
options GEOM_MIRROR
```

加入到核心設定檔(kernel configuration file)，重編並安裝核心。這應該能解決這個問題。

Notes

1. 請注意，將`fstab` 的swap file 那行註解起來，通常表示：您得用別的方法來重建swap。詳情請參考Section 11.14。

Chapter 20 The Vinum Volume Manager

Originally written by Greg Lehey.

20.1 Synopsis

No matter what disks you have, there are always potential problems:

- They can be too small.
- They can be too slow.
- They can be too unreliable.

One way some users safeguard themselves against such issues is through the use of multiple, and sometimes redundant, disks.

In addition to supporting various cards and controllers for hardware RAID systems, the base FreeBSD system includes the Vinum Volume Manager, a block device driver that implements virtual disk drives.

Vinum provides more flexibility, performance, and reliability than traditional disk storage, and implements RAID-0, RAID-1, and RAID-5 models both individually and in combination.

This chapter provides an overview of potential problems with traditional disk storage, and an introduction to the Vinum Volume Manager.

20.2 Disks Are Too Small

Vinum is a so-called *Volume Manager*, a virtual disk driver that addresses these three problems. Let us look at them in more detail. Various solutions to these problems have been proposed and implemented:

Disks are getting bigger, but so are data storage requirements. Often you will find you want a file system that is bigger than the disks you have available. Admittedly, this problem is not as acute as it was ten years ago, but it still exists. Some systems have solved this by creating an abstract device which stores its data on a number of disks.

20.3 Access Bottlenecks

Modern systems frequently need to access data in a highly concurrent manner. For example, large FTP or HTTP servers can maintain thousands of concurrent sessions and have multiple 100 Mbit/s connections to the outside world, well beyond the sustained transfer rate of most disks.

Current disk drives can transfer data sequentially at up to 70 MB/s, but this value is of little importance in an environment where many independent processes access a drive, where they may achieve only a fraction of these values. In such cases it is more interesting to view the problem from the viewpoint of the disk subsystem: the important parameter is the load that a transfer places on the subsystem, in other words the time for which a transfer occupies the drives involved in the transfer.

In any disk transfer, the drive must first position the heads, wait for the first sector to pass under the read head, and then perform the transfer. These actions can be considered to be atomic: it does not make any sense to interrupt them.

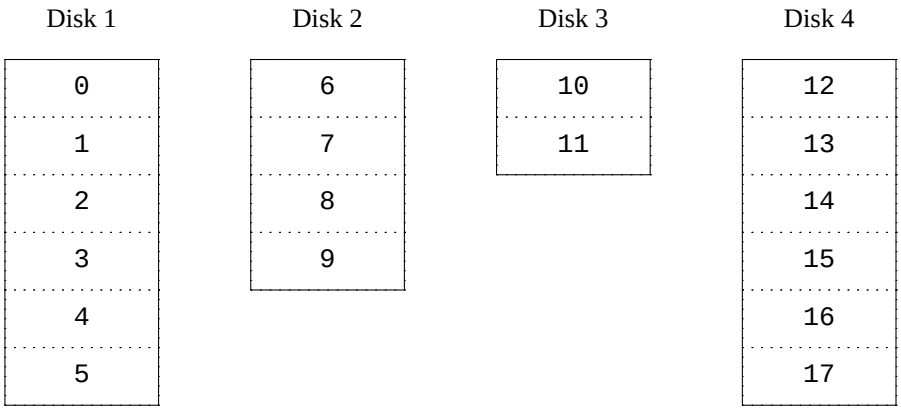
Consider a typical transfer of about 10 kB: the current generation of high-performance disks can position the heads in an average of 3.5 ms. The fastest drives spin at 15,000 rpm, so the average rotational latency (half a revolution) is 2 ms. At 70 MB/s, the transfer itself takes about 150 μs, almost nothing compared to the positioning time. In such a case, the effective transfer rate drops to a little over 1 MB/s and is clearly highly dependent on the transfer size.

The traditional and obvious solution to this bottleneck is "more spindles": rather than using one large disk, it uses several smaller disks with the same aggregate storage space. Each disk is capable of positioning and transferring independently, so the effective throughput increases by a factor close to the number of disks used.

The exact throughput improvement is, of course, smaller than the number of disks involved: although each drive is capable of transferring in parallel, there is no way to ensure that the requests are evenly distributed across the drives. Inevitably the load on one drive will be higher than on another.

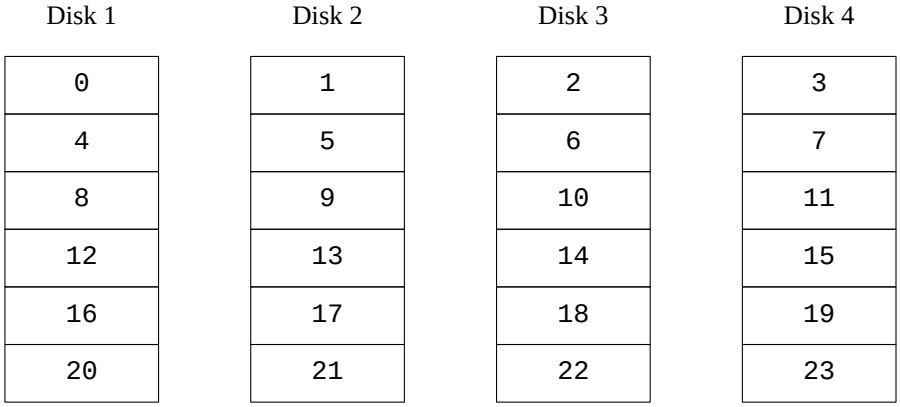
The evenness of the load on the disks is strongly dependent on the way the data is shared across the drives. In the following discussion, it is convenient to think of the disk storage as a large number of data sectors which are addressable by number, rather like the pages in a book. The most obvious method is to divide the virtual disk into groups of consecutive sectors the size of the individual physical disks and store them in this manner, rather like taking a large book and tearing it into smaller sections. This method is called *concatenation* and has the advantage that the disks are not required to have any specific size relationships. It works well when the access to the virtual disk is spread evenly about its address space. When access is concentrated on a smaller area, the improvement is less marked. Figure 20-1 illustrates the sequence in which storage units are allocated in a concatenated organization.

Figure 20-1. Concatenated Organization



An alternative mapping is to divide the address space into smaller, equal-sized components and store them sequentially on different devices. For example, the first 256 sectors may be stored on the first disk, the next 256 sectors on the next disk and so on. After filling the last disk, the process repeats until the disks are full. This mapping is called *striping* or RAID-0 [1]. Striping requires somewhat more effort to locate the data, and it can cause additional I/O load where a transfer is spread over multiple disks, but it can also provide a more constant load across the disks. Figure 20-2 illustrates the sequence in which storage units are allocated in a striped organization.

Figure 20-2. Striped Organization



20.4 Data Integrity

The final problem with current disks is that they are unreliable. Although disk drive reliability has increased tremendously over the last few years, they are still the most likely core component of a server to fail. When they do, the results can be catastrophic: replacing a failed disk drive and restoring data to it can take days.

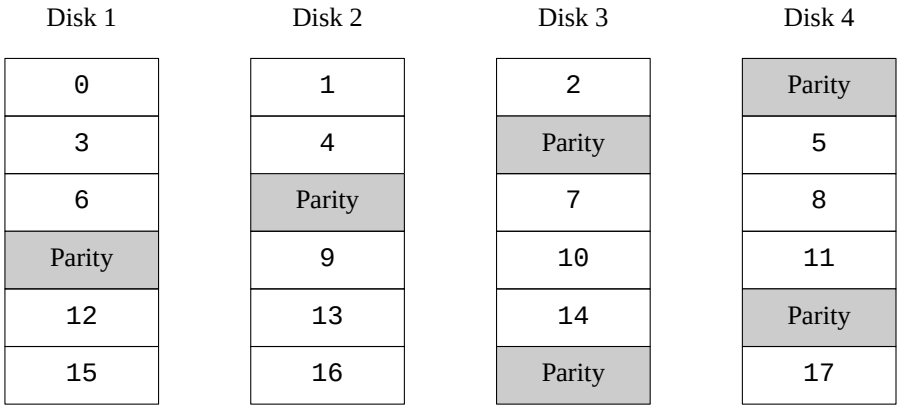
The traditional way to approach this problem has been *mirroring*, keeping two copies of the data on different physical hardware. Since the advent of the RAID levels, this technique has also been called RAID level 1 or RAID-1. Any write to the volume writes to both locations; a read can be satisfied from either, so if one drive fails, the data is still available on the other drive.

Mirroring has two problems:

- The price. It requires twice as much disk storage as a non-redundant solution.
- The performance impact. Writes must be performed to both drives, so they take up twice the bandwidth of a non-mirrored volume. Reads do not suffer from a performance penalty: it even looks as if they are faster.

An alternative solution is *parity*, implemented in the RAID levels 2, 3, 4 and 5. Of these, RAID-5 is the most interesting. As implemented in Vinum, it is a variant on a striped organization which dedicates one block of each stripe to parity of the other blocks. As implemented by Vinum, a RAID-5 plex is similar to a striped plex, except that it implements RAID-5 by including a parity block in each stripe. As required by RAID-5, the location of this parity block changes from one stripe to the next. The numbers in the data blocks indicate the relative block numbers.

Figure 20-3. RAID-5 Organization



Compared to mirroring, RAID-5 has the advantage of requiring significantly less storage space. Read access is similar to that of striped organizations, but write access is significantly slower, approximately 25% of the read performance. If one drive fails, the array can continue to operate in degraded mode: a read from one of the remaining accessible drives continues normally, but a read from the failed drive is recalculated from the corresponding block from all the remaining drives.

20.5 Vinum Objects

In order to address these problems, Vinum implements a four-level hierarchy of objects:

- The most visible object is the virtual disk, called a *volume*. Volumes have essentially the same properties as a UNIX disk drive, though there are some minor differences. They have no size limitations.
- Volumes are composed of *plexes*, each of which represent the total address space of a volume. This level in the hierarchy thus provides redundancy. Think of plexes as individual disks in a mirrored array, each containing the same data.
- Since Vinum exists within the UNIX disk storage framework, it would be possible to use UNIX partitions as the building block for multi-disk plexes, but in fact this turns out to be too inflexible: UNIX disks can have only a limited number of partitions. Instead, Vinum subdivides a single UNIX partition (the *drive*) into contiguous areas called *subdisks*, which it uses as building blocks for plexes.
- Subdisks reside on Vinum *drives*, currently UNIX partitions. Vinum drives can contain any number of subdisks. With the exception of a small area at the beginning of the drive, which is used for storing configuration and state information, the entire drive is available for data storage.

The following sections describe the way these objects provide the functionality required of Vinum.

20.5.1 Volume Size Considerations

Plexes can include multiple subdisks spread over all drives in the Vinum configuration. As a result, the size of an individual drive does not limit the size of a plex, and thus of a volume.

20.5.2 Redundant Data Storage

Vinum implements mirroring by attaching multiple plexes to a volume. Each plex is a representation of the data in a volume. A volume may contain between one and eight plexes.

Although a plex represents the complete data of a volume, it is possible for parts of the representation to be physically missing, either by design (by not defining a subdisk for parts of the plex) or by accident (as a result of the failure of a drive). As long as at least one plex can provide the data for the complete address range of the volume, the volume is fully functional.

20.5.3 Performance Issues

Vinum implements both concatenation and striping at the plex level:

- A *concatenated plex* uses the address space of each subdisk in turn.
- A *striped plex* stripes the data across each subdisk. The subdisks must all have the same size, and there must be at least two subdisks in order to distinguish it from a concatenated plex.

20.5.4 Which Plex Organization?

The version of Vinum supplied with FreeBSD 8.0 implements two kinds of plex:

- Concatenated plexes are the most flexible: they can contain any number of subdisks, and the subdisks may be of different length. The plex may be extended by adding additional subdisks. They require less CPU time than striped plexes, though the difference in CPU overhead is not measurable. On the other hand, they are most susceptible to hot spots, where one disk is very active and others are idle.
- The greatest advantage of striped (RAID-0) plexes is that they reduce hot spots: by choosing an optimum sized stripe (about 256 kB), you can even out the load on the component drives. The disadvantages of this approach are (fractionally) more complex code and restrictions on subdisks: they must be all the same size, and extending a plex by adding new subdisks is so complicated that Vinum currently does not implement it. Vinum imposes an additional, trivial restriction: a striped plex must have at least two subdisks, since otherwise it is indistinguishable from a concatenated plex.

Table 20-1 summarizes the advantages and disadvantages of each plex organization.

Table 20-1. Vinum Plex Organizations

Plex type	Minimum subdisks	Can add subdisks	Must be equal size	Application

Plex type	Minimum subdisks	Can add subdisks	Must be equal size	Application
concatenated	1	yes	no	Large data storage with maximum placement flexibility and moderate performance
striped	2	no	yes	High performance in combination with highly concurrent access

20.6 Some Examples

Vinum maintains a *configuration database* which describes the objects known to an individual system. Initially, the user creates the configuration database from one or more configuration files with the aid of the vinum(8) utility program. Vinum stores a copy of its configuration database on each disk slice (which Vinum calls a *device*) under its control. This database is updated on each state change, so that a restart accurately restores the state of each Vinum object.

20.6.1 The Configuration File

The configuration file describes individual Vinum objects. The definition of a simple volume might be:

```
drive a device /dev/da3h
volume myvol
  plex org concat
    sd length 512m drive a
```

This file describes four Vinum objects:

- The *drive* line describes a disk partition (*drive*) and its location relative to the underlying hardware. It is given the symbolic name *a*. This separation of the symbolic names from the device names allows disks to be moved from one location to another without confusion.
- The *volume* line describes a volume. The only required attribute is the name, in this case *myvol*.
- The *plex* line defines a plex. The only required parameter is the organization, in this case *concat*. No name is necessary: the system automatically generates a name from the volume name by adding the suffix *.px*, where *x* is the number of the plex in the volume. Thus this plex will be called *myvol.p0*.
- The *sd* line describes a subdisk. The minimum specifications are the name of a drive on which to store it, and the length of the subdisk. As with plexes, no name is necessary: the system automatically assigns names derived from the plex name by adding the suffix *.sx*, where *x* is the number of the subdisk in the plex. Thus Vinum gives this subdisk the name *myvol.p0.s0*.

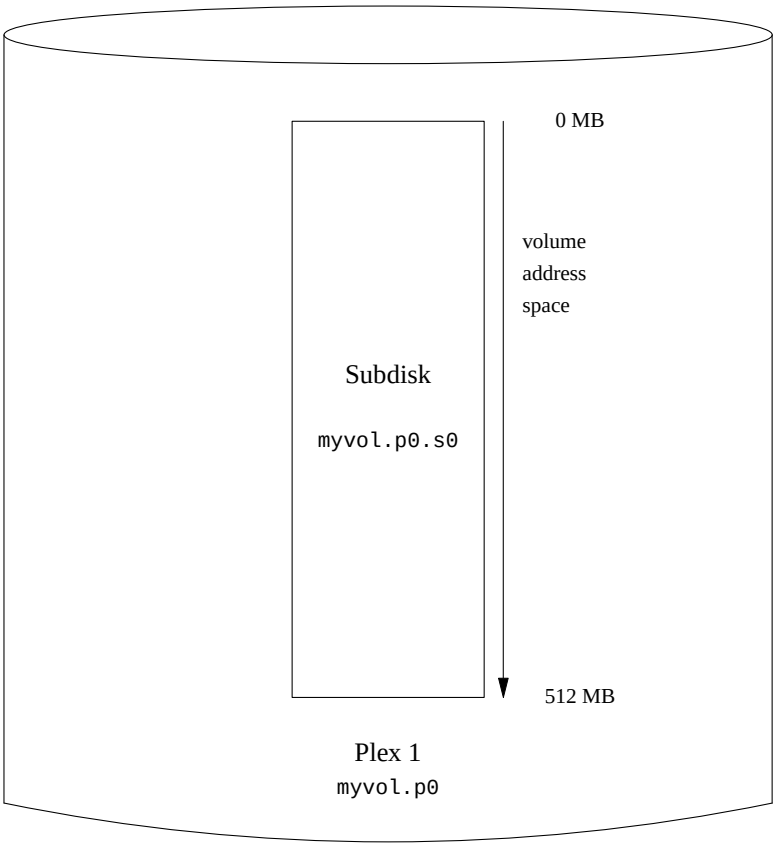
After processing this file, vinum(8) produces the following output:

```
# vinum -> create config1
```

```
     Configuration summary
     Drives:         1 (4 configured)
     Volumes:        1 (4 configured)
     Plexes:         1 (8 configured)
     Subdisks:       1 (16 configured)

D a                     State: up       Device /dev/da3h        Avail: 2061/2573 MB (80%)

V myvol                 State: up       Plexes:       1 Size:        512 MB

P myvol.p0            C State: up       Subdisks:     1 Size:        512 MB

S myvol.p0.s0           State: up       PO:        0  B Size:        512 MB
```

This output shows the brief listing format of vinum(8). It is represented graphically in Figure 20-4.

Figure 20-4. A Simple Vinum Volume



This figure, and the ones which follow, represent a volume, which contains the plexes, which in turn contain the subdisks. In this trivial example, the volume contains one plex, and the plex contains one subdisk.

This particular volume has no specific advantage over a conventional disk partition. It contains a single plex, so it is not redundant. The plex contains a single subdisk, so there is no difference in storage allocation from a conventional disk partition. The following sections illustrate various more interesting configuration methods.

20.6.2 Increased Resilience: Mirroring

The resilience of a volume can be increased by mirroring. When laying out a mirrored volume, it is important to ensure that the subdisks of each plex are on different drives, so that a drive failure will not take down both plexes. The following configuration mirrors a volume:

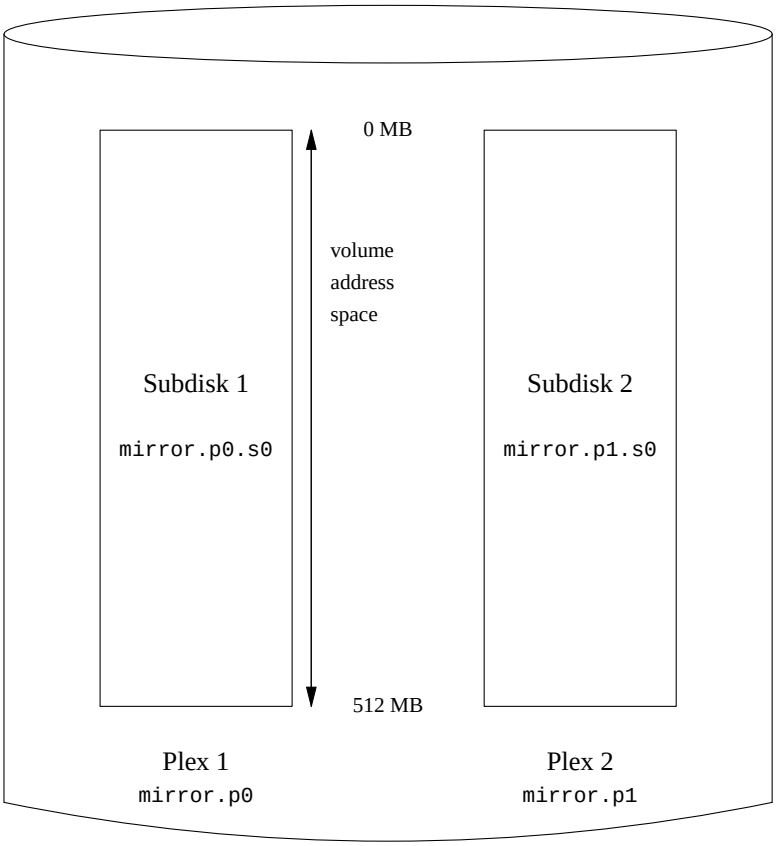
```
drive b device /dev/da4h
volume mirror
     plex org concat
       sd length 512m drive a
  plex org concat
    sd length 512m drive b
```

In this example, it was not necessary to specify a definition of drive *a* again, since Vinum keeps track of all objects in its configuration database. After processing this definition, the configuration looks like:

```
Drives:         2 (4 configured)
Volumes:        2 (4 configured)
Plexes:         3 (8 configured)
Subdisks:       3 (16 configured)

D a                     State: up       Device /dev/da3h        Avail: 1549/2573 MB (60%)
D b                     State: up       Device /dev/da4h        Avail: 2061/2573 MB (80%)

   V myvol                 State: up       Plexes:       1 Size:        512 MB
   V mirror                State: up       Plexes:       2 Size:        512 MB

   P myvol.p0            C State: up       Subdisks:     1 Size:        512 MB
   P mirror.p0           C State: up       Subdisks:     1 Size:        512 MB
   P mirror.p1           C State: initializing     Subdisks:     1 Size:        512 MB

   S myvol.p0.s0           State: up       PO:        0  B Size:        512 MB
S mirror.p0.s0          State: up       PO:        0  B Size:        512 MB
S mirror.p1.s0          State: empty    PO:        0  B Size:        512 MB
```

Figure 20-5 shows the structure graphically.

Figure 20-5. A Mirrored Vinum Volume



In this example, each plex contains the full 512 MB of address space. As in the previous example, each plex contains only a single subdisk.

20.6.3 Optimizing Performance

The mirrored volume in the previous example is more resistant to failure than an unmirrored volume, but its performance is less: each write to the volume requires a write to both drives, using up a greater proportion of the total disk bandwidth. Performance considerations demand a different approach: instead of mirroring, the data is striped across as many disk drives as possible. The following configuration shows a volume with a plex striped across four disk drives:

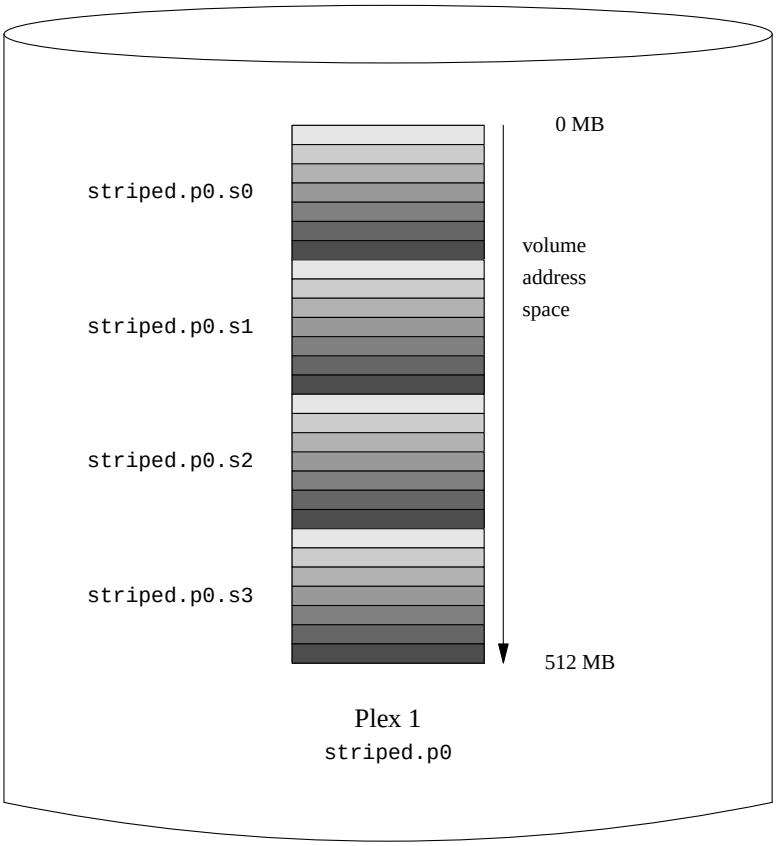
```
drive c device /dev/da5h
drive d device /dev/da6h
volume stripe
plex org striped 512k
  sd length 128m drive a
```

```
sd length 128m drive b
sd length 128m drive c
sd length 128m drive d
```

As before, it is not necessary to define the drives which are already known to Vinum. After processing this definition, the configuration looks like:

```
Drives:         4 (4 configured)
Volumes:        3 (4 configured)
Plexes:         4 (8 configured)
Subdisks:       7 (16 configured)

   D a                  State: up       Device /dev/da3h        Avail: 1421/2573 MB (55%)
   D b                  State: up       Device /dev/da4h        Avail: 1933/2573 MB (75%)
   D c                  State: up       Device /dev/da5h        Avail: 2445/2573 MB (95%)
   D d                  State: up       Device /dev/da6h        Avail: 2445/2573 MB (95%)

   V myvol              State: up       Plexes:       1 Size:        512 MB
   V mirror             State: up       Plexes:       2 Size:        512 MB
   V striped            State: up       Plexes:       1 Size:        512 MB

   P myvol.p0           C State: up     Subdisks:     1 Size:        512 MB
   P mirror.p0          C State: up     Subdisks:     1 Size:        512 MB
   P mirror.p1          C State: initializing     Subdisks:     1 Size:        512 MB
   P striped.p1           State: up     Subdisks:     1 Size:        512 MB

   S myvol.p0.s0          State: up     PO:        0  B Size:        512 MB
   S mirror.p0.s0         State: up     PO:        0  B Size:        512 MB
   S mirror.p1.s0         State: empty  PO:        0  B Size:        512 MB
   S striped.p0.s0        State: up     PO:        0  B Size:        128 MB
   S striped.p0.s1        State: up     PO:      512 kB Size:        128 MB
   S striped.p0.s2        State: up     PO:     1024 kB Size:        128 MB
   S striped.p0.s3        State: up     PO:     1536 kB Size:        128 MB
```

Figure 20-6. A Striped Vinum Volume



This volume is represented in Figure 20-6. The darkness of the stripes indicates the position within the plex address space: the lightest stripes come first, the darkest last.

20.6.4 Resilience and Performance

With sufficient hardware, it is possible to build volumes which show both increased resilience and increased performance compared to standard UNIX partitions. A typical configuration file might be:

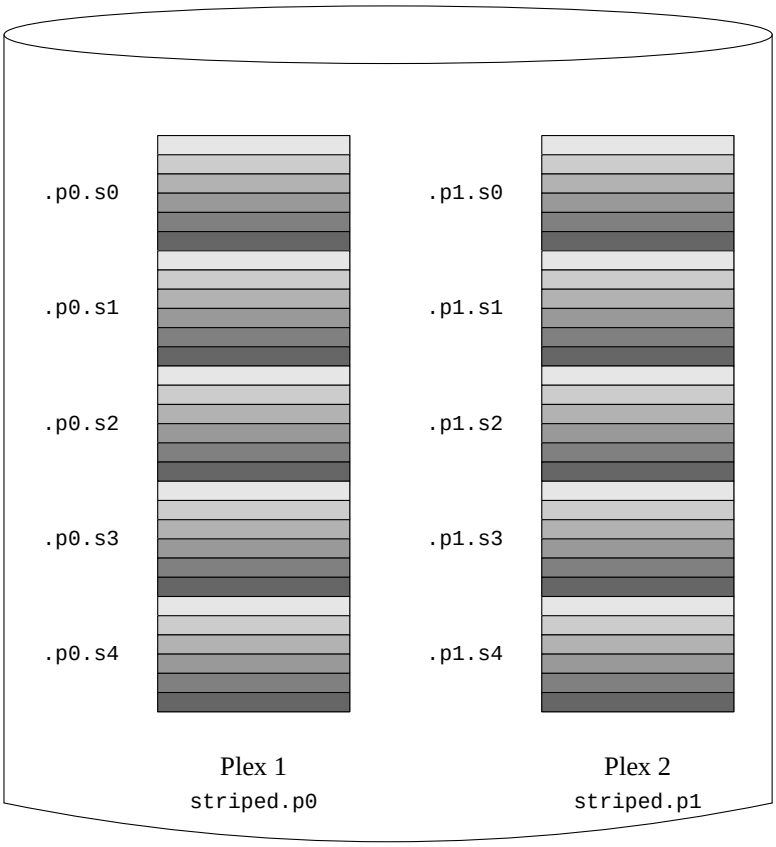
```
volume raid10
     plex org striped 512k
       sd length 102480k drive a
       sd length 102480k drive b
       sd length 102480k drive c
       sd length 102480k drive d
       sd length 102480k drive e
     plex org striped 512k
       sd length 102480k drive c
```

```
sd length 102480k drive d
sd length 102480k drive e
sd length 102480k drive a
sd length 102480k drive b
```

The subdisks of the second plex are offset by two drives from those of the first plex: this helps ensure that writes do not go to the same subdisks even if a transfer goes over two drives.

Figure 20-7 represents the structure of this volume.

Figure 20-7. A Mirrored, Striped Vinum Volume



20.7 Object Naming

As described above, Vinum assigns default names to plexes and subdisks, although they may be overridden. Overriding the default names is not recommended: experience with the VERITAS volume manager, which allows

arbitrary naming of objects, has shown that this flexibility does not bring a significant advantage, and it can cause confusion.

Names may contain any non-blank character, but it is recommended to restrict them to letters, digits and the underscore characters. The names of volumes, plexes and subdisks may be up to 64 characters long, and the names of drives may be up to 32 characters long.

Vinum objects are assigned device nodes in the hierarchy `/dev/vinum`. The configuration shown above would cause Vinum to create the following device nodes:

- The control devices `/dev/vinum/control` and `/dev/vinum/controld`, which are used by vinum(8) and the Vinum daemon respectively.
- Block and character device entries for each volume. These are the main devices used by Vinum. The block device names are the name of the volume, while the character device names follow the BSD tradition of prepending the letter *r* to the name. Thus the configuration above would include the block devices `/dev/vinum/myvol`, `/dev/vinum/mirror`, `/dev/vinum/striped`, `/dev/vinum/raid5` and `/dev/vinum/raid10`, and the character devices `/dev/vinum/rmyvol`, `/dev/vinum/rmirror`, `/dev/vinum/rstriped`, `/dev/vinum/rraid5` and `/dev/vinum/rraid10`. There is obviously a problem here: it is possible to have two volumes called *r* and *rr*, but there will be a conflict creating the device node `/dev/vinum/rr`: is it a character device for volume *r* or a block device for volume *rr*? Currently Vinum does not address this conflict: the first-defined volume will get the name.
- A directory `/dev/vinum/drive` with entries for each drive. These entries are in fact symbolic links to the corresponding disk nodes.
- A directory `/dev/vinum/volume` with entries for each volume. It contains subdirectories for each plex, which in turn contain subdirectories for their component subdisks.
- The directories `/dev/vinum/plex`, `/dev/vinum/sd`, and `/dev/vinum/rsd`, which contain block device nodes for each plex and block and character device nodes respectively for each subdisk.

For example, consider the following configuration file:

```
drive drive1 device /dev/sd1h
drive drive2 device /dev/sd2h
drive drive3 device /dev/sd3h
drive drive4 device /dev/sd4h
   volume s64 setupstate
     plex org striped 64k
       sd length 100m drive drive1
       sd length 100m drive drive2
       sd length 100m drive drive3
       sd length 100m drive drive4
```

After processing this file, vinum(8) creates the following structure in `/dev/vinum`:

```
brwx------  1 root  wheel   25, 0x40000001 Apr 13 16:46 Control
brwx------  1 root  wheel   25, 0x40000002 Apr 13 16:46 control
brwx------  1 root  wheel   25, 0x40000000 Apr 13 16:46 controld
drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 drive
drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 plex
crwxr-xr--  1 root  wheel   91,   2 Apr 13 16:46 rs64
drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 rsd
```

```
drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 rvol
brwxr-xr--  1 root  wheel   25,   2 Apr 13 16:46 s64
drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 sd
drwxr-xr-x  3 root  wheel       512 Apr 13 16:46 vol

/dev/vinum/drive:
   total 0
   lrwxr-xr-x  1 root  wheel  9 Apr 13 16:46 drive1 -> /dev/sd1h
   lrwxr-xr-x  1 root  wheel  9 Apr 13 16:46 drive2 -> /dev/sd2h
   lrwxr-xr-x  1 root  wheel  9 Apr 13 16:46 drive3 -> /dev/sd3h
   lrwxr-xr-x  1 root  wheel  9 Apr 13 16:46 drive4 -> /dev/sd4h

   /dev/vinum/plex:
   total 0
   brwxr-xr--  1 root  wheel   25, 0x10000002 Apr 13 16:46 s64.p0

   /dev/vinum/rsd:
   total 0
   crwxr-xr--  1 root  wheel   91, 0x20000002 Apr 13 16:46 s64.p0.s0
   crwxr-xr--  1 root  wheel   91, 0x20100002 Apr 13 16:46 s64.p0.s1
   crwxr-xr--  1 root  wheel   91, 0x20200002 Apr 13 16:46 s64.p0.s2
   crwxr-xr--  1 root  wheel   91, 0x20300002 Apr 13 16:46 s64.p0.s3

   /dev/vinum/rvol:
   total 0
   crwxr-xr--  1 root  wheel   91,   2 Apr 13 16:46 s64

   /dev/vinum/sd:
   total 0
   brwxr-xr--  1 root  wheel   25, 0x20000002 Apr 13 16:46 s64.p0.s0
   brwxr-xr--  1 root  wheel   25, 0x20100002 Apr 13 16:46 s64.p0.s1
   brwxr-xr--  1 root  wheel   25, 0x20200002 Apr 13 16:46 s64.p0.s2
   brwxr-xr--  1 root  wheel   25, 0x20300002 Apr 13 16:46 s64.p0.s3

   /dev/vinum/vol:
   total 1
   brwxr-xr--  1 root  wheel   25,   2 Apr 13 16:46 s64
   drwxr-xr-x  3 root  wheel       512 Apr 13 16:46 s64.plex

   /dev/vinum/vol/s64.plex:
   total 1
   brwxr-xr--  1 root  wheel   25, 0x10000002 Apr 13 16:46 s64.p0
   drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 s64.p0.sd

   /dev/vinum/vol/s64.plex/s64.p0.sd:
   total 0
   brwxr-xr--  1 root  wheel   25, 0x20000002 Apr 13 16:46 s64.p0.s0
   brwxr-xr--  1 root  wheel   25, 0x20100002 Apr 13 16:46 s64.p0.s1
   brwxr-xr--  1 root  wheel   25, 0x20200002 Apr 13 16:46 s64.p0.s2
   brwxr-xr--  1 root  wheel   25, 0x20300002 Apr 13 16:46 s64.p0.s3
```

Although it is recommended that plexes and subdisks should not be allocated specific names, Vinum drives must be named. This makes it possible to move a drive to a different location and still recognize it automatically. Drive names may be up to 32 characters long.

20.7.1 Creating File Systems

Volumes appear to the system to be identical to disks, with one exception. Unlike UNIX drives, Vinum does not partition volumes, which thus do not contain a partition table. This has required modification to some disk utilities, notably newfs(8), which previously tried to interpret the last letter of a Vinum volume name as a partition identifier. For example, a disk drive may have a name like `/dev/ad0a` or `/dev/da2h`. These names represent the first partition (`a`) on the first (0) IDE disk (`ad`) and the eighth partition (`h`) on the third (2) SCSI disk (`da`) respectively. By contrast, a Vinum volume might be called `/dev/vinum/concat`, a name which has no relationship with a partition name.

Normally, newfs(8) interprets the name of the disk and complains if it cannot understand it. For example:

```
# newfs /dev/vinum/concat
newfs: /dev/vinum/concat: can't figure out file system partition
```

Note: The following is only valid for FreeBSD versions prior to 5.0:

In order to create a file system on this volume, use the `-v` option to newfs(8):

```
# newfs -v /dev/vinum/concat
```

20.8 Configuring Vinum

The `GENERIC` kernel does not contain Vinum. It is possible to build a special kernel which includes Vinum, but this is not recommended. The standard way to start Vinum is as a kernel module (kld). You do not even need to use kldload(8) for Vinum: when you start vinum(8), it checks whether the module has been loaded, and if it is not, it loads it automatically.

20.8.1 Startup

Vinum stores configuration information on the disk slices in essentially the same form as in the configuration files. When reading from the configuration database, Vinum recognizes a number of keywords which are not allowed in the configuration files. For example, a disk configuration might contain the following text:

```
volume myvol state up
volume bigraid state down
plex name myvol.p0 state up org concat vol myvol
plex name myvol.p1 state up org concat vol myvol
plex name myvol.p2 state init org striped 512b vol myvol
plex name bigraid.p0 state initializing org raid5 512b vol bigraid
sd name myvol.p0.s0 drive a plex myvol.p0 state up len 1048576b driveoffset 265b plexoffset 0b
sd name myvol.p0.s1 drive b plex myvol.p0 state up len 1048576b driveoffset 265b plexoffset 1048576b
sd name myvol.p1.s0 drive c plex myvol.p1 state up len 1048576b driveoffset 265b plexoffset 0b
sd name myvol.p1.s1 drive d plex myvol.p1 state up len 1048576b driveoffset 265b plexoffset 1048576b
sd name myvol.p2.s0 drive a plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 0b
sd name myvol.p2.s1 drive b plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 524288b
sd name myvol.p2.s2 drive c plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 1048576b
sd name myvol.p2.s3 drive d plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 1572864b
sd name bigraid.p0.s0 drive a plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 0b
```

```
sd name bigraid.p0.s1 drive b plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 4194304b
sd name bigraid.p0.s2 drive c plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 8388608b
sd name bigraid.p0.s3 drive d plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 12582912b
sd name bigraid.p0.s4 drive e plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 16777216b
```

The obvious differences here are the presence of explicit location information and naming (both of which are also allowed, but discouraged, for use by the user) and the information on the states (which are not available to the user). Vinum does not store information about drives in the configuration information: it finds the drives by scanning the configured disk drives for partitions with a Vinum label. This enables Vinum to identify drives correctly even if they have been assigned different UNIX drive IDs.

20.8.1.1 Automatic Startup

In order to start Vinum automatically when you boot the system, ensure that you have the following line in your `/etc/rc.conf`:

```
start_vinum="YES"  # set to YES to start vinum
```

If you do not have a file `/etc/rc.conf`, create one with this content. This will cause the system to load the Vinum kld at startup, and to start any objects mentioned in the configuration. This is done before mounting file systems, so it is possible to automatically fsck(8) and mount file systems on Vinum volumes.

When you start Vinum with the `vinum start` command, Vinum reads the configuration database from one of the Vinum drives. Under normal circumstances, each drive contains an identical copy of the configuration database, so it does not matter which drive is read. After a crash, however, Vinum must determine which drive was updated most recently and read the configuration from this drive. It then updates the configuration if necessary from progressively older drives.

20.9 Using Vinum for the Root Filesystem

For a machine that has fully-mirrored filesystems using Vinum, it is desirable to also mirror the root filesystem. Setting up such a configuration is less trivial than mirroring an arbitrary filesystem because:

- The root filesystem must be available very early during the boot process, so the Vinum infrastructure must already be available at this time.
- The volume containing the root filesystem also contains the system bootstrap and the kernel, which must be read using the host system's native utilities (e. g. the BIOS on PC-class machines) which often cannot be taught about the details of Vinum.

In the following sections, the term "root volume" is generally used to describe the Vinum volume that contains the root filesystem. It is probably a good idea to use the name `"root"` for this volume, but this is not technically required in any way. All command examples in the following sections assume this name though.

20.9.1 Starting up Vinum Early Enough for the Root Filesystem

There are several measures to take for this to happen:

- Vinum must be available in the kernel at boot-time. Thus, the method to start Vinum automatically described in Section 20.8.1.1 is not applicable to accomplish this task, and the `start_vinum` parameter must actually *not* be set when the following setup is being arranged. The first option would be to compile Vinum statically into the kernel, so it is available all the time, but this is usually not desirable. There is another option as well, to have `/boot/loader` (Section 12.3.3) load the vinum kernel module early, before starting the kernel. This can be accomplished by putting the line:

  ```
  vinum_load="YES"
  ```

  into the file `/boot/loader.conf`.

- Vinum must be initialized early since it needs to supply the volume for the root filesystem. By default, the Vinum kernel part is not looking for drives that might contain Vinum volume information until the administrator (or one of the startup scripts) issues a `vinum start` command.

  > **Note:** The following paragraphs are outlining the steps needed for FreeBSD 5.X and above. The setup required for FreeBSD 4.X differs, and is described below in Section 20.9.5.

  By placing the line:

  ```
  vinum.autostart="YES"
  ```

  into `/boot/loader.conf`, Vinum is instructed to automatically scan all drives for Vinum information as part of the kernel startup.

  Note that it is not necessary to instruct the kernel where to look for the root filesystem. `/boot/loader` looks up the name of the root device in `/etc/fstab`, and passes this information on to the kernel. When it comes to mount the root filesystem, the kernel figures out from the device name provided which driver to ask to translate this into the internal device ID (major/minor number).

20.9.2 Making a Vinum-based Root Volume Accessible to the Bootstrap

Since the current FreeBSD bootstrap is only 7.5 KB of code, and already has the burden of reading files (like `/boot/loader`) from the UFS filesystem, it is sheer impossible to also teach it about internal Vinum structures so it could parse the Vinum configuration data, and figure out about the elements of a boot volume itself. Thus, some tricks are necessary to provide the bootstrap code with the illusion of a standard `"a"` partition that contains the root filesystem.

For this to be possible at all, the following requirements must be met for the root volume:

- The root volume must not be striped or RAID-5.
- The root volume must not contain more than one concatenated subdisk per plex.

Note that it is desirable and possible that there are multiple plexes, each containing one replica of the root filesystem. The bootstrap process will, however, only use one of these replica for finding the bootstrap and all the files, until the kernel will eventually mount the root filesystem itself. Each single subdisk within these plexes will then need its own `"a"` partition illusion, for the respective device to become bootable. It is not strictly needed that each of these faked `"a"` partitions is located at the same offset within its device, compared with other devices containing plexes of the root volume. However, it is probably a good idea to create the Vinum volumes that way so the resulting mirrored devices are symmetric, to avoid confusion.

In order to set up these "a" partitions, for each device containing part of the root volume, the following needs to be done:

1. The location (offset from the beginning of the device) and size of this device's subdisk that is part of the root volume need to be examined, using the command:

   ```
   # vinum l -rv root
   ```

   Note that Vinum offsets and sizes are measured in bytes. They must be divided by 512 in order to obtain the block numbers that are to be used in the `disklabel` command.

2. Run the command:

   ```
   # disklabel -e devname
   ```

   for each device that participates in the root volume. *devname* must be either the name of the disk (like `da0`) for disks without a slice (aka. fdisk) table, or the name of the slice (like `ad0s1`).

   If there is already an "a" partition on the device (presumably, containing a pre-Vinum root filesystem), it should be renamed to something else, so it remains accessible (just in case), but will no longer be used by default to bootstrap the system. Note that active partitions (like a root filesystem currently mounted) cannot be renamed, so this must be executed either when being booted from a "Fixit" medium, or in a two-step process, where (in a mirrored situation) the disk that has not been currently booted is being manipulated first.

   Then, the offset the Vinum partition on this device (if any) must be added to the offset of the respective root volume subdisk on this device. The resulting value will become the "offset" value for the new "a" partition. The "size" value for this partition can be taken verbatim from the calculation above. The "fstype" should be `4.2BSD`. The "fsize", "bsize", and "cpg" values should best be chosen to match the actual filesystem, though they are fairly unimportant within this context.

   That way, a new "a" partition will be established that overlaps the Vinum partition on this device. Note that the `disklabel` will only allow for this overlap if the Vinum partition has properly been marked using the "vinum" fstype.

3. That's all! A faked "a" partition does exist now on each device that has one replica of the root volume. It is highly recommendable to verify the result again, using a command like:

   ```
   # fsck -n /dev/devnamea
   ```

It should be remembered that all files containing control information must be relative to the root filesystem in the Vinum volume which, when setting up a new Vinum root volume, might not match the root filesystem that is currently active. So in particular, the files `/etc/fstab` and `/boot/loader.conf` need to be taken care of.

At next reboot, the bootstrap should figure out the appropriate control information from the new Vinum-based root filesystem, and act accordingly. At the end of the kernel initialization process, after all devices have been announced, the prominent notice that shows the success of this setup is a message like:

```
Mounting root from ufs:/dev/vinum/root
```

20.9.3 Example of a Vinum-based Root Setup

After the Vinum root volume has been set up, the output of `vinum l -rv root` could look like:

```
...
Subdisk root.p0.s0:
```

```
  Size:        125829120 bytes (120 MB)
  State: up
  Plex root.p0 at offset 0 (0  B)
  Drive disk0 (/dev/da0h) at offset 135680 (132 kB)

Subdisk root.p1.s0:
  Size:        125829120 bytes (120 MB)
  State: up
  Plex root.p1 at offset 0 (0  B)
  Drive disk1 (/dev/da1h) at offset 135680 (132 kB)
```

The values to note are `135680` for the offset (relative to partition `/dev/da0h`). This translates to 265 512-byte disk blocks in `disklabel`'s terms. Likewise, the size of this root volume is 245760 512-byte blocks. `/dev/da1h`, containing the second replica of this root volume, has a symmetric setup.

The disklabel for these devices might look like:

```
...
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  a:   245760      281    4.2BSD     2048 16384     0   # (Cyl.    0*- 15*)
  c: 71771688        0    unused        0     0         # (Cyl.    0 - 4467*)
  h: 71771672       16     vinum                        # (Cyl.    0*- 4467*)
```

It can be observed that the `"size"` parameter for the faked `"a"` partition matches the value outlined above, while the `"offset"` parameter is the sum of the offset within the Vinum partition `"h"`, and the offset of this partition within the device (or slice). This is a typical setup that is necessary to avoid the problem described in Section 20.9.4.3. It can also be seen that the entire `"a"` partition is completely within the `"h"` partition containing all the Vinum data for this device.

Note that in the above example, the entire device is dedicated to Vinum, and there is no leftover pre-Vinum root partition, since this has been a newly set-up disk that was only meant to be part of a Vinum configuration, ever.

20.9.4 Troubleshooting

If something goes wrong, a way is needed to recover from the situation. The following list contains few known pitfalls and solutions.

20.9.4.1 System Bootstrap Loads, but System Does Not Boot

If for any reason the system does not continue to boot, the bootstrap can be interrupted with by pressing the **space** key at the 10-seconds warning. The loader variables (like `vinum.autostart`) can be examined using the `show`, and manipulated using `set` or `unset` commands.

If the only problem was that the Vinum kernel module was not yet in the list of modules to load automatically, a simple `load vinum` will help.

When ready, the boot process can be continued with a `boot -as`. The options `-as` will request the kernel to ask for the root filesystem to mount (`-a`), and make the boot process stop in single-user mode (`-s`), where the root filesystem

is mounted read-only. That way, even if only one plex of a multi-plex volume has been mounted, no data inconsistency between plexes is being risked.

At the prompt asking for a root filesystem to mount, any device that contains a valid root filesystem can be entered. If `/etc/fstab` had been set up correctly, the default should be something like `ufs:/dev/vinum/root`. A typical alternate choice would be something like `ufs:da0d` which could be a hypothetical partition that contains the pre-Vinum root filesystem. Care should be taken if one of the alias `"a"` partitions are entered here that are actually reference to the subdisks of the Vinum root device, because in a mirrored setup, this would only mount one piece of a mirrored root device. If this filesystem is to be mounted read-write later on, it is necessary to remove the other plex(es) of the Vinum root volume since these plexes would otherwise carry inconsistent data.

20.9.4.2 Only Primary Bootstrap Loads

If `/boot/loader` fails to load, but the primary bootstrap still loads (visible by a single dash in the left column of the screen right after the boot process starts), an attempt can be made to interrupt the primary bootstrap at this point, using the **space** key. This will make the bootstrap stop in stage two, see Section 12.3.2. An attempt can be made here to boot off an alternate partition, like the partition containing the previous root filesystem that has been moved away from `"a"` above.

20.9.4.3 Nothing Boots, the Bootstrap Panics

This situation will happen if the bootstrap had been destroyed by the Vinum installation. Unfortunately, Vinum accidentally currently leaves only 4 KB at the beginning of its partition free before starting to write its Vinum header information. However, the stage one and two bootstraps plus the disklabel embedded between them currently require 8 KB. So if a Vinum partition was started at offset 0 within a slice or disk that was meant to be bootable, the Vinum setup will trash the bootstrap.

Similarly, if the above situation has been recovered, for example by booting from a "Fixit" medium, and the bootstrap has been re-installed using `disklabel -B` as described in Section 12.3.2, the bootstrap will trash the Vinum header, and Vinum will no longer find its disk(s). Though no actual Vinum configuration data or data in Vinum volumes will be trashed by this, and it would be possible to recover all the data by entering exact the same Vinum configuration data again, the situation is hard to fix at all. It would be necessary to move the entire Vinum partition by at least 4 KB off, in order to have the Vinum header and the system bootstrap no longer collide.

20.9.5 Differences for FreeBSD 4.X

Under FreeBSD 4.X, some internal functions required to make Vinum automatically scan all disks are missing, and the code that figures out the internal ID of the root device is not smart enough to handle a name like `/dev/vinum/root` automatically. Therefore, things are a little different here.

Vinum must explicitly be told which disks to scan, using a line like the following one in `/boot/loader.conf`:

```
vinum.drives="/dev/da0 /dev/da1"
```

It is important that all drives are mentioned that could possibly contain Vinum data. It does not harm if *more* drives are listed, nor is it necessary to add each slice and/or partition explicitly, since Vinum will scan all slices and partitions of the named drives for valid Vinum headers.

Since the routines used to parse the name of the root filesystem, and derive the device ID (major/minor number) are only prepared to handle "classical" device names like `/dev/ad0s1a`, they cannot make any sense out of a root volume name like `/dev/vinum/root`. For that reason, Vinum itself needs to pre-setup the internal kernel parameter that holds the ID of the root device during its own initialization. This is requested by passing the name of the root volume in the loader variable `vinum.root`. The entry in `/boot/loader.conf` to accomplish this looks like:

```
vinum.root="root"
```

Now, when the kernel initialization tries to find out the root device to mount, it sees whether some kernel module has already pre-initialized the kernel parameter for it. If that is the case, *and* the device claiming the root device matches the major number of the driver as figured out from the name of the root device string being passed (that is, `"vinum"` in our case), it will use the pre-allocated device ID, instead of trying to figure out one itself. That way, during the usual automatic startup, it can continue to mount the Vinum root volume for the root filesystem.

However, when `boot -a` has been requesting to ask for entering the name of the root device manually, it must be noted that this routine still cannot actually parse a name entered there that refers to a Vinum volume. If any device name is entered that does not refer to a Vinum device, the mismatch between the major numbers of the pre-allocated root parameter and the driver as figured out from the given name will make this routine enter its normal parser, so entering a string like `ufs:da0d` will work as expected. Note that if this fails, it is however no longer possible to re-enter a string like `ufs:vinum/root` again, since it cannot be parsed. The only way out is to reboot again, and start over then. (At the "askroot" prompt, the initial `/dev/` can always be omitted.)

Notes

1. RAID stands for *Redundant Array of Inexpensive Disks* and offers various forms of fault tolerance, though the latter term is somewhat misleading: it provides no redundancy.

Chapter 21 Virtualization(虛擬機器)

Contributed by Murray Stokely.

21.1 Synopsis

虛擬機器軟體可以讓同一台機器得以同時執行多種作業系統。在PC 上，通常這類系統都是在宿主(host)機器上裝虛擬機器軟體，來跑一堆guest OS 。

讀完這章，您將了解：

- host OS 以及guest OS 的區別。
- 如何在搭載Intel CPU 的Apple® Macintosh 電腦上安裝FreeBSD 。
- 如何在Linux 上以**Xen**™ 來安裝FreeBSD。
- 如何在Microsoft Windows 上以**Virtual PC** 安裝FreeBSD。
- 如何在虛擬機器對FreeBSD 系統作性能調校，以取得最佳效能。

在開始閱讀這章之前，您需要：

- 瞭解UNIX 及FreeBSD 相關基本概念(Chapter 3)。
- 知道如何安裝FreeBSD(Chapter 2)。
- 知道如何設定網路(Chapter 29)。
- 知道如何以ports/packages 來安裝應用程式(Chapter 4)。

21.2 安裝FreeBSD 為Guest OS

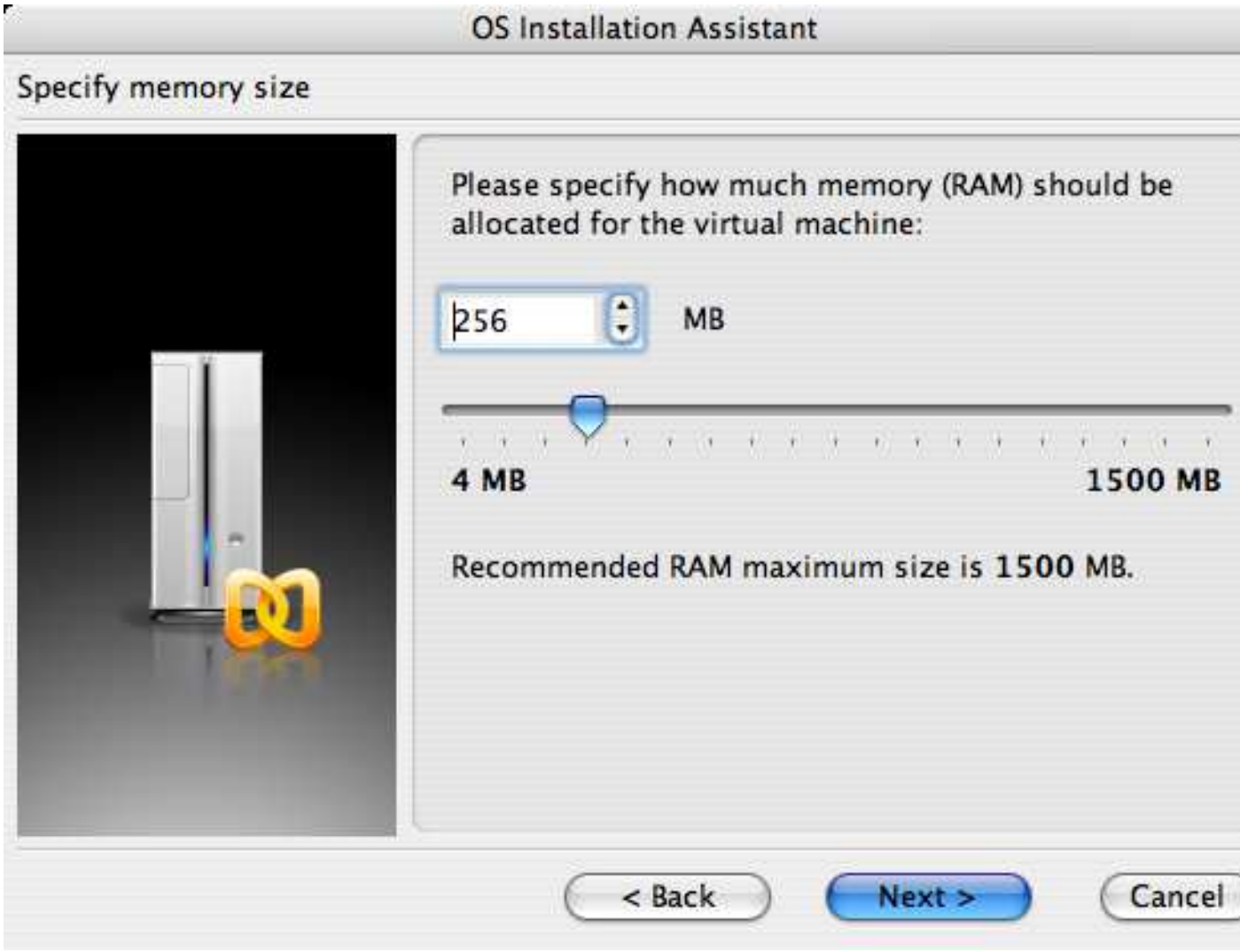
21.2.1 MacOS 上的Parallels

Mac 版的**Parallels Desktop** 乃是可用於搭配Intel CPU 以及Mac OS 10.4.6 以上的Apple Mac 電腦的商業軟體。FreeBSD 是其有完整支援的guest OS 之一。在Mac OS X 裝好**Parallels** 後，必須針對所欲安裝的guest OS 來作相關的虛擬機器設定。

21.2.1.1 在Parallels/Mac OS® X 上安裝FreeBSD

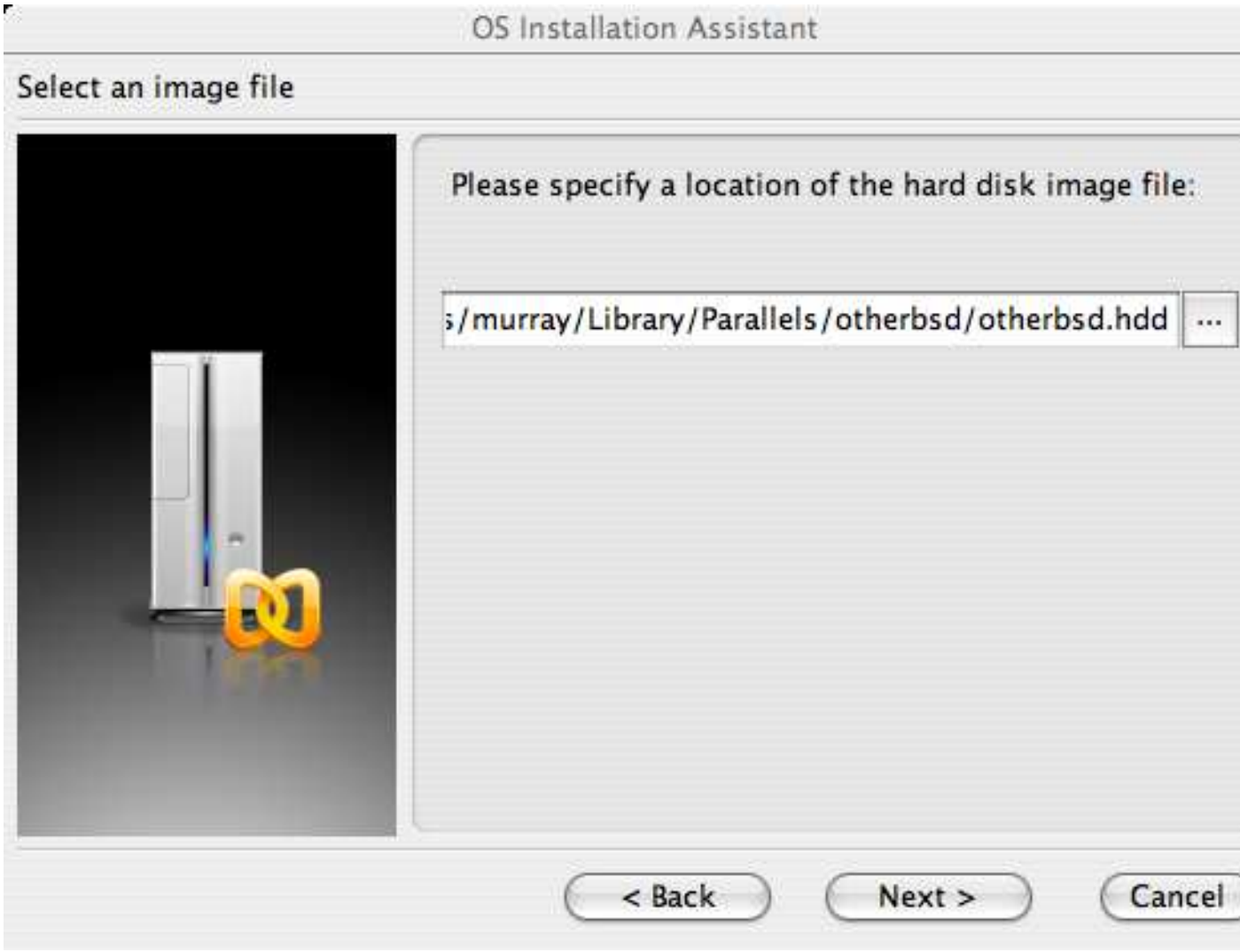
在Mac OS X/**Parallels** 上安裝FreeBSD 的第一步是新增虛擬機器。如下所示，在提示視窗內請將Guest OS Type 勾選為FreeBSD：

並依據自身需求來規劃硬碟容量跟記憶體的分配。對大多數在**Parallels** 使用的情況而言，大約4GB 硬碟以及512MB RAM 就夠用了：

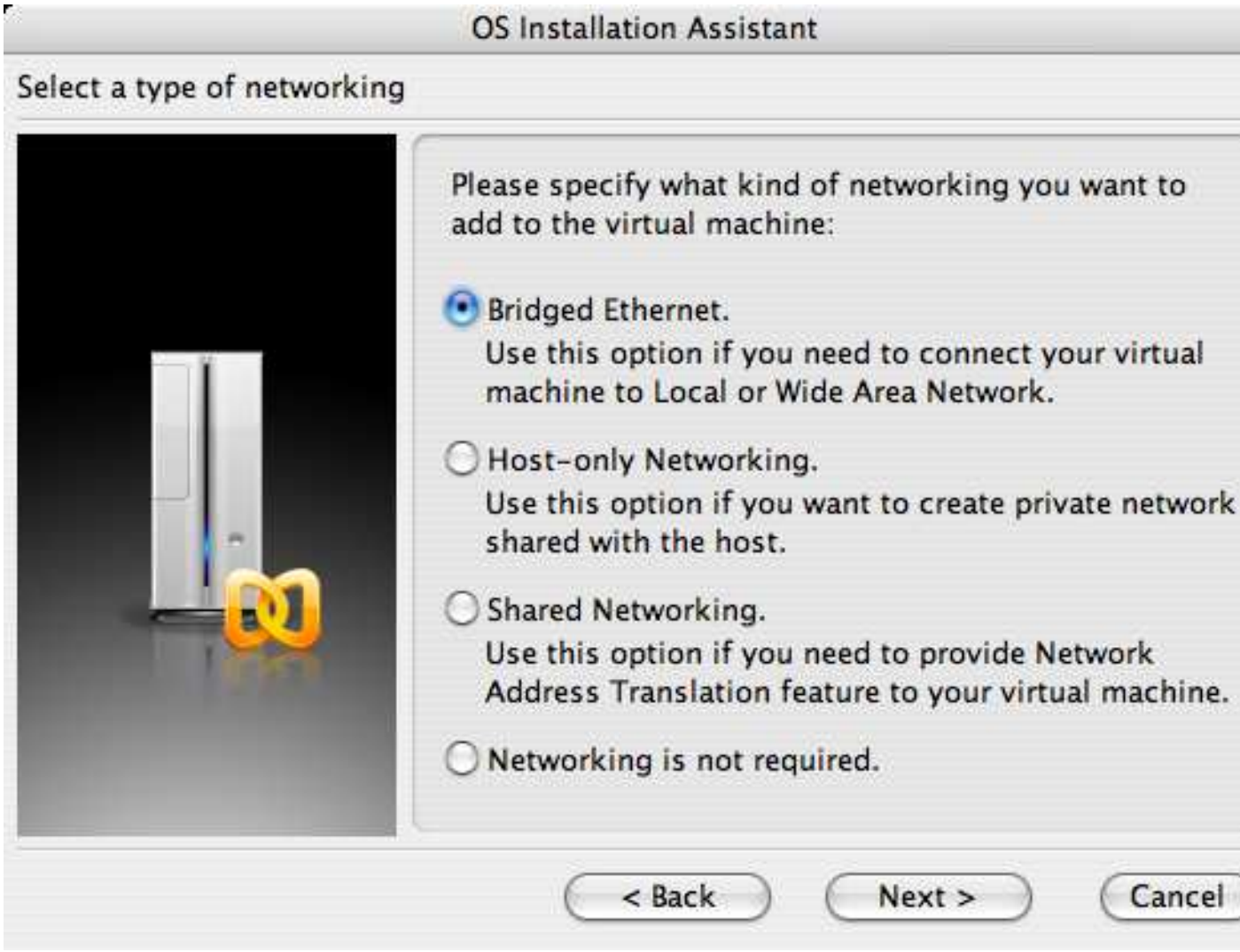

OS Installation Assistant
Specify memory size
Please specify how much memory (RAM) should be allocated for the virtual machine:
256
MB
4 MB
1500 MB
Recommended RAM maximum size is 1500 MB.
< Back
Next >
Cancel

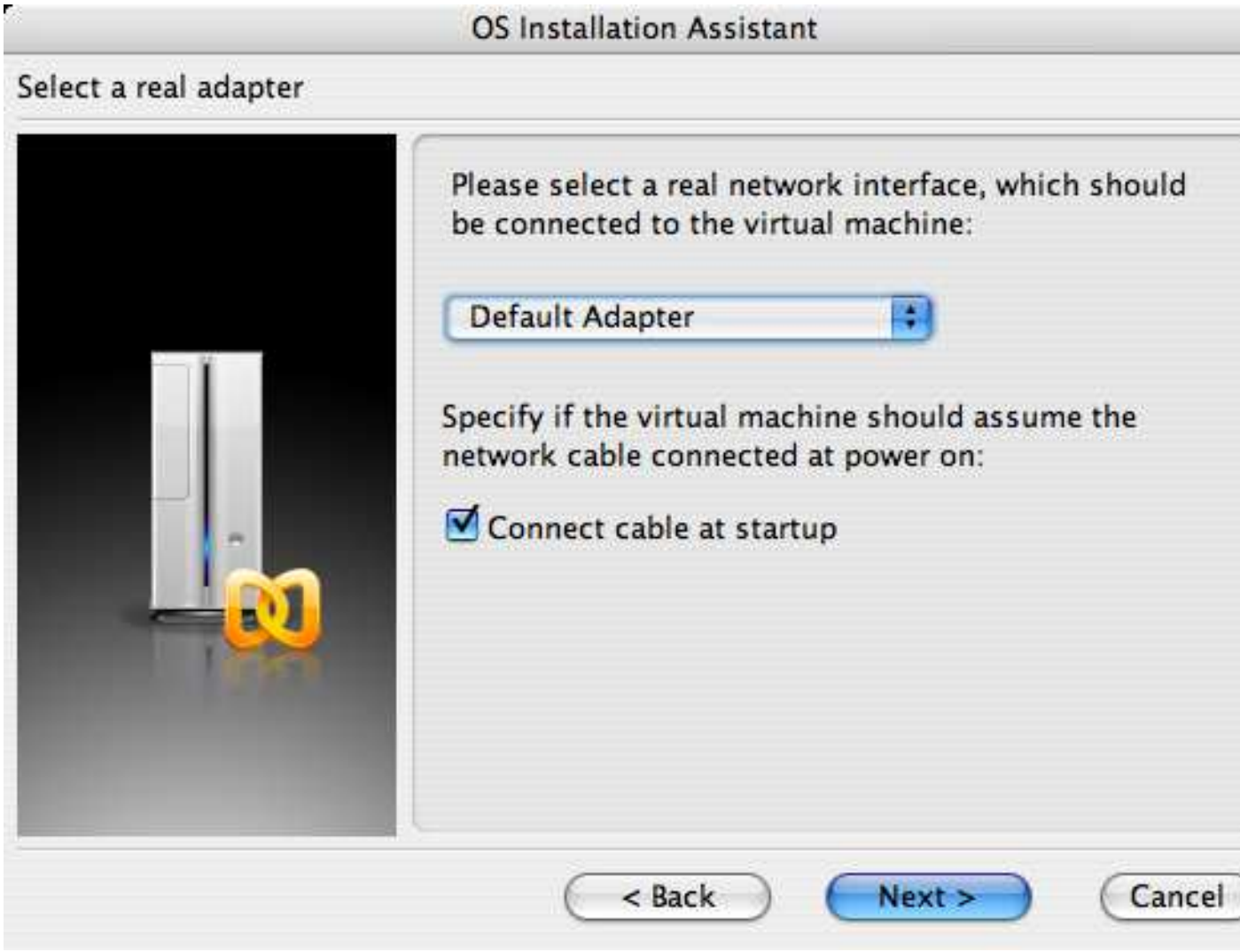

OS Installation Assistant
Select action type
Please specify what kind of hard disk you want to install to the virtual machine. If you do not want to add a hard disk now, select "Do not add hard disk" option. You will be able to add it later using a Configuration Editor.
Create a new virtual hard disk
Use an existing hard disk image
Do not add hard disk
< Back
Next >
Cancel


OS Installation Assistant
Specify hard disk options
Please specify a preferred virtual disk size:
8000
MB
Select disk format:
Expanding (recommended).
Disk image file is small initially and grows as you add more data to the virtual machine. This disk format takes less time to create and saves disk space on the host.
Plain.
Disk image file consumes all the allocated space right from the start. It takes more time to create but allows guest OS to operate faster.
< Back
Next >
Cancel

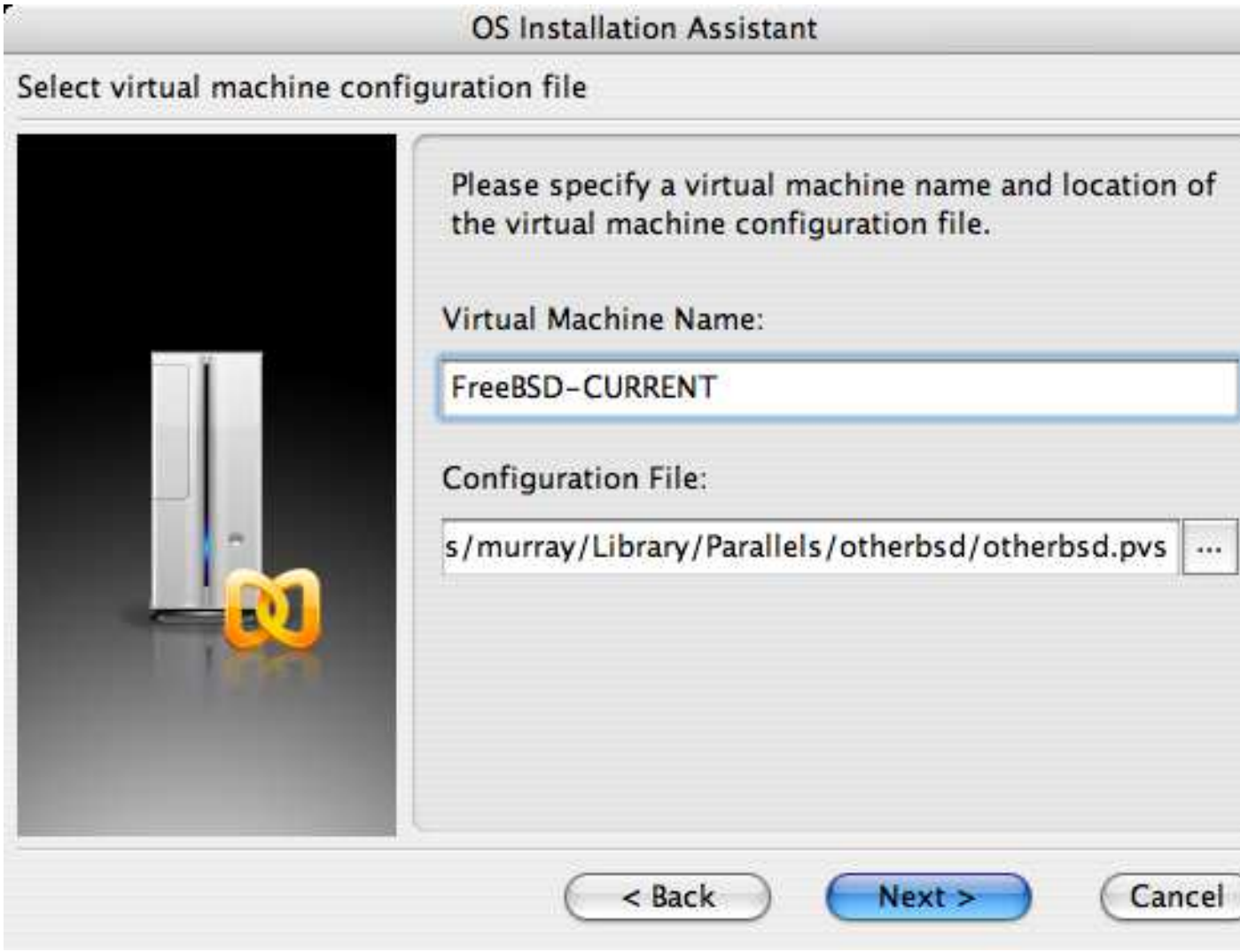


接下來，選擇網路種類以及網路卡：

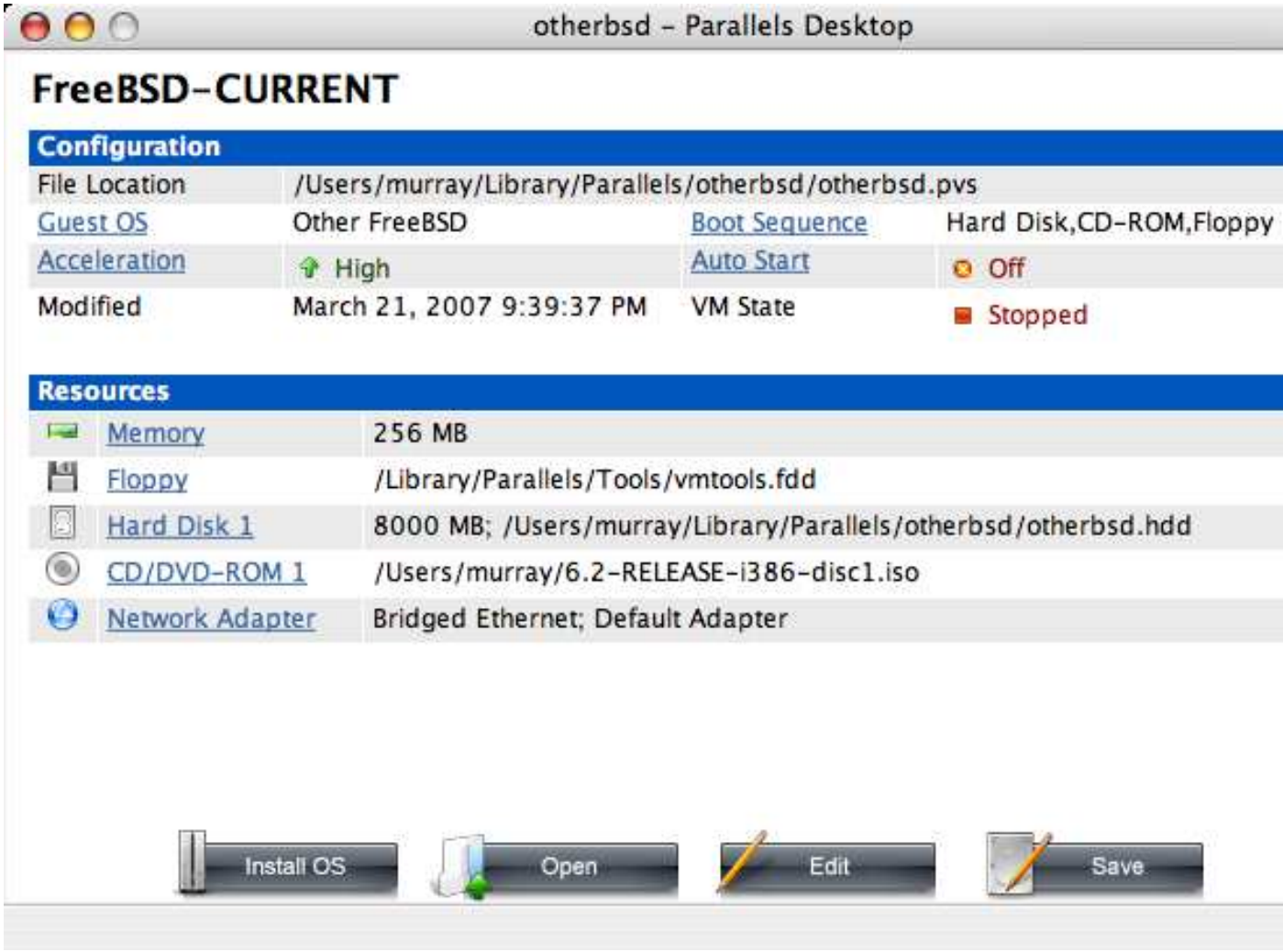

OS Installation Assistant
Select a type of networking
Please specify what kind of networking you want to add to the virtual machine:
Bridged Ethernet.
Use this option if you need to connect your virtual machine to Local or Wide Area Network.
Host-only Networking.
Use this option if you want to create private network shared with the host.
Shared Networking.
Use this option if you need to provide Network Address Translation feature to your virtual machine.
Networking is not required.
< Back
Next >
Cancel



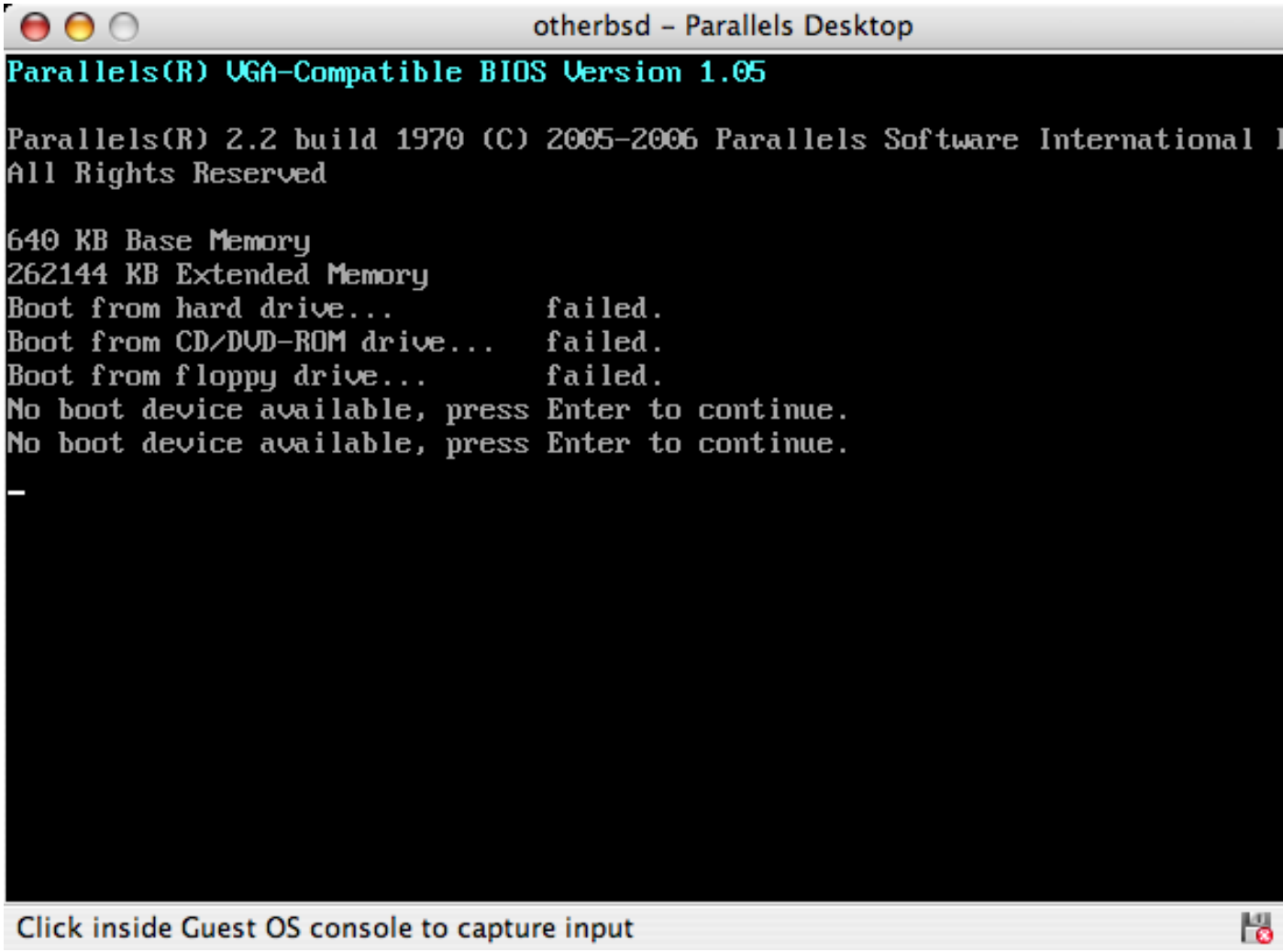
最後，儲存設定檔就完成設定了：


OS Installation Assistant
Select virtual machine configuration file
Please specify a virtual machine name and location of the virtual machine configuration file.
Virtual Machine Name:
FreeBSD-CURRENT
Configuration File:
s/murray/Library/Parallels/otherbsd/otherbsd.pvs
...
< Back
Next >
Cancel

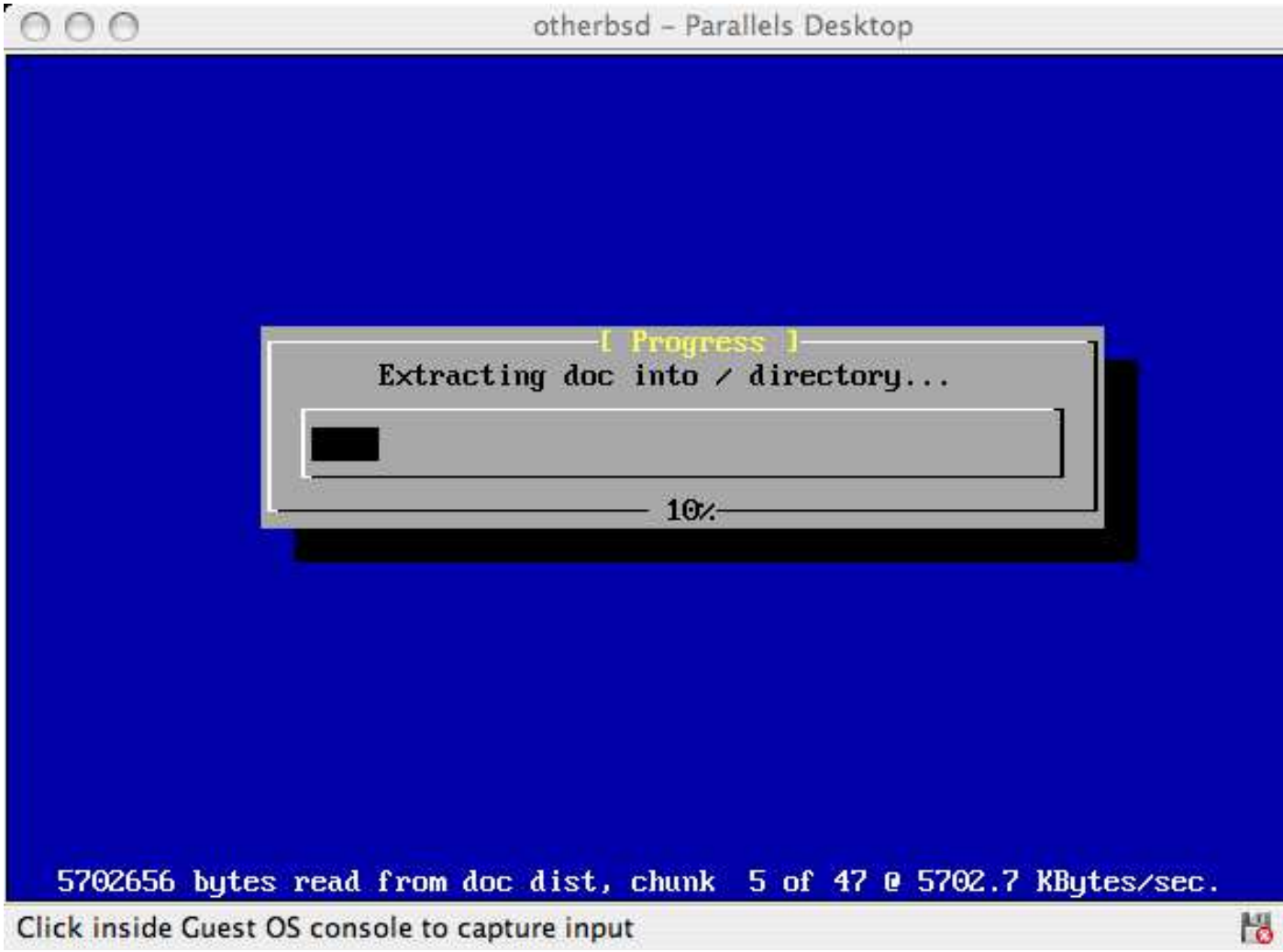
在FreeBSD 虛擬機器新增後，就可以繼續以其安裝FreeBSD。安裝方面，比較好的作法是使用官方的FreeBSD 光碟或者從官方FTP 站下載ISO image 檔。若您的Mac 本機已經有該ISO 檔，或者Mac 的光碟機內有放安裝片，那麼就可以在FreeBSD 的**Parallels** 視窗右下角按下光碟片圖示。接著會出現一個視窗，可以把虛擬機器內的光碟機設定到該ISO 檔，或者是實體光碟機。



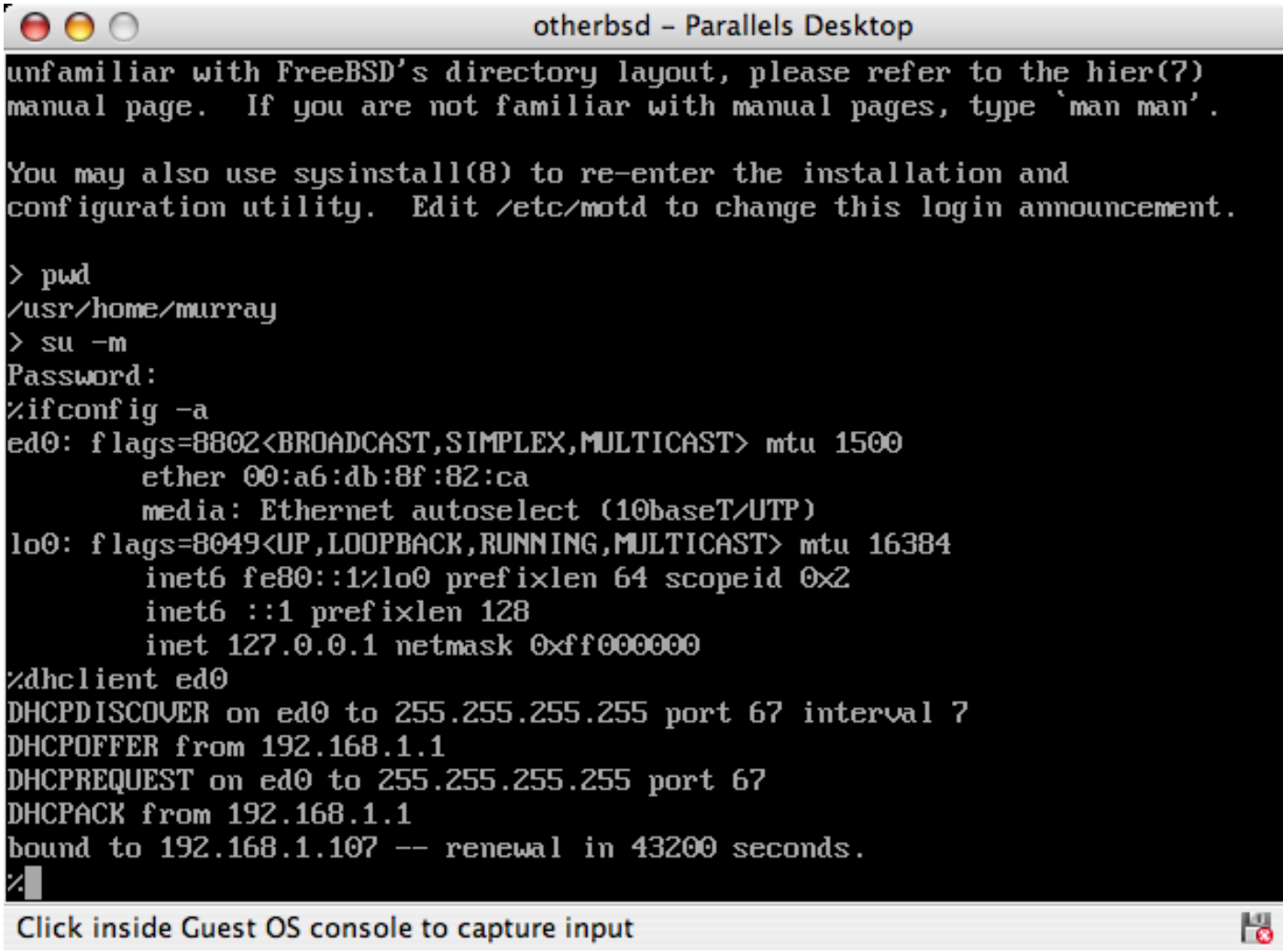
設好光碟片來源之後，就可以按下重開機圖示以重開FreeBSD 虛擬機器。**Parallels** 會以特殊BIOS 開機，並與普通的BIOS 一樣會先檢查是否有光碟機。



此時，它就會找到FreeBSD 安裝片，並開始在Chapter 2 內所介紹到的**sysinstall** 安裝過程。這時候也可順便裝X11，但先不要進行相關設定。



完成安裝過程之後，就可以重開剛裝的FreeBSD 虛擬機器。



21.2.1.2 在Mac OS X/Parallels 上設定FreeBSD

把FreeBSD 成功裝到Mac OS X 的**Parallels** 之後，還需要作一些設定步驟，以便將虛擬機器內的FreeBSD 最佳化。

1. 設定boot loader 參數

   最重要的步驟乃是藉由調降`kern.hz` 來降低**Parallels** 環境內FreeBSD 的CPU 佔用率。可以在`/boot/loader.conf` 內加上下列設定即可：

   ```
   kern.hz=100
   ```

   若不作這設定，那麼光是idle 狀態的FreeBSD (**Parallels** guest OS) 就會在僅單一處理器的iMac® 上佔了大約15% 的CPU 佔用率。作上述修改之後，佔用率就會降至大約5%。

2. 設定新的kernel 設定檔

   可以放心把所有SCSI、FireWire、USB 相關設備都移除。**Parallels** 有提供ed(4) 的虛擬網卡，因此，除了ed(4) 以及miibus(4) 以外的其他網路卡也都可以從kernel 中移除。

3. 設定網路

   可以替虛擬機器簡單用DHCP 來設定與Mac 相同的LAN 網路環境，只要在`/etc/rc.conf` 內加上`ifconfig_ed0="DHCP"` 即可完成。其他進階的網路設定方式，請參考Chapter 29。

21.2.2 在Linux 透過Xen™ 跑FreeBSD

Contributed by Fukang Chen (Loader).

Xen hypervisor 乃是開放源碼的paravirtualization 產品，並由商業公司(XenSource)提供支援。Guest OS 通常被稱為domU domains，而host OS 則是被稱為dom0。在Linux 上建立FreeBSD 虛擬機器的第一步，則是安裝Linux dom0 的**Xen**。在本例中，host OS 乃是Slackware Linux。

21.2.2.1 在Linux dom0 上設定Xen 3

1. 從XenSource 網站下載Xen 3.0

   從http://www.xensource.com/ 下載xen-3.0.4_1-src.tgz (http://bits.xensource.com/oss-xen/release/3.0.4-1/src.tgz/xen-3.0.4_1-src.tgz)。

2. 解壓縮

```
# cd xen-3.0.4_1-src
# KERNELS="linux-2.6-xen0 linux-2.6-xenU" make world
# make install
```

   Note: 為dom0 重新編譯kernel：

```
# cd xen-3.0.4_1-src/linux-2.6.16.33-xen0
# make menuconfig
# make
# make install
```

   舊版的**Xen** 可能需要用`make ARCH=xen menuconfig`

3. 增加選項到Grub 的menu.lst 選單

   修改`/boot/grub/menu.lst` 加上下列設定：

```
title Xen-3.0.4
root (hd0,0)
kernel /boot/xen-3.0.4-1.gz dom0_mem=262144
module /boot/vmlinuz-2.6.16.33-xen0 root=/dev/hda1 ro
```

4. 重開機並進入Xen

   首先，修改`/etc/xen/xend-config.sxp` 加上下列設定：

```
(network-script 'network-bridge netdev=eth0')
```

   接下來，就可以啓動**Xen**：

```
# /etc/init.d/xend start
# /etc/init.d/xendomains start
```

現在dom0 已經開始運作：

```
# xm list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0   256     1     r-----  54452.9
```

21.2.2.2 FreeBSD 7-CURRENT domU

從http://www.fsmware.com/ 下載搭配**Xen 3.0** 的FreeBSD domU kernel 相關檔案

- kernel-current (http://www.fsmware.com/xenofreebsd/7.0/download/kernel-current)
- mdroot-7.0.bz2 (http://www.fsmware.com/xenofreebsd/7.0/download/mdroot-7.0.bz2)
- xmexample1.bsd (http://www.fsmware.com/xenofreebsd/7.0/download/config/xmexample1.bsd)

把`xmexample1.bsd` 設定檔放到`/etc/xen/`，並修改kernel 及disk image 相關位置。以下是示範的例子：

```
kernel = "/opt/kernel-current"
memory = 256
name = "freebsd"
vif = [ ” ]
disk = [ ’file:/opt/mdroot-7.0,hda1,w’ ]
#on_crash    = ’preserve’
extra = "boot_verbose"
extra += ",boot_single"
extra += ",kern.hz=100"
extra += ",vfs.root.mountfrom=ufs:/dev/xbd769a"
```

其中`mdroot-7.0.bz2` 檔要記得解壓縮之。

接下來，要修改`kernel-current` 設定檔的__xen_guest 小節，並加上**Xen 3.0.3** 所需的VIRT_BASE：

```
# objcopy kernel-current -R __xen_guest
# perl -e ’print "LOADER=generic,GUEST_OS=freebsd,GUEST_VER=7.0,XEN_VER=xen-3.0,BSD_SYMTAB,VIRT_BASE=0xC0000
# objcopy kernel-current --add-section __xen_guest=tmp

# objdump -j __xen_guest -s kernel-current

kernel-current:     file format elf32-i386

Contents of section __xen_guest:
 0000 4c4f4144 45523d67 656e6572 69632c47  LOADER=generic,G
 0010 55455354 5f4f533d 66726565 6273642c  UEST_OS=freebsd,
 0020 47554553 545f5645 523d372e 302c5845  GUEST_VER=7.0,XE
 0030 4e5f5645 523d7865 6e2d332e 302c4253  N_VER=xen-3.0,BS
 0040 445f5359 4d544142 2c564952 545f4241  D_SYMTAB,VIRT_BA
 0050 53453d30 78433030 30303030 3000      SE=0xC0000000.
```

現在可以新增並啓動domU 囉：

```
# xm create /etc/xen/xmexample1.bsd -c
Using config file "/etc/xen/xmexample1.bsd".
Started domain freebsd
```

```
WARNING: loader(8) metadata is missing!
Copyright (c) 1992-2006 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
The Regents of the University of California. All rights reserved.
FreeBSD 7.0-CURRENT #113: Wed Jan  4 06:25:43 UTC 2006
    kmacy@freebsd7.gateway.2wire.net:/usr/home/kmacy/p4/freebsd7_xen3/src/sys/i386-xen/compile/XE
WARNING: DIAGNOSTIC option enabled, expect reduced performance.
Xen reported: 1796.927 MHz processor.
Timecounter "ixen" frequency 1796927000 Hz quality 0
CPU: Intel(R) Pentium(R) 4 CPU 1.80GHz (1796.93-MHz 686-class CPU)
  Origin = "GenuineIntel"  Id = 0xf29  Stepping = 9
  Features=0xbfebfbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,CLF
  DTS,ACPI,MMX,FXSR,SSE,SSE2,SS,HTT,TM,PBE>
  Features2=0x4400<CNTX-ID,<b14>>
real memory  = 265244672 (252 MB)
avail memory = 255963136 (244 MB)
xc0: <Xen Console> on motherboard
cpu0 on motherboard
Timecounters tick every 10.000 msec
[XEN] Initialising virtual ethernet driver.
xn0: Ethernet address: 00:16:3e:6b:de:3a
[XEN]
Trying to mount root from ufs:/dev/xbd769a
WARNING: / was not properly dismounted
Loading configuration files.
No suitable dump device was found.
Entropy harvesting: interrupts ethernet point_to_point kickstart.
Starting file system checks:
/dev/xbd769a: 18859 files, 140370 used, 113473 free (10769 frags, 12838 blocks, 4.2% fragmentatio
Setting hostname: demo.freebsd.org.
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
   inet6 ::1 prefixlen 128
   inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
   inet 127.0.0.1 netmask 0xff000000
Additional routing options:.
Mounting NFS file systems:.
Starting syslogd.
/etc/rc: WARNING: Dump device does not exist.  Savecore not run.
ELF ldconfig path: /lib /usr/lib /usr/lib/compat /usr/X11R6/lib /usr/local/lib
a.out ldconfig path: /usr/lib/aout /usr/lib/compat/aout /usr/X11R6/lib/aout
Starting usbd.
usb: Kernel module not available: No such file or directory
Starting local daemons:.
Updating motd.
Starting sshd.
Initial i386 initialization:.
Additional ABI support: linux.
Starting cron.
Local package initialization:.
Additional TCP options:.
Starting background file system checks in 60 seconds.

Sun Apr  1 02:11:43 UTC 2007
```

```
FreeBSD/i386 (demo.freebsd.org) (xc0)

login:
```

現在domU 應該可以跑FreeBSD 7.0-CURRENT kernel：

```
# uname -a
FreeBSD demo.freebsd.org 7.0-CURRENT FreeBSD 7.0-CURRENT #113: Wed Jan  4 06:25:43 UTC 2006
kmacy@freebsd7.gateway.2wire.net:/usr/home/kmacy/p4/freebsd7_xen3/src/sys/i386-xen/compile/XENCON
```

接下來是設定domU 的網路，FreeBSD domU 會用代號為xn0 的特殊網路卡：

```
# ifconfig xn0 10.10.10.200 netmask 255.0.0.0
# ifconfig
xn0: flags=843<UP,BROADCAST,RUNNING,SIMPLEX> mtu 1500
    inet 10.10.10.200 netmask 0xff000000 broadcast 10.255.255.255
    ether 00:16:3e:6b:de:3a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
      inet6 ::1 prefixlen 128
      inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
      inet 127.0.0.1 netmask 0xff000000
```

在dom0 Slackware 上應該會出現一些**Xen** 專用的網路卡：

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:07:E9:A0:02:C2
          inet addr:10.10.10.130  Bcast:0.0.0.0  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:815 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1400 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:204857 (200.0 KiB)  TX bytes:129915 (126.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:99 errors:0 dropped:0 overruns:0 frame:0
          TX packets:99 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9744 (9.5 KiB)  TX bytes:9744 (9.5 KiB)

peth0     Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:1853349 errors:0 dropped:0 overruns:0 frame:0
          TX packets:952923 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2432115831 (2.2 GiB)  TX bytes:86528526 (82.5 MiB)
          Base address:0xc000 Memory:ef020000-ef040000

vif0.1    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:1400 errors:0 dropped:0 overruns:0 frame:0
          TX packets:815 errors:0 dropped:0 overruns:0 carrier:0
```

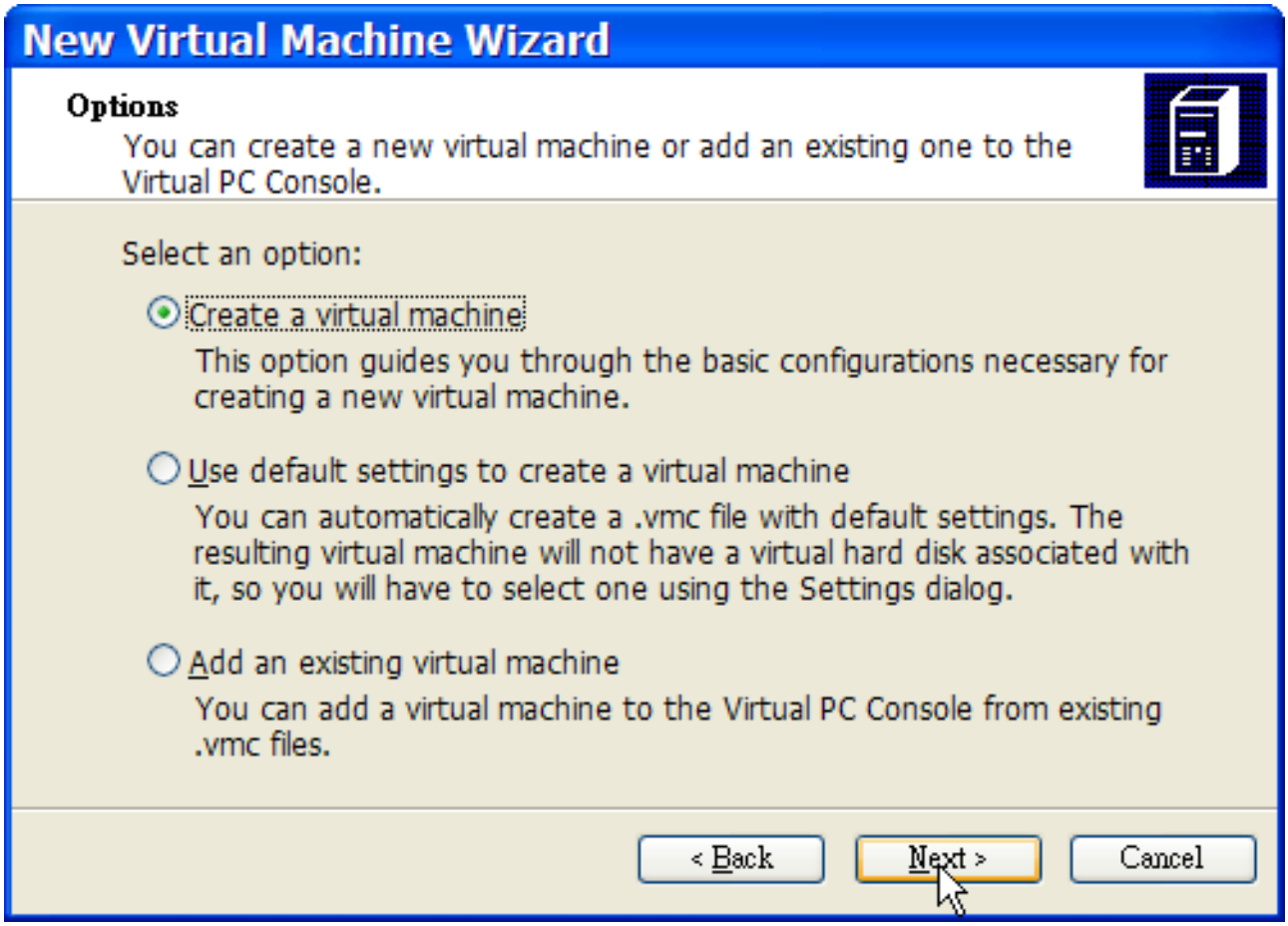
```
          collisions:0 txqueuelen:0
          RX bytes:129915 (126.8 KiB)  TX bytes:204857 (200.0 KiB)

vif1.0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:157 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:140 (140.0 b)  TX bytes:158 (158.0 b)

xenbr1    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:112 (112.0 b)  TX bytes:0 (0.0 b)

# brctl show
bridge name     bridge id           STP enabled         interfaces
xenbr1          8000.feffffffffff   no                  vif0.1
                                                        peth0
                                                        vif1.0
```

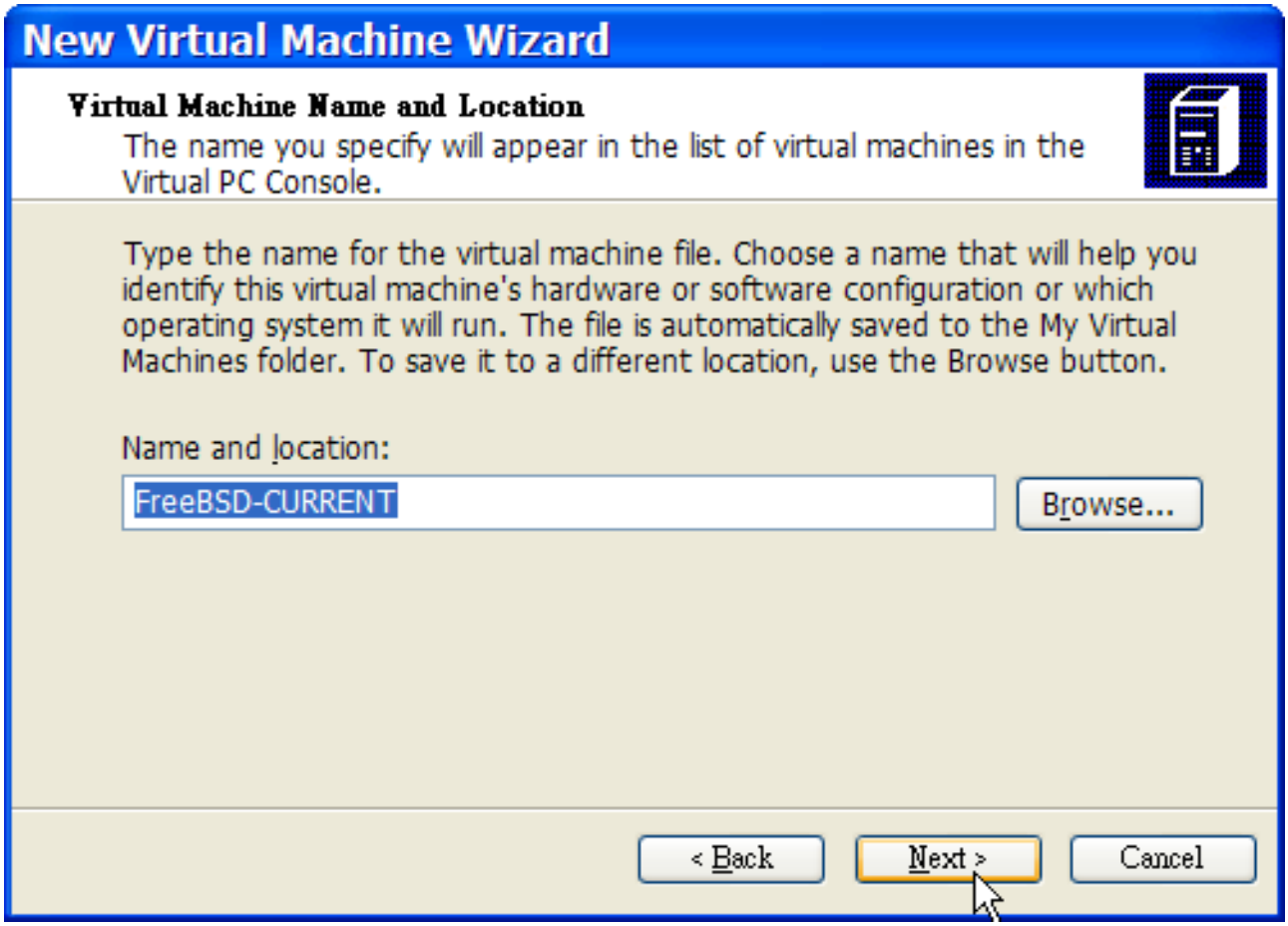
21.2.3 Windows 上的Virtual PC

Virtual PC 是Microsoft 的Windows 軟體產品，可以免費下載使用。相關系統需求，請參閱 system requirements (http://www.microsoft.com/windows/downloads/virtualpc/sysreq.mspx) 說明。在Microsoft Windows 裝完**Virtual PC** 之後，必須針對所欲安裝的虛擬機器來作相關設定。

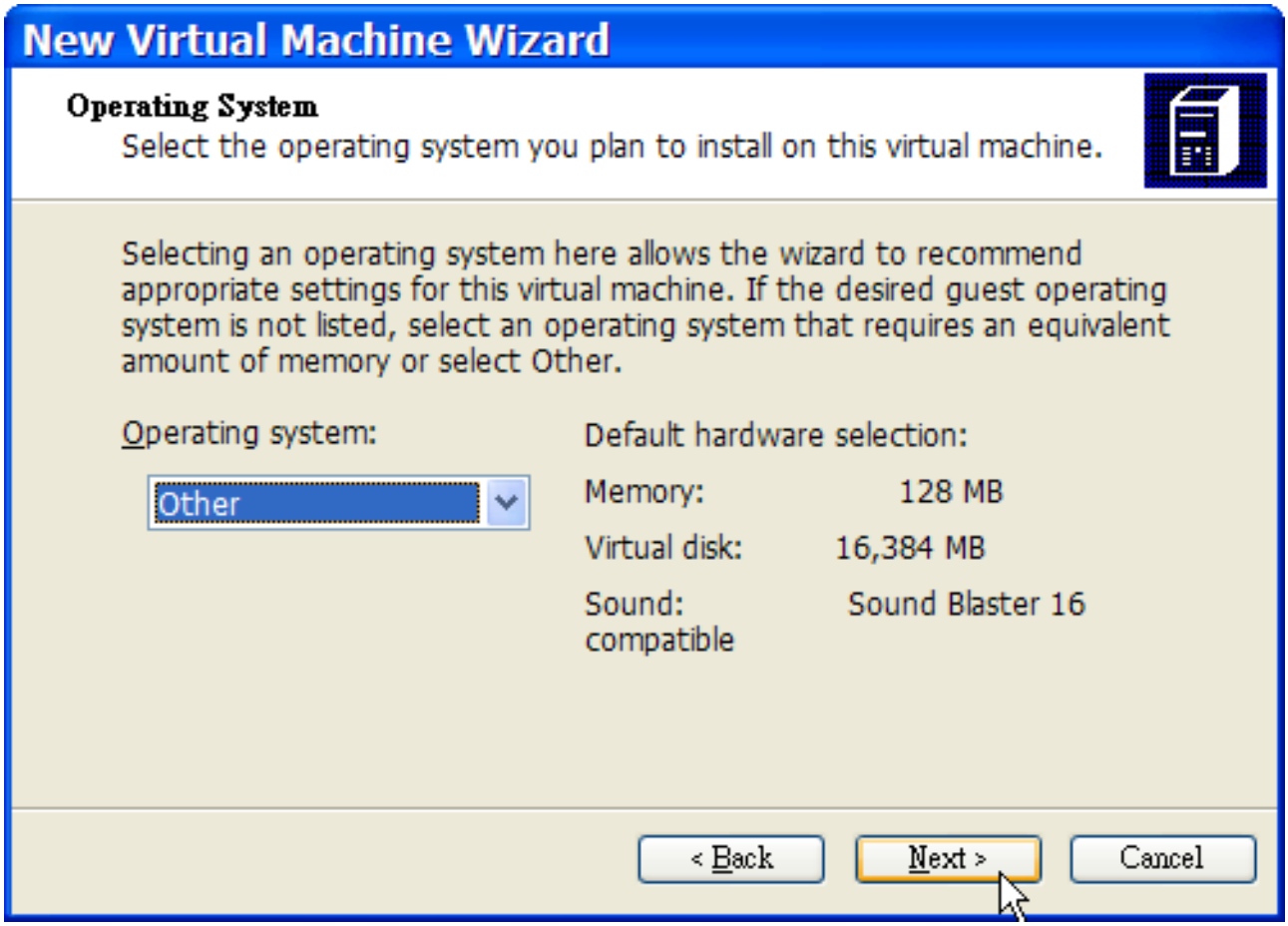
21.2.3.1 在Virtual PC/Microsoft® Windows 上安裝FreeBSD

在Microsoft Windows/**Virtual PC** 上安裝FreeBSD 的第一步是新增虛擬機器。如下所示，在提示視窗內請選擇Create a virtual machine：

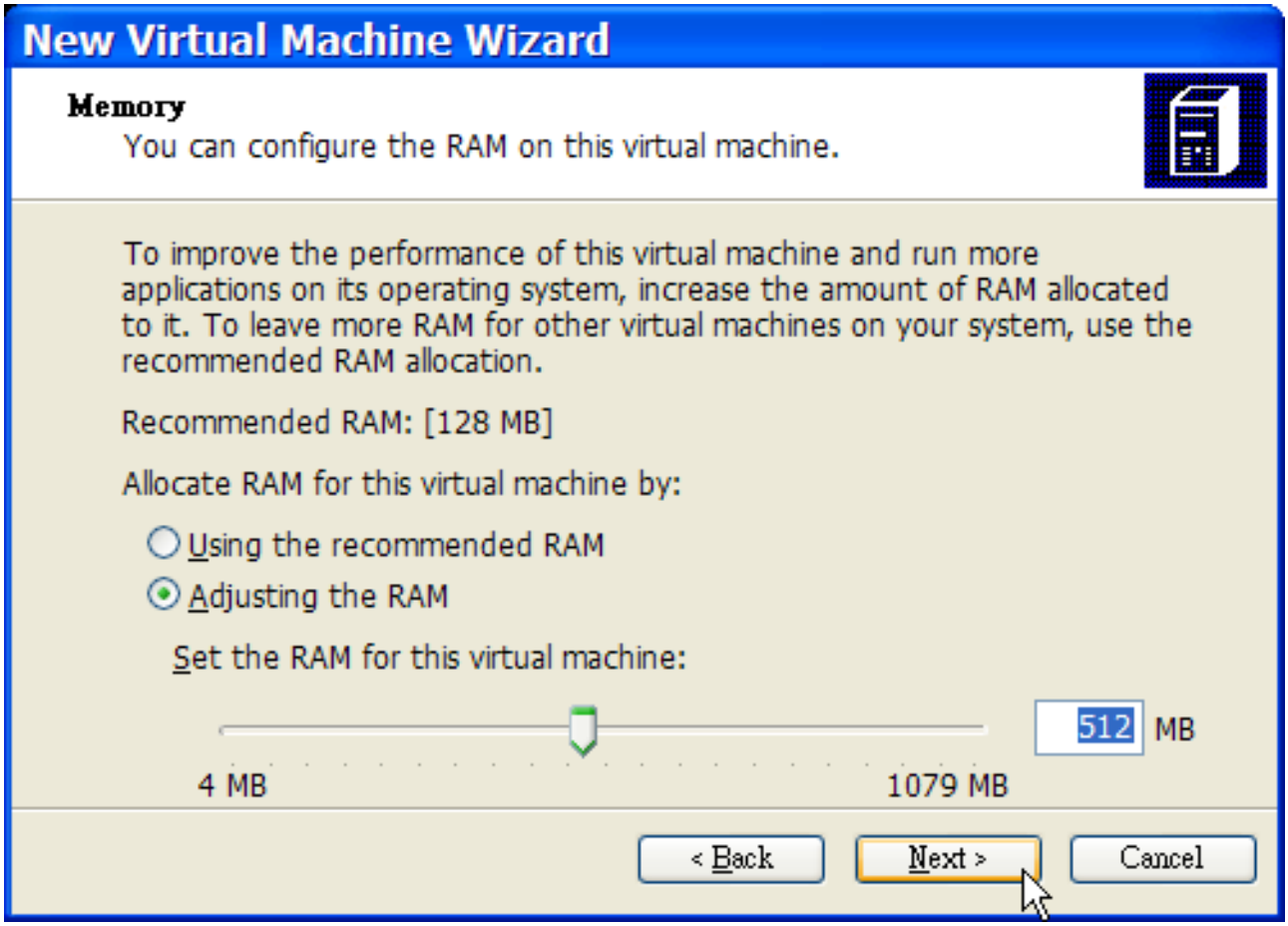

New Virtual Machine Wizard
Options
You can create a new virtual machine or add an existing one to the Virtual PC Console.
Select an option:
Create a virtual machine
This option guides you through the basic configurations necessary for creating a new virtual machine.
Use default settings to create a virtual machine
You can automatically create a .vmc file with default settings. The resulting virtual machine will not have a virtual hard disk associated with it, so you will have to select one using the Settings dialog.
Add an existing virtual machine
You can add a virtual machine to the Virtual PC Console from existing .vmc files.
< Back
Next >
Cancel

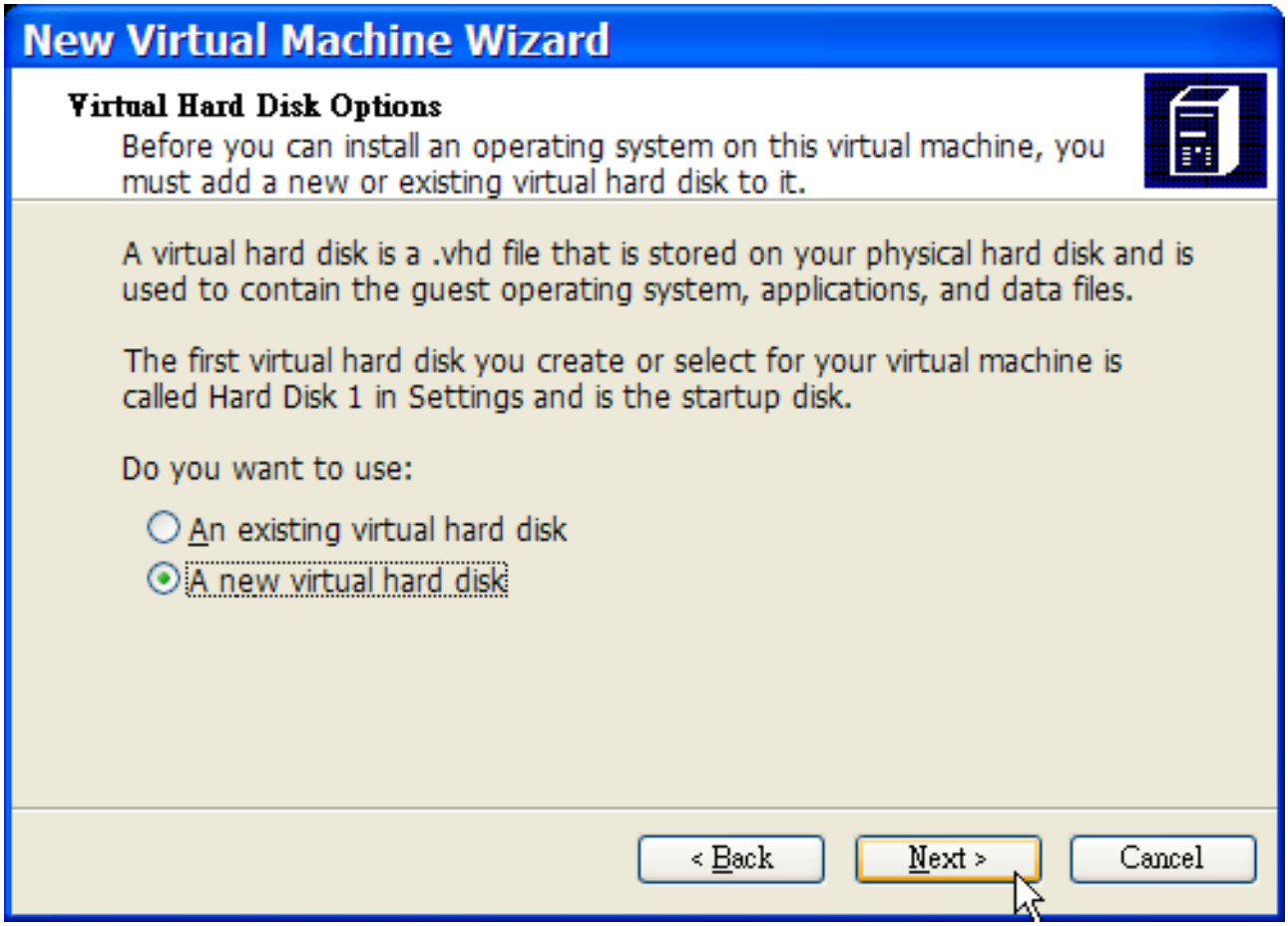


然後在Operating system 處選Other：

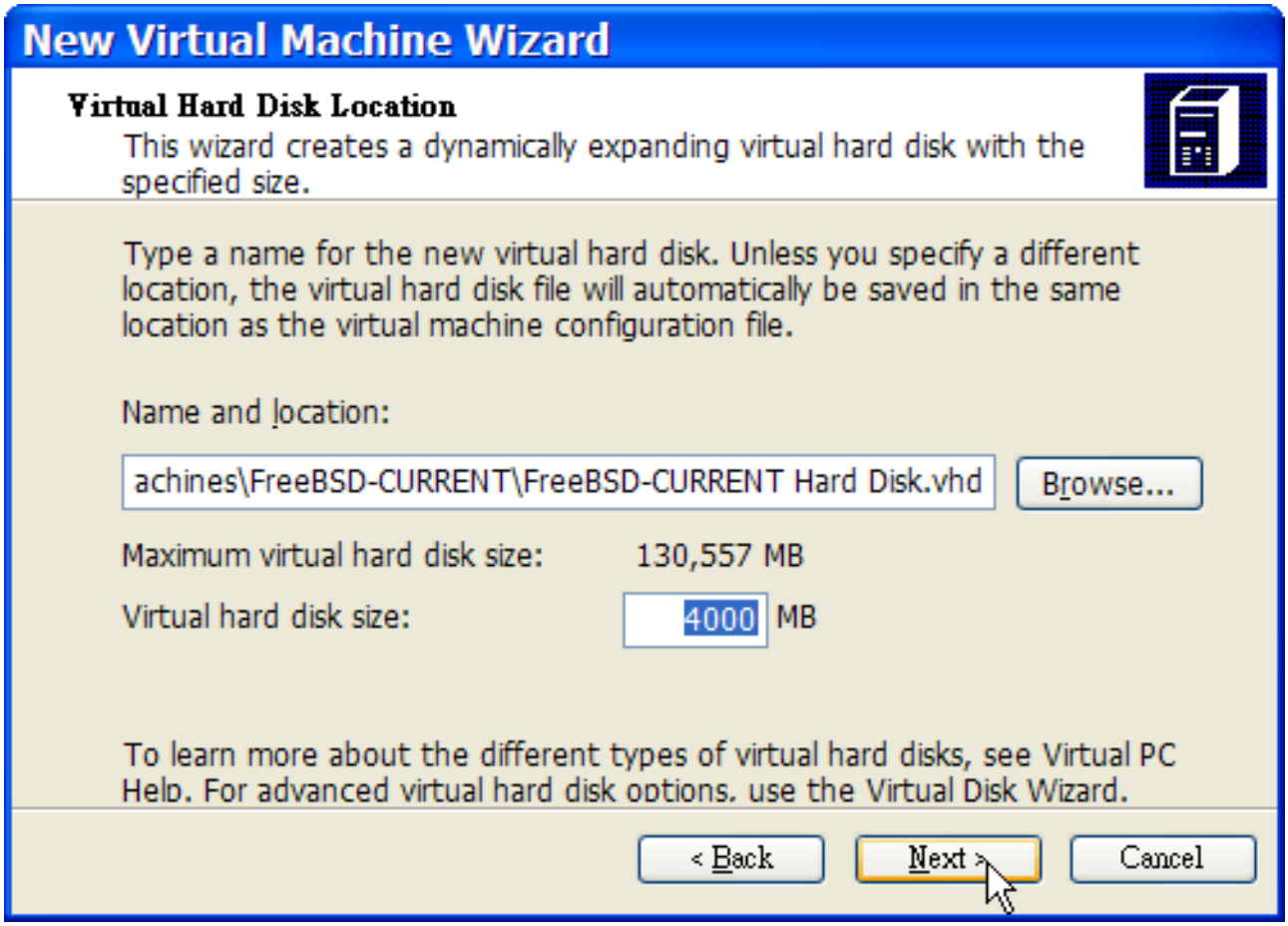


並依據自身需求來規劃硬碟容量跟記憶體的分配。對大多數在**Virtual PC** 使用FreeBSD 的情況而言，大約4GB 硬碟空間以及512MB RAM 就夠用了：

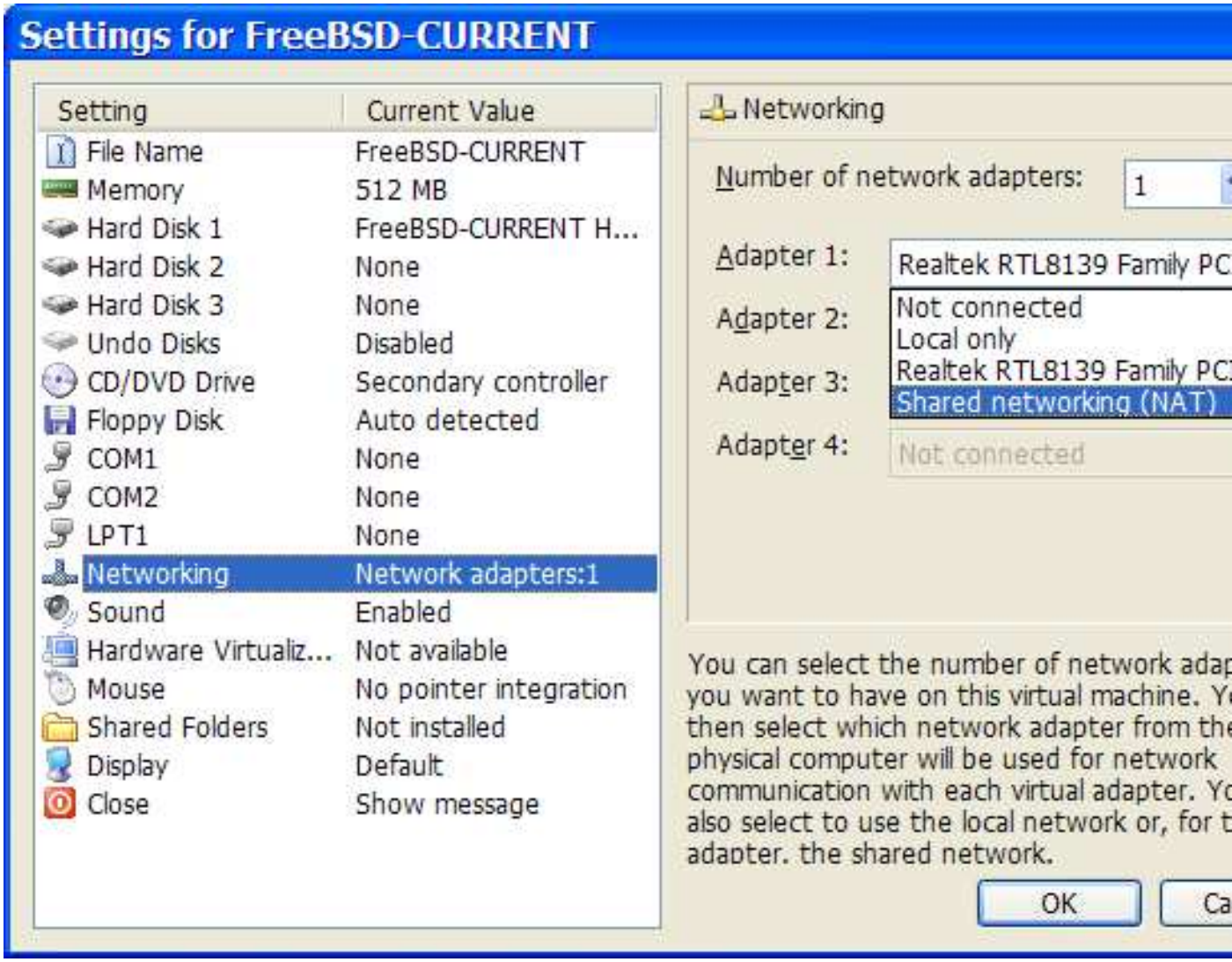

New Virtual Machine Wizard
Memory
You can configure the RAM on this virtual machine.
To improve the performance of this virtual machine and run more applications on its operating system, increase the amount of RAM allocated to it. To leave more RAM for other virtual machines on your system, use the recommended RAM allocation.
Recommended RAM: [128 MB]
Allocate RAM for this virtual machine by:
Using the recommended RAM
Adjusting the RAM
Set the RAM for this virtual machine:
512
MB
4 MB
1079 MB
< Back
Next >
Cancel


New Virtual Machine Wizard
Virtual Hard Disk Options
Before you can install an operating system on this virtual machine, you must add a new or existing virtual hard disk to it.
A virtual hard disk is a .vhd file that is stored on your physical hard disk and is used to contain the guest operating system, applications, and data files.
The first virtual hard disk you create or select for your virtual machine is called Hard Disk 1 in Settings and is the startup disk.
Do you want to use:
An existing virtual hard disk
A new virtual hard disk
< Back
Next >
Cancel

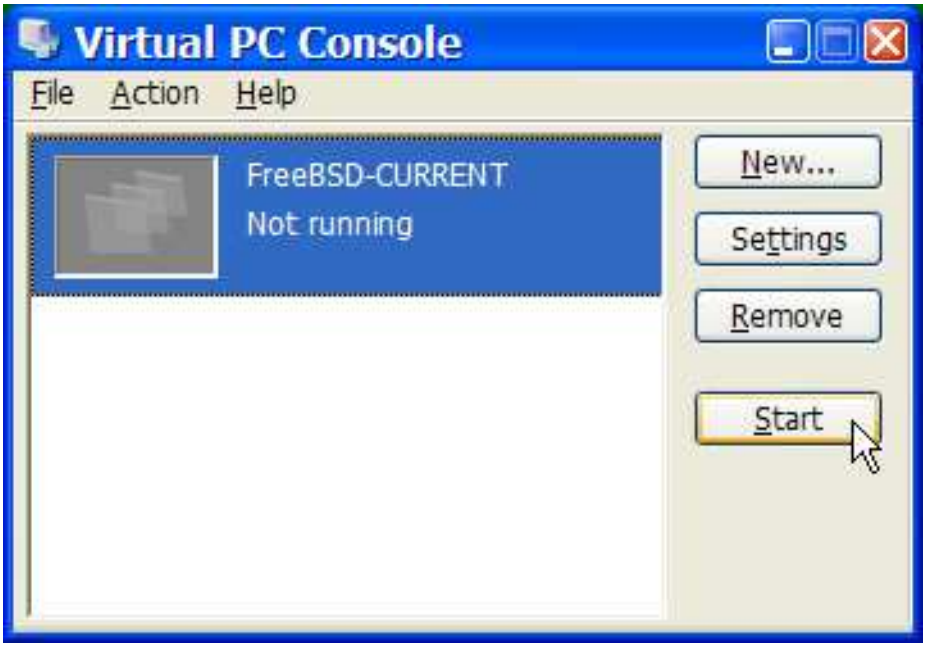
儲存設定檔：

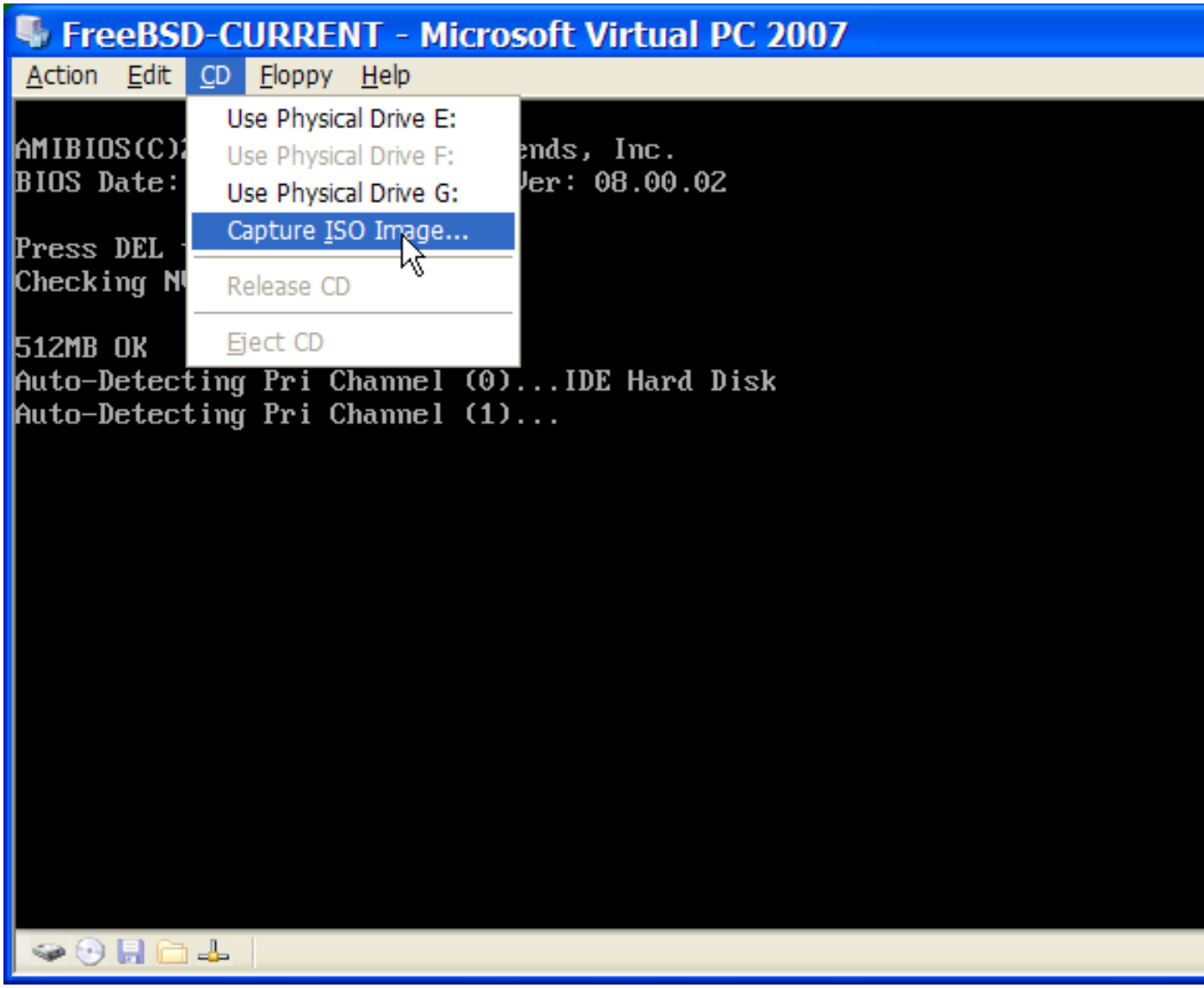


接下來選剛剛所新增的FreeBSD 虛擬機器，並按下Settings，以設定網路種類以及網路卡：

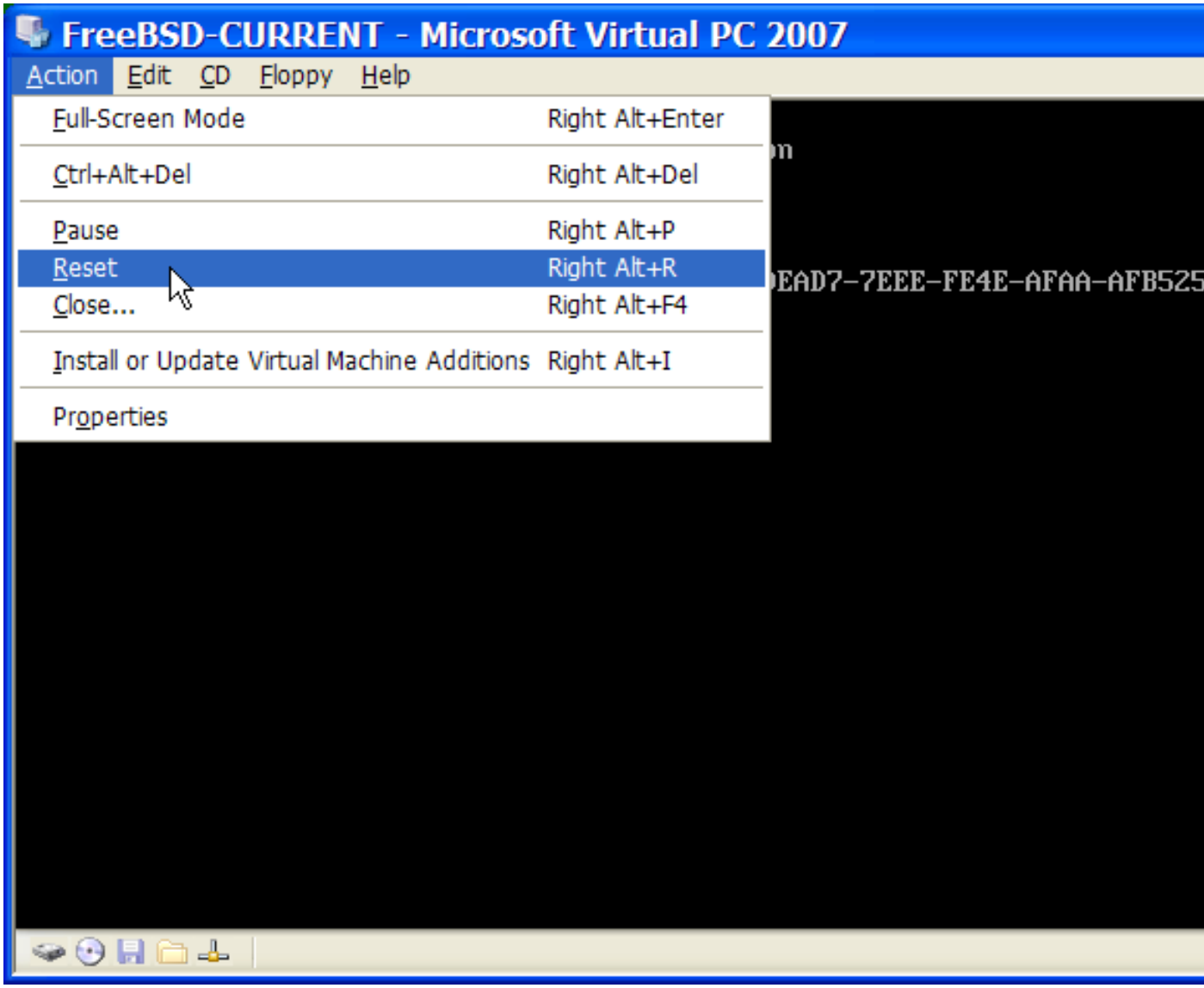


在FreeBSD 虛擬機器新增後，就可以繼續以其安裝FreeBSD。安裝方面，比較好的作法是使用官方的FreeBSD 光碟或者從官方FTP 站下載ISO image 檔。若您的Windows 檔案系統內已經有該ISO 檔，或者光碟機內有放安裝片，那麼就可以在FreeBSD 虛擬機器上連按兩下，以開始啓動。接著在**Virtual PC** 視窗內按CD 再按Capture ISO Image... 。接著會出現一個視窗，可以把虛擬機器內的光碟機設定到該ISO 檔，或者是實體光碟機。

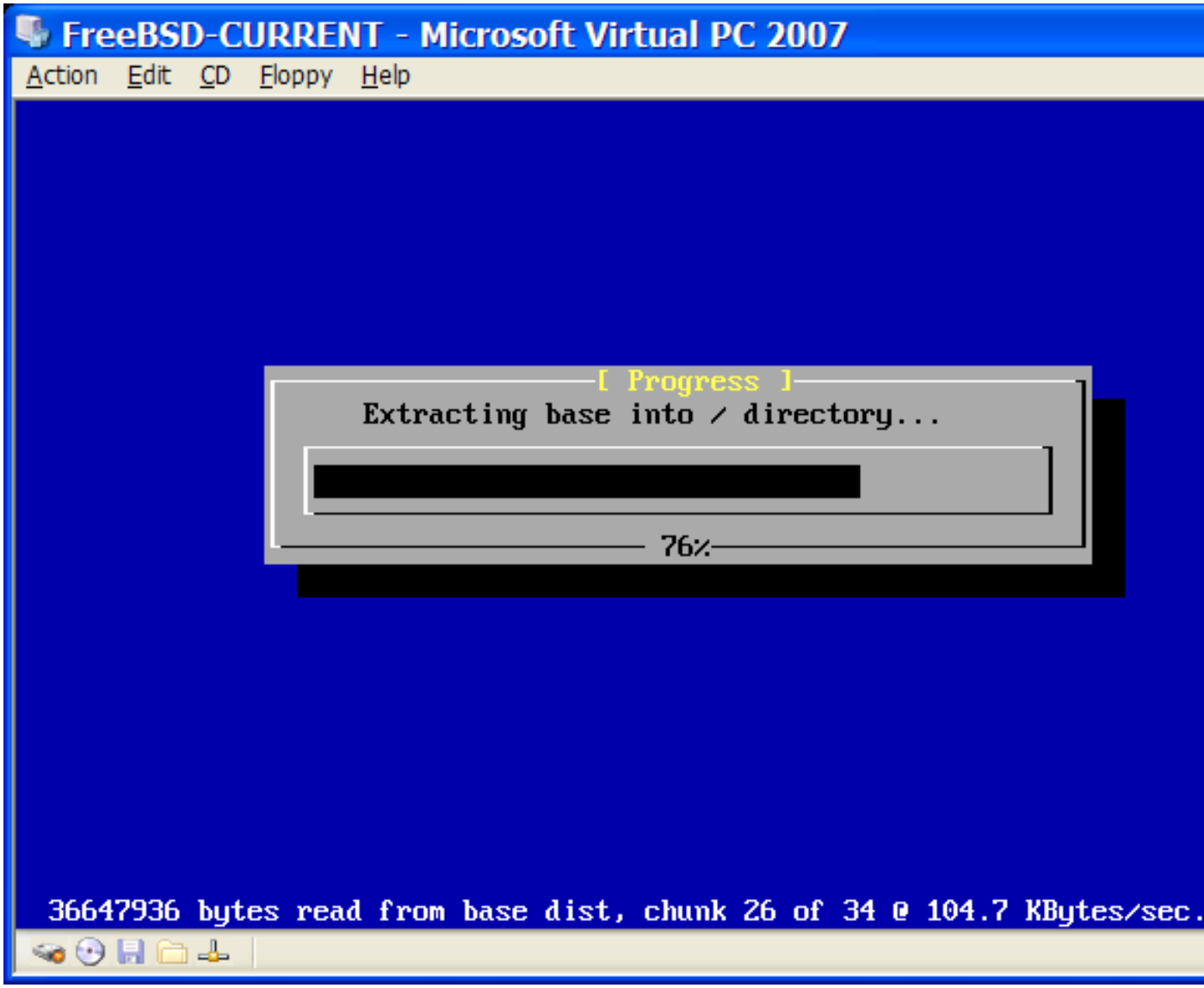

Virtual PC Console
File
Action
Help
FreeBSD-CURRENT
Not running
New...
Settings
Remove
Start



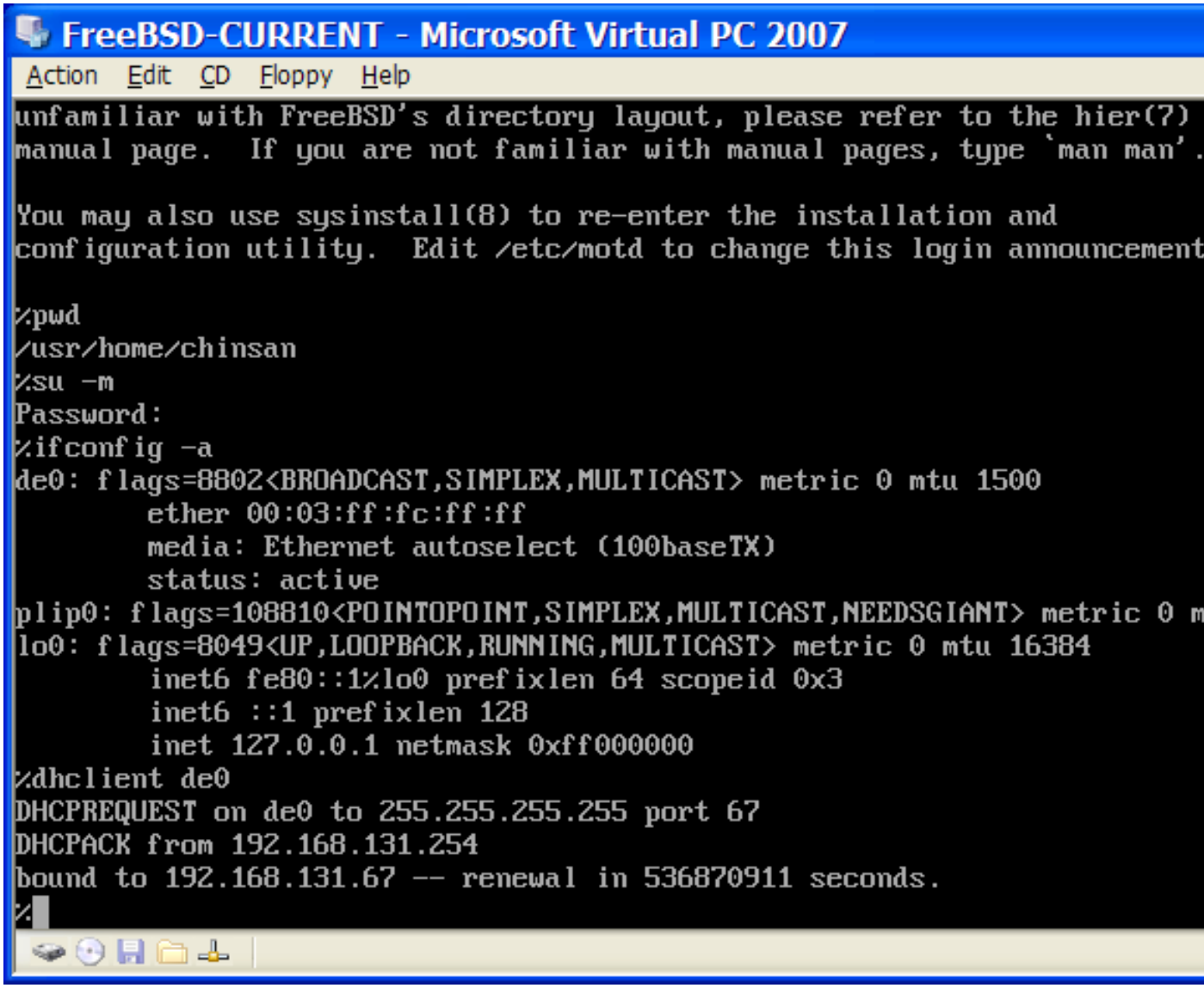
設好光碟片來源之後，就可以重開機，也就是先按Action 再按Reset 即可。**Virtual PC** 會以特殊BIOS 開機，並與普通BIOS 一樣會先檢查是否有光碟機。



此時，它就會找到FreeBSD 安裝片，並開始在Chapter 2 內所介紹到的**sysinstall** 安裝過程。這時候也可順便裝X11，但先不要進行相關設定。



完成安裝之後，記得把光碟片退出或者ISO image 退片。最後，把裝好的FreeBSD 虛擬機器重開機即可。



21.2.3.2 調整Microsoft Windows/Virtual PC 上的FreeBSD

在Microsoft Windows 上以**Virtual PC** 裝好FreeBSD 後，還需要作一些設定步驟，以便將虛擬機器內的FreeBSD 最佳化。

1. 設定boot loader 參數

   最重要的步驟乃是藉由調降`kern.hz` 來降低**Virtual PC** 環境內FreeBSD 的CPU 佔用率。可以在`/boot/loader.conf` 內加上下列設定即可：

   ```
   kern.hz=100
   ```

   若不作這設定，那麼光是idle 狀態的FreeBSD **Virtual PC** guest OS 就會在僅單一處理器的電腦上佔了大約40% 的CPU 佔用率。作上述修改之後，佔用率就會降至大約3%。

2. 設定新的kernel 設定檔

可以放心把所有SCSI、FireWire、USB 相關設備都移除。**Virtual PC** 有提供de(4) 的虛擬網卡，因此除了de(4) 以及miibus(4) 以外的其他網路卡也都可以從kernel 中移除。

3. 設定網路

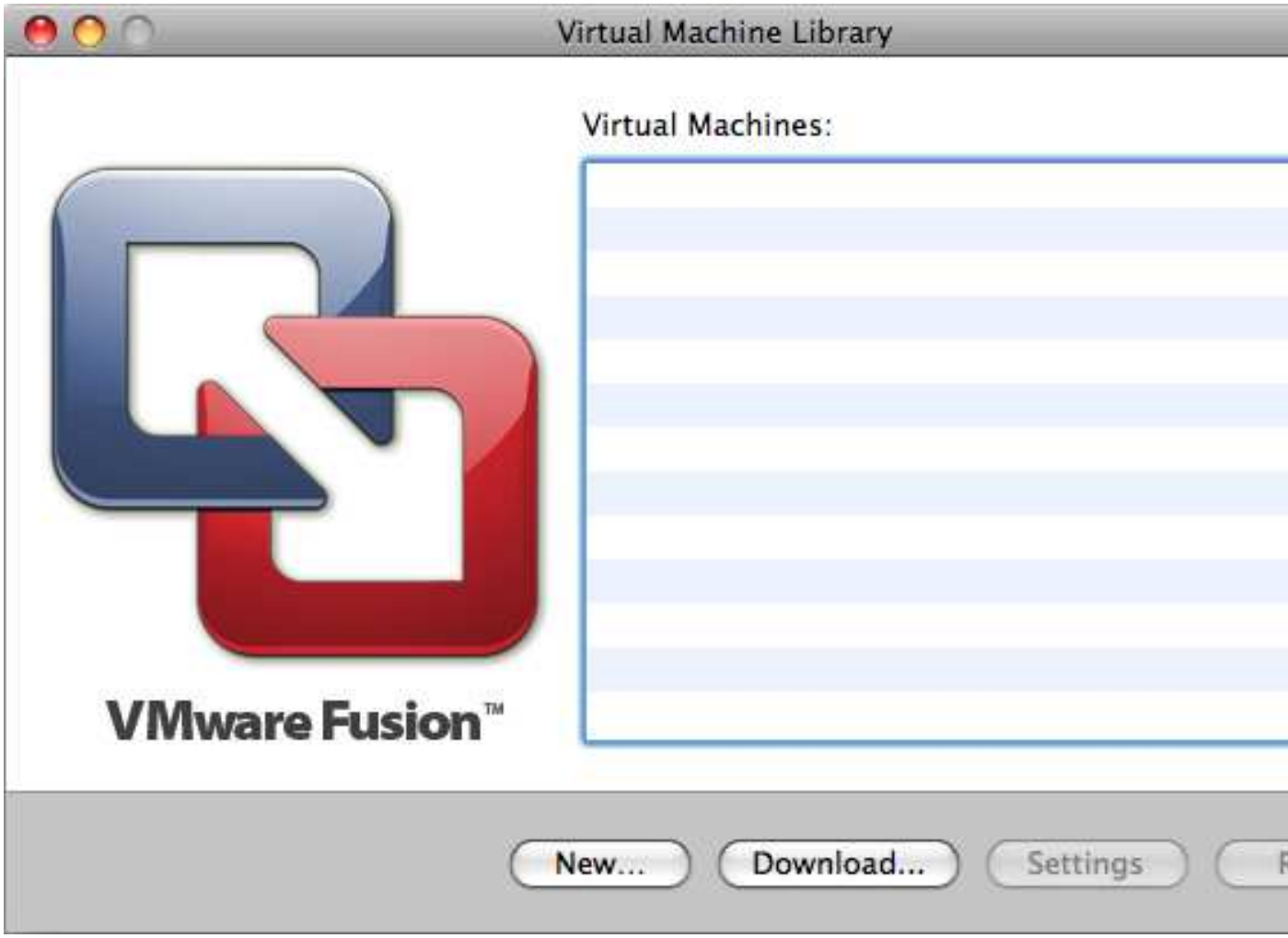
   可以替虛擬機器簡單用DHCP 來設定與host(Microsoft Windows) 相同的LAN 網路環境，只要在`/etc/rc.conf` 加上`ifconfig_de0="DHCP"` 即可完成。其他進階的網路設定方式，請參閱Chapter 29。

21.2.4 在MacOS 上的VMware

Mac 上的**VMWare Fusion** 乃是可用於搭配Intel CPU 以及Mac OS 10.4.9 之Apple Mac 以上的Apple Mac 電腦之商業軟體。FreeBSD 是其有完整支援的guest OS 之一。在Mac OS X 上裝完**VMWare Fusion** 之後，必須針對所欲安裝的guest OS 來作相關的虛擬機器設定。

21.2.4.1 在VMWare/Mac OS X 上安裝FreeBSD

首先執行VMWare Fusion，而其Virtual Machine Library 也會隨之一併載入，這時請按"New" 來建立VM(虛擬機器)：



接著會有New Virtual Machine Assistant 來協助您建立VM，請按Continue 繼續：

在Operating System 選Other，以及Version 處請選擇是否要FreeBSD 或FreeBSD 64-bit，這部份請依自身需求是否有要64-bit 支援而定：

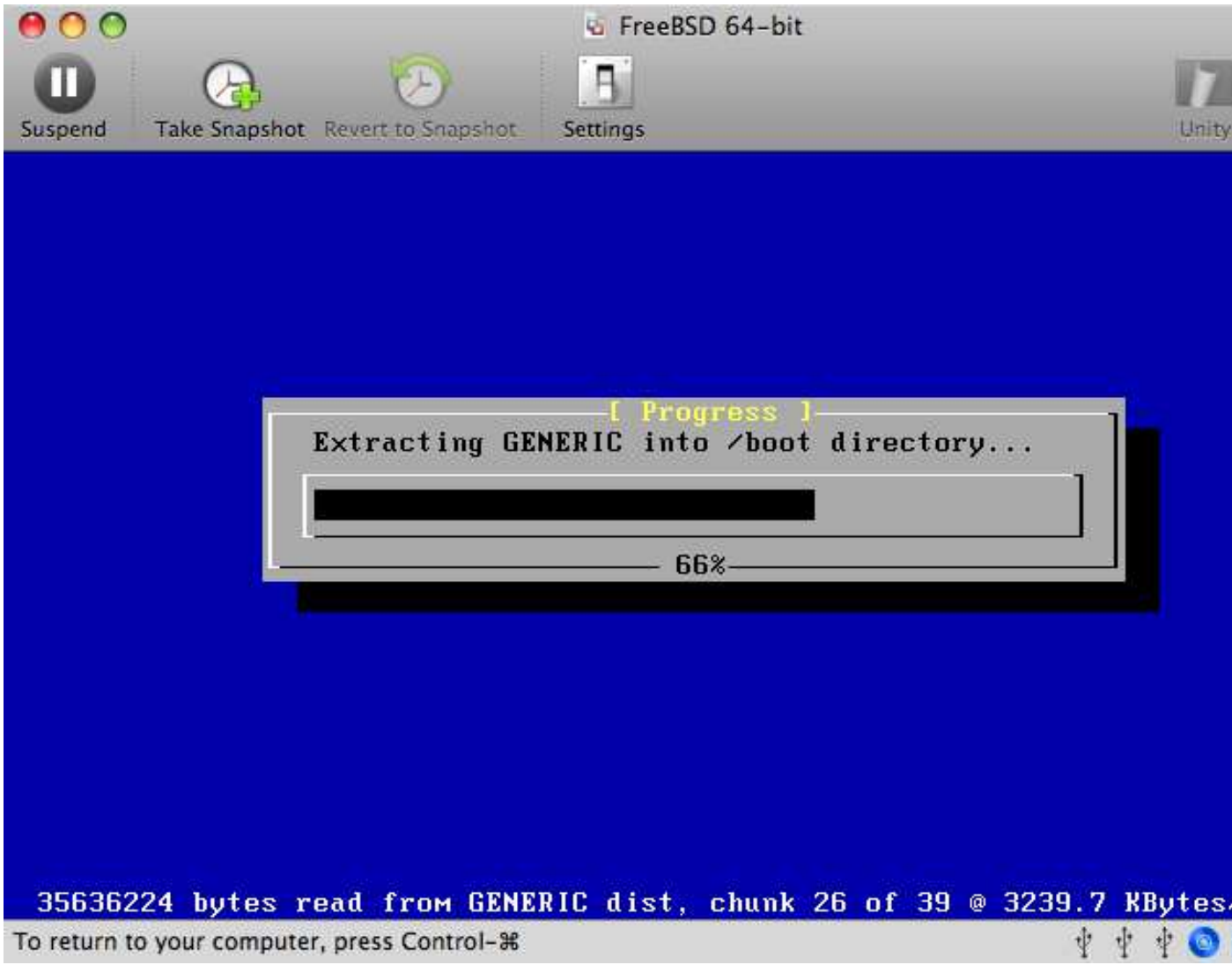
接著設定VM image 檔要存到何處，以及決定名稱：

決定該VM 的虛擬硬碟要用多大：

選擇要裝VM 的方式為何，要用ISO image 檔或者光碟機：

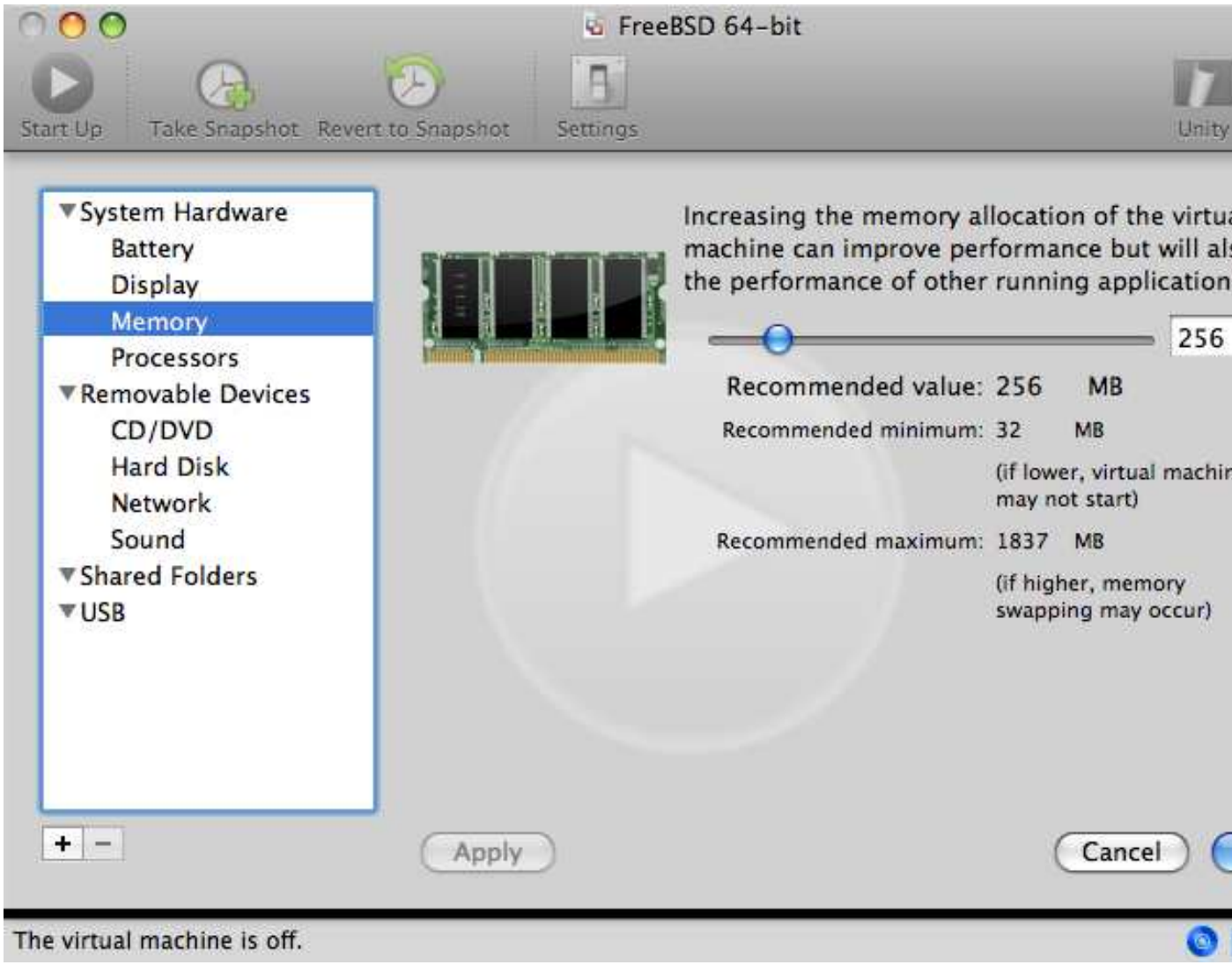
按Finish 以完畢，接著就會啓動該VM：

接著就照以往安裝FreeBSD 的方式來裝，若不熟的話請參閱Chapter 2：



裝完之後，就可以修改一些VM 設定，像是記憶體大小：

Note: VM 在運作之時，不能修改VM 的硬體設定。


FreeBSD 64-bit
Start Up
Take Snapshot
Revert to Snapshot
Settings
Unity
System Hardware
Battery
Display
Memory
Processors
Removable Devices
CD/DVD
Hard Disk
Network
Sound
Shared Folders
USB
Increasing the memory allocation of the virtu
machine can improve performance but will al
the performance of other running application
256
Recommended value: 256 MB
Recommended minimum: 32 MB
(if lower, virtual machir
may not start)
Recommended maximum: 1837 MB
(if higher, memory
swapping may occur)
Apply
Cancel
The virtual machine is off.

調整VM 的CPU 數量：

光碟機狀態，通常不再需要用的時候，就可以切斷其與VM 的連接：

最後要改的則是VM 的網路設定。若除了Host OS 之外的機器也能連到VM，那麼請選Connect directly to the physical network (Bridged)，否則就選Share the host's internet connection (NAT) 即可讓VM 連到Internet，但外面則無法連入該VM。

改完上述設定之後，就可以啓動新裝妥的FreeBSD 虛擬機器。

21.2.4.2 調整Mac OS X/VMWare 上的FreeBSD

把FreeBSD 成功裝到Mac OS X 的**VMWare** 之後，還需要作一些設定步驟，以便將虛擬機器內的FreeBSD 最佳化。

1. 設定boot loader 參數

   最重要的步驟乃是藉由調降`kern.hz` 來降低**VMWare** 環境內FreeBSD 的CPU 佔用率。可以在`/boot/loader.conf` 內加上下列設定即可：

   ```
   kern.hz=100
   ```

   若不作這設定，那麼光是idle 狀態的FreeBSD (**VMWare** guest OS) 就會在僅單一處理器的iMac 上佔了大約15% 的CPU 佔用率。作上述修改之後，佔用率就會降至大約5%。

2. 設定新的kernel 設定檔

可以放心把所有FireWire、USB 相關設備都移除。**VMWare** 有提供em(4) 的虛擬網卡，因此，除了em(4) 以及miibus(4) 以外的其他網路卡，也都可以從kernel 中移除。

3. 設定網路

   可以替虛擬機器簡單用DHCP 來設定與host Mac 相同的LAN 網路環境，只要在`/etc/rc.conf` 加上`ifconfig_em0="DHCP"` 即可。其他進階的網路設定方式，請參考Chapter 29。

21.3 以FreeBSD 為Host OS

目前，尚未有任何虛擬機器軟體有官方支援FreeBSD 作為host OS，但蠻多人都有在用舊版**VMware** 所提供的這項功能。不過，目前已經有人為讓**Xen** 能夠以FreeBSD 為host OS 為目標，而進行相關工作。

Chapter 22 語系設定- I18N/L10N 用法與設定

Contributed by Andrey Chernov. Rewritten by Michael C. Wu.

22.1 概述

由於FreeBSD 是分佈全世界的使用者及志工所支持的計畫，本章主要探討的是FreeBSD 的國際化、本土化議題，以便讓母語不是英語系的人也能順利完成各項工作。在作業系統、應用程式兩種層面，主要都是透過i18n 標準來實作的，所以，這裡我們將會介紹大致運作方式。

讀完這章，您將了解：

· 各種不同的語言與地區設定如何在作業系統上進行編碼。

· 如何設定登入用的shell 語系環境。

· 如何將你的console 設為英語以外的語系設定。

· 如何使用不同語系的設定，來讓X Window 運作更親切。

· 哪邊可以找到更多與i18n 規格相容的應用程式規格資料。

在開始閱讀這章之前，您需要：

· 知道如何以ports/packages 來安裝應用程式(Chapter 4)。

22.2 L10N 基礎概念

22.2.1 什麼是I18N/L10N?

程式開發人員習慣把internationalization 縮寫為I18N，中間的數字18 乃是最前與最後面字母之間的字母個數總和，而L10N 也是以一樣的方式，是“localization” 的縮寫。只要有符合I18N/L10N 規格、協定的應用程式，就可以讓使用者依各自語系而作設定。

I18N 應用程式是以I18N 開發工具來進行開發的，它可以讓程式開發人員透過寫簡單的文字檔，就可以把執行畫面上的選單、訊息翻譯為各語系的版本。我們強烈建議程式開發人員遵循這個遊戲規則。

22.2.2 為何該使用I18N/L10N？

只要有符合I18N/L10N 標準，就可以輕鬆地看、輸入、處理非英文的資料。

22.2.3 I18N 支援哪些語系？

I18N 和L10N 並非FreeBSD 所特有的，目前這世界上的幾乎任一主要語系都有支援，像是：中文、德文、日文、韓文、法文、俄文、越南文等等。

22.3 使用語系設定(Localization)

I18N 和L10N 並非FreeBSD 所特有的，而是共通的遊戲規則。我們鼓勵你在FreeBSD 世界中同樣遵守這項遊戲規則。

Locale 設定由三個部分所組成：語言代碼(Language Code)、國碼(Country Code)、編碼(Encoding)。所以，Locale 的設定名稱就是由這三個一起組成：

語言代碼_國碼.編碼

22.3.1 語言、國碼

使用者必須要先知道這些特定的國碼、語言代碼(國碼會告訴應用程式該使用哪一種語言)，才能讓FreeBSD 或其他支援I18N 的UNIX 類系統作locale 相關設定。此外，網頁瀏覽器(borwser)、SMTP/POP 主機、Web 主機等也都以這架構為主。下面是如何使用『語言代碼、國碼』的例子：

語言代碼/國碼	簡介
en_US	英文(美國)
ru_RU	俄文(俄國)
zh_TW	正體中文(台灣)

22.3.2 編碼

有些語言並非採用ASCII 編碼，可能是：8-bit、wide 或multibyte 字元，詳情請參閱multibyte(3)。較古早的程式可能無法正確判別、或誤判為特殊控制字元。而較新的程式都可以辨認8-bit 字元。由於各程式的作法不一，使用者可能需要在編譯程式時，加上wide 或multibyte 字元的支援設定，或是正確調整才行。要輸入、處理wide 或multibyte 字元的話，可多多利用FreeBSD Ports Collection (http://www.FreeBSD.org/ports/index.html) 內有各國語言版本的程式。詳情請參閱FreeBSD 各port 中的I18N 相關文件。

Specifically, the user needs to look at the application documentation to decide on how to configure it correctly or to pass correct values into the configure/Makefile/compiler.

Some things to keep in mind are:

· Language specific single C chars character sets (see multibyte(3)), e.g. ISO8859-1, ISO8859-15, KOI8-R, CP437.

· Wide or multibyte encodings, e.g. EUC, Big5.

You can check the active list of character sets at the IANA Registry (http://www.iana.org/assignments/character-sets).

> **Note:** FreeBSD use X11-compatible locale encodings instead.

22.3.3 I18N Applications

In the FreeBSD Ports and Package system, I18N applications have been named with `I18N` in their names for easy identification. However, they do not always support the language needed.

22.3.4 Setting Locale

Usually it is sufficient to export the value of the locale name as `LANG` in the login shell. This could be done in the user's `~/.login_conf` file or in the startup file of the user's shell (`~/.profile`, `~/.bashrc`, `~/.cshrc`). There is no need to set the locale subsets such as `LC_CTYPE`, `LC_CTIME`. Please refer to language-specific FreeBSD documentation for more information.

You should set the following two environment variables in your configuration files:

- `LANG` for POSIX setlocale(3) family functions
- `MM_CHARSET` for applications' MIME character set

This includes the user shell configuration, the specific application configuration, and the X11 configuration.

22.3.4.1 Setting Locale Methods

There are two methods for setting locale, and both are described below. The first (recommended one) is by assigning the environment variables in login class, and the second is by adding the environment variable assignments to the system's shell startup file.

22.3.4.1.1 Login Classes Method

This method allows environment variables needed for locale name and MIME character sets to be assigned once for every possible shell instead of adding specific shell assignments to each shell's startup file. User Level Setup can be done by an user himself and Administrator Level Setup require superuser privileges.

22.3.4.1.1.1 User Level Setup

Here is a minimal example of a `.login_conf` file in user's home directory which has both variables set for Latin-1 encoding:

```
me:\
 :charset=ISO-8859-1:\
 :lang=de_DE.ISO8859-1:
```

Here is an example of a `.login_conf` that sets the variables for Traditional Chinese in BIG-5 encoding. Notice the many more variables set because some software does not respect locale variables correctly for Chinese, Japanese, and Korean.

```
#Users who do not wish to use monetary units or time formats
#of Taiwan can manually change each variable
me:\
 :lang=zh_TW.Big5:\
 :lc_all=zh_TW.Big:\
 :lc_collate=zh_TW.Big5:\
 :lc_ctype=zh_TW.Big5:\
 :lc_messages=zh_TW.Big5:\
 :lc_monetary=zh_TW.Big5:\
 :lc_numeric=zh_TW.Big5:\
 :lc_time=zh_TW.Big5:\
 :charset=big5:\
 :xmodifiers="@im=xcin": #Setting the XIM Input Server
```

See Administrator Level Setup and login.conf(5) for more details.

22.3.4.1.1.2 Administrator Level Setup

Verify that the user's login class in `/etc/login.conf` sets the correct language. Make sure these settings appear in `/etc/login.conf`:

```
language_name:accounts_title:\
 :charset=MIME_charset:\
 :lang=locale_name:\
 :tc=default:
```

So sticking with our previous example using Latin-1, it would look like this:

```
german:German Users Accounts:\
 :charset=ISO-8859-1:\
 :lang=de_DE.ISO8859-1:\
 :tc=default:
```

Before changing users Login Classes execute the following command

```
# cap_mkdb /etc/login.conf
```

to make new configuration in `/etc/login.conf` visible to the system.

Changing Login Classes with vipw(8)

Use `vipw` to add new users, and make the entry look like this:

```
user:password:1111:11:language:0:0:User Name:/home/user:/bin/sh
```

Changing Login Classes with adduser(8)

Use `adduser` to add new users, and do the following:

- Set `defaultclass = language` in `/etc/adduser.conf`. Keep in mind you must enter a `default` class for all users of other languages in this case.
- An alternative variant is answering the specified language each time that

  ```
  Enter login class: default []:
  ```

  appears from adduser(8).
- Another alternative is to use the following for each user of a different language that you wish to add:

  ```
  # adduser -class language
  ```

Changing Login Classes with pw(8)

If you use pw(8) for adding new users, call it in this form:

```
# pw useradd user_name -L language
```

22.3.4.1.2 Shell Startup File Method

> **Note:** This method is not recommended because it requires a different setup for each possible shell program chosen. Use the Login Class Method instead.

To add the locale name and MIME character set, just set the two environment variables shown below in the `/etc/profile` and/or `/etc/csh.login` shell startup files. We will use the German language as an example below:

In `/etc/profile`:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

Or in `/etc/csh.login`:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Alternatively, you can add the above instructions to `/usr/share/skel/dot.profile` (similar to what was used in `/etc/profile` above), or `/usr/share/skel/dot.login` (similar to what was used in `/etc/csh.login` above).

For X11:

In `$HOME/.xinitrc`:

```
LANG=de_DE.ISO8859-1; export LANG
```

Or:

```
setenv LANG de_DE.ISO8859-1
```

Depending on your shell (see above).

22.3.5 Console Setup

For all single C chars character sets, set the correct console fonts in `/etc/rc.conf` for the language in question with:

```
font8x16=font_name
font8x14=font_name
font8x8=font_name
```

The *`font_name`* here is taken from the `/usr/share/syscons/fonts` directory, without the `.fnt` suffix.

Also be sure to set the correct keymap and screenmap for your single C chars character set through `sysinstall` (`/stand/sysinstall` in FreeBSD versions older than 5.2). Once inside **sysinstall**, choose Configure, then Console. Alternatively, you can add the following to `/etc/rc.conf`:

```
scrnmap=screenmap_name
keymap=keymap_name
keychange="fkey_number sequence"
```

The *screenmap_name* here is taken from the `/usr/share/syscons/scrnmaps` directory, without the `.scm` suffix. A screenmap with a corresponding mapped font is usually needed as a workaround for expanding bit 8 to bit 9 on a VGA adapter's font character matrix in pseudographics area, i.e., to move letters out of that area if screen font uses a bit 8 column.

If you have the **moused** daemon enabled by setting the following in your `/etc/rc.conf`:

```
moused_enable="YES"
```

then examine the mouse cursor information in the next paragraph.

By default the mouse cursor of the syscons(4) driver occupies the 0xd0-0xd3 range in the character set. If your language uses this range, you need to move the cursor's range outside of it. To enable the workaround for FreeBSD, add the following line to `/etc/rc.conf`:

```
mousechar_start=3
```

The *keymap_name* here is taken from the `/usr/share/syscons/keymaps` directory, without the `.kbd` suffix. If you are uncertain which keymap to use, you use can kbdmap(1) to test keymaps without rebooting.

The `keychange` is usually needed to program function keys to match the selected terminal type because function key sequences cannot be defined in the key map.

Also be sure to set the correct console terminal type in `/etc/ttys` for all `ttyv*` entries. Current pre-defined correspondences are:

Character Set	Terminal Type
ISO8859-1 or ISO8859-15	`cons25l1`
ISO8859-2	`cons25l2`
ISO8859-7	`cons25l7`
KOI8-R	`cons25r`
KOI8-U	`cons25u`
CP437 (VGA default)	`cons25`
US-ASCII	`cons25w`

For wide or multibyte characters languages, use the correct FreeBSD port in your `/usr/ports/`*language* directory. Some ports appear as console while the system sees it as serial vtty's, hence you must reserve enough vtty's for both X11 and the pseudo-serial console. Here is a partial list of applications for using other languages in console:

Language	Location
Traditional Chinese (BIG-5)	`chinese/big5con`
Japanese	`japanese/kon2-16dot` or `japanese/mule-freewnn`
Korean	`korean/han`

22.3.6 X11 Setup

Although X11 is not part of the FreeBSD Project, we have included some information here for FreeBSD users. For

more details, refer to the Xorg web site (http://www.x.org/) or whichever X11 Server you use.

In `~/.Xresources`, you can additionally tune application specific I18N settings (e.g., fonts, menus, etc.).

22.3.6.1 Displaying Fonts

Install **Xorg** server (`x11-servers/xorg-server`) or **XFree86** server (`x11-servers/XFree86-4-Server`), then install the language TrueType fonts. Setting the correct locale should allow you to view your selected language in menus and such.

22.3.6.2 Inputting Non-English Characters

The X11 Input Method (XIM) Protocol is a new standard for all X11 clients. All X11 applications should be written as XIM clients that take input from XIM Input servers. There are several XIM servers available for different languages.

22.3.7 Printer Setup

Some single C chars character sets are usually hardware coded into printers. Wide or multibyte character sets require special setup and we recommend using **apsfilter**. You may also convert the document to PostScript or PDF formats using language specific converters.

22.3.8 Kernel and File Systems

The FreeBSD fast filesystem (FFS) is 8-bit clean, so it can be used with any single C chars character set (see multibyte(3)), but there is no character set name stored in the filesystem; i.e., it is raw 8-bit and does not know anything about encoding order. Officially, FFS does not support any form of wide or multibyte character sets yet. However, some wide or multibyte character sets have independent patches for FFS enabling such support. They are only temporary unportable solutions or hacks and we have decided to not include them in the source tree. Refer to respective languages' web sites for more information and the patch files.

The FreeBSD MS-DOS filesystem has the configurable ability to convert between MS-DOS, Unicode character sets and chosen FreeBSD filesystem character sets. See mount_msdos(8) for details.

22.4 Compiling I18N Programs

Many FreeBSD Ports have been ported with I18N support. Some of them are marked with -I18N in the port name. These and many other programs have built in support for I18N and need no special consideration.

However, some applications such as **MySQL** need to be have the `Makefile` configured with the specific charset. This is usually done in the `Makefile` or done by passing a value to **configure** in the source.

22.5 Localizing FreeBSD to Specific Languages

22.5.1 Russian Language (KOI8-R Encoding)

Originally contributed by Andrey Chernov.

For more information about KOI8-R encoding, see the KOI8-R References (Russian Net Character Set) (http://koi8.pp.ru/).

22.5.1.1 Locale Setup

Put the following lines into your `~/.login_conf` file:

```
me:My Account:\
 :charset=KOI8-R:\
 :lang=ru_RU.KOI8-R:
```

See earlier in this chapter for examples of setting up the locale.

22.5.1.2 Console Setup

- Add the following line to your `/etc/rc.conf` file:

  ```
  mousechar_start=3
  ```

- Also, use following settings in `/etc/rc.conf`:

  ```
  keymap="ru.koi8-r"
  scrnmap="koi8-r2cp866"
  font8x16="cp866b-8x16"
  font8x14="cp866-8x14"
  font8x8="cp866-8x8"
  ```

- For each `ttyv*` entry in `/etc/ttys`, use `cons25r` as the terminal type.

See earlier in this chapter for examples of setting up the console.

22.5.1.3 Printer Setup

Since most printers with Russian characters come with hardware code page CP866, a special output filter is needed to convert from KOI8-R to CP866. Such a filter is installed by default as `/usr/libexec/lpr/ru/koi2alt`. A Russian printer `/etc/printcap` entry should look like:

```
lp|Russian local line printer:\
 :sh:of=/usr/libexec/lpr/ru/koi2alt:\
 :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

See printcap(5) for a detailed description.

22.5.1.4 MS-DOS FS and Russian Filenames

The following example fstab(5) entry enables support for Russian filenames in mounted MS-DOS filesystems:

```
/dev/ad0s2      /dos/c  msdos   rw,-Wkoi2dos,-Lru_RU.KOI8-R 0 0
```

The option `-L` selects the locale name used, and `-W` sets the character conversion table. To use the `-W` option, be sure to mount `/usr` before the MS-DOS partition because the conversion tables are located in `/usr/libdata/msdosfs`. For more information, see the mount_msdos(8) manual page.

22.5.1.5 X11 Setup

1. Do non-X locale setup first as described.
2. If you use **Xorg**, install `x11-fonts/xorg-fonts-cyrillic` package.

   Check the `"Files"` section in your `/etc/X11/xorg.conf` file. The following lines must be added *before* any other `FontPath` entries:

   ```
   FontPath   "/usr/X11R6/lib/X11/fonts/cyrillic/misc"
   FontPath   "/usr/X11R6/lib/X11/fonts/cyrillic/75dpi"
   FontPath   "/usr/X11R6/lib/X11/fonts/cyrillic/100dpi"
   ```

   If you use a high resolution video mode, swap the 75 dpi and 100 dpi lines.
3. To activate a Russian keyboard, add the following to the `"Keyboard"` section of your `xorg.conf` file.

   ```
   Option "XkbLayout"   "us,ru"
   Option "XkbOptions"  "grp:toggle"
   ```

   Also make sure that `XkbDisable` is turned off (commented out) there.

   For `grp:caps_toggle` the RUS/LAT switch will be **CapsLock**. The old **CapsLock** function is still available via **Shift**+**CapsLock** (in LAT mode only). For `grp:toggle` the RUS/LAT switch will be **Right Alt**. `grp:caps_toggle` does not work in **Xorg** for unknown reason.

   If you have “Windows” keys on your keyboard, and notice that some non-alphabetical keys are mapped incorrectly in RUS mode, add the following line in your `xorg.conf` file.

   ```
   Option "XkbVariant" ",winkeys"
   ```

   Note: The Russian XKB keyboard may not work with non-localized applications.

Note: Minimally localized applications should call a `XtSetLanguageProc (NULL, NULL, NULL);` function early in the program.

See KOI8-R for X Window (http://koi8.pp.ru/xwin.html) for more instructions on localizing X11 applications.

22.5.2 Traditional Chinese Localization for Taiwan

The FreeBSD-Taiwan Project has an Chinese HOWTO for FreeBSD at http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/ using many Chinese ports. Current editor for the `FreeBSD Chinese HOWTO` is Shen Chuan-Hsing <`statue@freebsd.sinica.edu.tw`>.

Chuan-Hsing Shen <`statue@freebsd.sinica.edu.tw`> has created the Chinese FreeBSD Collection (CFC) (http://netlab.cse.yzu.edu.tw/~statue/cfc/) using FreeBSD-Taiwan's `zh-L10N-tut`. The packages and the script files are available at ftp://freebsd.csie.nctu.edu.tw/pub/taiwan/CFC/.

22.5.3 German Language Localization (for All ISO 8859-1 Languages)

Slaven Rezic <`eserte@cs.tu-berlin.de`> wrote a tutorial how to use umlauts on a FreeBSD machine. The tutorial is written in German and available at http://www.de.FreeBSD.org/de/umlaute/.

22.5.4 Japanese and Korean Language Localization

For Japanese, refer to http://www.jp.FreeBSD.org/, and for Korean, refer to http://www.kr.FreeBSD.org/.

22.5.5 Non-English FreeBSD Documentation

Some FreeBSD contributors have translated parts of FreeBSD to other languages. They are available through links on the main site (http://www.FreeBSD.org/index.html) or in `/usr/share/doc`.

Chapter 23 更新、升級FreeBSD

Restructured, reorganized, and parts updated by Jim Mock. Original work by Jordan Hubbard, Poul-Henning Kamp, John Polstra, and Nik Clayton.

23.1 概述

FreeBSD 是個持續發展的作業系統。對於喜歡追求新鮮、刺激的使用者而言，有很多方法可以使您的系統輕鬆更新為最新版。注意：並非每個人都適合這麼做！　本章主要是協助您決定到底要跟開發版本，或是要使用較穩定的釋出版。

讀完這章，您將了解：

- FreeBSD-STABLE 與FreeBSD-CURRENT　這兩分支的不同之處；
- 如何以**CSup**, **CVSup**, **CVS** 或**CTM** 來更新你的系統
- 如何以`make buildworld` 等指令來重新編譯、安裝整個base system。

在開始閱讀這章之前，您需要：

- 先設好你的網路(Chapter 29)。
- 知道如何透過port/package 安裝軟體(Chapter 4)。

23.2 FreeBSD-CURRENT vs. FreeBSD-STABLE

FreeBSD 有兩個發展分支：FreeBSD-CURRENT 及FreeBSD-STABLE。本節將會陸續介紹，並介紹它們分別又是如何更新。首先，先介紹FreeBSD-CURRENT，接著再介紹FreeBSD-STABLE。

23.2.1 使用最新的FreeBSD CURRENT

這裡再次强調，FreeBSD-CURRENT 是FreeBSD 開發的“最前線”。FreeBSD-CURRENT 使用者須有較强的技術能力，而且應該要有能力自己解決困難的系統問題。若您是FreeBSD 新手，那麼請在安裝前最好先三思。

23.2.1.1 什麼是FreeBSD-CURRENT？

FreeBSD-CURRENT 是FreeBSD 的最新版。它包含：仍在研發階段、實驗性質的修改、過渡時期的機制，這些東西在下一次正式relase 的版本可能會有，也可能不會有的。儘管有許多FreeBSD 開發者每天都會編譯FreeBSD-CURRENT source code，但有時這些原始碼是無法編譯成功。雖然，這些問題通常會儘快解決，但FreeBSD-CURRENT 到底是帶來浩劫或是多了想要用的新功能、改善，這點主要取決於您更新原始碼的時機為何而定！

23.2.1.2 誰需要FreeBSD-CURRENT？

FreeBSD-CURRENT 適合下列這三類人：

1. FreeBSD 社群成員：積極專注於source tree 的某一部份，以及認為保持為“current(最新狀態)” 為絕對需求的人。
2. FreeBSD 社群成員：為了確保FreeBSD-CURRENT 能夠儘可能地維持在最穩定的狀態，而主動花時間解決問題的測試者。此外，還有對FreeBSD 能提出具體建議以及改善方向，並提出patch 修正檔的人。
3. 只是關心或者想參考(比如，只是*閱讀*，而非執行)的人。這些人有時也會做些註解，或貢獻原始碼。

23.2.1.3 FreeBSD-CURRENT *並不是* 什麼？

1. 追求最新功能。聽說裡面有些很酷的新功能，並希望成為您周圍的人中第一個嘗試的人，因此將FreeBSD-CURRENT 視為取得搶鮮版的捷徑。儘管，您能夠因此首先瞭解到最新的功能，但這也意味著若出現新的bug 時，您也是首當其衝。
2. 修復bug 的速成法。因為FreeBSD-CURRENT 的任何版本在修復已知bug 的同時，又可能會產生新的bug。
3. 無所不在的“officially supported”。我們會盡力協助上述FreeBSD-CURRENT 的那三種類別的“legitimate” 使用者，但我們*沒時間*為他們提供技術支援。這不代表我們很惡劣，或是不想幫助人(若是的話，我們也不會為FreeBSD 努力了) ，實在是因為我們分身乏術，無法每天回答數百個問題，*而同時*繼續開發FreeBSD。可以確定的一點就是，在改善FreeBSD 或是回答大量有關實驗碼的問題之間，若要做個選擇的話，開發者會選擇前者。

23.2.1.4 使用FreeBSD-CURRENT

1. 加入freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) 及cvs-all (http://lists.FreeBSD.org/mailman/listinfo/cvs-all) 論壇。這不單只是個建議，也是*必須* 作的。若您沒訂閱*freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current)* ，那麼就會錯過別人對目前系統狀態的說明，而枯耗在別人已解的問題。更重要的是，可能會錯失一些對己身所管系統安危相當重要的公告。

   在cvs-all (http://lists.FreeBSD.org/mailman/listinfo/cvs-all) 上則可以看到每個commit 紀錄，因為這些記錄會連帶影響其他相關資訊。

   要訂閱這些論壇或其他論壇，請參考http://lists.FreeBSD.org/mailman/listinfo 並點選想訂閱的部分即可。至於其他後續步驟如何進行，在那裡會有說明。
2. 從FreeBSD mirror 站 取得原始碼。有兩種方式可以達成：
   a. 以csup 或cvsup 程式搭配位於`/usr/share/examples/cvsup` 檔名為`standard-supfile` 的`supfile`。這是大家最常推薦的方式，因為它可以讓您把整個tree 都抓回來，之後就只取有更新的部分即可。此外，許多人會把`csup` 或`cvsup` 放到`cron` 以定期自動更新。您須要自訂前述的`supfile` 範例檔，並針對自身網路環境以調整csup 或cvsup 相關設定。
   b. 使用**CTM** 工具。若網路環境不佳(上網費用貴，或只能用email 而已) **CTM** 會比較適合您的需求。然而，這也有一些爭議並且常抓到一些有問題的檔案。因此，很少人會用它。這也註定了不能長期依賴這個更新方式。若是使用9600 bps modem 或頻寬更大的上網者，建議使用**CVSup** 。

3. 若抓source code 是要用來跑的，而不僅只是看看而已，那麼就抓*整個* FreeBSD-CURRENT，而不要只抓部分。因為大部分的source code 都會相依到其他source code 環節部分，若是您只編譯其中一部份，保證會很麻煩。

   在編譯FreeBSD-CURRENT 之前，請仔細閱讀`/usr/src` 內的`Makefile`。儘管只是升級部分東西而已，您至少也要先 裝新的kernel 以及重新編譯world。此外，多多閱讀FreeBSD-CURRENT 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) 以及`/usr/src/UPDATING` 也是必須的，才能知道目前進度是怎樣以及下一版會有什麼新東西。

4. 熱血！若您正在跑FreeBSD-CURRENT，我們很想知道您對於它的想法是什麼，尤其是加强哪些功能，或該修正哪些錯誤的建議。如果您在建議時能附上相關程式碼的話，那真是太棒了！

23.2.2 使用最新的FreeBSD STABLE

23.2.2.1 什麼是FreeBSD-STABLE？

FreeBSD-STABLE 是我們的開發分支，主要的發行版就由此而來。這個分支會以不同速度作修改變化，並且假設這些是第一次進入FreeBSD-CURRENT 進行測試。然而，這*仍然* 屬於開發中的分支，也就是說在某些時候，FreeBSD-STABLE 可能會、也可能不會符合一些特殊需求。它只不過是另一個開發分支而已，可能不太適合一般使用者。

23.2.2.2 誰需要FreeBSD-STABLE？

若您有興趣去追蹤、貢獻FreeBSD 開發過程或作些貢獻，尤其是會跟FreeBSD 接下來的“關鍵性” 發行有關，應該考慮採用FreeBSD-STABLE。

雖然安全漏洞的修補也會進入FreeBSD-STABLE 分支，但不必僅僅因此而*需要* 去用FreeBSD-STABLE。FreeBSD 每項security advisory(安全公告) 都會解說如何去修復有受到影響的版本[1]，若僅因為安全因素而去採用開發分支，雖然會解決現有已知問題，但也可能帶來一些潛藏的問題。

儘管我們盡力確保FreeBSD-STABLE 分支在任何時候均能正確編譯、運作，但沒人能夠擔保它隨時都可以符合上述目的。此外，雖然原始碼在進入FreeBSD-STABLE 之前，都會先在FreeBSD-CURRENT 開發完畢，但使用FreeBSD-CURRENT 的人畢竟遠比FreeBSD-STABLE 使用者來的少，所以通常有些問題，可能在FreeBSD-CURRENT 比較沒人注意到，隨著FreeBSD-STABLE 使用者的廣泛使用才會浮現。

由於上述這些理由，我們*並不推薦* 盲目追隨FreeBSD-STABLE，而且更重要的是，別在原始碼尚未經完整測試之前，就衝動把production server 轉移到FreeBSD-STABLE 環境。

若您沒有這些多的時間、精神的話，那推薦您使用最新的FreeBSD 發行版即可，並採用其所提供的binary 更新機制來完成升級轉移。

23.2.2.3 使用FreeBSD-STABLE

1. 訂閱freebsd-stable (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable) list。可以讓您隨時瞭解FreeBSD-STABLE 的軟體編譯時的相依關係，以及其他需特別注意的問題。開發者在考慮一些有爭議的修正或更新時，就會先在這裡發信說明，給使用者有機會可以反應，看他們對所提的更改是否有什麼建議或問題。

而cvs-all (http://lists.FreeBSD.org/mailman/listinfo/cvs-all) list 這邊可以看到每個commit log，其中包括了許多中肯的資訊，例如一些可能發生的邊際效應等等。

想要加入這些通信論壇的話，只要到http://lists.FreeBSD.org/mailman/listinfo 點下想訂閱的list 即可。其餘的步驟在網頁上會有說明。

2. 若打算要安裝一個全新的系統，並且希望裝FreeBSD-STABLE 每月定期的snapshot，那麼請參閱Snapshots (http://www.FreeBSD.org/snapshots/) 網頁以瞭解相關細節。此外，也可從mirror 站 來安裝最新的FreeBSD-STABLE 發行版，並透過下列的的說明來更新到最新的FreeBSD-STABLE 原始碼。

   若已裝的是FreeBSD 以前的版本，而想透過原始碼方式來升級，那麼也是可以利用FreeBSD mirror 站 來完成。以下介紹兩種方式：

   a. 以csup 或cvsup 程式搭配位於`/usr/share/examples/cvsup` 檔名為`stable-supfile` 的`supfile`。這是大家最常推薦的方式，因為它可以讓你把整個tree 都抓回來，之後就只取有更新的部分即可。此外，許多人會把`csup` 或`cvsup` 放到`cron` 以定期自動更新。您須要自訂前述的`supfile` 範例檔，並針對自身網路環境以調整csup 或cvsup 相關設定。

   b. 使用**CTM** 更新工具。若網路不快或網路費用貴，那麼可以考慮採用。

3. 一般而言，若常需存取最新原始碼，而不計較網路頻寬的話，可以使用`csup` 或`cvsup` 或`ftp`。否則，就考慮**CTM**。

4. 在編譯FreeBSD-STABLE 之前，請先仔細閱讀`/usr/src` 內的`Makefile` 檔。儘管只是升級部分東西而已，您至少也要先裝新的kernel 以及重新編譯world。此外，多多閱讀FreeBSD-STABLE 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable) 以及`/usr/src/UPDATING` 也是必備的，這樣才能知道目前進度是怎樣，以及下一版會有哪些新東西。

23.3 更新你的Source

FreeBSD 計劃原始碼有許多透過網路(或email)的方式來更新，無論是更新那一塊領域，這些全由您自行決定。我們主要提供的是Anonymous CVS、CVSup 、CTM。

> **Warning:** 雖然可以只更新部分原始碼，但唯一支援的更新流程是更新整個tree，並且重編userland(比如：由使用者去執行的所有程式，像是`/bin`、`/sbin` 內的程式)以及kernel 原始碼。若只更新部分的source tree、或只有kernel 部分、或只有userland 部分，通常會造成一些錯誤，像是：編譯錯誤、kernel panic、資料毀損等。

Anonymous CVS 及**CVSup** 均是採*pull* 模式來更新原始碼。以**CVSup** 為例，使用者(或`cron` script)會執行`cvsup` 程式，後者會與某一台`cvsupd` 伺服器作些互動，以更新相關原始碼檔案。您所收到更新會是當時最新的，而且只會收到需更新的部分。此外，也可以很輕鬆去設定要更新的範圍。更新會由伺服器跟本機比對之後，丟出當時您所需要的更新檔案給你。**Anonymous CVS** 的概念相對於**CVSup** 來得更簡單些，因為它只是**CVS** 的延伸而已，一樣讓你可從遠端的CVS repository 取出最新原始碼。然而**CVSup** 在這方面會更有效率，不過**Anonymous CVS** 對新手而言，是用起來比較簡單。

另一種方式則是**CTM**。它並不是以交談式介面來比對您所擁有的sources 和伺服器上的sources 或是您取得的更新部份。相反的，會有一個script 檔專門用來辨識變更過的檔案，這個程式是由CTM 伺服器來執行，每天會比對數次，並把兩次執行期間內變更過的檔案加以壓縮，並給它們一個序號，然後就加以編碼(只用printable ASCII 字元)，並以email 的方式寄出。當您收到它的時候，這些“CTM deltas” 就可以

由ctm_rmail(1) 程式來處理，該程式會自動解碼、確認、套用這些變更。這程序比**CVSup** 來說是快得多了，而且，這個模式對我們的伺服器來說是比較輕鬆的，因為這是一個*push* 的模式，而非*pull* 的模式。

當然，這樣做也會帶來一些不便。若不小心把您部份的程式清除掉了，**CVSup** 會偵測出來，並自動為您把不足的部份補齊。**CTM** 並不會為您做這些動作。若清掉了您的部份source (而且沒備份)，您可以從頭開始(從最新的CVS “base delta”)並用**CTM** 來重建它們，或是用**Anonymous CVS** 來完成，只要把不正確的地方砍掉，再重新做同步的動作即可。

23.4 重新編譯“world”

在更新FreeBSD 的source tree 到最新之後(無論是FreeBSD-STABLE、FreeBSD-CURRENT 等等)，接下來就可以用這些source tree 來重新編譯系統。

> **做好備份:** 在作任何大動作之*前* 要記得先把系統作備份的重要性無須強調。儘管重新編譯world 是(只要有照文件指示去作的話)一件很簡單的事情，但出錯也是在所難免的。另外，別人在source tree 不慎搞混的錯誤，也可能會造成系統無法開機。
>
> 請確認自己已作妥相關備份，並且手邊有fixit 磁片或開機光碟。您可能永遠也用不到這些東西，但安全第一總比事後說抱歉來得好吧！

> **訂閱相關的Mailing List:** FreeBSD-STABLE 以及FreeBSD-CURRENT 分支，本質上就是屬於 *開發階段*。為FreeBSD 作貢獻的也都是人，偶爾也會犯錯誤。
>
> 有時候這些錯誤並無大礙，只是會讓系統產生新的錯誤警告而已。有時則是災難，可能會導致不能開機或檔案系統的毀損(或更糟)。
>
> 若遇到類似問題，貼封標題為“heads up(注意)” 開頭的信到相關的mailing list，並講清楚問題點以及會影響哪些系統。在問題獲解決後，再貼標題為“all clear(已解決)” 開頭的聲明信。
>
> 若用的是FreeBSD-STABLE 或FreeBSD-CURRENT，卻又不閱讀FreeBSD-STABLE 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable) 或FreeBSD-CURRENT 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) 的討論，那麼會是自找麻煩而已。

> **不要用`make world`:** 一堆早期的舊文件都會建議說使用`make world`。這樣做會跳過一些重要步驟，建議只有在你知道自己在作什麼，再這麼做。在絕大多數的情況下，請不要亂用`make world`，而該改用下面介紹的方式。

23.4.1 更新系統的標準方式

要升級系統前，一定要先查閱`/usr/src/UPDATING` 文件，以瞭解buildworld 之前需要作哪些事情或注意事項，然後才用下列步驟：

```
# make buildworld
# make buildkernel
# make installkernel
# reboot
```

Note: 在少數狀況，可能需要先在`buildworld` 步驟之前先作`mergemaster -p` 才能完成。至於何時需要或不需要，請參閱`UPDATING` 內的說明。一般來說，只要不是進行跨版號(major)的FreeBSD 版本升級，就可略過這步驟。

完成`installkernel` 之後，需要重開機並切到single user 模式(舉例：也可以在loader 提示符號後面加上`boot -s`)。接下來執行：

```
# mergemaster -p
# make installworld
# mergemaster
# reboot
```

Read Further Explanations: 上述步驟只是協助您升級的簡單說明而已，若要清楚瞭解每一步驟，尤其是若欲自行打造kernel 設定，就更該閱讀下面的內容。

23.4.2 閱讀`/usr/src/UPDATING`

在作任何事情之前，請務必先閱讀`/usr/src/UPDATING` (或在source code 內類似的文件) 。這份文件會寫到可能遭遇的問題，或指定那些會執行的指令順序為何。如果你機器現在的`UPDATING` 文件與這邊的描述有衝突、矛盾之處，那麼請以機器上的`UPDATING` 為準。

Important: 然而，如同先前所述，單單只靠閱讀`UPDATING` 並不能完全取代mailing list。這兩者都是互補的，而不相排斥。

23.4.3 檢查`/etc/make.conf`

檢查`/usr/share/examples/etc/make.conf` 以及`/etc/make.conf`。第一份文件乃是一些系統預設值– 不過，大部分都被註解起來。為了在重新編譯時能夠使用這些，請把這些設定加到`/etc/make.conf`。請注意在`/etc/make.conf` 的任何設定也會影響到每次使用`make` 的結果，因此設定一些適合自己系統的選項會是不錯的作法。

一般使用者通常會從`/usr/share/examples/etc/make.conf` 複製`CFLAGS` 以及`NO_PROFILE` 之類的設定到`/etc/make.conf`，並解除相關註解印記。

此外，也可以試試看其他設定(`COPTFLAGS`、`NOPORTDOCS` 等等)，是否符合自己所需。

23.4.4 更新`/etc` 內的設定檔

在`/etc` 目錄會有系統的相關設定檔，以及開機時的各項服務啓動script。有些script 隨FreeBSD 版本的不同而有些差異。

其中有些設定檔會在每日運作的系統裡也會用到。尤其是`/etc/group`。

有時候在`make installworld` 安裝過程中，會需要先建立某些特定帳號或群組。在進行升級之前，它們可能並不存在，因此升級時就會造成問題。有時候`make buildworld` 會先檢查這些所需的帳號或群組是否已有存在。

舉個這樣的例子，像是某次升級之後必須新增`smmsp` 帳號。若使用者尚未新增該帳號就要完成升級操作的話，會在mtree(8) 嘗試建立`/var/spool/clientmqueue` 時發生失敗。

解法是在buildworld 階段之前，先執行mergemaster(8) 並搭配`-p` 選項。它會比對那些執行`buildworld` 或`installworld` 所需之關鍵設定檔。若你所用的是早期仍未支援`-p` 的`mergemaster` 版本，那麼直接使用source tree 內的新版即可：

```
# cd /usr/src/usr.sbin/mergemaster
# ./mergemaster.sh -p
```

> **Tip:** 若您是偏執狂(paranoid)，可以像下面這樣去試著檢查系統上有哪些檔案屬於已改名或被刪除的群組：
>
> ```
> # find / -group GID -print
> ```
>
> 這會顯示所有符合要找的*`GID`* 群組(可以是群組名稱，或者是群組的數字代號)的所有檔案。

23.4.5 切換到Single User 模式

您可能會想在single user 模式下編譯系統。除了可以明顯更快完成之外，安裝過程中將會牽涉許多重要的系統檔案，包括所有系統binaries、libraries、include 檔案等。若在運作中的系統(尤其有許多使用者在用的時候)內更改這些檔案，那簡直是自找麻煩的作法。

另一種模式是先在multi-user 模式下編譯好系統，然後再切到single user 模式去安裝。若您比較喜歡這種方式，只需在build(編譯過程) 完成之後，再去執行下面的步驟即可。一直到可切換single user 模式時，再去執行`installkernel` 或`installworld` 即可。

切換為root 身份打：

```
# shutdown now
```

這樣就會從原本的multi-user 模式切換到single user 模式。

除此之外也可以重開機，接著在開機選單處選擇“single user” 選項。如此一來就會進入single user 模式，然後在shell 提示符號處輸入：

```
# fsck -p
# mount -u /
# mount -a -t ufs
# swapon -a
```

這樣會先檢查檔案系統，並重新將`/` 改以可讀寫的模式掛載，以及`/etc/fstab` 內所設定的其他UFS 檔案系統，最後啓用swap 磁區。

> **Note:** 若CMOS 時鐘是設為當地時間，而非GMT 時區(若date(1) 指令沒顯示正確的時間、時區)，那可能需要再輸入下列指令：
>
> ```
> # adjkerntz -i
> ```

這步驟可以確認您的當地時區設定是否正確——否則日後會造成一些問題。

23.4.6 移除/usr/obj

在重新編譯系統的過程中，編譯結果會放到(預設情況) /usr/obj 內。這裡面的目錄會對應到/usr/src 的目錄結構。

砍掉這目錄，可以讓以後的make buildworld 過程更快一些，而且可避免以前編譯的東西跟現在的混淆在一起的相依錯亂。

而有些/usr/obj 內的檔案可能會設定不可更動的flag(細節請參閱chflags(1))，而必須先拿掉這些flag 設定才行。

```
# cd /usr/obj
# chflags -R noschg *
# rm -rf *
```

23.4.7 重新編譯Base System

23.4.7.1 保留編譯的紀錄

建議養成好習慣，把執行make(1) 時產生的紀錄存起來。這樣若有哪邊出錯，就會有錯誤訊息的紀錄。雖然單單這樣，你可能不知道如何分析是哪邊出了岔，但若把你問題記錄貼到FreeBSD 相關的mailing list 就可以有人可以幫忙看是怎麼一回事情。

最簡單的方是就是用script(1) 指令，並加上參數(你想存放記錄的檔案位置、檔名)即可。這步驟應該在重新編譯系統時就要作，然後在完成編譯後輸入**exit** 即可離開。

```
# script /var/tmp/mw.out
Script started, output file is /var/tmp/mw.out
# make TARGET
... compile, compile, compile ...
# exit
Script done, ...
```

對了，還有一點儘量*別把*檔案存到/tmp 目錄內。因為重開機之後，這目錄內的東西都會被清空。比較妥善的地方是/var/tmp (如上例所示) 或者是root 的家目錄。

23.4.7.2 編譯Base System

首先請先切換到/usr/src 目錄：

```
# cd /usr/src
```

(當然，除非你把source code 放到其他地方，若真是這樣，就切換到那個目錄即可)。

使用make(1) 指令來重新編譯world。這指令會從Makefile 檔(這檔會寫FreeBSD 的程式該如何重新編譯、以哪些順序來編譯等等)去讀取相關指令。

一般下指令的格式如下：

```
# make -x -DVARIABLE target
```

在這個例子，*-x* 是你想傳給make(1) 的選項，細節說明請參閱make(1) 說明，裡面有相關範例說明。

-DVARIABLE 則是把變數設定傳給Makefile。這些變數會控制Makefile 的行為。這些設定與/etc/make.conf 的變數設定是一樣，只是另一種設定方式而已。

```
# make -DNO_PROFILE target
```

上面的例子則是另一種設定方式，也就是哪些不要。這個例子中的意思是不去編譯profiled libraries，效果就如同設定在/etc/make.conf 的

```
NO_PROFILE=    true  #    Avoid compiling profiled libraries
```

target 則是告訴make(1) 該去做哪些。每個Makefile 都會定義不同的“targets”，然後依您所給的target 就會決定會做哪些動作。

Some targets are listed in the Makefile, but are not meant for you to run. Instead, they are used by the build process to break out the steps necessary to rebuild the system into a number of sub-steps.

Most of the time you will not need to pass any parameters to make(1), and so your command like will look like this:

```
# make target
```

Where *target* will be one of many build options. The first target should always be buildworld.

As the names imply, buildworld builds a complete new tree under /usr/obj, and installworld, another target, installs this tree on the current machine.

Having separate options is very useful for two reasons. First, it allows you to do the build safe in the knowledge that no components of your running system will be affected. The build is “self hosted”. Because of this, you can safely run buildworld on a machine running in multi-user mode with no fear of ill-effects. It is still recommended that you run the installworld part in single user mode, though.

Secondly, it allows you to use NFS mounts to upgrade multiple machines on your network. If you have three machines, A, B and C that you want to upgrade, run make buildworld and make installworld on A. B and C should then NFS mount /usr/src and /usr/obj from A, and you can then run make installworld to install the results of the build on B and C.

Although the world target still exists, you are strongly encouraged not to use it.

Run

```
# make buildworld
```

It is possible to specify a -j option to make which will cause it to spawn several simultaneous processes. This is most useful on multi-CPU machines. However, since much of the compiling process is IO bound rather than CPU bound it is also useful on single CPU machines.

On a typical single-CPU machine you would run:

```
# make -j4 buildworld
```

make(1) will then have up to 4 processes running at any one time. Empirical evidence posted to the mailing lists shows this generally gives the best performance benefit.

If you have a multi-CPU machine and you are using an SMP configured kernel try values between 6 and 10 and see how they speed things up.

23.4.7.3 Timings

Many factors influence the build time, but fairly recent machines may only take a one or two hours to build the FreeBSD-STABLE tree, with no tricks or shortcuts used during the process. A FreeBSD-CURRENT tree will take somewhat longer.

23.4.8 Compile and Install a New Kernel

To take full advantage of your new system you should recompile the kernel. This is practically a necessity, as certain memory structures may have changed, and programs like ps(1) and top(1) will fail to work until the kernel and source code versions are the same.

The simplest, safest way to do this is to build and install a kernel based on `GENERIC`. While `GENERIC` may not have all the necessary devices for your system, it should contain everything necessary to boot your system back to single user mode. This is a good test that the new system works properly. After booting from `GENERIC` and verifying that your system works you can then build a new kernel based on your normal kernel configuration file.

On FreeBSD it is important to build world before building a new kernel.

> **Note:** If you want to build a custom kernel, and already have a configuration file, just use `KERNCONF=MYKERNEL` like this:

```
# cd /usr/src
# make buildkernel KERNCONF=MYKERNEL
# make installkernel KERNCONF=MYKERNEL
```

Note that if you have raised `kern.securelevel` above 1 *and* you have set either the `noschg` or similar flags to your kernel binary, you might find it necessary to drop into single user mode to use `installkernel`. Otherwise you should be able to run both these commands from multi user mode without problems. See init(8) for details about `kern.securelevel` and chflags(1) for details about the various file flags.

23.4.9 Reboot into Single User Mode

You should reboot into single user mode to test the new kernel works. Do this by following the instructions in Section 23.4.5.

23.4.10 Install the New System Binaries

If you were building a version of FreeBSD recent enough to have used `make buildworld` then you should now use `installworld` to install the new system binaries.

Run

```
# cd /usr/src
# make installworld
```

> **Note:** If you specified variables on the `make buildworld` command line, you must specify the same variables in the `make installworld` command line. This does not necessarily hold true for other options; for example, `-j` must never be used with `installworld`.
>
> For example, if you ran:
>
> ```
> # make -DNO_PROFILE buildworld
> ```
>
> you must install the results with:
>
> ```
> # make -DNO_PROFILE installworld
> ```
>
> otherwise it would try to install profiled libraries that had not been built during the `make buildworld` phase.

23.4.11 Update Files Not Updated by `make installworld`

Remaking the world will not update certain directories (in particular, `/etc`, `/var` and `/usr`) with new or changed configuration files.

The simplest way to update these files is to use mergemaster(8), though it is possible to do it manually if you would prefer to do that. Regardless of which way you choose, be sure to make a backup of `/etc` in case anything goes wrong.

23.4.11.1 `mergemaster`

Contributed by Tom Rhodes.

The mergemaster(8) utility is a Bourne script that will aid you in determining the differences between your configuration files in `/etc`, and the configuration files in the source tree `/usr/src/etc`. This is the recommended solution for keeping the system configuration files up to date with those located in the source tree.

To begin simply type `mergemaster` at your prompt, and watch it start going. `mergemaster` will then build a temporary root environment, from `/` down, and populate it with various system configuration files. Those files are then compared to the ones currently installed in your system. At this point, files that differ will be shown in diff(1) format, with the + sign representing added or modified lines, and - representing lines that will be either removed completely, or replaced with a new line. See the diff(1) manual page for more information about the diff(1) syntax and how file differences are shown.

mergemaster(8) will then show you each file that displays variances, and at this point you will have the option of either deleting the new file (referred to as the temporary file), installing the temporary file in its unmodified state, merging the temporary file with the currently installed file, or viewing the diff(1) results again.

Choosing to delete the temporary file will tell mergemaster(8) that we wish to keep our current file unchanged, and to delete the new version. This option is not recommended, unless you see no reason to change the current file. You can get help at any time by typing **?** at the mergemaster(8) prompt. If the user chooses to skip a file, it will be presented again after all other files have been dealt with.

Choosing to install the unmodified temporary file will replace the current file with the new one. For most unmodified files, this is the best option.

Choosing to merge the file will present you with a text editor, and the contents of both files. You can now merge them by reviewing both files side by side on the screen, and choosing parts from both to create a finished product. When the files are compared side by side, the **l** key will select the left contents and the **r** key will select contents from your right. The final output will be a file consisting of both parts, which can then be installed. This option is customarily used for files where settings have been modified by the user.

Choosing to view the diff(1) results again will show you the file differences just like mergemaster(8) did before prompting you for an option.

After mergemaster(8) is done with the system files you will be prompted for other options. mergemaster(8) may ask if you want to rebuild the password file and will finish up with an option to remove left-over temporary files.

23.4.11.2 Manual Update

If you wish to do the update manually, however, you cannot just copy over the files from `/usr/src/etc` to `/etc` and have it work. Some of these files must be "installed" first. This is because the `/usr/src/etc` directory *is not* a copy of what your `/etc` directory should look like. In addition, there are files that should be in `/etc` that are not in `/usr/src/etc`.

If you are using mergemaster(8) (as recommended), you can skip forward to the next section.

The simplest way to do this by hand is to install the files into a new directory, and then work through them looking for differences.

> **Backup Your Existing `/etc`:** Although, in theory, nothing is going to touch this directory automatically, it is always better to be sure. So copy your existing `/etc` directory somewhere safe. Something like:
>
> ```
> # cp -Rp /etc /etc.old
> ```
>
> `-R` does a recursive copy, `-p` preserves times, ownerships on files and suchlike.

You need to build a dummy set of directories to install the new `/etc` and other files into. `/var/tmp/root` is a reasonable choice, and there are a number of subdirectories required under this as well.

```
# mkdir /var/tmp/root
# cd /usr/src/etc
# make DESTDIR=/var/tmp/root distrib-dirs distribution
```

This will build the necessary directory structure and install the files. A lot of the subdirectories that have been created under `/var/tmp/root` are empty and should be deleted. The simplest way to do this is to:

```
# cd /var/tmp/root
# find -d . -type d | xargs rmdir 2>/dev/null
```

This will remove all empty directories. (Standard error is redirected to `/dev/null` to prevent the warnings about the directories that are not empty.)

`/var/tmp/root` now contains all the files that should be placed in appropriate locations below `/`. You now have to go through each of these files, determining how they differ with your existing files.

Note that some of the files that will have been installed in `/var/tmp/root` have a leading “.”. At the time of writing the only files like this are shell startup files in `/var/tmp/root/` and `/var/tmp/root/root/`, although there may be others (depending on when you are reading this). Make sure you use `ls -a` to catch them.

The simplest way to do this is to use diff(1) to compare the two files:

```
# diff /etc/shells /var/tmp/root/etc/shells
```

This will show you the differences between your `/etc/shells` file and the new `/var/tmp/root/etc/shells` file. Use these to decide whether to merge in changes that you have made or whether to copy over your old file.

Name the New Root Directory (`/var/tmp/root`) with a Time Stamp, so You Can Easily Compare Differences Between Versions: Frequently rebuilding the world means that you have to update `/etc` frequently as well, which can be a bit of a chore.

You can speed this process up by keeping a copy of the last set of changed files that you merged into `/etc`. The following procedure gives one idea of how to do this.

1. Make the world as normal. When you want to update `/etc` and the other directories, give the target directory a name based on the current date. If you were doing this on the 14th of February 1998 you could do the following:

   ```
   # mkdir /var/tmp/root-19980214
   # cd /usr/src/etc
   # make DESTDIR=/var/tmp/root-19980214 \
       distrib-dirs distribution
   ```

2. Merge in the changes from this directory as outlined above.

   Do not remove the `/var/tmp/root-19980214` directory when you have finished.

3. When you have downloaded the latest version of the source and remade it, follow step 1. This will give you a new directory, which might be called `/var/tmp/root-19980221` (if you wait a week between doing updates).

4. You can now see the differences that have been made in the intervening week using diff(1) to create a recursive diff between the two directories:

   ```
   # cd /var/tmp
   # diff -r root-19980214 root-19980221
   ```

   Typically, this will be a much smaller set of differences than those between `/var/tmp/root-19980221/etc` and `/etc`. Because the set of differences is smaller, it is easier to migrate those changes across into your `/etc` directory.

5. You can now remove the older of the two `/var/tmp/root-*` directories:

   ```
   # rm -rf /var/tmp/root-19980214
   ```

6. Repeat this process every time you need to merge in changes to `/etc`.

You can use date(1) to automate the generation of the directory names:

```
# mkdir /var/tmp/root-‘date "+%Y%m%d"‘
```

23.4.12 Rebooting

You are now done. After you have verified that everything appears to be in the right place you can reboot the system. A simple shutdown(8) should do it:

```
# shutdown -r now
```

23.4.13 Finished

You should now have successfully upgraded your FreeBSD system. Congratulations.

If things went slightly wrong, it is easy to rebuild a particular piece of the system. For example, if you accidentally deleted `/etc/magic` as part of the upgrade or merge of `/etc`, the file(1) command will stop working. In this case, the fix would be to run:

```
# cd /usr/src/usr.bin/file
# make all install
```

23.4.14 Questions

1. Do I need to re-make the world for every change?

There is no easy answer to this one, as it depends on the nature of the change. For example, if you just ran **CVSup**, and it has shown the following files as being updated:

```
src/games/cribbage/instr.c
src/games/sail/pl_main.c
src/release/sysinstall/config.c
src/release/sysinstall/media.c
src/share/mk/bsd.port.mk
```

it probably is not worth rebuilding the entire world. You could just go to the appropriate sub-directories and `make all install`, and that's about it. But if something major changed, for example `src/lib/libc/stdlib` then you should either re-make the world, or at least those parts of it that are statically linked (as well as anything else you might have added that is statically linked).

At the end of the day, it is your call. You might be happy re-making the world every fortnight say, and let changes accumulate over that fortnight. Or you might want to re-make just those things that have changed, and be confident you can spot all the dependencies.

And, of course, this all depends on how often you want to upgrade, and whether you are tracking FreeBSD-STABLE or FreeBSD-CURRENT.

2. My compile failed with lots of signal 11 (or other signal number) errors. What has happened?

This is normally indicative of hardware problems. (Re)making the world is an effective way to stress test your hardware, and will frequently throw up memory problems. These normally manifest themselves as the compiler mysteriously dying on receipt of strange signals.

A sure indicator of this is if you can restart the make and it dies at a different point in the process.

In this instance there is little you can do except start swapping around the components in your machine to determine which one is failing.

3. Can I remove `/usr/obj` when I have finished?

The short answer is yes.

`/usr/obj` contains all the object files that were produced during the compilation phase. Normally, one of the first steps in the `make buildworld` process is to remove this directory and start afresh. In this case, keeping `/usr/obj` around after you have finished makes little sense, and will free up a large chunk of disk space (currently about 340 MB).

However, if you know what you are doing you can have `make buildworld` skip this step. This will make subsequent builds run much faster, since most of sources will not need to be recompiled. The flip side of this is that subtle dependency problems can creep in, causing your build to fail in odd ways. This frequently generates noise on the FreeBSD mailing lists, when one person complains that their build has failed, not realizing that it is because they have tried to cut corners.

4. Can interrupted builds be resumed?

This depends on how far through the process you got before you found a problem.

In general (and this is not a hard and fast rule) the `make buildworld` process builds new copies of essential tools (such as gcc(1), and make(1)) and the system libraries. These tools and libraries are then installed. The new tools and libraries are then used to rebuild themselves, and are installed again. The entire system (now including regular user programs, such as ls(1) or grep(1)) is then rebuilt with the new system files.

If you are at the last stage, and you know it (because you have looked through the output that you were storing) then you can (fairly safely) do:

```
... fix the problem ...
# cd /usr/src
# make -DNO_CLEAN all
```

This will not undo the work of the previous `make buildworld`.

If you see the message:

```
--------------------------------------------------------------
Building everything..
--------------------------------------------------------------
```

in the `make buildworld` output then it is probably fairly safe to do so.

If you do not see that message, or you are not sure, then it is always better to be safe than sorry, and restart the build from scratch.

5. How can I speed up making the world?

· Run in single user mode.

- Put the `/usr/src` and `/usr/obj` directories on separate file systems held on separate disks. If possible, put these disks on separate disk controllers.
- Better still, put these file systems across multiple disks using the ccd(4) (concatenated disk driver) device.
- Turn off profiling (set "NO_PROFILE=true" in `/etc/make.conf`). You almost certainly do not need it.
- Also in `/etc/make.conf`, set `CFLAGS` to something like `-O -pipe`. The optimization `-O2` is much slower, and the optimization difference between `-O` and `-O2` is normally negligible. `-pipe` lets the compiler use pipes rather than temporary files for communication, which saves disk access (at the expense of memory).
- Pass the `-j`*n* option to make(1) to run multiple processes in parallel. This usually helps regardless of whether you have a single or a multi processor machine.
- The file system holding `/usr/src` can be mounted (or remounted) with the `noatime` option. This prevents the file system from recording the file access time. You probably do not need this information anyway.

```
# mount -u -o noatime /usr/src
```

> **Warning:** The example assumes `/usr/src` is on its own file system. If it is not (if it is a part of `/usr` for example) then you will need to use that file system mount point, and not `/usr/src`.

- The file system holding `/usr/obj` can be mounted (or remounted) with the `async` option. This causes disk writes to happen asynchronously. In other words, the write completes immediately, and the data is written to the disk a few seconds later. This allows writes to be clustered together, and can be a dramatic performance boost.

> **Warning:** Keep in mind that this option makes your file system more fragile. With this option there is an increased chance that, should power fail, the file system will be in an unrecoverable state when the machine restarts.
>
> If `/usr/obj` is the only thing on this file system then it is not a problem. If you have other, valuable data on the same file system then ensure your backups are fresh before you enable this option.

```
# mount -u -o async /usr/obj
```

> **Warning:** As above, if `/usr/obj` is not on its own file system, replace it in the example with the name of the appropriate mount point.

6. What do I do if something goes wrong?

Make absolutely sure your environment has no extraneous cruft from earlier builds. This is simple enough.

```
# chflags -R noschg /usr/obj/usr
# rm -rf /usr/obj/usr
# cd /usr/src
# make cleandir
# make cleandir
```

Yes, `make cleandir` really should be run twice.

Then restart the whole process, starting with `make buildworld`.

If you still have problems, send the error and the output of `uname -a` to FreeBSD general questions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions). Be prepared to answer other questions about your setup!

23.5 Tracking for Multiple Machines

Contributed by Mike Meyer.

If you have multiple machines that you want to track the same source tree, then having all of them download sources and rebuild everything seems like a waste of resources: disk space, network bandwidth, and CPU cycles. It is, and the solution is to have one machine do most of the work, while the rest of the machines mount that work via NFS. This section outlines a method of doing so.

23.5.1 Preliminaries

First, identify a set of machines that is going to run the same set of binaries, which we will call a *build set*. Each machine can have a custom kernel, but they will be running the same userland binaries. From that set, choose a machine to be the *build machine*. It is going to be the machine that the world and kernel are built on. Ideally, it should be a fast machine that has sufficient spare CPU to run `make buildworld` and `make buildkernel`. You will also want to choose a machine to be the *test machine*, which will test software updates before they are put into production. This *must* be a machine that you can afford to have down for an extended period of time. It can be the build machine, but need not be.

All the machines in this build set need to mount `/usr/obj` and `/usr/src` from the same machine, and at the same point. Ideally, those are on two different drives on the build machine, but they can be NFS mounted on that machine as well. If you have multiple build sets, `/usr/src` should be on one build machine, and NFS mounted on the rest.

Finally make sure that `/etc/make.conf` on all the machines in the build set agrees with the build machine. That means that the build machine must build all the parts of the base system that any machine in the build set is going to install. Also, each build machine should have its kernel name set with `KERNCONF` in `/etc/make.conf`, and the build machine should list them all in `KERNCONF`, listing its own kernel first. The build machine must have the kernel configuration files for each machine in `/usr/src/sys/`*`arch`*`/conf` if it is going to build their kernels.

23.5.2 The Base System

Now that all that is done, you are ready to build everything. Build the kernel and world as described in Section 23.4.7.2 on the build machine, but do not install anything. After the build has finished, go to the test machine, and install the kernel you just built. If this machine mounts `/usr/src` and `/usr/obj` via NFS, when you reboot to single user you will need to enable the network and mount them. The easiest way to do this is to boot to multi-user, then run `shutdown now` to go to single user mode. Once there, you can install the new kernel and world and run `mergemaster` just as you normally would. When done, reboot to return to normal multi-user operations for this machine.

After you are certain that everything on the test machine is working properly, use the same procedure to install the new software on each of the other machines in the build set.

23.5.3 Ports

The same ideas can be used for the ports tree. The first critical step is mounting `/usr/ports` from the same machine to all the machines in the build set. You can then set up `/etc/make.conf` properly to share distfiles. You should set `DISTDIR` to a common shared directory that is writable by whichever user `root` is mapped to by your NFS mounts. Each machine should set `WRKDIRPREFIX` to a local build directory. Finally, if you are going to be building and distributing packages, you should set `PACKAGES` to a directory similar to `DISTDIR`.

Notes

1. 然而，這也不一定是正確，我們不可能永遠支援FreeBSD 昔日的各種發行版本，儘管每個發行版發佈之後，都仍會持續支援數年之久。若欲瞭解FreeBSD 目前對於舊版的支援政策細節，請參閱http://www.FreeBSD.org/security/ 。

IV. 網路通訊

FreeBSD 是一種廣泛的被使用在高效能的網路伺服器中的作業系統，這些章節包含了：

- 序列埠通訊
- PPP 和PPPoE
- 電子郵件
- 執行網路伺服程式
- 防火牆
- 其他的進階網路主題

這些章節是讓您在需要查資料的時候翻閱用的。您不需要依照特定的順序來讀，也不需要將這些章節全部讀過之後才將FreeBSD 用在網路環境下。

Chapter 24 Serial Communications

24.1 Synopsis

UNIX has always had support for serial communications. In fact, the very first UNIX machines relied on serial lines for user input and output. Things have changed a lot from the days when the average “terminal” consisted of a 10-character-per-second serial printer and a keyboard. This chapter will cover some of the ways in which FreeBSD uses serial communications.

After reading this chapter, you will know:

- How to connect terminals to your FreeBSD system.
- How to use a modem to dial out to remote hosts.
- How to allow remote users to login to your system with a modem.
- How to boot your system from a serial console.

Before reading this chapter, you should:

- Know how to configure and install a new kernel (Chapter 8).
- Understand UNIX permissions and processes (Chapter 3).
- Have access to the technical manual for the serial hardware (modem or multi-port card) that you would like to use with FreeBSD.

24.2 Introduction

24.2.1 Terminology

bps

Bits per Second ——the rate at which data is transmitted

DTE

Data Terminal Equipment ——for example, your computer

DCE

Data Communications Equipment ——your modem

RS-232

EIA standard for hardware serial communications

When talking about communications data rates, this section does not use the term “baud”. Baud refers to the number of electrical state transitions that may be made in a period of time, while “bps” (bits per second) is the *correct* term to use (at least it does not seem to bother the curmudgeons quite as much).

24.2.2 Cables and Ports

To connect a modem or terminal to your FreeBSD system, you will need a serial port on your computer and the proper cable to connect to your serial device. If you are already familiar with your hardware and the cable it requires, you can safely skip this section.

24.2.2.1 Cables

There are several different kinds of serial cables. The two most common types for our purposes are null-modem cables and standard (“straight”) RS-232 cables. The documentation for your hardware should describe the type of cable required.

24.2.2.1.1 Null-modem Cables

A null-modem cable passes some signals, such as “Signal Ground”, straight through, but switches other signals. For example, the “Transmitted Data” pin on one end goes to the “Received Data” pin on the other end.

You can also construct your own null-modem cable for use with terminals (e.g., for quality purposes). This table shows the RS-232C signals and the pin numbers on a DB-25 connector. Note that the standard also calls for a straight-through pin 1 to pin 1 *Protective Ground* line, but it is often omitted. Some terminals work OK using only pins 2, 3 and 7, while others require different configurations than the examples shown below.

Table 24-1. DB-25 to DB-25 Null-Modem Cable

Signal	Pin #		Pin #	Signal
SG	7	connects to	7	SG
TD	2	connects to	3	RD
RD	3	connects to	2	TD
RTS	4	connects to	5	CTS
CTS	5	connects to	4	RTS
DTR	20	connects to	6	DSR
DTR	20	connects to	8	DCD
DSR	6	connects to	20	DTR
DCD	8	connects to	20	DTR

Here are two other schemes more common nowadays.

Table 24-2. DB-9 to DB-9 Null-Modem Cable

Signal	Pin #		Pin #	Signal
RD	2	connects to	3	TD
TD	3	connects to	2	RD
DTR	4	connects to	6	DSR
DTR	4	connects to	1	DCD
SG	5	connects to	5	SG
DSR	6	connects to	4	DTR
DCD	1	connects to	4	DTR

Signal	Pin #		Pin #	Signal
RTS	7	connects to	8	CTS
CTS	8	connects to	7	RTS

Table 24-3. DB-9 to DB-25 Null-Modem Cable

Signal	Pin #		Pin #	Signal
RD	2	connects to	2	TD
TD	3	connects to	3	RD
DTR	4	connects to	6	DSR
DTR	4	connects to	8	DCD
SG	5	connects to	7	SG
DSR	6	connects to	20	DTR
DCD	1	connects to	20	DTR
RTS	7	connects to	5	CTS
CTS	8	connects to	4	RTS

Note: When one pin at one end connects to a pair of pins at the other end, it is usually implemented with one short wire between the pair of pins in their connector and a long wire to the other single pin.

The above designs seems to be the most popular. In another variation (explained in the book *RS-232 Made Easy*) SG connects to SG, TD connects to RD, RTS and CTS connect to DCD, DTR connects to DSR, and vice-versa.

24.2.2.1.2 Standard RS-232C Cables

A standard serial cable passes all of the RS-232C signals straight through. That is, the "Transmitted Data" pin on one end of the cable goes to the "Transmitted Data" pin on the other end. This is the type of cable to use to connect a modem to your FreeBSD system, and is also appropriate for some terminals.

24.2.2.2 Ports

Serial ports are the devices through which data is transferred between the FreeBSD host computer and the terminal. This section describes the kinds of ports that exist and how they are addressed in FreeBSD.

24.2.2.2.1 Kinds of Ports

Several kinds of serial ports exist. Before you purchase or construct a cable, you need to make sure it will fit the ports on your terminal and on the FreeBSD system.

Most terminals will have DB-25 ports. Personal computers, including PCs running FreeBSD, will have DB-25 or DB-9 ports. If you have a multiport serial card for your PC, you may have RJ-12 or RJ-45 ports.

See the documentation that accompanied the hardware for specifications on the kind of port in use. A visual inspection of the port often works too.

24.2.2.2.2 Port Names

In FreeBSD, you access each serial port through an entry in the `/dev` directory. There are two different kinds of entries:

- Call-in ports are named `/dev/ttydN` where *N* is the port number, starting from zero. Generally, you use the call-in port for terminals. Call-in ports require that the serial line assert the data carrier detect (DCD) signal to work correctly.
- Call-out ports are named `/dev/cuadN`. You usually do not use the call-out port for terminals, just for modems. You may use the call-out port if the serial cable or the terminal does not support the carrier detect signal.

> **Note:** Call-out ports are named `/dev/cuaaN` in FreeBSD 5.X and older.

If you have connected a terminal to the first serial port (`COM1` in MS-DOS), then you will use `/dev/ttyd0` to refer to the terminal. If the terminal is on the second serial port (also known as `COM2`), use `/dev/ttyd1`, and so forth.

24.2.3 Kernel Configuration

FreeBSD supports four serial ports by default. In the MS-DOS world, these are known as `COM1`, `COM2`, `COM3`, and `COM4`. FreeBSD currently supports "dumb" multiport serial interface cards, such as the BocaBoard 1008 and 2016, as well as more intelligent multi-port cards such as those made by Digiboard and Stallion Technologies. However, the default kernel only looks for the standard COM ports.

To see if your kernel recognizes any of your serial ports, watch for messages while the kernel is booting, or use the `/sbin/dmesg` command to replay the kernel's boot messages. In particular, look for messages that start with the characters `sio`.

> **Tip:** To view just the messages that have the word `sio`, use the command:
>
> ```
> # /sbin/dmesg | grep 'sio'
> ```

For example, on a system with four serial ports, these are the serial-port specific kernel boot messages:

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
sio2 at 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
sio3 at 0x2e8-0x2ef irq 9 on isa
sio3: type 16550A
```

If your kernel does not recognize all of your serial ports, you will probably need to configure your kernel in the `/boot/device.hints` file. You can also comment-out or completely remove lines for devices you do not have.

On FreeBSD 4.X you have to edit your kernel configuration file. For detailed information on configuring your kernel, please see Chapter 8. The relevant device lines would look like this:

```
device  sio0 at isa? port IO_COM1 irq 4
device  sio1 at isa? port IO_COM2 irq 3
device  sio2 at isa? port IO_COM3 irq 5
device  sio3 at isa? port IO_COM4 irq 9
```

Please refer to the sio(4) manual page for more information on serial ports and multiport boards configuration. Be careful if you are using a configuration file that was previously used for a different version of FreeBSD because the device flags and the syntax have changed between versions.

> **Note:** `port IO_COM1` is a substitution for `port 0x3f8`, `IO_COM2` is `0x2f8`, `IO_COM3` is `0x3e8`, and `IO_COM4` is `0x2e8`, which are fairly common port addresses for their respective serial ports; interrupts 4, 3, 5, and 9 are fairly common interrupt request lines. Also note that regular serial ports *cannot* share interrupts on ISA-bus PCs (multiport boards have on-board electronics that allow all the 16550A's on the board to share one or two interrupt request lines).

24.2.4 Device Special Files

Most devices in the kernel are accessed through "device special files", which are located in the `/dev` directory. The `sio` devices are accessed through the `/dev/ttydN` (dial-in) and `/dev/cuadN` (call-out) devices. FreeBSD also provides initialization devices (`/dev/ttydN.init` and `/dev/cuadN.init` on FreeBSD 6.X, `/dev/ttyidN` and `/dev/cuaidN` on FreeBSD 5.X and older) and locking devices (`/dev/ttydN.lock` and `/dev/cuadN.lock` on FreeBSD 6.X, `/dev/ttyldN` and `/dev/cualdN` on FreeBSD 5.X and older). The initialization devices are used to initialize communications port parameters each time a port is opened, such as `crtscts` for modems which use `RTS/CTS` signaling for flow control. The locking devices are used to lock flags on ports to prevent users or programs changing certain parameters; see the manual pages termios(4), sio(4), and stty(1) for information on the terminal settings, locking and initializing devices, and setting terminal options, respectively.

24.2.4.1 Making Device Special Files

> **Note:** FreeBSD 5.0 includes the devfs(5) filesystem which automatically creates device nodes as needed. If you are running a version of FreeBSD with `devfs` enabled then you can safely skip this section.

A shell script called `MAKEDEV` in the `/dev` directory manages the device special files. To use `MAKEDEV` to make dial-up device special files for `COM1` (port 0), `cd` to `/dev` and issue the command `MAKEDEV ttyd0`. Likewise, to make dial-up device special files for `COM2` (port 1), use `MAKEDEV ttyd1`.

`MAKEDEV` not only creates the `/dev/ttydN` device special files, but also the `/dev/cuaaN`, `/dev/cuaiaN`, `/dev/cualaN`, `/dev/ttyldN`, and `/dev/ttyidN` nodes.

After making new device special files, be sure to check the permissions on the files (especially the `/dev/cua*` files) to make sure that only users who should have access to those device special files can read and write on them ——you probably do not want to allow your average user to use your modems to dial-out. The default permissions on the `/dev/cua*` files should be sufficient:

```
crw-rw----    1 uucp     dialer    28, 129 Feb 15 14:38 /dev/cuaa1
crw-rw----    1 uucp     dialer    28, 161 Feb 15 14:38 /dev/cuaia1
crw-rw----    1 uucp     dialer    28, 193 Feb 15 14:38 /dev/cuala1
```

These permissions allow the user `uucp` and users in the group `dialer` to use the call-out devices.

24.2.5 Serial Port Configuration

The `ttydN` (or `cuadN`) device is the regular device you will want to open for your applications. When a process opens the device, it will have a default set of terminal I/O settings. You can see these settings with the command

```
# stty -a -f /dev/ttyd1
```

When you change the settings to this device, the settings are in effect until the device is closed. When it is reopened, it goes back to the default set. To make changes to the default set, you can open and adjust the settings of the "initial state" device. For example, to turn on `CLOCAL` mode, 8 bit communication, and `XON/XOFF` flow control by default for `ttyd5`, type:

```
# stty -f /dev/ttyd5.init clocal cs8 ixon ixoff
```

System-wide initialization of the serial devices is controlled in `/etc/rc.d/serial`. This file affects the default settings of serial devices.

> **Note:** On FreeBSD 4.X, system-wide initialization of the serial devices is controlled in `/etc/rc.serial`.

To prevent certain settings from being changed by an application, make adjustments to the "lock state" device. For example, to lock the speed of `ttyd5` to 57600 bps, type:

```
# stty -f /dev/ttyd5.lock 57600
```

Now, an application that opens `ttyd5` and tries to change the speed of the port will be stuck with 57600 bps.

Naturally, you should make the initial state and lock state devices writable only by the `root` account.

24.3 Terminals

Contributed by Sean Kelly.

Terminals provide a convenient and low-cost way to access your FreeBSD system when you are not at the computer's console or on a connected network. This section describes how to use terminals with FreeBSD.

24.3.1 Uses and Types of Terminals

The original UNIX systems did not have consoles. Instead, people logged in and ran programs through terminals that were connected to the computer's serial ports. It is quite similar to using a modem and terminal software to dial into a remote system to do text-only work.

Today's PCs have consoles capable of high quality graphics, but the ability to establish a login session on a serial port still exists in nearly every UNIX style operating system today; FreeBSD is no exception. By using a terminal attached to an unused serial port, you can log in and run any text program that you would normally run on the console or in an `xterm` window in the X Window System.

For the business user, you can attach many terminals to a FreeBSD system and place them on your employees' desktops. For a home user, a spare computer such as an older IBM PC or a Macintosh can be a terminal wired into a more powerful computer running FreeBSD. You can turn what might otherwise be a single-user computer into a powerful multiple user system.

For FreeBSD, there are three kinds of terminals:

- Dumb terminals
- PCs acting as terminals
- X terminals

The remaining subsections describe each kind.

24.3.1.1 Dumb Terminals

Dumb terminals are specialized pieces of hardware that let you connect to computers over serial lines. They are called "dumb" because they have only enough computational power to display, send, and receive text. You cannot run any programs on them. It is the computer to which you connect them that has all the power to run text editors, compilers, email, games, and so forth.

There are hundreds of kinds of dumb terminals made by many manufacturers, including Digital Equipment Corporation's VT-100 and Wyse's WY-75. Just about any kind will work with FreeBSD. Some high-end terminals can even display graphics, but only certain software packages can take advantage of these advanced features.

Dumb terminals are popular in work environments where workers do not need access to graphical applications such as those provided by the X Window System.

24.3.1.2 PCs Acting as Terminals

If a dumb terminal has just enough ability to display, send, and receive text, then certainly any spare personal computer can be a dumb terminal. All you need is the proper cable and some *terminal emulation* software to run on the computer.

Such a configuration is popular in homes. For example, if your spouse is busy working on your FreeBSD system's console, you can do some text-only work at the same time from a less powerful personal computer hooked up as a terminal to the FreeBSD system.

24.3.1.3 X Terminals

X terminals are the most sophisticated kind of terminal available. Instead of connecting to a serial port, they usually connect to a network like Ethernet. Instead of being relegated to text-only applications, they can display any X application.

We introduce X terminals just for the sake of completeness. However, this chapter does *not* cover setup, configuration, or use of X terminals.

24.3.2 Configuration

This section describes what you need to configure on your FreeBSD system to enable a login session on a terminal. It assumes you have already configured your kernel to support the serial port to which the terminal is connected——and that you have connected it.

Recall from Chapter 12 that the `init` process is responsible for all process control and initialization at system startup. One of the tasks performed by `init` is to read the `/etc/ttys` file and start a `getty` process on the available terminals. The `getty` process is responsible for reading a login name and starting the `login` program.

Thus, to configure terminals for your FreeBSD system the following steps should be taken as `root`:

1. Add a line to `/etc/ttys` for the entry in the `/dev` directory for the serial port if it is not already there.
2. Specify that `/usr/libexec/getty` be run on the port, and specify the appropriate *`getty`* type from the `/etc/gettytab` file.
3. Specify the default terminal type.
4. Set the port to "on."
5. Specify whether the port should be "secure."
6. Force `init` to reread the `/etc/ttys` file.

As an optional step, you may wish to create a custom *`getty`* type for use in step 2 by making an entry in `/etc/gettytab`. This chapter does not explain how to do so; you are encouraged to see the gettytab(5) and the getty(8) manual pages for more information.

24.3.2.1 Adding an Entry to `/etc/ttys`

The `/etc/ttys` file lists all of the ports on your FreeBSD system where you want to allow logins. For example, the first virtual console `ttyv0` has an entry in this file. You can log in on the console using this entry. This file also contains entries for the other virtual consoles, serial ports, and pseudo-ttys. For a hardwired terminal, just list the serial port's `/dev` entry without the `/dev` part (for example, `/dev/ttyv0` would be listed as `ttyv0`).

A default FreeBSD install includes an `/etc/ttys` file with support for the first four serial ports: `ttyd0` through `ttyd3`. If you are attaching a terminal to one of those ports, you do not need to add another entry.

Example 24-1. Adding Terminal Entries to `/etc/ttys`

Suppose we would like to connect two terminals to the system: a Wyse-50 and an old 286 IBM PC running **Procomm** terminal software emulating a VT-100 terminal. We connect the Wyse to the second serial port and the 286 to the sixth serial port (a port on a multiport serial card). The corresponding entries in the `/etc/ttys` file would look like this:

```
ttyd1❶  "/usr/libexec/getty std.38400"❷  wy50❸  on❹  insecure❺
ttyd5   "/usr/libexec/getty std.19200"  vt100  on  insecure
```

❶ The first field normally specifies the name of the terminal special file as it is found in `/dev`.

❷ The second field is the command to execute for this line, which is usually getty(8). `getty` initializes and opens the line, sets the speed, prompts for a user name and then executes the login(1) program.

The `getty` program accepts one (optional) parameter on its command line, the *`getty`* type. A *`getty`* type configures characteristics on the terminal line, like bps rate and parity. The `getty` program reads these characteristics from the file `/etc/gettytab`.

The file `/etc/gettytab` contains lots of entries for terminal lines both old and new. In almost all cases, the entries that start with the text `std` will work for hardwired terminals. These entries ignore parity. There is a `std` entry for each bps rate from 110 to 115200. Of course, you can add your own entries to this file. The gettytab(5) manual page provides more information.

When setting the *`getty`* type in the `/etc/ttys` file, make sure that the communications settings on the terminal match.

For our example, the Wyse-50 uses no parity and connects at 38400 bps. The 286 PC uses no parity and connects at 19200 bps.

❸ The third field is the type of terminal usually connected to that tty line. For dial-up ports, `unknown` or `dialup` is typically used in this field since users may dial up with practically any type of terminal or software. For hardwired terminals, the terminal type does not change, so you can put a real terminal type from the termcap(5) database file in this field.

For our example, the Wyse-50 uses the real terminal type while the 286 PC running **Procomm** will be set to emulate at VT-100.

❹ The fourth field specifies if the port should be enabled. Putting `on` here will have the `init` process start the program in the second field, `getty`. If you put `off` in this field, there will be no `getty`, and hence no logins on the port.

❺ The final field is used to specify whether the port is secure. Marking a port as secure means that you trust it enough to allow the `root` account (or any account with a user ID of 0) to login from that port. Insecure ports do not allow `root` logins. On an insecure port, users must login from unprivileged accounts and then use su(1) or a similar mechanism to gain superuser privileges.

It is highly recommended that you use "insecure" even for terminals that are behind locked doors. It is quite easy to login and use `su` if you need superuser privileges.

24.3.2.2 Force `init` to Reread `/etc/ttys`

After making the necessary changes to the `/etc/ttys` file you should send a SIGHUP (hangup) signal to the `init` process to force it to re-read its configuration file. For example:

```
# kill -HUP 1
```

Note: `init` is always the first process run on a system, therefore it will always have PID 1.

If everything is set up correctly, all cables are in place, and the terminals are powered up, then a `getty` process should be running on each terminal and you should see login prompts on your terminals at this point.

24.3.3 Troubleshooting Your Connection

Even with the most meticulous attention to detail, something could still go wrong while setting up a terminal. Here is a list of symptoms and some suggested fixes.

24.3.3.1 No Login Prompt Appears

Make sure the terminal is plugged in and powered up. If it is a personal computer acting as a terminal, make sure it is running terminal emulation software on the correct serial port.

Make sure the cable is connected firmly to both the terminal and the FreeBSD computer. Make sure it is the right kind of cable.

Make sure the terminal and FreeBSD agree on the bps rate and parity settings. If you have a video display terminal, make sure the contrast and brightness controls are turned up. If it is a printing terminal, make sure paper and ink are in good supply.

Make sure that a `getty` process is running and serving the terminal. For example, to get a list of running `getty` processes with `ps`, type:

```
# ps -axww|grep getty
```

You should see an entry for the terminal. For example, the following display shows that a `getty` is running on the second serial port `ttyd1` and is using the `std.38400` entry in `/etc/gettytab`:

```
22189  d1  Is+    0:00.03 /usr/libexec/getty std.38400 ttyd1
```

If no `getty` process is running, make sure you have enabled the port in `/etc/ttys`. Also remember to run `kill -HUP 1` after modifying the `ttys` file.

If the `getty` process is running but the terminal still does not display a login prompt, or if it displays a prompt but will not allow you to type, your terminal or cable may not support hardware handshaking. Try changing the entry in `/etc/ttys` from `std.38400` to `3wire.38400` remember to run `kill -HUP 1` after modifying `/etc/ttys`). The `3wire` entry is similar to `std`, but ignores hardware handshaking. You may need to reduce the baud rate or enable software flow control when using `3wire` to prevent buffer overflows.

24.3.3.2 If Garbage Appears Instead of a Login Prompt

Make sure the terminal and FreeBSD agree on the bps rate and parity settings. Check the `getty` processes to make sure the correct *`getty`* type is in use. If not, edit `/etc/ttys` and run `kill -HUP 1`.

24.3.3.3 Characters Appear Doubled; the Password Appears When Typed

Switch the terminal (or the terminal emulation software) from “half duplex” or “local echo” to “full duplex.”

24.4 Dial-in Service

Contributed by Guy Helmer. Additions by Sean Kelly.

Configuring your FreeBSD system for dial-in service is very similar to connecting terminals except that you are dealing with modems instead of terminals.

24.4.1 External vs. Internal Modems

External modems seem to be more convenient for dial-up, because external modems often can be semi-permanently configured via parameters stored in non-volatile RAM and they usually provide lighted indicators that display the state of important RS-232 signals. Blinking lights impress visitors, but lights are also very useful to see whether a modem is operating properly.

Internal modems usually lack non-volatile RAM, so their configuration may be limited only to setting DIP switches. If your internal modem has any signal indicator lights, it is probably difficult to view the lights when the system's cover is in place.

24.4.1.1 Modems and Cables

If you are using an external modem, then you will of course need the proper cable. A standard RS-232C serial cable should suffice as long as all of the normal signals are wired:

Table 24-4. Signal Names

Acronyms	Names
RD	Received Data
TD	Transmitted Data
DTR	Data Terminal Ready
DSR	Data Set Ready
DCD	Data Carrier Detect (RS-232's Received Line Signal Detector)
SG	Signal Ground
RTS	Request to Send
CTS	Clear to Send

FreeBSD needs the RTS and CTS signals for flow control at speeds above 2400 bps, the CD signal to detect when a call has been answered or the line has been hung up, and the DTR signal to reset the modem after a session is complete. Some cables are wired without all of the needed signals, so if you have problems, such as a login session not going away when the line hangs up, you may have a problem with your cable.

Like other UNIX like operating systems, FreeBSD uses the hardware signals to find out when a call has been answered or a line has been hung up and to hangup and reset the modem after a call. FreeBSD avoids sending commands to the modem or watching for status reports from the modem. If you are familiar with connecting modems to PC-based bulletin board systems, this may seem awkward.

24.4.2 Serial Interface Considerations

FreeBSD supports NS8250-, NS16450-, NS16550-, and NS16550A-based EIA RS-232C (CCITT V.24) communications interfaces. The 8250 and 16450 devices have single-character buffers. The 16550 device provides a 16-character buffer, which allows for better system performance. (Bugs in plain 16550's prevent the use of the 16-character buffer, so use 16550A's if possible). Because single-character-buffer devices require more work by the operating system than the 16-character-buffer devices, 16550A-based serial interface cards are much preferred. If the system has many active serial ports or will have a heavy load, 16550A-based cards are better for low-error-rate communications.

24.4.3 Quick Overview

As with terminals, `init` spawns a `getty` process for each configured serial port for dial-in connections. For example, if a modem is attached to `/dev/ttyd0`, the command `ps ax` might show this:

```
 4850 ??  I      0:00.09 /usr/libexec/getty V19200 ttyd0
```

When a user dials the modem's line and the modems connect, the CD (Carrier Detect) line is reported by the modem. The kernel notices that carrier has been detected and completes `getty`'s open of the port. `getty` sends a `login:` prompt at the specified initial line speed. `getty` watches to see if legitimate characters are received, and, in a typical configuration, if it finds junk (probably due to the modem's connection speed being different than `getty`'s speed), `getty` tries adjusting the line speeds until it receives reasonable characters.

After the user enters his/her login name, `getty` executes `/usr/bin/login`, which completes the login by asking for the user's password and then starting the user's shell.

24.4.4 Configuration Files

There are three system configuration files in the `/etc` directory that you will probably need to edit to allow dial-up access to your FreeBSD system. The first, `/etc/gettytab`, contains configuration information for the `/usr/libexec/getty` daemon. Second, `/etc/ttys` holds information that tells `/sbin/init` what `tty` devices should have `getty` processes running on them. Lastly, you can place port initialization commands in the `/etc/rc.d/serial` script.

There are two schools of thought regarding dial-up modems on UNIX. One group likes to configure their modems and systems so that no matter at what speed a remote user dials in, the local computer-to-modem RS-232 interface runs at a locked speed. The benefit of this configuration is that the remote user always sees a system login prompt immediately. The downside is that the system does not know what a user's true data rate is, so full-screen programs like Emacs will not adjust their screen-painting methods to make their response better for slower connections.

The other school configures their modems' RS-232 interface to vary its speed based on the remote user's connection speed. For example, V.32bis (14.4 Kbps) connections to the modem might make the modem run its RS-232 interface at 19.2 Kbps, while 2400 bps connections make the modem's RS-232 interface run at 2400 bps. Because `getty` does not understand any particular modem's connection speed reporting, `getty` gives a `login:` message at an initial speed and watches the characters that come back in response. If the user sees junk, it is assumed that they know they should press the Enter key until they see a recognizable prompt. If the data rates do not match, `getty` sees anything the user types as "junk", tries going to the next speed and gives the `login:` prompt again. This procedure can continue ad nauseam, but normally only takes a keystroke or two before the user sees a good prompt. Obviously, this login sequence does not look as clean as the former "locked-speed" method, but a user on a low-speed connection should receive better interactive response from full-screen programs.

This section will try to give balanced configuration information, but is biased towards having the modem's data rate follow the connection rate.

24.4.4.1 `/etc/gettytab`

`/etc/gettytab` is a termcap(5)-style file of configuration information for getty(8). Please see the gettytab(5) manual page for complete information on the format of the file and the list of capabilities.

24.4.4.1.1 Locked-speed Config

If you are locking your modem's data communications rate at a particular speed, you probably will not need to make any changes to `/etc/gettytab`.

24.4.4.1.2 Matching-speed Config

You will need to set up an entry in `/etc/gettytab` to give `getty` information about the speeds you wish to use for your modem. If you have a 2400 bps modem, you can probably use the existing `D2400` entry.

```
#
# Fast dialup terminals, 2400/1200/300 rotary (can start either way)
#
D2400|d2400|Fast-Dial-2400:\
        :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
        :nx=D300:tc=1200-baud:
5|D300|Fast-Dial-300:\
        :nx=D2400:tc=300-baud:
```

If you have a higher speed modem, you will probably need to add an entry in `/etc/gettytab`; here is an entry you could use for a 14.4 Kbps modem with a top interface speed of 19.2 Kbps:

```
#
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:\
        :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
        :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
        :nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
        :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
        :nx=V9600:tc=std.19200:
```

This will result in 8-bit, no parity connections.

The example above starts the communications rate at 19.2 Kbps (for a V.32bis connection), then cycles through 9600 bps (for V.32), 2400 bps, 1200 bps, 300 bps, and back to 19.2 Kbps. Communications rate cycling is implemented with the `nx=` ("next table") capability. Each of the lines uses a `tc=` ("table continuation") entry to pick up the rest of the "standard" settings for a particular data rate.

If you have a 28.8 Kbps modem and/or you want to take advantage of compression on a 14.4 Kbps modem, you need to use a higher communications rate than 19.2 Kbps. Here is an example of a `gettytab` entry starting a 57.6 Kbps:

```
#
# Additions for a V.32bis or V.34 Modem
# Starting at 57.6 Kbps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
        :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
        :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
        :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
        :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
        :nx=VH9600:tc=std.57600:
```

If you have a slow CPU or a heavily loaded system and do not have 16550A-based serial ports, you may receive "`sio`" "silo" errors at 57.6 Kbps.

24.4.4.2 `/etc/ttys`

Configuration of the `/etc/ttys` file was covered in Example 24-1. Configuration for modems is similar but we must pass a different argument to `getty` and specify a different terminal type. The general format for both locked-speed and matching-speed configurations is:

```
ttyd0   "/usr/libexec/getty xxx"   dialup on
```

The first item in the above line is the device special file for this entry ——`ttyd0` means `/dev/ttyd0` is the file that this `getty` will be watching. The second item, "`/usr/libexec/getty xxx`" (*xxx* will be replaced by the initial `gettytab` capability) is the process `init` will run on the device. The third item, `dialup`, is the default terminal type. The fourth parameter, `on`, indicates to `init` that the line is operational. There can be a fifth parameter, `secure`, but it should only be used for terminals which are physically secure (such as the system console).

The default terminal type (`dialup` in the example above) may depend on local preferences. `dialup` is the traditional default terminal type on dial-up lines so that users may customize their login scripts to notice when the terminal is `dialup` and automatically adjust their terminal type. However, the author finds it easier at his site to specify `vt102` as the default terminal type, since the users just use VT102 emulation on their remote systems.

After you have made changes to `/etc/ttys`, you may send the `init` process a HUP signal to re-read the file. You can use the command

```
# kill -HUP 1
```

to send the signal. If this is your first time setting up the system, you may want to wait until your modem(s) are properly configured and connected before signaling `init`.

24.4.4.2.1 Locked-speed Config

For a locked-speed configuration, your `ttys` entry needs to have a fixed-speed entry provided to `getty`. For a modem whose port speed is locked at 19.2 Kbps, the `ttys` entry might look like this:

```
ttyd0   "/usr/libexec/getty std.19200"   dialup on
```

If your modem is locked at a different data rate, substitute the appropriate value for `std.`*`speed`* instead of `std.19200`. Make sure that you use a valid type listed in `/etc/gettytab`.

24.4.4.2.2 Matching-speed Config

In a matching-speed configuration, your `ttys` entry needs to reference the appropriate beginning "auto-baud" (sic) entry in `/etc/gettytab`. For example, if you added the above suggested entry for a matching-speed modem that starts at 19.2 Kbps (the `gettytab` entry containing the `V19200` starting point), your `ttys` entry might look like this:

```
ttyd0   "/usr/libexec/getty V19200"   dialup on
```

24.4.4.3 `/etc/rc.d/serial`

High-speed modems, like V.32, V.32bis, and V.34 modems, need to use hardware (`RTS/CTS`) flow control. You can add `stty` commands to `/etc/rc.d/serial` to set the hardware flow control flag in the FreeBSD kernel for the modem ports.

For example to set the `termios` flag `crtscts` on serial port #1's (`COM2`) dial-in and dial-out initialization devices, the following lines could be added to `/etc/rc.d/serial`:

```
# Serial port initial configuration
stty -f /dev/ttyd1.init crtscts
stty -f /dev/cuad1.init crtscts
```

24.4.5 Modem Settings

If you have a modem whose parameters may be permanently set in non-volatile RAM, you will need to use a terminal program (such as Telix under MS-DOS or `tip` under FreeBSD) to set the parameters. Connect to the modem using the same communications speed as the initial speed `getty` will use and configure the modem's non-volatile RAM to match these requirements:

- CD asserted when connected
- DTR asserted for operation; dropping DTR hangs up line and resets modem
- CTS transmitted data flow control
- Disable XON/XOFF flow control
- RTS received data flow control
- Quiet mode (no result codes)

· No command echo

Please read the documentation for your modem to find out what commands and/or DIP switch settings you need to give it.

For example, to set the above parameters on a U.S. Robotics® Sportster® 14,400 external modem, one could give these commands to the modem:

```
ATZ
AT&C1&D2&H1&I0&R2&W
```

You might also want to take this opportunity to adjust other settings in the modem, such as whether it will use V.42bis and/or MNP5 compression.

The U.S. Robotics Sportster 14,400 external modem also has some DIP switches that need to be set; for other modems, perhaps you can use these settings as an example:

· Switch 1: UP ——DTR Normal

· Switch 2: N/A (Verbal Result Codes/Numeric Result Codes)

· Switch 3: UP ——Suppress Result Codes

· Switch 4: DOWN ——No echo, offline commands

· Switch 5: UP ——Auto Answer

· Switch 6: UP ——Carrier Detect Normal

· Switch 7: UP ——Load NVRAM Defaults

· Switch 8: N/A (Smart Mode/Dumb Mode)

Result codes should be disabled/suppressed for dial-up modems to avoid problems that can occur if `getty` mistakenly gives a `login:` prompt to a modem that is in command mode and the modem echoes the command or returns a result code. This sequence can result in a extended, silly conversation between `getty` and the modem.

24.4.5.1 Locked-speed Config

For a locked-speed configuration, you will need to configure the modem to maintain a constant modem-to-computer data rate independent of the communications rate. On a U.S. Robotics Sportster 14,400 external modem, these commands will lock the modem-to-computer data rate at the speed used to issue the commands:

```
ATZ
AT&B1&W
```

24.4.5.2 Matching-speed Config

For a variable-speed configuration, you will need to configure your modem to adjust its serial port data rate to match the incoming call rate. On a U.S. Robotics Sportster 14,400 external modem, these commands will lock the modem's error-corrected data rate to the speed used to issue the commands, but allow the serial port rate to vary for non-error-corrected connections:

```
ATZ
AT&B2&W
```

24.4.5.3 Checking the Modem's Configuration

Most high-speed modems provide commands to view the modem's current operating parameters in a somewhat human-readable fashion. On the U.S. Robotics Sportster 14,400 external modems, the command `ATI5` displays the settings that are stored in the non-volatile RAM. To see the true operating parameters of the modem (as influenced by the modem's DIP switch settings), use the commands `ATZ` and then `ATI4`.

If you have a different brand of modem, check your modem's manual to see how to double-check your modem's configuration parameters.

24.4.6 Troubleshooting

Here are a few steps you can follow to check out the dial-up modem on your system.

24.4.6.1 Checking Out the FreeBSD System

Hook up your modem to your FreeBSD system, boot the system, and, if your modem has status indication lights, watch to see whether the modem's DTR indicator lights when the `login:` prompt appears on the system's console ——if it lights up, that should mean that FreeBSD has started a `getty` process on the appropriate communications port and is waiting for the modem to accept a call.

If the DTR indicator does not light, login to the FreeBSD system through the console and issue a `ps ax` to see if FreeBSD is trying to run a `getty` process on the correct port. You should see lines like these among the processes displayed:

```
114 ??  I      0:00.10 /usr/libexec/getty V19200 ttyd0
115 ??  I      0:00.10 /usr/libexec/getty V19200 ttyd1
```

If you see something different, like this:

```
114 d0  I      0:00.10 /usr/libexec/getty V19200 ttyd0
```

and the modem has not accepted a call yet, this means that `getty` has completed its open on the communications port. This could indicate a problem with the cabling or a mis-configured modem, because `getty` should not be able to open the communications port until CD (carrier detect) has been asserted by the modem.

If you do not see any `getty` processes waiting to open the desired `ttydN` port, double-check your entries in `/etc/ttys` to see if there are any mistakes there. Also, check the log file `/var/log/messages` to see if there are any log messages from `init` or `getty` regarding any problems. If there are any messages, triple-check the configuration files `/etc/ttys` and `/etc/gettytab`, as well as the appropriate device special files `/dev/ttydN`, for any mistakes, missing entries, or missing device special files.

24.4.6.2 Try Dialing In

Try dialing into the system; be sure to use 8 bits, no parity, and 1 stop bit on the remote system. If you do not get a prompt right away, or get garbage, try pressing Enter about once per second. If you still do not see a `login:` prompt after a while, try sending a `BREAK`. If you are using a high-speed modem to do the dialing, try dialing again after locking the dialing modem's interface speed (via `AT&B1` on a U.S. Robotics Sportster modem, for example).

If you still cannot get a `login:` prompt, check `/etc/gettytab` again and double-check that

- The initial capability name specified in `/etc/ttys` for the line matches a name of a capability in `/etc/gettytab`
- Each `nx=` entry matches another `gettytab` capability name
- Each `tc=` entry matches another `gettytab` capability name

If you dial but the modem on the FreeBSD system will not answer, make sure that the modem is configured to answer the phone when DTR is asserted. If the modem seems to be configured correctly, verify that the DTR line is asserted by checking the modem's indicator lights (if it has any).

If you have gone over everything several times and it still does not work, take a break and come back to it later. If it still does not work, perhaps you can send an electronic mail message to the FreeBSD general questions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) describing your modem and your problem, and the good folks on the list will try to help.

24.5 Dial-out Service

The following are tips for getting your host to be able to connect over the modem to another computer. This is appropriate for establishing a terminal session with a remote host.

This is useful to log onto a BBS.

This kind of connection can be extremely helpful to get a file on the Internet if you have problems with PPP. If you need to FTP something and PPP is broken, use the terminal session to FTP it. Then use zmodem to transfer it to your machine.

24.5.1 My Stock Hayes Modem Is Not Supported, What Can I Do?

Actually, the manual page for `tip` is out of date. There is a generic Hayes dialer already built in. Just use `at=hayes` in your `/etc/remote` file.

The Hayes driver is not smart enough to recognize some of the advanced features of newer modems——messages like `BUSY`, `NO DIALTONE`, or `CONNECT 115200` will just confuse it. You should turn those messages off when you use `tip` (using `ATX0&W`).

Also, the dial timeout for `tip` is 60 seconds. Your modem should use something less, or else tip will think there is a communication problem. Try `ATS7=45&W`.

> **Note:** As shipped, `tip` does not yet support Hayes modems fully. The solution is to edit the file `tipconf.h` in the directory `/usr/src/usr.bin/tip/tip`. Obviously you need the source distribution to do this.
>
> Edit the line `#define HAYES 0` to `#define HAYES 1`. Then `make` and `make install`. Everything works nicely after that.

24.5.2 How Am I Expected to Enter These AT Commands?

Make what is called a "direct" entry in your `/etc/remote` file. For example, if your modem is hooked up to the first serial port, `/dev/cuad0`, then put in the following line:

```
cuad0:dv=/dev/cuad0:br#19200:pa=none
```

Use the highest bps rate your modem supports in the br capability. Then, type `tip cuad0` and you will be connected to your modem.

Or use `cu` as `root` with the following command:

```
# cu -lline -sspeed
```

`line` is the serial port (e.g.`/dev/cuad0`) and *`speed`* is the speed (e.g.`57600`). When you are done entering the AT commands hit **~.** to exit.

24.5.3 The @ Sign for the pn Capability Does Not Work!

The `@` sign in the phone number capability tells tip to look in `/etc/phones` for a phone number. But the `@` sign is also a special character in capability files like `/etc/remote`. Escape it with a backslash:

```
pn=\@
```

24.5.4 How Can I Dial a Phone Number on the Command Line?

Put what is called a "generic" entry in your `/etc/remote` file. For example:

```
tip115200|Dial any phone number at 115200 bps:\
        :dv=/dev/cuad0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
        :dv=/dev/cuad0:br#57600:at=hayes:pa=none:du:
```

Then you can do things like:

```
# tip -115200 5551234
```

If you prefer `cu` over `tip`, use a generic `cu` entry:

```
cu115200|Use cu to dial any number at 115200bps:\
        :dv=/dev/cuad1:br#57600:at=hayes:pa=none:du:
```

and type:

```
# cu 5551234 -s 115200
```

24.5.5 Do I Have to Type in the bps Rate Every Time I Do That?

Put in an entry for `tip1200` or `cu1200`, but go ahead and use whatever bps rate is appropriate with the br capability. `tip` thinks a good default is 1200 bps which is why it looks for a `tip1200` entry. You do not have to use 1200 bps, though.

24.5.6 I Access a Number of Hosts Through a Terminal Server

Rather than waiting until you are connected and typing `CONNECT <host>` each time, use tip's `cm` capability. For example, these entries in `/etc/remote`:

```
pain|pain.deep13.com|Forrester's machine:\
        :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank's machine:\
        :cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
        :dv=/dev/cuad2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

will let you type `tip pain` or `tip muffin` to connect to the hosts pain or muffin, and `tip deep13` to get to the terminal server.

24.5.7 Can Tip Try More Than One Line for Each Site?

This is often a problem where a university has several modem lines and several thousand students trying to use them.

Make an entry for your university in `/etc/remote` and use `@` for the `pn` capability:

```
big-university:\
        :pn=\@:tc=dialout
dialout:\
        :dv=/dev/cuad3:br#9600:at=courier:du:pa=none:
```

Then, list the phone numbers for the university in `/etc/phones`:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip` will try each one in the listed order, then give up. If you want to keep retrying, run `tip` in a while loop.

24.5.8 Why Do I Have to Hit Ctrl+P Twice to Send Ctrl+P Once?

Ctrl+**P** is the default "force" character, used to tell `tip` that the next character is literal data. You can set the force character to any other character with the `~s` escape, which means "set a variable."

Type `~sforce=`*`single-char`* followed by a newline. *`single-char`* is any single character. If you leave out *`single-char`*, then the force character is the nul character, which you can get by typing **Ctrl**+**2** or **Ctrl**+**Space**. A pretty good value for *`single-char`* is **Shift**+**Ctrl**+**6**, which is only used on some terminal servers.

You can have the force character be whatever you want by specifying the following in your `$HOME/.tiprc` file:

```
force=<single-char>
```

24.5.9 Suddenly Everything I Type Is in Upper Case??

You must have pressed **Ctrl**+**A**, `tip`'s "raise character," specially designed for people with broken caps-lock keys. Use `~s` as above and set the variable `raisechar` to something reasonable. In fact, you can set it to the same as the force character, if you never expect to use either of these features.

Here is a sample .tiprc file perfect for **Emacs** users who need to type **Ctrl**+**2** and **Ctrl**+**A** a lot:

```
force=^^
raisechar=^^
```

The ^^ is **Shift**+**Ctrl**+**6**.

24.5.10 How Can I Do File Transfers with `tip`?

If you are talking to another UNIX system, you can send and receive files with `~p` (put) and `~t` (take). These commands run `cat` and `echo` on the remote system to accept and send files. The syntax is:

~p local-file [remote-file]

~t remote-file [local-file]

There is no error checking, so you probably should use another protocol, like zmodem.

24.5.11 How Can I Run zmodem with `tip`?

To receive files, start the sending program on the remote end. Then, type `~C rz` to begin receiving them locally.

To send files, start the receiving program on the remote end. Then, type `~C sz` *`files`* to send them to the remote system.

24.6 Setting Up the Serial Console

Contributed by Kazutaka YOKOTA. Based on a document by Bill Paul.

24.6.1 Introduction

FreeBSD has the ability to boot on a system with only a dumb terminal on a serial port as a console. Such a configuration should be useful for two classes of people: system administrators who wish to install FreeBSD on machines that have no keyboard or monitor attached, and developers who want to debug the kernel or device drivers.

As described in Chapter 12, FreeBSD employs a three stage bootstrap. The first two stages are in the boot block code which is stored at the beginning of the FreeBSD slice on the boot disk. The boot block will then load and run the boot loader (`/boot/loader`) as the third stage code.

In order to set up the serial console you must configure the boot block code, the boot loader code and the kernel.

24.6.2 Serial Console Configuration, Terse Version

This section assumes that you are using the default setup and just want a fast overview of setting up the serial console.

1. Connect the serial cable to COM1 and the controlling terminal.
2. To see all boot messages on the serial console, issue the following command while logged in as the superuser:

   ```
   # echo 'console="comconsole"' >> /boot/loader.conf
   ```

3. Edit `/etc/ttys` and change `off` to `on` and `dialup` to `vt100` for the `ttyd0` entry. Otherwise a password will not be required to connect via the serial console, resulting in a potential security hole.
4. Reboot the system to see if the changes took effect.

If a different configuration is required, a more in depth configuration explanation exists in Section 24.6.3.

24.6.3 Serial Console Configuration

1. Prepare a serial cable.

   You will need either a null-modem cable or a standard serial cable and a null-modem adapter. See Section 24.2.2 for a discussion on serial cables.

2. Unplug your keyboard.

   Most PC systems probe for the keyboard during the Power-On Self-Test (POST) and will generate an error if the keyboard is not detected. Some machines complain loudly about the lack of a keyboard and will not continue to boot until it is plugged in.

   If your computer complains about the error, but boots anyway, then you do not have to do anything special. (Some machines with Phoenix BIOS installed merely say "`Keyboard failed`" and continue to boot normally.)

   If your computer refuses to boot without a keyboard attached then you will have to configure the BIOS so that it ignores this error (if it can). Consult your motherboard's manual for details on how to do this.

   > **Tip:** Set the keyboard to "Not installed" in the BIOS setup. You will still be able to use your keyboard. All this does is tell the BIOS not to probe for a keyboard at power-on. Your BIOS should not complain if the keyboard is absent. You can leave the keyboard plugged in even with this flag set to "Not installed" and the keyboard will still work.

   > **Note:** If your system has a PS/2® mouse, chances are very good that you may have to unplug your mouse as well as your keyboard. This is because PS/2 mice share some hardware with the keyboard and leaving the mouse plugged in can fool the keyboard probe into thinking the keyboard is still there. It is said that a Gateway 2000 Pentium 90 MHz system with an AMI BIOS that behaves this way. In general, this is not a problem since the mouse is not much good without the keyboard anyway.

3. Plug a dumb terminal into `COM1` (`sio0`).

If you do not have a dumb terminal, you can use an old PC/XT with a modem program, or the serial port on another UNIX box. If you do not have a `COM1` (`sio0`), get one. At this time, there is no way to select a port other than `COM1` for the boot blocks without recompiling the boot blocks. If you are already using `COM1` for another device, you will have to temporarily remove that device and install a new boot block and kernel once you get FreeBSD up and running. (It is assumed that `COM1` will be available on a file/compute/terminal server anyway; if you really need `COM1` for something else (and you cannot switch that something else to `COM2` (`sio1`)), then you probably should not even be bothering with all this in the first place.)

4. Make sure the configuration file of your kernel has appropriate flags set for `COM1` (`sio0`).

   Relevant flags are:

   `0x10`

   Enables console support for this unit. The other console flags are ignored unless this is set. Currently, at most one unit can have console support; the first one (in config file order) with this flag set is preferred. This option alone will not make the serial port the console. Set the following flag or use the `-h` option described below, together with this flag.

   `0x20`

   Forces this unit to be the console (unless there is another higher priority console), regardless of the `-h` option discussed below. The flag `0x20` must be used together with the `0x10` flag.

   `0x40`

   Reserves this unit (in conjunction with `0x10`) and makes the unit unavailable for normal access. You should not set this flag to the serial port unit which you want to use as the serial console. The only use of this flag is to designate the unit for kernel remote debugging. See The Developer's Handbook (http://www.FreeBSD.org/doc/zh_TW.Big5/books/developers-handbook/index.html) for more information on remote debugging.

   > **Note:** In FreeBSD 4.0 or later the semantics of the flag `0x40` are slightly different and there is another flag to specify a serial port for remote debugging.

   Example:

   ```
   device sio0 at isa? port IO_COM1 flags 0x10 irq 4
   ```

   See the sio(4) manual page for more details.

   If the flags were not set, you need to run UserConfig (on a different console) or recompile the kernel.

5. Create `boot.config` in the root directory of the `a` partition on the boot drive.

   This file will instruct the boot block code how you would like to boot the system. In order to activate the serial console, you need one or more of the following options——if you want multiple options, include them all on the same line:

`-h`

Toggles internal and serial consoles. You can use this to switch console devices. For instance, if you boot from the internal (video) console, you can use `-h` to direct the boot loader and the kernel to use the serial port as its console device. Alternatively, if you boot from the serial port, you can use the `-h` to tell the boot loader and the kernel to use the video display as the console instead.

`-D`

Toggles single and dual console configurations. In the single configuration the console will be either the internal console (video display) or the serial port, depending on the state of the `-h` option above. In the dual console configuration, both the video display and the serial port will become the console at the same time, regardless of the state of the `-h` option. However, note that the dual console configuration takes effect only during the boot block is running. Once the boot loader gets control, the console specified by the `-h` option becomes the only console.

`-P`

Makes the boot block probe the keyboard. If no keyboard is found, the `-D` and `-h` options are automatically set.

> **Note:** Due to space constraints in the current version of the boot blocks, the `-P` option is capable of detecting extended keyboards only. Keyboards with less than 101 keys (and without F11 and F12 keys) may not be detected. Keyboards on some laptop computers may not be properly found because of this limitation. If this is the case with your system, you have to abandon using the `-P` option. Unfortunately there is no workaround for this problem.

Use either the `-P` option to select the console automatically, or the `-h` option to activate the serial console.

You may include other options described in boot(8) as well.

The options, except for `-P`, will be passed to the boot loader (`/boot/loader`). The boot loader will determine which of the internal video or the serial port should become the console by examining the state of the `-h` option alone. This means that if you specify the `-D` option but not the `-h` option in `/boot.config`, you can use the serial port as the console only during the boot block; the boot loader will use the internal video display as the console.

6. Boot the machine.

   When you start your FreeBSD box, the boot blocks will echo the contents of `/boot.config` to the console. For example:

   ```
   /boot.config: -P
   Keyboard: no
   ```

   The second line appears only if you put `-P` in `/boot.config` and indicates presence/absence of the keyboard. These messages go to either serial or internal console, or both, depending on the option in `/boot.config`.

Options	Message goes to
none	internal console
`-h`	serial console

Options	Message goes to
`-D`	serial and internal consoles
`-Dh`	serial and internal consoles
`-P`, keyboard present	internal console
`-P`, keyboard absent	serial console

After the above messages, there will be a small pause before the boot blocks continue loading the boot loader and before any further messages printed to the console. Under normal circumstances, you do not need to interrupt the boot blocks, but you may want to do so in order to make sure things are set up correctly.

Hit any key, other than Enter, at the console to interrupt the boot process. The boot blocks will then prompt you for further action. You should now see something like:

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Verify the above message appears on either the serial or internal console or both, according to the options you put in `/boot.config`. If the message appears in the correct console, hit Enter to continue the boot process.

If you want the serial console but you do not see the prompt on the serial terminal, something is wrong with your settings. In the meantime, you enter `-h` and hit Enter/Return (if possible) to tell the boot block (and then the boot loader and the kernel) to choose the serial port for the console. Once the system is up, go back and check what went wrong.

After the boot loader is loaded and you are in the third stage of the boot process you can still switch between the internal console and the serial console by setting appropriate environment variables in the boot loader. See Section 24.6.6.

24.6.4 Summary

Here is the summary of various settings discussed in this section and the console eventually selected.

24.6.4.1 Case 1: You Set the Flags to 0x10 for `sio0`

```
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
```

Options in /boot.config	Console during boot blocks	Console during boot loader	Console in kernel
nothing	internal	internal	internal
`-h`	serial	serial	serial
`-D`	serial and internal	internal	internal
`-Dh`	serial and internal	serial	serial
`-P`, keyboard present	internal	internal	internal
`-P`, keyboard absent	serial and internal	serial	serial

24.6.4.2 Case 2: You Set the Flags to 0x30 for sio0

```
device sio0 at isa? port IO_COM1 flags 0x30 irq 4
```

Options in /boot.config	Console during boot blocks	Console during boot loader	Console in kernel
nothing	internal	internal	serial
`-h`	serial	serial	serial
`-D`	serial and internal	internal	serial
`-Dh`	serial and internal	serial	serial
`-P`, keyboard present	internal	internal	serial
`-P`, keyboard absent	serial and internal	serial	serial

24.6.5 Tips for the Serial Console

24.6.5.1 Setting a Faster Serial Port Speed

By default, the serial port settings are: 9600 baud, 8 bits, no parity, and 1 stop bit. If you wish to change the speed, you need to recompile at least the boot blocks. Add the following line to `/etc/make.conf` and compile new boot blocks:

```
BOOT_COMCONSOLE_SPEED=19200
```

See Section 24.6.5.2 for detailed instructions about building and installing new boot blocks.

If the serial console is configured in some other way than by booting with `-h`, or if the serial console used by the kernel is different from the one used by the boot blocks, then you must also add the following option to the kernel configuration file and compile a new kernel:

```
options CONSPEED=19200
```

24.6.5.2 Using Serial Port Other Than `sio0` for the Console

Using a port other than `sio0` as the console requires some recompiling. If you want to use another serial port for whatever reasons, recompile the boot blocks, the boot loader and the kernel as follows.

1. Get the kernel source. (See Chapter 23)
2. Edit `/etc/make.conf` and set `BOOT_COMCONSOLE_PORT` to the address of the port you want to use (0x3F8, 0x2F8, 0x3E8 or 0x2E8). Only `sio0` through `sio3` (`COM1` through `COM4`) can be used; multiport serial cards will not work. No interrupt setting is needed.
3. Create a custom kernel configuration file and add appropriate flags for the serial port you want to use. For example, if you want to make `sio1` (`COM2`) the console:

   ```
   device sio1 at isa? port IO_COM2 flags 0x10 irq 3
   ```

   or

```
device sio1 at isa? port IO_COM2 flags 0x30 irq 3
```

The console flags for the other serial ports should not be set.

4. Recompile and install the boot blocks and the boot loader:

```
# cd /sys/boot
# make clean
# make
# make install
```

5. Rebuild and install the kernel.
6. Write the boot blocks to the boot disk with disklabel(8) and boot from the new kernel.

24.6.5.3 Entering the DDB Debugger from the Serial Line

If you wish to drop into the kernel debugger from the serial console (useful for remote diagnostics, but also dangerous if you generate a spurious BREAK on the serial port!) then you should compile your kernel with the following options:

```
options BREAK_TO_DEBUGGER
options DDB
```

24.6.5.4 Getting a Login Prompt on the Serial Console

While this is not required, you may wish to get a *login* prompt over the serial line, now that you can see boot messages and can enter the kernel debugging session through the serial console. Here is how to do it.

Open the file `/etc/ttys` with an editor and locate the lines:

```
ttyd0 "/usr/libexec/getty std.9600" unknown off secure
ttyd1 "/usr/libexec/getty std.9600" unknown off secure
ttyd2 "/usr/libexec/getty std.9600" unknown off secure
ttyd3 "/usr/libexec/getty std.9600" unknown off secure
```

`ttyd0` through `ttyd3` corresponds to `COM1` through `COM4`. Change `off` to `on` for the desired port. If you have changed the speed of the serial port, you need to change `std.9600` to match the current setting, e.g. `std.19200`.

You may also want to change the terminal type from `unknown` to the actual type of your serial terminal.

After editing the file, you must `kill -HUP 1` to make this change take effect.

24.6.6 Changing Console from the Boot Loader

Previous sections described how to set up the serial console by tweaking the boot block. This section shows that you can specify the console by entering some commands and environment variables in the boot loader. As the boot loader is invoked at the third stage of the boot process, after the boot block, the settings in the boot loader will override the settings in the boot block.

24.6.6.1 Setting Up the Serial Console

You can easily specify the boot loader and the kernel to use the serial console by writing just one line in `/boot/loader.rc`:

```
set console="comconsole"
```

This will take effect regardless of the settings in the boot block discussed in the previous section.

You had better put the above line as the first line of `/boot/loader.rc` so as to see boot messages on the serial console as early as possible.

Likewise, you can specify the internal console as:

```
set console="vidconsole"
```

If you do not set the boot loader environment variable `console`, the boot loader, and subsequently the kernel, will use whichever console indicated by the `-h` option in the boot block.

In versions 3.2 or later, you may specify the console in `/boot/loader.conf.local` or `/boot/loader.conf`, rather than in `/boot/loader.rc`. In this method your `/boot/loader.rc` should look like:

```
include /boot/loader.4th
start
```

Then, create `/boot/loader.conf.local` and put the following line there.

```
console=comconsole
```

or

```
console=vidconsole
```

See loader.conf(5) for more information.

> **Note:** At the moment, the boot loader has no option equivalent to the `-P` option in the boot block, and there is no provision to automatically select the internal console and the serial console based on the presence of the keyboard.

24.6.6.2 Using a Serial Port Other Than `sio0` for the Console

You need to recompile the boot loader to use a serial port other than `sio0` for the serial console. Follow the procedure described in Section 24.6.5.2.

24.6.7 Caveats

The idea here is to allow people to set up dedicated servers that require no graphics hardware or attached keyboards. Unfortunately, while most systems will let you boot without a keyboard, there are quite a few that will not let you boot without a graphics adapter. Machines with AMI BIOSes can be configured to boot with no graphics adapter installed simply by changing the "graphics adapter" setting in the CMOS configuration to "Not installed."

However, many machines do not support this option and will refuse to boot if you have no display hardware in the system. With these machines, you will have to leave some kind of graphics card plugged in, (even if it is just a junky mono board) although you will not have to attach a monitor. You might also try installing an AMI BIOS.

Chapter 25 PPP and SLIP

Restructured, reorganized, and updated by Jim Mock.

25.1 Synopsis

FreeBSD has a number of ways to link one computer to another. To establish a network or Internet connection through a dial-up modem, or to allow others to do so through you, requires the use of PPP or SLIP. This chapter describes setting up these modem-based communication services in detail.

After reading this chapter, you will know:

- How to set up user PPP.
- How to set up kernel PPP.
- How to set up PPPoE (PPP over Ethernet).
- How to set up PPPoA (PPP over ATM).
- How to configure and set up a SLIP client and server.

Before reading this chapter, you should:

- Be familiar with basic network terminology.
- Understand the basics and purpose of a dialup connection and PPP and/or SLIP.

You may be wondering what the main difference is between user PPP and kernel PPP. The answer is simple: user PPP processes the inbound and outbound data in userland rather than in the kernel. This is expensive in terms of copying the data between the kernel and userland, but allows a far more feature-rich PPP implementation. User PPP uses the `tun` device to communicate with the outside world whereas kernel PPP uses the `ppp` device.

> **Note:** Throughout in this chapter, user PPP will simply be referred to as **ppp** unless a distinction needs to be made between it and any other PPP software such as **pppd**. Unless otherwise stated, all of the commands explained in this chapter should be executed as `root`.

25.2 Using User PPP

Updated and enhanced by Tom Rhodes. Originally contributed by Brian Somers. With input from Nik Clayton, Dirk Frömberg, and Peter Childs.

25.2.1 User PPP

25.2.1.1 Assumptions

This document assumes you have the following:

- An account with an Internet Service Provider (ISP) which you connect to using PPP.
- You have a modem or other device connected to your system and configured correctly which allows you to connect to your ISP.
- The dial-up number(s) of your ISP.
- Your login name and password. (Either a regular UNIX style login and password pair, or a PAP or CHAP login and password pair.)
- The IP address of one or more name servers. Normally, you will be given two IP addresses by your ISP to use for this. If they have not given you at least one, then you can use the `enable dns` command in `ppp.conf` and **ppp** will set the name servers for you. This feature depends on your ISPs PPP implementation supporting DNS negotiation.

The following information may be supplied by your ISP, but is not completely necessary:

- The IP address of your ISP's gateway. The gateway is the machine to which you will connect and will be set up as your *default route*. If you do not have this information, we can make one up and your ISP's PPP server will tell us the correct value when we connect.

  This IP number is referred to as `HISADDR` by **ppp**.
- The netmask you should use. If your ISP has not provided you with one, you can safely use `255.255.255.255`.
- If your ISP provides you with a static IP address and hostname, you can enter it. Otherwise, we simply let the peer assign whatever IP address it sees fit.

If you do not have any of the required information, contact your ISP.

> **Note:** Throughout this section, many of the examples showing the contents of configuration files are numbered by line. These numbers serve to aid in the presentation and discussion only and are not meant to be placed in the actual file. Proper indentation with tab and space characters is also important.

25.2.1.2 Creating PPP Device Nodes

Under normal circumstances, most users will only need one `tun` device (`/dev/tun0`). References to `tun0` below may be changed to `tun`*`N`* where *`N`* is any unit number corresponding to your system.

For FreeBSD installations that do not have devfs(5) enabled (FreeBSD 4.X and earlier), the existence of the `tun0` device should be verified (this is not necessary if devfs(5) is enabled as device nodes will be created on demand).

The easiest way to make sure that the `tun0` device is configured correctly is to remake the device. To remake the device, do the following:

```
# cd /dev
# sh MAKEDEV tun0
```

If you need 16 tunnel devices in your kernel, you will need to create them. This can be done by executing the following commands:

```
# cd /dev
# sh MAKEDEV tun15
```

25.2.1.3 Automatic PPP Configuration

Both ppp and pppd (the kernel level implementation of PPP) use the configuration files located in the /etc/ppp directory. Examples for user ppp can be found in /usr/share/examples/ppp/.

Configuring ppp requires that you edit a number of files, depending on your requirements. What you put in them depends to some extent on whether your ISP allocates IP addresses statically (i.e., you get given one IP address, and always use that one) or dynamically (i.e., your IP address changes each time you connect to your ISP).

25.2.1.3.1 PPP and Static IP Addresses

You will need to edit the /etc/ppp/ppp.conf configuration file. It should look similar to the example below.

> **Note:** Lines that end in a : start in the first column (beginning of the line)——all other lines should be indented as shown using spaces or tabs.

```
1     default:
2       set log Phase Chat LCP IPCP CCP tun command
3       ident user-ppp VERSION (built COMPILATIONDATE)
4       set device /dev/cuaa0
5       set speed 115200
6       set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7                 \"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
8       set timeout 180
9       enable dns
10
11    provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set login "TIMEOUT 10 \"\" \"\" gin:--gin: \\U word: \\P col: ppp"
16      set timeout 300
17      set ifaddr x.x.x.x y.y.y.y 255.255.255.255 0.0.0.0
18      add default HISADDR
```

Line 1:

Identifies the default entry. Commands in this entry are executed automatically when ppp is run.

Line 2:

Enables logging parameters. When the configuration is working satisfactorily, this line should be reduced to saying

```
set log phase tun
```

in order to avoid excessive log file sizes.

Line 3:

Tells PPP how to identify itself to the peer. PPP identifies itself to the peer if it has any trouble negotiating and setting up the link, providing information that the peers administrator may find useful when investigating such problems.

Line 4:

Identifies the device to which the modem is connected. `COM1` is `/dev/cuaa0` and `COM2` is `/dev/cuaa1`.

Line 5:

Sets the speed you want to connect at. If 115200 does not work (it should with any reasonably new modem), try 38400 instead.

Line 6 & 7:

The dial string. User PPP uses an expect-send syntax similar to the chat(8) program. Refer to the manual page for information on the features of this language.

Note that this command continues onto the next line for readability. Any command in `ppp.conf` may do this if the last character on the line is a "\" character.

Line 8:

Sets the idle timeout for the link. 180 seconds is the default, so this line is purely cosmetic.

Line 9:

Tells PPP to ask the peer to confirm the local resolver settings. If you run a local name server, this line should be commented out or removed.

Line 10:

A blank line for readability. Blank lines are ignored by PPP.

Line 11:

Identifies an entry for a provider called "provider". This could be changed to the name of your ISP so that later you can use the `load ISP` to start the connection.

Line 12:

Sets the phone number for this provider. Multiple phone numbers may be specified using the colon (`:`) or pipe character (`|`)as a separator. The difference between the two separators is described in ppp(8). To summarize, if you want to rotate through the numbers, use a colon. If you want to always attempt to dial the first number first and only use the other numbers if the first number fails, use the pipe character. Always quote the entire set of phone numbers as shown.

You must enclose the phone number in quotation marks (`"`) if there is any intention on using spaces in the phone number. This can cause a simple, yet subtle error.

Line 13 & 14:

Identifies the user name and password. When connecting using a UNIX style login prompt, these values are referred to by the `set login` command using the \U and \P variables. When connecting using PAP or CHAP, these values are used at authentication time.

Line 15:

If you are using PAP or CHAP, there will be no login at this point, and this line should be commented out or removed. See PAP and CHAP authentication for further details.

The login string is of the same chat-like syntax as the dial string. In this example, the string works for a service whose login session looks like this:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

You will need to alter this script to suit your own needs. When you write this script for the first time, you should ensure that you have enabled "chat" logging so you can determine if the conversation is going as expected.

Line 16:

Sets the default idle timeout (in seconds) for the connection. Here, the connection will be closed automatically after 300 seconds of inactivity. If you never want to timeout, set this value to zero or use the `-ddial` command line switch.

Line 17:

Sets the interface addresses. The string *`x.x.x.x`* should be replaced by the IP address that your provider has allocated to you. The string *`y.y.y.y`* should be replaced by the IP address that your ISP indicated for their gateway (the machine to which you connect). If your ISP has not given you a gateway address, use `10.0.0.2/0`. If you need to use a "guessed" address, make sure that you create an entry in `/etc/ppp/ppp.linkup` as per the instructions for PPP and Dynamic IP addresses. If this line is omitted, `ppp` cannot run in `-auto` mode.

Line 18:

Adds a default route to your ISP's gateway. The special word `HISADDR` is replaced with the gateway address specified on line 17. It is important that this line appears after line 17, otherwise `HISADDR` will not yet be initialized.

If you do not wish to run ppp in `-auto`, this line should be moved to the `ppp.linkup` file.

It is not necessary to add an entry to `ppp.linkup` when you have a static IP address and are running ppp in `-auto` mode as your routing table entries are already correct before you connect. You may however wish to create an entry to invoke programs after connection. This is explained later with the sendmail example.

Example configuration files can be found in the `/usr/share/examples/ppp/` directory.

25.2.1.3.2 PPP and Dynamic IP Addresses

If your service provider does not assign static IP addresses, `ppp` can be configured to negotiate the local and remote addresses. This is done by "guessing" an IP address and allowing `ppp` to set it up correctly using the IP Configuration Protocol (IPCP) after connecting. The `ppp.conf` configuration is the same as PPP and Static IP Addresses, with the following change:

```
17      set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255
```

Again, do not include the line number, it is just for reference. Indentation of at least one space is required.

Line 17:

The number after the / character is the number of bits of the address that ppp will insist on. You may wish to use IP numbers more appropriate to your circumstances, but the above example will always work.

The last argument (0.0.0.0) tells PPP to start negotiations using address 0.0.0.0 rather than 10.0.0.1 and is necessary for some ISPs. Do not use 0.0.0.0 as the first argument to set ifaddr as it prevents PPP from setting up an initial route in -auto mode.

If you are not running in -auto mode, you will need to create an entry in /etc/ppp/ppp.linkup. ppp.linkup is used after a connection has been established. At this point, ppp will have assigned the interface addresses and it will now be possible to add the routing table entries:

```
1     provider:
2      add default HISADDR
```

Line 1:

On establishing a connection, ppp will look for an entry in ppp.linkup according to the following rules: First, try to match the same label as we used in ppp.conf. If that fails, look for an entry for the IP address of our gateway. This entry is a four-octet IP style label. If we still have not found an entry, look for the MYADDR entry.

Line 2:

This line tells ppp to add a default route that points to HISADDR. HISADDR will be replaced with the IP number of the gateway as negotiated by the IPCP.

See the pmdemand entry in the files /usr/share/examples/ppp/ppp.conf.sample and /usr/share/examples/ppp/ppp.linkup.sample for a detailed example.

25.2.1.3.3 Receiving Incoming Calls

When you configure **ppp** to receive incoming calls on a machine connected to a LAN, you must decide if you wish to forward packets to the LAN. If you do, you should allocate the peer an IP number from your LAN's subnet, and use the command enable proxy in your /etc/ppp/ppp.conf file. You should also confirm that the /etc/rc.conf file contains the following:

```
gateway_enable="YES"
```

25.2.1.3.4 Which getty?

Configuring FreeBSD for Dial-up Services provides a good description on enabling dial-up services using getty(8).

An alternative to getty is mgetty (http://www.leo.org/~doering/mgetty/index.html), a smarter version of getty designed with dial-up lines in mind.

The advantages of using mgetty is that it actively *talks* to modems, meaning if port is turned off in /etc/ttys then your modem will not answer the phone.

Later versions of mgetty (from 0.99beta onwards) also support the automatic detection of PPP streams, allowing your clients script-less access to your server.

Refer to Mgetty and AutoPPP for more information on mgetty.

25.2.1.3.5 PPP Permissions

The ppp command must normally be run as the root user. If however, you wish to allow ppp to run in server mode as a normal user by executing ppp as described below, that user must be given permission to run ppp by adding them to the network group in /etc/group.

You will also need to give them access to one or more sections of the configuration file using the allow command:

```
allow users fred mary
```

If this command is used in the default section, it gives the specified users access to everything.

25.2.1.3.6 PPP Shells for Dynamic-IP Users

Create a file called /etc/ppp/ppp-shell containing the following:

```
#!/bin/sh
IDENT=‘echo $0 | sed -e ’s/^.*-\(.*\)$/\1/’‘
CALLEDAS="$IDENT"
TTY=‘tty‘

if [ x$IDENT = xdialup ]; then
        IDENT=‘basename $TTY‘
fi

echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"

exec /usr/sbin/ppp -direct $IDENT
```

This script should be executable. Now make a symbolic link called ppp-dialup to this script using the following commands:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

You should use this script as the *shell* for all of your dialup users. This is an example from /etc/passwd for a dialup PPP user with username pchilds (remember do not directly edit the password file, use vipw(8)).

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

Create a /home/ppp directory that is world readable containing the following 0 byte files:

```
-r--r--r--   1 root     wheel           0 May 27 02:23 .hushlogin
-r--r--r--   1 root     wheel           0 May 27 02:22 .rhosts
```

which prevents /etc/motd from being displayed.

25.2.1.3.7 PPP Shells for Static-IP Users

Create the ppp-shell file as above, and for each account with statically assigned IPs create a symbolic link to ppp-shell.

For example, if you have three dialup customers, `fred`, `sam`, and `mary`, that you route class C networks for, you would type the following:

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary
```

Each of these users dialup accounts should have their shell set to the symbolic link created above (for example, `mary`'s shell should be `/etc/ppp/ppp-mary`).

25.2.1.3.8 Setting Up `ppp.conf` for Dynamic-IP Users

The `/etc/ppp/ppp.conf` file should contain something along the lines of:

```
default:
  set debug phase lcp chat
  set timeout 0

ttyd0:
  set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
  enable proxy

ttyd1:
  set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
  enable proxy
```

> **Note:** The indenting is important.

The `default:` section is loaded for each session. For each dialup line enabled in `/etc/ttys` create an entry similar to the one for `ttyd0:` above. Each line should get a unique IP address from your pool of IP addresses for dynamic users.

25.2.1.3.9 Setting Up `ppp.conf` for Static-IP Users

Along with the contents of the sample `/usr/share/examples/ppp/ppp.conf` above you should add a section for each of the statically assigned dialup users. We will continue with our `fred`, `sam`, and `mary` example.

```
fred:
  set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
  set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
  set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

The file `/etc/ppp/ppp.linkup` should also contain routing information for each static IP user if required. The line below would add a route for the `203.14.101.0` class C via the client's ppp link.

```
fred:
  add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
  add 203.14.102.0 netmask 255.255.255.0 HISADDR

mary:
  add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

25.2.1.3.10 `mgetty` and AutoPPP

Configuring and compiling `mgetty` with the `AUTO_PPP` option enabled allows `mgetty` to detect the LCP phase of PPP connections and automatically spawn off a ppp shell. However, since the default login/password sequence does not occur it is necessary to authenticate users using either PAP or CHAP.

This section assumes the user has successfully configured, compiled, and installed a version of `mgetty` with the `AUTO_PPP` option (v0.99beta or later).

Make sure your `/usr/local/etc/mgetty+sendfax/login.config` file has the following in it:

```
/AutoPPP/ -     -        /etc/ppp/ppp-pap-dialup
```

This will tell `mgetty` to run the `ppp-pap-dialup` script for detected PPP connections.

Create a file called `/etc/ppp/ppp-pap-dialup` containing the following (the file should be executable):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

For each dialup line enabled in `/etc/ttys`, create a corresponding entry in `/etc/ppp/ppp.conf`. This will happily co-exist with the definitions we created above.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Each user logging in with this method will need to have a username/password in `/etc/ppp/ppp.secret` file, or alternatively add the following option to authenticate users via PAP from the `/etc/passwd` file.

```
enable passwdauth
```

If you wish to assign some users a static IP number, you can specify the number as the third argument in `/etc/ppp/ppp.secret`. See `/usr/share/examples/ppp/ppp.secret.sample` for examples.

25.2.1.3.11 MS Extensions

It is possible to configure PPP to supply DNS and NetBIOS nameserver addresses on demand.

To enable these extensions with PPP version 1.x, the following lines might be added to the relevant section of `/etc/ppp/ppp.conf`.

```
enable msext
```

```
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

And for PPP version 2 and above:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

This will tell the clients the primary and secondary name server addresses, and a NetBIOS nameserver host.

In version 2 and above, if the `set dns` line is omitted, PPP will use the values found in `/etc/resolv.conf`.

25.2.1.3.12 PAP and CHAP Authentication

Some ISPs set their system up so that the authentication part of your connection is done using either of the PAP or CHAP authentication mechanisms. If this is the case, your ISP will not give a `login:` prompt when you connect, but will start talking PPP immediately.

PAP is less secure than CHAP, but security is not normally an issue here as passwords, although being sent as plain text with PAP, are being transmitted down a serial line only. There is not much room for crackers to "eavesdrop".

Referring back to the PPP and Static IP addresses or PPP and Dynamic IP addresses sections, the following alterations must be made:

```
13      set authname MyUserName
14      set authkey MyPassword
15      set login
```

Line 13:

This line specifies your PAP/CHAP user name. You will need to insert the correct value for *`MyUserName`*.

Line 14:

This line specifies your PAP/CHAP password. You will need to insert the correct value for *`MyPassword`*. You may want to add an additional line, such as:

```
16      accept PAP
```

or

```
16      accept CHAP
```

to make it obvious that this is the intention, but PAP and CHAP are both accepted by default.

Line 15:

Your ISP will not normally require that you log into the server if you are using PAP or CHAP. You must therefore disable your "set login" string.

25.2.1.3.13 Changing Your ppp Configuration on the Fly

It is possible to talk to the `ppp` program while it is running in the background, but only if a suitable diagnostic port has been set up. To do this, add the following line to your configuration:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

This will tell PPP to listen to the specified UNIX domain socket, asking clients for the specified password before allowing access. The `%d` in the name is replaced with the `tun` device number that is in use.

Once a socket has been set up, the pppctl(8) program may be used in scripts that wish to manipulate the running program.

25.2.1.4 Using PPP Network Address Translation Capability

PPP has ability to use internal NAT without kernel diverting capabilities. This functionality may be enabled by the following line in `/etc/ppp/ppp.conf`:

```
nat enable yes
```

Alternatively, PPP NAT may be enabled by command-line option `-nat`. There is also `/etc/rc.conf` knob named `ppp_nat`, which is enabled by default.

If you use this feature, you may also find useful the following `/etc/ppp/ppp.conf` options to enable incoming connections forwarding:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

or do not trust the outside at all

```
nat deny_incoming yes
```

25.2.1.5 Final System Configuration

You now have `ppp` configured, but there are a few more things to do before it is ready to work. They all involve editing the `/etc/rc.conf` file.

Working from the top down in this file, make sure the `hostname=` line is set, e.g.:

```
hostname="foo.example.com"
```

If your ISP has supplied you with a static IP address and name, it is probably best that you use this name as your host name.

Look for the `network_interfaces` variable. If you want to configure your system to dial your ISP on demand, make sure the `tun0` device is added to the list, otherwise remove it.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```

> **Note:** The `ifconfig_tun0` variable should be empty, and a file called `/etc/start_if.tun0` should be created. This file should contain the line:
>
> ```
> ppp -auto mysystem
> ```

This script is executed at network configuration time, starting your ppp daemon in automatic mode. If you have a LAN for which this machine is a gateway, you may also wish to use the `-alias` switch. Refer to the manual page for further details.

Make sure that the router program is set to `NO` with the following line in your `/etc/rc.conf`:

```
router_enable="NO"
```

It is important that the `routed` daemon is not started, as `routed` tends to delete the default routing table entries created by `ppp`.

It is probably worth your while ensuring that the `sendmail_flags` line does not include the `-q` option, otherwise `sendmail` will attempt to do a network lookup every now and then, possibly causing your machine to dial out. You may try:

```
sendmail_flags="-bd"
```

The downside of this is that you must force `sendmail` to re-examine the mail queue whenever the ppp link is up by typing:

```
# /usr/sbin/sendmail -q
```

You may wish to use the `!bg` command in `ppp.linkup` to do this automatically:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

If you do not like this, it is possible to set up a "dfilter" to block SMTP traffic. Refer to the sample files for further details.

All that is left is to reboot the machine. After rebooting, you can now either type:

```
# ppp
```

and then `dial provider` to start the PPP session, or, if you want `ppp` to establish sessions automatically when there is outbound traffic (and you have not created the `start_if.tun0` script), type:

```
# ppp -auto provider
```

25.2.1.6 Summary

To recap, the following steps are necessary when setting up ppp for the first time:

Client side:

1. Ensure that the `tun` device is built into your kernel.
2. Ensure that the `tun`*`N`* device file is available in the `/dev` directory.
3. Create an entry in `/etc/ppp/ppp.conf`. The `pmdemand` example should suffice for most ISPs.

4. If you have a dynamic IP address, create an entry in `/etc/ppp/ppp.linkup`.
5. Update your `/etc/rc.conf` file.
6. Create a `start_if.tun0` script if you require demand dialing.

Server side:

1. Ensure that the `tun` device is built into your kernel.
2. Ensure that the `tunN` device file is available in the `/dev` directory.
3. Create an entry in `/etc/passwd` (using the vipw(8) program).
4. Create a profile in this users home directory that runs `ppp -direct direct-server` or similar.
5. Create an entry in `/etc/ppp/ppp.conf`. The `direct-server` example should suffice.
6. Create an entry in `/etc/ppp/ppp.linkup`.
7. Update your `/etc/rc.conf` file.

25.3 Using Kernel PPP

Parts originally contributed by Gennady B. Sorokopud and Robert Huff.

25.3.1 Setting Up Kernel PPP

Before you start setting up PPP on your machine, make sure that `pppd` is located in `/usr/sbin` and the directory `/etc/ppp` exists.

`pppd` can work in two modes:

1. As a "client" ——you want to connect your machine to the outside world via a PPP serial connection or modem line.
2. As a "server" ——your machine is located on the network, and is used to connect other computers using PPP.

In both cases you will need to set up an options file (`/etc/ppp/options` or `~/.ppprc` if you have more than one user on your machine that uses PPP).

You will also need some modem/serial software (preferably `comms/kermit`), so you can dial and establish a connection with the remote host.

25.3.2 Using `pppd` as a Client

Based on information provided by Trev Roydhouse.

The following `/etc/ppp/options` might be used to connect to a Cisco terminal server PPP line.

```
crtscts         # enable hardware flow control
modem           # modem control line
```

```
noipdefault     # remote PPP server must supply your IP address
                # if the remote host does not send your IP during IPCP
                # negotiation, remove this option
passive         # wait for LCP packets
domain ppp.foo.com      # put your domain name here

:<remote_ip>    # put the IP of remote PPP host here
                # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to <local_ip>:<remote_ip>

defaultroute    # put this if you want that PPP server will be your
                # default router
```

To connect:

1. Dial to the remote host using **Kermit** (or some other modem program), and enter your user name and password (or whatever is needed to enable PPP on the remote host).
2. Exit **Kermit** (without hanging up the line).
3. Enter the following:

   ```
   # /usr/src/usr.sbin/pppd.new/pppd /dev/tty01 19200
   ```

   Be sure to use the appropriate speed and device name.

Now your computer is connected with PPP. If the connection fails, you can add the `debug` option to the `/etc/ppp/options` file, and check console messages to track the problem.

Following `/etc/ppp/pppup` script will make all 3 stages automatic:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=‘ps ax |grep pppd |grep -v grep|awk ’{print $1;}’‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing pppd, PID=’ ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=‘ps ax |grep kermit |grep -v grep|awk ’{print $1;}’‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing kermit, PID=’ ${pid}
        kill -9 ${pid}
fi

ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

`/etc/ppp/kermit.dial` is a **Kermit** script that dials and makes all necessary authorization on the remote host (an example of such a script is attached to the end of this document).

Use the following `/etc/ppp/pppdown` script to disconnect the PPP line:

```
#!/bin/sh
pid=‘ps ax |grep pppd |grep -v grep|awk ’{print $1;}’‘
if [ X${pid} != "X" ] ; then
        echo ’killing pppd, PID=’ ${pid}
        kill -TERM ${pid}
fi

ps ax |grep kermit |grep -v grep
pid=‘ps ax |grep kermit |grep -v grep|awk ’{print $1;}’‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing kermit, PID=’ ${pid}
        kill -9 ${pid}
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

Check to see if `pppd` is still running by executing `/usr/etc/ppp/ppptest`, which should look like this:

```
#!/bin/sh
pid=‘ps ax| grep pppd |grep -v grep|awk ’{print $1;}’‘
if [ X${pid} != "X" ] ; then
        echo ’pppd running: PID=’ ${pid-NONE}
else
        echo ’No pppd running.’
fi
set -x
netstat -n -I ppp0
ifconfig ppp0
```

To hang up the modem, execute `/etc/ppp/kermit.hup`, which should contain:

```
set line /dev/tty01 ; put your modem device here
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
echo \13
```

```
exit
```

Here is an alternate method using `chat` instead of `kermit`:

The following two files are sufficient to accomplish a `pppd` connection.

`/etc/ppp/options`:

```
/dev/cuaa1 115200

crtscts  # enable hardware flow control
modem  # modem control line
connect "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault # remote PPP serve must supply your IP address
         # if the remote host doesn't send your IP during
                # IPCP negotiation, remove this option
passive         # wait for LCP packets
domain <your.domain> # put your domain name here

:  # put the IP of remote PPP host here
         # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to <local_ip>:<remote_ip>

defaultroute # put this if you want that PPP server will be
         # your default router
```

`/etc/ppp/login.chat.script`:

Note: The following should go on a single line.

```
ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDT<phone.number>
  CONNECT "" TIMEOUT 10 ogin:-\\r-ogin: <login-id>
  TIMEOUT 5 sword: <password>
```

Once these are installed and modified correctly, all you need to do is run `pppd`, like so:

```
# pppd
```

25.3.3 Using `pppd` as a Server

`/etc/ppp/options` should contain something similar to the following:

```
crtscts                         # Hardware flow control
netmask 255.255.255.0           # netmask (not required)
192.114.208.20:192.114.208.165  # IP's of local and remote hosts
                                # local ip must be different from one
                                # you assigned to the Ethernet (or other)
                                # interface on your machine.
                                # remote IP is IP address that will be
                                # assigned to the remote machine
```

```
domain ppp.foo.com              # your domain
passive                         # wait for LCP
modem                           # modem line
```

The following /etc/ppp/pppserv script will tell **pppd** to behave as a server:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=‘ps ax |grep pppd |grep -v grep|awk ’{print $1;}’‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing pppd, PID=’ ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=‘ps ax |grep kermit |grep -v grep|awk ’{print $1;}’‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing kermit, PID=’ ${pid}
        kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
ifconfig ppp0 delete

# enable autoanswer mode
kermit -y /etc/ppp/kermit.ans

# run ppp
pppd /dev/tty01 19200
```

Use this /etc/ppp/pppservdown script to stop the server:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=‘ps ax |grep pppd |grep -v grep|awk ’{print $1;}’‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing pppd, PID=’ ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=‘ps ax |grep kermit |grep -v grep|awk ’{print $1;}’‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing kermit, PID=’ ${pid}
        kill -9 ${pid}
fi
ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.noans
```

The following **Kermit** script (/etc/ppp/kermit.ans) will enable/disable autoanswer mode on your modem. It should look like this:

```
set line /dev/tty01
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
inp 5 OK
echo \13
out ATS0=1\13   ; change this to out ATS0=0\13 if you want to disable
                ; autoanswer mode
inp 5 OK
echo \13
exit
```

A script named `/etc/ppp/kermit.dial` is used for dialing and authenticating on the remote host. You will need to customize it for your needs. Put your login and password in this script; you will also need to change the input statement depending on responses from your modem and remote host.

```
;
; put the com line attached to the modem here:
;
set line /dev/tty01
;
; put the modem speed here:
;
set speed 19200
set file type binary            ; full 8 bit file xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
set carrier auto                ; Then SET CARRIER if necessary,
set dial display on             ; Then SET DIAL if necessary,
set input echo on
set input timeout proceed
set input case ignore
def \%x 0                       ; login prompt counter
```

```
goto slhup

:slcmd                          ; put the modem in command mode
echo Put the modem in command mode.
clear                           ; Clear unread characters from input buffer
pause 1
output +++                      ; hayes escape sequence
input 1 OK\13\10                ; wait for OK
if success goto slhup
output \13
pause 1
output at\13
input 1 OK\13\10
if fail goto slcmd              ; if modem doesn't answer OK, try again

:slhup                          ; hang up the phone
clear                           ; Clear unread characters from input buffer
pause 1
echo Hanging up the phone.
output ath0\13                  ; hayes command for on hook
input 2 OK\13\10
if fail goto slcmd              ; if no OK answer, put modem in command mode

:sldial                         ; dial the number
pause 1
echo Dialing.
output atdt9,550311\13\10               ; put phone number here
assign \%x 0                    ; zero the time counter

:look
clear                           ; Clear unread characters from input buffer
increment \%x                   ; Count the seconds
input 1 {CONNECT }
if success goto sllogin
reinput 1 {NO CARRIER\13\10}
if success goto sldial
reinput 1 {NO DIALTONE\13\10}
if success goto slnodial
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 60 goto look
else goto slhup

:sllogin                        ; login
assign \%x 0                    ; zero the time counter
pause 1
echo Looking for login prompt.

:slloop
increment \%x                   ; Count the seconds
clear                           ; Clear unread characters from input buffer
```

```
output \13
;
; put your expected login prompt here:
;
input 1 {Username: }
if success goto sluid
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 10 goto slloop         ; try 10 times to get a login prompt
else goto slhup                 ; hang up and start again if 10 failures

:sluid
;
; put your userid here:
;
output ppp-login\13
input 1 {Password: }
;
; put your password here:
;
output ppp-password\13
input 1 {Entering SLIP mode.}
echo
quit

:slnodial
echo \7No dialtone.  Check the telephone line!\7
exit 1

; local variables:
; mode: csh
; comment-start: "; "
; comment-start-skip: "; "
; end:
```

25.4 Troubleshooting PPP Connections

Contributed by Tom Rhodes.

This section covers a few issues which may arise when using PPP over a modem connection. For instance, perhaps you need to know exactly what prompts the system you are dialing into will present. Some ISPs present the `ssword` prompt, and others will present `password`; if the `ppp` script is not written accordingly, the login attempt will fail. The most common way to debug `ppp` connections is by connecting manually. The following information will walk you through a manual connection step by step.

25.4.1 Check the Device Nodes

If you reconfigured your kernel then you recall the `sio` device. If you did not configure your kernel, there is no reason to worry. Just check the `dmesg` output for the modem device with:

```
# dmesg | grep sio
```

You should get some pertinent output about the `sio` devices. These are the COM ports we need. If your modem acts like a standard serial port then you should see it listed on `sio1`, or `COM2`. If so, you are not required to rebuild the kernel, you just need to make the serial device. You can do this by changing your directory to `/dev` and running the `MAKEDEV` script like above. Now make the serial devices with:

```
# sh MAKEDEV cuaa0 cuaa1 cuaa2 cuaa3
```

which will create the serial devices for your system. When matching up sio modem is on `sio1` or `COM2` if you are in DOS, then your modem device would be `/dev/cuaa1`.

25.4.2 Connecting Manually

Connecting to the Internet by manually controlling `ppp` is quick, easy, and a great way to debug a connection or just get information on how your ISP treats `ppp` client connections. Lets start **PPP** from the command line. Note that in all of our examples we will use *example* as the hostname of the machine running **PPP**. You start `ppp` by just typing `ppp`:

```
# ppp
```

We have now started `ppp`.

```
ppp ON example> set device /dev/cuaa1
```

We set our modem device, in this case it is `cuaa1`.

```
ppp ON example> set speed 115200
```

Set the connection speed, in this case we are using 115,200 kbps.

```
ppp ON example> enable dns
```

Tell `ppp` to configure our resolver and add the nameserver lines to `/etc/resolv.conf`. If `ppp` cannot determine our hostname, we can set one manually later.

```
ppp ON example> term
```

Switch to “terminal” mode so that we can manually control the modem.

```
deflink: Entering terminal mode on /dev/cuaa1
type '~h' for help

at
OK
atdt123456789
```

Use `at` to initialize the modem, then use `atdt` and the number for your ISP to begin the dial in process.

```
CONNECT
```

Confirmation of the connection, if we are going to have any connection problems, unrelated to hardware, here is where we will attempt to resolve them.

```
ISP Login:myusername
```

Here you are prompted for a username, return the prompt with the username that was provided by the ISP.

```
ISP Pass:mypassword
```

This time we are prompted for a password, just reply with the password that was provided by the ISP. Just like logging into FreeBSD, the password will not echo.

```
Shell or PPP:ppp
```

Depending on your ISP this prompt may never appear. Here we are being asked if we wish to use a shell on the provider, or to start ppp. In this example, we have chosen to use ppp as we want an Internet connection.

```
Ppp ON example>
```

Notice that in this example the first p has been capitalized. This shows that we have successfully connected to the ISP.

```
PPp ON example>
```

We have successfully authenticated with our ISP and are waiting for the assigned IP address.

```
PPP ON example>
```

We have made an agreement on an IP address and successfully completed our connection.

```
PPP ON example>add default HISADDR
```

Here we add our default route, we need to do this before we can talk to the outside world as currently the only established connection is with the peer. If this fails due to existing routes you can put a bang character ! in front of the add. Alternatively, you can set this before making the actual connection and it will negotiate a new route accordingly.

If everything went good we should now have an active connection to the Internet, which could be thrown into the background using **CTRL+z** If you notice the PPP return to ppp then we have lost our connection. This is good to know because it shows our connection status. Capital P's show that we have a connection to the ISP and lowercase p's show that the connection has been lost for whatever reason. ppp only has these 2 states.

25.4.2.1 Debugging

If you have a direct line and cannot seem to make a connection, then turn hardware flow CTS/RTS to off with the `set ctsrts off`. This is mainly the case if you are connected to some **PPP** capable terminal servers, where **PPP** hangs when it tries to write data to your communication link, so it would be waiting for a CTS, or Clear To Send signal which may never come. If you use this option however, you should also use the `set accmap` option, which may be required to defeat hardware dependent on passing certain characters from end to end, most of the time XON/XOFF. See the ppp(8) manual page for more information on this option, and how it is used.

If you have an older modem, you may need to use the `set parity even`. Parity is set at none be default, but is used for error checking (with a large increase in traffic) on older modems and some ISPs. You may need this option for the Compuserve ISP.

PPP may not return to the command mode, which is usually a negotiation error where the ISP is waiting for your side to start negotiating. At this point, using the `~p` command will force ppp to start sending the configuration information.

If you never obtain a login prompt, then most likely you need to use PAP or CHAP authentication instead of the UNIX style in the example above. To use PAP or CHAP just add the following options to **PPP** before going into terminal mode:

```
ppp ON example> set authname myusername
```

Where *myusername* should be replaced with the username that was assigned by the ISP.

```
ppp ON example> set authkey mypassword
```

Where *mypassword* should be replaced with the password that was assigned by the ISP.

If you connect fine, but cannot seem to find any domain name, try to use ping(8) with an IP address and see if you can get any return information. If you experience 100 percent (100%) packet loss, then it is most likely that you were not assigned a default route. Double check that the option `add default HISADDR` was set during the connection. If you can connect to a remote IP address then it is possible that a resolver address has not been added to the `/etc/resolv.conf`. This file should look like:

```
domain example.com
nameserver x.x.x.x
nameserver y.y.y.y
```

Where *x.x.x.x* and *y.y.y.y* should be replaced with the IP address of your ISP's DNS servers. This information may or may not have been provided when you signed up, but a quick call to your ISP should remedy that.

You could also have syslog(3) provide a logging function for your **PPP** connection. Just add:

```
!ppp
*.*     /var/log/ppp.log
```

to `/etc/syslog.conf`. In most cases, this functionality already exists.

25.5 Using PPP over Ethernet (PPPoE)

Contributed (from http://node.to/freebsd/how-tos/how-to-freebsd-pppoe.html) by Jim Mock.

This section describes how to set up PPP over Ethernet (PPPoE).

25.5.1 Configuring the Kernel

No kernel configuration is necessary for PPPoE any longer. If the necessary netgraph support is not built into the kernel, it will be dynamically loaded by **ppp**.

25.5.2 Setting Up `ppp.conf`

Here is an example of a working `ppp.conf`:

```
default:
  set log Phase tun command # you can add more detailed logging if you wish
  set ifaddr 10.0.0.1/0 10.0.0.2/0

name_of_service_provider:
  set device PPPoE:xl1 # replace xl1 with your Ethernet device
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
  set dial
  set login
  add default HISADDR
```

25.5.3 Running ppp

As `root`, you can run:

```
# ppp -ddial name_of_service_provider
```

25.5.4 Starting ppp at Boot

Add the following to your `/etc/rc.conf` file:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES" # if you want to enable nat for your local network, otherwise NO
ppp_profile="name_of_service_provider"
```

25.5.5 Using a PPPoE Service Tag

Sometimes it will be necessary to use a service tag to establish your connection. Service tags are used to distinguish between different PPPoE servers attached to a given network.

You should have been given any required service tag information in the documentation provided by your ISP. If you cannot locate it there, ask your ISP's tech support personnel.

As a last resort, you could try the method suggested by the Roaring Penguin PPPoE (http://www.roaringpenguin.com/pppoe/) program which can be found in the Ports Collection. Bear in mind however, this may de-program your modem and render it useless, so think twice before doing it. Simply install the program shipped with the modem by your provider. Then, access the System menu from the program. The name of your profile should be listed there. It is usually *ISP*.

The profile name (service tag) will be used in the PPPoE configuration entry in `ppp.conf` as the provider part of the `set device` command (see the ppp(8) manual page for full details). It should look like this:

```
set device PPPoE:xl1:ISP
```

Do not forget to change *xl1* to the proper device for your Ethernet card.

Do not forget to change *ISP* to the profile you have just found above.

For additional information, see:

- Cheaper Broadband with FreeBSD on DSL (http://renaud.waldura.com/doc/freebsd/pppoe/) by Renaud Waldura.
- Nutzung von T-DSL und T-Online mit FreeBSD (http://www.ruhr.de/home/nathan/FreeBSD/tdsl-freebsd.html) by Udo Erdelhoff (in German).

25.5.6 PPPoE with a 3Com® HomeConnect® ADSL Modem Dual Link

This modem does not follow RFC 2516 (http://www.faqs.org/rfcs/rfc2516.html) (*A Method for transmitting PPP over Ethernet (PPPoE)*, written by L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler). Instead, different packet type codes have been used for the Ethernet frames. Please complain to 3Com (http://www.3com.com/) if you think it should comply with the PPPoE specification.

In order to make FreeBSD capable of communicating with this device, a sysctl must be set. This can be done automatically at boot time by updating `/etc/sysctl.conf`:

```
net.graph.nonstandard_pppoe=1
```

or can be done immediately with the command:

```
# sysctl net.graph.nonstandard_pppoe=1
```

Unfortunately, because this is a system-wide setting, it is not possible to talk to a normal PPPoE client or server and a 3Com HomeConnect® ADSL Modem at the same time.

25.6 Using PPP over ATM (PPPoA)

The following describes how to set up PPP over ATM (PPPoA). PPPoA is a popular choice among European DSL providers.

25.6.1 Using PPPoA with the Alcatel SpeedTouch™ USB

PPPoA support for this device is supplied as a port in FreeBSD because the firmware is distributed under Alcatel's license agreement (http://www.speedtouchdsl.com/disclaimer_lx.htm) and can not be redistributed freely with the base system of FreeBSD.

To install the software, simply use the Ports Collection. Install the `net/pppoa` port and follow the instructions provided with it.

Like many USB devices, the Alcatel SpeedTouch™ USB needs to download firmware from the host computer to operate properly. It is possible to automate this process in FreeBSD so that this transfer takes place whenever the device is plugged into a USB port. The following information can be added to the `/etc/usbd.conf` file to enable this automatic firmware transfer. This file must be edited as the `root` user.

```
device "Alcatel SpeedTouch USB"
```

```
    devname "ugen[0-9]+"
    vendor 0x06b9
    product 0x4061
    attach "/usr/local/sbin/modem_run -f /usr/local/libdata/mgmt.o"
```

To enable the USB daemon, **usbd**, put the following the line into `/etc/rc.conf`:

```
usbd_enable="YES"
```

It is also possible to set up **ppp** to dial up at startup. To do this add the following lines to `/etc/rc.conf`. Again, for this procedure you will need to be logged in as the `root` user.

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_profile="adsl"
```

For this to work correctly you will need to have used the sample `ppp.conf` which is supplied with the `net/pppoa` port.

25.6.2 Using mpd

You can use **mpd** to connect to a variety of services, in particular PPTP services. You can find **mpd** in the Ports Collection, `net/mpd`. Many ADSL modems require that a PPTP tunnel is created between the modem and computer, one such modem is the Alcatel SpeedTouch Home.

First you must install the port, and then you can configure **mpd** to suit your requirements and provider settings. The port places a set of sample configuration files which are well documented in *`PREFIX`*`/etc/mpd/`. Note here that *`PREFIX`* means the directory into which your ports are installed, this defaults to `/usr/local/`. A complete guide to configure **mpd** is available in HTML format once the port has been installed. It is placed in *`PREFIX`*`/share/doc/mpd/`. Here is a sample configuration for connecting to an ADSL service with **mpd**. The configuration is spread over two files, first the `mpd.conf`:

```
default:
    load adsl

adsl:
    new -i ng0 adsl adsl
    set bundle authname username ❶
    set bundle password password ❷
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
```

```
    set iface idle 0

    open
```

❶ The username used to authenticate with your ISP.

❷ The password used to authenticate with your ISP.

The `mpd.links` file contains information about the link, or links, you wish to establish. An example `mpd.links` to accompany the above example is given beneath:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate outcall
    set pptp self 10.0.0.1 ❶
    set pptp peer 10.0.0.138 ❷
```

❶ The IP address of your FreeBSD computer which you will be using **mpd** from.

❷ The IP address of your ADSL modem. For the Alcatel SpeedTouch Home this address defaults to `10.0.0.138`.

It is possible to initialize the connection easily by issuing the following command as `root`:

```
# mpd -b adsl
```

You can see the status of the connection with the following command:

```
% ifconfig ng0
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
     inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Using **mpd** is the recommended way to connect to an ADSL service with FreeBSD.

25.6.3 Using pptpclient

It is also possible to use FreeBSD to connect to other PPPoA services using `net/pptpclient`.

To use `net/pptpclient` to connect to a DSL service, install the port or package and edit your `/etc/ppp/ppp.conf`. You will need to be `root` to perform both of these operations. An example section of `ppp.conf` is given below. For further information on `ppp.conf` options consult the **ppp** manual page, ppp(8).

```
adsl:
 set log phase chat lcp ipcp ccp tun command
 set timeout 0
 enable dns
 set authname username ❶
 set authkey password ❷
 set ifaddr 0 0
 add default HISADDR
```

❶ The username of your account with the DSL provider.

❷ The password for your account.

Warning: Because you must put your account's password in the `ppp.conf` file in plain text form you should make sure than nobody can read the contents of this file. The following series of commands will make sure the file is only readable by the `root` account. Refer to the manual pages for chmod(1) and chown(8) for further information.

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

This will open a tunnel for a PPP session to your DSL router. Ethernet DSL modems have a preconfigured LAN IP address which you connect to. In the case of the Alcatel SpeedTouch Home this address is `10.0.0.138`. Your router documentation should tell you which address your device uses. To open the tunnel and start a PPP session execute the following command:

```
# pptp address adsl
```

Tip: You may wish to add an ampersand ("&") to the end of the previous command because **pptp** will not return your prompt to you otherwise.

A `tun` virtual tunnel device will be created for interaction between the **pptp** and **ppp** processes. Once you have been returned to your prompt, or the **pptp** process has confirmed a connection you can examine the tunnel like so:

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
        inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffff00
        Opened by PID 918
```

If you are unable to connect, check the configuration of your router, which is usually accessible via **telnet** or with a web browser. If you still cannot connect you should examine the output of the `pptp` command and the contents of the **ppp** log file, `/var/log/ppp.log` for clues.

25.7 Using SLIP

Originally contributed by Satoshi Asami. With input from Guy Helmer and Piero Serini.

25.7.1 Setting Up a SLIP Client

The following is one way to set up a FreeBSD machine for SLIP on a static host network. For dynamic hostname assignments (your address changes each time you dial up), you probably need to have a more complex setup.

First, determine which serial port your modem is connected to. Many people set up a symbolic link, such as `/dev/modem`, to point to the real device name, `/dev/cuaaN` (or `/dev/cuadN` under FreeBSD 6.X). This allows you to abstract the actual device name should you ever need to move the modem to a different port. It can become quite cumbersome when you need to fix a bunch of files in `/etc` and `.kermrc` files all over the system!

Note: `/dev/cuaa0` (or `/dev/cuad0` under FreeBSD 6.X) is `COM1`, `cuaa1` (or `/dev/cuad1`) is `COM2`, etc.

Make sure you have the following in your kernel configuration file:

```
device   sl
```

Under FreeBSD 4.X, use instead the following line:

```
pseudo-device   sl      1
```

It is included in the `GENERIC` kernel, so this should not be a problem unless you have deleted it.

25.7.1.1 Things You Have to Do Only Once

1. Add your home machine, the gateway and nameservers to your `/etc/hosts` file. Ours looks like this:

   ```
   127.0.0.1               localhost loghost
   136.152.64.181          water.CS.Example.EDU water.CS water
   136.152.64.1            inr-3.CS.Example.EDU inr-3 slip-gateway
   128.32.136.9            ns1.Example.EDU ns1
   128.32.136.12           ns2.Example.EDU ns2
   ```

2. Make sure you have `hosts` before `bind` in your `/etc/host.conf` on FreeBSD versions prior to 5.0. Since FreeBSD 5.0, the system uses the file `/etc/nsswitch.conf` instead, make sure you have `files` before `dns` in the `hosts` line of this file. Without these parameters funny things may happen.
3. Edit the `/etc/rc.conf` file.
   1. Set your hostname by editing the line that says:

      ```
      hostname="myname.my.domain"
      ```

      Your machine's full Internet hostname should be placed here.
   2. Designate the default router by changing the line:

      ```
      defaultrouter="NO"
      ```

      to:

      ```
      defaultrouter="slip-gateway"
      ```
4. Make a file `/etc/resolv.conf` which contains:

   ```
   domain CS.Example.EDU
   nameserver 128.32.136.9
   nameserver 128.32.136.12
   ```

   As you can see, these set up the nameserver hosts. Of course, the actual domain names and addresses depend on your environment.
5. Set the password for `root` and `toor` (and any other accounts that do not have a password).
6. Reboot your machine and make sure it comes up with the correct hostname.

25.7.1.2 Making a SLIP Connection

1. Dial up, type `slip` at the prompt, enter your machine name and password. What is required to be entered depends on your environment. If you use **Kermit**, you can try a script like this:

```
# kermit setup
set modem hayes
set line /dev/modem
set speed 115200
set parity none
set flow rts/cts
set terminal bytesize 8
set file type binary
# The next macro will dial up and login
define slip dial 643-9600, input 10 =>, if failure stop, -
output slip\x0d, input 10 Username:, if failure stop, -
output silvia\x0d, input 10 Password:, if failure stop, -
output ***\x0d, echo \x0aCONNECTED\x0a
```

   Of course, you have to change the username and password to fit yours. After doing so, you can just type `slip` from the **Kermit** prompt to connect.

   > **Note:** Leaving your password in plain text anywhere in the filesystem is generally a *bad* idea. Do it at your own risk.

2. Leave the **Kermit** there (you can suspend it by **Ctrl**-**z**) and as `root`, type:

```
# slattach -h -c -s 115200 /dev/modem
```

   If you are able to `ping` hosts on the other side of the router, you are connected! If it does not work, you might want to try `-a` instead of `-c` as an argument to `slattach`.

25.7.1.3 How to Shutdown the Connection

Do the following:

```
# kill -INT ‘cat /var/run/slattach.modem.pid‘
```

to kill `slattach`. Keep in mind you must be `root` to do the above. Then go back to `kermit` (by running `fg` if you suspended it) and exit from it (**q**).

The slattach(8) manual page says you have to use `ifconfig sl0 down` to mark the interface down, but this does not seem to make any difference. (`ifconfig sl0` reports the same thing.)

Some times, your modem might refuse to drop the carrier. In that case, simply start `kermit` and quit it again. It usually goes out on the second try.

25.7.1.4 Troubleshooting

If it does not work, feel free to ask on freebsd-net (http://lists.FreeBSD.org/mailman/listinfo/freebsd-net) mailing list. The things that people tripped over so far:

- Not using `-c` or `-a` in `slattach` (This should not be fatal, but some users have reported that this solves their problems.)
- Using `s10` instead of `sl0` (might be hard to see the difference on some fonts).
- Try `ifconfig sl0` to see your interface status. For example, you might get:

```
# ifconfig sl0
sl0: flags=10<POINTOPOINT>
        inet 136.152.64.181 --> 136.152.64.1 netmask ffffff00
```

- If you get "`no route to host`" messages from ping(8), there may be a problem with your routing table. You can use the `netstat -r` command to display the current routes :

```
# netstat -r
Routing tables
Destination      Gateway            Flags     Refs     Use  IfaceMTU    Rtt    Netmasks:

(root node)
(root node)

Route Tree for Protocol Family inet:
(root node) =>
default          inr-3.Example.EDU  UG          8   224515  sl0 -      -
localhost.Exampl localhost.Example. UH          5    42127  lo0 -       0.438
inr-3.Example.ED water.CS.Example.E UH          1        0  sl0 -      -
water.CS.Example localhost.Example. UGH        34 47641234  lo0 -       0.438
(root node)
```

The preceding examples are from a relatively busy system. The numbers on your system will vary depending on network activity.

25.7.2 Setting Up a SLIP Server

This document provides suggestions for setting up SLIP Server services on a FreeBSD system, which typically means configuring your system to automatically startup connections upon login for remote SLIP clients.

25.7.2.1 Prerequisites

This section is very technical in nature, so background knowledge is required. It is assumed that you are familiar with the TCP/IP network protocol, and in particular, network and node addressing, network address masks, subnetting, routing, and routing protocols, such as RIP. Configuring SLIP services on a dial-up server requires a knowledge of these concepts, and if you are not familiar with them, please read a copy of either Craig Hunt's *TCP/IP Network Administration* published by O'Reilly & Associates, Inc. (ISBN Number 0-937175-82-X), or Douglas Comer's books on the TCP/IP protocol.

It is further assumed that you have already set up your modem(s) and configured the appropriate system files to allow logins through your modems. If you have not prepared your system for this yet, please see Section 24.4 for details on

dialup services configuration. You may also want to check the manual pages for sio(4) for information on the serial port device driver and ttys(5), gettytab(5), getty(8), & init(8) for information relevant to configuring the system to accept logins on modems, and perhaps stty(1) for information on setting serial port parameters (such as `clocal` for directly-connected serial interfaces).

25.7.2.2 Quick Overview

In its typical configuration, using FreeBSD as a SLIP server works as follows: a SLIP user dials up your FreeBSD SLIP Server system and logs in with a special SLIP login ID that uses `/usr/sbin/sliplogin` as the special user's shell. The `sliplogin` program browses the file `/etc/sliphome/slip.hosts` to find a matching line for the special user, and if it finds a match, connects the serial line to an available SLIP interface and then runs the shell script `/etc/sliphome/slip.login` to configure the SLIP interface.

25.7.2.2.1 An Example of a SLIP Server Login

For example, if a SLIP user ID were `Shelmerg`, `Shelmerg`'s entry in `/etc/master.passwd` would look something like this:

```
Shelmerg:password:1964:89::0:0:Guy Helmer - SLIP:/usr/users/Shelmerg:/usr/sbin/sliplogin
```

When `Shelmerg` logs in, `sliplogin` will search `/etc/sliphome/slip.hosts` for a line that had a matching user ID; for example, there may be a line in `/etc/sliphome/slip.hosts` that reads:

```
Shelmerg        dc-slip sl-helmer       0xfffffc00    autocomp
```

`sliplogin` will find that matching line, hook the serial line into the next available SLIP interface, and then execute `/etc/sliphome/slip.login` like this:

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer 0xfffffc00 autocomp
```

If all goes well, `/etc/sliphome/slip.login` will issue an `ifconfig` for the SLIP interface to which `sliplogin` attached itself (SLIP interface 0, in the above example, which was the first parameter in the list given to `slip.login`) to set the local IP address (`dc-slip`), remote IP address (`sl-helmer`), network mask for the SLIP interface (`0xfffffc00`), and any additional flags (`autocomp`). If something goes wrong, `sliplogin` usually logs good informational messages via the **syslogd** daemon facility, which usually logs to `/var/log/messages` (see the manual pages for syslogd(8) and syslog.conf(5) and perhaps check `/etc/syslog.conf` to see to what **syslogd** is logging and where it is logging to).

25.7.2.3 Kernel Configuration

FreeBSD's default kernel (`GENERIC`) comes with SLIP (sl(4)) support; in case of a custom kernel, you have to add the following line to your kernel configuration file:

```
device   sl
```

Under FreeBSD 4.X, use instead the following line:

```
pseudo-device   sl      2
```

Note: The number at the end of the line is the maximum number of SLIP connections that may be operating simultaneously. Since FreeBSD 5.0, the sl(4) driver is "auto-cloning".

By default, your FreeBSD machine will not forward packets. If you want your FreeBSD SLIP Server to act as a router, you will have to edit the `/etc/rc.conf` file and change the setting of the `gateway_enable` variable to `YES`.

You will then need to reboot for the new settings to take effect.

Please refer to Chapter 8 on Configuring the FreeBSD Kernel for help in reconfiguring your kernel.

25.7.2.4 Sliplogin Configuration

As mentioned earlier, there are three files in the `/etc/sliphome` directory that are part of the configuration for `/usr/sbin/sliplogin` (see sliplogin(8) for the actual manual page for `sliplogin`): `slip.hosts`, which defines the SLIP users and their associated IP addresses; `slip.login`, which usually just configures the SLIP interface; and (optionally) `slip.logout`, which undoes `slip.login`'s effects when the serial connection is terminated.

25.7.2.4.1 `slip.hosts` Configuration

`/etc/sliphome/slip.hosts` contains lines which have at least four items separated by whitespace:

- SLIP user's login ID
- Local address (local to the SLIP server) of the SLIP link
- Remote address of the SLIP link
- Network mask

The local and remote addresses may be host names (resolved to IP addresses by `/etc/hosts` or by the domain name service, depending on your specifications in the file `/etc/nsswitch.conf`, or in `/etc/host.conf` if you use FreeBSD 4.X), and the network mask may be a name that can be resolved by a lookup into `/etc/networks`. On a sample system, `/etc/sliphome/slip.hosts` looks like this:

```
#
# login local-addr      remote-addr     mask            opt1    opt2
#                                               (normal,compress,noicmp)
#
Shelmerg  dc-slip       sl-helmerg      0xfffffc00      autocomp
```

At the end of the line is one or more of the options:

- `normal` ——no header compression
- `compress` ——compress headers
- `autocomp` ——compress headers if the remote end allows it
- `noicmp` ——disable ICMP packets (so any "ping" packets will be dropped instead of using up your bandwidth)

Your choice of local and remote addresses for your SLIP links depends on whether you are going to dedicate a TCP/IP subnet or if you are going to use "proxy ARP" on your SLIP server (it is not "true" proxy ARP, but that is the terminology used in this section to describe it). If you are not sure which method to select or how to assign IP

addresses, please refer to the TCP/IP books referenced in the SLIP Prerequisites (Section 25.7.2.1) and/or consult your IP network manager.

If you are going to use a separate subnet for your SLIP clients, you will need to allocate the subnet number out of your assigned IP network number and assign each of your SLIP client's IP numbers out of that subnet. Then, you will probably need to configure a static route to the SLIP subnet via your SLIP server on your nearest IP router.

Otherwise, if you will use the "proxy ARP" method, you will need to assign your SLIP client's IP addresses out of your SLIP server's Ethernet subnet, and you will also need to adjust your `/etc/sliphome/slip.login` and `/etc/sliphome/slip.logout` scripts to use arp(8) to manage the proxy-ARP entries in the SLIP server's ARP table.

25.7.2.4.2 `slip.login` Configuration

The typical `/etc/sliphome/slip.login` file looks like this:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

This `slip.login` file merely runs `ifconfig` for the appropriate SLIP interface with the local and remote addresses and network mask of the SLIP interface.

If you have decided to use the "proxy ARP" method (instead of using a separate subnet for your SLIP clients), your `/etc/sliphome/slip.login` file will need to look something like this:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Answer ARP requests for the SLIP client with our Ethernet addr
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

The additional line in this `slip.login`, `arp -s $5 00:11:22:33:44:55 pub`, creates an ARP entry in the SLIP server's ARP table. This ARP entry causes the SLIP server to respond with the SLIP server's Ethernet MAC address whenever another IP node on the Ethernet asks to speak to the SLIP client's IP address.

When using the example above, be sure to replace the Ethernet MAC address (`00:11:22:33:44:55`) with the MAC address of your system's Ethernet card, or your "proxy ARP" will definitely not work! You can discover your SLIP server's Ethernet MAC address by looking at the results of running `netstat -i`; the second line of the output should look something like:

```
ed0   1500  <Link>0.2.c1.28.5f.4a         191923 0   129457     0   116
```

This indicates that this particular system's Ethernet MAC address is `00:02:c1:28:5f:4a` ——the periods in the Ethernet MAC address given by `netstat -i` must be changed to colons and leading zeros should be added to each single-digit hexadecimal number to convert the address into the form that arp(8) desires; see the manual page on arp(8) for complete information on usage.

> **Note:** When you create `/etc/sliphome/slip.login` and `/etc/sliphome/slip.logout`, the "execute" bit (i.e., `chmod 755 /etc/sliphome/slip.login /etc/sliphome/slip.logout`) must be set, or `sliplogin` will be unable to execute it.

25.7.2.4.3 `slip.logout` Configuration

`/etc/sliphome/slip.logout` is not strictly needed (unless you are implementing "proxy ARP"), but if you decide to create it, this is an example of a basic `slip.logout` script:

```
#!/bin/sh -
#
#       slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

If you are using "proxy ARP", you will want to have `/etc/sliphome/slip.logout` remove the ARP entry for the SLIP client:

```
#!/bin/sh -
#
#       @(#)slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
# Quit answering ARP requests for the SLIP client
/usr/sbin/arp -d $5
```

The `arp -d $5` removes the ARP entry that the "proxy ARP" `slip.login` added when the SLIP client logged in.

It bears repeating: make sure `/etc/sliphome/slip.logout` has the execute bit set after you create it (i.e., `chmod 755 /etc/sliphome/slip.logout`).

25.7.2.5 Routing Considerations

If you are not using the "proxy ARP" method for routing packets between your SLIP clients and the rest of your network (and perhaps the Internet), you will probably have to add static routes to your closest default router(s) to route your SLIP clients subnet via your SLIP server.

25.7.2.5.1 Static Routes

Adding static routes to your nearest default routers can be troublesome (or impossible if you do not have authority to do so...). If you have a multiple-router network in your organization, some routers, such as those made by Cisco and Proteon, may not only need to be configured with the static route to the SLIP subnet, but also need to be told which static routes to tell other routers about, so some expertise and troubleshooting/tweaking may be necessary to get static-route-based routing to work.

*25.7.2.5.2 Running **GateD®***

> **Note: GateD** is proprietary software now and will not be available as source code to the public anymore (more info on the GateD (http://www.gated.org/) website). This section only exists to ensure backwards compatibility for those that are still using an older version.

An alternative to the headaches of static routes is to install **GateD** on your FreeBSD SLIP server and configure it to use the appropriate routing protocols (RIP/OSPF/BGP/EGP) to tell other routers about your SLIP subnet. You will need to write a `/etc/gated.conf` file to configure your **GateD**; here is a sample, similar to what the author used on a FreeBSD SLIP server:

```
#
# gated configuration file for dc.dsu.edu; for gated version 3.5alpha5
# Only broadcast RIP information for xxx.xxx.yy out the ed Ethernet interface
#
#
# tracing options
#
traceoptions "/var/tmp/gated.output" replace size 100k files 2 general ;

rip yes {
  interface sl noripout noripin ;
  interface ed ripin ripout version 1 ;
  traceoptions route ;
} ;

#
# Turn on a bunch of tracing info for the interface to the kernel:
```

```
kernel {
  traceoptions remnants request routes info interface ;
} ;

#
# Propagate the route to xxx.xxx.yy out the Ethernet interface via RIP
#

export proto rip interface ed {
  proto direct {
      xxx.xxx.yy mask 255.255.252.0 metric 1; # SLIP connections
  } ;
} ;

#
# Accept routes from RIP via ed Ethernet interfaces

import proto rip interface ed {
  all ;
} ;
```

The above sample `gated.conf` file broadcasts routing information regarding the SLIP subnet *`xxx.xxx.yy`* via RIP onto the Ethernet; if you are using a different Ethernet driver than the `ed` driver, you will need to change the references to the `ed` interface appropriately. This sample file also sets up tracing to `/var/tmp/gated.output` for debugging **GateD**'s activity; you can certainly turn off the tracing options if **GateD** works correctly for you. You will need to change the *`xxx.xxx.yy`*'s into the network address of your own SLIP subnet (be sure to change the net mask in the `proto direct` clause as well).

Once you have installed and configured **GateD** on your system, you will need to tell the FreeBSD startup scripts to run **GateD** in place of **routed**. The easiest way to accomplish this is to set the `router` and `router_flags` variables in `/etc/rc.conf`. Please see the manual page for **GateD** for information on command-line parameters.

Chapter 26 電子郵件

Original work by Bill Lloyd. Rewritten by Jim Mock.

26.1 概述

“電子郵件”或者俗稱的email，乃是現今使用最廣泛的溝通方式之一。本章主要介紹如何在FreeBSD 上安裝、設定email 服務，以及如何在FreeBSD 收發信件；然而這並不是完整的參考手冊，實際上許多需考量的重要事項並未提及，若欲瞭解細節請參閱Appendix B 內的參考書籍。

讀完這章，您將了解：

- 哪些軟體元件與收發電子郵件有關。
- FreeBSD 內的**sendmail** 基本設定檔在哪。
- 遠端信箱與本機信箱的區別。
- 如何阻擋spammer(垃圾郵件製造者)非法運用您的郵件伺服器作為relay(轉發中繼點)。
- 如何安裝、設定其他Mail Transfer Agent(MTA) 來取代**sendmail**。
- 如何處理常見的郵件伺服器問題。
- 如何使用UUCP 來進行SMTP。
- 如何設定系統，使其只能發送郵件。
- 如何在撥接上網環境中，收發郵件。
- 如何設定SMTP 驗證，以加強安全性。
- 如何安裝、使用Mail User Agent(MUA) 程式，比如**mutt** 來收發郵件。
- 如何從遠端POP 或IMAP 主機去下載郵件。
- 如何在收信方面，自動套用郵件過濾。

在開始閱讀這章之前，您需要：

- 先設好你的網路(Chapter 29)。
- 能正確為郵件伺服器設定DNS (Chapter 27)。
- 知道如何透過port/package 安裝軟體(Chapter 4)。

26.2 使用電子郵件

在email 交換的過程中有5 個主要部分，分別是：MUA、MTA、DNS、 遠端或本機的信箱，當然還有郵件主機本身。

26.2.1 MUA 程式

包括一些文字介面的程式，像是**mutt**、**pine**、**elm**、and `mail`，以及GUI 介面的程式，像是**balsa**、**xfmail** 等等。此外，還有更“複雜的” 像是WWW 瀏覽器。這些程式會郵件處理交給“郵件主機”，或者透過呼叫MTA(若有的話)或者是透過TCP 來傳遞郵件。

26.2.2 Mailhost Server Daemon

FreeBSD ships with **sendmail** by default, but also support numerous other mail server daemons, just some of which include:

- **exim**;
- **postfix**;
- **qmail**.

The server daemon usually has two functions——it is responsible for receiving incoming mail as well as delivering outgoing mail. It is *not* responsible for the collection of mail using protocols such as POP or IMAP to read your email, nor does it allow connecting to local `mbox` or Maildir mailboxes. You may require an additional daemon for that.

> **Warning:** Older versions of **sendmail** have some serious security issues which may result in an attacker gaining local and/or remote access to your machine. Make sure that you are running a current version to avoid these problems. Optionally, install an alternative MTA from the FreeBSD Ports Collection.

26.2.3 Email and DNS

The Domain Name System (DNS) and its daemon `named` play a large role in the delivery of email. In order to deliver mail from your site to another, the server daemon will look up the remote site in the DNS to determine the host that will receive mail for the destination. This process also occurs when mail is sent from a remote host to your mail server.

DNS is responsible for mapping hostnames to IP addresses, as well as for storing information specific to mail delivery, known as MX records. The MX (Mail eXchanger) record specifies which host, or hosts, will receive mail for a particular domain. If you do not have an MX record for your hostname or domain, the mail will be delivered directly to your host provided you have an A record pointing your hostname to your IP address.

You may view the MX records for any domain by using the host(1) command, as seen in the example below:

```
% host -t mx FreeBSD.org
FreeBSD.org mail is handled (pri=10) by mx1.FreeBSD.org
```

26.2.4 Receiving Mail

Receiving mail for your domain is done by the mail host. It will collect all mail sent to your domain and store it either in `mbox` (the default method for storing mail) or Maildir format, depending on your configuration. Once mail has been stored, it may either be read locally using applications such as mail(1) or **mutt**, or remotely accessed and

collected using protocols such as POP or IMAP. This means that should you only wish to read mail locally, you are not required to install a POP or IMAP server.

26.2.4.1 Accessing remote mailboxes using POP and IMAP

In order to access mailboxes remotely, you are required to have access to a POP or IMAP server. These protocols allow users to connect to their mailboxes from remote locations with ease. Though both POP and IMAP allow users to remotely access mailboxes, IMAP offers many advantages, some of which are:

- IMAP can store messages on a remote server as well as fetch them.
- IMAP supports concurrent updates.
- IMAP can be extremely useful over low-speed links as it allows users to fetch the structure of messages without downloading them; it can also perform tasks such as searching on the server in order to minimize data transfer between clients and servers.

In order to install a POP or IMAP server, the following steps should be performed:

1. Choose an IMAP or POP server that best suits your needs. The following POP and IMAP servers are well known and serve as some good examples:
   - **qpopper**;
   - **teapop**;
   - **imap-uw**;
   - **courier-imap**;
2. Install the POP or IMAP daemon of your choosing from the ports collection.
3. Where required, modify `/etc/inetd.conf` to load the POP or IMAP server.

> **Warning:** It should be noted that both POP and IMAP transmit information, including username and password credentials in clear-text. This means that if you wish to secure the transmission of information across these protocols, you should consider tunneling sessions over ssh(1). Tunneling sessions is described in Section 14.11.8.

26.2.4.2 Accessing local mailboxes

Mailboxes may be accessed locally by directly utilizing MUAs on the server on which the mailbox resides. This can be done using applications such as **mutt** or mail(1).

26.2.5 The Mail Host

The mail host is the name given to a server that is responsible for delivering and receiving mail for your host, and possibly your network.

26.3 sendmail Configuration

Contributed by Christopher Shumway.

sendmail(8) is the default Mail Transfer Agent (MTA) in FreeBSD. **sendmail**'s job is to accept mail from Mail User Agents (MUA) and deliver it to the appropriate mailer as defined by its configuration file. **sendmail** can also accept network connections and deliver mail to local mailboxes or deliver it to another program.

sendmail uses the following configuration files:

Filename	Function
`/etc/mail/access`	**sendmail** access database file
`/etc/mail/aliases`	Mailbox aliases
`/etc/mail/local-host-names`	Lists of hosts **sendmail** accepts mail for
`/etc/mail/mailer.conf`	Mailer program configuration
`/etc/mail/mailertable`	Mailer delivery table
`/etc/mail/sendmail.cf`	**sendmail** master configuration file
`/etc/mail/virtusertable`	Virtual users and domain tables

26.3.1 `/etc/mail/access`

The access database defines what host(s) or IP addresses have access to the local mail server and what kind of access they have. Hosts can be listed as `OK`, `REJECT`, `RELAY` or simply passed to **sendmail**'s error handling routine with a given mailer error. Hosts that are listed as `OK`, which is the default, are allowed to send mail to this host as long as the mail's final destination is the local machine. Hosts that are listed as `REJECT` are rejected for all mail connections. Hosts that have the `RELAY` option for their hostname are allowed to send mail for any destination through this mail server.

Example 26-1. Configuring the sendmail Access Database

```
cyberspammer.com                550 We do not accept mail from spammers
FREE.STEALTH.MAILER@            550 We do not accept mail from spammers
another.source.of.spam          REJECT
okay.cyberspammer.com           OK
128.32                          RELAY
```

In this example we have five entries. Mail senders that match the left hand side of the table are affected by the action on the right side of the table. The first two examples give an error code to **sendmail**'s error handling routine. The message is printed to the remote host when a mail matches the left hand side of the table. The next entry rejects mail from a specific host on the Internet, `another.source.of.spam`. The next entry accepts mail connections from a host `okay.cyberspammer.com`, which is more exact than the `cyberspammer.com` line above. More specific matches override less exact matches. The last entry allows relaying of electronic mail from hosts with an IP address that begins with `128.32`. These hosts would be able to send mail through this mail server that are destined for other mail servers.

When this file is updated, you need to run `make` in `/etc/mail/` to update the database.

26.3.2 /etc/mail/aliases

The aliases database contains a list of virtual mailboxes that are expanded to other user(s), files, programs or other aliases. Here are a few examples that can be used in /etc/mail/aliases:

Example 26-2. Mail Aliases

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

The file format is simple; the mailbox name on the left side of the colon is expanded to the target(s) on the right. The first example simply expands the mailbox root to the mailbox localuser, which is then looked up again in the aliases database. If no match is found, then the message is delivered to the local user localuser. The next example shows a mail list. Mail to the mailbox ftp-bugs is expanded to the three local mailboxes joe, eric, and paul. Note that a remote mailbox could be specified as <user@example.com>. The next example shows writing mail to a file, in this case /dev/null. The last example shows sending mail to a program, in this case the mail message is written to the standard input of /usr/local/bin/procmail through a UNIX pipe.

When this file is updated, you need to run make in /etc/mail/ to update the database.

26.3.3 /etc/mail/local-host-names

This is a list of hostnames sendmail(8) is to accept as the local host name. Place any domains or hosts that **sendmail** is to be receiving mail for. For example, if this mail server was to accept mail for the domain example.com and the host mail.example.com, its local-host-names might look something like this:

```
example.com
mail.example.com
```

When this file is updated, sendmail(8) needs to be restarted to read the changes.

26.3.4 /etc/mail/sendmail.cf

sendmail's master configuration file, sendmail.cf controls the overall behavior of **sendmail**, including everything from rewriting e-mail addresses to printing rejection messages to remote mail servers. Naturally, with such a diverse role, this configuration file is quite complex and its details are a bit out of the scope of this section. Fortunately, this file rarely needs to be changed for standard mail servers.

The master **sendmail** configuration file can be built from m4(1) macros that define the features and behavior of **sendmail**. Please see /usr/src/contrib/sendmail/cf/README for some of the details.

When changes to this file are made, **sendmail** needs to be restarted for the changes to take effect.

26.3.5 /etc/mail/virtusertable

The virtusertable maps mail addresses for virtual domains and mailboxes to real mailboxes. These mailboxes can be local, remote, aliases defined in /etc/mail/aliases or files.

Example 26-3. Example Virtual Domain Mail Map

```
root@example.com                root
postmaster@example.com          postmaster@noc.example.net
@example.com                    joe
```

In the above example, we have a mapping for a domain `example.com`. This file is processed in a first match order down the file. The first item maps <`root@example.com`> to the local mailbox `root`. The next entry maps <`postmaster@example.com`> to the mailbox `postmaster` on the host `noc.example.net`. Finally, if nothing from `example.com` has matched so far, it will match the last mapping, which matches every other mail message addressed to someone at `example.com`. This will be mapped to the local mailbox `joe`.

26.4 Changing Your Mail Transfer Agent

Written by Andrew Boothman. Information taken from e-mails written by Gregory Neil Shapiro.

As already mentioned, FreeBSD comes with **sendmail** already installed as your MTA (Mail Transfer Agent). Therefore by default it is in charge of your outgoing and incoming mail.

However, for a variety of reasons, some system administrators want to change their system's MTA. These reasons range from simply wanting to try out another MTA to needing a specific feature or package which relies on another mailer. Fortunately, whatever the reason, FreeBSD makes it easy to make the change.

26.4.1 Install a New MTA

You have a wide choice of MTAs available. A good starting point is the FreeBSD Ports Collection where you will be able to find many. Of course you are free to use any MTA you want from any location, as long as you can make it run under FreeBSD.

Start by installing your new MTA. Once it is installed it gives you a chance to decide if it really fulfills your needs, and also gives you the opportunity to configure your new software before getting it to take over from **sendmail**. When doing this, you should be sure that installing the new software will not attempt to overwrite system binaries such as `/usr/bin/sendmail`. Otherwise, your new mail software has essentially been put into service before you have configured it.

Please refer to your chosen MTA's documentation for information on how to configure the software you have chosen.

26.4.2 Disable sendmail

The procedure used to start **sendmail** changed significantly between 4.5-RELEASE, 4.6-RELEASE, and later releases. Therefore, the procedure used to disable it is subtly different.

> **Warning:** If you disable **sendmail**'s outgoing mail service, it is important that you replace it with an alternative mail delivery system. If you choose not to, system functions such as periodic(8) will be unable to deliver their results by e-mail as they would normally expect to. Many parts of your system may expect to have a functional **sendmail**-compatible system. If applications continue to use **sendmail**'s binaries to try to send e-mail after you have disabled them, mail could go into an inactive **sendmail** queue, and never be delivered.

26.4.2.1 FreeBSD 4.5-STABLE before 2002/4/4 and Earlier (Including 4.5-RELEASE and Earlier)

Enter:

```
sendmail_enable="NO"
```

into `/etc/rc.conf`. This will disable **sendmail**'s incoming mail service, but if `/etc/mail/mailer.conf` (see below) is not changed, **sendmail** will still be used to send e-mail.

26.4.2.2 FreeBSD 4.5-STABLE after 2002/4/4 (Including 4.6-RELEASE and Later)

In order to completely disable **sendmail**, including the outgoing mail service, you must use

```
sendmail_enable="NONE"
```

in `/etc/rc.conf`.

If you only want to disable **sendmail**'s incoming mail service, you should set

```
sendmail_enable="NO"
```

in `/etc/rc.conf`. However, if incoming mail is disabled, local delivery will still function. More information on **sendmail**'s startup options is available from the rc.sendmail(8) manual page.

26.4.2.3 FreeBSD 5.0-STABLE and Later

In order to completely disable **sendmail**, including the outgoing mail service, you must use

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

in `/etc/rc.conf`.

If you only want to disable **sendmail**'s incoming mail service, you should set

```
sendmail_enable="NO"
```

in `/etc/rc.conf`. More information on **sendmail**'s startup options is available from the rc.sendmail(8) manual page.

26.4.3 Running Your New MTA on Boot

You may have a choice of two methods for running your new MTA on boot, again depending on what version of FreeBSD you are running.

26.4.3.1 FreeBSD 4.5-STABLE before 2002/4/11 (Including 4.5-RELEASE and Earlier)

Add a script to `/usr/local/etc/rc.d/` that ends in `.sh` and is executable by `root`. The script should accept `start` and `stop` parameters. At startup time the system scripts will execute the command

```
/usr/local/etc/rc.d/supermailer.sh start
```

which you can also use to manually start the server. At shutdown time, the system scripts will use the `stop` option, running the command

```
/usr/local/etc/rc.d/supermailer.sh stop
```

which you can also use to manually stop the server while the system is running.

26.4.3.2 FreeBSD 4.5-STABLE after 2002/4/11 (Including 4.6-RELEASE and Later)

With later versions of FreeBSD, you can use the above method or you can set

```
mta_start_script="filename"
```

in `/etc/rc.conf`, where *`filename`* is the name of some script that you want executed at boot to start your MTA.

26.4.4 Replacing sendmail as the System's Default Mailer

The program **sendmail** is so ubiquitous as standard software on UNIX systems that some software just assumes it is already installed and configured. For this reason, many alternative MTA's provide their own compatible implementations of the **sendmail** command-line interface; this facilitates using them as "drop-in" replacements for **sendmail**.

Therefore, if you are using an alternative mailer, you will need to make sure that software trying to execute standard **sendmail** binaries such as `/usr/bin/sendmail` actually executes your chosen mailer instead. Fortunately, FreeBSD provides a system called mailwrapper(8) that does this job for you.

When **sendmail** is operating as installed, you will find something like the following in `/etc/mail/mailer.conf`:

```
sendmail  /usr/libexec/sendmail/sendmail
send-mail /usr/libexec/sendmail/sendmail
mailq  /usr/libexec/sendmail/sendmail
newaliases /usr/libexec/sendmail/sendmail
hoststat /usr/libexec/sendmail/sendmail
purgestat /usr/libexec/sendmail/sendmail
```

This means that when any of these common commands (such as `sendmail` itself) are run, the system actually invokes a copy of mailwrapper named `sendmail`, which checks `mailer.conf` and executes `/usr/libexec/sendmail/sendmail` instead. This system makes it easy to change what binaries are actually executed when these default `sendmail` functions are invoked.

Therefore if you wanted `/usr/local/supermailer/bin/sendmail-compat` to be run instead of **sendmail**, you could change `/etc/mail/mailer.conf` to read:

```
sendmail  /usr/local/supermailer/bin/sendmail-compat
send-mail /usr/local/supermailer/bin/sendmail-compat
mailq  /usr/local/supermailer/bin/mailq-compat
newaliases /usr/local/supermailer/bin/newaliases-compat
hoststat /usr/local/supermailer/bin/hoststat-compat
purgestat /usr/local/supermailer/bin/purgestat-compat
```

26.4.5 Finishing

Once you have everything configured the way you want it, you should either kill the **sendmail** processes that you no longer need and start the processes belonging to your new software, or simply reboot. Rebooting will also give you the opportunity to ensure that you have correctly configured your system to start your new MTA automatically on boot.

26.5 Troubleshooting

1. Why do I have to use the FQDN for hosts on my site?

You will probably find that the host is actually in a different domain; for example, if you are in `foo.bar.edu` and you wish to reach a host called `mumble` in the `bar.edu` domain, you will have to refer to it by the fully-qualified domain name, `mumble.bar.edu`, instead of just `mumble`.

Traditionally, this was allowed by BSD BIND resolvers. However the current version of **BIND** that ships with FreeBSD no longer provides default abbreviations for non-fully qualified domain names other than the domain you are in. So an unqualified host `mumble` must either be found as `mumble.foo.bar.edu`, or it will be searched for in the root domain.

This is different from the previous behavior, where the search continued across `mumble.bar.edu`, and `mumble.edu`. Have a look at RFC 1535 for why this was considered bad practice, or even a security hole.

As a good workaround, you can place the line:

```
search foo.bar.edu bar.edu
```

instead of the previous:

```
domain foo.bar.edu
```

into your `/etc/resolv.conf`. However, make sure that the search order does not go beyond the “boundary between local and public administration”, as RFC 1535 calls it.

2. sendmail says “`mail loops back to myself`”

This is answered in the **sendmail** FAQ as follows:

```
I'm getting these error messages:

553 MX list for domain.net points back to relay.domain.net
554 <user@domain.net>... Local configuration error

How can I solve this problem?

You have asked mail to the domain (e.g., domain.net) to be
forwarded to a specific host (in this case, relay.domain.net)
by using an MX record, but the relay machine does not recognize
itself as domain.net. Add domain.net to /etc/mail/local-host-names
[known as /etc/sendmail.cw prior to version 8.10]
(if you are using FEATURE(use_cw_file)) or add “Cw domain.net”
```

```
to /etc/mail/sendmail.cf.
```

The **sendmail** FAQ can be found at http://www.sendmail.org/faq/ and is recommended reading if you want to do any "tweaking" of your mail setup.

3. How can I run a mail server on a dial-up PPP host?

You want to connect a FreeBSD box on a LAN to the Internet. The FreeBSD box will be a mail gateway for the LAN. The PPP connection is non-dedicated.

There are at least two ways to do this. One way is to use UUCP.

Another way is to get a full-time Internet server to provide secondary MX services for your domain. For example, if your company's domain is `example.com` and your Internet service provider has set `example.net` up to provide secondary MX services to your domain:

```
example.com.          MX        10      example.com.
                      MX        20      example.net.
```

Only one host should be specified as the final recipient (add `Cw example.com` in `/etc/mail/sendmail.cf` on `example.com`).

When the sending `sendmail` is trying to deliver the mail it will try to connect to you (`example.com`) over the modem link. It will most likely time out because you are not online. The program **sendmail** will automatically deliver it to the secondary MX site, i.e. your Internet provider (`example.net`). The secondary MX site will then periodically try to connect to your host and deliver the mail to the primary MX host (`example.com`).

You might want to use something like this as a login script:

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

If you are going to create a separate login script for a user you could use `sendmail -qRexample.com` instead in the script above. This will force all mail in your queue for `example.com` to be processed immediately.

A further refinement of the situation is as follows:

Message stolen from the FreeBSD Internet service provider's 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp).

```
> we provide the secondary MX for a customer. The customer connects to
> our services several times a day automatically to get the mails to
> his primary MX (We do not call his site when a mail for his domains
> arrived). Our sendmail sends the mailqueue every 30 minutes. At the
> moment he has to stay 30 minutes online to be sure that all mail is
> gone to the primary MX.
>
> Is there a command that would initiate sendmail to send all the mails
> now? The user has not root-privileges on our machine of course.

In the "privacy flags" section of sendmail.cf, there is a
definition Opgoaway,restrictqrun
```

```
Remove restrictqrun to allow non-root users to start the queue processing.
You might also like to rearrange the MXs. We are the 1st MX for our
customers like this, and we have defined:

# If we are the best MX for a host, try directly instead of generating
# local config error.
OwTrue

That way a remote site will deliver straight to you, without trying
the customer connection.  You then send to your customer.  Only works for
“hosts”, so you need to get your customer to name their mail
machine “customer.com” as well as
“hostname.customer.com” in the DNS.  Just put an A record in
the DNS for “customer.com”.
```

4. Why do I keep getting “`Relaying Denied`” errors when sending mail from other hosts?

In default FreeBSD installations, **sendmail** is configured to only send mail from the host it is running on. For example, if a POP server is available, then users will be able to check mail from school, work, or other remote locations but they still will not be able to send outgoing emails from outside locations. Typically, a few moments after the attempt, an email will be sent from **MAILER-DAEMON** with a “`5.7 Relaying Denied`” error message.

There are several ways to get around this. The most straightforward solution is to put your ISP’s address in a relay-domains file at `/etc/mail/relay-domains`. A quick way to do this would be:

```
# echo "your.isp.example.com" > /etc/mail/relay-domains
```

After creating or editing this file you must restart **sendmail**. This works great if you are a server administrator and do not wish to send mail locally, or would like to use a point and click client/system on another machine or even another ISP. It is also very useful if you only have one or two email accounts set up. If there is a large number of addresses to add, you can simply open this file in your favorite text editor and then add the domains, one per line:

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

Now any mail sent through your system, by any host in this list (provided the user has an account on your system), will succeed. This is a very nice way to allow users to send mail from your system remotely without allowing people to send SPAM through your system.

26.6 Advanced Topics

The following section covers more involved topics such as mail configuration and setting up mail for your entire domain.

26.6.1 Basic Configuration

Out of the box, you should be able to send email to external hosts as long as you have set up `/etc/resolv.conf` or are running your own name server. If you would like to have mail for your host delivered to the MTA (e.g., **sendmail**) on your own FreeBSD host, there are two methods:

- Run your own name server and have your own domain. For example, `FreeBSD.org`
- Get mail delivered directly to your host. This is done by delivering mail directly to the current DNS name for your machine. For example, `example.FreeBSD.org`.

Regardless of which of the above you choose, in order to have mail delivered directly to your host, it must have a permanent static IP address (not a dynamic address, as with most PPP dial-up configurations). If you are behind a firewall, it must pass SMTP traffic on to you. If you want to receive mail directly at your host, you need to be sure of either of two things:

- Make sure that the (lowest-numbered) MX record in your DNS points to your host's IP address.
- Make sure there is no MX entry in your DNS for your host.

Either of the above will allow you to receive mail directly at your host.

Try this:

```
# hostname
example.FreeBSD.org
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

If that is what you see, mail directly to <`yourlogin@example.FreeBSD.org`> should work without problems (assuming **sendmail** is running correctly on `example.FreeBSD.org`).

If instead you see something like this:

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by hub.FreeBSD.org
```

All mail sent to your host (`example.FreeBSD.org`) will end up being collected on `hub` under the same username instead of being sent directly to your host.

The above information is handled by your DNS server. The DNS record that carries mail routing information is the *M*ail e*X*change entry. If no MX record exists, mail will be delivered directly to the host by way of its IP address.

The MX entry for `freefall.FreeBSD.org` at one time looked like this:

```
freefall  MX 30 mail.crl.net
freefall  MX 40 agora.rdrop.com
freefall  MX 10 freefall.FreeBSD.org
freefall  MX 20 who.cdrom.com
```

As you can see, `freefall` had many MX entries. The lowest MX number is the host that receives mail directly if available; if it is not accessible for some reason, the others (sometimes called "backup MXes") accept messages temporarily, and pass it along when a lower-numbered host becomes available, eventually to the lowest-numbered host.

Alternate MX sites should have separate Internet connections from your own in order to be most useful. Your ISP or another friendly site should have no problem providing this service for you.

26.6.2 Mail for Your Domain

In order to set up a "mailhost" (a.k.a. mail server) you need to have any mail sent to various workstations directed to it. Basically, you want to "claim" any mail for any hostname in your domain (in this case `*.FreeBSD.org`) and divert it to your mail server so your users can receive their mail on the master mail server.

To make life easiest, a user account with the same *username* should exist on both machines. Use adduser(8) to do this.

The mailhost you will be using must be the designated mail exchanger for each workstation on the network. This is done in your DNS configuration like so:

```
example.FreeBSD.org A 204.216.27.XX  ; Workstation
   MX 10 hub.FreeBSD.org ; Mailhost
```

This will redirect mail for the workstation to the mailhost no matter where the A record points. The mail is sent to the MX host.

You cannot do this yourself unless you are running a DNS server. If you are not, or cannot run your own DNS server, talk to your ISP or whoever provides your DNS.

If you are doing virtual email hosting, the following information will come in handy. For this example, we will assume you have a customer with his own domain, in this case `customer1.org`, and you want all the mail for `customer1.org` sent to your mailhost, `mail.myhost.com`. The entry in your DNS should look like this:

```
customer1.org  MX 10 mail.myhost.com
```

You do *not* need an A record for `customer1.org` if you only want to handle email for that domain.

> **Note:** Be aware that pinging `customer1.org` will not work unless an A record exists for it.

The last thing that you must do is tell **sendmail** on your mailhost what domains and/or hostnames it should be accepting mail for. There are a few different ways this can be done. Either of the following will work:

- Add the hosts to your `/etc/mail/local-host-names` file if you are using the `FEATURE(use_cw_file)`. If you are using a version of **sendmail** earlier than 8.10, the file is `/etc/sendmail.cw`.
- Add a `Cwyour.host.com` line to your `/etc/sendmail.cf` or `/etc/mail/sendmail.cf` if you are using **sendmail** 8.10 or higher.

26.7 SMTP with UUCP

The **sendmail** configuration that ships with FreeBSD is designed for sites that connect directly to the Internet. Sites that wish to exchange their mail via UUCP must install another **sendmail** configuration file.

Tweaking /etc/mail/sendmail.cf manually is an advanced topic. **sendmail** version 8 generates config files via m4(1) preprocessing, where the actual configuration occurs on a higher abstraction level. The m4(1) configuration files can be found under /usr/share/sendmail/cf. The file README in the cf directory can serve as a basic introduction to m4(1) configuration.

The best way to support UUCP delivery is to use the mailertable feature. This creates a database that **sendmail** can use to make routing decisions.

First, you have to create your .mc file. The directory /usr/share/sendmail/cf/cf contains a few examples. Assuming you have named your file foo.mc, all you need to do in order to convert it into a valid sendmail.cf is:

```
# cd /etc/mail
# make foo.cf
# cp foo.cf /etc/mail/sendmail.cf
```

A typical .mc file might look like:

```
VERSIONID('Your version number') OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, 'hash -o /etc/mail/mailertable')

define('UUCP_RELAY', your.uucp.relay)
define('UUCP_MAX_SIZE', 200000)
define('confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw    your.alias.host.name
Cw    youruucpnodename.UUCP
```

The lines containing accept_unresolvable_domains, nocanonify, and confDONT_PROBE_INTERFACES features will prevent any usage of the DNS during mail delivery. The UUCP_RELAY clause is needed to support UUCP delivery. Simply put an Internet hostname there that is able to handle .UUCP pseudo-domain addresses; most likely, you will enter the mail relay of your ISP there.

Once you have this, you need an /etc/mail/mailertable file. If you have only one link to the outside that is used for all your mails, the following file will suffice:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
.                           uucp-dom:your.uucp.relay
```

A more complex example might look like this:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
horus.interface-business.de   uucp-dom:horus
.interface-business.de        uucp-dom:if-bus
interface-business.de         uucp-dom:if-bus
```

```
.heep.sax.de                    smtp8:%1
horus.UUCP                      uucp-dom:horus
if-bus.UUCP                     uucp-dom:if-bus
.                               uucp-dom:
```

The first three lines handle special cases where domain-addressed mail should not be sent out to the default route, but instead to some UUCP neighbor in order to “shortcut” the delivery path. The next line handles mail to the local Ethernet domain that can be delivered using SMTP. Finally, the UUCP neighbors are mentioned in the .UUCP pseudo-domain notation, to allow for a *`uucp-neighbor !recipient`* override of the default rules. The last line is always a single dot, matching everything else, with UUCP delivery to a UUCP neighbor that serves as your universal mail gateway to the world. All of the node names behind the `uucp-dom:` keyword must be valid UUCP neighbors, as you can verify using the command `uuname`.

As a reminder that this file needs to be converted into a DBM database file before use. The command line to accomplish this is best placed as a comment at the top of the `mailertable` file. You always have to execute this command each time you change your `mailertable` file.

Final hint: if you are uncertain whether some particular mail routing would work, remember the `-bt` option to **sendmail**. It starts **sendmail** in *address test mode*; simply enter `3,0`, followed by the address you wish to test for the mail routing. The last line tells you the used internal mail agent, the destination host this agent will be called with, and the (possibly translated) address. Leave this mode by typing **Ctrl**+**D**.

```
% sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@example.com
canonify           input: foo @ example . com
...
parse            returns: $# uucp-dom $@ your.uucp.relay $: foo < @ example . com . >
> ^D
```

26.8 Setting Up to Send Only

Contributed by Bill Moran.

There are many instances where you may only want to send mail through a relay. Some examples are:

- Your computer is a desktop machine, but you want to use programs such as send-pr(1). To do so, you should use your ISP’s mail relay.
- The computer is a server that does not handle mail locally, but needs to pass off all mail to a relay for processing.

Just about any MTA is capable of filling this particular niche. Unfortunately, it can be very difficult to properly configure a full-featured MTA just to handle offloading mail. Programs such as **sendmail** and **postfix** are largely overkill for this use.

Additionally, if you are using a typical Internet access service, your agreement may forbid you from running a “mail server”.

The easiest way to fulfill those needs is to install the `mail/ssmtp` port. Execute the following commands as `root`:

```
# cd /usr/ports/mail/ssmtp
```

```
# make install replace clean
```

Once installed, `mail/ssmtp` can be configured with a four-line file located at `/usr/local/etc/ssmtp/ssmtp.conf`:

```
root=yourrealemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Make sure you use your real email address for `root`. Enter your ISP's outgoing mail relay in place of `mail.example.com` (some ISPs call this the "outgoing mail server" or "SMTP server").

Make sure you disable **sendmail**, including the outgoing mail service. See Section 26.4.2 for details.

`mail/ssmtp` has some other options available. See the example configuration file in `/usr/local/etc/ssmtp` or the manual page of **ssmtp** for some examples and more information.

Setting up **ssmtp** in this manner will allow any software on your computer that needs to send mail to function properly, while not violating your ISP's usage policy or allowing your computer to be hijacked for spamming.

26.9 Using Mail with a Dialup Connection

If you have a static IP address, you should not need to adjust anything from the defaults. Set your host name to your assigned Internet name and **sendmail** will do the rest.

If you have a dynamically assigned IP number and use a dialup PPP connection to the Internet, you will probably have a mailbox on your ISPs mail server. Let's assume your ISP's domain is `example.net`, and that your user name is `user`, you have called your machine `bsd.home`, and your ISP has told you that you may use `relay.example.net` as a mail relay.

In order to retrieve mail from your mailbox, you must install a retrieval agent. The **fetchmail** utility is a good choice as it supports many different protocols. This program is available as a package or from the Ports Collection (`mail/fetchmail`). Usually, your ISP will provide POP. If you are using user PPP, you can automatically fetch your mail when an Internet connection is established with the following entry in `/etc/ppp/ppp.linkup`:

```
MYADDR:
!bg su user -c fetchmail
```

If you are using **sendmail** (as shown below) to deliver mail to non-local accounts, you probably want to have **sendmail** process your mailqueue as soon as your Internet connection is established. To do this, put this command after the `fetchmail` command in `/etc/ppp/ppp.linkup`:

```
  !bg su user -c "sendmail -q"
```

Assume that you have an account for `user` on `bsd.home`. In the home directory of `user` on `bsd.home`, create a `.fetchmailrc` file:

```
poll example.net protocol pop3 fetchall pass MySecret
```

This file should not be readable by anyone except `user` as it contains the password `MySecret`.

In order to send mail with the correct `from:` header, you must tell **sendmail** to use <`user@example.net`> rather than <`user@bsd.home`>. You may also wish to tell **sendmail** to send all mail via `relay.example.net`, allowing quicker mail transmission.

The following `.mc` file should suffice:

```
VERSIONID('bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS('example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define('SMART_HOST', 'relay.example.net')
Dmbsd.home
define('confDOMAIN_NAME','bsd.home')dnl
define('confDELIVERY_MODE','deferred')dnl
```

Refer to the previous section for details of how to turn this `.mc` file into a `sendmail.cf` file. Also, do not forget to restart **sendmail** after updating `sendmail.cf`.

26.10 SMTP Authentication

Written by James Gorham.

Having SMTP Authentication in place on your mail server has a number of benefits. SMTP Authentication can add another layer of security to **sendmail**, and has the benefit of giving mobile users who switch hosts the ability to use the same mail server without the need to reconfigure their mail client settings each time.

1. Install `security/cyrus-sasl2` from the ports. You can find this port in `security/cyrus-sasl2`. The `security/cyrus-sasl2` port supports a number of compile-time options. For the SMTP Authentication method we will be using here, make sure that the `LOGIN` option is not disabled.
2. After installing `security/cyrus-sasl2`, edit `/usr/local/lib/sasl2/Sendmail.conf` (or create it if it does not exist) and add the following line:

   ```
   pwcheck_method: saslauthd
   ```
3. Next, install `security/cyrus-sasl2-saslauthd`, edit `/etc/rc.conf` to add the following line:

   ```
   saslauthd_enable="YES"
   ```

   and finally start the saslauthd daemon:

   ```
   # /usr/local/etc/rc.d/saslauthd start
   ```

   This daemon serves as a broker for **sendmail** to authenticate against your FreeBSD `passwd` database. This saves the trouble of creating a new set of usernames and passwords for each user that needs to use SMTP authentication, and keeps the login and mail password the same.

4. Now edit `/etc/make.conf` and add the following lines:

   ```
   SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
   SENDMAIL_LDFLAGS=-L/usr/local/lib
   SENDMAIL_LDADD=-lsasl2
   ```

   These lines will give **sendmail** the proper configuration options for linking to `cyrus-sasl2` at compile time. Make sure that `cyrus-sasl2` has been installed before recompiling **sendmail**.

5. Recompile **sendmail** by executing the following commands:

   ```
   # cd /usr/src/lib/libsmutil
   # make cleandir && make obj && make
   # cd /usr/src/lib/libsm
   # make cleandir && make obj && make
   # cd /usr/src/usr.sbin/sendmail
   # make cleandir && make obj && make && make install
   ```

   The compile of **sendmail** should not have any problems if `/usr/src` has not been changed extensively and the shared libraries it needs are available.

6. After **sendmail** has been compiled and reinstalled, edit your `/etc/mail/freebsd.mc` file (or whichever file you use as your `.mc` file. Many administrators choose to use the output from hostname(1) as the `.mc` file for uniqueness). Add these lines to it:

   ```
   dnl set SASL options
   TRUST_AUTH_MECH(‘GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN’)dnl
   define(‘confAUTH_MECHANISMS’, ‘GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN’)dnl
   ```

   These options configure the different methods available to **sendmail** for authenticating users. If you would like to use a method other than **pwcheck**, please see the included documentation.

7. Finally, run make(1) while in `/etc/mail`. That will run your new `.mc` file and create a `.cf` file named `freebsd.cf` (or whatever name you have used for your `.mc` file). Then use the command `make install restart`, which will copy the file to `sendmail.cf`, and will properly restart **sendmail**. For more information about this process, you should refer to `/etc/mail/Makefile`.

If all has gone correctly, you should be able to enter your login information into the mail client and send a test message. For further investigation, set the `LogLevel` of **sendmail** to 13 and watch `/var/log/maillog` for any errors.

For more information, please see the **sendmail** page regarding SMTP authentication (http://www.sendmail.org/~ca/email/auth.html).

26.11 Mail User Agents

Contributed by Marc Silver.

A Mail User Agent (MUA) is an application that is used to send and receive email. Furthermore, as email “evolves” and becomes more complex, MUA’s are becoming increasingly powerful in the way they interact with email; this gives users increased functionality and flexibility. FreeBSD contains support for numerous mail user agents, all of which can be easily installed using the FreeBSD Ports Collection. Users may choose between graphical email clients such as **evolution** or **balsa**, console based clients such as **mutt**, **pine** or `mail`, or the web interfaces used by some large organizations.

26.11.1 mail

mail(1) is the default Mail User Agent (MUA) in FreeBSD. It is a console based MUA that offers all the basic functionality required to send and receive text-based email, though it is limited in interaction abilities with attachments and can only support local mailboxes.

Although `mail` does not natively support interaction with POP or IMAP servers, these mailboxes may be downloaded to a local `mbox` file using an application such as **fetchmail**, which will be discussed later in this chapter (Section 26.12).

In order to send and receive email, simply invoke the `mail` command as per the following example:

```
% mail
```

The contents of the user mailbox in `/var/mail` are automatically read by the `mail` utility. Should the mailbox be empty, the utility exits with a message indicating that no mails could be found. Once the mailbox has been read, the application interface is started, and a list of messages will be displayed. Messages are automatically numbered, as can be seen in the following example:

```
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/marcs": 3 messages 3 new
>N  1 root@localhost        Mon Mar  8 14:05  14/510   "test"
 N  2 root@localhost        Mon Mar  8 14:05  14/509   "user account"
 N  3 root@localhost        Mon Mar  8 14:05  14/509   "sample"
```

Messages can now be read by using the **t** `mail` command, suffixed by the message number that should be displayed. In this example, we will read the first email:

```
& t 1
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Mon,  8 Mar 2004 14:05:52 +0200 (SAST)
From: root@localhost (Charlie Root)

This is a test message, please reply if you receive it.
```

As can be seen in the example above, the **t** key will cause the message to be displayed with full headers. To display the list of messages again, the **h** key should be used.

If the email requires a response, you may use `mail` to reply, by using either the **R** or **r** `mail` keys. The **R** key instructs `mail` to reply only to the sender of the email, while **r** replies not only to the sender, but also to other recipients of the message. You may also suffix these commands with the mail number which you would like make a reply to. Once this has been done, the response should be entered, and the end of the message should be marked by a single **.** on a new line. An example can be seen below:

```
& R 1
To: root@localhost
Subject: Re: test

Thank you, I did get your email.
```

```
.
EOT
```

In order to send new email, the **m** key should be used, followed by the recipient email address. Multiple recipients may also be specified by separating each address with the **,** delimiter. The subject of the message may then be entered, followed by the message contents. The end of the message should be specified by putting a single **.** on a new line.

```
& mail root@localhost
Subject: I mastered mail

Now I can send and receive email using mail ... :)
.
EOT
```

While inside the `mail` utility, the **?** command may be used to display help at any time, the mail(1) manual page should also be consulted for more help with `mail`.

> **Note:** As previously mentioned, the mail(1) command was not originally designed to handle attachments, and thus deals with them very poorly. Newer MUAs such as **mutt** handle attachments in a much more intelligent way. But should you still wish to use the `mail` command, the `converters/mpack` port may be of considerable use.

26.11.2 mutt

mutt is a small yet very powerful Mail User Agent, with excellent features, just some of which include:

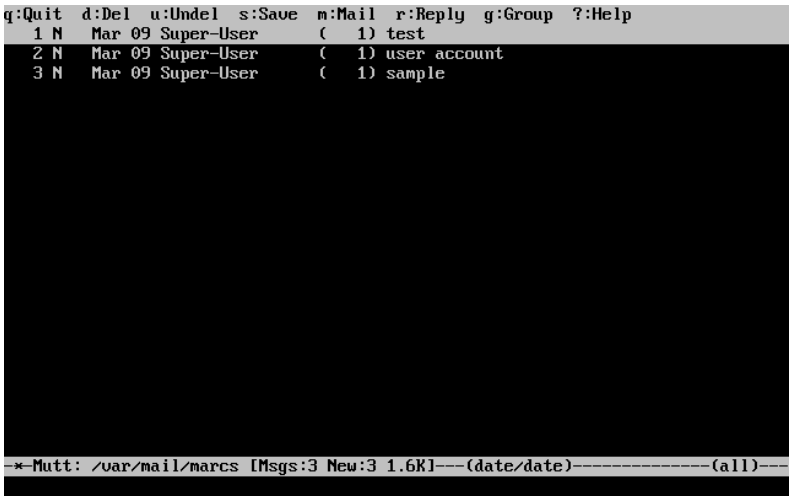
- The ability to thread messages;
- PGP support for digital signing and encryption of email;
- MIME Support;
- Maildir Support;
- Highly customizable.

All of these features help to make **mutt** one of the most advanced mail user agents available. See http://www.mutt.org for more information on **mutt**.

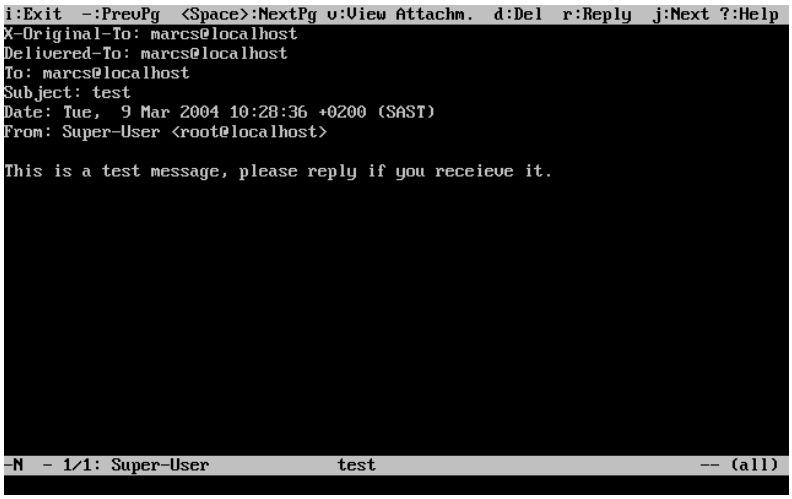
The stable version of **mutt** may be installed using the `mail/mutt` port, while the current development version may be installed via the `mail/mutt-devel` port. After the port has been installed, **mutt** can be started by issuing the following command:

```
% mutt
```

mutt will automatically read the contents of the user mailbox in `/var/mail` and display the contents if applicable. If no mails are found in the user mailbox, then **mutt** will wait for commands from the user. The example below shows **mutt** displaying a list of messages:



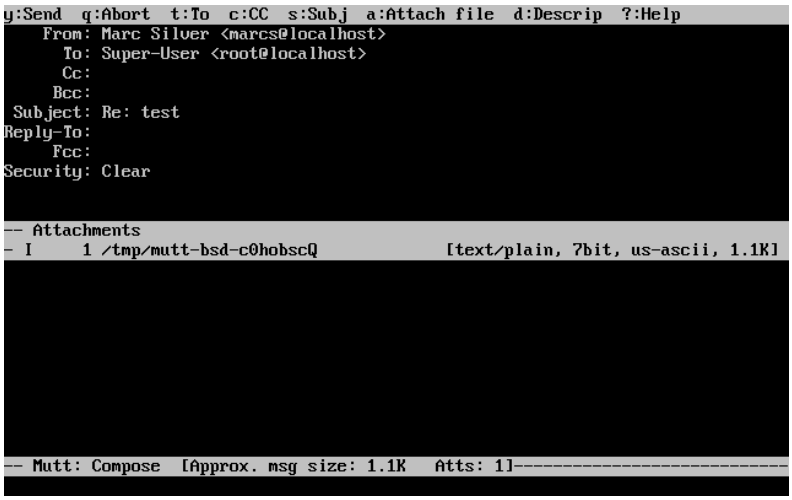
In order to read an email, simply select it using the cursor keys, and press the **Enter** key. An example of **mutt** displaying email can be seen below:



As with the mail(1) command, **mutt** allows users to reply only to the sender of the message as well as to all recipients. To reply only to the sender of the email, use the **r** keyboard shortcut. To send a group reply, which will be sent to the original sender as well as all the message recipients, use the **g** shortcut.

> **Note: mutt** makes use of the vi(1) command as an editor for creating and replying to emails. This may be customized by the user by creating or editing their own `.muttrc` file in their home directory and setting the `editor` variable or by setting the `EDITOR` environment variable. See http://www.mutt.org/ for more information about configuring **mutt**.

In order to compose a new mail message, press **m**. After a valid subject has been given, **mutt** will start vi(1) and the mail can be written. Once the contents of the mail are complete, save and quit from `vi` and **mutt** will resume, displaying a summary screen of the mail that is to be delivered. In order to send the mail, press **y**. An example of the summary screen can be seen below:



mutt also contains extensive help, which can be accessed from most of the menus by pressing the **?** key. The top line also displays the keyboard shortcuts where appropriate.

26.11.3 pine

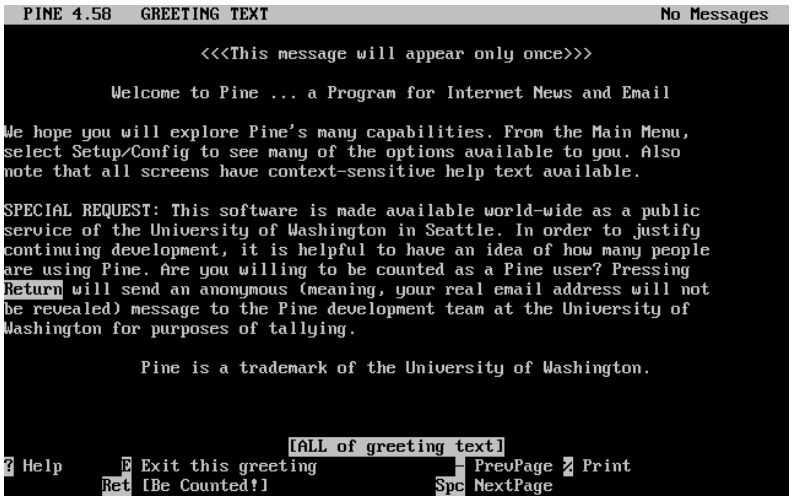
pine is aimed at a beginner user, but also includes some advanced features.

> **Warning:** The **pine** software has had several remote vulnerabilities discovered in the past, which allowed remote attackers to execute arbitrary code as users on the local system, by the action of sending a specially-prepared email. All such *known* problems have been fixed, but the **pine** code is written in a very insecure style and the FreeBSD Security Officer believes there are likely to be other undiscovered vulnerabilities. You install **pine** at your own risk.

The current version of **pine** may be installed using the `mail/pine4` port. Once the port has installed, **pine** can be started by issuing the following command:

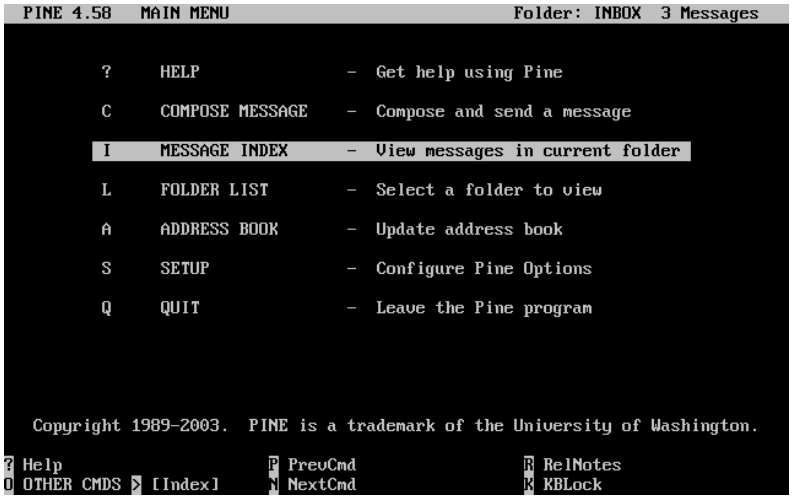
```
% pine
```

The first time that **pine** is run it displays a greeting page with a brief introduction, as well as a request from the **pine** development team to send an anonymous email message allowing them to judge how many users are using their client. To send this anonymous message, press **Enter**, or alternatively press **E** to exit the greeting without sending an anonymous message. An example of the greeting page can be seen below:

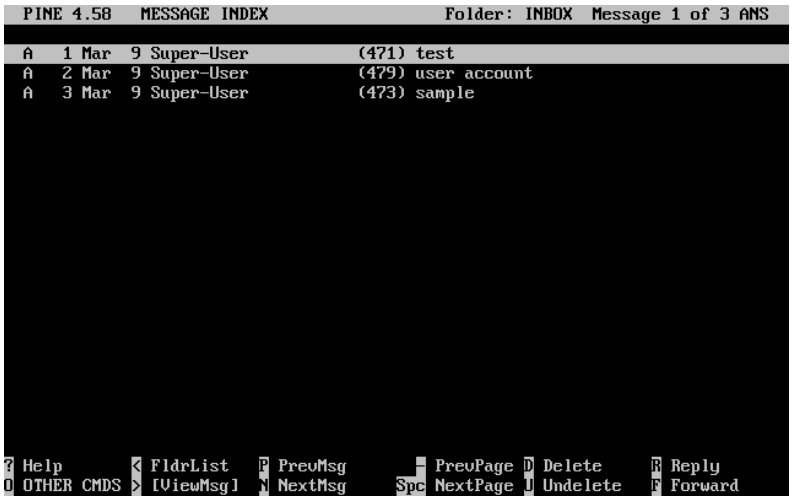


Users are then presented with the main menu, which can be easily navigated using the cursor keys. This main menu provides shortcuts for the composing new mails, browsing of mail directories, and even the administration of address book entries. Below the main menu, relevant keyboard shortcuts to perform functions specific to the task at hand are shown.

The default directory opened by **pine** is the `inbox`. To view the message index, press **I**, or select the MESSAGE INDEX option as seen below:

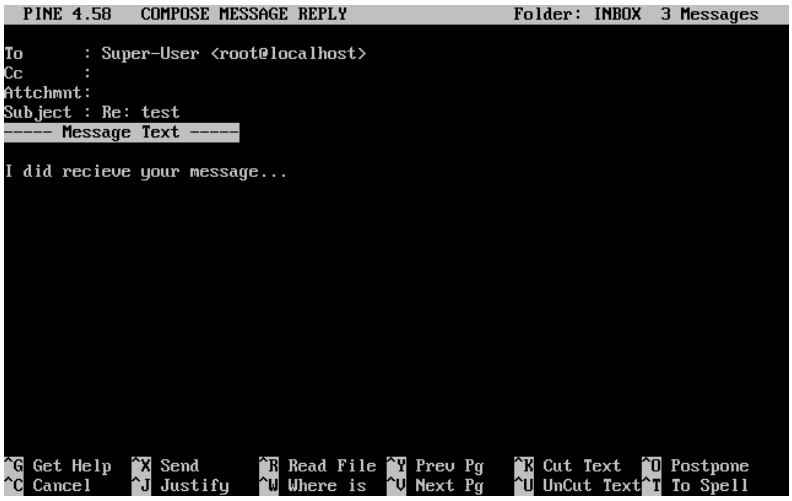


The message index shows messages in the current directory, and can be navigated by using the cursor keys. Highlighted messages can be read by pressing the **Enter** key.



In the screenshot below, a sample message is displayed by **pine**. Keyboard shortcuts are displayed as a reference at the bottom of the screen. An example of one of these shortcuts is the **r** key, which tells the MUA to reply to the current message being displayed.

Replying to an email in **pine** is done using the **pico** editor, which is installed by default with **pine**. The **pico** utility makes it easy to navigate around the message and is slightly more forgiving on novice users than vi(1) or mail(1). Once the reply is complete, the message can be sent by pressing **Ctrl**+**X**. The **pine** application will ask for confirmation.



The **pine** application can be customized using the SETUP option from the main menu. Consult http://www.washington.edu/pine/ for more information.

26.12 Using fetchmail

Contributed by Marc Silver.

fetchmail is a full-featured IMAP and POP client which allows users to automatically download mail from remote IMAP and POP servers and save it into local mailboxes; there it can be accessed more easily. **fetchmail** can be installed using the `mail/fetchmail` port, and offers various features, some of which include:

- Support of POP3, APOP, KPOP, IMAP, ETRN and ODMR protocols.
- Ability to forward mail using SMTP, which allows filtering, forwarding, and aliasing to function normally.
- May be run in daemon mode to check periodically for new messages.
- Can retrieve multiple mailboxes and forward them based on configuration, to different local users.

While it is outside the scope of this document to explain all of **fetchmail**'s features, some basic features will be explained. The **fetchmail** utility requires a configuration file known as `.fetchmailrc`, in order to run correctly. This file includes server information as well as login credentials. Due to the sensitive nature of the contents of this file, it is advisable to make it readable only by the owner, with the following command:

```
% chmod 600 .fetchmailrc
```

The following `.fetchmailrc` serves as an example for downloading a single user mailbox using POP. It tells **fetchmail** to connect to `example.com` using a username of `joesoap` and a password of `XXX`. This example assumes that the user `joesoap` is also a user on the local system.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

The next example connects to multiple POP and IMAP servers and redirects to different local usernames where applicable:

```
poll example.com proto pop3:
user "joesoap", with password "XXX", is "jsoap" here;
user "andrea", with password "XXXX";
poll example2.net proto imap:
user "john", with password "XXXXX", is "myth" here;
```

The **fetchmail** utility can be run in daemon mode by running it with the `-d` flag, followed by the interval (in seconds) that **fetchmail** should poll servers listed in the `.fetchmailrc` file. The following example would cause **fetchmail** to poll every 600 seconds:

```
% fetchmail -d 600
```

More information on **fetchmail** can be found at http://fetchmail.berlios.de/.

26.13 Using procmail

Contributed by Marc Silver.

The **procmail** utility is an incredibly powerful application used to filter incoming mail. It allows users to define "rules" which can be matched to incoming mails to perform specific functions or to reroute mail to alternative mailboxes and/or email addresses. **procmail** can be installed using the `mail/procmail` port. Once installed, it can be directly integrated into most MTAs; consult your MTA documentation for more information. Alternatively, **procmail** can be integrated by adding the following line to a `.forward` in the home directory of the user utilizing **procmail** features:

```
"|exec /usr/local/bin/procmail || exit 75"
```

The following section will display some basic **procmail** rules, as well as brief descriptions on what they do. These rules, and others must be inserted into a `.procmailrc` file, which must reside in the user's home directory.

The majority of these rules can also be found in the procmailex(5) manual page.

Forward all mail from <`user@example.com`> to an external address of <`goodmail@example2.com`>:

```
:0
* ^From.*user@example.com
! goodmail@example2.com
```

Forward all mails shorter than 1000 bytes to an external address of <`goodmail@example2.com`>:

```
:0
* < 1000
! goodmail@example2.com
```

Send all mail sent to <`alternate@example.com`> into a mailbox called `alternate`:

```
:0
* ^TOalternate@example.com
alternate
```

Send all mail with a subject of "Spam" to `/dev/null`:

```
:0
```

```
^Subject:.*Spam
/dev/null
```

A useful recipe that parses incoming `FreeBSD.org` mailing lists and places each list in its own mailbox:

```
:0
* ^Sender:.owner-freebsd-\/[^@]+@FreeBSD.ORG
{
 LISTNAME=${MATCH}
 :0
 * LISTNAME??^\/[^@]+
 FreeBSD-${MATCH}
}
```

Chapter 27 Network Servers

Reorganized by Murray Stokely.

27.1 概述

This chapter will cover some of the more frequently used network services on UNIX systems. We will cover how to install, configure, test, and maintain many different types of network services. Example configuration files are included throughout this chapter for you to benefit from.

After reading this chapter, you will know:

- How to manage the **inetd** daemon.
- How to set up a network file system.
- How to set up a network information server for sharing user accounts.
- How to set up automatic network settings using DHCP.
- How to set up a domain name server.
- How to set up the **Apache** HTTP Server.
- How to set up a File Transfer Protocol (FTP) Server.
- How to set up a file and print server for Windows clients using **Samba**.
- How to synchronize the time and date, and set up a time server, with the NTP protocol.

Before reading this chapter, you should:

- Understand the basics of the `/etc/rc` scripts.
- Be familiar with basic network terminology.
- Know how to install additional third-party software (Chapter 4).

27.2 The inetd “Super-Server”

Contributed by Chern Lee.

27.2.1 Overview

inetd(8) is referred to as the “Internet Super-Server” because it manages connections for several services. When a connection is received by **inetd**, it determines which program the connection is destined for, spawns the particular process and delegates the socket to it (the program is invoked with the service socket as its standard input, output and error descriptors). Running one instance of **inetd** reduces the overall system load as compared to running each daemon individually in stand-alone mode.

Primarily, **inetd** is used to spawn other daemons, but several trivial protocols are handled directly, such as **chargen**, **auth**, and **daytime**.

This section will cover the basics in configuring **inetd** through its command-line options and its configuration file, `/etc/inetd.conf`.

27.2.2 Settings

inetd is initialized through the `/etc/rc.conf` system. The `inetd_enable` option is set to `NO` by default, but is often times turned on by **sysinstall** with the medium security profile. Placing:

```
inetd_enable="YES"
```

or

```
inetd_enable="NO"
```

into `/etc/rc.conf` can enable or disable **inetd** starting at boot time.

Additionally, different command-line options can be passed to **inetd** via the `inetd_flags` option.

27.2.3 Command-Line Options

inetd synopsis:

```
inetd [-d] [-l] [-w] [-W] [-c maximum] [-C rate] [-a address | hostname] [-p filename]
[-R rate] [configuration file]
```

-d

Turn on debugging.

-l

Turn on logging of successful connections.

-w

Turn on TCP Wrapping for external services (on by default).

-W

Turn on TCP Wrapping for internal services which are built into **inetd** (on by default).

-c maximum

Specify the default maximum number of simultaneous invocations of each service; the default is unlimited. May be overridden on a per-service basis with the `max-child` parameter.

-C rate

Specify the default maximum number of times a service can be invoked from a single IP address in one minute; the default is unlimited. May be overridden on a per-service basis with the `max-connections-per-ip-per-minute` parameter.

-R rate

Specify the maximum number of times a service can be invoked in one minute; the default is 256. A rate of 0 allows an unlimited number of invocations.

-a

Specify one specific IP address to bind to. Alternatively, a hostname can be specified, in which case the IPv4 or IPv6 address which corresponds to that hostname is used. Usually a hostname is specified when **inetd** is run inside a jail(8), in which case the hostname corresponds to the jail(8) environment.

When hostname specification is used and both IPv4 and IPv6 bindings are desired, one entry with the appropriate protocol type for each binding is required for each service in `/etc/inetd.conf`. For example, a TCP-based service would need two entries, one using `tcp4` for the protocol and the other using `tcp6`.

-p

Specify an alternate file in which to store the process ID.

These options can be passed to **inetd** using the `inetd_flags` option in `/etc/rc.conf`. By default, `inetd_flags` is set to `-wW`, which turns on TCP wrapping for **inetd**'s internal and external services. For novice users, these parameters usually do not need to be modified or even entered in `/etc/rc.conf`.

Note: An external service is a daemon outside of **inetd**, which is invoked when a connection is received for it. On the other hand, an internal service is one that **inetd** has the facility of offering within itself.

27.2.4 `inetd.conf`

Configuration of **inetd** is controlled through the `/etc/inetd.conf` file.

When a modification is made to `/etc/inetd.conf`, **inetd** can be forced to re-read its configuration file by sending a HangUP signal to the **inetd** process as shown:

Example 27-1. Sending inetd a HangUP Signal

```
# kill -HUP ‘cat /var/run/inetd.pid‘
```

Each line of the configuration file specifies an individual daemon. Comments in the file are preceded by a "#". The format of `/etc/inetd.conf` is as follows:

```
service-name
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute]]
user[:group][/login-class]
server-program
server-program-arguments
```

An example entry for the **ftpd** daemon using IPv4:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

service-name

This is the service name of the particular daemon. It must correspond to a service listed in `/etc/services`. This determines which port **inetd** must listen to. If a new service is being created, it must be placed in `/etc/services` first.

socket-type

Either `stream`, `dgram`, `raw`, or `seqpacket`. `stream` must be used for connection-based, TCP daemons, while `dgram` is used for daemons utilizing the UDP transport protocol.

protocol

One of the following:

Protocol	Explanation
tcp, tcp4	TCP IPv4
udp, udp4	UDP IPv4
tcp6	TCP IPv6
udp6	UDP IPv6
tcp46	Both TCP IPv4 and v6
udp46	Both UDP IPv4 and v6

{wait|nowait}[/max-child[/max-connections-per-ip-per-minute]]

`wait|nowait` indicates whether the daemon invoked from **inetd** is able to handle its own socket or not. `dgram` socket types must use the `wait` option, while stream socket daemons, which are usually multi-threaded, should use `nowait`. `wait` usually hands off multiple sockets to a single daemon, while `nowait` spawns a child daemon for each new socket.

The maximum number of child daemons **inetd** may spawn can be set using the `max-child` option. If a limit of ten instances of a particular daemon is needed, a `/10` would be placed after `nowait`.

In addition to `max-child`, another option limiting the maximum connections from a single place to a particular daemon can be enabled. `max-connections-per-ip-per-minute` does just this. A value of ten here would limit any particular IP address connecting to a particular service to ten attempts per minute. This is useful to prevent intentional or unintentional resource consumption and Denial of Service (DoS) attacks to a machine.

In this field, `wait` or `nowait` is mandatory. `max-child` and `max-connections-per-ip-per-minute` are optional.

A stream-type multi-threaded daemon without any `max-child` or `max-connections-per-ip-per-minute` limits would simply be: `nowait`.

The same daemon with a maximum limit of ten daemons would read: `nowait/10`.

Additionally, the same setup with a limit of twenty connections per IP address per minute and a maximum total limit of ten child daemons would read: `nowait/10/20`.

These options are all utilized by the default settings of the **fingerd** daemon, as seen here:

```
finger stream  tcp     nowait/3/10 nobody /usr/libexec/fingerd fingerd -s
```

user

This is the username that the particular daemon should run as. Most commonly, daemons run as the `root` user. For security purposes, it is common to find some servers running as the `daemon` user, or the least privileged `nobody` user.

server-program

The full path of the daemon to be executed when a connection is received. If the daemon is a service provided by **inetd** internally, then `internal` should be used.

server-program-arguments

This works in conjunction with `server-program` by specifying the arguments, starting with `argv[0]`, passed to the daemon on invocation. If `mydaemon -d` is the command line, `mydaemon -d` would be the value of `server-program-arguments`. Again, if the daemon is an internal service, use `internal` here.

27.2.5 Security

Depending on the security profile chosen at install, many of **inetd**'s daemons may be enabled by default. If there is no apparent need for a particular daemon, disable it! Place a "#" in front of the daemon in question in `/etc/inetd.conf`, and then send a hangup signal to inetd. Some daemons, such as **fingerd**, may not be desired at all because they provide an attacker with too much information.

Some daemons are not security-conscious and have long, or non-existent timeouts for connection attempts. This allows an attacker to slowly send connections to a particular daemon, thus saturating available resources. It may be a good idea to place `max-connections-per-ip-per-minute` and `max-child` limitations on certain daemons.

By default, TCP wrapping is turned on. Consult the hosts_access(5) manual page for more information on placing TCP restrictions on various **inetd** invoked daemons.

27.2.6 Miscellaneous

daytime, **time**, **echo**, **discard**, **chargen**, and **auth** are all internally provided services of **inetd**.

The **auth** service provides identity (**ident**, **identd**) network services, and is configurable to a certain degree.

Consult the inetd(8) manual page for more in-depth information.

27.3 Network File System (NFS)

Reorganized and enhanced by Tom Rhodes. Written by Bill Swingle.

Among the many different file systems that FreeBSD supports is the Network File System, also known as NFS. NFS allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems almost as if they were local files.

Some of the most notable benefits that NFS can provide are:

- Local workstations use less disk space because commonly used data can be stored on a single machine and still remain accessible to others over the network.
- There is no need for users to have separate home directories on every network machine. Home directories could be set up on the NFS server and made available throughout the network.
- Storage devices such as floppy disks, CDROM drives, and Zip® drives can be used by other machines on the network. This may reduce the number of removable media drives throughout the network.

27.3.1 How NFS Works

NFS consists of at least two main parts: a server and one or more clients. The client remotely accesses the data that is stored on the server machine. In order for this to function properly a few processes have to be configured and running.

> **Note:** Under FreeBSD 4.X, the **portmap** utility is used in place of the **rpcbind** utility. Thus, in FreeBSD 4.X the user is required to replace every instance of **rpcbind** with **portmap** in the forthcoming examples.

The server has to be running the following daemons:

Daemon	Description
nfsd	The NFS daemon which services requests from the NFS clients.
mountd	The NFS mount daemon which carries out the requests that nfsd(8) passes on to it.
rpcbind	This daemon allows NFS clients to discover which port the NFS server is using.

The client can also run a daemon, known as **nfsiod**. The **nfsiod** daemon services the requests from the NFS server. This is optional, and improves performance, but is not required for normal and correct operation. See the nfsiod(8) manual page for more information.

27.3.2 Configuring NFS

NFS configuration is a relatively straightforward process. The processes that need to be running can all start at boot time with a few modifications to your `/etc/rc.conf` file.

On the NFS server, make sure that the following options are configured in the `/etc/rc.conf` file:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_flags="-r"
```

mountd runs automatically whenever the NFS server is enabled.

On the client, make sure this option is present in `/etc/rc.conf`:

```
nfs_client_enable="YES"
```

The `/etc/exports` file specifies which file systems NFS should export (sometimes referred to as “share”). Each line in `/etc/exports` specifies a file system to be exported and which machines have access to that file system. Along with what machines have access to that file system, access options may also be specified. There are many such

options that can be used in this file but only a few will be mentioned here. You can easily discover other options by reading over the exports(5) manual page.

Here are a few example `/etc/exports` entries:

The following examples give an idea of how to export file systems, although the settings may be different depending on your environment and network configuration. For instance, to export the `/cdrom` directory to three example machines that have the same domain name as the server (hence the lack of a domain name for each) or have entries in your `/etc/hosts` file. The `-ro` flag makes the exported file system read-only. With this flag, the remote system will not be able to write any changes to the exported file system.

```
/cdrom -ro host1 host2 host3
```

The following line exports `/home` to three hosts by IP address. This is a useful setup if you have a private network without a DNS server configured. Optionally the `/etc/hosts` file could be configured for internal hostnames; please review hosts(5) for more information. The `-alldirs` flag allows the subdirectories to be mount points. In other words, it will not mount the subdirectories but permit the client to mount only the directories that are required or needed.

```
/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

The following line exports `/a` so that two clients from different domains may access the file system. The `-maproot=root` flag allows the `root` user on the remote system to write data on the exported file system as `root`. If the `-maproot=root` flag is not specified, then even if a user has `root` access on the remote system, he will not be able to modify files on the exported file system.

```
/a  -maproot=root  host.example.com box.example.org
```

In order for a client to access an exported file system, the client must have permission to do so. Make sure the client is listed in your `/etc/exports` file.

In `/etc/exports`, each line represents the export information for one file system to one host. A remote host can only be specified once per file system, and may only have one default entry. For example, assume that `/usr` is a single file system. The following `/etc/exports` would be invalid:

```
# Invalid when /usr is one file system
/usr/src   client
/usr/ports client
```

One file system, `/usr`, has two lines specifying exports to the same host, `client`. The correct format for this situation is:

```
/usr/src /usr/ports  client
```

The properties of one file system exported to a given host must all occur on one line. Lines without a client specified are treated as a single host. This limits how you can export file systems, but for most people this is not an issue.

The following is an example of a valid export list, where `/usr` and `/exports` are local file systems:

```
# Export src and ports to client01 and client02, but only
# client01 has root privileges on it
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports               client02
# The client machines have root and can mount anywhere
```

```
# on /exports. Anyone in the world can mount /exports/obj read-only
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

You must restart **mountd** whenever you modify `/etc/exports` so the changes can take effect. This can be accomplished by sending the HUP signal to the `mountd` process:

```
# kill -HUP ‘cat /var/run/mountd.pid‘
```

Alternatively, a reboot will make FreeBSD set everything up properly. A reboot is not necessary though. Executing the following commands as `root` should start everything up.

On the NFS server:

```
# rpcbind
# nfsd -u -t -n 4
# mountd -r
```

On the NFS client:

```
# nfsiod -n 4
```

Now everything should be ready to actually mount a remote file system. In these examples the server’s name will be `server` and the client’s name will be `client`. If you only want to temporarily mount a remote file system or would rather test the configuration, just execute a command like this as `root` on the client:

```
# mount server:/home /mnt
```

This will mount the `/home` directory on the server at `/mnt` on the client. If everything is set up correctly you should be able to enter `/mnt` on the client and see all the files that are on the server.

If you want to automatically mount a remote file system each time the computer boots, add the file system to the `/etc/fstab` file. Here is an example:

```
server:/home /mnt nfs rw 0 0
```

The fstab(5) manual page lists all the available options.

27.3.3 Practical Uses

NFS has many practical uses. Some of the more common ones are listed below:

- Set several machines to share a CDROM or other media among them. This is cheaper and often a more convenient method to install software on multiple machines.
- On large networks, it might be more convenient to configure a central NFS server in which to store all the user home directories. These home directories can then be exported to the network so that users would always have the same home directory, regardless of which workstation they log in to.
- Several machines could have a common `/usr/ports/distfiles` directory. That way, when you need to install a port on several machines, you can quickly access the source without downloading it on each machine.

27.3.4 Automatic Mounts with amd

Contributed by Wylie Stilwell. Rewritten by Chern Lee.

amd(8) (the automatic mounter daemon) automatically mounts a remote file system whenever a file or directory within that file system is accessed. Filesystems that are inactive for a period of time will also be automatically unmounted by **amd**. Using **amd** provides a simple alternative to permanent mounts, as permanent mounts are usually listed in `/etc/fstab`.

amd operates by attaching itself as an NFS server to the `/host` and `/net` directories. When a file is accessed within one of these directories, **amd** looks up the corresponding remote mount and automatically mounts it. `/net` is used to mount an exported file system from an IP address, while `/host` is used to mount an export from a remote hostname.

An access to a file within `/host/foobar/usr` would tell **amd** to attempt to mount the `/usr` export on the host `foobar`.

Example 27-2. Mounting an Export with amd

You can view the available mounts of a remote host with the `showmount` command. For example, to view the mounts of a host named `foobar`, you can use:

```
% showmount -e foobar
Exports list on foobar:
/usr                               10.10.10.0
/a                                 10.10.10.0
% cd /host/foobar/usr
```

As seen in the example, the `showmount` shows `/usr` as an export. When changing directories to `/host/foobar/usr`, **amd** attempts to resolve the hostname `foobar` and automatically mount the desired export.

amd can be started by the startup scripts by placing the following lines in `/etc/rc.conf`:

```
amd_enable="YES"
```

Additionally, custom flags can be passed to **amd** from the `amd_flags` option. By default, `amd_flags` is set to:

```
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map"
```

The `/etc/amd.map` file defines the default options that exports are mounted with. The `/etc/amd.conf` file defines some of the more advanced features of **amd**.

Consult the amd(8) and amd.conf(5) manual pages for more information.

27.3.5 Problems Integrating with Other Systems

Contributed by John Lind.

Certain Ethernet adapters for ISA PC systems have limitations which can lead to serious network problems, particularly with NFS. This difficulty is not specific to FreeBSD, but FreeBSD systems are affected by it.

The problem nearly always occurs when (FreeBSD) PC systems are networked with high-performance workstations, such as those made by Silicon Graphics, Inc., and Sun Microsystems, Inc. The NFS mount will work fine, and some operations may succeed, but suddenly the server will seem to become unresponsive to the client, even though requests to and from other systems continue to be processed. This happens to the client system, whether the client is

the FreeBSD system or the workstation. On many systems, there is no way to shut down the client gracefully once this problem has manifested itself. The only solution is often to reset the client, because the NFS situation cannot be resolved.

Though the "correct" solution is to get a higher performance and capacity Ethernet adapter for the FreeBSD system, there is a simple workaround that will allow satisfactory operation. If the FreeBSD system is the *server*, include the option `-w=1024` on the mount from the client. If the FreeBSD system is the *client*, then mount the NFS file system with the option `-r=1024`. These options may be specified using the fourth field of the `fstab` entry on the client for automatic mounts, or by using the `-o` parameter of the mount(8) command for manual mounts.

It should be noted that there is a different problem, sometimes mistaken for this one, when the NFS servers and clients are on different networks. If that is the case, make *certain* that your routers are routing the necessary UDP information, or you will not get anywhere, no matter what else you are doing.

In the following examples, `fastws` is the host (interface) name of a high-performance workstation, and `freebox` is the host (interface) name of a FreeBSD system with a lower-performance Ethernet adapter. Also, `/sharedfs` will be the exported NFS file system (see exports(5)), and `/project` will be the mount point on the client for the exported file system. In all cases, note that additional options, such as `hard` or `soft` and `bg` may be desirable in your application.

Examples for the FreeBSD system (`freebox`) as the client in `/etc/fstab` on `freebox`:

```
fastws:/sharedfs /project nfs rw,-r=1024 0 0
```

As a manual mount command on `freebox`:

```
# mount -t nfs -o -r=1024 fastws:/sharedfs /project
```

Examples for the FreeBSD system as the server in `/etc/fstab` on `fastws`:

```
freebox:/sharedfs /project nfs rw,-w=1024 0 0
```

As a manual mount command on `fastws`:

```
# mount -t nfs -o -w=1024 freebox:/sharedfs /project
```

Nearly any 16-bit Ethernet adapter will allow operation without the above restrictions on the read or write size.

For anyone who cares, here is what happens when the failure occurs, which also explains why it is unrecoverable. NFS typically works with a "block" size of 8 K (though it may do fragments of smaller sizes). Since the maximum Ethernet packet is around 1500 bytes, the NFS "block" gets split into multiple Ethernet packets, even though it is still a single unit to the upper-level code, and must be received, assembled, and *acknowledged* as a unit. The high-performance workstations can pump out the packets which comprise the NFS unit one right after the other, just as close together as the standard allows. On the smaller, lower capacity cards, the later packets overrun the earlier packets of the same unit before they can be transferred to the host and the unit as a whole cannot be reconstructed or acknowledged. As a result, the workstation will time out and try again, but it will try again with the entire 8 K unit, and the process will be repeated, ad infinitum.

By keeping the unit size below the Ethernet packet size limitation, we ensure that any complete Ethernet packet received can be acknowledged individually, avoiding the deadlock situation.

Overruns may still occur when a high-performance workstations is slamming data out to a PC system, but with the better cards, such overruns are not guaranteed on NFS "units". When an overrun occurs, the units affected will be retransmitted, and there will be a fair chance that they will be received, assembled, and acknowledged.

27.4 Network Information System (NIS/YP)

Written by Bill Swingle. Enhanced by Eric Ogren and Udo Erdelhoff.

27.4.1 What Is It?

NIS, which stands for Network Information Services, was developed by Sun Microsystems to centralize administration of UNIX (originally SunOS) systems. It has now essentially become an industry standard; all major UNIX like systems (Solaris, HP-UX, AIX®, Linux, NetBSD, OpenBSD, FreeBSD, etc) support NIS.

NIS was formerly known as Yellow Pages, but because of trademark issues, Sun changed the name. The old term (and yp) is still often seen and used.

It is a RPC-based client/server system that allows a group of machines within an NIS domain to share a common set of configuration files. This permits a system administrator to set up NIS client systems with only minimal configuration data and add, remove or modify configuration data from a single location.

It is similar to the Windows NT® domain system; although the internal implementation of the two are not at all similar, the basic functionality can be compared.

27.4.2 Terms/Processes You Should Know

There are several terms and several important user processes that you will come across when attempting to implement NIS on FreeBSD, whether you are trying to create an NIS server or act as an NIS client:

Term	Description
NIS domainname	An NIS master server and all of its clients (including its slave servers) have a NIS domainname. Similar to an Windows NT domain name, the NIS domainname does not have anything to do with DNS.
rpcbind	Must be running in order to enable RPC (Remote Procedure Call, a network protocol used by NIS). If **rpcbind** is not running, it will be impossible to run an NIS server, or to act as an NIS client (Under FreeBSD 4.X **portmap** is used in place of **rpcbind**).
ypbind	"Binds" an NIS client to its NIS server. It will take the NIS domainname from the system, and using RPC, connect to the server. **ypbind** is the core of client-server communication in an NIS environment; if **ypbind** dies on a client machine, it will not be able to access the NIS server.
ypserv	Should only be running on NIS servers; this is the NIS server process itself. If ypserv(8) dies, then the server will no longer be able to respond to NIS requests (hopefully, there is a slave server to take over for it). There are some implementations of NIS (but not the FreeBSD one), that do not try to reconnect to another server if the server it used before dies. Often, the only thing that helps in this case is to restart the server process (or even the whole server) or the **ypbind** process on the client.
rpc.yppasswdd	Another process that should only be running on NIS master servers; this is a daemon that will allow NIS clients to change their NIS passwords. If this daemon is not running, users will have to login to the NIS master server and change their passwords there.

27.4.3 How Does It Work?

There are three types of hosts in an NIS environment: master servers, slave servers, and clients. Servers act as a central repository for host configuration information. Master servers hold the authoritative copy of this information, while slave servers mirror this information for redundancy. Clients rely on the servers to provide this information to them.

Information in many files can be shared in this manner. The `master.passwd`, `group`, and `hosts` files are commonly shared via NIS. Whenever a process on a client needs information that would normally be found in these files locally, it makes a query to the NIS server that it is bound to instead.

27.4.3.1 Machine Types

- A *NIS master server*. This server, analogous to a Windows NT primary domain controller, maintains the files used by all of the NIS clients. The `passwd`, `group`, and other various files used by the NIS clients live on the master server.

  Note: It is possible for one machine to be an NIS master server for more than one NIS domain. However, this will not be covered in this introduction, which assumes a relatively small-scale NIS environment.

- *NIS slave servers*. Similar to the Windows NT backup domain controllers, NIS slave servers maintain copies of the NIS master's data files. NIS slave servers provide the redundancy, which is needed in important environments. They also help to balance the load of the master server: NIS Clients always attach to the NIS server whose response they get first, and this includes slave-server-replies.
- *NIS clients*. NIS clients, like most Windows NT workstations, authenticate against the NIS server (or the Windows NT domain controller in the Windows NT workstations case) to log on.

27.4.4 Using NIS/YP

This section will deal with setting up a sample NIS environment.

Note: This section assumes that you are running FreeBSD 3.3 or later. The instructions given here will *probably* work for any version of FreeBSD greater than 3.0, but there are no guarantees that this is true.

27.4.4.1 Planning

Let us assume that you are the administrator of a small university lab. This lab, which consists of 15 FreeBSD machines, currently has no centralized point of administration; each machine has its own `/etc/passwd` and `/etc/master.passwd`. These files are kept in sync with each other only through manual intervention; currently, when you add a user to the lab, you must run `adduser` on all 15 machines. Clearly, this has to change, so you have decided to convert the lab to use NIS, using two of the machines as servers.

Therefore, the configuration of the lab now looks something like:

Machine name	IP address	Machine role
ellington	10.0.0.2	NIS master
coltrane	10.0.0.3	NIS slave
basie	10.0.0.4	Faculty workstation
bird	10.0.0.5	Client machine
cli[1-11]	10.0.0.[6-17]	Other client machines

If you are setting up a NIS scheme for the first time, it is a good idea to think through how you want to go about it. No matter what the size of your network, there are a few decisions that need to be made.

27.4.4.1.1 Choosing a NIS Domain Name

This might not be the "domainname" that you are used to. It is more accurately called the "NIS domainname". When a client broadcasts its requests for info, it includes the name of the NIS domain that it is part of. This is how multiple servers on one network can tell which server should answer which request. Think of the NIS domainname as the name for a group of hosts that are related in some way.

Some organizations choose to use their Internet domainname for their NIS domainname. This is not recommended as it can cause confusion when trying to debug network problems. The NIS domainname should be unique within your network and it is helpful if it describes the group of machines it represents. For example, the Art department at Acme Inc. might be in the "acme-art" NIS domain. For this example, assume you have chosen the name `test-domain`.

However, some operating systems (notably SunOS) use their NIS domain name as their Internet domain name. If one or more machines on your network have this restriction, you *must* use the Internet domain name as your NIS domain name.

27.4.4.1.2 Physical Server Requirements

There are several things to keep in mind when choosing a machine to use as a NIS server. One of the unfortunate things about NIS is the level of dependency the clients have on the server. If a client cannot contact the server for its NIS domain, very often the machine becomes unusable. The lack of user and group information causes most systems to temporarily freeze up. With this in mind you should make sure to choose a machine that will not be prone to being rebooted regularly, or one that might be used for development. The NIS server should ideally be a stand alone machine whose sole purpose in life is to be an NIS server. If you have a network that is not very heavily used, it is acceptable to put the NIS server on a machine running other services, just keep in mind that if the NIS server becomes unavailable, it will affect *all* of your NIS clients adversely.

27.4.4.2 NIS Servers

The canonical copies of all NIS information are stored on a single machine called the NIS master server. The databases used to store the information are called NIS maps. In FreeBSD, these maps are stored in `/var/yp/[domainname]` where `[domainname]` is the name of the NIS domain being served. A single NIS server can support several domains at once, therefore it is possible to have several such directories, one for each supported domain. Each domain will have its own independent set of maps.

NIS master and slave servers handle all NIS requests with the `ypserv` daemon. `ypserv` is responsible for receiving incoming requests from NIS clients, translating the requested domain and map name to a path to the corresponding database file and transmitting data from the database back to the client.

27.4.4.2.1 Setting Up a NIS Master Server

Setting up a master NIS server can be relatively straight forward, depending on your needs. FreeBSD comes with support for NIS out-of-the-box. All you need is to add the following lines to `/etc/rc.conf`, and FreeBSD will do the rest for you.

1. `nisdomainname="test-domain"`
   This line will set the NIS domainname to `test-domain` upon network setup (e.g. after reboot).
2. `nis_server_enable="YES"`
   This will tell FreeBSD to start up the NIS server processes when the networking is next brought up.
3. `nis_yppasswdd_enable="YES"`
   This will enable the `rpc.yppasswdd` daemon which, as mentioned above, will allow users to change their NIS password from a client machine.

> **Note:** Depending on your NIS setup, you may need to add further entries. See the section about NIS servers that are also NIS clients, below, for details.

Now, all you have to do is to run the command `/etc/netstart` as superuser. It will set up everything for you, using the values you defined in `/etc/rc.conf`.

27.4.4.2.2 Initializing the NIS Maps

The *NIS maps* are database files, that are kept in the `/var/yp` directory. They are generated from configuration files in the `/etc` directory of the NIS master, with one exception: the `/etc/master.passwd` file. This is for a good reason, you do not want to propagate passwords to your `root` and other administrative accounts to all the servers in the NIS domain. Therefore, before we initialize the NIS maps, you should:

```
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```

You should remove all entries regarding system accounts (`bin`, `tty`, `kmem`, `games`, etc), as well as any accounts that you do not want to be propagated to the NIS clients (for example `root` and any other UID 0 (superuser) accounts).

> **Note:** Make sure the `/var/yp/master.passwd` is neither group nor world readable (mode 600)! Use the `chmod` command, if appropriate.

When you have finished, it is time to initialize the NIS maps! FreeBSD includes a script named `ypinit` to do this for you (see its manual page for more information). Note that this script is available on most UNIX Operating Systems, but not on all. On Digital UNIX/Compaq Tru64 UNIX it is called `ypsetup`. Because we are generating maps for an

NIS master, we are going to pass the `-m` option to `ypinit`. To generate the NIS maps, assuming you already performed the steps above, run:

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   :  ellington
next host to add:  coltrane
next host to add:  ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct?  [y/n: y] y

[..output from map generation..]

NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

`ypinit` should have created `/var/yp/Makefile` from `/var/yp/Makefile.dist`. When created, this file assumes that you are operating in a single server NIS environment with only FreeBSD machines. Since `test-domain` has a slave server as well, you must edit `/var/yp/Makefile`:

```
ellington# vi /var/yp/Makefile
```

You should comment out the line that says

```
NOPUSH = "True"
```

(if it is not commented out already).

27.4.4.2.3 Setting up a NIS Slave Server

Setting up an NIS slave server is even more simple than setting up the master. Log on to the slave server and edit the file `/etc/rc.conf` as you did before. The only difference is that we now must use the `-s` option when running `ypinit`. The `-s` option requires the name of the NIS master be passed to it as well, so our command line looks like:

```
coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington

Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
```

```
Do you want this procedure to quit on non-fatal errors? [y/n: n]  n

Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
Transferring netgroup...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byhost...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred
Transferring group.byname...
ypxfr: Exiting: Map successfully transferred
Transferring services.byname...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors.
Don't forget to update map ypservers on ellington.
```

You should now have a directory called `/var/yp/test-domain`. Copies of the NIS master server's maps should be in this directory. You will need to make sure that these stay updated. The following `/etc/crontab` entries on your slave servers should do the job:

```
20      *       *       *       *       root    /usr/libexec/ypxfr passwd.byname
21      *       *       *       *       root    /usr/libexec/ypxfr passwd.byuid
```

These two lines force the slave to sync its maps with the maps on the master server. Although these entries are not mandatory, since the master server attempts to ensure any changes to its NIS maps are communicated to its slaves and because password information is vital to systems depending on the server, it is a good idea to force the updates. This is more important on busy networks where map updates might not always complete.

Now, run the command `/etc/netstart` on the slave server as well, which again starts the NIS server.

27.4.4.3 NIS Clients

An NIS client establishes what is called a binding to a particular NIS server using the `ypbind` daemon. `ypbind` checks the system's default domain (as set by the `domainname` command), and begins broadcasting RPC requests on the local network. These requests specify the name of the domain for which `ypbind` is attempting to establish a binding. If a server that has been configured to serve the requested domain receives one of the broadcasts, it will respond to `ypbind`, which will record the server's address. If there are several servers available (a master and several slaves, for example), `ypbind` will use the address of the first one to respond. From that point on, the client system will direct all of its NIS requests to that server. `ypbind` will occasionally "ping" the server to make sure it is still up and running. If it fails to receive a reply to one of its pings within a reasonable amount of time, `ypbind` will mark the domain as unbound and begin broadcasting again in the hopes of locating another server.

27.4.4.3.1 Setting Up a NIS Client

Setting up a FreeBSD machine to be a NIS client is fairly straightforward.

1. Edit the file `/etc/rc.conf` and add the following lines in order to set the NIS domainname and start `ypbind` upon network startup:

   ```
   nisdomainname="test-domain"
   nis_client_enable="YES"
   ```

2. To import all possible password entries from the NIS server, remove all user accounts from your `/etc/master.passwd` file and use `vipw` to add the following line to the end of the file:

   ```
   +:::::::::
   ```

   Note: This line will afford anyone with a valid account in the NIS server's password maps an account. There are many ways to configure your NIS client by changing this line. See the netgroups section below for more information. For more detailed reading see O'Reilly's book on `Managing NFS and NIS`.

   Note: You should keep at least one local account (i.e. not imported via NIS) in your `/etc/master.passwd` and this account should also be a member of the group `wheel`. If there is something wrong with NIS, this account can be used to log in remotely, become `root`, and fix things.

3. To import all possible group entries from the NIS server, add this line to your `/etc/group` file:

   ```
   +:*::
   ```

After completing these steps, you should be able to run `ypcat passwd` and see the NIS server's passwd map.

27.4.5 NIS Security

In general, any remote user can issue an RPC to ypserv(8) and retrieve the contents of your NIS maps, provided the remote user knows your domainname. To prevent such unauthorized transactions, ypserv(8) supports a feature called "securenets" which can be used to restrict access to a given set of hosts. At startup, ypserv(8) will attempt to load the securenets information from a file called `/var/yp/securenets`.

> **Note:** This path varies depending on the path specified with the `-p` option. This file contains entries that consist of a network specification and a network mask separated by white space. Lines starting with "#" are considered to be comments. A sample securenets file might look like this:

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
10.0.0.0      255.255.240.0
```

If ypserv(8) receives a request from an address that matches one of these rules, it will process the request normally. If the address fails to match a rule, the request will be ignored and a warning message will be logged. If the `/var/yp/securenets` file does not exist, `ypserv` will allow connections from any host.

The `ypserv` program also has support for Wietse Venema's **TCP Wrapper** package. This allows the administrator to use the **TCP Wrapper** configuration files for access control instead of `/var/yp/securenets`.

> **Note:** While both of these access control mechanisms provide some security, they, like the privileged port test, are vulnerable to "IP spoofing" attacks. All NIS-related traffic should be blocked at your firewall.
>
> Servers using `/var/yp/securenets` may fail to serve legitimate NIS clients with archaic TCP/IP implementations. Some of these implementations set all host bits to zero when doing broadcasts and/or fail to observe the subnet mask when calculating the broadcast address. While some of these problems can be fixed by changing the client configuration, other problems may force the retirement of the client systems in question or the abandonment of `/var/yp/securenets`.
>
> Using `/var/yp/securenets` on a server with such an archaic implementation of TCP/IP is a really bad idea and will lead to loss of NIS functionality for large parts of your network.
>
> The use of the **TCP Wrapper** package increases the latency of your NIS server. The additional delay may be long enough to cause timeouts in client programs, especially in busy networks or with slow NIS servers. If one or more of your client systems suffers from these symptoms, you should convert the client systems in question into NIS slave servers and force them to bind to themselves.

27.4.6 Barring Some Users from Logging On

In our lab, there is a machine `basie` that is supposed to be a faculty only workstation. We do not want to take this machine out of the NIS domain, yet the `passwd` file on the master NIS server contains accounts for both faculty and students. What can we do?

There is a way to bar specific users from logging on to a machine, even if they are present in the NIS database. To do this, all you must do is add *`-username`* to the end of the `/etc/master.passwd` file on the client machine, where *`username`* is the username of the user you wish to bar from logging in. This should preferably be done using `vipw`, since `vipw` will sanity check your changes to `/etc/master.passwd`, as well as automatically rebuild the password database when you finish editing. For example, if we wanted to bar user `bill` from logging on to `basie` we would:

```
basie# vipw
[add -bill to the end, exit]
vipw: rebuilding the database...
vipw: done

basie# cat /etc/master.passwd

root:[password]:0:0::0:0:The super-user:/root:/bin/csh
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5::0:0:System &:/:/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/share/man:/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/sbin/nologin
+:::::::::
-bill

basie#
```

27.4.7 Using Netgroups

Contributed by Udo Erdelhoff.

The method shown in the previous section works reasonably well if you need special rules for a very small number of users and/or machines. On larger networks, you *will* forget to bar some users from logging onto sensitive machines, or you may even have to modify each machine separately, thus losing the main benefit of NIS: *centralized* administration.

The NIS developers' solution for this problem is called *netgroups*. Their purpose and semantics can be compared to the normal groups used by UNIX file systems. The main differences are the lack of a numeric ID and the ability to define a netgroup by including both user accounts and other netgroups.

Netgroups were developed to handle large, complex networks with hundreds of users and machines. On one hand, this is a Good Thing if you are forced to deal with such a situation. On the other hand, this complexity makes it almost impossible to explain netgroups with really simple examples. The example used in the remainder of this section demonstrates this problem.

Let us assume that your successful introduction of NIS in your laboratory caught your superiors' interest. Your next job is to extend your NIS domain to cover some of the other machines on campus. The two tables contain the names of the new users and new machines as well as brief descriptions of them.

User Name(s)	Description
`alpha`, `beta`	Normal employees of the IT department
`charlie`, `delta`	The new apprentices of the IT department
`echo`, `foxtrott`, `golf`, ...	Ordinary employees
`able`, `baker`, ...	The current interns

Machine Name(s)	Description
`war`, `death`, `famine`, `pollution`	Your most important servers. Only the IT employees are allowed to log onto these machines.
`pride`, `greed`, `envy`, `wrath`, `lust`, `sloth`	Less important servers. All members of the IT department are allowed to login onto these machines.
`one`, `two`, `three`, `four`, ...	Ordinary workstations. Only the *real* employees are allowed to use these machines.
`trashcan`	A very old machine without any critical data. Even the intern is allowed to use this box.

If you tried to implement these restrictions by separately blocking each user, you would have to add one `-user` line to each system's `passwd` for each user who is not allowed to login onto that system. If you forget just one entry, you could be in trouble. It may be feasible to do this correctly during the initial setup, however you *will* eventually forget to add the lines for new users during day-to-day operations. After all, Murphy was an optimist.

Handling this situation with netgroups offers several advantages. Each user need not be handled separately; you assign a user to one or more netgroups and allow or forbid logins for all members of the netgroup. If you add a new machine, you will only have to define login restrictions for netgroups. If a new user is added, you will only have to add the user to one or more netgroups. Those changes are independent of each other: no more "for each combination of user and machine do..." If your NIS setup is planned carefully, you will only have to modify exactly one central configuration file to grant or deny access to machines.

The first step is the initialization of the NIS map netgroup. FreeBSD's ypinit(8) does not create this map by default, but its NIS implementation will support it once it has been created. To create an empty map, simply type

```
ellington# vi /var/yp/netgroup
```

and start adding content. For our example, we need at least four netgroups: IT employees, IT apprentices, normal employees and interns.

```
IT_EMP  (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain)     (,foxtrott,test-domain) \
        (,golf,test-domain)
```

```
INTERNS (,able,test-domain)     (,baker,test-domain)
```

`IT_EMP`, `IT_APP` etc. are the names of the netgroups. Each bracketed group adds one or more user accounts to it. The three fields inside a group are:

1. The name of the host(s) where the following items are valid. If you do not specify a hostname, the entry is valid on all hosts. If you do specify a hostname, you will enter a realm of darkness, horror and utter confusion.
2. The name of the account that belongs to this netgroup.
3. The NIS domain for the account. You can import accounts from other NIS domains into your netgroup if you are one of the unlucky fellows with more than one NIS domain.

Each of these fields can contain wildcards. See netgroup(5) for details.

> **Note:** Netgroup names longer than 8 characters should not be used, especially if you have machines running other operating systems within your NIS domain. The names are case sensitive; using capital letters for your netgroup names is an easy way to distinguish between user, machine and netgroup names.
>
> Some NIS clients (other than FreeBSD) cannot handle netgroups with a large number of entries. For example, some older versions of SunOS start to cause trouble if a netgroup contains more than 15 *entries*. You can circumvent this limit by creating several sub-netgroups with 15 users or less and a real netgroup that consists of the sub-netgroups:
>
> ```
> BIGGRP1  (,joe1,domain)  (,joe2,domain)  (,joe3,domain) [...]
> BIGGRP2  (,joe16,domain)  (,joe17,domain) [...]
> BIGGRP3  (,joe31,domain)  (,joe32,domain)
> BIGGROUP  BIGGRP1 BIGGRP2 BIGGRP3
> ```
>
> You can repeat this process if you need more than 225 users within a single netgroup.

Activating and distributing your new NIS map is easy:

```
ellington# cd /var/yp
ellington# make
```

This will generate the three NIS maps `netgroup`, `netgroup.byhost` and `netgroup.byuser`. Use ypcat(1) to check if your new NIS maps are available:

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

The output of the first command should resemble the contents of `/var/yp/netgroup`. The second command will not produce output if you have not specified host-specific netgroups. The third command can be used to get the list of netgroups for a user.

The client setup is quite simple. To configure the server `war`, you only have to start vipw(8) and replace the line

```
+::::::::::
```

with

```
+@IT_EMP:::::::::
```

Now, only the data for the users defined in the netgroup `IT_EMP` is imported into `war`'s password database and only these users are allowed to login.

Unfortunately, this limitation also applies to the ~ function of the shell and all routines converting between user names and numerical user IDs. In other words, `cd ~user` will not work, `ls -l` will show the numerical ID instead of the username and `find . -user joe -print` will fail with "`No such user`". To fix this, you will have to import all user entries *without allowing them to login onto your servers*.

This can be achieved by adding another line to `/etc/master.passwd`. This line should contain:

`+::::::::::/sbin/nologin`, meaning "Import all entries but replace the shell with `/sbin/nologin` in the imported entries". You can replace any field in the `passwd` entry by placing a default value in your `/etc/master.passwd`.

> **Warning:** Make sure that the line `+::::::::::/sbin/nologin` is placed after `+@IT_EMP:::::::::`. Otherwise, all user accounts imported from NIS will have `/sbin/nologin` as their login shell.

After this change, you will only have to change one NIS map if a new employee joins the IT department. You could use a similar approach for the less important servers by replacing the old `+::::::::::` in their local version of `/etc/master.passwd` with something like this:

```
+@IT_EMP:::::::::
+@IT_APP:::::::::
+::::::::::/sbin/nologin
```

The corresponding lines for the normal workstations could be:

```
+@IT_EMP:::::::::
+@USERS:::::::::
+::::::::::/sbin/nologin
```

And everything would be fine until there is a policy change a few weeks later: The IT department starts hiring interns. The IT interns are allowed to use the normal workstations and the less important servers; and the IT apprentices are allowed to login onto the main servers. You add a new netgroup `IT_INTERN`, add the new IT interns to this netgroup and start to change the configuration on each and every machine... As the old saying goes: "Errors in centralized planning lead to global mess".

NIS' ability to create netgroups from other netgroups can be used to prevent situations like these. One possibility is the creation of role-based netgroups. For example, you could create a netgroup called `BIGSRV` to define the login restrictions for the important servers, another netgroup called `SMALLSRV` for the less important servers and a third netgroup called `USERBOX` for the normal workstations. Each of these netgroups contains the netgroups that are allowed to login onto these machines. The new entries for your NIS map netgroup should look like this:

```
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP  ITINTERN
USERBOX   IT_EMP  ITINTERN USERS
```

This method of defining login restrictions works reasonably well if you can define groups of machines with identical restrictions. Unfortunately, this is the exception and not the rule. Most of the time, you will need the ability to define login restrictions on a per-machine basis.

Machine-specific netgroup definitions are the other possibility to deal with the policy change outlined above. In this scenario, the `/etc/master.passwd` of each box contains two lines starting with "+". The first of them adds a netgroup with the accounts allowed to login onto this machine, the second one adds all other accounts with `/sbin/nologin` as shell. It is a good idea to use the "ALL-CAPS" version of the machine name as the name of the netgroup. In other words, the lines should look like this:

```
+@BOXNAME:::::::::
+::::::::::/sbin/nologin
```

Once you have completed this task for all your machines, you will not have to modify the local versions of `/etc/master.passwd` ever again. All further changes can be handled by modifying the NIS map. Here is an example of a possible netgroup map for this scenario with some additional goodies:

```
# Define groups of users first
IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
ITINTERN  (,kilo,test-domain)     (,lima,test-domain)
D_INTERNS (,able,test-domain)     (,baker,test-domain)
#
# Now, define some groups based on roles
USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS
#
# And a groups for a special tasks
# Allow echo and golf to access our anti-virus-machine
SECURITY  IT_EMP  (,echo,test-domain)  (,golf,test-domain)
#
# machine-based netgroups
# Our main servers
WAR       BIGSRV
FAMINE    BIGSRV
# User india needs access to this server
POLLUTION  BIGSRV  (,india,test-domain)
#
# This one is really important and needs more access restrictions
DEATH     IT_EMP
#
# The anti-virus-machine mentioned above
ONE       SECURITY
#
# Restrict a machine to a single user
TWO       (,hotel,test-domain)
# [...more groups to follow]
```

If you are using some kind of database to manage your user accounts, you should be able to create the first part of the map with your database's report tools. This way, new users will automatically have access to the boxes.

One last word of caution: It may not always be advisable to use machine-based netgroups. If you are deploying a couple of dozen or even hundreds of identical machines for student labs, you should use role-based netgroups instead of machine-based netgroups to keep the size of the NIS map within reasonable limits.

27.4.8 Important Things to Remember

There are still a couple of things that you will need to do differently now that you are in an NIS environment.

- Every time you wish to add a user to the lab, you must add it to the master NIS server *only*, and *you must remember to rebuild the NIS maps*. If you forget to do this, the new user will not be able to login anywhere except on the NIS master. For example, if we needed to add a new user `jsmith` to the lab, we would:

  ```
  # pw useradd jsmith
  # cd /var/yp
  # make test-domain
  ```

  You could also run `adduser jsmith` instead of `pw useradd jsmith`.

- *Keep the administration accounts out of the NIS maps*. You do not want to be propagating administrative accounts and passwords to machines that will have users that should not have access to those accounts.

- *Keep the NIS master and slave secure, and minimize their downtime*. If somebody either hacks or simply turns off these machines, they have effectively rendered many people without the ability to login to the lab.

  This is the chief weakness of any centralized administration system. If you do not protect your NIS servers, you will have a lot of angry users!

27.4.9 NIS v1 Compatibility

FreeBSD's **ypserv** has some support for serving NIS v1 clients. FreeBSD's NIS implementation only uses the NIS v2 protocol, however other implementations include support for the v1 protocol for backwards compatibility with older systems. The **ypbind** daemons supplied with these systems will try to establish a binding to an NIS v1 server even though they may never actually need it (and they may persist in broadcasting in search of one even after they receive a response from a v2 server). Note that while support for normal client calls is provided, this version of **ypserv** does not handle v1 map transfer requests; consequently, it cannot be used as a master or slave in conjunction with older NIS servers that only support the v1 protocol. Fortunately, there probably are not any such servers still in use today.

27.4.10 NIS Servers That Are Also NIS Clients

Care must be taken when running **ypserv** in a multi-server domain where the server machines are also NIS clients. It is generally a good idea to force the servers to bind to themselves rather than allowing them to broadcast bind requests and possibly become bound to each other. Strange failure modes can result if one server goes down and others are dependent upon it. Eventually all the clients will time out and attempt to bind to other servers, but the delay involved can be considerable and the failure mode is still present since the servers might bind to each other all over again.

You can force a host to bind to a particular server by running `ypbind` with the `-S` flag. If you do not want to do this manually each time you reboot your NIS server, you can add the following lines to your `/etc/rc.conf`:

```
nis_client_enable="YES" # run client stuff as well
```

```
nis_client_flags="-S NIS domain,server"
```

See ypbind(8) for further information.

27.4.11 Password Formats

One of the most common issues that people run into when trying to implement NIS is password format compatibility. If your NIS server is using DES encrypted passwords, it will only support clients that are also using DES. For example, if you have Solaris NIS clients in your network, then you will almost certainly need to use DES encrypted passwords.

To check which format your servers and clients are using, look at `/etc/login.conf`. If the host is configured to use DES encrypted passwords, then the `default` class will contain an entry like this:

```
default:\
 :passwd_format=des:\
 :copyright=/etc/COPYRIGHT:\
 [Further entries elided]
```

Other possible values for the `passwd_format` capability include `blf` and `md5` (for Blowfish and MD5 encrypted passwords, respectively).

If you have made changes to `/etc/login.conf`, you will also need to rebuild the login capability database, which is achieved by running the following command as `root`:

```
# cap_mkdb /etc/login.conf
```

> **Note:** The format of passwords already in `/etc/master.passwd` will not be updated until a user changes his password for the first time *after* the login capability database is rebuilt.

Next, in order to ensure that passwords are encrypted with the format that you have chosen, you should also check that the `crypt_default` in `/etc/auth.conf` gives precedence to your chosen password format. To do this, place the format that you have chosen first in the list. For example, when using DES encrypted passwords, the entry would be:

```
crypt_default = des blf md5
```

Having followed the above steps on each of the FreeBSD based NIS servers and clients, you can be sure that they all agree on which password format is used within your network. If you have trouble authenticating on an NIS client, this is a pretty good place to start looking for possible problems. Remember: if you want to deploy an NIS server for a heterogenous network, you will probably have to use DES on all systems because it is the lowest common standard.

27.5 Automatic Network Configuration (DHCP)

Written by Greg Sutter.

27.5.1 What Is DHCP?

DHCP, the Dynamic Host Configuration Protocol, describes the means by which a system can connect to a network and obtain the necessary information for communication upon that network. FreeBSD versions prior to 6.0 use the ISC (Internet Software Consortium) DHCP client (dhclient(8)) implementation. Later versions use the OpenBSD `dhclient` taken from OpenBSD 3.7. All information here regarding `dhclient` is for use with either of the ISC or OpenBSD DHCP clients. The DHCP server is the one included in the ISC distribution.

27.5.2 What This Section Covers

This section describes both the client-side components of the ISC and OpenBSD DHCP client and server-side components of the ISC DHCP system. The client-side program, `dhclient`, comes integrated within FreeBSD, and the server-side portion is available from the `net/isc-dhcp3-server` port. The dhclient(8), dhcp-options(5), and dhclient.conf(5) manual pages, in addition to the references below, are useful resources.

27.5.3 How It Works

When `dhclient`, the DHCP client, is executed on the client machine, it begins broadcasting requests for configuration information. By default, these requests are on UDP port 68. The server replies on UDP 67, giving the client an IP address and other relevant network information such as netmask, router, and DNS servers. All of this information comes in the form of a DHCP "lease" and is only valid for a certain time (configured by the DHCP server maintainer). In this manner, stale IP addresses for clients no longer connected to the network can be automatically reclaimed.

DHCP clients can obtain a great deal of information from the server. An exhaustive list may be found in dhcp-options(5).

27.5.4 FreeBSD Integration

FreeBSD fully integrates the ISC or OpenBSD DHCP client, `dhclient` (according to the FreeBSD version you run). DHCP client support is provided within both the installer and the base system, obviating the need for detailed knowledge of network configurations on any network that runs a DHCP server. `dhclient` has been included in all FreeBSD distributions since 3.2.

DHCP is supported by **sysinstall**. When configuring a network interface within **sysinstall**, the second question asked is: "Do you want to try DHCP configuration of the interface?". Answering affirmatively will execute `dhclient`, and if successful, will fill in the network configuration information automatically.

There are two things you must do to have your system use DHCP upon startup:

- Make sure that the `bpf` device is compiled into your kernel. To do this, add `device bpf` (`pseudo-device bpf` under FreeBSD 4.X) to your kernel configuration file, and rebuild the kernel. For more information about building kernels, see Chapter 8.

The `bpf` device is already part of the `GENERIC` kernel that is supplied with FreeBSD, so if you do not have a custom kernel, you should not need to create one in order to get DHCP working.

> **Note:** For those who are particularly security conscious, you should be warned that `bpf` is also the device that allows packet sniffers to work correctly (although they still have to be run as `root`). `bpf` *is* required to use DHCP, but if you are very sensitive about security, you probably should not add `bpf` to your kernel in the expectation that at some point in the future you will be using DHCP.

- Edit your `/etc/rc.conf` to include the following:

  ```
  ifconfig_fxp0="DHCP"
  ```

  > **Note:** Be sure to replace `fxp0` with the designation for the interface that you wish to dynamically configure, as described in Section 11.8.

  If you are using a different location for `dhclient`, or if you wish to pass additional flags to `dhclient`, also include the following (editing as necessary):

  ```
  dhcp_program="/sbin/dhclient"
  dhcp_flags=""
  ```

The DHCP server, **dhcpd**, is included as part of the `net/isc-dhcp3-server` port in the ports collection. This port contains the ISC DHCP server and documentation.

27.5.5 Files

- `/etc/dhclient.conf`

  `dhclient` requires a configuration file, `/etc/dhclient.conf`. Typically the file contains only comments, the defaults being reasonably sane. This configuration file is described by the dhclient.conf(5) manual page.

- `/sbin/dhclient`

  `dhclient` is statically linked and resides in `/sbin`. The dhclient(8) manual page gives more information about `dhclient`.

- `/sbin/dhclient-script`

  `dhclient-script` is the FreeBSD-specific DHCP client configuration script. It is described in dhclient-script(8), but should not need any user modification to function properly.

- `/var/db/dhclient.leases`

  The DHCP client keeps a database of valid leases in this file, which is written as a log. dhclient.leases(5) gives a slightly longer description.

27.5.6 Further Reading

The DHCP protocol is fully described in RFC 2131 (http://www.freesoft.org/CIE/RFC/2131/). An informational resource has also been set up at http://www.dhcp.org/.

27.5.7 Installing and Configuring a DHCP Server

27.5.7.1 What This Section Covers

This section provides information on how to configure a FreeBSD system to act as a DHCP server using the ISC (Internet Software Consortium) implementation of the DHCP suite.

The server portion of the suite is not provided as part of FreeBSD, and so you will need to install the `net/isc-dhcp3-server` port to provide this service. See Chapter 4 for more information on using the Ports Collection.

27.5.7.2 DHCP Server Installation

In order to configure your FreeBSD system as a DHCP server, you will need to ensure that the bpf(4) device is compiled into your kernel. To do this, add `device bpf` (`pseudo-device bpf` under FreeBSD 4.X) to your kernel configuration file, and rebuild the kernel. For more information about building kernels, see Chapter 8.

The `bpf` device is already part of the `GENERIC` kernel that is supplied with FreeBSD, so you do not need to create a custom kernel in order to get DHCP working.

> **Note:** Those who are particularly security conscious should note that `bpf` is also the device that allows packet sniffers to work correctly (although such programs still need privileged access). `bpf` *is* required to use DHCP, but if you are very sensitive about security, you probably should not include `bpf` in your kernel purely because you expect to use DHCP at some point in the future.

The next thing that you will need to do is edit the sample `dhcpd.conf` which was installed by the `net/isc-dhcp3-server` port. By default, this will be `/usr/local/etc/dhcpd.conf.sample`, and you should copy this to `/usr/local/etc/dhcpd.conf` before proceeding to make changes.

27.5.7.3 Configuring the DHCP Server

`dhcpd.conf` is comprised of declarations regarding subnets and hosts, and is perhaps most easily explained using an example :

```
option domain-name "example.com";❶
option domain-name-servers 192.168.4.100;❷
option subnet-mask 255.255.255.0;❸

default-lease-time 3600;❹
max-lease-time 86400;❺
ddns-update-style none;❻

subnet 192.168.4.0 netmask 255.255.255.0 {
```

```
  range 192.168.4.129 192.168.4.254;❼
  option routers 192.168.4.1;❽
}

host mailhost {
  hardware ethernet 02:03:04:05:06:07;❾
  fixed-address mailhost.example.com;(10)
}
```

❶ This option specifies the domain that will be provided to clients as the default search domain. See resolv.conf(5) for more information on what this means.

❷ This option specifies a comma separated list of DNS servers that the client should use.

❸ The netmask that will be provided to clients.

❹ A client may request a specific length of time that a lease will be valid. Otherwise the server will assign a lease with this expiry value (in seconds).

❺ This is the maximum length of time that the server will lease for. Should a client request a longer lease, a lease will be issued, although it will only be valid for `max-lease-time` seconds.

❻ This option specifies whether the DHCP server should attempt to update DNS when a lease is accepted or released. In the ISC implementation, this option is *required*.

❼ This denotes which IP addresses should be used in the pool reserved for allocating to clients. IP addresses between, and including, the ones stated are handed out to clients.

❽ Declares the default gateway that will be provided to clients.

❾ The hardware MAC address of a host (so that the DHCP server can recognize a host when it makes a request).

(10) Specifies that the host should always be given the same IP address. Note that using a hostname is correct here, since the DHCP server will resolve the hostname itself before returning the lease information.

Once you have finished writing your `dhcpd.conf`, you can proceed to start the server by issuing the following command:

```
# /usr/local/etc/rc.d/isc-dhcpd.sh start
```

Should you need to make changes to the configuration of your server in the future, it is important to note that sending a `SIGHUP` signal to **dhcpd** does *not* result in the configuration being reloaded, as it does with most daemons. You will need to send a `SIGTERM` signal to stop the process, and then restart it using the command above.

27.5.7.4 Files

- `/usr/local/sbin/dhcpd`

  dhcpd is statically linked and resides in `/usr/local/sbin`. The dhcpd(8) manual page installed with the port gives more information about **dhcpd**.

- `/usr/local/etc/dhcpd.conf`

  dhcpd requires a configuration file, `/usr/local/etc/dhcpd.conf` before it will start providing service to clients. This file needs to contain all the information that should be provided to clients that are being serviced,

along with information regarding the operation of the server. This configuration file is described by the dhcpd.conf(5) manual page installed by the port.

- `/var/db/dhcpd.leases`

  The DHCP server keeps a database of leases it has issued in this file, which is written as a log. The manual page dhcpd.leases(5), installed by the port gives a slightly longer description.

- `/usr/local/sbin/dhcrelay`

  dhcrelay is used in advanced environments where one DHCP server forwards a request from a client to another DHCP server on a separate network. If you require this functionality, then install the `net/isc-dhcp3-relay` port. The dhcrelay(8) manual page provided with the port contains more detail.

27.6 Domain Name System (DNS)

Contributed by Chern Lee.

27.6.1 Overview

FreeBSD utilizes, by default, a version of BIND (Berkeley Internet Name Domain), which is the most common implementation of the DNS protocol. DNS is the protocol through which names are mapped to IP addresses, and vice versa. For example, a query for `www.FreeBSD.org` will receive a reply with the IP address of The FreeBSD Project's web server, whereas, a query for `ftp.FreeBSD.org` will return the IP address of the corresponding FTP machine. Likewise, the opposite can happen. A query for an IP address can resolve its hostname. It is not necessary to run a name server to perform DNS lookups on a system.

DNS is coordinated across the Internet through a somewhat complex system of authoritative root name servers, and other smaller-scale name servers who host and cache individual domain information.

This document refers to BIND 8.x, as it is the stable version used in FreeBSD. Versions of FreeBSD 5.3 and beyond include BIND9 and the configuration instructions may be found later in this chapter. Users of FreeBSD 5.2 and other previous versions may install BIND9 from the `net/bind9` port.

RFC1034 and RFC1035 dictate the DNS protocol.

Currently, BIND is maintained by the Internet Software Consortium http://www.isc.org/.

27.6.2 Terminology

To understand this document, some terms related to DNS must be understood.

Term	Definition
Forward DNS	Mapping of hostnames to IP addresses
Origin	Refers to the domain covered in a particular zone file
named, BIND, name server	Common names for the BIND name server package within FreeBSD
Resolver	A system process through which a machine queries a name server for zone information

Term	Definition
Reverse DNS	The opposite of forward DNS; mapping of IP addresses to hostnames
Root zone	The beginning of the Internet zone hierarchy. All zones fall under the root zone, similar to how all files in a file system fall under the root directory.
Zone	An individual domain, subdomain, or portion of the DNS administered by the same authority

Examples of zones:

- `.` is the root zone
- `org.` is a zone under the root zone
- `example.org.` is a zone under the `org.` zone
- `foo.example.org.` is a subdomain, a zone under the `example.org.` zone
- `1.2.3.in-addr.arpa` is a zone referencing all IP addresses which fall under the `3.2.1.*` IP space.

As one can see, the more specific part of a hostname appears to its left. For example, `example.org.` is more specific than `org.`, as `org.` is more specific than the root zone. The layout of each part of a hostname is much like a file system: the `/dev` directory falls within the root, and so on.

27.6.3 Reasons to Run a Name Server

Name servers usually come in two forms: an authoritative name server, and a caching name server.

An authoritative name server is needed when:

- one wants to serve DNS information to the world, replying authoritatively to queries.
- a domain, such as `example.org`, is registered and IP addresses need to be assigned to hostnames under it.
- an IP address block requires reverse DNS entries (IP to hostname).
- a backup name server, called a slave, must reply to queries when the primary is down or inaccessible.

A caching name server is needed when:

- a local DNS server may cache and respond more quickly than querying an outside name server.
- a reduction in overall network traffic is desired (DNS traffic has been measured to account for 5% or more of total Internet traffic).

When one queries for `www.FreeBSD.org`, the resolver usually queries the uplink ISP's name server, and retrieves the reply. With a local, caching DNS server, the query only has to be made once to the outside world by the caching DNS server. Every additional query will not have to look to the outside of the local network, since the information is cached locally.

27.6.4 How It Works

In FreeBSD, the BIND daemon is called **named** for obvious reasons.

File	Description
named	the BIND daemon
`ndc`	name daemon control program
`/etc/namedb`	directory where BIND zone information resides
`/etc/namedb/named.conf`	daemon configuration file

Zone files are usually contained within the `/etc/namedb` directory, and contain the DNS zone information served by the name server.

27.6.5 Starting BIND

Since BIND is installed by default, configuring it all is relatively simple.

To ensure the **named** daemon is started at boot, put the following line in `/etc/rc.conf`:

```
named_enable="YES"
```

To start the daemon manually (after configuring it):

```
# ndc start
```

27.6.6 Configuration Files

27.6.6.1 Using `make-localhost`

Be sure to:

```
# cd /etc/namedb
# sh make-localhost
```

to properly create the local reverse DNS zone file in `/etc/namedb/master/localhost.rev`.

27.6.6.2 `/etc/namedb/named.conf`

```
// $FreeBSD$
//
// Refer to the named(8) manual page for details.  If you are ever going
// to setup a primary server, make sure you've understood the hairy
// details of how DNS is working.  Even with simple mistakes, you can
// break connectivity for affected parties, or cause huge amount of
// useless Internet traffic.

options {
        directory "/etc/namedb";

// In addition to the "forwarders" clause, you can force your name
// server to never initiate queries of its own, but always ask its
// forwarders only, by enabling the following line:
```

```
//
//      forward only;

// If you've got a DNS server around at your upstream provider, enter
// its IP address here, and enable the line below.  This will make you
// benefit from its cache, thus reduce overall DNS traffic in the
Internet.
/*
        forwarders {
                127.0.0.1;
        };
*/
```

Just as the comment says, to benefit from an uplink's cache, `forwarders` can be enabled here. Under normal circumstances, a name server will recursively query the Internet looking at certain name servers until it finds the answer it is looking for. Having this enabled will have it query the uplink's name server (or name server provided) first, taking advantage of its cache. If the uplink name server in question is a heavily trafficked, fast name server, enabling this may be worthwhile.

Warning: `127.0.0.1` will *not* work here. Change this IP address to a name server at your uplink.

```
        /*
         * If there is a firewall between you and name servers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        // query-source address * port 53;

        /*
         * If running in a sandbox, you may have to specify a different
         * location for the dumpfile.
         */
        // dump-file "s/named_dump.db";
};

// Note: the following will be supported in a future release.
/*
host { any; } {
        topology {
                127.0.0.0/8;
        };
};
*/

// Setting up secondaries is way easier and the rough picture for this
// is explained below.
//
// If you enable a local name server, don't forget to enter 127.0.0.1
// into your /etc/resolv.conf so this server will be queried first.
```

```
// Also, make sure to enable it in /etc/rc.conf.

zone "." {
        type hint;
        file "named.root";
};

zone "0.0.127.IN-ADDR.ARPA" {
        type master;
        file "localhost.rev";
};

// NB: Do not use the IP addresses below, they are faked, and only
// serve demonstration/documentation purposes!
//
// Example secondary config entries.  It can be convenient to become
// a secondary at least for the zone where your own domain is in.  Ask
// your network administrator for the IP address of the responsible
// primary.
//
// Never forget to include the reverse lookup (IN-ADDR.ARPA) zone!
// (This is the first bytes of the respective IP address, in reverse
// order, with ".IN-ADDR.ARPA" appended.)
//
// Before starting to setup a primary zone, better make sure you fully
// understand how DNS and BIND works, however.  There are sometimes
// unobvious pitfalls.  Setting up a secondary is comparably simpler.
//
// NB: Don't blindly enable the examples below. :-)  Use actual names
// and addresses instead.
//
// NOTE!!! FreeBSD runs BIND in a sandbox (see named_flags in rc.conf).
// The directory containing the secondary zones must be write accessible
// to BIND.  The following sequence is suggested:
//
//      mkdir /etc/namedb/s
//      chown bind:bind /etc/namedb/s
//      chmod 750 /etc/namedb/s
```

For more information on running BIND in a sandbox, see Running named in a sandbox.

```
/*
zone "example.com" {
        type slave;
        file "s/example.com.bak";
        masters {
                192.168.1.1;
        };
};

zone "0.168.192.in-addr.arpa" {
        type slave;
        file "s/0.168.192.in-addr.arpa.bak";
```

```
        masters {
                192.168.1.1;
        };
};
*/
```

In `named.conf`, these are examples of slave entries for a forward and reverse zone.

For each new zone served, a new zone entry must be added to `named.conf`.

For example, the simplest zone entry for `example.org` can look like:

```
zone "example.org" {
 type master;
 file "example.org";
};
```

The zone is a master, as indicated by the `type` statement, holding its zone information in `/etc/namedb/example.org` indicated by the `file` statement.

```
zone "example.org" {
 type slave;
 file "example.org";
};
```

In the slave case, the zone information is transferred from the master name server for the particular zone, and saved in the file specified. If and when the master server dies or is unreachable, the slave name server will have the transferred zone information and will be able to serve it.

27.6.6.3 Zone Files

An example master zone file for `example.org` (existing within `/etc/namedb/example.org`) is as follows:

```
$TTL 3600

example.org. IN SOA ns1.example.org. admin.example.org. (
                        5               ; Serial
                        10800           ; Refresh
                        3600            ; Retry
                        604800          ; Expire
                        86400 )         ; Minimum TTL

; DNS Servers
@       IN NS           ns1.example.org.
@       IN NS           ns2.example.org.

; Machine Names
localhost       IN A    127.0.0.1
ns1             IN A    3.2.1.2
ns2             IN A    3.2.1.3
mail            IN A    3.2.1.10
@               IN A    3.2.1.30
```

```
; Aliases
www             IN CNAME        @

; MX Record
@               IN MX   10      mail.example.org.
```

Note that every hostname ending in a "." is an exact hostname, whereas everything without a trailing "." is referenced to the origin. For example, `www` is translated into `www.`*`origin`*. In our fictitious zone file, our origin is `example.org.`, so `www` would translate to `www.example.org.`

The format of a zone file follows:

```
recordname      IN recordtype   value
```

The most commonly used DNS records:

SOA

: start of zone authority

NS

: an authoritative name server

A

: a host address

CNAME

: the canonical name for an alias

MX

: mail exchanger

PTR

: a domain name pointer (used in reverse DNS)

```
example.org. IN SOA ns1.example.org. admin.example.org. (
                        5               ; Serial
                        10800           ; Refresh after 3 hours
                        3600            ; Retry after 1 hour
                        604800          ; Expire after 1 week
                        86400 )         ; Minimum TTL of 1 day
```

`example.org.`

: the domain name, also the origin for this zone file.

`ns1.example.org.`

: the primary/authoritative name server for this zone.

`admin.example.org.`

the responsible person for this zone, email address with "@" replaced. (<admin@example.org> becomes `admin.example.org`)

`5`

the serial number of the file. This must be incremented each time the zone file is modified. Nowadays, many admins prefer a `yyyymmddrr` format for the serial number. `2001041002` would mean last modified 04/10/2001, the latter `02` being the second time the zone file has been modified this day. The serial number is important as it alerts slave name servers for a zone when it is updated.

```
@       IN NS           ns1.example.org.
```

This is an NS entry. Every name server that is going to reply authoritatively for the zone must have one of these entries. The `@` as seen here could have been `example.org.` The `@` translates to the origin.

```
localhost       IN A    127.0.0.1
ns1             IN A    3.2.1.2
ns2             IN A    3.2.1.3
mail            IN A    3.2.1.10
@               IN A    3.2.1.30
```

The A record indicates machine names. As seen above, `ns1.example.org` would resolve to `3.2.1.2`. Again, the origin symbol, `@`, is used here, thus meaning `example.org` would resolve to `3.2.1.30`.

```
www             IN CNAME        @
```

The canonical name record is usually used for giving aliases to a machine. In the example, `www` is aliased to the machine addressed to the origin, or `example.org` (`3.2.1.30`). CNAMEs can be used to provide alias hostnames, or round robin one hostname among multiple machines.

```
@               IN MX   10      mail.example.org.
```

The MX record indicates which mail servers are responsible for handling incoming mail for the zone. `mail.example.org` is the hostname of the mail server, and 10 being the priority of that mail server.

One can have several mail servers, with priorities of 3, 2, 1. A mail server attempting to deliver to `example.org` would first try the highest priority MX, then the second highest, etc, until the mail can be properly delivered.

For in-addr.arpa zone files (reverse DNS), the same format is used, except with PTR entries instead of A or CNAME.

```
$TTL 3600

1.2.3.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (
                        5               ; Serial
                        10800           ; Refresh
                        3600            ; Retry
                        604800          ; Expire
                        3600 )          ; Minimum

@       IN NS   ns1.example.org.
@       IN NS   ns2.example.org.

2       IN PTR  ns1.example.org.
```

```
3       IN PTR  ns2.example.org.
10      IN PTR  mail.example.org.
30      IN PTR  example.org.
```

This file gives the proper IP address to hostname mappings of our above fictitious domain.

27.6.7 Caching Name Server

A caching name server is a name server that is not authoritative for any zones. It simply asks queries of its own, and remembers them for later use. To set one up, just configure the name server as usual, omitting any inclusions of zones.

27.6.8 Running named in a Sandbox

For added security you may want to run named(8) as an unprivileged user, and configure it to chroot(8) into a sandbox directory. This makes everything outside of the sandbox inaccessible to the **named** daemon. Should **named** be compromised, this will help to reduce the damage that can be caused. By default, FreeBSD has a user and a group called `bind`, intended for this use.

> **Note:** Various people would recommend that instead of configuring **named** to `chroot`, you should run **named** inside a jail(8). This section does not attempt to cover this situation.

Since **named** will not be able to access anything outside of the sandbox (such as shared libraries, log sockets, and so on), there are a number of steps that need to be followed in order to allow **named** to function correctly. In the following checklist, it is assumed that the path to the sandbox is `/etc/namedb` and that you have made no prior modifications to the contents of this directory. Perform the following steps as `root`:

- Create all directories that **named** expects to see:

```
# cd /etc/namedb
# mkdir -p bin dev etc var/tmp var/run master slave
# chown bind:bind slave var/*❶
```

❶ **named** only needs write access to these directories, so that is all we give it.

- Rearrange and create basic zone and configuration files:

```
# cp /etc/localtime etc❶
# mv named.conf etc && ln -sf etc/named.conf
# mv named.root master
# sh make-localhost
# cat > master/named.localhost
$ORIGIN localhost.
$TTL 6h
@ IN SOA localhost. postmaster.localhost. (
   1 ; serial
   3600 ; refresh
```

```
   1800 ; retry
   604800 ; expiration
   3600 ) ; minimum
 IN NS localhost.
 IN A  127.0.0.1
^D
```

❶ This allows **named** to log the correct time to syslogd(8).

.

If you are running a version of FreeBSD prior to 4.9-RELEASE, build a statically linked copy of **named-xfer**, and copy it into the sandbox:

```
# cd /usr/src/lib/libisc
# make cleandir && make cleandir && make depend && make all
# cd /usr/src/lib/libbind
# make cleandir && make cleandir && make depend && make all
# cd /usr/src/libexec/named-xfer
# make cleandir && make cleandir && make depend && make NOSHARED=yes all
# cp named-xfer /etc/namedb/bin && chmod 555 /etc/namedb/bin/named-xfer❶
```

After your statically linked `named-xfer` is installed some cleaning up is required, to avoid leaving stale copies of libraries or programs in your source tree:

```
# cd /usr/src/lib/libisc
# make cleandir
# cd /usr/src/lib/libbind
# make cleandir
# cd /usr/src/libexec/named-xfer
# make cleandir
```

❶ This step has been reported to fail occasionally. If this happens to you, then issue the command:

```
# cd /usr/src && make cleandir && make cleandir
```

and delete your `/usr/obj` tree:

```
# rm -fr /usr/obj && mkdir /usr/obj
```

This will clean out any "cruft" from your source tree, and retrying the steps above should then work.

If you are running FreeBSD version 4.9-RELEASE or later, then the copy of `named-xfer` in `/usr/libexec` is statically linked by default, and you can simply use cp(1) to copy it into your sandbox.

· Make a `dev/null` that **named** can see and write to:

```
# cd /etc/namedb/dev && mknod null c 2 2
# chmod 666 null
```

· Symlink `/var/run/ndc` to `/etc/namedb/var/run/ndc`:

```
# ln -sf /etc/namedb/var/run/ndc /var/run/ndc
```

Note: This simply avoids having to specify the `-c` option to ndc(8) every time you run it. Since the contents of `/var/run` are deleted on boot, it may be useful to add this command to `root`'s crontab(5), using the `@reboot` option.

.

Configure syslogd(8) to create an extra `log` socket that **named** can write to. To do this, add `-l /etc/namedb/dev/log` to the `syslogd_flags` variable in `/etc/rc.conf`.

.

Arrange to have **named** start and `chroot` itself to the sandbox by adding the following to `/etc/rc.conf`:

```
named_enable="YES"
named_flags="-u bind -g bind -t /etc/namedb /etc/named.conf"
```

> **Note:** Note that the configuration file */etc/named.conf* is denoted by a full pathname *relative to the sandbox*, i.e. in the line above, the file referred to is actually `/etc/namedb/etc/named.conf`.

The next step is to edit `/etc/namedb/etc/named.conf` so that **named** knows which zones to load and where to find them on the disk. There follows a commented example (anything not specifically commented here is no different from the setup for a DNS server not running in a sandbox):

```
options {
        directory "/";❶
        named-xfer "/bin/named-xfer";❷
        version "";  // Don't reveal BIND version
        query-source address * port 53;
};
// ndc control socket
controls {
        unix "/var/run/ndc" perm 0600 owner 0 group 0;
};
// Zones follow:
zone "localhost" IN {
        type master;
        file "master/named.localhost";❸
        allow-transfer { localhost; };
        notify no;
};
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "master/localhost.rev";
        allow-transfer { localhost; };
        notify no;
};
zone "." IN {
        type hint;
        file "master/named.root";
};
zone "private.example.net" in {
        type master;
        file "master/private.example.net.db";
 allow-transfer { 192.168.10.0/24; };
```

```
};
zone "10.168.192.in-addr.arpa" in {
        type slave;
        masters { 192.168.10.2; };
        file "slave/192.168.10.db";❹
};
```

❶ The `directory` statement is specified as `/`, since all files that **named** needs are within this directory (recall that this is equivalent to a "normal" user's `/etc/namedb`).

❷ Specifies the full path to the `named-xfer` binary (from **named**'s frame of reference). This is necessary since **named** is compiled to look for `named-xfer` in `/usr/libexec` by default.

❸ Specifies the filename (relative to the `directory` statement above) where **named** can find the zone file for this zone.

❹ Specifies the filename (relative to the `directory` statement above) where **named** should write a copy of the zone file for this zone after successfully transferring it from the master server. This is why we needed to change the ownership of the directory `slave` to `bind` in the setup stages above.

After completing the steps above, either reboot your server or restart syslogd(8) and start named(8), making sure to use the new options specified in `syslogd_flags` and `named_flags`. You should now be running a sandboxed copy of **named**!

27.6.9 Security

Although BIND is the most common implementation of DNS, there is always the issue of security. Possible and exploitable security holes are sometimes found.

It is a good idea to read CERT (http://www.cert.org/)'s security advisories and to subscribe to the FreeBSD security notifications 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications) to stay up to date with the current Internet and FreeBSD security issues.

> **Tip:** If a problem arises, keeping sources up to date and having a fresh build of **named** would not hurt.

27.6.10 Further Reading

BIND/**named** manual pages: ndc(8) named(8) named.conf(5)

- Official ISC BIND Page (http://www.isc.org/products/BIND/)
- BIND FAQ (http://www.nominum.com/getOpenSourceResource.php?id=6)
- O'Reilly DNS and BIND 4th Edition (http://www.oreilly.com/catalog/dns4/)
- RFC1034 - Domain Names - Concepts and Facilities (ftp://ftp.isi.edu/in-notes/rfc1034.txt)
- RFC1035 - Domain Names - Implementation and Specification (ftp://ftp.isi.edu/in-notes/rfc1035.txt)

27.7 BIND9 and FreeBSD

Written by Tom Rhodes.

The release of FreeBSD 5.3 brought the BIND9 DNS server software into the distribution. New security features, a new file system layout and automated chroot(8) configuration came with the import. This section has been written in two parts, the first will discuss new features and their configuration; the latter will cover upgrades to aid in move to FreeBSD 5.3. From this moment on, the server will be referred to simply as named(8) in place of BIND. This section skips over the terminology described in the previous section as well as some of the theoretical discussions; thus, it is recommended that the previous section be consulted before reading any further here.

Configuration files for **named** currently reside in `/var/named/etc/namedb/` and will need modification before use. This is where most of the configuration will be performed.

27.7.1 Configuration of a Master Zone

To configure a master zone visit `/var/named/etc/namedb/` and run the following command:

```
# sh make-localhost
```

If all went well a new file should exist in the `master` directory. The filenames should be `localhost.rev` for the local domain name and `localhost-v6.rev` for IPv6 configurations. As the default configuration file, configuration for its use will already be present in the `named.conf` file.

27.7.2 Configuration of a Slave Zone

Configuration for extra domains or sub domains may be done properly by setting them as a slave zone. In most cases, the `master/localhost.rev` file could just be copied over into the `slave` directory and modified. Once completed, the files need to be properly added in `named.conf` such as in the following configuration for `example.com`:

```
zone "example.com" {
        type slave;
        file "slave/example.com";
        masters {
                10.0.0.1;
        };
};

zone "0.168.192.in-addr.arpa" {
        type slave;
        file "slave/0.168.192.in-addr.arpa";
        masters {
                10.0.0.1;
        };
};
```

Note well that in this example, the master IP address is the primary domain server from which the zones are transferred; it does not necessary serve as DNS server itself.

27.7.3 System Initialization Configuration

In order for the **named** daemon to start when the system is booted, the following option must be present in the `rc.conf` file:

```
named_enable="YES"
```

While other options exist, this is the bare minimal requirement. Consult the rc.conf(5) manual page for a list of the other options. If nothing is entered in the `rc.conf` file then **named** may be started on the command line by invoking:

```
# /etc/rc.d/named start
```

27.7.4 BIND9 Security

While FreeBSD automatically drops **named** into a chroot(8) environment; there are several other security mechanisms in place which could help to lure off possible DNS service attacks.

27.7.4.1 Query Access Control Lists

A query access control list can be used to restrict queries against the zones. The configuration works by defining the network inside of the `acl` token and then listing IP addresses in the zone configuration. To permit domains to query the example host, just define it like this:

```
acl "example.com" {
        192.168.0.0/24;
};

zone "example.com" {
        type slave;
        file "slave/example.com";
        masters {
                10.0.0.1;
        };
 allow-query { example.com; };
};

zone "0.168.192.in-addr.arpa" {
        type slave;
        file "slave/0.168.192.in-addr.arpa";
        masters {
                10.0.0.1;
        };
 allow-query { example.com; };
};
```

27.7.4.2 Restrict Version

Permitting version lookups on the DNS server could be opening the doors for an attacker. A malicious user may use this information to hunt up known exploits or bugs to utilize against the host.

Warning: Setting a false version will not protect the server from exploits. Only upgrading to a version that is not vulnerable will protect your server.

A false version string can be placed the `options` section of `named.conf`:

```
options {
        directory       "/etc/namedb";
        pid-file        "/var/run/named/pid";
        dump-file       "/var/dump/named_dump.db";
        statistics-file "/var/stats/named.stats";
        version  "None of your business";
};
```

27.8 Apache HTTP Server

Contributed by Murray Stokely.

27.8.1 Overview

FreeBSD is used to run some of the busiest web sites in the world. The majority of web servers on the Internet are using the **Apache HTTP Server**. **Apache** software packages should be included on your FreeBSD installation media. If you did not install **Apache** when you first installed FreeBSD, then you can install it from the `www/apache13` or `www/apache20` port.

Once **Apache** has been installed successfully, it must be configured.

Note: This section covers version 1.3.X of the **Apache HTTP Server** as that is the most widely used version for FreeBSD. **Apache** 2.X introduces many new technologies but they are not discussed here. For more information about **Apache** 2.X, please see http://httpd.apache.org/.

27.8.2 Configuration

The main **Apache HTTP Server** configuration file is installed as `/usr/local/etc/apache/httpd.conf` on FreeBSD. This file is a typical UNIX text configuration file with comment lines beginning with the # character. A comprehensive description of all possible configuration options is outside the scope of this book, so only the most frequently modified directives will be described here.

`ServerRoot "/usr/local"`

This specifies the default directory hierarchy for the **Apache** installation. Binaries are stored in the `bin` and `sbin` subdirectories of the server root, and configuration files are stored in `etc/apache`.

`ServerAdmin you@your.address`

The address to which problems with the server should be emailed. This address appears on some server-generated pages, such as error documents.

`ServerName www.example.com`

`ServerName` allows you to set a host name which is sent back to clients for your server if it is different to the one that the host is configured with (i.e., use `www` instead of the host's real name).

`DocumentRoot "/usr/local/www/data"`

`DocumentRoot`: The directory out of which you will serve your documents. By default, all requests are taken from this directory, but symbolic links and aliases may be used to point to other locations.

It is always a good idea to make backup copies of your **Apache** configuration file before making changes. Once you are satisfied with your initial configuration you are ready to start running **Apache**.

27.8.3 Running Apache

Apache does not run from the **inetd** super server as many other network servers do. It is configured to run standalone for better performance for incoming HTTP requests from client web browsers. A shell script wrapper is included to make starting, stopping, and restarting the server as simple as possible. To start up **Apache** for the first time, just run:

```
# /usr/local/sbin/apachectl start
```

You can stop the server at any time by typing:

```
# /usr/local/sbin/apachectl stop
```

After making changes to the configuration file for any reason, you will need to restart the server:

```
# /usr/local/sbin/apachectl restart
```

To restart **Apache** without aborting current connections, run:

```
# /usr/local/sbin/apachectl graceful
```

Additional information available at apachectl(8) manual page.

To launch **Apache** at system startup, add the following line to `/etc/rc.conf`:

```
apache_enable="YES"
```

If you would like to supply additional command line options for the **Apache** `httpd` program started at system boot, you may specify them with an additional line in `rc.conf`:

```
apache_flags=""
```

Now that the web server is running, you can view your web site by pointing a web browser to `http://localhost/`. The default web page that is displayed is `/usr/local/www/data/index.html`.

27.8.4 Virtual Hosting

Apache supports two different types of Virtual Hosting. The first method is Name-based Virtual Hosting. Name-based virtual hosting uses the clients HTTP/1.1 headers to figure out the hostname. This allows many different domains to share the same IP address.

To setup **Apache** to use Name-based Virtual Hosting add an entry like the following to your `httpd.conf`:

```
NameVirtualHost *
```

If your webserver was named `www.domain.tld` and you wanted to setup a virtual domain for `www.someotherdomain.tld` then you would add the following entries to `httpd.conf`:

```
<VirtualHost *>
ServerName www.domain.tld
DocumentRoot /www/domain.tld
</VirtualHost>

<VirtualHost *>
ServerName www.someotherdomain.tld
DocumentRoot /www/someotherdomain.tld
</VirtualHost>
```

Replace the addresses with the addresses you want to use and the path to the documents with what you are using.

For more information about setting up virtual hosts, please consult the official **Apache** documentation at: http://httpd.apache.org/docs/vhosts/.

27.8.5 Apache Modules

There are many different **Apache** modules available to add functionality to the basic server. The FreeBSD Ports Collection provides an easy way to install **Apache** together with some of the more popular add-on modules.

27.8.5.1 mod_ssl

The **mod_ssl** module uses the OpenSSL library to provide strong cryptography via the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. This module provides everything necessary to request a signed certificate from a trusted certificate signing authority so that you can run a secure web server on FreeBSD.

If you have not yet installed **Apache**, then a version of **Apache** 1.3.X that includes **mod_ssl** may be installed with the `www/apache13-modssl` port. SSL support is also available for **Apache** 2.X in the `www/apache20` port, where it is enabled by default.

27.8.5.2 Dynamic Websites with Perl & PHP

In the past few years, more businesses have turned to the Internet in order to enhance their revenue and increase exposure. This has also increased the need for interactive web content. While some companies, such as Microsoft, have introduced solutions into their proprietary products, the open source community answered the call. Two options for dynamic web content include mod_perl & mod_php.

27.8.5.2.1 mod_perl

The **Apache**/Perl integration project brings together the full power of the Perl programming language and the **Apache HTTP Server**. With the **mod_perl** module it is possible to write **Apache** modules entirely in Perl. In addition, the persistent interpreter embedded in the server avoids the overhead of starting an external interpreter and the penalty of Perl start-up time.

mod_perl is available a few different ways. To use **mod_perl** remember that **mod_perl** 1.0 only works with **Apache** 1.3 and **mod_perl** 2.0 only works with **Apache** 2. **mod_perl** 1.0 is available in `www/mod_perl` and a statically compiled version is available in `www/apache13-modperl`. **mod_perl** 2.0 is avaliable in `www/mod_perl2`.

27.8.5.2.2 mod_php

Written by Tom Rhodes.

PHP, also known as "PHP: Hypertext Preprocessor" is a general-purpose scripting language that is especially suited for Web development. Capable of being embedded into HTML its syntax draws upon C, Java, and Perl with the intention of allowing web developers to write dynamically generated webpages quickly.

To gain support for PHP5 for the **Apache** web server, begin by installing the `www/mod_php5` port.

This will install and configure the modules required to support dynamic PHP applications. Check to ensure the following lines have been added to `/usr/local/etc/apache/httpd.conf`:

```
LoadModule php5_module        libexec/apache/libphp5.so
AddModule mod_php5.c
    <IfModule mod_php5.c>
        DirectoryIndex index.php index.html
    </IfModule>

    <IfModule mod_php5.c>
        AddType application/x-httpd-php .php
        AddType application/x-httpd-php-source .phps
    </IfModule>
```

Once completed, a simple call to the `apachectl` command for a graceful restart is needed to load the PHP module:

```
# apachectl graceful
```

The PHP support in FreeBSD is extremely modular so the base install is very limited. It is very easy to add support using the `lang/php5-extensions` port. This port provides a menu driven interface to PHP extension installation. Alternatively, individual extensions can be installed using the appropriate port.

For instance, to add support for the **MySQL** database server to PHP5, simply install the `databases/php5-mysql` port.

After installing an extension, the **Apache** server must be reloaded to pick up the new configuration changes.

```
# apachectl graceful
```

27.9 File Transfer Protocol (FTP)

Contributed by Murray Stokely.

27.9.1 Overview

The File Transfer Protocol (FTP) provides users with a simple way to transfer files to and from an FTP server. FreeBSD includes FTP server software, **ftpd**, in the base system. This makes setting up and administering an FTP server on FreeBSD very straightforward.

27.9.2 Configuration

The most important configuration step is deciding which accounts will be allowed access to the FTP server. A normal FreeBSD system has a number of system accounts used for various daemons, but unknown users should not be allowed to log in with these accounts. The `/etc/ftpusers` file is a list of users disallowed any FTP access. By default, it includes the aforementioned system accounts, but it is possible to add specific users here that should not be allowed access to FTP.

You may want to restrict the access of some users without preventing them completely from using FTP. This can be accomplished with the `/etc/ftpchroot` file. This file lists users and groups subject to FTP access restrictions. The ftpchroot(5) manual page has all of the details so it will not be described in detail here.

If you would like to enable anonymous FTP access to your server, then you must create a user named `ftp` on your FreeBSD system. Users will then be able to log on to your FTP server with a username of `ftp` or `anonymous` and with any password (by convention an email address for the user should be used as the password). The FTP server will call chroot(2) when an anonymous user logs in, to restrict access to only the home directory of the `ftp` user.

There are two text files that specify welcome messages to be displayed to FTP clients. The contents of the file `/etc/ftpwelcome` will be displayed to users before they reach the login prompt. After a successful login, the contents of the file `/etc/ftpmotd` will be displayed. Note that the path to this file is relative to the login environment, so the file `~ftp/etc/ftpmotd` would be displayed for anonymous users.

Once the FTP server has been configured properly, it must be enabled in `/etc/inetd.conf`. All that is required here is to remove the comment symbol “#” from in front of the existing **ftpd** line :

```
ftp stream tcp nowait root /usr/libexec/ftpd ftpd -l
```

As explained in Example 27-1, a HangUP Signal must be sent to **inetd** after this configuration file is changed.

You can now log on to your FTP server by typing:

```
% ftp localhost
```

27.9.3 Maintaining

The **ftpd** daemon uses syslog(3) to log messages. By default, the system log daemon will put messages related to FTP in the `/var/log/xferlog` file. The location of the FTP log can be modified by changing the following line in `/etc/syslog.conf`:

```
ftp.info      /var/log/xferlog
```

Be aware of the potential problems involved with running an anonymous FTP server. In particular, you should think twice about allowing anonymous users to upload files. You may find that your FTP site becomes a forum for the trade of unlicensed commercial software or worse. If you do need to allow anonymous FTP uploads, then you should set up the permissions so that these files can not be read by other anonymous users until they have been reviewed.

27.10 File and Print Services for Microsoft Windows clients (Samba)

Contributed by Murray Stokely.

27.10.1 Overview

Samba is a popular open source software package that provides file and print services for Microsoft Windows clients. Such clients can connect to and use FreeBSD filespace as if it was a local disk drive, or FreeBSD printers as if they were local printers.

Samba software packages should be included on your FreeBSD installation media. If you did not install **Samba** when you first installed FreeBSD, then you can install it from the `net/samba3` port or package.

27.10.2 Configuration

A default **Samba** configuration file is installed as `/usr/local/etc/smb.conf.default`. This file must be copied to `/usr/local/etc/smb.conf` and customized before **Samba** can be used.

The `smb.conf` file contains runtime configuration information for **Samba**, such as definitions of the printers and "file system shares" that you would like to share with Windows clients. The **Samba** package includes a web based tool called **swat** which provides a simple way of configuring the `smb.conf` file.

27.10.2.1 Using the Samba Web Administration Tool (SWAT)

The Samba Web Administration Tool (SWAT) runs as a daemon from **inetd**. Therefore, the following line in `/etc/inetd.conf` should be uncommented before **swat** can be used to configure **Samba**:

```
swat   stream  tcp     nowait/400      root    /usr/local/sbin/swat
```

As explained in Example 27-1, a HangUP Signal must be sent to **inetd** after this configuration file is changed.

Once **swat** has been enabled in `inetd.conf`, you can use a browser to connect to http://localhost:901. You will first have to log on with the system `root` account.

Once you have successfully logged on to the main **Samba** configuration page, you can browse the system documentation, or begin by clicking on the Globals tab. The Globals section corresponds to the variables that are set in the `[global]` section of `/usr/local/etc/smb.conf`.

27.10.2.2 Global Settings

Whether you are using **swat** or editing `/usr/local/etc/smb.conf` directly, the first directives you are likely to encounter when configuring **Samba** are:

`workgroup`

NT Domain-Name or Workgroup-Name for the computers that will be accessing this server.

`netbios name`

This sets the NetBIOS name by which a **Samba** server is known. By default it is the same as the first component of the host's DNS name.

`server string`

This sets the string that will be displayed with the `net view` command and some other networking tools that seek to display descriptive text about the server.

27.10.2.3 Security Settings

Two of the most important settings in `/usr/local/etc/smb.conf` are the security model chosen, and the backend password format for client users. The following directives control these options:

`security`

The two most common options here are `security = share` and `security = user`. If your clients use usernames that are the same as their usernames on your FreeBSD machine then you will want to use user level security. This is the default security policy and it requires clients to first log on before they can access shared resources.

In share level security, client do not need to log onto the server with a valid username and password before attempting to connect to a shared resource. This was the default security model for older versions of **Samba**.

`passdb backend`

Samba has several different backend authentication models. You can authenticate clients with LDAP, NIS+, a SQL database, or a modified password file. The default authentication method is `smbpasswd`, and that is all that will be covered here.

Assuming that the default `smbpasswd` backend is used, the `/usr/local/private/smbpasswd` file must be created to allow **Samba** to authenticate clients. If you would like to give all of your UNIX user accounts access from Windows clients, use the following command:

```
# grep -v "^#" /etc/passwd | make_smbpasswd > /usr/local/private/smbpasswd
# chmod 600 /usr/local/private/smbpasswd
```

Please see the **Samba** documentation for additional information about configuration options. With the basics outlined here, you should have everything you need to start running **Samba**.

27.10.3 Starting Samba

To enable **Samba** when your system boots, add the following line to `/etc/rc.conf`:

```
samba_enable="YES"
```

You can then start **Samba** at any time by typing:

```
# /usr/local/etc/rc.d/samba.sh start
Starting SAMBA: removing stale tdbs :
Starting nmbd.
Starting smbd.
```

Samba actually consists of three separate daemons. You should see that both the **nmbd** and **smbd** daemons are started by the `samba.sh` script. If you enabled winbind name resolution services in `smb.conf`, then you will also see that the **winbindd** daemon is started.

You can stop **Samba** at any time by typing :

```
# /usr/local/etc/rc.d/samba.sh stop
```

Samba is a complex software suite with functionality that allows broad integration with Microsoft Windows networks. For more information about functionality beyond the basic installation described here, please see http://www.samba.org.

27.11 Clock Synchronization with NTP

Contributed by Tom Hukins.

27.11.1 Overview

Over time, a computer's clock is prone to drift. The Network Time Protocol (NTP) is one way to ensure your clock stays accurate.

Many Internet services rely on, or greatly benefit from, computers' clocks being accurate. For example, a web server may receive requests to send a file if it has been modified since a certain time. In a local area network environment, it is essential that computers sharing files from the same file server have synchronized clocks so that file timestamps stay consistent. Services such as cron(8) also rely on an accurate system clock to run commands at the specified times.

FreeBSD ships with the ntpd(8) NTP server which can be used to query other NTP servers to set the clock on your machine or provide time services to others.

27.11.2 Choosing Appropriate NTP Servers

In order to synchronize your clock, you will need to find one or more NTP servers to use. Your network administrator or ISP may have set up an NTP server for this purpose——check their documentation to see if this is the case. There is an online list of publicly accessible NTP servers (http://ntp.isc.org/bin/view/Servers/WebHome) which you can use to find an NTP server near to you. Make sure you are aware of the policy for any servers you choose, and ask for permission if required.

Choosing several unconnected NTP servers is a good idea in case one of the servers you are using becomes unreachable or its clock is unreliable. ntpd(8) uses the responses it receives from other servers intelligently——it will favor unreliable servers less than reliable ones.

27.11.3 Configuring Your Machine

27.11.3.1 Basic Configuration

If you only wish to synchronize your clock when the machine boots up, you can use ntpdate(8). This may be appropriate for some desktop machines which are frequently rebooted and only require infrequent synchronization, but most machines should run ntpd(8).

Using ntpdate(8) at boot time is also a good idea for machines that run ntpd(8). The ntpd(8) program changes the clock gradually, whereas ntpdate(8) sets the clock, no matter how great the difference between a machine's current clock setting and the correct time.

To enable ntpdate(8) at boot time, add `ntpdate_enable="YES"` to `/etc/rc.conf`. You will also need to specify all servers you wish to synchronize with and any flags to be passed to ntpdate(8) in `ntpdate_flags`.

27.11.3.2 General Configuration

NTP is configured by the `/etc/ntp.conf` file in the format described in ntp.conf(5). Here is a simple example:

```
server ntplocal.example.com prefer
server timeserver.example.org
server ntp2a.example.net

driftfile /var/db/ntp.drift
```

The `server` option specifies which servers are to be used, with one server listed on each line. If a server is specified with the `prefer` argument, as with `ntplocal.example.com`, that server is preferred over other servers. A response from a preferred server will be discarded if it differs significantly from other servers' responses, otherwise it will be used without any consideration to other responses. The `prefer` argument is normally used for NTP servers that are known to be highly accurate, such as those with special time monitoring hardware.

The `driftfile` option specifies which file is used to store the system clock's frequency offset. The ntpd(8) program uses this to automatically compensate for the clock's natural drift, allowing it to maintain a reasonably correct setting even if it is cut off from all external time sources for a period of time.

The `driftfile` option specifies which file is used to store information about previous responses from the NTP servers you are using. This file contains internal information for NTP. It should not be modified by any other process.

27.11.3.3 Controlling Access to Your Server

By default, your NTP server will be accessible to all hosts on the Internet. The `restrict` option in `/etc/ntp.conf` allows you to control which machines can access your server.

If you want to deny all machines from accessing your NTP server, add the following line to `/etc/ntp.conf`:

```
restrict default ignore
```

If you only want to allow machines within your own network to synchronize their clocks with your server, but ensure they are not allowed to configure the server or used as peers to synchronize against, add

```
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

instead, where `192.168.1.0` is an IP address on your network and `255.255.255.0` is your network's netmask.

`/etc/ntp.conf` can contain multiple `restrict` options. For more details, see the `Access Control Support` subsection of ntp.conf(5).

27.11.4 Running the NTP Server

To ensure the NTP server is started at boot time, add the line `ntpd_enable="YES"` to `/etc/rc.conf`. If you wish to pass additional flags to ntpd(8), edit the `ntpd_flags` parameter in `/etc/rc.conf`.

To start the server without rebooting your machine, run `ntpd` being sure to specify any additional parameters from `ntpd_flags` in `/etc/rc.conf`. For example:

```
# ntpd -p /var/run/ntpd.pid
```

> **Note:** Under FreeBSD 4.X, you have to replace every instance of `ntpd` with `xntpd` in the options above.

27.11.5 Using ntpd with a Temporary Internet Connection

The ntpd(8) program does not need a permanent connection to the Internet to function properly. However, if you have a temporary connection that is configured to dial out on demand, it is a good idea to prevent NTP traffic from triggering a dial out or keeping the connection alive. If you are using user PPP, you can use `filter` directives in `/etc/ppp/ppp.conf`. For example:

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

For more details see the `PACKET FILTERING` section in ppp(8) and the examples in `/usr/share/examples/ppp/`.

> **Note:** Some Internet access providers block low-numbered ports, preventing NTP from functioning since replies never reach your machine.

27.11.6 Further Information

Documentation for the NTP server can be found in `/usr/share/doc/ntp/` in HTML format.

Chapter 28 防火牆

Contributed by Joseph J. Barbish. Converted to SGML and updated by Brad Davis.

28.1 概述

防火牆能夠過濾你的系統中進出的流量。防火牆也能藉由設置一或多組「規則(rules)」來檢查你的網路連結中進出的網路封包(network packets)，並且能允許或阻擋其通過。這些防火牆的規則可以檢查封包中的特徵，這些特徵涵蓋，但不限於某些通訊協定類型、主機位址的來源或目的，以及連接埠(port)的來源及目的。

防火牆能夠大幅地增強主機或是網路的安全性。它也能夠用來執行下列事項：

- 保護或隔離你內部網路的應用程式、服務以及機器，免於被來自Internet 中你不想要的傳輸所影響
- 限制或禁止內部網路對Internet 的存取服務
- 支援「網路位址轉換」(network address translation , NAT)，它可以允許你的內部網路使用private IP 位址並可以共同分享一個單一連線到網際網路上(可同時用單一IP位址或是一組公共網址)

讀完這章之後，你將會知道：

- 如何適當地訂出封包過濾的規則。
- FreeBSD 中內建的防火牆之間的差異。
- 如何使用及設定OpenBSD 的**PF** 防火牆。
- 如何使用及設定**IPFILTER**。
- 如何使用及設定**IPFW**。

在閱讀這章之前，你必須：

- 了解基本的FreeBSD 和Internet 觀念

28.2 淺談防火牆概念

基本上防火牆規則可分為兩種型態，分別為：「exclusive」以及「inclusive」。「exclusive」類似「黑名單」，它先允許所有封包通過，然後違反規則的封包則禁止通過防火牆。相反的，「inclusive」類似「白名單」，它先擋住所有封包通過，然後只允許有符合規則的才可通過防火牆。

整體來說，「inclusive」式防火牆會比「exclusive」式防火牆安全些。因為「inclusive」明顯降低了不必要的風險。

此外，使用「stateful firewall」可讓安全性更嚴密。它會持續記錄通過防火牆開放的連線，並且只允許符合現存或開啓新的連線才能通過防火牆。狀態防火牆的缺點是如果在非常快的速度下開啓許多新連線，就可能會受到阻絕式服務攻擊(DoS, Denial of Service)。在大多數的防火牆方案中，也可以交叉運用「stateful 」及「non-stateful」防火牆的組合，讓該網站的防火牆達到最佳化。

28.3 防火牆相關軟體

在FreeBSD 基本系統中內建有三種不同的防火牆軟體套件。它們分別是*IPFILTER* (也就是IPF)、*IPFIREWALL* (也就是IPFW)，以及 *OpenBSD 的PacketFilter* (即有名的PF)。FreeBSD 也有兩個內建的流量控管套件(基本上是控制頻寬的使用)：altq(4) 以及dummynet(4)。通常我們習慣把Dummynet 與IPFW 一併運用，而ALTQ 則是搭配IPF/PF 一同使用。雖然IPF、IPFW 以及PF 是使用不同的實做方式及規則語法，但是它們都使用規則來控制是否允許資料封包進出你的系統。

FreeBSD 為何會內建許多不同的防火牆軟體套件，這是因為不同人會有不同的需求、偏好，很難說哪一個防火牆軟體套件是最好的。

而筆者偏好IPFILTER 的原因，是因為運用在NAT 環境的時候，它的狀態規則是相對簡單許多的。而且它內建的FTP 代理，也簡化了如何設定安全的對外FTP 服務規則。

正由於所有的防火牆都是以「檢查、控制所選定之封包」的實作，所以，制定防火牆規則的人就更必須了解TCP/IP 如何運作，以及如何控制封包在正常session 的各種作用。更詳盡的說明，請參閱：http://www.ipprimer.com/overview.cfm。

28.4 OpenBSD 封包過濾器(Packet Filter, PF)及ALTQ

在2003 年6 月份，OpenBSD 的防火牆軟體PF 被移植到FreeBSD 中，並且收錄於Ports Collection 內。而2004 年11 月份所發行的FreeBSD 5.3 版也是第一次將PF 整合為基礎系統的一部分。PF是個完備、全功能的防火牆，並且具有選擇性ALTQ(交錯佇列，Alternate Queuing) 的功能。ALTQ提供了「QoS」(Quality of Service)頻寬管制功能，它可以用過濾規則的方式來保障各種不同服務的頻寬。另外，OpenBSD 計劃中已經對PF 的使用指南提供了詳盡的解說，因此在這本手冊中我們不會作重複的贅述，而只介紹概要。

更多關於PF 的資訊可於下列網址查詢：http://pf4freebsd.love2party.net/.

28.4.1 啓用PF

PF 在FreeBSD 5.3 之後的系統中，就可以輕鬆使用kernel 動態模組來載入。在rc.conf 中加入`pf_enable="YES"` 後，系統就會載入PF 的kernel 動態模組。這模組會在建立時也啓用pflog(4) 記錄功能。

> **Note:** 這個模組會假設kernel 內已有`options INET` 和`device bpf`。除非編譯kernel 時已在像是make.conf(5) 設定檔中加入`NOINET6`(FreeBSD 6.0 以後的版本則是`NO_INET6`) 這樣才會避免不打開IPv6 支援，否則pf 模組同時也需要`options INET6`，也就是IPv6 支援。

一旦載入PF 的kernel 模組或是靜態編譯入kernel 內，就可以使用`pfctl` 來啓動或關閉**pf**。

下面這個例子示範如何啓動**pf**：

```
# pfctl -e
```

`pfctl` 是使用**pf** 防火牆的指令。若要了解更詳盡的`pfctl` 運用，請查閱pfctl(8) 線上手冊。

28.4.2 kernel 選項

在編譯FreeBSD kernel 時，並不必完全加入下列的選項來啓用PF。在這裡只是要列出給你參考的一些資訊而已。將PF 編譯入kernel 中，會導致無法使用kernel 的動態載入模組。

設定PF 的kernel 選項範例在kernel 原始碼中的/usr/src/sys/conf/NOTES，轉貼內容如下：

```
device pf
device pflog
device pfsync
```

device pf 是用來啓動「packet filter(封包過濾)」的防火牆支援。

而device pflog，此功能要裝不裝皆可，它會啓動pflog(4)，以bpf(4) 格式來記錄網路流量。pflogd(8) daemon 則是用來紀錄這些訊息，並存在硬碟上。

device pfsync，此功能要裝不裝皆可，它會啓動pfsync(4)，可以用來監控「狀態的改變」。請注意：device pfsync並不是kernel 動態模組，要使用的話，必須要編入自訂的kernel 中才行。

這些設定將會在你編譯及安裝好新kernel 後才會生效。

28.4.3 rc.conf 其他相關的選項

你需要在/etc/rc.conf 中加入下列的設定，以便在系統啓動時啓用PF：

```
pf_enable="YES"                 # 啓用 PF (如果需要的話載入模組)
pf_rules="/etc/pf.conf"         # PF 防火牆規則設定檔
pf_flags=""                     # pfctl 啓動時的附加選項
pflog_enable="YES"              # 啓動 pflogd(8)
pflog_logfile="/var/log/pflog"  # pflogd 儲存記錄檔案的地方
pflog_flags=""                  # pflogd 啓動時附加的選項
```

如果您的防火牆後面有個LAN(區域網路)，並要透過它來轉送封包，就必須要設定下列選項：

```
gateway_enable="YES"            # 啓用 LAN Gateway
```

28.4.4 啓用ALTQ

ALTQ 只有在編入FreeBSD kernel 中才能生效。不是所有的網路卡驅動程式都支援ALTQ。請看altq(4) 線上手冊來了解你使用的FreeBSD 版本中支援驅動程式的清單。下面所列的將會啓用ALTQ 及其他附加功能：

```
options         ALTQ
options         ALTQ_CBQ        # Class Bases Queuing (CBQ)
options         ALTQ_RED        # Random Early Detection (RED)
options         ALTQ_RIO        # RED In/Out
options         ALTQ_HFSC       # Hierarchical Packet Scheduler (HFSC)
options         ALTQ_PRIQ       # Priority Queuing (PRIQ)
options         ALTQ_NOPCC      # Required for SMP build
```

options ALTQ 是啓用ALTQ 主架構。

options ALTQ_CBQ 會啓用「CBQ」(Class Based Queuing)支援。CBQ 允許你divide a connection’s bandwidth into different classes or queues to prioritize traffic based on filter rules.

`options ALTQ_RED` enables Random Early Detection (RED). RED is used to avoid network congestion. RED does this by measuring the length of the queue and comparing it to the minimum and maximum thresholds for the queue. If the queue is over the maximum all new packets will be dropped. True to its name, RED drops packets from different connections randomly.

`options ALTQ_RIO` enables Random Early Detection In and Out.

`options ALTQ_HFSC` enables the Hierarchical Fair Service Curve Packet Scheduler. For more information about HFSC see: http://www-2.cs.cmu.edu/~hzhang/HFSC/main.html.

`options ALTQ_PRIQ` enables Priority Queuing (PRIQ). PRIQ will always pass traffic that is in a higher queue first.

`options ALTQ_NOPCC` enables SMP support for ALTQ. This option is required on SMP systems.

28.4.5 Creating Filtering Rules

The Packet Filter reads its configuration rules from the pf.conf(5) file and it modifies, drops or passes packets according to the rules or definitions specified there. The FreeBSD installation comes with a default `/etc/pf.conf` which contains useful examples and explanations.

Although FreeBSD has its own `/etc/pf.conf` the syntax is the same as one used in OpenBSD. A great resource for configuring the **pf** firewall has been written by OpenBSD team and is available at http://www.openbsd.org/faq/pf/.

> **Warning:** When browsing the pf user's guide, please keep in mind that different versions of FreeBSD contain different versions of pf. The **pf** firewall in FreeBSD 5.X is at the level of OpenBSD version 3.5 and in FreeBSD 6.X is at the level of OpenBSD version 3.7.

The FreeBSD packet filter 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf) is a good place to ask questions about configuring and running the **pf** firewall. Do not forget to check the mailing list archives before asking questions.

28.5 IPFILTER (IPF) 防火牆

> **Note:** 此一節的內容仍在陸續補充、更新，所以本節內容可能並未完全符合現況。

IPFILTER 的作者為Darren Reed。IPFILTER 並非得綁某特定作業系統才行：它是個跨OS 平台的open source 應用程式，且已被移植到FreeBSD、NetBSD、OpenBSD、SunOS、HP/UX 以及Solaris 這些作業系統上。此外，IPFILTER 的支援、維護也相當積極，也有定期釋出的更新版。

IPFILTER is based on a kernel-side firewall and NAT mechanism that can be controlled and monitored by userland interface programs. The firewall rules can be set or deleted with the ipf(8) utility. The NAT rules can be set or deleted with the ipnat(1) utility. The ipfstat(8) utility can print run-time statistics for the kernel parts of IPFILTER. The ipmon(8) program can log IPFILTER actions to the system log files.

IPF was originally written using a rule processing logic of 「the last matching rule wins」and used only stateless type of rules. Over time IPF has been enhanced to include a 「quick」option and a stateful 「keep state」option which drastically modernized the rules processing logic. IPF's official documentation covers the legacy rule coding

parameters and the legacy rule file processing logic. The modernized functions are only included as additional options, completely understating their benefits in producing a far superior secure firewall.

The instructions contained in this section are based on using rules that contain the 「quick」option and the stateful 「keep state」option. This is the basic framework for coding an inclusive firewall rule set.

An inclusive firewall only allows packets matching the rules to pass through. This way you can control what services can originate behind the firewall destined for the public Internet and also control the services which can originate from the public Internet accessing your private network. Everything else is blocked and logged by default design. Inclusive firewalls are much, much more secure than exclusive firewall rule sets and is the only rule set type covered herein.

For detailed explanation of the legacy rules processing method see:
http://www.obfuscation.org/ipf/ipf-howto.html#TOC_1 and http://coombs.anu.edu.au/~avalon/ip-filter.html.

IPF 的FAQ 位於http://www.phildev.net/ipf/index.html.

28.5.1 啓用IPF

IPF is included in the basic FreeBSD install as a separate run time loadable module. The system will dynamically load the IPF kernel loadable module when the rc.conf statement `ipfilter_enable="YES"` is used. The loadable module was created with logging enabled and the `default pass all` options. You do not need to compile IPF into the FreeBSD kernel just to change the default to `block all`, you can do that by just coding a block all rule at the end of your rule set.

28.5.2 kernel 選項

在編譯FreeBSD kernel 時，並不必完全加入下列的選項來啓用IPF。在這裡只是要列出給你參考的一些資訊而已。將IPF 編譯入kernel 中，會導致無法使用kernel 的動態載入模組。

Sample kernel config IPF option statements are in the `/usr/src/sys/conf/NOTES` kernel source (`/usr/src/sys/`*`arch`*`/conf/LINT` for FreeBSD 4.X) and are reproduced here:

```
options IPFILTER
options IPFILTER_LOG
options IPFILTER_DEFAULT_BLOCK
```

`options IPFILTER` enables support for the 「IPFILTER」firewall.

`options IPFILTER_LOG` enables the option to have IPF log traffic by writing to the `ipl` packet logging pseudo——device for every rule that has the `log` keyword.

`options IPFILTER_DEFAULT_BLOCK` changes the default behavior so any packet not matching a firewall `pass` rule gets blocked.

These settings will take effect only after you have built and installed a kernel with them set.

28.5.3 可用的rc.conf 選項

須在`/etc/rc.conf` 內加入下列內容，以便在開機時就會啓用IPF：

```
ipfilter_enable="YES"             # Start ipf firewall
ipfilter_rules="/etc/ipf.rules"   # IPF 防火牆規則設定檔
```

```
ipmon_enable="YES"                # 啓用 IP 監控記錄
ipmon_flags="-Ds"                 # D = 使用服務程序 (daemon) 啓動
                                  # s = 使用 syslog 記錄
                                  # v = 記錄於 tcp window, ack, seq
                                  # n = 將 IP 及 port 對應至名稱中
```

If you have a LAN behind this firewall that uses the reserved private IP address ranges, then you need to add the following to enable NAT functionality:

```
gateway_enable="YES"              # 啓用 LAN Gateway
ipnat_enable="YES"                # Start ipnat function
ipnat_rules="/etc/ipnat.rules"    # rules definition file for ipnat
```

28.5.4 IPF

The ipf command is used to load your rules file. Normally you create a file containing your custom rules and use this command to replace in mass the currently running firewall internal rules:

```
# ipf -Fa -f /etc/ipf.rules
```

`-Fa` means flush all internal rules tables.

`-f` means this is the file to read for the rules to load.

This gives you the ability to make changes to your custom rules file, run the above IPF command, and thus update the running firewall with a fresh copy of all the rules without having to reboot the system. This method is very convenient for testing new rules as the procedure can be executed as many times as needed.

See the ipf(8) manual page for details on the other flags available with this command.

The ipf(8) command expects the rules file to be a standard text file. It will not accept a rules file written as a script with symbolic substitution.

There is a way to build IPF rules that utilizes the power of script symbolic substitution. For more information, see Section 28.5.9.

28.5.5 IPFSTAT

The default behavior of ipfstat(8) is to retrieve and display the totals of the accumulated statistics gathered as a result of applying the user coded rules against packets going in and out of the firewall since it was last started, or since the last time the accumulators were reset to zero by the `ipf -Z` command.

See the ipfstat(8) manual page for details.

The default ipfstat(8) command output will look something like this:

```
input packets: blocked 99286 passed 1255609 nomatch 14686 counted 0
 output packets: blocked 4200 passed 1284345 nomatch 14687 counted 0
 input packets logged: blocked 99286 passed 0
 output packets logged: blocked 0 passed 0
 packets logged: input 0 output 0
 log failures: input 3898 output 0
 fragment state(in): kept 0 lost 0
```

```
fragment state(out): kept 0 lost 0
packet state(in): kept 169364 lost 0
packet state(out): kept 431395 lost 0
ICMP replies: 0 TCP RSTs sent: 0
Result cache hits(in): 1215208 (out): 1098963
IN Pullups succeeded: 2 failed: 0
OUT Pullups succeeded: 0 failed: 0
Fastroute successes: 0 failures: 0
TCP cksum fails(in): 0 (out): 0
Packet log flags set: (0)
```

When supplied with either `-i` for inbound or `-o` for outbound, it will retrieve and display the appropriate list of filter rules currently installed and in use by the kernel.

`ipfstat -in` displays the inbound internal rules table with rule number.

`ipfstat -on` displays the outbound internal rules table with the rule number.

The output will look something like this:

```
@1 pass out on xl0 from any to any
@2 block out on dc0 from any to any
@3 pass out quick on dc0 proto tcp/udp from any to any keep state
```

`ipfstat -ih` displays the inbound internal rules table, prefixing each rule with a count of how many times the rule was matched.

`ipfstat -oh` displays the outbound internal rules table, prefixing each rule with a count of how many times the rule was matched.

The output will look something like this:

```
2451423 pass out on xl0 from any to any
354727 block out on dc0 from any to any
430918 pass out quick on dc0 proto tcp/udp from any to any keep state
```

One of the most important functions of the `ipfstat` command is the `-t` flag which displays the state table in a way similar to the way top(1) shows the FreeBSD running process table. When your firewall is under attack this function gives you the ability to identify, drill down to, and see the attacking packets. The optional sub-flags give the ability to select the destination or source IP, port, or protocol that you want to monitor in real time. See the ipfstat(8) manual page for details.

28.5.6 IPMON

In order for `ipmon` to work properly, the kernel option IPFILTER_LOG must be turned on. This command has two different modes that it can be used in. Native mode is the default mode when you type the command on the command line without the `-D` flag.

Daemon mode is for when you want to have a continuous system log file available so that you can review logging of past events. This is how FreeBSD and IPFILTER are configured to work together. FreeBSD has a built in facility to automatically rotate system logs. That is why outputting the log information to syslogd is better than the default of outputting to a regular file. In the default `rc.conf` file you see the ipmon_flags statement uses the `-Ds` flags:

```
ipmon_flags="-Ds" # D = start as daemon
```

```
# s = log to syslog
# v = log tcp window, ack, seq
# n = map IP & port to names
```

The benefits of logging are obvious. It provides the ability to review, after the fact, information such as which packets had been dropped, what addresses they came from and where they were going. These all give you a significant edge in tracking down attackers.

Even with the logging facility enabled, IPF will not generate any rule logging on its own. The firewall administrator decides what rules in the rule set he wants to log and adds the log keyword to those rules. Normally only deny rules are logged.

It is very customary to include a default deny everything rule with the log keyword included as your last rule in the rule set. This way you get to see all the packets that did not match any of the rules in the rule set.

28.5.7 IPMON Logging

Syslogd uses its own special method for segregation of log data. It uses special groupings called 「facility」and "level". IPMON in `-Ds` mode uses `security` (`local0` in 4.X) as the 「facility」name. All IPMON logged data goes to `security` (`local0` in 4.X). The following levels can be used to further segregate the logged data if desired:

```
LOG_INFO - packets logged using the "log" keyword as the action rather than pass or block.
LOG_NOTICE - packets logged which are also passed
LOG_WARNING - packets logged which are also blocked
LOG_ERR - packets which have been logged and which can be considered short
```

To setup IPFILTER to log all data to `/var/log/ipfilter.log`, you will need to create the file. The following command will do that:

```
# touch /var/log/ipfilter.log
```

The syslog function is controlled by definition statements in the `/etc/syslog.conf` file. The `syslog.conf` file offers considerable flexibility in how syslog will deal with system messages issued by software applications like IPF.

Add the following statement to `/etc/syslog.conf` for FreeBSD 5.X and later:

```
security.* /var/log/ipfilter.log
```

Or add the following statement to `/etc/syslog.conf` for FreeBSD 4.X:

```
local0.* /var/log/ipfilter.log
```

The `security.*` (`local0` for 4.X) means to write all the logged messages to the coded file location.

To activate the changes to `/etc/syslog.conf` you can reboot or bump the syslog task into re-reading `/etc/syslog.conf` by running `/etc/rc.d/syslogd reload` (`killall -HUP syslogd` in FreeBSD 4.X).

Do not forget to change `/etc/newsyslog.conf` to rotate the new log you just created above.

28.5.8 The Format of Logged Messages

Messages generated by `ipmon` consist of data fields separated by white space. Fields common to all messages are:

1. The date of packet receipt.
2. The time of packet receipt. This is in the form HH:MM:SS.F, for hours, minutes, seconds, and fractions of a second (which can be several digits long).
3. The name of the interface the packet was processed on, e.g. `dc0`.
4. The group and rule number of the rule, e.g. `@0:17`.

These can be viewed with `ipfstat -in`.

1. The action: p for passed, b for blocked, S for a short packet, n did not match any rules, L for a log rule. The order of precedence in showing flags is: S, p, b, n, L. A capital P or B means that the packet has been logged due to a global logging setting, not a particular rule.
2. The addresses. This is actually three fields: the source address and port (separated by a comma), the -> symbol, and the destination address and port. 209.53.17.22,80 -> 198.73.220.17,1722.
3. `PR` followed by the protocol name or number, e.g. PR tcp.
4. `len` followed by the header length and total length of the packet, e.g. len 20 40.

If the packet is a TCP packet, there will be an additional field starting with a hyphen followed by letters corresponding to any flags that were set. See the ipmon(8) manual page for a list of letters and their flags.

If the packet is an ICMP packet, there will be two fields at the end, the first always being 「ICMP」, and the next being the ICMP message and sub-message type, separated by a slash, e.g. ICMP 3/3 for a port unreachable message.

28.5.9 Building the Rule Script with Symbolic Substitution

Some experienced IPF users create a file containing the rules and code them in a manner compatible with running them as a script with symbolic substitution. The major benefit of doing this is that you only have to change the value associated with the symbolic name and when the script is run all the rules containing the symbolic name will have the value substituted in the rules. Being a script, you can use symbolic substitution to code frequently used values and substitute them in multiple rules. You will see this in the following example.

The script syntax used here is compatible with the sh, csh, and tcsh shells.

Symbolic substitution fields are prefixed with a dollar sign: `$`.

Symbolic fields do not have the $ prefix.

The value to populate the symbolic field must be enclosed with double quotes (").

Start your rule file with something like this:

```
############# IPF 規則命令稿的開始  ########################

oif="dc0"            # 對外網路裝置的名稱
odns="192.0.2.11"    # ISP 的 DNS 伺服器 IP 位址
myip="192.0.2.7"     # 從我的 ISP 提供的靜態 IP
ks="keep state"
fks="flags S keep state"

# You can choose between building /etc/ipf.rules file
# from this script or running this script "as is".
#
```

```
# Uncomment only one line and comment out another.
#
# 1) This can be used for building /etc/ipf.rules:
#cat > /etc/ipf.rules << EOF
#
# 2) This can be used to run script "as is":
/sbin/ipf -Fa -f - << EOF

# Allow out access to my ISP's Domain name server.
pass out quick on $oif proto tcp from any to $odns port = 53 $fks
pass out quick on $oif proto udp from any to $odns port = 53 $ks

# Allow out non-secure standard www function
pass out quick on $oif proto tcp from $myip to any port = 80 $fks

# Allow out secure www function https over TLS SSL
pass out quick on $oif proto tcp from $myip to any port = 443 $fks
EOF
################## End of IPF rules script ########################
```

That is all there is to it. The rules are not important in this example; how the symbolic substitution fields are populated and used are. If the above example was in a file named `/etc/ipf.rules.script`, you could reload these rules by entering the following command:

```
# sh /etc/ipf.rules.script
```

There is one problem with using a rules file with embedded symbolics: IPF does not understand symbolic substitution, and cannot read such scripts directly.

This script can be used in one of two ways:

- Uncomment the line that begins with `cat`, and comment out the line that begins with `/sbin/ipf`. Place `ipfilter_enable="YES"` into `/etc/rc.conf` as usual, and run script once after each modification to create or update `/etc/ipf.rules`.
- Disable IPFILTER in system startup scripts by adding `ipfilter_enable="NO"` (this is default value) into `/etc/rc.conf` file.

  Add a script like the following to your `/usr/local/etc/rc.d/` startup directory. The script should have an obvious name like `ipf.loadrules.sh`. The `.sh` extension is mandatory.

  ```
  #!/bin/sh
  sh /etc/ipf.rules.script
  ```

  The permissions on this script file must be read, write, execute for owner `root`.

  ```
  # chmod 700 /usr/local/etc/rc.d/ipf.loadrules.sh
  ```

從現在起，當系統開機時就會載入你所設的IPF 規則。

28.5.10 IPF 規則

A rule set is a group of ipf rules coded to pass or block packets based on the values contained in the packet. The bi-directional exchange of packets between hosts comprises a session conversation. The firewall rule set processes

the packet two times, once on its arrival from the public Internet host and again as it leaves for its return trip back to the public Internet host. Each TCP/IP service (i.e. telnet, www, mail, etc.) is predefined by its protocol, source and destination IP address, or the source and destination port number. This is the basic selection criteria used to create rules which will pass or block services.

IPF was originally written using a rules processing logic of 「the last matching rule wins」 and used only stateless rules. Over time IPF has been enhanced to include a 「quick」 option and a stateful 「keep state」 option which drastically modernized the rule processing logic.

The instructions contained in this section are based on using rules that contain the 「quick」 option and the stateful 「keep state」 option. This is the basic framework for coding an inclusive firewall rule set.

An inclusive firewall only allows services matching the rules through. This way you can control what services can originate behind the firewall destined for the public Internet and also control the services which can originate from the public Internet accessing your private network. Everything else is blocked and logged by default design. Inclusive firewalls are much, much securer than exclusive firewall rule sets and is the only rule set type covered herein.

> **Warning:** When working with the firewall rules, be *very careful*. Some configurations *will lock you out* of the server. To be on the safe side, you may wish to consider performing the initial firewall configuration from the local console rather than doing it remotely e.g. via **ssh**.

28.5.11 Rule Syntax

The rule syntax presented here has been simplified to only address the modern stateful rule context and 「first matching rule wins」 logic. For the complete legacy rule syntax description see the ipf(8) manual page.

A # character is used to mark the start of a comment and may appear at the end of a rule line or on its own line. Blank lines are ignored.

Rules contain keywords. These keywords have to be coded in a specific order from left to right on the line. Keywords are identified in bold type. Some keywords have sub-options which may be keywords themselves and also include more sub-options. Each of the headings in the below syntax has a bold section header which expands on the content.

ACTION IN-OUT OPTIONS SELECTION STATEFUL PROTO SRC_ADDR,DST_ADDR OBJECT PORT_NUM TCP_FLAG STATEFUL

ACTION = block | pass

IN-OUT = in | out

OPTIONS = log | quick | on interface-name

SELECTION = proto value | source/destination IP | port = number | flags flag-value

PROTO = tcp/udp | udp | tcp | icmp

SRC_ADD,DST_ADDR = all | from object to object

OBJECT = IP address | any

PORT_NUM = port number

TCP_FLAG = S

STATEFUL = keep state

28.5.11.1 ACTION

The action indicates what to do with the packet if it matches the rest of the filter rule. Each rule *must* have a action. The following actions are recognized:

`block` indicates that the packet should be dropped if the selection parameters match the packet.

`pass` indicates that the packet should exit the firewall if the selection parameters match the packet.

28.5.11.2 IN-OUT

A mandatory requirement is that each filter rule explicitly state which side of the I/O it is to be used on. The next keyword must be either in or out and one or the other has to be coded or the rule will not pass syntax checks.

`in` means this rule is being applied against an inbound packet which has just been received on the interface facing the public Internet.

`out` means this rule is being applied against an outbound packet destined for the interface facing the public Internet.

28.5.11.3 OPTIONS

> **Note:** These options must be used in the order shown here.

`log` indicates that the packet header will be written to the `ipl` log (as described in the LOGGING section below) if the selection parameters match the packet.

`quick` indicates that if the selection parameters match the packet, this rule will be the last rule checked, allowing a 「short-circuit」 path to avoid processing any following rules for this packet. This option is a mandatory requirement for the modernized rules processing logic.

`on` indicates the interface name to be incorporated into the selection parameters. Interface names are as displayed by ifconfig(8). Using this option, the rule will only match if the packet is going through that interface in the specified direction (in/out). This option is a mandatory requirement for the modernized rules processing logic.

When a packet is logged, the headers of the packet are written to the IPL packet logging pseudo-device. Immediately following the log keyword, the following qualifiers may be used (in this order):

`body` indicates that the first 128 bytes of the packet contents will be logged after the headers.

`first` If the `log` keyword is being used in conjunction with a 「keep state」 option, it is recommended that this option is also applied so that only the triggering packet is logged and not every packet which thereafter matches the 「keep state」 information.

28.5.11.4 SELECTION

The keywords described in this section are used to describe attributes of the packet to be interrogated when determining whether rules match or not. There is a keyword subject, and it has sub-option keywords, one of which has to be selected. The following general-purpose attributes are provided for matching, and must be used in this order:

28.5.11.5 PROTO

`proto` is the subject keyword and must be coded along with one of its corresponding keyword sub-option values. The value allows a specific protocol to be matched against. This option is a mandatory requirement for the modernized rules processing logic.

`tcp/udp | udp | tcp | icmp` or any protocol names found in `/etc/protocols` are recognized and may be used. The special protocol keyword `tcp/udp` may be used to match either a TCP or a UDP packet, and has been added as a convenience to save duplication of otherwise identical rules.

28.5.11.6 SRC_ADDR/DST_ADDR

The `all` keyword is essentially a synonym for 「from any to any」with no other match parameters.

`from src to dst`: the from and to keywords are used to match against IP addresses. Rules must specify BOTH source and destination parameters. `any` is a special keyword that matches any IP address. Examples of use: 「from any to any」or 「from 0.0.0.0/0 to any」or “from any to 0.0.0.0/0」or 「from 0.0.0.0 to any” or 「from any to 0.0.0.0」.

IP addresses may be specified as a dotted IP address numeric form/mask-length, or as single dotted IP address numeric form.

There is no way to match ranges of IP addresses which do not express themselves easily as mask-length. See this web page for help on writing mask-length: http://jodies.de/ipcalc.

28.5.11.7 PORT

If a port match is included, for either or both of source and destination, then it is only applied to TCP and UDP packets. When composing port comparisons, either the service name from `/etc/services` or an integer port number may be used. When the port appears as part of the from object, it matches the source port number; when it appears as part of the to object, it matches the destination port number. The use of the port option with the `to` object is a mandatory requirement for the modernized rules processing logic. Example of use: 「from any to any port = 80」

Port comparisons may be done in a number of forms, with a number of comparison operators, or port ranges may be specified.

port "=" | "!=" | "<" | ">" | "<=" | ">=" | "eq" | "ne" | "lt" | "gt" | "le" | "ge".

To specify port ranges, port "<>" | "><"

> **Warning:** Following the source and destination matching parameters, the following two parameters are mandatory requirements for the modernized rules processing logic.

28.5.11.8 TCP_FLAG

Flags are only effective for TCP filtering. The letters represents one of the possible flags that can be interrogated in the TCP packet header.

The modernized rules processing logic uses the `flags S` parameter to identify the tcp session start request.

28.5.11.9 STATEFUL

`keep state` indicates that on a pass rule, any packets that match the rules selection parameters should activate the stateful filtering facility.

> **Note:** This option is a mandatory requirement for the modernized rules processing logic.

28.5.12 Stateful Filtering

Stateful filtering treats traffic as a bi-directional exchange of packets comprising a session conversation. When activated, keep-state dynamically generates internal rules for each anticipated packet being exchanged during the bi-directional session conversation. It has the interrogation abilities to determine if the session conversation between the originating sender and the destination are following the valid procedure of bi-directional packet exchange. Any packets that do not properly fit the session conversation template are automatically rejected as impostors.

Keep state will also allow ICMP packets related to a TCP or UDP session through. So if you get ICMP type 3 code 4 in response to some web surfing allowed out by a keep state rule, they will be automatically allowed in. Any packet that IPF can be certain is part of an active session, even if it is a different protocol, will be let in.

What happens is:

Packets destined to go out the interface connected to the public Internet are first checked against the dynamic state table, if the packet matches the next expected packet comprising in a active session conversation, then it exits the firewall and the state of the session conversation flow is updated in the dynamic state table, the remaining packets get checked against the outbound rule set.

Packets coming in to the interface connected to the public Internet are first checked against the dynamic state table, if the packet matches the next expected packet comprising a active session conversation, then it exits the firewall and the state of the session conversation flow is updated in the dynamic state table, the remaining packets get checked against the inbound rule set.

When the conversation completes it is removed from the dynamic state table.

Stateful filtering allows you to focus on blocking/passing new sessions. If the new session is passed, all its subsequent packets will be allowed through automatically and any impostors automatically rejected. If a new session is blocked, none of its subsequent packets will be allowed through. Stateful filtering has technically advanced interrogation abilities capable of defending against the flood of different attack methods currently employed by attackers.

28.5.13 Inclusive Rule Set Example

The following rule set is an example of how to code a very secure inclusive type of firewall. An inclusive firewall only allows services matching pass rules through and blocks all other by default. All firewalls have at the minimum two interfaces which have to have rules to allow the firewall to function.

All UNIX flavored systems including FreeBSD are designed to use interface `lo0` and IP address `127.0.0.1` for internal communication within the operating system. The firewall rules must contain rules to allow free unmolested movement of these special internally used packets.

The interface which faces the public Internet is the one where you place your rules to authorize and control access out to the public Internet and access requests arriving from the public Internet. This can be your user PPP `tun0` interface or your NIC that is connected to your DSL or cable modem.

In cases where one or more NICs are cabled to private LANs behind the firewall, those interfaces must have a rule coded to allow free unmolested movement of packets originating from those LAN interfaces.

The rules should be first organized into three major sections: all the free unmolested interfaces, the public interface outbound, and the public interface inbound.

The rules in each of the public interface sections should have the most frequently matched rules placed before less commonly matched rules, with the last rule in the section blocking and logging all packets on that interface and direction.

The Outbound section in the following rule set only contains ’pass’ rules which contain selection values that uniquely identify the service that is authorized for public Internet access. All the rules have the ’quick’, ’on’, ’proto’, ’port’, and ’keep state’ option coded. The ’proto tcp’ rules have the ’flag’ option included to identify the session start request as the triggering packet to activate the stateful facility.

The Inbound section has all the blocking of undesirable packets first, for two different reasons. The first is that these things being blocked may be part of an otherwise valid packet which may be allowed in by the later authorized service rules. The second reason is that by having a rule that explicitly blocks selected packets that I receive on an infrequent basis and that I do not want to see in the log, they will not be caught by the last rule in the section which blocks and logs all packets which have fallen through the rules. The last rule in the section which blocks and logs all packets is how you create the legal evidence needed to prosecute the people who are attacking your system.

Another thing you should take note of, is there is no response returned for any of the undesirable stuff, their packets just get dropped and vanish. This way the attacker has no knowledge if his packets have reached your system. The less the attackers can learn about your system, the more time they must invest before actually doing something bad. The inbound ’nmap OS fingerprint’ attempts rule I log the first occurrence because this is something a attacker would do.

Any time you see log messages on a rule with ’log first’. You should do an `ipfstat -hio` command to see the number of times the rule has been matched so you know if you are being flooded, i.e. under attack.

When you log packets with port numbers you do not recognize, look it up in `/etc/services` or go to http://www.securitystats.com/tools/portsearch.php and do a port number lookup to find what the purpose of that port number is.

Check out this link for port numbers used by Trojans http://www.simovits.com/trojans/trojans.html.

The following rule set is a complete very secure ’inclusive’ type of firewall rule set that I have used on my system. You can not go wrong using this rule set for your own. Just comment out any pass rules for services that you do not want to authorize.

If you see messages in your log that you want to stop seeing just add a block rule in the inbound section.

You have to change the `dc0` interface name in every rule to the interface name of the Nic card that connects your system to the public Internet. For user PPP it would be `tun0`.

Add the following statements to `/etc/ipf.rules`:

```
#####################################################################
# No restrictions on Inside LAN Interface for private network
# Not needed unless you have LAN
#####################################################################
```

```
#pass out quick on xl0 all
#pass in quick on xl0 all

######################################################################
# No restrictions on Loopback Interface
######################################################################
pass in quick on lo0 all
pass out quick on lo0 all

######################################################################
# Interface facing Public Internet (Outbound Section)
# Interrogate session start requests originating from behind the
# firewall on the private network
# or from this gateway server destine for the public Internet.
######################################################################

# Allow out access to my ISP's Domain name server.
# xxx must be the IP address of your ISP's DNS.
# Dup these lines if your ISP has more than one DNS server
# Get the IP addresses from /etc/resolv.conf file
pass out quick on dc0 proto tcp from any to xxx port = 53 flags S keep state
pass out quick on dc0 proto udp from any to xxx port = 53 keep state

# Allow out access to my ISP's DHCP server for cable or DSL networks.
# This rule is not needed for 'user ppp' type connection to the
# public Internet, so you can delete this whole group.
# Use the following rule and check log for IP address.
# Then put IP address in commented out rule & delete first rule
pass out log quick on dc0 proto udp from any to any port = 67 keep state
#pass out quick on dc0 proto udp from any to z.z.z.z port = 67 keep state


# Allow out non-secure standard www function
pass out quick on dc0 proto tcp from any to any port = 80 flags S keep state

# Allow out secure www function https over TLS SSL
pass out quick on dc0 proto tcp from any to any port = 443 flags S keep state

# Allow out send & get email function
pass out quick on dc0 proto tcp from any to any port = 110 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 25 flags S keep state

# Allow out Time
pass out quick on dc0 proto tcp from any to any port = 37 flags S keep state

# Allow out nntp news
pass out quick on dc0 proto tcp from any to any port = 119 flags S keep state

# Allow out gateway & LAN users non-secure FTP ( both passive & active modes)
# This function uses the IPNAT built in FTP proxy function coded in
# the nat rules file to make this single rule function correctly.
# If you want to use the pkg_add command to install application packages
```

```
# on your gateway system you need this rule.
pass out quick on dc0 proto tcp from any to any port = 21 flags S keep state

# Allow out secure FTP, Telnet, and SCP
# This function is using SSH (secure shell)
pass out quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Allow out non-secure Telnet
pass out quick on dc0 proto tcp from any to any port = 23 flags S keep state

# Allow out FBSD CVSUP function
pass out quick on dc0 proto tcp from any to any port = 5999 flags S keep state

# Allow out ping to public Internet
pass out quick on dc0 proto icmp from any to any icmp-type 8 keep state

# Allow out whois for LAN PC to public Internet
pass out quick on dc0 proto tcp from any to any port = 43 flags S keep state

# Block and log only the first occurrence of everything
# else that's trying to get out.
# This rule enforces the block all by default logic.
block out log first quick on dc0 all

######################################################################
# Interface facing Public Internet (Inbound Section)
# Interrogate packets originating from the public Internet
# destine for this gateway server or the private network.
######################################################################

# Block all inbound traffic from non-routable or reserved address spaces
block in quick on dc0 from 192.168.0.0/16 to any    #RFC 1918 private IP
block in quick on dc0 from 172.16.0.0/12 to any     #RFC 1918 private IP
block in quick on dc0 from 10.0.0.0/8 to any        #RFC 1918 private IP
block in quick on dc0 from 127.0.0.0/8 to any       #loopback
block in quick on dc0 from 0.0.0.0/8 to any         #loopback
block in quick on dc0 from 169.254.0.0/16 to any    #DHCP auto-config
block in quick on dc0 from 192.0.2.0/24 to any      #reserved for docs
block in quick on dc0 from 204.152.64.0/23 to any   #Sun cluster interconnect
block in quick on dc0 from 224.0.0.0/3 to any       #Class D & E multicast

##### Block a bunch of different nasty things. ############
# That I do not want to see in the log

# Block frags
block in quick on dc0 all with frags

# Block short tcp packets
block in quick on dc0 proto tcp all with short

# block source routed packets
block in quick on dc0 all with opt lsrr
block in quick on dc0 all with opt ssrr
```

```
# Block nmap OS fingerprint attempts
# Log first occurrence of these so I can get their IP address
block in log first quick on dc0 proto tcp from any to any flags FUP

# Block anything with special options
block in quick on dc0 all with ipopts

# Block public pings
block in quick on dc0 proto icmp all icmp-type 8

# Block ident
block in quick on dc0 proto tcp from any to any port = 113

# Block all Netbios service. 137=name, 138=datagram, 139=session
# Netbios is MS/Windows sharing services.
# Block MS/Windows hosts2 name server requests 81
block in log first quick on dc0 proto tcp/udp from any to any port = 137
block in log first quick on dc0 proto tcp/udp from any to any port = 138
block in log first quick on dc0 proto tcp/udp from any to any port = 139
block in log first quick on dc0 proto tcp/udp from any to any port = 81

# Allow traffic in from ISP’s DHCP server. This rule must contain
# the IP address of your ISP’s DHCP server as it’s the only
# authorized source to send this packet type. Only necessary for
# cable or DSL configurations. This rule is not needed for
# ’user ppp’ type connection to the public Internet.
# This is the same IP address you captured and
# used in the outbound section.
pass in quick on dc0 proto udp from z.z.z.z to any port = 68 keep state

# Allow in standard www function because I have apache server
pass in quick on dc0 proto tcp from any to any port = 80 flags S keep state

# Allow in non-secure Telnet session from public Internet
# labeled non-secure because ID/PW passed over public Internet as clear text.
# Delete this sample group if you do not have telnet server enabled.
#pass in quick on dc0 proto tcp from any to any port = 23 flags S keep state

# Allow in secure FTP, Telnet, and SCP from public Internet
# This function is using SSH (secure shell)
pass in quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Block and log only first occurrence of all remaining traffic
# coming into the firewall. The logging of only the first
# occurrence stops a .denial of service. attack targeted
# at filling up your log file space.
# This rule enforces the block all by default logic.
block in log first quick on dc0 all
##################### End of rules file ########################################
```

28.5.14 NAT

NAT stands for Network Address Translation. To those familiar with Linux, this concept is called IP Masquerading; NAT and IP Masquerading are the same thing. One of the many things the IPF NAT function enables is the ability to have a private Local Area Network (LAN) behind the firewall sharing a single ISP assigned IP address on the public Internet.

You may ask why would someone want to do this. ISPs normally assign a dynamic IP address to their non-commercial users. Dynamic means that the IP address can be different each time you dial in and log on to your ISP, or for cable and DSL modem users when you power off and then power on your modems you can get assigned a different IP address. This IP address is how you are known to the public Internet.

Now lets say you have five PCs at home and each one needs Internet access. You would have to pay your ISP for an individual Internet account for each PC and have five phone lines.

With NAT you only need a single account with your ISP, then cable your other four PCs to a switch and the switch to the NIC in your FreeBSD system which is going to service your LAN as a gateway. NAT will automatically translate the private LAN IP address for each separate PC on the LAN to the single public IP address as it exits the firewall bound for the public Internet. It also does the reverse translation for returning packets.

NAT is most often accomplished without the approval, or knowledge, of your ISP and in most cases is grounds for your ISP terminating your account if found out. Commercial users pay a lot more for their Internet connection and usually get assigned a block of static IP address which never change. The ISP also expects and consents to their Commercial customers using NAT for their internal private LANs.

There is a special range of IP addresses reserved for NATed private LAN IP address. According to RFC 1918, you can use the following IP ranges for private nets which will never be routed directly to the public Internet:

Start IP `10.0.0.0`	-	Ending IP `10`
Start IP `172.16.0.0`	-	Ending IP `17`
Start IP `192.168.0.0`	-	Ending IP `19`

28.5.15 IPNAT

NAT rules are loaded by using the `ipnat` command. Typically the NAT rules are stored in `/etc/ipnat.rules`. See ipnat(1) for details.

When changing the NAT rules after NAT has been started, make your changes to the file containing the NAT rules, then run ipnat command with the `-CF` flags to delete the internal in use NAT rules and flush the contents of the translation table of all active entries.

To reload the NAT rules issue a command like this:

```
# ipnat -CF -f /etc/ipnat.rules
```

To display some statistics about your NAT, use this command:

```
# ipnat -s
```

To list the NAT table’s current mappings, use this command:

```
# ipnat -l
```

To turn verbose mode on, and display information relating to rule processing and active rules/table entries:

```
# ipnat -v
```

28.5.16 IPNAT Rules

NAT rules are very flexible and can accomplish many different things to fit the needs of commercial and home users.

The rule syntax presented here has been simplified to what is most commonly used in a non-commercial environment. For a complete rule syntax description see the ipnat(5) manual page.

The syntax for a NAT rule looks something like this:

```
map IF LAN_IP_RANGE -> PUBLIC_ADDRESS
```

The keyword `map` starts the rule.

Replace *`IF`* with the external interface.

The *`LAN_IP_RANGE`* is what your internal clients use for IP Addressing, usually this is something like `192.168.1.0/24`.

The *`PUBLIC_ADDRESS`* can either be the external IP address or the special keyword `0/32`, which means to use the IP address assigned to *`IF`*.

28.5.17 How NAT works

A packet arrives at the firewall from the LAN with a public destination. It passes through the outbound filter rules, NAT gets his turn at the packet and applies its rules top down, first matching rule wins. NAT tests each of its rules against the packets interface name and source IP address. When a packets interface name matches a NAT rule then the [source IP address, i.e. private LAN IP address] of the packet is checked to see if it falls within the IP address range specified to the left of the arrow symbol on the NAT rule. On a match the packet has its source IP address rewritten with the public IP address obtained by the `0/32` keyword. NAT posts a entry in its internal NAT table so when the packet returns from the public Internet it can be mapped back to its original private IP address and then passed to the filter rules for processing.

28.5.18 Enabling IPNAT

To enable IPNAT add these statements to `/etc/rc.conf`.

To enable your machine to route traffic between interfaces:

```
gateway_enable="YES"
```

To start IPNAT automatically each time:

```
ipnat_enable="YES"
```

To specify where to load the IPNAT rules from:

```
ipnat_rules="/etc/ipnat.rules"
```

28.5.19 NAT for a very large LAN

For networks that have large numbers of PC's on the LAN or networks with more than a single LAN, the process of funneling all those private IP addresses into a single public IP address becomes a resource problem that may cause problems with the same port numbers being used many times across many NATed LAN PC's, causing collisions. There are two ways to relieve this resource problem.

28.5.19.1 Assigning Ports to Use

A normal NAT rule would look like:

```
map dc0 192.168.1.0/24 -> 0/32
```

In the above rule the packet's source port is unchanged as the packet passes through IPNAT. By adding the portmap keyword you can tell IPNAT to only use source ports in a range. For example the following rule will tell IPNAT to modify the source port to be within that range:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp 20000:60000
```

Additionally we can make things even easier by using the `auto` keyword to tell IPNAT to determine by itself which ports are available to use:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp auto
```

28.5.19.2 Using a pool of public addresses

In very large LANs there comes a point where there are just too many LAN addresses to fit into a single public address. By changing the following rule:

```
map dc0 192.168.1.0/24 -> 204.134.75.1
```

Currently this rule maps all connections through `204.134.75.1`. This can be changed to specify a range:

```
map dc0 192.168.1.0/24 -> 204.134.75.1-10
```

Or a subnet using CIDR notation such as:

```
map dc0 192.168.1.0/24 -> 204.134.75.0/24
```

28.5.20 Port Redirection

A very common practice is to have a web server, email server, database server and DNS server each segregated to a different PC on the LAN. In this case the traffic from these servers still have to be NATed, but there has to be some way to direct the inbound traffic to the correct LAN PCs. IPNAT has the redirection facilities of NAT to solve this problem. Lets say you have your web server on LAN address `10.0.10.25` and your single public IP address is `20.20.20.5` you would code the rule like this:

```
rdr dc0 20.20.20.5/32 port 80 -> 10.0.10.25 port 80
```

or:

```
rdr dc0 0/32 port 80 -> 10.0.10.25 port 80
```

or for a LAN DNS Server on LAN address of `10.0.10.33` that needs to receive public DNS requests:

```
rdr dc0 20.20.20.5/32 port 53 -> 10.0.10.33 port 53 udp
```

28.5.21 FTP and NAT

FTP is a dinosaur left over from the time before the Internet as it is known today, when research universities were leased lined together and FTP was used to share files among research Scientists. This was a time when data security was not a consideration. Over the years the FTP protocol became buried into the backbone of the emerging Internet and its username and password being sent in clear text was never changed to address new security concerns. FTP has two flavors, it can run in active mode or passive mode. The difference is in how the data channel is acquired. Passive mode is more secure as the data channel is acquired be the ordinal ftp session requester. For a real good explanation of FTP and the different modes see http://www.slacksite.com/other/ftp.html.

28.5.21.1 IPNAT Rules

IPNAT has a special built in FTP proxy option which can be specified on the NAT map rule. It can monitor all outbound packet traffic for FTP active or passive start session requests and dynamically create temporary filter rules containing only the port number really in use for the data channel. This eliminates the security risk FTP normally exposes the firewall to from having large ranges of high order port numbers open.

This rule will handle all the traffic for the internal LAN:

```
map dc0 10.0.10.0/29 -> 0/32 proxy port 21 ftp/tcp
```

This rule handles the FTP traffic from the gateway:

```
map dc0 0.0.0.0/0 -> 0/32 proxy port 21 ftp/tcp
```

This rule handles all non-FTP traffic from the internal LAN:

```
map dc0 10.0.10.0/29 -> 0/32
```

The FTP map rule goes before our regular map rule. All packets are tested against the first rule from the top. Matches on interface name, then private LAN source IP address, and then is it a FTP packet. If all that matches then the special FTP proxy creates temp filter rules to let the FTP session packets pass in and out, in addition to also NATing the FTP packets. All LAN packets that are not FTP do not match the first rule and fall through to the third rule and are tested, matching on interface and source IP, then are NATed.

28.5.21.2 IPNAT FTP Filter Rules

Only one filter rule is needed for FTP if the NAT FTP proxy is used.

Without the FTP Proxy you will need the following three rules:

```
# Allow out LAN PC client FTP to public Internet
# Active and passive modes
pass out quick on rl0 proto tcp from any to any port = 21 flags S keep state
```

```
# Allow out passive mode data channel high order port numbers
pass out quick on rl0 proto tcp from any to any port > 1024 flags S keep state

# Active mode let data channel in from FTP server
pass in quick on rl0 proto tcp from any to any port = 20 flags S keep state
```

28.5.21.3 FTP NAT Proxy Bug

As of FreeBSD 4.9 which includes IPFILTER version 3.4.31 the FTP proxy works as documented during the FTP session until the session is told to close. When the close happens packets returning from the remote FTP server are blocked and logged coming in on port 21. The NAT FTP/proxy appears to remove its temp rules prematurely, before receiving the response from the remote FTP server acknowledging the close. A problem report was posted to the IPF mailing list.

The solution is to add a filter rule to get rid of these unwanted log messages or do nothing and ignore FTP inbound error messages in your log. Most people do not use outbound FTP too often.

```
block in quick on rl0 proto tcp from any to any port = 21
```

28.6 IPFW

> **Note:** 此一節的內容仍在陸續補充、更新，所以本節內容可能並未完全符合現況。.

The IPFIREWALL (IPFW) is a FreeBSD sponsored firewall software application authored and maintained by FreeBSD volunteer staff members. It uses the legacy stateless rules and a legacy rule coding technique to achieve what is referred to as Simple Stateful logic.

The IPFW sample rule set (found in `/etc/rc.firewall`) in the standard FreeBSD install is rather simple and it is not expected that it used directly without modifications. The example does not use stateful filtering, which is beneficial in most setups, so it will not be used as base for this section.

The IPFW stateless rule syntax is empowered with technically sophisticated selection capabilities which far surpasses the knowledge level of the customary firewall installer. IPFW is targeted at the professional user or the advanced technical computer hobbyist who have advanced packet selection requirements. A high degree of detailed knowledge into how different protocols use and create their unique packet header information is necessary before the power of the IPFW rules can be unleashed. Providing that level of explanation is out of the scope of this section of the handbook.

IPFW is composed of seven components, the primary component is the kernel firewall filter rule processor and its integrated packet accounting facility, the logging facility, the ’divert’ rule which triggers the NAT facility, and the advanced special purpose facilities, the dummynet traffic shaper facilities, the ’fwd rule’ forward facility, the bridge facility, and the ipstealth facility.

28.6.1 Enabling IPFW

IPFW is included in the basic FreeBSD install as a separate run time loadable module. The system will dynamically load the kernel module when the `rc.conf` statement `firewall_enable="YES"` is used. You do not need to compile IPFW into the FreeBSD kernel unless you want NAT function enabled.

After rebooting your system with `firewall_enable="YES"` in `rc.conf` the following white highlighted message is displayed on the screen as part of the boot process:

```
ipfw2 initialized, divert disabled, rule-based forwarding disabled, default to deny, logging disa
```

The loadable module does have logging ability compiled in. To enable logging and set the verbose logging limit, there is a knob you can set in `/etc/sysctl.conf` by adding these statements, logging will be enabled on future reboots:

```
net.inet.ip.fw.verbose=1
net.inet.ip.fw.verbose_limit=5
```

28.6.2 Kernel Options

It is not a mandatory requirement that you enable IPFW by compiling the following options into the FreeBSD kernel unless you need NAT function. It is presented here as background information.

```
options    IPFIREWALL
```

This option enables IPFW as part of the kernel

```
options    IPFIREWALL_VERBOSE
```

Enables logging of packets that pass through IPFW and have the 'log' keyword specified in the rule set.

```
options    IPFIREWALL_VERBOSE_LIMIT=5
```

Limits the number of packets logged through syslogd(8) on a per entry basis. You may wish to use this option in hostile environments which you want to log firewall activity. This will close a possible denial of service attack via syslog flooding.

```
options    IPFIREWALL_DEFAULT_TO_ACCEPT
```

This option will allow everything to pass through the firewall by default, which is a good idea when you are first setting up your firewall.

```
options    IPV6FIREWALL
options    IPV6FIREWALL_VERBOSE
options    IPV6FIREWALL_VERBOSE_LIMIT
options    IPV6FIREWALL_DEFAULT_TO_ACCEPT
```

These options are exactly the same as the IPv4 options but they are for IPv6. If you do not use IPv6 you might want to use IPV6FIREWALL without any rules to block all IPv6

```
options    IPDIVERT
```

This enables the use of NAT functionality.

Note: If you do not include IPFIREWALL_DEFAULT_TO_ACCEPT or set your rules to allow incoming packets you will block all packets going to and from this machine.

28.6.3 `/etc/rc.conf` Options

If you do not have IPFW compiled into your kernel you will need to load it with the following statement in your `/etc/rc.conf`:

```
firewall_enable="YES"
```

Set the script to run to activate your rules:

```
firewall_script="/etc/ipfw.rules"
```

Enable logging:

```
firewall_logging="YES"
```

Warning: The only thing that the `firewall_logging` variable will do is setting the `net.inet.ip.fw.verbose` sysctl variable to the value of `1` (see Section 28.6.1). There is no `rc.conf` variable to set log limitations, but it can be set via sysctl variable, manually or from the `/etc/sysctl.conf` file:

```
net.inet.ip.fw.verbose_limit=5
```

If your machine is acting as a gateway, i.e. providing Network Address Translation (NAT) via natd(8), please refer to Section 29.9 for information regarding the required `/etc/rc.conf` options.

28.6.4 The IPFW Command

The ipfw command is the normal vehicle for making manual single rule additions or deletions to the firewall active internal rules while it is running. The problem with using this method is once your system is shutdown or halted all the rules you added or changed or deleted are lost. Writing all your rules in a file and using that file to load the rules at boot time, or to replace in mass the currently running firewall rules with changes you made to the files content is the recommended method used here.

The ipfw command is still a very useful to display the running firewall rules to the console screen. The IPFW accounting facility dynamically creates a counter for each rule that counts each packet that matches the rule. During the process of testing a rule, listing the rule with its counter is the one of the ways of determining if the rule is functioning.

To list all the rules in sequence:

```
# ipfw list
```

To list all the rules with a time stamp of when the last time the rule was matched:

```
# ipfw -t list
```

To list the accounting information, packet count for matched rules along with the rules themselves. The first column is the rule number, followed by the number of outgoing matched packets, followed by the number of incoming matched packets, and then the rule itself.

```
# ipfw -a list
```

List the dynamic rules in addition to the static rules:

```
# ipfw -d list
```

Also show the expired dynamic rules:

```
# ipfw -d -e list
```

Zero the counters:

```
# ipfw zero
```

Zero the counters for just rule *NUM*:

```
# ipfw zero NUM
```

28.6.5 IPFW Rule Sets

A rule set is a group of ipfw rules coded to allow or deny packets based on the values contained in the packet. The bi-directional exchange of packets between hosts comprises a session conversation. The firewall rule set processes the packet twice: once on its arrival from the public Internet host and again as it leaves for its return trip back to the public Internet host. Each tcp/ip service (i.e. telnet, www, mail, etc.) is predefined by its protocol, and port number. This is the basic selection criteria used to create rules which will allow or deny services.

When a packet enters the firewall it is compared against the first rule in the rule set and progress one rule at a time moving from top to bottom of the set in ascending rule number sequence order. When the packet matches a rule selection parameters, the rules action field value is executed and the search of the rule set terminates for that packet. This is referred to as 「the first match wins」search method. If the packet does not match any of the rules, it gets caught by the mandatory ipfw default rule, number 65535 which denies all packets and discards them without any reply back to the originating destination.

> **Note:** The search continues after `count`, `skipto` and `tee` rules.

The instructions contained here are based on using rules that contain the stateful 'keep state', 'limit', 'in'/'out', and via options. This is the basic framework for coding an inclusive type firewall rule set.

An inclusive firewall only allows services matching the rules through. This way you can control what services can originate behind the firewall destine for the public Internet and also control the services which can originate from the public Internet accessing your private network. Everything else is denied by default design. Inclusive firewalls are much, much more secure than exclusive firewall rule sets and is the only rule set type covered here in.

> **Warning:** When working with the firewall rules be careful, you can end up locking your self out.

28.6.5.1 Rule Syntax

The rule syntax presented here has been simplified to what is necessary to create a standard inclusive type firewall rule set. For a complete rule syntax description see the ipfw(8) manual page.

Rules contain keywords: these keywords have to be coded in a specific order from left to right on the line. Keywords are identified in bold type. Some keywords have sub-options which may be keywords them selves and also include more sub-options.

is used to mark the start of a comment and may appear at the end of a rule line or on its own lines. Blank lines are ignored.

```
CMD RULE_NUMBER ACTION LOGGING SELECTION STATEFUL
```

28.6.5.1.1 CMD

Each new rule has to be prefixed with *add* to add the rule to the internal table.

28.6.5.1.2 RULE_NUMBER

Each rule has to have a rule number to go with it.

28.6.5.1.3 ACTION

A rule can be associated with one of the following actions, which will be executed when the packet matches the selection criterion of the rule.

allow | *accept* | *pass* | *permit*

These all mean the same thing which is to allow packets that match the rule to exit the firewall rule processing. The search terminates at this rule.

check-state

Checks the packet against the dynamic rules table. If a match is found, execute the action associated with the rule which generated this dynamic rule, otherwise move to the next rule. The check-state rule does not have selection criterion. If no check-state rule is present in the rule set, the dynamic rules table is checked at the first keep-state or limit rule.

deny | *drop*

Both words mean the same thing which is to discard packets that match this rule. The search terminates.

28.6.5.1.4 Logging

log or *logamount*

When a packet matches a rule with the log keyword, a message will be logged to syslogd with a facility name of SECURITY. The logging only occurs if the number of packets logged so far for that particular rule does not exceed the logamount parameter. If no logamount is specified, the limit is taken from the sysctl variable net.inet.ip.fw.verbose_limit. In both cases, a value of zero removes the logging limit. Once the limit is reached, logging can be re-enabled by clearing the logging counter or the packet counter for that rule, see the ipfw reset log command.

Note: Logging is done after all other packet matching conditions have been successfully verified, and before performing the final action (accept, deny) on the packet. It is up to you to decide which rules you want to enable logging on.

28.6.5.1.5 Selection

The keywords described in this section are used to describe attributes of the packet to be interrogated when determining whether rules match the packet or not. The following general-purpose attributes are provided for matching, and must be used in this order:

udp | tcp | icmp

or any protocol names found in `/etc/protocols` are recognized and may be used. The value specified is protocol to be matched against. This is a mandatory requirement.

from src to dst

The from and to keywords are used to match against IP addresses. Rules must specify BOTH source and destination parameters. `any` is a special keyword that matches any IP address. `me` is a special keyword that matches any IP address configured on an interface in your FreeBSD system to represent the PC the firewall is running on (i.e. this box) as in 'from me to any' or 'from any to me' or 'from 0.0.0.0/0 to any' or 'from any to 0.0.0.0/0' or 'from 0.0.0.0 to any' or 'from any to 0.0.0.0' or 'from me to 0.0.0.0'. IP addresses are specified as a dotted IP address numeric form/mask-length, or as single dotted IP address numeric form. This is a mandatory requirement. See this link for help on writing mask-lengths. http://jodies.de/ipcalc

port number

For protocols which support port numbers (such as TCP and UDP). It is mandatory that you code the port number of the service you want to match on. Service names (from `/etc/services`) may be used instead of numeric port values.

in | out

Matches incoming or outgoing packets, respectively. The in and out are keywords and it is mandatory that you code one or the other as part of your rule matching criterion.

via IF

Matches packets going through the interface specified by exact name. The `via` keyword causes the interface to always be checked as part of the match process.

setup

This is a mandatory keyword that identifies the session start request for TCP packets.

keep-state

This is a mandatory> keyword. Upon a match, the firewall will create a dynamic rule, whose default behavior is to match bidirectional traffic between source and destination IP/port using the same protocol.

limit {src-addr | src-port | dst-addr | dst-port}

The firewall will only allow *N* connections with the same set of parameters as specified in the rule. One or more of source and destination addresses and ports can be specified. The 'limit' and 'keep-state' can not be used on same rule. Limit provides the same stateful function as 'keep-state' plus its own functions.

28.6.5.2 Stateful Rule Option

Stateful filtering treats traffic as a bi-directional exchange of packets comprising a session conversation. It has the interrogation abilities to determine if the session conversation between the originating sender and the destination are following the valid procedure of bi-directional packet exchange. Any packets that do not properly fit the session conversation template are automatically rejected as impostors.

'check-state' is used to identify where in the IPFW rules set the packet is to be tested against the dynamic rules facility. On a match the packet exits the firewall to continue on its way and a new rule is dynamic created for the next anticipated packet being exchanged during this bi-directional session conversation. On a no match the packet advances to the next rule in the rule set for testing.

The dynamic rules facility is vulnerable to resource depletion from a SYN-flood attack which would open a huge number of dynamic rules. To counter this attack, FreeBSD version 4.5 added another new option named limit. This option is used to limit the number of simultaneous session conversations by interrogating the rules source or destinations fields as directed by the limit option and using the packet's IP address found there, in a search of the open dynamic rules counting the number of times this rule and IP address combination occurred, if this count is greater that the value specified on the limit option, the packet is discarded.

28.6.5.3 Logging Firewall Messages

The benefits of logging are obvious: it provides the ability to review after the fact the rules you activated logging on which provides information like, what packets had been dropped, what addresses they came from, where they were going, giving you a significant edge in tracking down attackers.

Even with the logging facility enabled, IPFW will not generate any rule logging on it's own. The firewall administrator decides what rules in the rule set he wants to log and adds the log verb to those rules. Normally only deny rules are logged, like the deny rule for incoming ICMP pings. It is very customary to duplicate the ipfw default deny everything rule with the log verb included as your last rule in the rule set. This way you get to see all the packets that did not match any of the rules in the rule set.

Logging is a two edged sword, if you are not careful, you can lose yourself in the over abundance of log data and fill your disk up with growing log files. DoS attacks that fill up disk drives is one of the oldest attacks around. These log message are not only written to syslogd, but also are displayed on the root console screen and soon become very annoying.

The `IPFIREWALL_VERBOSE_LIMIT=5` kernel option limits the number of consecutive messages sent to the system logger syslogd, concerning the packet matching of a given rule. When this option is enabled in the kernel, the number of consecutive messages concerning a particular rule is capped at the number specified. There is nothing to be gained from 200 log messages saying the same identical thing. For instance, five consecutive messages concerning a particular rule would be logged to syslogd, the remainder identical consecutive messages would be counted and posted to the syslogd with a phrase like this:

```
last message repeated 45 times
```

All logged packets messages are written by default to `/var/log/security` file, which is defined in the `/etc/syslog.conf` file.

28.6.5.4 Building a Rule Script

Most experienced IPFW users create a file containing the rules and code them in a manner compatible with running

them as a script. The major benefit of doing this is the firewall rules can be refreshed in mass without the need of rebooting the system to activate the new rules. This method is very convenient in testing new rules as the procedure can be executed as many times as needed. Being a script, you can use symbolic substitution to code frequent used values and substitution them in multiple rules. You will see this in the following example.

The script syntax used here is compatible with the 'sh', 'csh', 'tcsh' shells. Symbolic substitution fields are prefixed with a dollar sign $. Symbolic fields do not have the $ prefix. The value to populate the Symbolic field must be enclosed to "double quotes".

Start your rules file like this:

```
############### start of example ipfw rules script #############
#
ipfw -q -f flush       # Delete all rules
# Set defaults
oif="tun0"             # out interface
odns="192.0.2.11"      # ISP's DNS server IP address
cmd="ipfw -q add "     # build rule prefix
ks="keep-state"        # just too lazy to key this each time
$cmd 00500 check-state
$cmd 00502 deny all from any to any frag
$cmd 00501 deny tcp from any to any established
$cmd 00600 allow tcp from any to any 80 out via $oif setup $ks
$cmd 00610 allow tcp from any to $odns 53 out via $oif setup $ks
$cmd 00611 allow udp from any to $odns 53 out via $oif $ks
################### End of example ipfw rules script ############
```

That is all there is to it. The rules are not important in this example, how the Symbolic substitution field are populated and used are.

If the above example was in `/etc/ipfw.rules` file, you could reload these rules by entering on the command line.

```
# sh /etc/ipfw.rules
```

The `/etc/ipfw.rules` file could be located anywhere you want and the file could be named any thing you would like.

The same thing could also be accomplished by running these commands by hand:

```
# ipfw -q -f flush
# ipfw -q add check-state
# ipfw -q add deny all from any to any frag
# ipfw -q add deny tcp from any to any established
# ipfw -q add allow tcp from any to any 80 out via tun0 setup keep-state
# ipfw -q add allow tcp from any to 192.0.2.11 53 out via tun0 setup keep-state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0 keep-state
```

28.6.5.5 Stateful Ruleset

The following non-NATed rule set is an example of how to code a very secure 'inclusive' type of firewall. An inclusive firewall only allows services matching pass rules through and blocks all other by default. All firewalls have at the minimum two interfaces which have to have rules to allow the firewall to function.

All UNIX flavored operating systems, FreeBSD included, are designed to use interface `lo0` and IP address `127.0.0.1` for internal communication with in the operating system. The firewall rules must contain rules to allow free unmolested movement of these special internally used packets.

The interface which faces the public Internet, is the one which you code your rules to authorize and control access out to the public Internet and access requests arriving from the public Internet. This can be your ppp `tun0` interface or your NIC that is connected to your DSL or cable modem.

In cases where one or more than one NIC are connected to a private LANs behind the firewall, those interfaces must have rules coded to allow free unmolested movement of packets originating from those LAN interfaces.

The rules should be first organized into three major sections, all the free unmolested interfaces, public interface outbound, and the public interface inbound.

The order of the rules in each of the public interface sections should be in order of the most used rules being placed before less often used rules with the last rule in the section being a block log all packets on that interface and direction.

The Outbound section in the following rule set only contains ’allow’ rules which contain selection values that uniquely identify the service that is authorized for public Internet access. All the rules have the, proto, port, in/out, via and keep state option coded. The ’proto tcp’ rules have the ’setup’ option included to identify the start session request as the trigger packet to be posted to the keep state stateful table.

The Inbound section has all the blocking of undesirable packets first for two different reasons. First is these things being blocked may be part of an otherwise valid packet which may be allowed in by the later authorized service rules. Second reason is that by having a rule that explicitly blocks selected packets that I receive on an infrequent bases and do not want to see in the log, this keeps them from being caught by the last rule in the section which blocks and logs all packets which have fallen through the rules. The last rule in the section which blocks and logs all packets is how you create the legal evidence needed to prosecute the people who are attacking your system.

Another thing you should take note of, is there is no response returned for any of the undesirable stuff, their packets just get dropped and vanish. This way the attackers has no knowledge if his packets have reached your system. The less the attackers can learn about your system the more secure it is. When you log packets with port numbers you do not recognize, look the numbers up in `/etc/services/` or go to http://www.securitystats.com/tools/portsearch.php and do a port number lookup to find what the purpose of that port number is. Check out this link for port numbers used by Trojans: http://www.simovits.com/trojans/trojans.html.

28.6.5.6 An Example Inclusive Ruleset

The following non-NATed rule set is a complete inclusive type ruleset. You can not go wrong using this rule set for you own. Just comment out any pass rules for services you do not want. If you see messages in your log that you want to stop seeing just add a deny rule in the inbound section. You have to change the ’dc0’ interface name in every rule to the interface name of the NIC that connects your system to the public Internet. For user ppp it would be ’tun0’.

You will see a pattern in the usage of these rules.

- All statements that are a request to start a session to the public Internet use keep-state.
- All the authorized services that originate from the public Internet have the limit option to stop flooding.
- All rules use in or out to clarify direction.
- All rules use via interface name to specify the interface the packet is traveling over.

The following rules go into `/etc/ipfw.rules`.

```
################ Start of IPFW rules file ###############################
# Flush out the list before we begin.
ipfw -q -f flush

# Set rules command prefix
cmd="ipfw -q add"
pif="dc0"     # public interface name of NIC
              # facing the public Internet

######################################################################
# No restrictions on Inside LAN Interface for private network
# Not needed unless you have LAN.
# Change xl0 to your LAN NIC interface name
######################################################################
#$cmd 00005 allow all from any to any via xl0

######################################################################
# No restrictions on Loopback Interface
######################################################################
$cmd 00010 allow all from any to any via lo0

######################################################################
# Allow the packet through if it has previous been added to the
# the "dynamic" rules table by a allow keep-state statement.
######################################################################
$cmd 00015 check-state

######################################################################
# Interface facing Public Internet (Outbound Section)
# Interrogate session start requests originating from behind the
# firewall on the private network or from this gateway server
# destine for the public Internet.
######################################################################

# Allow out access to my ISP's Domain name server.
# x.x.x.x must be the IP address of your ISP.s DNS
# Dup these lines if your ISP has more than one DNS server
# Get the IP addresses from /etc/resolv.conf file
$cmd 00110 allow tcp from any to x.x.x.x 53 out via $pif setup keep-state
$cmd 00111 allow udp from any to x.x.x.x 53 out via $pif keep-state

# Allow out access to my ISP's DHCP server for cable/DSL configurations.
# This rule is not needed for .user ppp. connection to the public Internet.
# so you can delete this whole group.
# Use the following rule and check log for IP address.
# Then put IP address in commented out rule & delete first rule
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
#$cmd 00120 allow udp from any to x.x.x.x 67 out via $pif keep-state

# Allow out non-secure standard www function
$cmd 00200 allow tcp from any to any 80 out via $pif setup keep-state

# Allow out secure www function https over TLS SSL
```

```
$cmd 00220 allow tcp from any to any 443 out via $pif setup keep-state

# Allow out send & get email function
$cmd 00230 allow tcp from any to any 25 out via $pif setup keep-state
$cmd 00231 allow tcp from any to any 110 out via $pif setup keep-state

# Allow out FBSD (make install & CVSUP) functions
# Basically give user root "GOD" privileges.
$cmd 00240 allow tcp from me to any out via $pif setup keep-state uid root

# Allow out ping
$cmd 00250 allow icmp from any to any out via $pif keep-state

# Allow out Time
$cmd 00260 allow tcp from any to any 37 out via $pif setup keep-state

# Allow out nntp news (i.e. news groups)
$cmd 00270 allow tcp from any to any 119 out via $pif setup keep-state

# Allow out secure FTP, Telnet, and SCP
# This function is using SSH (secure shell)
$cmd 00280 allow tcp from any to any 22 out via $pif setup keep-state

# Allow out whois
$cmd 00290 allow tcp from any to any 43 out via $pif setup keep-state

# deny and log everything else that.s trying to get out.
# This rule enforces the block all by default logic.
$cmd 00299 deny log all from any to any out via $pif

################################################################
# Interface facing Public Internet (Inbound Section)
# Interrogate packets originating from the public Internet
# destine for this gateway server or the private network.
################################################################

# Deny all inbound traffic from non-routable reserved address spaces
$cmd 00300 deny all from 192.168.0.0/16 to any in via $pif  #RFC 1918 private IP
$cmd 00301 deny all from 172.16.0.0/12 to any in via $pif     #RFC 1918 private IP
$cmd 00302 deny all from 10.0.0.0/8 to any in via $pif          #RFC 1918 private IP
$cmd 00303 deny all from 127.0.0.0/8 to any in via $pif        #loopback
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif            #loopback
$cmd 00305 deny all from 169.254.0.0/16 to any in via $pif   #DHCP auto-config
$cmd 00306 deny all from 192.0.2.0/24 to any in via $pif       #reserved for docs
$cmd 00307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster interconnect
$cmd 00308 deny all from 224.0.0.0/3 to any in via $pif         #Class D & E multicast

# Deny public pings
$cmd 00310 deny icmp from any to any in via $pif

# Deny ident
$cmd 00315 deny tcp from any to any 113 in via $pif
```

```
# Deny all Netbios service. 137=name, 138=datagram, 139=session
# Netbios is MS/Windows sharing services.
# Block MS/Windows hosts2 name server requests 81
$cmd 00320 deny tcp from any to any 137 in via $pif
$cmd 00321 deny tcp from any to any 138 in via $pif
$cmd 00322 deny tcp from any to any 139 in via $pif
$cmd 00323 deny tcp from any to any 81 in via $pif

# Deny any late arriving packets
$cmd 00330 deny all from any to any frag in via $pif

# Deny ACK packets that did not match the dynamic rule table
$cmd 00332 deny tcp from any to any established in via $pif

# Allow traffic in from ISP's DHCP server. This rule must contain
# the IP address of your ISP.s DHCP server as it.s the only
# authorized source to send this packet type.
# Only necessary for cable or DSL configurations.
# This rule is not needed for .user ppp. type connection to
# the public Internet. This is the same IP address you captured
# and used in the outbound section.
#$cmd 00360 allow udp from any to x.x.x.x 67 in via $pif keep-state

# Allow in standard www function because I have apache server
$cmd 00400 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Allow in secure FTP, Telnet, and SCP from public Internet
$cmd 00410 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Allow in non-secure Telnet session from public Internet
# labeled non-secure because ID & PW are passed over public
# Internet as clear text.
# Delete this sample group if you do not have telnet server enabled.
$cmd 00420 allow tcp from any to me 23 in via $pif setup limit src-addr 2

# Reject & Log all incoming connections from the outside
$cmd 00499 deny log all from any to any in via $pif

# Everything else is denied by default
# deny and log all packets that fell through to see what they are
$cmd 00999 deny log all from any to any
################ End of IPFW rules file ################################
```

28.6.5.7 An Example NAT and Stateful Ruleset

There are some additional configuration statements that need to be enabled to activate the NAT function of IPFW. The kernel source needs 'option divert' statement added to the other IPFIREWALL statements compiled into a custom kernel.

In addition to the normal IPFW options in `/etc/rc.conf`, the following are needed.

```
natd_enable="YES"                  # Enable NATD function
```

```
natd_interface="rl0"                # interface name of public Internet NIC
natd_flags="-dynamic -m"            # -m = preserve port numbers if possible
```

Utilizing stateful rules with divert natd rule (Network Address Translation) greatly complicates the rule set coding logic. The positioning of the check-state, and 'divert natd' rules in the rule set becomes very critical. This is no longer a simple fall-through logic flow. A new action type is used, called 'skipto'. To use the skipto command it is mandatory that you number each rule so you know exactly where the skipto rule number is you are really jumping to.

The following is an uncommented example of one coding method, selected here to explain the sequence of the packet flow through the rule sets.

The processing flow starts with the first rule from the top of the rule file and progress one rule at a time deeper into the file until the end is reach or the packet being tested to the selection criteria matches and the packet is released out of the firewall. It is important to take notice of the location of rule numbers 100 101, 450, 500, and 510. These rules control the translation of the outbound and inbound packets so their entries in the keep-state dynamic table always register the private LAN IP address. Next notice that all the allow and deny rules specified the direction the packet is going (IE outbound or inbound) and the interface. Also notice that all the start outbound session requests all skipto rule 500 for the network address translation.

Lets say a LAN user uses their web browser to get a web page. Web pages use port 80 to communicate over. So the packet enters the firewall, It does not match 100 because it is headed out not in. It passes rule 101 because this is the first packet so it has not been posted to the keep-state dynamic table yet. The packet finally comes to rule 125 a matches. It is outbound through the NIC facing the public Internet. The packet still has it's source IP address as a private LAN IP address. On the match to this rule, two actions take place. The keep-state option will post this rule into the keep-state dynamic rules table and the specified action is executed. The action is part of the info posted to the dynamic table. In this case it is "skipto rule 500". Rule 500 NATs the packet IP address and out it goes. Remember this, this is very important. This packet makes its way to the destination and returns and enters the top of the rule set. This time it does match rule 100 and has it destination IP address mapped back to its corresponding LAN IP address. It then is processed by the check-state rule, it's found in the table as an existing session conversation and released to the LAN. It goes to the LAN PC that sent it and a new packet is sent requesting another segment of the data from the remote server. This time it gets checked by the check-state rule and its outbound entry is found, the associated action, 'skipto 500', is executed. The packet jumps to rule 500 gets NATed and released on it's way out.

On the inbound side, everything coming in that is part of an existing session conversation is being automatically handled by the check-state rule and the properly placed divert natd rules. All we have to address is denying all the bad packets and only allowing in the authorized services. Lets say there is a apache server running on the firewall box and we want people on the public Internet to be able to access the local web site. The new inbound start request packet matches rule 100 and its IP address is mapped to LAN IP for the firewall box. The packet is them matched against all the nasty things we want to check for and finally matches against rule 425. On a match two things occur. The packet rule is posted to the keep-state dynamic table but this time any new session requests originating from that source IP address is limited to 2. This defends against DoS attacks of service running on the specified port number. The action is allow so the packet is released to the LAN. On return the check-state rule recognizes the packet as belonging to an existing session conversation sends it to rule 500 for NATing and released to outbound interface.

Example Ruleset #1:

```
#!/bin/sh
cmd="ipfw -q add"
skip="skipto 500"
pif=rl0
ks="keep-state"
good_tcpo="22,25,37,43,53,80,443,110,119"
```

```
ipfw -q -f flush

$cmd 002 allow all from any to any via xl0  # exclude LAN traffic
$cmd 003 allow all from any to any via lo0  # exclude loopback traffic

$cmd 100 divert natd ip from any to any in via $pif
$cmd 101 check-state

# Authorized outbound packets
$cmd 120 $skip udp from any to xx.168.240.2 53 out via $pif $ks
$cmd 121 $skip udp from any to xx.168.240.5 53 out via $pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via $pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks
$cmd 135 $skip udp from any to any 123 out via $pif $ks


# Deny all inbound traffic from non-routable reserved address spaces
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 private IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 private IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 private IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #reserved for docs
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Class D & E multicast

# Authorized inbound packets
$cmd 400 allow udp from xx.70.207.54 to any 68 in $ks
$cmd 420 allow tcp from any to me 80 in via $pif setup limit src-addr 1


$cmd 450 deny log ip from any to any

# This is skipto location for outbound stateful rules
$cmd 500 divert natd ip from any to any out via $pif
$cmd 510 allow ip from any to any

######################## end of rules  #################
```

The following is pretty much the same as above, but uses a self documenting coding style full of description comments to help the inexperienced IPFW rule writer to better understand what the rules are doing.

Example Ruleset #2:

```
#!/bin/sh
################ Start of IPFW rules file ###############################
# Flush out the list before we begin.
ipfw -q -f flush

# Set rules command prefix
cmd="ipfw -q add"
skip="skipto 800"
```

```
pif="rl0"     # public interface name of NIC
              # facing the public Internet

#####################################################################
# No restrictions on Inside LAN Interface for private network
# Change xl0 to your LAN NIC interface name
#####################################################################
$cmd 005 allow all from any to any via xl0

#####################################################################
# No restrictions on Loopback Interface
#####################################################################
$cmd 010 allow all from any to any via lo0

#####################################################################
# check if packet is inbound and nat address if it is
#####################################################################
$cmd 014 divert natd ip from any to any in via $pif

#####################################################################
# Allow the packet through if it has previous been added to the
# the "dynamic" rules table by a allow keep-state statement.
#####################################################################
$cmd 015 check-state

#####################################################################
# Interface facing Public Internet (Outbound Section)
# Interrogate session start requests originating from behind the
# firewall on the private network or from this gateway server
# destine for the public Internet.
#####################################################################

# Allow out access to my ISP's Domain name server.
# x.x.x.x must be the IP address of your ISP's DNS
# Dup these lines if your ISP has more than one DNS server
# Get the IP addresses from /etc/resolv.conf file
$cmd 020 $skip tcp from any to x.x.x.x 53 out via $pif setup keep-state


# Allow out access to my ISP's DHCP server for cable/DSL configurations.
$cmd 030 $skip udp from any to x.x.x.x 67 out via $pif keep-state

# Allow out non-secure standard www function
$cmd 040 $skip tcp from any to any 80 out via $pif setup keep-state

# Allow out secure www function https over TLS SSL
$cmd 050 $skip tcp from any to any 443 out via $pif setup keep-state

# Allow out send & get email function
$cmd 060 $skip tcp from any to any 25 out via $pif setup keep-state
$cmd 061 $skip tcp from any to any 110 out via $pif setup keep-state

# Allow out FreeBSD (make install & CVSUP) functions
```

```
# Basically give user root "GOD" privileges.
$cmd 070 $skip tcp from me to any out via $pif setup keep-state uid root

# Allow out ping
$cmd 080 $skip icmp from any to any out via $pif keep-state

# Allow out Time
$cmd 090 $skip tcp from any to any 37 out via $pif setup keep-state

# Allow out nntp news (i.e. news groups)
$cmd 100 $skip tcp from any to any 119 out via $pif setup keep-state

# Allow out secure FTP, Telnet, and SCP
# This function is using SSH (secure shell)
$cmd 110 $skip tcp from any to any 22 out via $pif setup keep-state

# Allow out whois
$cmd 120 $skip tcp from any to any 43 out via $pif setup keep-state

# Allow ntp time server
$cmd 130 $skip udp from any to any 123 out via $pif keep-state

################################################################
# Interface facing Public Internet (Inbound Section)
# Interrogate packets originating from the public Internet
# destine for this gateway server or the private network.
################################################################

# Deny all inbound traffic from non-routable reserved address spaces
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 private IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 private IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 private IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #reserved for docs
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Class D & E multicast

# 拒絶 ident
$cmd 315 deny tcp from any to any 113 in via $pif

# 拒絶所有的 Netbios 服務. 137=name, 138=datagram, 139=session
# Netbios 是 MS/Windows 網路分享服務
# 阻擋所有的 MS/Windows 主機名稱伺服器hosts2 name server requests 81
$cmd 320 deny tcp from any to any 137 in via $pif
$cmd 321 deny tcp from any to any 138 in via $pif
$cmd 322 deny tcp from any to any 139 in via $pif
$cmd 323 deny tcp from any to any 81  in via $pif

# 拒絶任何的延遲到達之封包
$cmd 330 deny all from any to any frag in via $pif
```

```
# Deny ACK packets that did not match the dynamic rule table
$cmd 332 deny tcp from any to any established in via $pif

# Allow traffic in from ISP's DHCP server. This rule must contain
# the IP address of your ISP's DHCP server as it's the only
# authorized source to send this packet type.
# Only necessary for cable or DSL configurations.
# This rule is not needed for 'user ppp' type connection to
# the public Internet. This is the same IP address you captured
# and used in the outbound section.
$cmd 360 allow udp from x.x.x.x to any 68 in via $pif keep-state

# Allow in standard www function because I have Apache server
$cmd 370 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Allow in secure FTP, Telnet, and SCP from public Internet
$cmd 380 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Allow in non-secure Telnet session from public Internet
# labeled non-secure because ID & PW are passed over public
# Internet as clear text.
# Delete this sample group if you do not have telnet server enabled.
$cmd 390 allow tcp from any to me 23 in via $pif setup limit src-addr 2

# Reject & Log all unauthorized incoming connections from the public Internet
$cmd 400 deny log all from any to any in via $pif

# Reject & Log all unauthorized out going connections to the public Internet
$cmd 450 deny log all from any to any out via $pif

# This is skipto location for outbound stateful rules
$cmd 800 divert natd ip from any to any out via $pif
$cmd 801 allow ip from any to any

# Everything else is denied by default
# deny and log all packets that fell through to see what they are
$cmd 999 deny log all from any to any
################ End of IPFW rules file ################################
```

Chapter 29 網路進階練功房

29.1 概述

本章將介紹一些進階的網路設定主題。

讀完這章，您將了解：

- gateway(閘道)及route(路由)的概念。
- 如何設定IEEE 802.11 以及藍芽(Bluetooth®)設備。
- 如何以FreeBSD 作為bridge(橋接)。
- 如何為無碟系統設定網路開機。
- 如何設定NAT(Network Address Translation)。
- 如何透過PLIP 方式來連接兩台電腦。
- 如何在FreeBSD 內設定IPv6。
- 如何設定ATM。
- 如何去善用FreeBSD 的CARP(Common Access Redundancy Protocol)功能。

在開始閱讀這章之前，您需要：

- 瞭解`/etc/rc` 相關script 的概念。
- 熟悉基本常用的網路術語。
- 知道如何設定、安裝新的FreeBSD kernel (Chapter 8)。
- 知道如何透過port/package 安裝軟體(Chapter 4) 。

29.2 Gateways and Routes

Contributed by Coranth Gryphon.

為了讓一部電腦能找到另一部電腦，因此必需要有一種機制，讓這部電腦知道該怎麼做，這個機制就是路由選擇(*routing*)。一條路由(“route”)是由一對位址所定義的：一個是“目的地(destination)”以及另一個則是閘道(“gateway”)。這對位址表示要送到*目的地*的封包，必須經過*閘道*。目的地分為三種類型：主機、子網路(subnet)、預設路由(“default route”。若都沒有其它的路由可以使用，這時就會使用預設路由，稍後我們會對預設路由作進一步的說明。此外，閘道也可分為三種類型：主機、傳輸介面(interface，也稱為“links”)、乙太網路硬體位址(MAC addresses)。

29.2.1 範例

為了方便說明不同類型的路由選擇(routing)，以下使用`netstat` 指令的結果作為介紹範例：

```
% netstat -r
Routing tables
```

```
Destination      Gateway            Flags     Refs     Use     Netif Expire

default          outside-gw         UGSc       37      418      ppp0
localhost        localhost          UH          0      181       lo0
test0            0:e0:b5:36:cf:4f   UHLW        5    63288       ed0     77
10.20.30.255     link#1             UHLW        1     2421
example.com      link#1             UC          0        0
host1            0:e0:a8:37:8:1e    UHLW        3     4601       lo0
host2            0:e0:a8:37:8:1e    UHLW        0        5       lo0 =>
host2.example.com link#1             UC          0        0
224              link#1             UC          0        0
```

The first two lines specify the default route (which we will cover in the next section) and the `localhost` route.

The interface (`Netif` column) that this routing table specifies to use for `localhost` is `lo0`, also known as the loopback device. This says to keep all traffic for this destination internal, rather than sending it out over the LAN, since it will only end up back where it started.

The next thing that stands out are the addresses beginning with `0:e0:`. These are Ethernet hardware addresses, which are also known as MAC addresses. FreeBSD will automatically identify any hosts (`test0` in the example) on the local Ethernet and add a route for that host, directly to it over the Ethernet interface, `ed0`. There is also a timeout (`Expire` column) associated with this type of route, which is used if we fail to hear from the host in a specific amount of time. When this happens, the route to this host will be automatically deleted. These hosts are identified using a mechanism known as RIP (Routing Information Protocol), which figures out routes to local hosts based upon a shortest path determination.

FreeBSD will also add subnet routes for the local subnet (`10.20.30.255` is the broadcast address for the subnet `10.20.30`, and `example.com` is the domain name associated with that subnet). The designation `link#1` refers to the first Ethernet card in the machine. You will notice no additional interface is specified for those.

Both of these groups (local network hosts and local subnets) have their routes automatically configured by a daemon called **routed**. If this is not run, then only routes which are statically defined (i.e. entered explicitly) will exist.

The `host1` line refers to our host, which it knows by Ethernet address. Since we are the sending host, FreeBSD knows to use the loopback interface (`lo0`) rather than sending it out over the Ethernet interface.

The two `host2` lines are an example of what happens when we use an ifconfig(8) alias (see the section on Ethernet for reasons why we would do this). The => symbol after the `lo0` interface says that not only are we using the loopback (since this address also refers to the local host), but specifically it is an alias. Such routes only show up on the host that supports the alias; all other hosts on the local network will simply have a `link#1` line for such routes.

The final line (destination subnet `224`) deals with multicasting, which will be covered in another section.

Finally, various attributes of each route can be seen in the `Flags` column. Below is a short table of some of these flags and their meanings:

U	Up: The route is active.
H	Host: The route destination is a single host.
G	Gateway: Send anything for this destination on to this remote system, which will figure out from there where to send it.
S	Static: This route was configured manually, not automatically generated by the system.
C	Clone: Generates a new route based upon this route for machines we connect to. This type of route is normally used for local networks.

W	WasCloned: Indicated a route that was auto-configured based upon a local area network (Clone) route.
L	Link: Route involves references to Ethernet hardware.

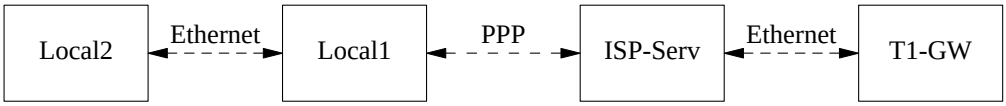
29.2.2 Default Routes

When the local system needs to make a connection to a remote host, it checks the routing table to determine if a known path exists. If the remote host falls into a subnet that we know how to reach (Cloned routes), then the system checks to see if it can connect along that interface.

If all known paths fail, the system has one last option: the "default" route. This route is a special type of gateway route (usually the only one present in the system), and is always marked with a `c` in the flags field. For hosts on a local area network, this gateway is set to whatever machine has a direct connection to the outside world (whether via PPP link, DSL, cable modem, T1, or another network interface).

If you are configuring the default route for a machine which itself is functioning as the gateway to the outside world, then the default route will be the gateway machine at your Internet Service Provider's (ISP) site.

Let us look at an example of default routes. This is a common configuration:



The hosts `Local1` and `Local2` are at your site. `Local1` is connected to an ISP via a dial up PPP connection. This PPP server computer is connected through a local area network to another gateway computer through an external interface to the ISPs Internet feed.

The default routes for each of your machines will be:

Host	Default Gateway	Interface
Local2	Local1	Ethernet
Local1	T1-GW	PPP

A common question is "Why (or how) would we set the `T1-GW` to be the default gateway for `Local1`, rather than the ISP server it is connected to?".

Remember, since the PPP interface is using an address on the ISP's local network for your side of the connection, routes for any other machines on the ISP's local network will be automatically generated. Hence, you will already know how to reach the `T1-GW` machine, so there is no need for the intermediate step of sending traffic to the ISP server.

It is common to use the address `X.X.X.1` as the gateway address for your local network. So (using the same example), if your local class-C address space was `10.20.30` and your ISP was using `10.9.9` then the default routes would be:

Host	Default Route
Local2 (10.20.30.2)	Local1 (10.20.30.1)
Local1 (10.20.30.1, 10.9.9.30)	T1-GW (10.9.9.1)

You can easily define the default route via the `/etc/rc.conf` file. In our example, on the `Local2` machine, we added the following line in `/etc/rc.conf`:

```
defaultrouter="10.20.30.1"
```

It is also possible to do it directly from the command line with the route(8) command:

```
# route add default 10.20.30.1
```

For more information on manual manipulation of network routing tables, consult route(8) manual page.

29.2.3 Dual Homed Hosts

There is one other type of configuration that we should cover, and that is a host that sits on two different networks. Technically, any machine functioning as a gateway (in the example above, using a PPP connection) counts as a dual-homed host. But the term is really only used to refer to a machine that sits on two local-area networks.

In one case, the machine has two Ethernet cards, each having an address on the separate subnets. Alternately, the machine may only have one Ethernet card, and be using ifconfig(8) aliasing. The former is used if two physically separate Ethernet networks are in use, the latter if there is one physical network segment, but two logically separate subnets.

Either way, routing tables are set up so that each subnet knows that this machine is the defined gateway (inbound route) to the other subnet. This configuration, with the machine acting as a router between the two subnets, is often used when we need to implement packet filtering or firewall security in either or both directions.

If you want this machine to actually forward packets between the two interfaces, you need to tell FreeBSD to enable this ability. See the next section for more details on how to do this.

29.2.4 Building a Router

A network router is simply a system that forwards packets from one interface to another. Internet standards and good engineering practice prevent the FreeBSD Project from enabling this by default in FreeBSD. You can enable this feature by changing the following variable to `YES` in rc.conf(5):

```
gateway_enable=YES          # Set to YES if this host will be a gateway
```

This option will set the sysctl(8) variable `net.inet.ip.forwarding` to `1`. If you should need to stop routing temporarily, you can reset this to `0` temporarily.

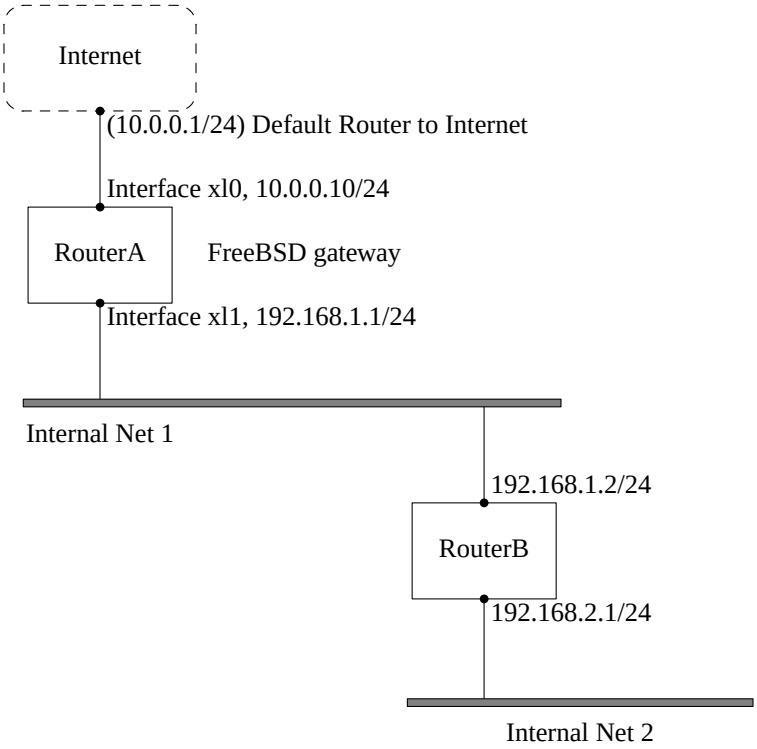
Your new router will need routes to know where to send the traffic. If your network is simple enough you can use static routes. FreeBSD also comes with the standard BSD routing daemon routed(8), which speaks RIP (both version 1 and version 2) and IRDP. Support for BGP v4, OSPF v2, and other sophisticated routing protocols is available with the `net/zebra` package. Commercial products such as **GateD** are also available for more complex network routing solutions.

29.2.5 Setting Up Static Routes

Contributed by Al Hoang.

29.2.5.1 Manual Configuration

Let us assume we have a network as follows:



In this scenario, `RouterA` is our FreeBSD machine that is acting as a router to the rest of the Internet. It has a default route set to `10.0.0.1` which allows it to connect with the outside world. We will assume that `RouterB` is already configured properly and knows how to get wherever it needs to go. (This is simple in this picture. Just add a default route on `RouterB` using `192.168.1.1` as the gateway.)

If we look at the routing table for `RouterA` we would see something like the following:

```
% netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif  Expire
default            10.0.0.1           UGS         0    49378    xl0
127.0.0.1          127.0.0.1          UH          0        6    lo0
10.0.0/24          link#1             UC          0        0    xl0
192.168.1/24       link#2             UC          0        0    xl1
```

With the current routing table `RouterA` will not be able to reach our Internal Net 2. It does not have a route for `192.168.2.0/24`. One way to alleviate this is to manually add the route. The following command would add the Internal Net 2 network to `RouterA`'s routing table using `192.168.1.2` as the next hop:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Now RouterA can reach any hosts on the 192.168.2.0/24 network.

29.2.5.2 Persistent Configuration

The above example is perfect for configuring a static route on a running system. However, one problem is that the routing information will not persist if you reboot your FreeBSD machine. The way to handle the addition of a static route is to put it in your /etc/rc.conf file:

```
# Add Internal Net 2 as a static route
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

The static_routes configuration variable is a list of strings separated by a space. Each string references to a route name. In our above example we only have one string in static_routes. This string is *internalnet2*. We then add a configuration variable called route_*internalnet2* where we put all of the configuration parameters we would give to the route(8) command. For our example above we would have used the command:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

so we need "-net 192.168.2.0/24 192.168.1.2".

As said above, we can have more than one string in static_routes. This allows us to create multiple static routes. The following lines shows an example of adding static routes for the 192.168.0.0/24 and 192.168.1.0/24 networks on an imaginary router:

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

29.2.6 Routing Propagation

We have already talked about how we define our routes to the outside world, but not about how the outside world finds us.

We already know that routing tables can be set up so that all traffic for a particular address space (in our examples, a class-C subnet) can be sent to a particular host on that network, which will forward the packets inbound.

When you get an address space assigned to your site, your service provider will set up their routing tables so that all traffic for your subnet will be sent down your PPP link to your site. But how do sites across the country know to send to your ISP?

There is a system (much like the distributed DNS information) that keeps track of all assigned address-spaces, and defines their point of connection to the Internet Backbone. The "Backbone" are the main trunk lines that carry Internet traffic across the country, and around the world. Each backbone machine has a copy of a master set of tables, which direct traffic for a particular network to a specific backbone carrier, and from there down the chain of service providers until it reaches your network.

It is the task of your service provider to advertise to the backbone sites that they are the point of connection (and thus the path inward) for your site. This is known as route propagation.

29.2.7 Troubleshooting

Sometimes, there is a problem with routing propagation, and some sites are unable to connect to you. Perhaps the most useful command for trying to figure out where routing is breaking down is the traceroute(8) command. It is equally useful if you cannot seem to make a connection to a remote machine (i.e. ping(8) fails).

The traceroute(8) command is run with the name of the remote host you are trying to connect to. It will show the gateway hosts along the path of the attempt, eventually either reaching the target host, or terminating because of a lack of connection.

For more information, see the manual page for traceroute(8).

29.2.8 Multicast Routing

FreeBSD supports both multicast applications and multicast routing natively. Multicast applications do not require any special configuration of FreeBSD; applications will generally run out of the box. Multicast routing requires that support be compiled into the kernel:

```
options MROUTING
```

In addition, the multicast routing daemon, mrouted(8) must be configured to set up tunnels and DVMRP via `/etc/mrouted.conf`. More details on multicast configuration may be found in the manual page for mrouted(8).

29.3 Wireless Networking

Loader, Marc Fonvieille, and Murray Stokely.

29.3.1 Wireless Networking Basics

Most wireless networks are based on the IEEE 802.11 standards. A basic wireless network consists of multiple stations communicating with radios that broadcast in either the 2.4GHz or 5GHz band (though this varies according to the locale and is also changing to enable communication in the 2.3GHz and 4.9GHz ranges).

802.11 networks are organized in two ways: in *infrastructure mode* one station acts as a master with all the other stations associating to it; the network is known as a BSS and the master station is termed an access point (AP). In a BSS all communication passes through the AP; even when one station wants to communicate with another wireless station messages must go through the AP. In the second form of network there is no master and stations communicate directly. This form of network is termed an IBSS and is commonly known as an *ad-hoc network*.

802.11 networks were first deployed in the 2.4GHz band using protocols defined by the IEEE 802.11 and 802.11b standard. These specifications include the operating frequencies, MAC layer characteristics including framing and transmission rates (communication can be done at various rates). Later the 802.11a standard defined operation in the 5GHz band, including different signalling mechanisms and higher transmission rates. Still later the 802.11g standard was defined to enable use of 802.11a signalling and transmission mechanisms in the 2.4GHz band in such a way as to be backwards compatible with 802.11b networks.

Separate from the underlying transmission techniques 802.11 networks have a variety of security mechanisms. The original 802.11 specifications defined a simple security protocol called WEP. This protocol uses a fixed pre-shared key and the RC4 cryptographic cipher to encode data transmitted on a network. Stations must all agree on the fixed

key in order to communicate. This scheme was shown to be easily broken and is now rarely used except to discourage transient users from joining networks. Current security practice is given by the IEEE 802.11i specification that defines new cryptographic ciphers and an additional protocol to authenticate stations to an access point and exchange keys for doing data communication. Further, cryptographic keys are periodically refreshed and there are mechanisms for detecting intrusion attempts (and for countering intrusion attempts). Another security protocol specification commonly used in wireless networks is termed WPA. This was a precursor to 802.11i defined by an industry group as an interim measure while waiting for 802.11i to be ratified. WPA specifies a subset of the requirements found in 802.11i and is designed for implementation on legacy hardware. Specifically WPA requires only the TKIP cipher that is derived from the original WEP cipher. 802.11i permits use of TKIP but also requires support for a stronger cipher, AES-CCM, for encrypting data. (The AES cipher was not required in WPA because it was deemed too computationally costly to be implemented on legacy hardware.)

Other than the above protocol standards the other important standard to be aware of is 802.11e. This defines protocols for deploying multi-media applications such as streaming video and voice over IP (VoIP) in an 802.11 network. Like 802.11i, 802.11e also has a precursor specification termed WME (later renamed WMM) that has been defined by an industry group as a subset of 802.11e that can be deployed now to enable multi-media applications while waiting for the final ratification of 802.11e. The most important thing to know about 802.11e and WME/WMM is that it enables prioritized traffic use of a wireless network through Quality of Service (QoS) protocols and enhanced media access protocols. Proper implementation of these protocols enable high speed bursting of data and prioritized traffic flow.

Since the 6.0 version, FreeBSD supports networks that operate using 802.11a, 802.11b, and 802.11g. The WPA and 802.11i security protocols are likewise supported (in conjunction with any of 11a, 11b, and 11g) and QoS and traffic prioritization required by the WME/WMM protocols are supported for a limited set of wireless devices.

29.3.2 Basic Setup

29.3.2.1 Kernel Configuration

To use wireless networking you need a wireless networking card and to configure the kernel with the appropriate wireless networking support. The latter is separated into multiple modules so that you only need to configure the software you are actually going to use.

The first thing you need is a wireless device. The most commonly used devices are those that use parts made by Atheros. These devices are supported by the ath(4) driver and require the following line to be added to the `/boot/loader.conf` file:

```
if_ath_load="YES"
```

The Atheros driver is split up into three separate pieces: the driver proper (ath(4)), the hardware support layer that handles chip-specific functions (ath_hal(4)), and an algorithm for selecting which of several possible rates for transmitting frames (ath_rate_sample here). When you load this support as modules these dependencies are automatically handled for you. If instead of an Atheros device you had another device you would select the module for that device; e.g.:

```
if_wi_load="YES"
```

for devices based on the Intersil Prism parts (wi(4) driver).

> **Note:** In the rest of this document, we will use an ath(4) device, the device name in the examples must be changed according to your configuration. A list of available wireless drivers can be found at the beginning of the

wlan(4) manual page. If a native FreeBSD driver for your wireless device does not exist, it may be possible to directly use the Windows driver with the help of the NDIS driver wrapper.

With a device driver configured you need to also bring in the 802.11 networking support required by the driver. For the ath(4) driver this is at least the wlan(4) module; this module is automatically loaded with the wireless device driver. With that you will need the modules that implement cryptographic support for the security protocols you intend to use. These are intended to be dynamically loaded on demand by the wlan(4) module but for now they must be manually configured. The following modules are available: wlan_wep(4), wlan_ccmp(4) and wlan_tkip(4). Both wlan_ccmp(4) and wlan_tkip(4) drivers are only needed if you intend to use the WPA and/or 802.11i security protocols. If your network is to run totally open (i.e., with no encryption) then you do not even need the wlan_wep(4) support. To load these modules at boot time, add the following lines to `/boot/loader.conf`:

```
wlan_wep_load="YES"
wlan_ccmp_load="YES"
wlan_tkip_load="YES"
```

With this information in the system bootstrap configuration file (i.e., `/boot/loader.conf`), you have to reboot your FreeBSD box. If you do not want to reboot your machine for the moment, you can just load the modules by hand using kldload(8).

Note: If you do not want to use modules, it is possible to compile these drivers into the kernel by adding the following lines to your kernel configuration file:

```
device ath               # Atheros IEEE 802.11 wireless network driver
device ath_hal           # Atheros Hardware Access Layer
device ath_rate_sample   # John Bicket's SampleRate control algorithm.
device wlan              # 802.11 support (Required)
device wlan_wep          # WEP crypto support for 802.11 devices
device wlan_ccmp         # AES-CCMP crypto support for 802.11 devices
device wlan_tkip         # TKIP and Michael crypto support for 802.11 devices
```

With this information in the kernel configuration file, recompile the kernel and reboot your FreeBSD machine.

When the system is up, we could find some information about the wireless device in the boot messages, like this:

```
ath0: <Atheros 5212> mem 0xff9f0000-0xff9fffff irq 17 at device 2.0 on pci2
ath0: Ethernet address: 00:11:95:d5:43:62
ath0: mac 7.9 phy 4.5 radio 5.6
```

29.3.3 Infrastructure Mode

The infrastructure mode or BSS mode is the mode that is typically used. In this mode, a number of wireless access points are connected to a wired network. Each wireless network has its own name, this name is called the SSID of the network. Wireless clients connect to the wireless access points.

29.3.3.1 FreeBSD Clients

29.3.3.1.1 How to Find Access Points

To scan for networks, use the `ifconfig` command. This request may take a few moments to complete as it requires that the system switches to each available wireless frequency and probes for available access points. Only the super-user can initiate such a scan:

```
# ifconfig ath0 up scan
SSID            BSSID              CHAN RATE  S:N   INT CAPS
dlinkap         00:13:46:49:41:76    6   54M 29:0   100 EPS  WPA WME
freebsdap       00:11:95:c3:0d:ac    1   54M 22:0   100 EPS  WPA
```

> **Note:** You must mark the interface `up` before you can scan. Subsequent scan requests do not require you to mark the interface up again.

The output of a scan request lists each BSS/IBSS network found. Beside the name of the network, `SSID`, we find the `BSSID` which is the MAC address of the access point. The `CAPS` field identifies the type of each network and the capabilities of the stations operating there:

`E`

Extended Service Set (ESS). Indicates that the station is part of an infrastructure network (in contrast to an IBSS/ad-hoc network).

`I`

IBSS/ad-hoc network. Indicates that the station is part of an ad-hoc network (in contrast to an ESS network).

`P`

Privacy. Data confidentiality is required for all data frames exchanged within the BSS. This means that this BSS requires the station to use cryptographic means such as WEP, TKIP or AES-CCMP to encrypt/decrypt data frames being exchanged with others.

`S`

Short Preamble. Indicates that the network is using short preambles (defined in 802.11b High Rate/DSSS PHY, short preamble utilizes a 56 bit sync field in contrast to a 128 bit field used in long preamble mode).

`s`

Short slot time. Indicates that the 802.11g network is using a short slot time because there are no legacy (802.11b) stations present.

One can also display the current list of known networks with:

```
# ifconfig ath0 list scan
```

This information may be updated automatically by the adapter or manually with a `scan` request. Old data is automatically removed from the cache, so over time this list may shrink unless more scans are done.

29.3.3.1.2 Basic Settings

This section provides a simple example of how to make the wireless network adapter work in FreeBSD without encryption. After you are familiar with these concepts, we strongly recommend using WPA to set up your wireless network.

There are three basic steps to configure a wireless network: selecting an access point, authenticating your station, and configuring an IP address. The following sections discuss each step.

29.3.3.1.2.1 Selecting an Access Point

Most of time it is sufficient to let the system choose an access point using the builtin heuristics. This is the default behaviour when you mark an interface up or otherwise configure an interface by listing it in `/etc/rc.conf`, e.g.:

```
ifconfig_ath0="DHCP"
```

If there are multiple access points and you want to select a specific one, you can select it by its SSID:

```
ifconfig_ath0="ssid your_ssid_here DHCP"
```

In an environment where there are multiple access points with the same SSID (often done to simplify roaming) it may be necessary to associate to one specific device. In this case you can also specify the BSSID of the access point (you can also leave off the SSID):

```
ifconfig_ath0="ssid your_ssid_here bssid xx:xx:xx:xx:xx:xx DHCP"
```

There are other ways to constrain the choice of an access point such as limiting the set of frequencies the system will scan on. This may be useful if you have a multi-band wireless card as scanning all the possible channels can be time-consuming. To limit operation to a specific band you can use the `mode` parameter; e.g.:

```
ifconfig_ath0="mode 11g ssid your_ssid_here DHCP"
```

will force the card to operate in 802.11g which is defined only for 2.4GHz frequencies so any 5GHz channels will not be considered. Other ways to do this are the `channel` parameter, to lock operation to one specific frequency, and the `chanlist` parameter, to specify a list of channels for scanning. More information about these parameters can be found in the ifconfig(8) manual page.

29.3.3.1.2.2 Authentication

Once you have selected an access point your station needs to authenticate before it can pass data. Authentication can happen in several ways. The most common scheme used is termed open authentication and allows any station to join the network and communicate. This is the authentication you should use for test purpose the first time you set up a wireless network. Other schemes require cryptographic handshakes be completed before data traffic can flow; either using pre-shared keys or secrets, or more complex schemes that involve backend services such as RADIUS. Most users will use open authentication which is the default setting. Next most common setup is WPA-PSK, also known as WPA Personal, which is described below.

> **Note:** If you have an Apple AirPort® Extreme base station for an access point you may need to configure shared-key authentication together with a WEP key. This can be done in the `/etc/rc.conf` file or using the wpa_supplicant(8) program. If you have a single AirPort base station you can setup access with something like:
>
> ```
> ifconfig_ath0="authmode shared wepmode on weptxkey 1 wepkey 01234567 DHCP"
> ```

In general shared key authentication is to be avoided because it uses the WEP key material in a highly-constrained manner making it even easier to crack the key. If WEP must be used (e.g., for compatibility with legacy devices) it is better to use WEP with `open` authentication. More information regarding WEP can be found in the Section 29.3.3.1.4.

29.3.3.1.2.3 Getting an IP Address with DHCP

Once you have selected an access point and set the authentication parameters, you will have to get an IP address to communicate. Most of time you will obtain your wireless IP address via DHCP. To achieve that, simply edit `/etc/rc.conf` and add `DHCP` to the configuration for your device as shown in various examples above:

```
ifconfig_ath0="DHCP"
```

At this point, you are ready to bring up the wireless interface:

```
# /etc/rc.d/netif start
```

Once the interface is running, use `ifconfig` to see the status of the interface `ath0`:

```
# ifconfig ath0
ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
        inet 192.168.1.100 netmask 0xffffff00 broadcast 192.168.1.255
        ether 00:11:95:d5:43:62
        media: IEEE 802.11 Wireless Ethernet autoselect (OFDM/54Mbps)
        status: associated
        ssid dlinkap channel 6 bssid 00:13:46:49:41:76
        authmode OPEN privacy OFF txpowmax 36 protmode CTS bintval 100
```

The `status: associated` means you are connected to the wireless network (to the `dlinkap` network in our case). The `bssid 00:13:46:49:41:76` part is the MAC address of your access point; the `authmode` line informs you that the communication is not encrypted (`OPEN`).

29.3.3.1.2.4 Static IP Address

In the case you cannot obtain an IP address from a DHCP server, you can set a fixed IP address. Replace the `DHCP` keyword shown above with the address information. Be sure to retain any other parameters you have set up for selecting an access point:

```
ifconfig_ath0="inet 192.168.1.100 netmask 255.255.255.0 ssid your_ssid_here"
```

29.3.3.1.3 WPA

WPA (Wi-Fi Protected Access) is a security protocol used together with 802.11 networks to address the lack of proper authentication and the weakness of WEP. WPA leverages the 802.1X authentication protocol and uses one of several ciphers instead of WEP for data integrity. The only cipher required by WPA is TKIP (Temporary Key Integrity Protocol) which is a cipher that extends the basic RC4 cipher used by WEP by adding integrity checking,

tamper detection, and measures for responding to any detected intrusions. TKIP is designed to work on legacy hardware with only software modification; it represents a compromise that improves security but is still not entirely immune to attack. WPA also specifies the AES-CCMP cipher as an alternative to TKIP and that is preferred when possible; for this specification the term WPA2 (or RSN) is commonly used.

WPA defines authentication and encryption protocols. Authentication is most commonly done using one of two techniques: by 802.1X and a backend authentication service such as RADIUS, or by a minimal handshake between the station and the access point using a pre-shared secret. The former is commonly termed WPA Enterprise with the latter known as WPA Personal. Since most people will not set up a RADIUS backend server for wireless network, WPA-PSK is by far the most commonly encountered configuration for WPA.

The control of the wireless connection and the authentication (key negotiation or authentication with a server) is done with the wpa_supplicant(8) utility. This program requires a configuration file, `/etc/wpa_supplicant.conf`, to run. More information regarding this file can be found in the wpa_supplicant.conf(5) manual page.

29.3.3.1.3.1 WPA-PSK

WPA-PSK also known as WPA-Personal is based on a pre-shared key (PSK) generated from a given password and that will be used as the master key in the wireless network. This means every wireless user will share the same key. WPA-PSK is intended for small networks where the use of an authentication server is not possible or desired.

> **Warning:** Always use strong passwords that are sufficiently long and made from a rich alphabet so they will not be guessed and/or attacked.

The first step is the configuration of the `/etc/wpa_supplicant.conf` file with the SSID and the pre-shared key of your network:

```
network={
  ssid="freebsdap"
  psk="freebsdmall"
}
```

Then, in `/etc/rc.conf`, we indicate that the wireless device configuration will be done with WPA and the IP address will be obtained with DHCP:

```
ifconfig_ath0="WPA DHCP"
```

Then, we can bring up the interface:

```
# /etc/rc.d/netif start
Starting wpa_supplicant.
DHCPDISCOVER on ath0 to 255.255.255.255 port 67 interval 5
DHCPDISCOVER on ath0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.0.1
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      ether 00:11:95:d5:43:62
```

```
      media: IEEE 802.11 Wireless Ethernet autoselect (OFDM/36Mbps)
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA privacy ON deftxkey UNDEF TKIP 2:128-bit txpowmax 36
      protmode CTS roaming MANUAL bintval 100
```

Or you can try to configure it manually using the same `/etc/wpa_supplicant.conf` above, and run:

```
# wpa_supplicant -i ath0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:11:95:c3:0d:ac (SSID='freebsdap' freq=2412 MHz)
Associated with 00:11:95:c3:0d:ac
WPA: Key negotiation completed with 00:11:95:c3:0d:ac [PTK=TKIP GTK=TKIP]
```

The next operation is the launch of the `dhclient` command to get the IP address from the DHCP server:

```
# dhclient ath0
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
# ifconfig ath0
ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      ether 00:11:95:d5:43:62
      media: IEEE 802.11 Wireless Ethernet autoselect (OFDM/48Mbps)
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA privacy ON deftxkey UNDEF TKIP 2:128-bit txpowmax 36
      protmode CTS roaming MANUAL bintval 100
```

> **Note:** If the `/etc/rc.conf` is set up with the line `ifconfig_ath0="DHCP"` then it is no need to run the `dhclient` command manually, `dhclient` will be launched after `wpa_supplicant` plumbs the keys.

In the case where the use of DHCP is not possible, you can set a static IP address after `wpa_supplicant` has authenticated the station:

```
# ifconfig ath0 inet 192.168.0.100 netmask 255.255.255.0
# ifconfig ath0
ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
      inet 192.168.0.100 netmask 0xffffff00 broadcast 192.168.0.255
      ether 00:11:95:d5:43:62
      media: IEEE 802.11 Wireless Ethernet autoselect (OFDM/36Mbps)
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA privacy ON deftxkey UNDEF TKIP 2:128-bit txpowmax 36
      protmode CTS roaming MANUAL bintval 100
```

When DHCP is not used, you also have to manually set up the default gateway and the nameserver:

```
# route add default your_default_router
```

```
# echo "nameserver your_DNS_server" >> /etc/resolv.conf
```

29.3.3.1.3.2 WPA with EAP-TLS

The second way to use WPA is with an 802.1X backend authentication server, in this case WPA is called WPA-Enterprise to make difference with the less secure WPA-Personal with its pre-shared key. The authentication in WPA-Enterprise is based on EAP (Extensible Authentication Protocol).

EAP does not come with an encryption method, it was decided to embed EAP inside an encrypted tunnel. Many types of EAP authentication methods have been designed, the most common methods are EAP-TLS, EAP-TTLS and EAP-PEAP.

EAP-TLS (EAP with Transport Layer Security) is a very well-supported authentication protocol in the wireless world since it was the first EAP method to be certified by the Wi-Fi alliance (http://www.wi-fi.org/). EAP-TLS will require three certificates to run: the CA certificate (installed on all machines), the server certificate for your authentication server, and one client certificate for each wireless client. In this EAP method, both authentication server and wireless client authenticate each other in presenting their respective certificates, and they verify that these certificates were signed by your organization's certificate authority (CA).

As previously, the configuration is done via `/etc/wpa_supplicant.conf`:

```
network={
  ssid="freebsdap" ❶
  proto=RSN  ❷
  key_mgmt=WPA-EAP ❸
  eap=TLS ❹
  identity="loader" ❺
  ca_cert="/etc/certs/cacert.pem" ❻
  client_cert="/etc/certs/clientcert.pem" ❼
  private_key="/etc/certs/clientkey.pem" ❽
  private_key_passwd="freebsdmallclient" ❾
}
```

❶ This field indicates the network name (SSID).

❷ Here, we use RSN (IEEE 802.11i) protocol, i.e., WPA2.

❸ The `key_mgmt` line refers to the key management protocol we use. In our case it is WPA using EAP authentication: `WPA-EAP`.

❹ In this field, we mention the EAP method for our connection.

❺ The `identity` field contains the identity string for EAP.

❻ The `ca_cert` field indicates the pathname of the CA certificate file. This file is needed to verify the server certificat.

❼ The `client_cert` line gives the pathname to the client certificate file. This certificate is unique to each wireless client of the network.

❽ The `private_key` field is the pathname to the client certificate private key file.

❾ The `private_key_passwd` field contains the passphrase for the private key.

Then add the following line to `/etc/rc.conf`:

```
ifconfig_ath0="WPA DHCP"
```

The next step is to bring up the interface with the help of the `rc.d` facility:

```
# /etc/rc.d/netif start
Starting wpa_supplicant.
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      ether 00:11:95:d5:43:62
      media: IEEE 802.11 Wireless Ethernet autoselect (DS/11Mbps)
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA2/802.11i privacy ON deftxkey UNDEF TKIP 2:128-bit
      txpowmax 36 protmode CTS roaming MANUAL bintval 100
```

As previously shown, it is also possible to bring up the interface manually with both `wpa_supplicant` and `ifconfig` commands.

29.3.3.1.3.3 WPA with EAP-TTLS

With EAP-TLS both the authentication server and the client need a certificate, with EAP-TTLS (EAP-Tunneled Transport Layer Security) a client certificate is optional. This method is close to what some secure web sites do , where the web server can create a secure SSL tunnel even if the visitors do not have client-side certificates. EAP-TTLS will use the encrypted TLS tunnel for safe transport of the authentication data.

The configuration is done via the `/etc/wpa_supplicant.conf` file:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=TTLS ❶
  identity="test" ❷
  password="test" ❸
  ca_cert="/etc/certs/cacert.pem" ❹
  phase2="auth=MD5" ❺
}
```

❶ In this field, we mention the EAP method for our connection.

❷ The `identity` field contains the identity string for EAP authentication inside the encrypted TLS tunnel.

❸ The `password` field contains the passphrase for the EAP authentication.

❹ The `ca_cert` field indicates the pathname of the CA certificate file. This file is needed to verify the server certificat.

❺ In this field, we mention the authentication method used in the encrypted TLS tunnel. In our case, EAP with MD5-Challenge has been used. The “inner authentication” phase is often called “phase2”.

You also have to add the following line to `/etc/rc.conf`:

```
ifconfig_ath0="WPA DHCP"
```

The next step is to bring up the interface:

```
# /etc/rc.d/netif start
Starting wpa_supplicant.
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      ether 00:11:95:d5:43:62
      media: IEEE 802.11 Wireless Ethernet autoselect (DS/11Mbps)
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA2/802.11i privacy ON deftxkey UNDEF TKIP 2:128-bit
      txpowmax 36 protmode CTS roaming MANUAL bintval 100
```

29.3.3.1.3.4 WPA with EAP-PEAP

PEAP (Protected EAP) has been designed as an alternative to EAP-TTLS. There are two types of PEAP methods, the most common one is PEAPv0/EAP-MSCHAPv2. In the rest of this document, we will use the PEAP term to refer to that EAP method. PEAP is the most used EAP standard after EAP-TLS, in other words if you have a network with mixed OSes, PEAP should be the most supported standard after EAP-TLS.

PEAP is similar to EAP-TTLS: it uses a server-side certificate to authenticate clients by creating an encrypted TLS tunnel between the client and the authentication server, which protects the ensuing exchange of authentication information. In term of security the difference between EAP-TTLS and PEAP is that PEAP authentication broadcasts the username in clear, only the password is sent in the encrypted TLS tunnel. EAP-TTLS will use the TLS tunnel for both username and password.

We have to edit the `/etc/wpa_supplicant.conf` file and add the EAP-PEAP related settings:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP ❶
  identity="test" ❷
  password="test" ❸
  ca_cert="/etc/certs/cacert.pem" ❹
  phase1="peaplabel=0" ❺
  phase2="auth=MSCHAPV2" ❻
}
```

❶ In this field, we mention the EAP method for our connection.

❷ The `identity` field contains the identity string for EAP authentication inside the encrypted TLS tunnel.

❸ The `password` field contains the passphrase for the EAP authentication.

❹ The `ca_cert` field indicates the pathname of the CA certificate file. This file is needed to verify the server certificat.

❺ This field contains the parameters for the first phase of the authentication (the TLS tunnel). According to the authentication server used, you will have to specify a specific label for the authentication. Most of time, the label will be "client EAP encryption" which is set by using `peaplabel=0`. More information can be found in the wpa_supplicant.conf(5) manual page.

❻ In this field, we mention the authentication protocol used in the encrypted TLS tunnel. In the case of PEAP, it is `auth=MSCHAPV2`.

The following must be added to `/etc/rc.conf`:

```
ifconfig_ath0="WPA DHCP"
```

Then, we can bring up the interface:

```
# /etc/rc.d/netif start
Starting wpa_supplicant.
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      ether 00:11:95:d5:43:62
      media: IEEE 802.11 Wireless Ethernet autoselect (DS/11Mbps)
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA2/802.11i privacy ON deftxkey UNDEF TKIP 2:128-bit
      txpowmax 36 protmode CTS roaming MANUAL bintval 100
```

29.3.3.1.4 WEP

WEP (Wired Equivalent Privacy) is part of the original 802.11 standard. There is no authentication mechanism, only a weak form of access control, and it is easily to be cracked.

WEP can be set up with `ifconfig`:

```
# ifconfig ath0 inet 192.168.1.100 netmask 255.255.255.0 ssid my_net \
     wepmode on weptxkey 3 wepkey 3:0x3456789012
```

· The `weptxkey` means which WEP key will be used in the transmission. Here we used the third key. This must match the setting in the access point.

- The `wepkey` means setting the selected WEP key. It should in the format *`index:key`*, if the index is not given, key `1` is set. That is to say we need to set the index if we use keys other than the first key.

> **Note:** You must replace the `0x3456789012` with the key configured for use on the access point.

You are encouraged to read ifconfig(8) manual page for further information.

The `wpa_supplicant` facility also can be used to configure your wireless interface with WEP. The example above can be set up by adding the following lines to `/etc/wpa_supplicant.conf`:

```
network={
  ssid="my_net"
  key_mgmt=NONE
  wep_key3=3456789012
  wep_tx_keyidx=3
}
```

Then:

```
# wpa_supplicant -i ath0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:13:46:49:41:76 (SSID='dlinkap' freq=2437 MHz)
Associated with 00:13:46:49:41:76
```

29.3.4 Ad-hoc Mode

IBSS mode, also called ad-hoc mode, is designed for point to point connections. For example, to establish an ad-hoc network between the machine `A` and the machine `B` we will just need to choose two IP adresses and a SSID.

On the box `A`:

```
# ifconfig ath0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mediaopt adhoc
# ifconfig ath0
  ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
   inet6 fe80::211:95ff:fec3:dac%ath0 prefixlen 64 scopeid 0x4
   ether 00:11:95:c3:0d:ac
   media: IEEE 802.11 Wireless Ethernet autoselect <adhoc> (autoselect <adhoc>)
   status: associated
   ssid freebsdap channel 2 bssid 02:11:95:c3:0d:ac
   authmode OPEN privacy OFF txpowmax 36 protmode CTS bintval 100
```

The `adhoc` parameter indicates the interface is running in the IBSS mode.

On `B`, we should be able to detect `A`:

```
# ifconfig ath0 up scan
  SSID            BSSID              CHAN RATE  S:N   INT CAPS
  freebsdap       02:11:95:c3:0d:ac    2   54M 19:0   100 IS
```

The I in the output confirms the machine A is in ad-hoc mode. We just have to configure B with a different IP address:

```
# ifconfig ath0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap mediaopt adhoc
# ifconfig ath0
  ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
   inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
   ether 00:11:95:d5:43:62
   media: IEEE 802.11 Wireless Ethernet autoselect <adhoc> (autoselect <adhoc>)
   status: associated
   ssid freebsdap channel 2 bssid 02:11:95:c3:0d:ac
   authmode OPEN privacy OFF txpowmax 36 protmode CTS bintval 100
```

Both A and B are now ready to exchange informations.

29.3.5 FreeBSD Host Access Points

FreeBSD can act as an Access Point (AP) which eliminates the need to buy a hardware AP or run an ad-hoc network. This can be particularly useful when your FreeBSD machine is acting as a gateway to another network (e.g., the Internet).

29.3.5.1 Basic Settings

Before configuring your FreeBSD machine as an AP, the kernel must be configured with the appropriate wireless networking support for your wireless card. You also have to add the support for the security protocols you intend to use. For more details, see Section 29.3.2.

> **Note:** The use of the NDIS driver wrapper and the Windows drivers do not allow currently the AP operation. Only native FreeBSD wireless drivers support AP mode.

Once the wireless networking support is loaded, you can check if your wireless device supports the host-based access point mode (also know as hostap mode):

```
# ifconfig ath0 list caps
ath0=783ed0f<WEP,TKIP,AES,AES_CCM,IBSS,HOSTAP,AHDEMO,TXPMGT,SHSLOT,SHPREAMBLE,MONITOR,TKIPMIC,WPA
```

This output displays the card capabilities; the HOSTAP word confirms this wireless card can act as an Access Point. Various supported ciphers are also mentioned: WEP, TKIP, WPA2, etc., these informations are important to know what security protocols could be set on the Access Point.

The wireless device can now be put into hostap mode and configured with the correct SSID and IP address:

```
# ifconfig ath0 ssid freebsdap mode 11g mediaopt hostap inet 192.168.0.1 netmask 255.255.255.0
```

Use again ifconfig to see the status of the ath0 interface:

```
# ifconfig ath0
  ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
   inet6 fe80::211:95ff:fec3:dac%ath0 prefixlen 64 scopeid 0x4
```

```
   ether 00:11:95:c3:0d:ac
   media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
   status: associated
   ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
   authmode OPEN privacy OFF txpowmax 38 bmiss 7 protmode CTS burst dtimperiod 1 bintval 100
```

The `hostap` parameter indicates the interface is running in the host-based access point mode.

The interface configuration can be done automatically at boot time by adding the following line to `/etc/rc.conf`:

```
ifconfig_ath0="ssid freebsdap mode 11g mediaopt hostap inet 192.168.0.1 netmask 255.255.255.0"
```

29.3.5.2 Host-based Access Point without Authentication or Encryption

Although it is not recommended to run an AP without any authentication or encryption, this is a simple way to check if your AP is working. This configuration is also important for debugging client issues.

Once the AP configured as previously shown, it is possible from another wireless machine to initiate a scan to find the AP:

```
# ifconfig ath0 up scan
SSID            BSSID              CHAN RATE  S:N   INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M 22:1   100 ES
```

The client machine found the Access Point and can be associated with it:

```
# ifconfig ath0 ssid freebsdap inet 192.168.0.2 netmask 255.255.255.0
# ifconfig ath0
  ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   inet6 fe80::211:95ff:fed5:4362%ath0 prefixlen 64 scopeid 0x1
   inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
   ether 00:11:95:d5:43:62
   media: IEEE 802.11 Wireless Ethernet autoselect (OFDM/54Mbps)
   status: associated
   ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
   authmode OPEN privacy OFF txpowmax 36 protmode CTS bintval 100
```

29.3.5.3 WPA Host-based Access Point

This section will focus on setting up FreeBSD Access Point using the WPA security protocol. More details regarding WPA and the configuration of WPA-based wireless clients can be found in the Section 29.3.3.1.3.

The **hostapd** daemon is used to deal with client authentication and keys management on the WPA enabled Access Point.

In the following, all the configuration operations will be performed on the FreeBSD machine acting as AP. Once the AP is correctly working, **hostapd** should be automatically enabled at boot with the following line in `/etc/rc.conf`:

```
hostapd_enable="YES"
```

Before trying to configure **hostapd**, be sure you have done the basic settings introduced in the Section 29.3.5.1.

29.3.5.3.1 WPA-PSK

WPA-PSK is intended for small networks where the use of an backend authentication server is not possible or desired.

The configuration is done in the `/etc/hostapd.conf` file:

```
interface=ath0 ❶
debug=1 ❷
ctrl_interface=/var/run/hostapd ❸
ctrl_interface_group=wheel ❹
ssid=freebsdap ❺
wpa=1 ❻
wpa_passphrase=freebsdmall ❼
wpa_key_mgmt=WPA-PSK ❽
wpa_pairwise=CCMP TKIP ❾
```

❶ This field indicates the wireless interface used for the Access Point.

❷ This field sets the level of verbosity during the execution of **hostapd**. A value of `1` represents the minimal level.

❸ The `ctrl_interface` field gives the pathname of the directory used by **hostapd** to stores its domain socket files for the communication with external programs such as hostapd_cli(8). The default value is used here.

❹ The `ctrl_interface_group` line sets the group (here, it is the `wheel` group) allowed to access to the control interface files.

❺ This field sets the network name.

❻ The `wpa` field enables WPA and specifies which WPA authentication protocol will be required. A value of `1` configures the AP for WPA-PSK.

❼ The `wpa_passphrase` field contains the ASCII passphrase for the WPA authentication.

> **Warning:** Always use strong passwords that are sufficiently long and made from a rich alphabet so they will not be guessed and/or attacked.

❽ The `wpa_key_mgmt` line refers to the key management protocol we use. In our case it is WPA-PSK.

❾ The `wpa_pairwise` field indicates the set of accepted encryption algorithms by the Access Point. Here both TKIP (WPA) and CCMP (WPA2) ciphers are accepted. CCMP cipher is an alternative to TKIP and that is strongly preferred when possible; TKIP should be used solely for stations incapable of doing CCMP.

The next step is to start **hostapd**:

```
# /etc/rc.d/hostapd forcestart

# ifconfig ath0
  ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2290
   inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
   inet6 fe80::211:95ff:fec3:dac%ath0 prefixlen 64 scopeid 0x4
   ether 00:11:95:c3:0d:ac
   media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
   status: associated
   ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
```

```
   authmode WPA2/802.11i privacy MIXED deftxkey 2 TKIP 2:128-bit txpowmax 36 protmode CTS dtimperi
```

The Access Point is running, the clients can now be associated with it, see Section 29.3.3.1.3 for more details. It is possible to see the stations associated with the AP using the `ifconfig ath0 list sta` command.

29.3.5.4 WEP Host-based Access Point

It is not recommended to use WEP for setting up an Access Point since there is no authentication mechanism and it is easily to be cracked. Some legacy wireless cards only support WEP as security protocol, these cards will only allow to set up AP without authentication or encryption or using the WEP protocol.

The wireless device can now be put into hostap mode and configured with the correct SSID and IP address:

```
# ifconfig ath0 ssid freebsdap wepmode on weptxkey 3 wepkey 3:0x3456789012 mode 11g mediaopt hostap \
 inet 192.168.0.1 netmask 255.255.255.0
```

- The `weptxkey` means which WEP key will be used in the transmission. Here we used the third key (note that the key numbering starts with `1`). This parameter must be specified to really encrypt the data.
- The `wepkey` means setting the selected WEP key. It should in the format *`index:key`*, if the index is not given, key `1` is set. That is to say we need to set the index if we use keys other than the first key.

Use again `ifconfig` to see the status of the `ath0` interface:

```
# ifconfig ath0
  ath0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
   inet6 fe80::211:95ff:fec3:dac%ath0 prefixlen 64 scopeid 0x4
   ether 00:11:95:c3:0d:ac
   media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
   status: associated
   ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
   authmode OPEN privacy ON deftxkey 3 wepkey 3:40-bit txpowmax 36 protmode CTS dtimperiod 1 bintv
```

From another wireless machine, it is possible to initiate a scan to find the AP:

```
# ifconfig ath0 up scan
SSID            BSSID              CHAN RATE  S:N   INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M 22:1   100 EPS
```

The client machine found the Access Point and can be associated with it using the correct parameters (key, etc.), see Section 29.3.3.1.4 for more details.

29.3.6 Troubleshooting

If you are having trouble with wireless networking, there are a number of steps you can take to help troubleshoot the problem.

- If you do not see the access point listed when scanning be sure you have not configured your wireless device to a limited set of channels.
- If you cannot associate to an access point verify the configuration of your station matches the one of the access point. This includes the authentication scheme and any security protocols. Simplify your configuration as much as possible. If you are using a security protocol such as WPA or WEP configure the access point for open authentication and no security to see if you can get traffic to pass.
- Once you can associate to the access point diagnose any security configuration using simple tools like ping(8).

  The `wpa_supplicant` has much debugging support; try running it manually with the `-dd` option and look at the system logs.
- There are also many lower-level debugging tools. You can enable debugging messages in the 802.11 protocol support layer using the `wlandebug` program found in `/usr/src/tools/tools/net80211`. For example:

  ```
  # wlandebug -i ath0 +scan+auth+debug+assoc
    net.wlan.0.debug: 0 => 0xc80000<assoc,auth,scan>
  ```

  can be used to enable console messages related to scanning for access points and doing the 802.11 protocol handshakes required to arrange communication.

  There are also many useful statistics maintained by the 802.11 layer; the `wlanstats` tool will dump these informations. These statistics should identify all errors identified by the 802.11 layer. Beware however that some errors are identified in the device drivers that lie below the 802.11 layer so they may not show up. To diagnose device-specific problems you need to refer to the drivers' documentation.

If the above information does not help to clarify the problem, please submit a problem report and include output from the above tools.

29.4 Bluetooth

Written by Pav Lucistnik.

29.4.1 Introduction

Bluetooth is a wireless technology for creating personal networks operating in the 2.4 GHz unlicensed band, with a range of 10 meters. Networks are usually formed ad-hoc from portable devices such as cellular phones, handhelds and laptops. Unlike the other popular wireless technology, Wi-Fi, Bluetooth offers higher level service profiles, e.g. FTP-like file servers, file pushing, voice transport, serial line emulation, and more.

The Bluetooth stack in FreeBSD is implemented using the Netgraph framework (see netgraph(4)). A broad variety of Bluetooth USB dongles is supported by the ng_ubt(4) driver. The Broadcom BCM2033 chip based Bluetooth devices are supported via the ubtbcmfw(4) and ng_ubt(4) drivers. The 3Com Bluetooth PC Card 3CRWB60-A is supported by the ng_bt3c(4) driver. Serial and UART based Bluetooth devices are supported via sio(4), ng_h4(4) and hcseriald(8). This section describes the use of the USB Bluetooth dongle.

29.4.2 Plugging in the Device

By default Bluetooth device drivers are available as kernel modules. Before attaching a device, you will need to load the driver into the kernel:

```
# kldload ng_ubt
```

If the Bluetooth device is present in the system during system startup, load the module from `/boot/loader.conf`:

```
ng_ubt_load="YES"
```

Plug in your USB dongle. The output similar to the following will appear on the console (or in syslog):

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

> **Note:** The Bluetooth stack has to be started manually on FreeBSD 6.0, and on FreeBSD 5.X before 5.5. It is done automatically from devd(8) on FreeBSD 5.5, 6.1 and newer.
>
> Copy `/usr/share/examples/netgraph/bluetooth/rc.bluetooth` into some convenient place, like `/etc/rc.bluetooth`. This script is used to start and stop the Bluetooth stack. It is a good idea to stop the stack before unplugging the device, but it is not (usually) fatal. When starting the stack, you will receive output similar to the following:
>
> ```
> # /etc/rc.bluetooth start ubt0
> BD_ADDR: 00:02:72:00:d4:1a
> Features: 0xff 0xff 0xf 00 00 00 00 00
> <3-Slot> <5-Slot> <Encryption> <Slot offset>
> <Timing accuracy> <Switch> <Hold mode> <Sniff mode>
> <Park mode> <RSSI> <Channel quality> <SCO link>
> <HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
> <Paging scheme> <Power control> <Transparent SCO data>
> Max. ACL packet size: 192 bytes
> Number of ACL packets: 8
> Max. SCO packet size: 64 bytes
> Number of SCO packets: 8
> ```

29.4.3 Host Controller Interface (HCI)

Host Controller Interface (HCI) provides a command interface to the baseband controller and link manager, and access to hardware status and control registers. This interface provides a uniform method of accessing the Bluetooth baseband capabilities. HCI layer on the Host exchanges data and commands with the HCI firmware on the Bluetooth hardware. The Host Controller Transport Layer (i.e. physical bus) driver provides both HCI layers with the ability to exchange information with each other.

A single Netgraph node of type *hci* is created for a single Bluetooth device. The HCI node is normally connected to the Bluetooth device driver node (downstream) and the L2CAP node (upstream). All HCI operations must be performed on the HCI node and not on the device driver node. Default name for the HCI node is “devicehci”. For more details refer to the ng_hci(4) manual page.

One of the most common tasks is discovery of Bluetooth devices in RF proximity. This operation is called *inquiry*. Inquiry and other HCI related operations are done with the hccontrol(8) utility. The example below shows how to find out which Bluetooth devices are in range. You should receive the list of devices in a few seconds. Note that a remote device will only answer the inquiry if it put into *discoverable* mode.

```
% hccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
       BD_ADDR: 00:80:37:29:19:a4
       Page Scan Rep. Mode: 0x1
       Page Scan Period Mode: 00
       Page Scan Mode: 00
       Class: 52:02:04
       Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

BD_ADDR is unique address of a Bluetooth device, similar to MAC addresses of a network card. This address is needed for further communication with a device. It is possible to assign human readable name to a BD_ADDR. The /etc/bluetooth/hosts file contains information regarding the known Bluetooth hosts. The following example shows how to obtain human readable name that was assigned to the remote device:

```
% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39
```

If you perform an inquiry on a remote Bluetooth device, it will find your computer as "your.host.name (ubt0)". The name assigned to the local device can be changed at any time.

The Bluetooth system provides a point-to-point connection (only two Bluetooth units involved), or a point-to-multipoint connection. In the point-to-multipoint connection the connection is shared among several Bluetooth devices. The following example shows how to obtain the list of active baseband connections for the local device:

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN
```

A *connection handle* is useful when termination of the baseband connection is required. Note, that it is normally not required to do it by hand. The stack will automatically terminate inactive baseband connections.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

Refer to hccontrol help for a complete listing of available HCI commands. Most of the HCI commands do not require superuser privileges.

29.4.4 Logical Link Control and Adaptation Protocol (L2CAP)

Logical Link Control and Adaptation Protocol (L2CAP) provides connection-oriented and connectionless data services to upper layer protocols with protocol multiplexing capability and segmentation and reassembly operation. L2CAP permits higher level protocols and applications to transmit and receive L2CAP data packets up to 64 kilobytes in length.

L2CAP is based around the concept of *channels*. Channel is a logical connection on top of baseband connection. Each channel is bound to a single protocol in a many-to-one fashion. Multiple channels can be bound to the same

protocol, but a channel cannot be bound to multiple protocols. Each L2CAP packet received on a channel is directed to the appropriate higher level protocol. Multiple channels can share the same baseband connection.

A single Netgraph node of type *l2cap* is created for a single Bluetooth device. The L2CAP node is normally connected to the Bluetooth HCI node (downstream) and Bluetooth sockets nodes (upstream). Default name for the L2CAP node is "devicel2cap". For more details refer to the ng_l2cap(4) manual page.

A useful command is l2ping(8), which can be used to ping other devices. Some Bluetooth implementations might not return all of the data sent to them, so `0 bytes` in the following example is normal.

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

The l2control(8) utility is used to perform various operations on L2CAP nodes. This example shows how to obtain the list of logical connections (channels) and the list of baseband connections for the local device:

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca    66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR    Handle Flags Pending State
00:07:e0:00:0b:ca     41 O           0 OPEN
```

Another diagnostic tool is btsockstat(1). It does a job similar to as netstat(1) does, but for Bluetooth network-related data structures. The example below shows the same logical connection as l2control(8) above.

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM       Foreign address   CID   State
c2afe900      0      0 00:02:72:00:d4:1a/3     00:07:e0:00:0b:ca 66    OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127   0     Yes  OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address     Foreign address   Chan DLCI State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6    OPEN
```

29.4.5 RFCOMM Protocol

The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10. RFCOMM is a simple transport protocol, with additional provisions for emulating the 9 circuits of RS-232 (EIATIA-232-E) serial ports. The RFCOMM protocol supports up to 60 simultaneous connections (RFCOMM channels) between two Bluetooth devices.

For the purposes of RFCOMM, a complete communication path involves two applications running on different devices (the communication endpoints) with a communication segment between them. RFCOMM is intended to

cover applications that make use of the serial ports of the devices in which they reside. The communication segment is a Bluetooth link from one device to another (direct connect).

RFCOMM is only concerned with the connection between the devices in the direct connect case, or between the device and a modem in the network case. RFCOMM can support other configurations, such as modules that communicate via Bluetooth wireless technology on one side and provide a wired interface on the other side.

In FreeBSD the RFCOMM protocol is implemented at the Bluetooth sockets layer.

29.4.6 Pairing of Devices

By default, Bluetooth communication is not authenticated, and any device can talk to any other device. A Bluetooth device (for example, cellular phone) may choose to require authentication to provide a particular service (for example, Dial-Up service). Bluetooth authentication is normally done with *PIN codes*. A PIN code is an ASCII string up to 16 characters in length. User is required to enter the same PIN code on both devices. Once user has entered the PIN code, both devices will generate a *link key*. After that the link key can be stored either in the devices themselves or in a persistent storage. Next time both devices will use previously generated link key. The described above procedure is called *pairing*. Note that if the link key is lost by any device then pairing must be repeated.

The hcsecd(8) daemon is responsible for handling of all Bluetooth authentication requests. The default configuration file is `/etc/bluetooth/hcsecd.conf`. An example section for a cellular phone with the PIN code arbitrarily set to “1234” is shown below:

```
device {
        bdaddr  00:80:37:29:19:a4;
        name    "Pav's T39";
        key     nokey;
        pin     "1234";
      }
```

There is no limitation on PIN codes (except length). Some devices (for example Bluetooth headsets) may have a fixed PIN code built in. The `-d` switch forces the hcsecd(8) daemon to stay in the foreground, so it is easy to see what is happening. Set the remote device to receive pairing and initiate the Bluetooth connection to the remote device. The remote device should say that pairing was accepted, and request the PIN code. Enter the same PIN code as you have in `hcsecd.conf`. Now your PC and the remote device are paired. Alternatively, you can initiate pairing on the remote device.

On FreeBSD 5.5, 6.1 and newer, the following line can be added to the `/etc/rc.conf` file to have **hcsecd** started automatically on system start:

```
hcsecd_enable="YES"
```

The following is a sample of the **hcsecd** daemon output:

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', link key do
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', PIN code ex
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
```

29.4.7 Service Discovery Protocol (SDP)

The Service Discovery Protocol (SDP) provides the means for client applications to discover the existence of services provided by server applications as well as the attributes of those services. The attributes of a service include the type or class of service offered and the mechanism or protocol information needed to utilize the service.

SDP involves communication between a SDP server and a SDP client. The server maintains a list of service records that describe the characteristics of services associated with the server. Each service record contains information about a single service. A client may retrieve information from a service record maintained by the SDP server by issuing a SDP request. If the client, or an application associated with the client, decides to use a service, it must open a separate connection to the service provider in order to utilize the service. SDP provides a mechanism for discovering services and their attributes, but it does not provide a mechanism for utilizing those services.

Normally, a SDP client searches for services based on some desired characteristics of the services. However, there are times when it is desirable to discover which types of services are described by an SDP server's service records without any a priori information about the services. This process of looking for any offered services is called *browsing*.

The Bluetooth SDP server sdpd(8) and command line client sdpcontrol(8) are included in the standard FreeBSD installation. The following example shows how to perform a SDP browse query.

```
% sdpcontrol -a 00:01:03:fc:6e:ec browse
Record Handle: 00000000
Service Class ID List:
        Service Discovery Server (0x1000)
Protocol Descriptor List:
        L2CAP (0x0100)
                Protocol specific parameter #1: u/int/uuid16 1
                Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
        Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
        LAN Access Using PPP (0x1102)
Protocol Descriptor List:
        L2CAP (0x0100)
        RFCOMM (0x0003)
                Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
        LAN Access Using PPP (0x1102) ver. 1.0
```

... and so on. Note that each service has a list of attributes (RFCOMM channel for example). Depending on the service you might need to make a note of some of the attributes. Some Bluetooth implementations do not support service browsing and may return an empty list. In this case it is possible to search for the specific service. The example below shows how to search for the OBEX Object Push (OPUSH) service:

```
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

Offering services on FreeBSD to Bluetooth clients is done with the sdpd(8) server. On FreeBSD 5.5, 6.1 and newer, the following line can be added to the `/etc/rc.conf` file:

```
sdpd_enable="YES"
```

Then the **sdpd** daemon can be started with:

```
# /etc/rc.d/sdpd start
```

On FreeBSD 6.0, and on FreeBSD 5.X before 5.5, **sdpd** is not integrated into the system startup scripts. It has to be started manually with:

```
# sdpd
```

The local server application that wants to provide Bluetooth service to the remote clients will register service with the local SDP daemon. The example of such application is rfcomm_pppd(8). Once started it will register Bluetooth LAN service with the local SDP daemon.

The list of services registered with the local SDP server can be obtained by issuing SDP browse query via local control channel:

```
# sdpcontrol -l browse
```

29.4.8 Dial-Up Networking (DUN) and Network Access with PPP (LAN) Profiles

The Dial-Up Networking (DUN) profile is mostly used with modems and cellular phones. The scenarios covered by this profile are the following:

- use of a cellular phone or modem by a computer as a wireless modem for connecting to a dial-up Internet access server, or using other dial-up services;
- use of a cellular phone or modem by a computer to receive data calls.

Network Access with PPP (LAN) profile can be used in the following situations:

- LAN access for a single Bluetooth device;
- LAN access for multiple Bluetooth devices;
- PC to PC (using PPP networking over serial cable emulation).

In FreeBSD both profiles are implemented with ppp(8) and rfcomm_pppd(8) - a wrapper that converts RFCOMM Bluetooth connection into something PPP can operate with. Before any profile can be used, a new PPP label in the `/etc/ppp/ppp.conf` must be created. Consult rfcomm_pppd(8) manual page for examples.

In the following example rfcomm_pppd(8) will be used to open RFCOMM connection to remote device with BD_ADDR 00:80:37:29:19:a4 on DUN RFCOMM channel. The actual RFCOMM channel number will be obtained from the remote device via SDP. It is possible to specify RFCOMM channel by hand, and in this case rfcomm_pppd(8) will not perform SDP query. Use sdpcontrol(8) to find out RFCOMM channel on the remote device.

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

In order to provide Network Access with PPP (LAN) service the sdpd(8) server must be running. A new entry for LAN clients must be created in the `/etc/ppp/ppp.conf` file. Consult rfcomm_pppd(8) manual page for examples. Finally, start RFCOMM PPP server on valid RFCOMM channel number. The RFCOMM PPP server will

automatically register Bluetooth LAN service with the local SDP daemon. The example below shows how to start RFCOMM PPP server.

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

29.4.9 OBEX Object Push (OPUSH) Profile

OBEX is a widely used protocol for simple file transfers between mobile devices. Its main use is in infrared communication, where it is used for generic file transfers between notebooks or PDAs, and for sending business cards or calendar entries between cellular phones and other devices with PIM applications.

The OBEX server and client are implemented as a third-party package **obexapp**, which is available as `comms/obexapp` port.

OBEX client is used to push and/or pull objects from the OBEX server. An object can, for example, be a business card or an appointment. The OBEX client can obtain RFCOMM channel number from the remote device via SDP. This can be done by specifying service name instead of RFCOMM channel number. Supported service names are: IrMC, FTRN and OPUSH. It is possible to specify RFCOMM channel as a number. Below is an example of an OBEX session, where device information object is pulled from the cellular phone, and a new object (business card) is pushed into the phone's directory.

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get telecom/devinfo.txt devinfo-t39.txt
Success, response: OK, Success (0x20)
obex> put new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

In order to provide OBEX Object Push service, sdpd(8) server must be running. A root folder, where all incoming objects will be stored, must be created. The default path to the root folder is `/var/spool/obex`. Finally, start OBEX server on valid RFCOMM channel number. The OBEX server will automatically register OBEX Object Push service with the local SDP daemon. The example below shows how to start OBEX server.

```
# obexapp -s -C 10
```

29.4.10 Serial Port Profile (SPP)

The Serial Port Profile (SPP) allows Bluetooth devices to perform RS232 (or similar) serial cable emulation. The scenario covered by this profile deals with legacy applications using Bluetooth as a cable replacement, through a virtual serial port abstraction.

The rfcomm_sppd(1) utility implements the Serial Port profile. A pseudo tty is used as a virtual serial port abstraction. The example below shows how to connect to a remote device Serial Port service. Note that you do not have to specify a RFCOMM channel - rfcomm_sppd(1) can obtain it from the remote device via SDP. If you would like to override this, specify a RFCOMM channel on the command line.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t /dev/ttyp6
rfcomm_sppd[94692]: Starting on /dev/ttyp6...
```

Once connected, the pseudo tty can be used as serial port:

```
# cu -l ttyp6
```

29.4.11 Troubleshooting

29.4.11.1 A remote device cannot connect

Some older Bluetooth devices do not support role switching. By default, when FreeBSD is accepting a new connection, it tries to perform a role switch and become master. Devices, which do not support this will not be able to connect. Note that role switching is performed when a new connection is being established, so it is not possible to ask the remote device if it does support role switching. There is a HCI option to disable role switching on the local side:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

29.4.11.2 Something is going wrong, can I see what exactly is happening?

Yes, you can. Use the third-party package **hcidump**, which is available as `comms/hcidump` port. The **hcidump** utility is similar to tcpdump(1). It can be used to display the content of the Bluetooth packets on the terminal and to dump the Bluetooth packets to a file.

29.5 Bridging

Written by Andrew Thompson.

29.5.1 Introduction

It is sometimes useful to divide one physical network (such as an Ethernet segment) into two separate network segments without having to create IP subnets and use a router to connect the segments together. A device that connects two networks together in this fashion is called a “bridge”. A FreeBSD system with two network interface cards can act as a bridge.

The bridge works by learning the MAC layer addresses (Ethernet addresses) of the devices on each of its network interfaces. It forwards traffic between two networks only when its source and destination are on different networks.

In many respects, a bridge is like an Ethernet switch with very few ports.

29.5.2 Situations Where Bridging Is Appropriate

There are many common situations in which a bridge is used today.

29.5.2.1 Connecting Networks

The basic operation of a bridge is to join two or more network segments together. There are many reasons to use a host based bridge over plain networking equipment such as cabling constraints, firewalling or connecting pseudo

networks such as a Virtual Machine interface. A bridge can also connect a wireless interface running in hostap mode to a wired network and act as an access point.

29.5.2.2 Filtering/Traffic Shaping Firewall

A common situation is where firewall functionality is needed without routing or network address translation (NAT).

An example is a small company that is connected via DSL or ISDN to their ISP. They have a 13 globally-accessible IP addresses from their ISP and have 10 PCs on their network. In this situation, using a router-based firewall is difficult because of subnetting issues.

A bridge-based firewall can be configured and dropped into the path just downstream of their DSL/ISDN router without any IP numbering issues.

29.5.2.3 Network Tap

A bridge can join two network segments and be used to inspect all Ethernet frames that pass between them. This can either be from using bpf(4)/tcpdump(1) on the bridge interface or by sending a copy of all frames out an additional interface (span port).

29.5.2.4 Layer 2 VPN

Two Ethernet networks can be joined across an IP link by bridging the networks to an EtherIP tunnel or a tap(4) based solution such as OpenVPN.

29.5.2.5 Layer 2 Redundancy

A network can be connected together with multiple links and use the Spanning Tree Protocol to block redundant paths. For an Ethernet network to function properly only one active path can exist between two devices, Spanning Tree will detect loops and put the redundant links into a blocked state. Should one of the active links fail then the protocol will calculate a different tree and reenable one of the blocked paths to restore connectivity to all points in the network.

29.5.3 Kernel Configuration

This section covers if_bridge(4) bridge implementation, a netgraph bridging driver is also available, for more information see ng_bridge(4) manual page.

The bridge driver is a kernel module and will be automatically loaded by ifconfig(8) when creating a bridge interface. It is possible to compile the bridge in to the kernel by adding `device if_bridge` to your kernel configuration file.

Packet filtering can be used with any firewall package that hooks in via the pfil(9) framework. The firewall can be loaded as a module or compiled into the kernel.

The bridge can be used as a traffic shaper with altq(4) or dummynet(4).

29.5.4 Enabling the Bridge

The bridge is created using interface cloning. To create a bridge use ifconfig(8), if the bridge driver is not present in the kernel then it will be loaded automatically.

```
# ifconfig bridge create
bridge0
# ifconfig bridge0
bridge0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:00:00:00:00:00 priority 0 ifcost 0 port 0
```

A bridge interface is created and is automatically assigned a randomly generated Ethernet address. The `maxaddr` and `timeout` parameters control how many MAC addresses the bridge will keep in its forwarding table and how many seconds before each entry is removed after it is last seen. The other parameters control how Spanning Tree operates.

Add the member network interfaces to the bridge. For the bridge to forward packets all member interfaces and the bridge need to be up:

```
# ifconfig bridge0 addm fxp0 addm fxp1 up
# ifconfig fxp0 up
# ifconfig fxp1 up
```

The bridge is now forwarding Ethernet frames between `fxp0` and `fxp1`. The equivalent configuration in `/etc/rc.conf` so the bridge is created at startup is:

```
cloned_interfaces="bridge0"
ifconfig_bridge0="addm fxp0 addm fxp1 up"
ifconfig_fxp0="up"
ifconfig_fxp1="up"
```

If the bridge host needs an IP address then the correct place to set this is on the bridge interface itself rather than one of the member interfaces. This can be set statically or via DHCP:

```
# ifconfig bridge0 inet 192.168.0.1/24
```

It is also possible to assign an IPv6 address to a bridge interface.

29.5.5 Firewalling

When packet filtering is enabled, bridged packets will pass through the filter inbound on the originating interface, on the bridge interface and outbound on the appropriate interfaces. Either stage can be disabled. When direction of the packet flow is important it is best to firewall on the member interfaces rather than the bridge itself.

The bridge has several configurable settings for passing non-IP and ARP packets, and layer2 firewalling with IPFW. See if_bridge(4) for more information.

29.5.6 Spanning Tree

The bridge driver implements the Rapid Spanning Tree Protocol (RSTP or 802.1w) with backwards compatibility with the legacy Spanning Tree Protocol (STP). Spanning Tree is used to detect and remove loops in a network topology. RSTP provides faster Spanning Tree convergence than legacy STP, the protocol will exchange information with neighbouring switches to quickly transition to forwarding without creating loops.

The following table shows the supported operating modes:

OS Version	STP Modes	Default Mode
FreeBSD 5.4——FreeBSD 6.2	STP	STP
FreeBSD 6.3+	RSTP or STP	STP
FreeBSD 7.0+	RSTP or STP	RSTP

Spanning Tree can be enabled on member interfaces using the `stp` command. For a bridge with `fxp0` and `fxp1` as the current interfaces, enable STP with the following:

```
# ifconfig bridge0 stp fxp0 stp fxp1
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether d6:cf:d5:a0:94:6d
        id 00:01:02:4b:d4:50 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 0 port 0
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 3 priority 128 path cost 200000 proto rstp
                role designated state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

This bridge has a spanning tree ID of `00:01:02:4b:d4:50` and a priority of `32768`. As the `root id` is the same it indicates that this is the root bridge for the tree.

Another bridge on the network also has spanning tree enabled:

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:13:d4:9a:06:7a priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role root state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 5 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

The line `root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4` shows that the root bridge is `00:01:02:4b:d4:50` as above and has a path cost of `400000` from this bridge, the path to the root bridge is via `port 4` which is `fxp0`.

29.5.7 Advanced Bridging

29.5.7.1 Reconstruct Traffic Flows

The bridge supports monitor mode, where the packets are discarded after bpf(4) processing, and are not processed or forwarded further. This can be used to multiplex the input of two or more interfaces into a single bpf(4) stream. This is useful for reconstructing the traffic for network taps that transmit the RX/TX signals out through two separate interfaces.

To read the input from four network interfaces as one stream:

```
# ifconfig bridge0 addm fxp0 addm fxp1 addm fxp2 addm fxp3 monitor up
# tcpdump -i bridge0
```

29.5.7.2 Span Ports

A copy of every Ethernet frame received by the bridge will be transmitted out a designated span port. The number of span ports configured on a bridge is unlimited, if an interface is designated as a span port then it may not also be used as a regular bridge port. This is most useful for snooping a bridged network passively on another host connected to one of the span ports of the bridge.

To send a copy of all frames out the interface named `fxp4`:

```
# ifconfig bridge0 span fxp4
```

29.5.7.3 Private Interfaces

A private interface does not forward any traffic to any other port that is also a private interface. The traffic is blocked unconditionally so no Ethernet frames will be forwarded, including ARP. If traffic needs to be selectively blocked then a firewall should be used instead.

29.5.7.4 Sticky Interfaces

If a bridge member interface is marked as sticky then dynamically learned address entries are treated at static once entered into the forwarding cache. Sticky entries are never aged out of the cache or replaced, even if the address is seen on a different interface. This gives the benefit of static address entries without the need to pre-populate the forwarding table, clients learnt on a particular segment of the bridge can not roam to another segment.

Another example of using sticky addresses would be to combine the bridge with VLANs to create a router where customer networks are isolated without wasting IP address space. Consider that `CustomerA` is on `vlan100` and `CustomerB` is on `vlan101`. The bridge has the address `192.168.0.1` and is also an internet router.

```
# ifconfig bridge0 addm vlan100 sticky vlan100 addm vlan101 sticky vlan101
# ifconfig bridge0 inet 192.168.0.1/24
```

Both clients see `192.168.0.1` as their default gateway and since the bridge cache is sticky they can not spoof the MAC address of the other customer to intercept their traffic.

Any communication between the VLANs can be blocked using private interfaces (or a firewall):

```
# ifconfig bridge0 private vlan100 private vlan101
```

The customers are completely isolated from each other, the full `/24` address range can be allocated without subnetting.

29.5.7.5 SNMP Monitoring

The bridge interface and STP parameters can be monitored via the SNMP daemon which is included in the FreeBSD base system. The exported bridge MIBs conform to the IETF standards so any SNMP client or monitoring package can be used to retrieve the data.

On the bridge machine uncomment the `begemotSnmpdModulePath."bridge" = "/usr/lib/snmp_bridge.so"` line from `/etc/snmp.config` and start the **bsnmpd** daemon. Other configuration such as community names and access lists may need to be modified. See bsnmpd(1) and snmp_bridge(3) for more information.

The following examples use the **Net-SNMP** software (`net-mgmt/net-snmp`) to query a bridge, the `net-mgmt/bsnmptools` port can also be used. From the SNMP client host add to `$HOME/.snmp/snmp.conf` the following lines to import the bridge MIB definitions in to **Net-SNMP**:

```
mibdirs +/usr/share/snmp/mibs
mibs +BRIDGE-MIB:RSTP-MIB:BEGEMOT-MIB:BEGEMOT-BRIDGE-MIB
```

To monitor a single bridge via the IETF BRIDGE-MIB (RFC4188) do

```
% snmpwalk -v 2c -c public bridge1.example.com mib-2.dot1dBridge
BRIDGE-MIB::dot1dBaseBridgeAddress.0 = STRING: 66:fb:9b:6e:5c:44
BRIDGE-MIB::dot1dBaseNumPorts.0 = INTEGER: 1 ports
BRIDGE-MIB::dot1dStpTimeSinceTopologyChange.0 = Timeticks: (189959) 0:31:39.59 centi-seconds
BRIDGE-MIB::dot1dStpTopChanges.0 = Counter32: 2
BRIDGE-MIB::dot1dStpDesignatedRoot.0 = Hex-STRING: 80 00 00 01 02 4B D4 50
...
BRIDGE-MIB::dot1dStpPortState.3 = INTEGER: forwarding(5)
BRIDGE-MIB::dot1dStpPortEnable.3 = INTEGER: enabled(1)
BRIDGE-MIB::dot1dStpPortPathCost.3 = INTEGER: 200000
BRIDGE-MIB::dot1dStpPortDesignatedRoot.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedCost.3 = INTEGER: 0
BRIDGE-MIB::dot1dStpPortDesignatedBridge.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedPort.3 = Hex-STRING: 03 80
BRIDGE-MIB::dot1dStpPortForwardTransitions.3 = Counter32: 1
RSTP-MIB::dot1dStpVersion.0 = INTEGER: rstp(2)
```

The `dot1dStpTopChanges.0` value is two which means that the STP bridge topology has changed twice, a topology change means that one or more links in the network have changed or failed and a new tree has been calculated. The `dot1dStpTimeSinceTopologyChange.0` value will show when this happened.

To monitor multiple bridge interfaces one may use the private BEGEMOT-BRIDGE-MIB:

```
% snmpwalk -v 2c -c public bridge1.example.com
enterprises.fokus.begemot.begemotBridge
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge0" = STRING: bridge0
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge2" = STRING: bridge2
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge0" = STRING: e:ce:3b:5a:9e:13
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge2" = STRING: 12:5e:4d:74:d:fc
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge0" = INTEGER: 1
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge2" = INTEGER: 1
...
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge0" = Timeticks: (116927) 0:19:1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge2" = Timeticks: (82773) 0:13:47
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge0" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge2" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge0" = Hex-STRING: 80 00 00 40 95 30 5E 31
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge2" = Hex-STRING: 80 00 00 50 8B B8 C6 A9
```

To change the bridge interface being monitored via the `mib-2.dot1dBridge` subtree do:

```
% snmpset -v 2c -c private bridge1.example.com
BEGEMOT-BRIDGE-MIB::begemotBridgeDefaultBridgeIf.0 s bridge2
```

29.6 Link Aggregation and Failover

Written by Andrew Thompson.

29.6.1 Introduction

The lagg(4) interface allows aggregation of multiple network interfaces as one virtual interface for the purpose of providing fault-tolerance and high-speed links.

29.6.2 Operating Modes

failover

Sends and receives traffic only through the master port. If the master port becomes unavailable, the next active port is used. The first interface added is the master port; any interfaces added after that are used as failover devices.

fec

Supports Cisco EtherChannel. This is a static setup and does not negotiate aggregation with the peer or exchange frames to monitor the link, if the switch supports LACP then that should be used instead.

Balances outgoing traffic across the active ports based on hashed protocol header information and accepts incoming traffic from any active port. The hash includes the Ethernet source and destination address, and, if available, the VLAN tag, and the IPv4/IPv6 source and destination address.

lacp

Supports the IEEE 802.3ad Link Aggregation Control Protocol (LACP) and the Marker Protocol. LACP will negotiate a set of aggregable links with the peer in to one or more Link Aggregated Groups. Each LAG is composed of ports of the same speed, set to full-duplex operation. The traffic will be balanced across the ports in the LAG with the greatest total speed, in most cases there will only be one LAG which contains all ports. In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration.

Balances outgoing traffic across the active ports based on hashed protocol header information and accepts incoming traffic from any active port. The hash includes the Ethernet source and destination address, and, if available, the VLAN tag, and the IPv4/IPv6 source and destination address.

loadbalance

This is an alias of *fec* mode.

roundrobin

Distributes outgoing traffic using a round-robin scheduler through all active ports and accepts incoming traffic from any active port. This mode will violate Ethernet frame ordering and should be used with caution.

29.6.3 Examples

Example 29-1. LACP aggregation with a Cisco switch

This example connects two interfaces on a FreeBSD machine to the switch as a single load balanced and fault tolerant link. More interfaces can be added to increase throughput and fault tolerance. Since frame ordering is mandatory on Ethernet links then any traffic between two stations always flows over the same physical link limiting the maximum speed to that of one interface. The transmit algorithm attempts to use as much information as it can to distinguish different traffic flows and balance across the available interfaces.

On the Cisco switch add the interfaces to the channel group.

```
interface FastEthernet0/1
 channel-group 1 mode active
 channel-protocol lacp
!
interface FastEthernet0/2
 channel-group 1 mode active
 channel-protocol lacp
!
```

On the FreeBSD machine create the lagg interface.

```
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto lacp laggport fxp0 laggport fxp1
```

View the interface status from ifconfig; ports marked as *ACTIVE* are part of the active aggregation group that has been negotiated with the remote switch and traffic will be transmitted and received. Use the verbose output of ifconfig(8) to view the LAG identifiers.

```
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        media: Ethernet autoselect
        status: active
        laggproto lacp
        laggport: fxp1 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
        laggport: fxp0 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
```

The switch will show which ports are active. For more detail use **`show lacp neighbor detail`**.

```
switch# show lacp neighbor
```

```
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors

Partner's information:

                  LACP port                         Oper    Port     Port
Port      Flags   Priority  Dev ID          Age     Key     Number   State
Fa0/1     SA      32768     0005.5d71.8db8  29s     0x146   0x3      0x3D
Fa0/2     SA      32768     0005.5d71.8db8  29s     0x146   0x4      0x3D
```

Example 29-2. Failover mode

Failover mode can be used to switch over to another interface if the link is lost on the master.

```
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport fxp0 laggport fxp1
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        media: Ethernet autoselect
        status: active
        laggproto failover
        laggport: fxp1 flags=0<>
        laggport: fxp0 flags=5<MASTER,ACTIVE>
```

Traffic will be transmitted and received on `fxp0`. If the link is lost on `fxp0` then `fxp1` will become the active link. If the link is restored on the master interface then it will once again become the active link.

29.7 Diskless Operation

Updated by Jean-François Dockès. Reorganized and enhanced by Alex Dupre.

A FreeBSD machine can boot over the network and operate without a local disk, using file systems mounted from an NFS server. No system modification is necessary, beyond standard configuration files. Such a system is relatively easy to set up because all the necessary elements are readily available:

- There are at least two possible methods to load the kernel over the network:
  - PXE: The Intel Preboot eXecution Environment system is a form of smart boot ROM built into some networking cards or motherboards. See pxeboot(8) for more details.
  - The **Etherboot** port (`net/etherboot`) produces ROM-able code to boot kernels over the network. The code can be either burnt into a boot PROM on a network card, or loaded from a local floppy (or hard) disk drive, or from a running MS-DOS system. Many network cards are supported.

- A sample script (`/usr/share/examples/diskless/clone_root`) eases the creation and maintenance of the workstation's root file system on the server. The script will probably require a little customization but it will get you started very quickly.
- Standard system startup files exist in `/etc` to detect and support a diskless system startup.
- Swapping, if needed, can be done either to an NFS file or to a local disk.

There are many ways to set up diskless workstations. Many elements are involved, and most can be customized to suit local taste. The following will describe variations on the setup of a complete system, emphasizing simplicity and compatibility with the standard FreeBSD startup scripts. The system described has the following characteristics:

- The diskless workstations use a shared read-only `/` file system, and a shared read-only `/usr`.

  The root file system is a copy of a standard FreeBSD root (typically the server's), with some configuration files overridden by ones specific to diskless operation or, possibly, to the workstation they belong to.

  The parts of the root which have to be writable are overlaid with md(4) file systems. Any changes will be lost when the system reboots.
- The kernel is transferred and loaded either with **Etherboot** or PXE as some situations may mandate the use of either method.

> **Caution:** As described, this system is insecure. It should live in a protected area of a network, and be untrusted by other hosts.

All the information in this section has been tested using FreeBSD 5.2.1-RELEASE.

29.7.1 Background Information

Setting up diskless workstations is both relatively straightforward and prone to errors. These are sometimes difficult to diagnose for a number of reasons. For example:

- Compile time options may determine different behaviors at runtime.
- Error messages are often cryptic or totally absent.

In this context, having some knowledge of the background mechanisms involved is very useful to solve the problems that may arise.

Several operations need to be performed for a successful bootstrap:

- The machine needs to obtain initial parameters such as its IP address, executable filename, server name, root path. This is done using the DHCP or BOOTP protocols. DHCP is a compatible extension of BOOTP, and uses the same port numbers and basic packet format.

  It is possible to configure a system to use only BOOTP. The bootpd(8) server program is included in the base FreeBSD system.

  However, DHCP has a number of advantages over BOOTP (nicer configuration files, possibility of using PXE, plus many others not directly related to diskless operation), and we will describe mainly a DHCP configuration, with equivalent examples using bootpd(8) when possible. The sample configuration will use the **ISC DHCP** software package (release 3.0.1.r12 was installed on the test server).

- The machine needs to transfer one or several programs to local memory. Either TFTP or NFS are used. The choice between TFTP and NFS is a compile time option in several places. A common source of error is to specify filenames for the wrong protocol: TFTP typically transfers all files from a single directory on the server, and would expect filenames relative to this directory. NFS needs absolute file paths.
- The possible intermediate bootstrap programs and the kernel need to be initialized and executed. There are several important variations in this area:
  - PXE will load pxeboot(8), which is a modified version of the FreeBSD third stage loader. The loader(8) will obtain most parameters necessary to system startup, and leave them in the kernel environment before transferring control. It is possible to use a `GENERIC` kernel in this case.
  - **Etherboot**, will directly load the kernel, with less preparation. You will need to build a kernel with specific options.

  PXE and **Etherboot** work equally well; however, because kernels normally let the loader(8) do more work for them, PXE is the preferred method.

  If your BIOS and network cards support PXE, you should probably use it.
- Finally, the machine needs to access its file systems. NFS is used in all cases.

See also diskless(8) manual page.

29.7.2 Setup Instructions

29.7.2.1 Configuration Using ISC DHCP

The **ISC DHCP** server can answer both BOOTP and DHCP requests.

ISC DHCP 3.0 is not part of the base system. You will first need to install the `net/isc-dhcp3-server` port or the corresponding package.

Once **ISC DHCP** is installed, it needs a configuration file to run (normally named `/usr/local/etc/dhcpd.conf`). Here follows a commented example, where host `margaux` uses **Etherboot** and host `corbieres` uses PXE:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

option domain-name "example.com";
option domain-name-servers 192.168.4.1;
option routers 192.168.4.1;

subnet 192.168.4.0 netmask 255.255.255.0 {
  use-host-decl-names on; ❶
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.4.255;

  host margaux {
    hardware ethernet 01:23:45:67:89:ab;
    fixed-address margaux.example.com;
    next-server 192.168.4.4; ❷
```

```
    filename "/data/misc/kernel.diskless"; ❸
    option root-path "192.168.4.4:/data/misc/diskless"; ❹
  }
  host corbieres {
    hardware ethernet 00:02:b3:27:62:df;
    fixed-address corbieres.example.com;
    next-server 192.168.4.4;
    filename "pxeboot";
    option root-path "192.168.4.4:/data/misc/diskless";
  }
}
```

❶ This option tells **dhcpd** to send the value in the `host` declarations as the hostname for the diskless host. An alternate way would be to add an `option host-name` *`margaux`* inside the `host` declarations.

❷ The `next-server` directive designates the TFTP or NFS server to use for loading loader or kernel file (the default is to use the same host as the DHCP server).

❸ The `filename` directive defines the file that **Etherboot** or PXE will load for the next execution step. It must be specified according to the transfer method used. **Etherboot** can be compiled to use NFS or TFTP. The FreeBSD port configures NFS by default. PXE uses TFTP, which is why a relative filename is used here (this may depend on the TFTP server configuration, but would be fairly typical). Also, PXE loads `pxeboot`, not the kernel. There are other interesting possibilities, like loading `pxeboot` from a FreeBSD CD-ROM `/boot` directory (as pxeboot(8) can load a `GENERIC` kernel, this makes it possible to use PXE to boot from a remote CD-ROM).

❹ The `root-path` option defines the path to the root file system, in usual NFS notation. When using PXE, it is possible to leave off the host's IP as long as you do not enable the kernel option BOOTP. The NFS server will then be the same as the TFTP one.

29.7.2.2 Configuration Using BOOTP

Here follows an equivalent **bootpd** configuration (reduced to one client). This would be found in `/etc/bootptab`.

Please note that **Etherboot** must be compiled with the non-default option `NO_DHCP_SUPPORT` in order to use BOOTP, and that PXE *needs* DHCP. The only obvious advantage of **bootpd** is that it exists in the base system.

```
.def100:\
  :hn:ht=1:sa=192.168.4.4:vm=rfc1048:\
  :sm=255.255.255.0:\
  :ds=192.168.4.1:\
  :gw=192.168.4.1:\
  :hd="/tftpboot":\
  :bf="/kernel.diskless":\
  :rp="192.168.4.4:/data/misc/diskless":

margaux:ha=0123456789ab:tc=.def100
```

29.7.2.3 Preparing a Boot Program with Etherboot

Etherboot's Web site (http://etherboot.sourceforge.net) contains extensive documentation (http://etherboot.sourceforge.net/doc/html/userman/t1.html) mainly intended for Linux systems, but nonetheless containing useful information. The following will just outline how you would use **Etherboot** on a FreeBSD system.

You must first install the `net/etherboot` package or port.

You can change the **Etherboot** configuration (i.e. to use TFTP instead of NFS) by editing the `Config` file in the **Etherboot** source directory.

For our setup, we shall use a boot floppy. For other methods (PROM, or MS-DOS program), please refer to the **Etherboot** documentation.

To make a boot floppy, insert a floppy in the drive on the machine where you installed **Etherboot**, then change your current directory to the `src` directory in the **Etherboot** tree and type:

```
# gmake bin32/devicetype.fd0
```

`devicetype` depends on the type of the Ethernet card in the diskless workstation. Refer to the `NIC` file in the same directory to determine the right *`devicetype`*.

29.7.2.4 Booting with PXE

By default, the pxeboot(8) loader loads the kernel via NFS. It can be compiled to use TFTP instead by specifying the `LOADER_TFTP_SUPPORT` option in `/etc/make.conf`. See the comments in `/usr/share/examples/etc/make.conf` for instructions.

There are two other `make.conf` options which may be useful for setting up a serial console diskless machine: `BOOT_PXELDR_PROBE_KEYBOARD`, and `BOOT_PXELDR_ALWAYS_SERIAL`.

To use PXE when the machine starts, you will usually need to select the `Boot from network` option in your BIOS setup, or type a function key during the PC initialization.

29.7.2.5 Configuring the TFTP and NFS Servers

If you are using PXE or **Etherboot** configured to use TFTP, you need to enable **tftpd** on the file server:

1. Create a directory from which **tftpd** will serve the files, e.g. `/tftpboot`.
2. Add this line to your `/etc/inetd.conf`:

   ```
   tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /tftpboot
   ```

   > **Note:** It appears that at least some PXE versions want the TCP version of TFTP. In this case, add a second line, replacing `dgram udp` with `stream tcp`.

3. Tell **inetd** to reread its configuration file. The `inetd_enable="YES"` must be in the `/etc/rc.conf` file for this command to execute correctly:

   ```
   # /etc/rc.d/inetd restart
   ```

You can place the `tftpboot` directory anywhere on the server. Make sure that the location is set in both `inetd.conf` and `dhcpd.conf`.

In all cases, you also need to enable NFS and export the appropriate file system on the NFS server.

1. Add this to `/etc/rc.conf`:

   ```
   nfs_server_enable="YES"
   ```

2. Export the file system where the diskless root directory is located by adding the following to `/etc/exports` (adjust the volume mount point and replace *`margaux corbieres`* with the names of the diskless workstations):

   ```
   /data/misc -alldirs -ro margaux corbieres
   ```

3. Tell **mountd** to reread its configuration file. If you actually needed to enable NFS in `/etc/rc.conf` at the first step, you probably want to reboot instead.

   ```
   # /etc/rc.d/mountd restart
   ```

29.7.2.6 Building a Diskless Kernel

If using **Etherboot**, you need to create a kernel configuration file for the diskless client with the following options (in addition to the usual ones):

```
options     BOOTP          # Use BOOTP to obtain IP address/hostname
options     BOOTP_NFSROOT  # NFS mount root file system using BOOTP info
```

You may also want to use `BOOTP_NFSV3`, `BOOT_COMPAT` and `BOOTP_WIRED_TO` (refer to `NOTES`).

These option names are historical and slightly misleading as they actually enable indifferent use of DHCP and BOOTP inside the kernel (it is also possible to force strict BOOTP or DHCP use).

Build the kernel (see Chapter 8), and copy it to the place specified in `dhcpd.conf`.

> **Note:** When using PXE, building a kernel with the above options is not strictly necessary (though suggested). Enabling them will cause more DHCP requests to be issued during kernel startup, with a small risk of inconsistency between the new values and those retrieved by pxeboot(8) in some special cases. The advantage of using them is that the host name will be set as a side effect. Otherwise you will need to set the host name by another method, for example in a client-specific `rc.conf` file.

> **Note:** In order to be loadable with **Etherboot**, a kernel needs to have the device hints compiled in. You would typically set the following option in the configuration file (see the `NOTES` configuration comments file):
>
> ```
> hints  "GENERIC.hints"
> ```

29.7.2.7 Preparing the Root Filesystem

You need to create a root file system for the diskless workstations, in the location listed as `root-path` in `dhcpd.conf`.

29.7.2.7.1 Using `make world` to populate root

This method is quick and will install a complete virgin system (not only the root file system) into `DESTDIR`. All you have to do is simply execute the following script:

```
#!/bin/sh
export DESTDIR=/data/misc/diskless
mkdir -p ${DESTDIR}
cd /usr/src; make buildworld && make buildkernel
cd /usr/src/etc; make distribution
```

Once done, you may need to customize your `/etc/rc.conf` and `/etc/fstab` placed into `DESTDIR` according to your needs.

29.7.2.8 Configuring Swap

If needed, a swap file located on the server can be accessed via NFS.

29.7.2.8.1 NFS Swap

The kernel does not support enabling NFS swap at boot time. Swap must be enabled by the startup scripts, by mounting a writable file system and creating and enabling a swap file. To create a swap file of appropriate size, you can do like this:

```
# dd if=/dev/zero of=/path/to/swapfile bs=1k count=1 oseek=100000
```

To enable it you have to add the following line to your `rc.conf`:

```
swapfile=/path/to/swapfile
```

29.7.2.9 Miscellaneous Issues

29.7.2.9.1 Running with a Read-only `/usr`

If the diskless workstation is configured to run X, you will have to adjust the **XDM** configuration file, which puts the error log on `/usr` by default.

29.7.2.9.2 Using a Non-FreeBSD Server

When the server for the root file system is not running FreeBSD, you will have to create the root file system on a FreeBSD machine, then copy it to its destination, using `tar` or `cpio`.

In this situation, there are sometimes problems with the special files in `/dev`, due to differing major/minor integer sizes. A solution to this problem is to export a directory from the non-FreeBSD server, mount this directory onto a FreeBSD machine, and use devfs(5) to allocate device nodes transparently for the user.

29.8 ISDN

A good resource for information on ISDN technology and hardware is Dan Kegel's ISDN Page (http://www.alumni.caltech.edu/~dank/isdn/).

A quick simple road map to ISDN follows:

- If you live in Europe you might want to investigate the ISDN card section.
- If you are planning to use ISDN primarily to connect to the Internet with an Internet Provider on a dial-up non-dedicated basis, you might look into Terminal Adapters. This will give you the most flexibility, with the fewest problems, if you change providers.
- If you are connecting two LANs together, or connecting to the Internet with a dedicated ISDN connection, you might consider the stand alone router/bridge option.

Cost is a significant factor in determining what solution you will choose. The following options are listed from least expensive to most expensive.

29.8.1 ISDN Cards

Contributed by Hellmuth Michaelis.

FreeBSD's ISDN implementation supports only the DSS1/Q.931 (or Euro-ISDN) standard using passive cards. Some active cards are supported where the firmware also supports other signaling protocols; this also includes the first supported Primary Rate (PRI) ISDN card.

The **isdn4bsd** software allows you to connect to other ISDN routers using either IP over raw HDLC or by using synchronous PPP: either by using kernel PPP with `isppp`, a modified sppp(4) driver, or by using userland ppp(8). By using userland ppp(8), channel bonding of two or more ISDN B-channels is possible. A telephone answering machine application is also available as well as many utilities such as a software 300 Baud modem.

Some growing number of PC ISDN cards are supported under FreeBSD and the reports show that it is successfully used all over Europe and in many other parts of the world.

The passive ISDN cards supported are mostly the ones with the Infineon (formerly Siemens) ISAC/HSCX/IPAC ISDN chipsets, but also ISDN cards with chips from Cologne Chip (ISA bus only), PCI cards with Winbond W6692 chips, some cards with the Tiger300/320/ISAC chipset combinations and some vendor specific chipset based cards such as the AVM Fritz!Card PCI V.1.0 and the AVM Fritz!Card PnP.

Currently the active supported ISDN cards are the AVM B1 (ISA and PCI) BRI cards and the AVM T1 PCI PRI cards.

For documentation on **isdn4bsd**, have a look at `/usr/share/examples/isdn/` directory on your FreeBSD system or at the homepage of isdn4bsd (http://www.freebsd-support.de/i4b/) which also has pointers to hints, erratas and much more documentation such as the isdn4bsd handbook (http://people.FreeBSD.org/~hm/).

In case you are interested in adding support for a different ISDN protocol, a currently unsupported ISDN PC card or otherwise enhancing **isdn4bsd**, please get in touch with Hellmuth Michaelis <hm@FreeBSD.org>.

For questions regarding the installation, configuration and troubleshooting **isdn4bsd**, a freebsd-isdn (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn) mailing list is available.

29.8.2 ISDN Terminal Adapters

Terminal adapters (TA), are to ISDN what modems are to regular phone lines.

Most TA's use the standard Hayes modem AT command set, and can be used as a drop in replacement for a modem.

A TA will operate basically the same as a modem except connection and throughput speeds will be much faster than your old modem. You will need to configure PPP exactly the same as for a modem setup. Make sure you set your serial speed as high as possible.

The main advantage of using a TA to connect to an Internet Provider is that you can do Dynamic PPP. As IP address space becomes more and more scarce, most providers are not willing to provide you with a static IP anymore. Most stand-alone routers are not able to accommodate dynamic IP allocation.

TA's completely rely on the PPP daemon that you are running for their features and stability of connection. This allows you to upgrade easily from using a modem to ISDN on a FreeBSD machine, if you already have PPP set up. However, at the same time any problems you experienced with the PPP program and are going to persist.

If you want maximum stability, use the kernel PPP option, not the userland PPP.

The following TA's are known to work with FreeBSD:

- Motorola BitSurfer and Bitsurfer Pro
- Adtran

Most other TA's will probably work as well, TA vendors try to make sure their product can accept most of the standard modem AT command set.

The real problem with external TA's is that, like modems, you need a good serial card in your computer.

You should read the FreeBSD Serial Hardware (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/serial-uart/index.html) tutorial for a detailed understanding of serial devices, and the differences between asynchronous and synchronous serial ports.

A TA running off a standard PC serial port (asynchronous) limits you to 115.2 Kbs, even though you have a 128 Kbs connection. To fully utilize the 128 Kbs that ISDN is capable of, you must move the TA to a synchronous serial card.

Do not be fooled into buying an internal TA and thinking you have avoided the synchronous/asynchronous issue. Internal TA's simply have a standard PC serial port chip built into them. All this will do is save you having to buy another serial cable and find another empty electrical socket.

A synchronous card with a TA is at least as fast as a stand-alone router, and with a simple 386 FreeBSD box driving it, probably more flexible.

The choice of synchronous card/TA v.s. stand-alone router is largely a religious issue. There has been some discussion of this in the mailing lists. We suggest you search the archives (http://www.FreeBSD.org/search/index.html) for the complete discussion.

29.8.3 Stand-alone ISDN Bridges/Routers

ISDN bridges or routers are not at all specific to FreeBSD or any other operating system. For a more complete description of routing and bridging technology, please refer to a networking reference book.

In the context of this section, the terms router and bridge will be used interchangeably.

As the cost of low end ISDN routers/bridges comes down, it will likely become a more and more popular choice. An ISDN router is a small box that plugs directly into your local Ethernet network, and manages its own connection to the other bridge/router. It has built in software to communicate via PPP and other popular protocols.

A router will allow you much faster throughput than a standard TA, since it will be using a full synchronous ISDN connection.

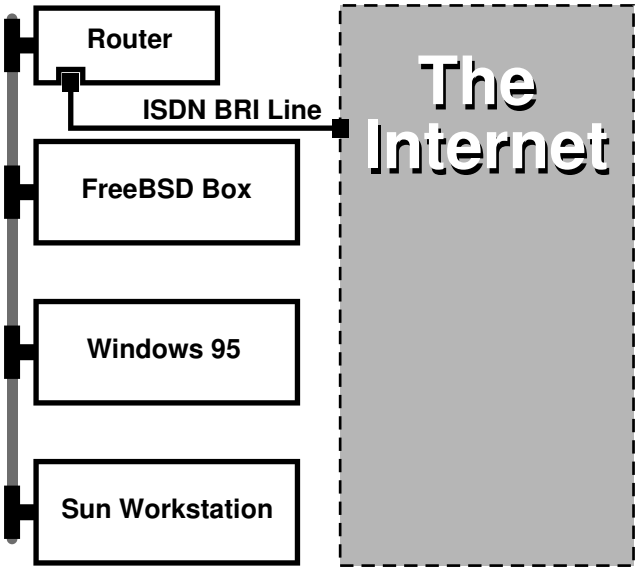
The main problem with ISDN routers and bridges is that interoperability between manufacturers can still be a problem. If you are planning to connect to an Internet provider, you should discuss your needs with them.

If you are planning to connect two LAN segments together, such as your home LAN to the office LAN, this is the simplest lowest maintenance solution. Since you are buying the equipment for both sides of the connection you can be assured that the link will work.

For example to connect a home computer or branch office network to a head office network the following setup could be used:

Example 29-3. Branch Office or Home Network

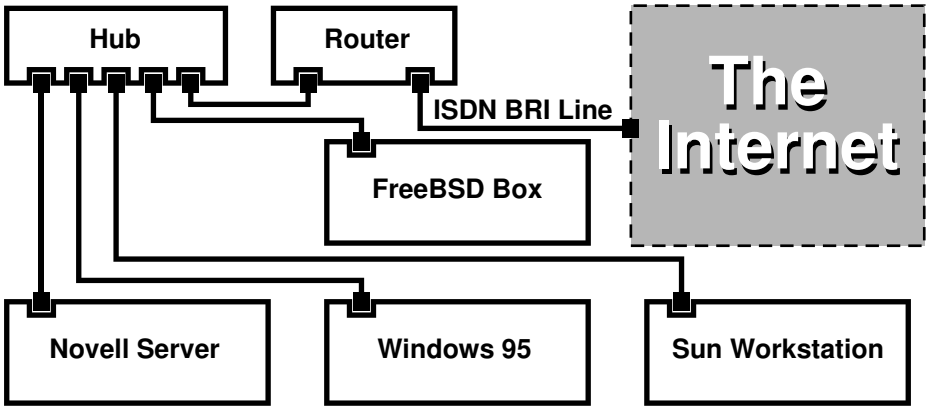
Network uses a bus based topology with 10 base 2 Ethernet (“thinnet”). Connect router to network cable with AUI/10BT transceiver, if necessary.



If your home/branch office is only one computer you can use a twisted pair crossover cable to connect to the stand-alone router directly.

Example 29-4. Head Office or Other LAN

Network uses a star topology with 10 base T Ethernet (“Twisted Pair”).



One large advantage of most routers/bridges is that they allow you to have 2 *separate independent* PPP connections to 2 separate sites at the *same* time. This is not supported on most TA’s, except for specific (usually expensive) models that have two serial ports. Do not confuse this with channel bonding, MPP, etc.

This can be a very useful feature if, for example, you have an dedicated ISDN connection at your office and would like to tap into it, but do not want to get another ISDN line at work. A router at the office location can manage a dedicated B channel connection (64 Kbps) to the Internet and use the other B channel for a separate data connection. The second B channel can be used for dial-in, dial-out or dynamically bonding (MPP, etc.) with the first B channel for more bandwidth.

An Ethernet bridge will also allow you to transmit more than just IP traffic. You can also send IPX/SPX or whatever other protocols you use.

29.9 Network Address Translation

Contributed by Chern Lee.

29.9.1 Overview

FreeBSD’s Network Address Translation daemon, commonly known as natd(8) is a daemon that accepts incoming raw IP packets, changes the source to the local machine and re-injects these packets back into the outgoing IP packet stream. natd(8) does this by changing the source IP address and port such that when data is received back, it is able to determine the original location of the data and forward it back to its original requester.

The most common use of NAT is to perform what is commonly known as Internet Connection Sharing.

29.9.2 Setup

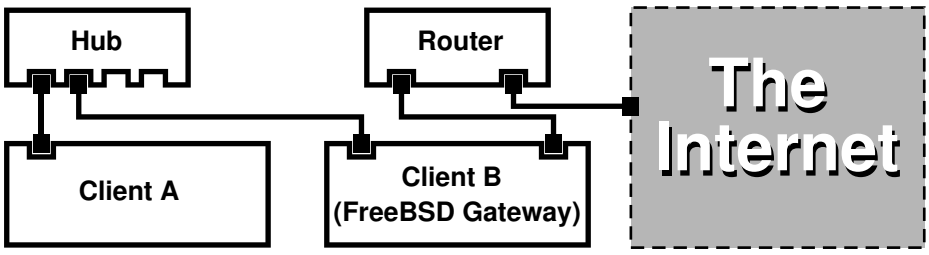
Due to the diminishing IP space in IPv4, and the increased number of users on high-speed consumer lines such as cable or DSL, people are increasingly in need of an Internet Connection Sharing solution. The ability to connect

several computers online through one connection and IP address makes natd(8) a reasonable choice.

Most commonly, a user has a machine connected to a cable or DSL line with one IP address and wishes to use this one connected computer to provide Internet access to several more over a LAN.

To do this, the FreeBSD machine on the Internet must act as a gateway. This gateway machine must have two NICs——one for connecting to the Internet router, the other connecting to a LAN. All the machines on the LAN are connected through a hub or switch.

> **Note:** There are many ways to get a LAN connected to the Internet through a FreeBSD gateway. This example will only cover a gateway with at least two NICs.



A setup like this is commonly used to share an Internet connection. One of the LAN machines is connected to the Internet. The rest of the machines access the Internet through that “gateway” machine.

29.9.3 Configuration

The following options must be in the kernel configuration file:

```
options IPFIREWALL
options IPDIVERT
```

Additionally, at choice, the following may also be suitable:

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
options IPFIREWALL_VERBOSE
```

The following must be in `/etc/rc.conf`:

```
gateway_enable="YES" ❶
firewall_enable="YES" ❷
firewall_type="OPEN" ❸
natd_enable="YES"
natd_interface="fxp0" ❹
natd_flags="" ❺
```

❶ Sets up the machine to act as a gateway. Running `sysctl net.inet.ip.forwarding=1` would have the same effect.

❷ Enables the firewall rules in `/etc/rc.firewall` at boot.

❸ This specifies a predefined firewall ruleset that allows anything in. See `/etc/rc.firewall` for additional types.

❹ Indicates which interface to forward packets through (the interface connected to the Internet).

❺ Any additional configuration options passed to natd(8) on boot.

Having the previous options defined in `/etc/rc.conf` would run `natd -interface fxp0` at boot. This can also be run manually.

> **Note:** It is also possible to use a configuration file for natd(8) when there are too many options to pass. In this case, the configuration file must be defined by adding the following line to `/etc/rc.conf`:
>
> ```
> natd_flags="-f /etc/natd.conf"
> ```
>
> The `/etc/natd.conf` file will contain a list of configuration options, one per line. For example the next section case would use the following file:
>
> ```
> redirect_port tcp 192.168.0.2:6667 6667
> redirect_port tcp 192.168.0.3:80 80
> ```
>
> For more information about the configuration file, consult the natd(8) manual page about the `-f` option.

Each machine and interface behind the LAN should be assigned IP address numbers in the private network space as defined by RFC 1918 (ftp://ftp.isi.edu/in-notes/rfc1918.txt) and have a default gateway of the **natd** machine's internal IP address.

For example, client `A` and `B` behind the LAN have IP addresses of `192.168.0.2` and `192.168.0.3`, while the natd machine's LAN interface has an IP address of `192.168.0.1`. Client `A` and `B`'s default gateway must be set to that of the **natd** machine, `192.168.0.1`. The **natd** machine's external, or Internet interface does not require any special modification for natd(8) to work.

29.9.4 Port Redirection

The drawback with natd(8) is that the LAN clients are not accessible from the Internet. Clients on the LAN can make outgoing connections to the world but cannot receive incoming ones. This presents a problem if trying to run Internet services on one of the LAN client machines. A simple way around this is to redirect selected Internet ports on the **natd** machine to a LAN client.

For example, an IRC server runs on client `A`, and a web server runs on client `B`. For this to work properly, connections received on ports 6667 (IRC) and 80 (web) must be redirected to the respective machines.

The `-redirect_port` must be passed to natd(8) with the proper options. The syntax is as follows:

```
 -redirect_port proto targetIP:targetPORT[-targetPORT]
            [aliasIP:]aliasPORT[-aliasPORT]
            [remoteIP[:remotePORT[-remotePORT]]]
```

In the above example, the argument should be:

```
-redirect_port tcp 192.168.0.2:6667 6667
-redirect_port tcp 192.168.0.3:80 80
```

This will redirect the proper *tcp* ports to the LAN client machines.

The `-redirect_port` argument can be used to indicate port ranges over individual ports. For example, *`tcp 192.168.0.2:2000-3000 2000-3000`* would redirect all connections received on ports 2000 to 3000 to ports 2000 to 3000 on client `A`.

These options can be used when directly running natd(8), placed within the `natd_flags=""` option in `/etc/rc.conf`, or passed via a configuration file.

For further configuration options, consult natd(8)

29.9.5 Address Redirection

Address redirection is useful if several IP addresses are available, yet they must be on one machine. With this, natd(8) can assign each LAN client its own external IP address. natd(8) then rewrites outgoing packets from the LAN clients with the proper external IP address and redirects all traffic incoming on that particular IP address back to the specific LAN client. This is also known as static NAT. For example, the IP addresses `128.1.1.1`, `128.1.1.2`, and `128.1.1.3` belong to the **natd** gateway machine. `128.1.1.1` can be used as the **natd** gateway machine's external IP address, while `128.1.1.2` and `128.1.1.3` are forwarded back to LAN clients `A` and `B`.

The `-redirect_address` syntax is as follows:

```
-redirect_address localIP publicIP
```

localIP	The internal IP address of the LAN client.
publicIP	The external IP address corresponding to the LAN client.

In the example, this argument would read:

```
-redirect_address 192.168.0.2 128.1.1.2
-redirect_address 192.168.0.3 128.1.1.3
```

Like `-redirect_port`, these arguments are also placed within the `natd_flags=""` option of `/etc/rc.conf`, or passed via a configuration file. With address redirection, there is no need for port redirection since all data received on a particular IP address is redirected.

The external IP addresses on the **natd** machine must be active and aliased to the external interface. Look at rc.conf(5) to do so.

29.10 Parallel Line IP (PLIP)

PLIP lets us run TCP/IP between parallel ports. It is useful on machines without network cards, or to install on laptops. In this section, we will discuss:

- Creating a parallel (laplink) cable.
- Connecting two computers with PLIP.

29.10.1 Creating a Parallel Cable

You can purchase a parallel cable at most computer supply stores. If you cannot do that, or you just want to know how it is done, the following table shows how to make one out of a normal parallel printer cable.

Table 29-1. Wiring a Parallel Cable for Networking

A-name	A-End	B-End	Descr.	Post/Bit
DATA0 -ERROR	2 15	15 2	Data	0/0x01 1/0x08
DATA1 +SLCT	3 13	13 3	Data	0/0x02 1/0x10
DATA2 +PE	4 12	12 4	Data	0/0x04 1/0x20
DATA3 -ACK	5 10	10 5	Strobe	0/0x08 1/0x40
DATA4 BUSY	6 11	11 6	Data	0/0x10 1/0x80
GND	18-25	18-25	GND	-

29.10.2 Setting Up PLIP

First, you have to get a laplink cable. Then, confirm that both computers have a kernel with lpt(4) driver support:

```
# grep lp /var/run/dmesg.boot
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
```

The parallel port must be an interrupt driven port, you should have lines similar to the following in your in the `/boot/device.hints` file:

```
hint.ppc.0.at="isa"
hint.ppc.0.irq="7"
```

Then check if the kernel configuration file has a `device plip` line or if the `plip.ko` kernel module is loaded. In both cases the parallel networking interface should appear when you use the ifconfig(8) command to display it:

```
# ifconfig plip0
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

Plug the laplink cable into the parallel interface on both computers.

Configure the network interface parameters on both sites as `root`. For example, if you want to connect the host `host1` with another machine `host2`:

```
                host1 <-----> host2
IP Address    10.0.0.1      10.0.0.2
```

Configure the interface on `host1` by doing:

```
# ifconfig plip0 10.0.0.1 10.0.0.2
```

Configure the interface on `host2` by doing:

```
# ifconfig plip0 10.0.0.2 10.0.0.1
```

You now should have a working connection. Please read the manual pages lp(4) and lpt(4) for more details.

You should also add both hosts to `/etc/hosts`:

```
127.0.0.1               localhost.my.domain localhost
10.0.0.1                host1.my.domain host1
10.0.0.2                host2.my.domain
```

To confirm the connection works, go to each host and ping the other. For example, on `host1`:

```
# ifconfig plip0
plip0: flags=8851<UP,POINTOPINT,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 10.0.0.1 --> 10.0.0.2 netmask 0xff000000
# netstat -r
Routing tables

Internet:
Destination        Gateway          Flags     Refs     Use      Netif Expire
host2              host1            UH          0       0       plip0
# ping -c 4 host2
PING host2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=255 time=2.774 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=255 time=2.530 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=255 time=2.556 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=255 time=2.714 ms

--- host2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.530/2.643/2.774/0.103 ms
```

29.11 IPv6

Originally Written by Aaron Kaplan. Restructured and Added by Tom Rhodes. Extended by Brad Davis.

IPv6 (also known as IPng “IP next generation”) is the new version of the well known IP protocol (also known as IPv4). Like the other current *BSD systems, FreeBSD includes the KAME IPv6 reference implementation. So your FreeBSD system comes with all you will need to experiment with IPv6. This section focuses on getting IPv6 configured and running.

In the early 1990s, people became aware of the rapidly diminishing address space of IPv4. Given the expansion rate of the Internet there were two major concerns:

- Running out of addresses. Today this is not so much of a concern anymore since RFC1918 private address space (`10.0.0.0/8`, `172.16.0.0/12`, and `192.168.0.0/16`) and Network Address Translation (NAT) are being employed.
- Router table entries were getting too large. This is still a concern today.

IPv6 deals with these and many other issues:

- 128 bit address space. In other words theoretically there are 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses available. This means there are approximately 6.67 * 10^27 IPv6 addresses per square meter on our planet.
- Routers will only store network aggregation addresses in their routing tables thus reducing the average space of a routing table to 8192 entries.

There are also lots of other useful features of IPv6 such as:

- Address autoconfiguration (RFC2462 (http://www.ietf.org/rfc/rfc2462.txt))
- Anycast addresses (“one-out-of many”)
- Mandatory multicast addresses
- IPsec (IP security)
- Simplified header structure
- Mobile IP
- IPv6-to-IPv4 transition mechanisms

For more information see:

- IPv6 overview at playground.sun.com (http://playground.sun.com/pub/ipng/html/ipng-main.html)
- KAME.net (http://www.kame.net)

29.11.1 Background on IPv6 Addresses

There are different types of IPv6 addresses: Unicast, Anycast and Multicast.

Unicast addresses are the well known addresses. A packet sent to a unicast address arrives exactly at the interface belonging to the address.

Anycast addresses are syntactically indistinguishable from unicast addresses but they address a group of interfaces. The packet destined for an anycast address will arrive at the nearest (in router metric) interface. Anycast addresses may only be used by routers.

Multicast addresses identify a group of interfaces. A packet destined for a multicast address will arrive at all interfaces belonging to the multicast group.

Note: The IPv4 broadcast address (usually `xxx.xxx.xxx.255`) is expressed by multicast addresses in IPv6.

Table 29-2. Reserved IPv6 addresses

IPv6 address	Prefixlength (Bits)	Description	Notes
`::`	128 bits	unspecified	cf. `0.0.0.0` in IPv4
`::1`	128 bits	loopback address	cf. `127.0.0.1` in IPv4

IPv6 address	Prefixlength (Bits)	Description	Notes
`::00:xx:xx:xx:xx`	96 bits	embedded IPv4	The lower 32 bits are the IPv4 address. Also called "IPv4 compatible IPv6 address"
`::ff:xx:xx:xx:xx`	96 bits	IPv4 mapped IPv6 address	The lower 32 bits are the IPv4 address. For hosts which do not support IPv6.
`fe80::-feb::`	10 bits	link-local	cf. loopback address in IPv4
`fec0::-fef::`	10 bits	site-local	
`ff::`	8 bits	multicast	
`001` (base 2)	3 bits	global unicast	All global unicast addresses are assigned from this pool. The first 3 bits are "001".

29.11.2 Reading IPv6 Addresses

The canonical form is represented as: `x:x:x:x:x:x:x:x`, each "x" being a 16 Bit hex value. For example `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`

Often an address will have long substrings of all zeros therefore one such substring per address can be abbreviated by "::". Also up to three leading "0"s per hexquad can be omitted. For example `fe80::1` corresponds to the canonical form `fe80:0000:0000:0000:0000:0000:0000:0001`.

A third form is to write the last 32 Bit part in the well known (decimal) IPv4 style with dots "." as separators. For example `2002::10.0.0.1` corresponds to the (hexadecimal) canonical representation `2002:0000:0000:0000:0000:0000:0a00:0001` which in turn is equivalent to writing `2002::a00:1`.

By now the reader should be able to understand the following:

```
# ifconfig

rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
         inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
         inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
         ether 00:00:21:03:08:e1
         media: Ethernet autoselect (100baseTX )
         status: active
```

`fe80::200:21ff:fe03:8e1%rl0` is an auto configured link-local address. It is generated from the MAC address as part of the auto configuration.

For further information on the structure of IPv6 addresses see RFC3513 (http://www.ietf.org/rfc/rfc3513.txt).

29.11.3 Getting Connected

Currently there are four ways to connect to other IPv6 hosts and networks:

- Getting an IPv6 network from your upstream provider. Talk to your Internet provider for instructions.
- Tunnel via 6-to-4 (RFC3068 (http://www.ietf.org/rfc/rfc3068.txt))
- Use the `net/freenet6` port if you are on a dial-up connection.

29.11.4 DNS in the IPv6 World

There used to be two types of DNS records for IPv6. The IETF has declared A6 records obsolete. AAAA records are the standard now.

Using AAAA records is straightforward. Assign your hostname to the new IPv6 address you just received by adding:

```
MYHOSTNAME            AAAA     MYIPv6ADDR
```

To your primary zone DNS file. In case you do not serve your own DNS zones ask your DNS provider. Current versions of **bind** (version 8.3 and 9) and `dns/djbdns` (with the IPv6 patch) support AAAA records.

29.11.5 Applying the needed changes to `/etc/rc.conf`

29.11.5.1 IPv6 Client Settings

These settings will help you configure a machine that will be on your LAN and act as a client, not a router. To have rtsol(8) autoconfigure your interface on boot all you need to add is:

```
ipv6_enable="YES"
```

To statically assign an IP address such as `2001:471:1f11:251:290:27ff:fee0:2093`, to your `fxp0` interface, add:

```
ipv6_ifconfig_fxp0="2001:471:1f11:251:290:27ff:fee0:2093"
```

To assign a default router of `2001:471:1f11:251::1` add the following to `/etc/rc.conf`:

```
ipv6_defaultrouter="2001:471:1f11:251::1"
```

29.11.5.2 IPv6 Router/Gateway Settings

This will help you take the directions that your tunnel provider has given you and convert it into settings that will persist through reboots. To restore your tunnel on startup use something like the following in `/etc/rc.conf`:

List the Generic Tunneling interfaces that will be configured, for example `gif0`:

```
gif_interfaces="gif0"
```

To configure the interface with a local endpoint of *`MY_IPv4_ADDR`* to a remote endpoint of *`REMOTE_IPv4_ADDR`*:

```
gifconfig_gif0="MY_IPv4_ADDR REMOTE_IPv4_ADDR"
```

To apply the IPv6 address you have been assigned for use as your IPv6 tunnel endpoint, add:

```
ipv6_ifconfig_gif0="MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR"
```

Then all you have to do is set the default route for IPv6. This is the other side of the IPv6 tunnel:

```
ipv6_defaultrouter="MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR"
```

29.11.5.3 IPv6 Tunnel Settings

If the server is to route IPv6 between the rest of your network and the world, the following `/etc/rc.conf` setting will also be needed:

```
ipv6_gateway_enable="YES"
```

29.11.6 Router Advertisement and Host Auto Configuration

This section will help you setup rtadvd(8) to advertise the IPv6 default route.

To enable rtadvd(8) you will need the following in your `/etc/rc.conf`:

```
rtadvd_enable="YES"
```

It is important that you specify the interface on which to do IPv6 router solicitation. For example to tell rtadvd(8) to use `fxp0`:

```
rtadvd_interfaces="fxp0"
```

Now we must create the configuration file, `/etc/rtadvd.conf`. Here is an example:

```
fxp0:\
 :addrs#1:addr="2001:471:1f11:246::":prefixlen#64:tc=ether:
```

Replace `fxp0` with the interface you are going to be using.

Next, replace `2001:471:1f11:246::` with the prefix of your allocation.

If you are dedicated a `/64` subnet you will not need to change anything else. Otherwise, you will need to change the `prefixlen#` to the correct value.

29.12 Asynchronous Transfer Mode (ATM)

Contributed by Harti Brandt.

29.12.1 Configuring classical IP over ATM (PVCs)

Classical IP over ATM (CLIP) is the simplest method to use Asynchronous Transfer Mode (ATM) with IP. It can be used with switched connections (SVCs) and with permanent connections (PVCs). This section describes how to set up a network based on PVCs.

29.12.1.1 Fully meshed configurations

The first method to set up a CLIP with PVCs is to connect each machine to each other machine in the network via a dedicated PVC. While this is simple to configure it tends to become impractical for a larger number of machines. The example supposes that we have four machines in the network, each connected to the ATM network with an ATM adapter card. The first step is the planning of the IP addresses and the ATM connections between the machines. We use the following:

Host	IP Address
hostA	192.168.173.1
hostB	192.168.173.2
hostC	192.168.173.3
hostD	192.168.173.4

To build a fully meshed net we need one ATM connection between each pair of machines:

Machines	VPI.VCI couple
hostA - hostB	0.100
hostA - hostC	0.101
hostA - hostD	0.102
hostB - hostC	0.103
hostB - hostD	0.104
hostC - hostD	0.105

The VPI and VCI values at each end of the connection may of course differ, but for simplicity we assume that they are the same. Next we need to configure the ATM interfaces on each host:

```
hostA# ifconfig hatm0 192.168.173.1 up
hostB# ifconfig hatm0 192.168.173.2 up
hostC# ifconfig hatm0 192.168.173.3 up
hostD# ifconfig hatm0 192.168.173.4 up
```

assuming that the ATM interface is `hatm0` on all hosts. Now the PVCs need to be configured on `hostA` (we assume that they are already configured on the ATM switches, you need to consult the manual for the switch on how to do this).

```
hostA# atmconfig natm add 192.168.173.2 hatm0 0 100 llc/snap ubr
hostA# atmconfig natm add 192.168.173.3 hatm0 0 101 llc/snap ubr
hostA# atmconfig natm add 192.168.173.4 hatm0 0 102 llc/snap ubr

hostB# atmconfig natm add 192.168.173.1 hatm0 0 100 llc/snap ubr
hostB# atmconfig natm add 192.168.173.3 hatm0 0 103 llc/snap ubr
hostB# atmconfig natm add 192.168.173.4 hatm0 0 104 llc/snap ubr

hostC# atmconfig natm add 192.168.173.1 hatm0 0 101 llc/snap ubr
hostC# atmconfig natm add 192.168.173.2 hatm0 0 103 llc/snap ubr
hostC# atmconfig natm add 192.168.173.4 hatm0 0 105 llc/snap ubr

hostD# atmconfig natm add 192.168.173.1 hatm0 0 102 llc/snap ubr
```

```
hostD# atmconfig natm add 192.168.173.2 hatm0 0 104 llc/snap ubr
hostD# atmconfig natm add 192.168.173.3 hatm0 0 105 llc/snap ubr
```

Of course other traffic contracts than UBR can be used given the ATM adapter supports those. In this case the name of the traffic contract is followed by the parameters of the traffic. Help for the atmconfig(8) tool can be obtained with:

```
# atmconfig help natm add
```

or in the atmconfig(8) manual page.

The same configuration can also be done via `/etc/rc.conf`. For `hostA` this would look like:

```
network_interfaces="lo0 hatm0"
ifconfig_hatm0="inet 192.168.173.1 up"
natm_static_routes="hostB hostC hostD"
route_hostB="192.168.173.2 hatm0 0 100 llc/snap ubr"
route_hostC="192.168.173.3 hatm0 0 101 llc/snap ubr"
route_hostD="192.168.173.4 hatm0 0 102 llc/snap ubr"
```

The current state of all CLIP routes can be obtained with:

```
hostA# atmconfig natm show
```

29.13 Common Access Redundancy Protocol (CARP)

Contributed by Tom Rhodes.

The Common Access Redundancy Protocol, or CARP allows multiple hosts to share the same IP address. In some configurations, this may be used for availability or load balancing. Hosts may use separate IP addresses as well, as in the example provided here.

To enable support for CARP, the FreeBSD kernel must be rebuilt with the following option:

```
device carp
```

CARP functionality should now be available and may be tuned via several `sysctl` OIDs. Devices themselves may be loaded via the `ifconfig` command:

```
# ifconfig carp0 create
```

In a real environment, these interfaces will need unique identification numbers known as a VHID. This VHID or Virtual Host Identification will be used to distinguish the host on the network.

29.13.1 Using CARP For Server Availability (CARP)

One use of CARP, as noted above, is for server availability. This example will provide failover support for three hosts, all with unique IP addresses and providing the same web content. These machines will act in conjunction with a Round Robin DNS configuration. The failover machine will have two additional CARP interfaces, one for each of the content server’s IPs. When a failure occurs, the failover server should pick up the failed machine’s IP address.

This means the failure should go completely unnoticed to the user. The failover server requires identical content and services as the other content servers it is expected to pick up load for.

The two machines should be configured identically other than their issued hostnames and VHIDs. This example calls these machines `hosta.example.org` and `hostb.example.org` respectively. First, the required lines for a CARP configuration have to be added to `rc.conf`. For `hosta.example.org`, the `rc.conf` file should contain the following lines:

```
hostname="hosta.example.org"
ifconfig_fxp0="inet 192.168.1.3 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 pass testpast 192.168.1.50/24"
```

On `hostb.example.org` the following lines should be in `rc.conf`:

```
hostname="hostb.example.org"
ifconfig_fxp0="inet 192.168.1.4 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 2 pass testpass 192.168.1.51/24"
```

> **Note:** It is very important that the passwords, specified by the `pass` option to `ifconfig`, are identical. The `carp` devices will only listen to and accept advertisements from machines with the correct password. The VHID must also be different for each machine.

The third machine, `provider.example.org`, should be prepared so that it may handle failover from either host. This machine will require two `carp` devices, one to handle each host. The appropriate `rc.conf` configuration lines will be similar to the following:

```
hostname="provider.example.org"
ifconfig_fxp0="inet 192.168.1.5 netmask 255.255.255.0"
cloned_interfaces="carp0 carp1"
ifconfig_carp0="vhid 1 advskew 100 pass testpass 192.168.1.50/24"
ifconfig_carp1="vhid 2 advskew 100 pass testpass 192.168.1.51/24"
```

Having the two `carp` devices will allow `provider.example.org` to notice and pick up the IP address of either machine should it stop responding.

> **Note:** The default FreeBSD kernel *may* have preemption enabled. If so, `provider.example.org` may not relinquish the IP address back to the original content server. In this case, an administrator may “nudge” the interface. The following command should be issued on `provider.example.org`:
>
> ```
> # ifconfig carp0 down && ifconfig carp0 up
> ```
>
> This should be done on the `carp` interface which corresponds to the correct host.

At this point, CARP should be completely enabled and available for testing. For testing, either networking has to be restarted or the machines need to be rebooted.

More information is always available in the carp(4) manual page.

V. 附錄

Appendix A. 取得FreeBSD 的方式

A.1 CDROM 及DVD 發行商

A.1.1 盒裝產品的零售處：

FreeBSD 盒裝產品(含FreeBSD 光碟及其他一些軟體、書面文件)的零售業者：

· CompUSA
  WWW: http://www.compusa.com/

· Frys Electronics
  WWW: http://www.frys.com/

A.1.2 CD 及DVD 合集

FreeBSD 光碟(CD 及DVD)的網路零售業者：

· BSD Mall by Daemon News
  PO Box 161
  Nauvoo, IL 62354
  USA
  Phone: +1 866 273-6255
  Fax: +1 217 453-9956
  Email: <sales@bsdmall.com>
  WWW: http://www.bsdmall.com/freebsd1.html

· BSD-Systems
  Email: <info@bsd-systems.co.uk>
  WWW: http://www.bsd-systems.co.uk

· FreeBSD Mall, Inc.
  3623 Sanford Street
  Concord, CA 94520-1405
  USA
  Phone: +1 925 674-0783
  Fax: +1 925 674-0821
  Email: <info@freebsdmall.com>
  WWW: http://www.freebsdmall.com/

· Hinner EDV
  St. Augustinus-Str. 10
  D-81825 München
  Germany
  Phone: (089) 428 419
  WWW: http://www.hinner.de/linux/freebsd.html

· Ikarios

22-24 rue Voltaire
92000 Nanterre
France
WWW: http://ikarios.com/form/#freebsd

· JMC Software
Ireland
Phone: 353 1 6291282
WWW: http://www.thelinuxmall.com

· Linux CD Mall
Private Bag MBE N348
Auckland 1030
New Zealand
Phone: +64 21 866529
WWW: http://www.linuxcdmall.co.nz/

· The Linux Emporium
Hilliard House, Lester Way
Wallingford
OX10 9TA
United Kingdom
Phone: +44 1491 837010
Fax: +44 1491 837016
WWW: http://www.linuxemporium.co.uk/products/freebsd/

· Linux+ DVD Magazine
Lewartowskiego 6
Warsaw
00-190
Poland
Phone: +48 22 860 18 18
Email: <editors@lpmagazine.org>
WWW: http://www.lpmagazine.org/

· Linux System Labs Australia
21 Ray Drive
Balwyn North
VIC - 3104
Australia
Phone: +61 3 9857 5918
Fax: +61 3 9857 8974
WWW: http://www.lsl.com.au

· LinuxCenter.Ru
Galernaya Street, 55
Saint-Petersburg
190000
Russia
Phone: +7-812-3125208
Email: <info@linuxcenter.ru>

WWW: http://linuxcenter.ru/freebsd

A.1.3 經銷商(Distributors)

若你是區域經銷商，並想代理經銷FreeBSD 光碟產品的話，請與下列的代理商聯繫：

· Cylogistics
809B Cuesta Dr., #2149
Mountain View, CA 94040
USA
Phone: +1 650 694-4949
Fax: +1 650 694-4953
Email: <sales@cylogistics.com>
WWW: http://www.cylogistics.com/

· Ingram Micro
1600 E. St. Andrew Place
Santa Ana, CA 92705-4926
USA
Phone: 1 (800) 456-8000
WWW: http://www.ingrammicro.com/

· Kudzu, LLC
7375 Washington Ave. S.
Edina, MN 55439
USA
Phone: +1 952 947-0822
Fax: +1 952 947-0876
Email: <sales@kudzuenterprises.com>

· LinuxCenter.Ru
Galernaya Street, 55
Saint-Petersburg
190000
Russia
Phone: +7-812-3125208
Email: <info@linuxcenter.ru>
WWW: http://linuxcenter.ru/freebsd

· Navarre Corp
7400 49th Ave South
New Hope, MN 55428
USA
Phone: +1 763 535-8333
Fax: +1 763 535-0341
WWW: http://www.navarre.com/

A.2 FTP 站

The official sources for FreeBSD are available via anonymous FTP from a worldwide set of mirror sites. The site ftp://ftp.FreeBSD.org/pub/FreeBSD/ is well connected and allows a large number of connections to it, but you are probably better off finding a “closer” mirror site (especially if you decide to set up some sort of mirror site).

The FreeBSD mirror sites database (http://mirrorlist.FreeBSD.org/) is more accurate than the mirror listing in the Handbook, as it gets its information from the DNS rather than relying on static lists of hosts.

Additionally, FreeBSD is available via anonymous FTP from the following mirror sites. If you choose to obtain FreeBSD via anonymous FTP, please try to use a site near you. The mirror sites listed as “Primary Mirror Sites” typically have the entire FreeBSD archive (all the currently available versions for each of the architectures) but you will probably have faster download times from a site that is in your country or region. The regional sites carry the most recent versions for the most popular architecture(s) but might not carry the entire FreeBSD archive. All sites provide access via anonymous FTP but some sites also provide access via other methods. The access methods available for each site are provided in parentheses after the hostname.

Central Servers, Primary Mirror Sites, Argentina, Armenia, Australia, Austria, Brazil, Bulgaria, Canada, China, Croatia, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hong Kong, Hungary, Iceland, Indonesia, Ireland, Israel, Italy, Japan, Korea, Latvia, Lithuania, Netherlands, New Zealand, Norway, Poland, Portugal, Romania, Russia, Saudi Arabia, Singapore, Slovak Republic, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Turkey, Ukraine, United Kingdom, USA.

(as of 2009/05/21 21:02:58 UTC)

Central Servers

- ftp://ftp.FreeBSD.org/pub/FreeBSD/ (ftp)

Primary Mirror Sites

In case of problems, please contact the hostmaster <mirror-admin@FreeBSD.org> for this domain.

- ftp://ftp1.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp12.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp13.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.FreeBSD.org/pub/FreeBSD/ (ftp)

Argentina

In case of problems, please contact the hostmaster <hostmaster@ar.FreeBSD.org> for this domain.

- ftp://ftp.ar.FreeBSD.org/pub/FreeBSD/ (ftp)

Armenia

In case of problems, please contact the hostmaster <hostmaster@am.FreeBSD.org> for this domain.

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD/ (ftp)

Australia

In case of problems, please contact the hostmaster <hostmaster@au.FreeBSD.org> for this domain.

- ftp://ftp.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/ (ftp)

Austria

In case of problems, please contact the hostmaster <hostmaster@at.FreeBSD.org> for this domain.

- ftp://ftp.at.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.at.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp2.at.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.at.freebsd.org/pub/FreeBSD/) / rsync)

Brazil

In case of problems, please contact the hostmaster <hostmaster@br.FreeBSD.org> for this domain.

- ftp://ftp.br.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.br.FreeBSD.org/))
- ftp://ftp2.br.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.br.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.br.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.br.FreeBSD.org/pub/FreeBSD/ (ftp)

Bulgaria

In case of problems, please contact the hostmaster <hostmaster@bg.FreeBSD.org> for this domain.

- ftp://ftp.bg.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.bg.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Canada

In case of problems, please contact the hostmaster <hostmaster@ca.FreeBSD.org> for this domain.

- ftp://ftp.ca.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.ca.FreeBSD.org/ (ftp)
- ftp://ftp3.ca.FreeBSD.org/pub/FreeBSD/ (ftp)

China

In case of problems, please contact the hostmaster <hostmaster@cn.FreeBSD.org> for this domain.

- ftp://ftp.cn.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.cn.FreeBSD.org/pub/FreeBSD/ (ftp)

Croatia

In case of problems, please contact the hostmaster <hostmaster@hr.FreeBSD.org> for this domain.

- ftp://ftp.hr.FreeBSD.org/pub/FreeBSD/ (ftp)

Czech Republic

In case of problems, please contact the hostmaster <hostmaster@cz.FreeBSD.org> for this domain.

- ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.cz.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp2.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.cz.FreeBSD.org/pub/FreeBSD/))

Denmark

In case of problems, please contact the hostmaster <hostmaster@dk.FreeBSD.org> for this domain.

- ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6)
- ftp://ftp2.dk.FreeBSD.org/pub/FreeBSD/ (ftp)

Estonia

In case of problems, please contact the hostmaster <hostmaster@ee.FreeBSD.org> for this domain.

- ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/ (ftp)

Finland

In case of problems, please contact the hostmaster <hostmaster@fi.FreeBSD.org> for this domain.

- ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.fi.FreeBSD.org/pub/FreeBSD/ (ftp)

France

In case of problems, please contact the hostmaster <hostmaster@fr.FreeBSD.org> for this domain.

- ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/ (ftp)

Germany

In case of problems, please contact the hostmaster <de-bsd-hubs@de.FreeBSD.org > for this domain.

- ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.de.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp3.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.de.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp4.de.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp7.de.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp8.de.FreeBSD.org/pub/FreeBSD/ (ftp)

Greece

In case of problems, please contact the hostmaster <hostmaster@gr.FreeBSD.org> for this domain.

- ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/ (ftp)

Hong Kong

- ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/ (ftp)

Hungary

In case of problems, please contact the hostmaster <hostmaster@hu.FreeBSD.org> for this domain.

- ftp://ftp.hu.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.hu.FreeBSD.org/) / rsync)
- ftp://ftp2.hu.FreeBSD.org/pub/FreeBSD/ (ftp)

Iceland

In case of problems, please contact the hostmaster <hostmaster@is.FreeBSD.org> for this domain.

- ftp://ftp.is.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Indonesia

In case of problems, please contact the hostmaster <hostmaster@id.FreeBSD.org> for this domain.

- ftp://ftp.id.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.id.FreeBSD.org/) / rsync)

Ireland

In case of problems, please contact the hostmaster <hostmaster@ie.FreeBSD.org> for this domain.

- ftp://ftp.ie.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.ie.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.ie.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp3.ie.FreeBSD.org/pub/FreeBSD/) / rsync)

Israel

In case of problems, please contact the hostmaster <hostmaster@il.FreeBSD.org> for this domain.

- ftp://ftp.il.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6)

Italy

In case of problems, please contact the hostmaster <hostmaster@it.FreeBSD.org> for this domain.

- ftp://ftp.it.FreeBSD.org/pub/FreeBSD/ (ftp)

Japan

In case of problems, please contact the hostmaster <hostmaster@jp.FreeBSD.org> for this domain.

- ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

Korea

In case of problems, please contact the hostmaster <hostmaster@kr.FreeBSD.org> for this domain.

- ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/ (ftp)

Latvia

In case of problems, please contact the hostmaster <hostmaster@lv.FreeBSD.org> for this domain.

- ftp://ftp.lv.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.lv.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp2.lv.FreeBSD.org/pub/FreeBSD/ (ftp)

Lithuania

In case of problems, please contact the hostmaster <hostmaster@lt.FreeBSD.org> for this domain.

- ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/ (ftp)

Netherlands

In case of problems, please contact the hostmaster <hostmaster@nl.FreeBSD.org> for this domain.

- ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.nl.FreeBSD.org/os/FreeBSD/) / rsync)
- ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/ (ftp)

New Zealand

- ftp://ftp.nz.FreeBSD.org/pub/FreeBSD/ (ftp)

Norway

In case of problems, please contact the hostmaster <hostmaster@no.FreeBSD.org> for this domain.

- ftp://ftp.no.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp3.no.FreeBSD.org/pub/FreeBSD/ (ftp)

Poland

In case of problems, please contact the hostmaster <hostmaster@pl.FreeBSD.org> for this domain.

- ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.pl.FreeBSD.org/pub/FreeBSD/ (ftp)

Portugal

In case of problems, please contact the hostmaster <hostmaster@pt.FreeBSD.org> for this domain.

- ftp://ftp.pt.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.pt.FreeBSD.org/pub/freebsd/ (ftp)
- ftp://ftp4.pt.FreeBSD.org/pub/ISO/FreeBSD/ (ftp)

Romania

In case of problems, please contact the hostmaster <hostmaster@ro.FreeBSD.org> for this domain.

- ftp://ftp.ro.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.ro.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6)

Russia

In case of problems, please contact the hostmaster <hostmaster@ru.FreeBSD.org> for this domain.

- ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.ru.FreeBSD.org/FreeBSD/) / rsync)
- ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.ru.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp3.ru.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp5.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp5.ru.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp6.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

Saudi Arabia

In case of problems, please contact the hostmaster <ftpadmin@isu.net.sa> for this domain.

- ftp://ftp.isu.net.sa/pub/mirrors/ftp.freebsd.org/ (ftp)

Singapore

In case of problems, please contact the hostmaster <hostmaster@sg.FreeBSD.org> for this domain.

- ftp://ftp.sg.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.sg.FreeBSD.org/pub/FreeBSD/) / rsync)

Slovak Republic

In case of problems, please contact the hostmaster <hostmaster@sk.FreeBSD.org> for this domain.

- ftp://ftp.sk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp.sk.FreeBSD.org/pub/FreeBSD/) / http (http://ftp.sk.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp.sk.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)

Slovenia

In case of problems, please contact the hostmaster <hostmaster@si.FreeBSD.org> for this domain.

- ftp://ftp.si.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.si.FreeBSD.org/pub/FreeBSD/ (ftp)

South Africa

In case of problems, please contact the hostmaster <hostmaster@za.FreeBSD.org> for this domain.

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/ (ftp)

Spain

In case of problems, please contact the hostmaster <hostmaster@es.FreeBSD.org> for this domain.

- ftp://ftp.es.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.es.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/ (ftp)

Sweden

In case of problems, please contact the hostmaster <hostmaster@se.FreeBSD.org> for this domain.

- ftp://ftp.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.se.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp5.se.FreeBSD.org/) / rsync)

Switzerland

In case of problems, please contact the hostmaster <hostmaster@ch.FreeBSD.org> for this domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.ch.FreeBSD.org/pub/FreeBSD/ (ftp)

Taiwan

In case of problems, please contact the hostmaster <hostmaster@tw.FreeBSD.org> for this domain.

- ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)
- ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/) / http (http://ftp2.tw.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp2.tw.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)
- ftp://ftp3.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp6.tw.FreeBSD.org/) / rsync)
- ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp11.tw.FreeBSD.org/FreeBSD/))
- ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/ (ftp)

Turkey

- ftp://ftp.tr.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.tr.freebsd.org/pub/FreeBSD/) / rsync)
- ftp://ftp2.tr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Ukraine

- ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.ua.freebsd.org/pub/FreeBSD/))
- ftp://ftp2.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.ua.freebsd.org/pub/FreeBSD/))
- ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp8.ua.freebsd.org/FreeBSD/))
- ftp://ftp11.ua.FreeBSD.org/pub/FreeBSD/ (ftp)

United Kingdom

In case of problems, please contact the hostmaster <`hostmaster@uk.FreeBSD.org`> for this domain.

- ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.uk.FreeBSD.org/) / rsync)
- ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.uk.FreeBSD.org/pub/FreeBSD/ (ftp)

USA

In case of problems, please contact the hostmaster <`hostmaster@us.FreeBSD.org`> for this domain.

- ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6)
- ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp6.us.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp7.us.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp7.us.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.us.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp9.us.FreeBSD.org/pub/os/FreeBSD/))
- ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp12.us.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp13.us.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/ (ftp)

A.3 Anonymous CVS

A.3.1 anoncvs 簡介

Anonymous CVS (or, as it is otherwise known, *anoncvs*) is a feature provided by the CVS utilities bundled with FreeBSD for synchronizing with a remote CVS repository. Among other things, it allows users of FreeBSD to perform, with no special privileges, read-only CVS operations against one of the FreeBSD project's official anoncvs servers. To use it, one simply sets the `CVSROOT` environment variable to point at the appropriate anoncvs server, provides the well-known password "anoncvs" with the `cvs login` command, and then uses the cvs(1) command to access it like any local repository.

> **Note:** The `cvs login` command, stores the passwords that are used for authenticating to the CVS server in a file called `.cvspass` in your `HOME` directory. If this file does not exist, you might get an error when trying to use `cvs login` for the first time. Just make an empty `.cvspass` file, and retry to login.

While it can also be said that the CVSup and *anoncvs* services both perform essentially the same function, there are various trade-offs which can influence the user's choice of synchronization methods. In a nutshell, **CVSup** is much more efficient in its usage of network resources and is by far the most technically sophisticated of the two, but at a price. To use **CVSup**, a special client must first be installed and configured before any bits can be grabbed, and then only in the fairly large chunks which **CVSup** calls *collections*.

Anoncvs, by contrast, can be used to examine anything from an individual file to a specific program (like `ls` or `grep`) by referencing the CVS module name. Of course, **anoncvs** is also only good for read-only operations on the CVS repository, so if it is your intention to support local development in one repository shared with the FreeBSD project bits then **CVSup** is really your only option.

A.3.2 Using Anonymous CVS

Configuring cvs(1) to use an Anonymous CVS repository is a simple matter of setting the `CVSROOT` environment variable to point to one of the FreeBSD project's *anoncvs* servers. At the time of this writing, the following servers are available:

- *Austria(奧地利)*: :pserver:anoncvs@anoncvs.at.FreeBSD.org:/home/ncvs (Use `cvs login` and enter any password when prompted.)
- *France(法國)*: :pserver:anoncvs@anoncvs.fr.FreeBSD.org:/home/ncvs (pserver (password "anoncvs"), ssh (no password))

- *Germany(德國)*: :pserver:anoncvs@anoncvs.de.FreeBSD.org:/home/ncvs (Use `cvs login` and enter the password “anoncvs” when prompted.)
- *Germany(德國)*: :pserver:anoncvs@anoncvs2.de.FreeBSD.org:/home/ncvs (rsh, pserver, ssh, ssh/2022)
- *Japan(日本)*: :pserver:anoncvs@anoncvs.jp.FreeBSD.org:/home/ncvs (Use `cvs login` and enter the password “anoncvs” when prompted.)
- *USA(美國)*: freebsdanoncvs@anoncvs.FreeBSD.org:/home/ncvs (ssh only - no password)

  ```
  SSH HostKey: 1024 a1:e7:46:de:fb:56:ef:05:bc:73:aa:91:09:da:f7:f4 root@sanmateo.ecn.purdue.edu
  SSH2 HostKey: 1024 52:02:38:1a:2f:a8:71:d3:f5:83:93:8d:aa:00:6f:65 ssh_host_dsa_key.pub
  ```

- *USA(美國)*: anoncvs@anoncvs1.FreeBSD.org:/home/ncvs (ssh only - no password)

  ```
  SSH HostKey: 1024 8b:c4:6f:9a:7e:65:8a:eb:50:50:29:7c:a1:47:03:bc root@ender.liquidneon.com
  SSH2 HostKey: 2048 4d:59:19:7b:ea:9b:76:0b:ca:ee:da:26:e2:3a:83:b8 ssh_host_dsa_key.pub
  ```

Since CVS allows one to “check out” virtually any version of the FreeBSD sources that ever existed (or, in some cases, will exist), you need to be familiar with the revision (`-r`) flag to cvs(1) and what some of the permissible values for it in the FreeBSD Project repository are.

There are two kinds of tags, revision tags and branch tags. A revision tag refers to a specific revision. Its meaning stays the same from day to day. A branch tag, on the other hand, refers to the latest revision on a given line of development, at any given time. Because a branch tag does not refer to a specific revision, it may mean something different tomorrow than it means today.

Section A.7 contains revision tags that users might be interested in. Again, none of these are valid for the Ports Collection since the Ports Collection does not have multiple revisions.

When you specify a branch tag, you normally receive the latest versions of the files on that line of development. If you wish to receive some past version, you can do so by specifying a date with the `-D date` flag. See the cvs(1) manual page for more details.

A.3.3 Examples

While it really is recommended that you read the manual page for cvs(1) thoroughly before doing anything, here are some quick examples which essentially show how to use Anonymous CVS:

Example A-1. Checking Out Something from -CURRENT (ls(1)):

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.jp.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password “anoncvs”.
% cvs co ls
```

Example A-2. Using SSH to check out the `src/` tree:

```
% cvs -d freebsdanoncvs@anoncvs.FreeBSD.org:/home/ncvs co src
The authenticity of host 'anoncvs.freebsd.org (128.46.156.46)' can't be established.
DSA key fingerprint is 52:02:38:1a:2f:a8:71:d3:f5:83:93:8d:aa:00:6f:65.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'anoncvs.freebsd.org' (DSA) to the list of known hosts.
```

Example A-3. Checking Out the Version of ls(1) in the 6-STABLE Branch:

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.jp.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password “anoncvs”.
% cvs co -rRELENG_6 ls
```

Example A-4. Creating a List of Changes (as Unified Diffs) to ls(1)

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.jp.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password “anoncvs”.
% cvs rdiff -u -rRELENG_5_3_0_RELEASE -rRELENG_5_4_0_RELEASE ls
```

Example A-5. Finding Out What Other Module Names Can Be Used:

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.jp.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password “anoncvs”.
% cvs co modules
% more modules/modules
```

A.3.4 Other Resources

The following additional resources may be helpful in learning CVS:

- CVS Tutorial (http://www.csc.calpoly.edu/~dbutler/tutorials/winter96/cvs/) from Cal Poly.
- CVS Home (http://ximbiot.com/cvs/wiki/), the CVS development and support community.
- CVSweb (http://www.FreeBSD.org/cgi/cvsweb.cgi) is the FreeBSD Project web interface for CVS.

A.4 Using CTM

CTM is a method for keeping a remote directory tree in sync with a central one. It has been developed for usage with FreeBSD’s source trees, though other people may find it useful for other purposes as time goes by. Little, if any, documentation currently exists at this time on the process of creating deltas, so contact the ctm-users (http://lists.FreeBSD.org/mailman/listinfo/ctm-users) mailing list for more information and if you wish to use **CTM** for other things.

A.4.1 Why Should I Use CTM?

CTM will give you a local copy of the FreeBSD source trees. There are a number of “flavors” of the tree available. Whether you wish to track the entire CVS tree or just one of the branches, **CTM** can provide you the information. If you are an active developer on FreeBSD, but have lousy or non-existent TCP/IP connectivity, or simply wish to have the changes automatically sent to you, **CTM** was made for you. You will need to obtain up to three deltas per day for the most active branches. However, you should consider having them sent by automatic email. The sizes of the updates are always kept as small as possible. This is typically less than 5K, with an occasional (one in ten) being 10-50K and every now and then a large 100K+ or more coming around.

You will also need to make yourself aware of the various caveats related to working directly from the development sources rather than a pre-packaged release. This is particularly true if you choose the “current” sources. It is recommended that you read Staying current with FreeBSD.

A.4.2 What Do I Need to Use CTM?

You will need two things: The **CTM** program, and the initial deltas to feed it (to get up to “current” levels).

The **CTM** program has been part of FreeBSD ever since version 2.0 was released, and lives in `/usr/src/usr.sbin/ctm` if you have a copy of the source available.

The “deltas” you feed **CTM** can be had two ways, FTP or email. If you have general FTP access to the Internet then the following FTP sites support access to **CTM**:

ftp://ftp.FreeBSD.org/pub/FreeBSD/CTM/

or see section mirrors.

FTP the relevant directory and fetch the `README` file, starting from there.

If you wish to get your deltas via email:

Subscribe to one of the **CTM** distribution lists. ctm-cvs-cur (http://lists.FreeBSD.org/mailman/listinfo/ctm-cvs-cur) supports the entire CVS tree. ctm-src-cur (http://lists.FreeBSD.org/mailman/listinfo/ctm-src-cur) supports the head of the development branch. ctm-src-4 (http://lists.FreeBSD.org/mailman/listinfo/ctm-src-4) supports the 4.X release branch, etc.. (If you do not know how to subscribe yourself to a list, click on the list name above or go to http://lists.FreeBSD.org/mailman/listinfo and click on the list that you wish to subscribe to. The list page should contain all of the necessary subscription instructions.)

When you begin receiving your **CTM** updates in the mail, you may use the `ctm_rmail` program to unpack and apply them. You can actually use the `ctm_rmail` program directly from a entry in `/etc/aliases` if you want to have the process run in a fully automated fashion. Check the `ctm_rmail` manual page for more details.

> **Note:** No matter what method you use to get the **CTM** deltas, you should subscribe to the ctm-announce (http://lists.FreeBSD.org/mailman/listinfo/ctm-announce) mailing list. In the future, this will be the only place where announcements concerning the operations of the **CTM** system will be posted. Click on the list name above and follow the instructions to subscribe to the list.

A.4.3 Using CTM for the First Time

Before you can start using **CTM** deltas, you will need to get to a starting point for the deltas produced subsequently to it.

First you should determine what you already have. Everyone can start from an “empty” directory. You must use an initial “Empty” delta to start off your **CTM** supported tree. At some point it is intended that one of these “started” deltas be distributed on the CD for your convenience, however, this does not currently happen.

Since the trees are many tens of megabytes, you should prefer to start from something already at hand. If you have a -RELEASE CD, you can copy or extract an initial source from it. This will save a significant transfer of data.

You can recognize these “starter” deltas by the `X` appended to the number (`src-cur.3210XEmpty.gz` for instance). The designation following the `X` corresponds to the origin of your initial “seed”. `Empty` is an empty directory. As a rule a base transition from `Empty` is produced every 100 deltas. By the way, they are large! 70 to 80 Megabytes of `gzip`’d data is common for the `XEmpty` deltas.

Once you have picked a base delta to start from, you will also need all deltas with higher numbers following it.

A.4.4 Using CTM in Your Daily Life

To apply the deltas, simply say:

```
# cd /where/ever/you/want/the/stuff
# ctm -v -v /where/you/store/your/deltas/src-xxx.*
```

CTM understands deltas which have been put through `gzip`, so you do not need to `gunzip` them first, this saves disk space.

Unless it feels very secure about the entire process, **CTM** will not touch your tree. To verify a delta you can also use the `-c` flag and **CTM** will not actually touch your tree; it will merely verify the integrity of the delta and see if it would apply cleanly to your current tree.

There are other options to **CTM** as well, see the manual pages or look in the sources for more information.

That is really all there is to it. Every time you get a new delta, just run it through **CTM** to keep your sources up to date.

Do not remove the deltas if they are hard to download again. You just might want to keep them around in case something bad happens. Even if you only have floppy disks, consider using `fdwrite` to make a copy.

A.4.5 Keeping Your Local Changes

As a developer one would like to experiment with and change files in the source tree. **CTM** supports local modifications in a limited way: before checking for the presence of a file `foo`, it first looks for `foo.ctm`. If this file exists, **CTM** will operate on it instead of `foo`.

This behavior gives us a simple way to maintain local changes: simply copy the files you plan to modify to the corresponding file names with a `.ctm` suffix. Then you can freely hack the code, while **CTM** keeps the `.ctm` file up-to-date.

A.4.6 Other Interesting CTM Options

A.4.6.1 Finding Out Exactly What Would Be Touched by an Update

You can determine the list of changes that **CTM** will make on your source repository using the `-l` option to **CTM**.

This is useful if you would like to keep logs of the changes, pre- or post- process the modified files in any manner, or just are feeling a tad paranoid.

A.4.6.2 Making Backups Before Updating

Sometimes you may want to backup all the files that would be changed by a **CTM** update.

Specifying the `-B backup-file` option causes **CTM** to backup all files that would be touched by a given **CTM** delta to `backup-file`.

A.4.6.3 Restricting the Files Touched by an Update

Sometimes you would be interested in restricting the scope of a given **CTM** update, or may be interested in extracting just a few files from a sequence of deltas.

You can control the list of files that **CTM** would operate on by specifying filtering regular expressions using the `-e` and `-x` options.

For example, to extract an up-to-date copy of `lib/libc/Makefile` from your collection of saved **CTM** deltas, run the commands:

```
# cd /where/ever/you/want/to/extract/it/
# ctm -e '^lib/libc/Makefile' ~ctm/src-xxx.*
```

For every file specified in a **CTM** delta, the `-e` and `-x` options are applied in the order given on the command line. The file is processed by **CTM** only if it is marked as eligible after all the `-e` and `-x` options are applied to it.

A.4.7 Future Plans for CTM

Tons of them:

- Use some kind of authentication into the **CTM** system, so as to allow detection of spoofed **CTM** updates.
- Clean up the options to **CTM**, they became confusing and counter intuitive.

A.4.8 Miscellaneous Stuff

There is a sequence of deltas for the `ports` collection too, but interest has not been all that high yet.

A.4.9 CTM Mirrors

CTM/FreeBSD is available via anonymous FTP from the following mirror sites. If you choose to obtain **CTM** via anonymous FTP, please try to use a site near you.

In case of problems, please contact the ctm-users (http://lists.FreeBSD.org/mailman/listinfo/ctm-users) mailing list.

California, Bay Area, official source

- ftp://ftp.FreeBSD.org/pub/FreeBSD/development/CTM/

South Africa, backup server for old deltas

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/CTM/

Taiwan/R.O.C.(台灣/中華民國)

- ftp://ctm.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm2.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm3.tw.FreeBSD.org/pub/FreeBSD/development/CTM/

If you did not find a mirror near to you or the mirror is incomplete, try to use a search engine such as alltheweb (http://www.alltheweb.com/).

A.5 Using CVSup

A.5.1 CVSup 簡介

CVSup is a software package for distributing and updating source trees from a master CVS repository on a remote server host. The FreeBSD sources are maintained in a CVS repository on a central development machine in California. With **CVSup**, FreeBSD users can easily keep their own source trees up to date.

CVSup uses the so-called *pull* model of updating. Under the pull model, each client asks the server for updates, if and when they are wanted. The server waits passively for update requests from its clients. Thus all updates are instigated by the client. The server never sends unsolicited updates. Users must either run the **CVSup** client manually to get an update, or they must set up a `cron` job to run it automatically on a regular basis.

The term **CVSup**, capitalized just so, refers to the entire software package. Its main components are the client `cvsup` which runs on each user's machine, and the server `cvsupd` which runs at each of the FreeBSD mirror sites.

As you read the FreeBSD documentation and mailing lists, you may see references to **sup**. **Sup** was the predecessor of **CVSup**, and it served a similar purpose. **CVSup** is used much in the same way as sup and, in fact, uses

configuration files which are backward-compatible with `sup`'s. **Sup** is no longer used in the FreeBSD project, because **CVSup** is both faster and more flexible.

A.5.2 Installation

The easiest way to install **CVSup** is to use the precompiled `net/cvsup` package from the FreeBSD packages collection. If you prefer to build **CVSup** from source, you can use the `net/cvsup` port instead. But be forewarned: the `net/cvsup` port depends on the Modula-3 system, which takes a substantial amount of time and disk space to download and build.

> **Note:** If you are going to be using **CVSup** on a machine which will not have **XFree86** or **Xorg** installed, such as a server, be sure to use the port which does not include the **CVSup** GUI, `net/cvsup-without-gui`.

A.5.3 CVSup Configuration

CVSup's operation is controlled by a configuration file called the `supfile`. There are some sample `supfiles` in the directory `/usr/share/examples/cvsup/`.

The information in a `supfile` answers the following questions for **CVSup**:

- Which files do you want to receive?
- Which versions of them do you want?
- Where do you want to get them from?
- Where do you want to put them on your own machine?
- Where do you want to put your status files?

In the following sections, we will construct a typical `supfile` by answering each of these questions in turn. First, we describe the overall structure of a `supfile`.

A `supfile` is a text file. Comments begin with # and extend to the end of the line. Lines that are blank and lines that contain only comments are ignored.

Each remaining line describes a set of files that the user wishes to receive. The line begins with the name of a "collection", a logical grouping of files defined by the server. The name of the collection tells the server which files you want. After the collection name come zero or more fields, separated by white space. These fields answer the questions listed above. There are two types of fields: flag fields and value fields. A flag field consists of a keyword standing alone, e.g., `delete` or `compress`. A value field also begins with a keyword, but the keyword is followed without intervening white space by = and a second word. For example, `release=cvs` is a value field.

A `supfile` typically specifies more than one collection to receive. One way to structure a `supfile` is to specify all of the relevant fields explicitly for each collection. However, that tends to make the `supfile` lines quite long, and it is inconvenient because most fields are the same for all of the collections in a `supfile`. **CVSup** provides a defaulting mechanism to avoid these problems. Lines beginning with the special pseudo-collection name `*default` can be used to set flags and values which will be used as defaults for the subsequent collections in the `supfile`. A default value can be overridden for an individual collection, by specifying a different value with the collection itself. Defaults can also be changed or augmented in mid-supfile by additional `*default` lines.

With this background, we will now proceed to construct a `supfile` for receiving and updating the main source tree of FreeBSD-CURRENT.

- Which files do you want to receive?

  The files available via **CVSup** are organized into named groups called “collections”. The collections that are available are described in the following section. In this example, we wish to receive the entire main source tree for the FreeBSD system. There is a single large collection `src-all` which will give us all of that. As a first step toward constructing our `supfile`, we simply list the collections, one per line (in this case, only one line):

  ```
  src-all
  ```

- Which version(s) of them do you want?

  With **CVSup**, you can receive virtually any version of the sources that ever existed. That is possible because the **cvsupd** server works directly from the CVS repository, which contains all of the versions. You specify which one of them you want using the `tag=` and `date=` value fields.

  > **Warning:** Be very careful to specify any `tag=` fields correctly. Some tags are valid only for certain collections of files. If you specify an incorrect or misspelled tag, **CVSup** will delete files which you probably do not want deleted. In particular, use *only* `tag=.` for the `ports-*` collections.

  The `tag=` field names a symbolic tag in the repository. There are two kinds of tags, revision tags and branch tags. A revision tag refers to a specific revision. Its meaning stays the same from day to day. A branch tag, on the other hand, refers to the latest revision on a given line of development, at any given time. Because a branch tag does not refer to a specific revision, it may mean something different tomorrow than it means today.

  Section A.7 contains branch tags that users might be interested in. When specifying a tag in **CVSup**’s configuration file, it must be preceded with `tag=` (`RELENG_4` will become `tag=RELENG_4`). Keep in mind that only the `tag=.` is relevant for the Ports Collection.

  > **Warning:** Be very careful to type the tag name exactly as shown. **CVSup** cannot distinguish between valid and invalid tags. If you misspell the tag, **CVSup** will behave as though you had specified a valid tag which happens to refer to no files at all. It will delete your existing sources in that case.

  When you specify a branch tag, you normally receive the latest versions of the files on that line of development. If you wish to receive some past version, you can do so by specifying a date with the `date=` value field. The cvsup(1) manual page explains how to do that.

  For our example, we wish to receive FreeBSD-CURRENT. We add this line at the beginning of our `supfile`:

  ```
  *default tag=.
  ```

  There is an important special case that comes into play if you specify neither a `tag=` field nor a `date=` field. In that case, you receive the actual RCS files directly from the server’s CVS repository, rather than receiving a particular version. Developers generally prefer this mode of operation. By maintaining a copy of the repository itself on their systems, they gain the ability to browse the revision histories and examine past versions of files. This gain is achieved at a large cost in terms of disk space, however.

- Where do you want to get them from?

We use the `host=` field to tell `cvsup` where to obtain its updates. Any of the CVSup mirror sites will do, though you should try to select one that is close to you in cyberspace. In this example we will use a fictional FreeBSD distribution site, `cvsup99.FreeBSD.org`:

```
*default host=cvsup99.FreeBSD.org
```

You will need to change the host to one that actually exists before running **CVSup**. On any particular run of `cvsup`, you can override the host setting on the command line, with `-h` *`hostname`*.

- Where do you want to put them on your own machine?

  The `prefix=` field tells `cvsup` where to put the files it receives. In this example, we will put the source files directly into our main source tree, `/usr/src`. The `src` directory is already implicit in the collections we have chosen to receive, so this is the correct specification:

  ```
  *default prefix=/usr
  ```

- Where should `cvsup` maintain its status files?

  The **CVSup** client maintains certain status files in what is called the "base" directory. These files help **CVSup** to work more efficiently, by keeping track of which updates you have already received. We will use the standard base directory, `/var/db`:

  ```
  *default base=/var/db
  ```

  If your base directory does not already exist, now would be a good time to create it. The `cvsup` client will refuse to run if the base directory does not exist.

- Miscellaneous `supfile` settings:

  There is one more line of boiler plate that normally needs to be present in the `supfile`:

  ```
  *default release=cvs delete use-rel-suffix compress
  ```

  `release=cvs` indicates that the server should get its information out of the main FreeBSD CVS repository. This is virtually always the case, but there are other possibilities which are beyond the scope of this discussion.

  `delete` gives **CVSup** permission to delete files. You should always specify this, so that **CVSup** can keep your source tree fully up-to-date. **CVSup** is careful to delete only those files for which it is responsible. Any extra files you happen to have will be left strictly alone.

  `use-rel-suffix` is ... arcane. If you really want to know about it, see the cvsup(1) manual page. Otherwise, just specify it and do not worry about it.

  `compress` enables the use of gzip-style compression on the communication channel. If your network link is T1 speed or faster, you probably should not use compression. Otherwise, it helps substantially.

- Putting it all together:

  Here is the entire `supfile` for our example:

  ```
  *default tag=.
  *default host=cvsup99.FreeBSD.org
  *default prefix=/usr
  *default base=/var/db
  *default release=cvs delete use-rel-suffix compress

  src-all
  ```

A.5.3.1 The `refuse` File

As mentioned above, **CVSup** uses a *pull method*. Basically, this means that you connect to the **CVSup** server, and it says, "Here is what you can download from me..." , and your client responds "OK, I will take this, this, this, and this." In the default configuration, the **CVSup** client will take every file associated with the collection and tag you chose in the configuration file. However, this is not always what you want, especially if you are synching the `doc`, `ports`, or `www` trees ——most people cannot read four or five languages, and therefore they do not need to download the language-specific files. If you are **CVSup**ing the Ports Collection, you can get around this by specifying each collection individually (e.g., *ports-astrology*, *ports-biology*, etc instead of simply saying *ports-all*). However, since the `doc` and `www` trees do not have language-specific collections, you must use one of **CVSup**'s many nifty features: the `refuse` file.

The `refuse` file essentially tells **CVSup** that it should not take every single file from a collection; in other words, it tells the client to *refuse* certain files from the server. The `refuse` file can be found (or, if you do not yet have one, should be placed) in *`base`*`/sup/`. *`base`* is defined in your `supfile`; our defined *`base`* is `/var/db`, which means that by default the `refuse` file is `/var/db/sup/refuse`.

The `refuse` file has a very simple format; it simply contains the names of files or directories that you do not wish to download. For example, if you cannot speak any languages other than English and some German, and you do not feel the need to read the German translation of documentation, you can put the following in your `refuse` file:

```
doc/bn_*
doc/da_*
doc/de_*
doc/el_*
doc/es_*
doc/fr_*
doc/it_*
doc/ja_*
doc/nl_*
doc/no_*
doc/pl_*
doc/pt_*
doc/ru_*
doc/sr_*
doc/tr_*
doc/zh_*
```

and so forth for the other languages (you can find the full list by browsing the FreeBSD CVS repository (http://www.FreeBSD.org/cgi/cvsweb.cgi/)).

With this very useful feature, those users who are on slow links or pay by the minute for their Internet connection will be able to save valuable time as they will no longer need to download files that they will never use. For more information on `refuse` files and other neat features of **CVSup**, please view its manual page.

A.5.4 Running CVSup

You are now ready to try an update. The command line for doing this is quite simple:

```
# cvsup supfile
```

where *supfile* is of course the name of the supfile you have just created. Assuming you are running under X11, cvsup will display a GUI window with some buttons to do the usual things. Press the go button, and watch it run.

Since you are updating your actual /usr/src tree in this example, you will need to run the program as root so that cvsup has the permissions it needs to update your files. Having just created your configuration file, and having never used this program before, that might understandably make you nervous. There is an easy way to do a trial run without touching your precious files. Just create an empty directory somewhere convenient, and name it as an extra argument on the command line:

```
# mkdir /var/tmp/dest
# cvsup supfile /var/tmp/dest
```

The directory you specify will be used as the destination directory for all file updates. **CVSup** will examine your usual files in /usr/src, but it will not modify or delete any of them. Any file updates will instead land in /var/tmp/dest/usr/src. **CVSup** will also leave its base directory status files untouched when run this way. The new versions of those files will be written into the specified directory. As long as you have read access to /usr/src, you do not even need to be root to perform this kind of trial run.

If you are not running X11 or if you just do not like GUIs, you should add a couple of options to the command line when you run cvsup:

```
# cvsup -g -L 2 supfile
```

The -g tells **CVSup** not to use its GUI. This is automatic if you are not running X11, but otherwise you have to specify it.

The -L 2 tells **CVSup** to print out the details of all the file updates it is doing. There are three levels of verbosity, from -L 0 to -L 2. The default is 0, which means total silence except for error messages.

There are plenty of other options available. For a brief list of them, type cvsup -H. For more detailed descriptions, see the manual page.

Once you are satisfied with the way updates are working, you can arrange for regular runs of **CVSup** using cron(8). Obviously, you should not let **CVSup** use its GUI when running it from cron(8).

A.5.5 CVSup File Collections

The file collections available via **CVSup** are organized hierarchically. There are a few large collections, and they are divided into smaller sub-collections. Receiving a large collection is equivalent to receiving each of its sub-collections. The hierarchical relationships among collections are reflected by the use of indentation in the list below.

The most commonly used collections are src-all, and ports-all. The other collections are used only by small groups of people for specialized purposes, and some mirror sites may not carry all of them.

cvs-all release=cvs

The main FreeBSD CVS repository, including the cryptography code.

distrib release=cvs

Files related to the distribution and mirroring of FreeBSD.

`doc-all release=cvs`

Sources for the FreeBSD Handbook and other documentation. This does not include files for the FreeBSD web site.

`ports-all release=cvs`

The FreeBSD Ports Collection.

> **Important:** If you do not want to update the whole of `ports-all` (the whole ports tree), but use one of the subcollections listed below, make sure that you *always* update the `ports-base` subcollection! Whenever something changes in the ports build infrastructure represented by `ports-base`, it is virtually certain that those changes will be used by "real" ports real soon. Thus, if you only update the "real" ports and they use some of the new features, there is a very high chance that their build will fail with some mysterious error message. The *very first* thing to do in this case is to make sure that your `ports-base` subcollection is up to date.

> **Important:** If you are going to be building your own local copy of `ports/INDEX`, you *must* accept `ports-all` (the whole ports tree). Building `ports/INDEX` with a partial tree is not supported. See the FAQ (http://www.FreeBSD.org/doc/zh_TW.Big5/books/faq/applications.html#MAKE-INDEX).

`ports-accessibility release=cvs`

Software to help disabled users.

`ports-arabic release=cvs`

Arabic language support.

`ports-archivers release=cvs`

Archiving tools.

`ports-astro release=cvs`

Astronomical ports.

`ports-audio release=cvs`

Sound support.

`ports-base release=cvs`

The Ports Collection build infrastructure - various files located in the `Mk/` and `Tools/` subdirectories of `/usr/ports`.

> **Note:** Please see the important warning above: you should *always* update this subcollection, whenever you update any part of the FreeBSD Ports Collection!

`ports-benchmarks release=cvs`

Benchmarks.

`ports-biology release=cvs`

Biology.

`ports-cad release=cvs`

Computer aided design tools.

`ports-chinese release=cvs`

Chinese language support.

`ports-comms release=cvs`

Communication software.

`ports-converters release=cvs`

character code converters.

`ports-databases release=cvs`

Databases.

`ports-deskutils release=cvs`

Things that used to be on the desktop before computers were invented.

`ports-devel release=cvs`

Development utilities.

`ports-dns release=cvs`

DNS related software.

`ports-editors release=cvs`

Editors.

`ports-emulators release=cvs`

Emulators for other operating systems.

`ports-finance release=cvs`

Monetary, financial and related applications.

`ports-ftp release=cvs`

FTP client and server utilities.

`ports-games release=cvs`

Games.

`ports-german release=cvs`

German language support.

`ports-graphics release=cvs`

Graphics utilities.

`ports-hebrew release=cvs`

Hebrew language support.

`ports-hungarian release=cvs`

Hungarian language support.

`ports-irc release=cvs`

Internet Relay Chat utilities.

`ports-japanese release=cvs`

Japanese language support.

`ports-java release=cvs`

Java utilities.

`ports-korean release=cvs`

Korean language support.

`ports-lang release=cvs`

Programming languages.

`ports-mail release=cvs`

Mail software.

`ports-math release=cvs`

Numerical computation software.

`ports-mbone release=cvs`

MBone applications.

`ports-misc release=cvs`

Miscellaneous utilities.

`ports-multimedia release=cvs`

Multimedia software.

`ports-net release=cvs`

Networking software.

`ports-net-im release=cvs`

Instant messaging software.

`ports-net-mgmt release=cvs`

Network management software.

`ports-net-p2p release=cvs`

Peer to peer networking.

`ports-news release=cvs`

USENET news software.

`ports-palm release=cvs`

Software support for Palm™ series.

`ports-polish release=cvs`

Polish language support.

`ports-portuguese release=cvs`

Portuguese language support.

`ports-print release=cvs`

Printing software.

`ports-russian release=cvs`

Russian language support.

`ports-science release=cvs`

Science.

`ports-security release=cvs`

Security utilities.

`ports-shells release=cvs`

Command line shells.

`ports-sysutils release=cvs`

System utilities.

`ports-textproc release=cvs`

text processing utilities (does not include desktop publishing).

`ports-ukrainian release=cvs`

Ukrainian language support.

`ports-vietnamese release=cvs`

Vietnamese language support.

`ports-www release=cvs`

Software related to the World Wide Web.

`ports-x11 release=cvs`

Ports to support the X window system.

`ports-x11-clocks release=cvs`

X11 clocks.

`ports-x11-fm release=cvs`

X11 file managers.

`ports-x11-fonts release=cvs`

X11 fonts and font utilities.

`ports-x11-toolkits release=cvs`

X11 toolkits.

`ports-x11-servers release=cvs`

X11 servers.

`ports-x11-themes release=cvs`

X11 themes.

`ports-x11-wm release=cvs`

X11 window managers.

`projects-all release=cvs`

Sources for the FreeBSD projects repository.

`src-all release=cvs`

The main FreeBSD sources, including the cryptography code.

`src-base release=cvs`

Miscellaneous files at the top of `/usr/src`.

`src-bin release=cvs`

User utilities that may be needed in single-user mode (`/usr/src/bin`).

`src-contrib release=cvs`

Utilities and libraries from outside the FreeBSD project, used relatively unmodified (`/usr/src/contrib`).

`src-crypto release=cvs`

Cryptography utilities and libraries from outside the FreeBSD project, used relatively unmodified (`/usr/src/crypto`).

`src-eBones release=cvs`

Kerberos and DES (`/usr/src/eBones`). Not used in current releases of FreeBSD.

`src-etc release=cvs`

System configuration files (`/usr/src/etc`).

`src-games release=cvs`

Games (`/usr/src/games`).

`src-gnu release=cvs`

Utilities covered by the GNU Public License (`/usr/src/gnu`).

`src-include release=cvs`

Header files (`/usr/src/include`).

`src-kerberos5 release=cvs`

Kerberos5 security package (`/usr/src/kerberos5`).

`src-kerberosIV release=cvs`

KerberosIV security package (`/usr/src/kerberosIV`).

`src-lib release=cvs`

Libraries (`/usr/src/lib`).

`src-libexec release=cvs`

System programs normally executed by other programs (`/usr/src/libexec`).

`src-release release=cvs`

Files required to produce a FreeBSD release (`/usr/src/release`).

`src-sbin release=cvs`

System utilities for single-user mode (`/usr/src/sbin`).

`src-secure release=cvs`

Cryptographic libraries and commands (`/usr/src/secure`).

`src-share release=cvs`

Files that can be shared across multiple systems (`/usr/src/share`).

`src-sys release=cvs`

The kernel (`/usr/src/sys`).

`src-sys-crypto release=cvs`

Kernel cryptography code (`/usr/src/sys/crypto`).

`src-tools release=cvs`

Various tools for the maintenance of FreeBSD (`/usr/src/tools`).

`src-usrbin release=cvs`

User utilities (`/usr/src/usr.bin`).

`src-usrsbin release=cvs`

System utilities (`/usr/src/usr.sbin`).

`www release=cvs`

The sources for the FreeBSD WWW site.

`distrib release=self`

The **CVSup** server's own configuration files. Used by **CVSup** mirror sites.

`gnats release=current`

The GNATS bug-tracking database.

`mail-archive release=current`

FreeBSD mailing list archive.

`www release=current`

The pre-processed FreeBSD WWW site files (not the source files). Used by WWW mirror sites.

A.5.6 For More Information

For the **CVSup** FAQ and other information about **CVSup**, see The CVSup Home Page (http://www.polstra.com/projects/freeware/CVSup/).

Most FreeBSD-related discussion of **CVSup** takes place on the FreeBSD technical discussions 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers). New versions of the software are announced there, as well as on the FreeBSD announcements 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce).

Questions and bug reports should be addressed to the author of the program at <`cvsup-bugs@polstra.com`>.

A.5.7 CVSup Sites

CVSup servers for FreeBSD are running at the following sites:

Central Servers, Primary Mirror Sites, Argentina, Armenia, Australia, Austria, Brazil, Bulgaria, Canada, China, Costa Rica, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Indonesia, Ireland, Israel, Italy, Japan, Korea, Kuwait, Kyrgyzstan, Latvia, Lithuania, Netherlands, New Zealand, Norway, Philippines, Poland, Portugal, Romania, Russia, San Marino, Singapore, Slovak Republic, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Thailand, Turkey, Ukraine, United Kingdom, USA.

(as of 2009/05/21 21:02:58 UTC)

Central Servers

- cvsup.FreeBSD.org

Primary Mirror Sites

- cvsup1.FreeBSD.org
- cvsup2.FreeBSD.org
- cvsup3.FreeBSD.org
- cvsup4.FreeBSD.org
- cvsup5.FreeBSD.org
- cvsup6.FreeBSD.org
- cvsup7.FreeBSD.org
- cvsup8.FreeBSD.org
- cvsup9.FreeBSD.org
- cvsup10.FreeBSD.org
- cvsup11.FreeBSD.org
- cvsup12.FreeBSD.org
- cvsup13.FreeBSD.org
- cvsup14.FreeBSD.org
- cvsup15.FreeBSD.org
- cvsup16.FreeBSD.org
- cvsup18.FreeBSD.org

Argentina

- cvsup.ar.FreeBSD.org

Armenia

- cvsup1.am.FreeBSD.org

Australia

- cvsup.au.FreeBSD.org

Austria

- cvsup.at.FreeBSD.org
- cvsup2.at.FreeBSD.org

Brazil

- cvsup.br.FreeBSD.org
- cvsup2.br.FreeBSD.org
- cvsup3.br.FreeBSD.org
- cvsup4.br.FreeBSD.org
- cvsup5.br.FreeBSD.org

Bulgaria

- cvsup.bg.FreeBSD.org

Canada

- cvsup1.ca.FreeBSD.org

China

- cvsup.cn.FreeBSD.org
- cvsup2.cn.FreeBSD.org
- cvsup3.cn.FreeBSD.org
- cvsup4.cn.FreeBSD.org
- cvsup5.cn.FreeBSD.org

Costa Rica

- cvsup1.cr.FreeBSD.org

Czech Republic

- cvsup.cz.FreeBSD.org

Denmark

- cvsup.dk.FreeBSD.org
- cvsup2.dk.FreeBSD.org

Estonia

- cvsup.ee.FreeBSD.org

Finland

- cvsup.fi.FreeBSD.org
- cvsup2.fi.FreeBSD.org

France

- cvsup.fr.FreeBSD.org

- cvsup1.fr.FreeBSD.org
- cvsup2.fr.FreeBSD.org
- cvsup3.fr.FreeBSD.org
- cvsup4.fr.FreeBSD.org
- cvsup5.fr.FreeBSD.org
- cvsup8.fr.FreeBSD.org

Germany

- cvsup.de.FreeBSD.org
- cvsup2.de.FreeBSD.org
- cvsup3.de.FreeBSD.org
- cvsup4.de.FreeBSD.org
- cvsup5.de.FreeBSD.org
- cvsup6.de.FreeBSD.org
- cvsup7.de.FreeBSD.org
- cvsup8.de.FreeBSD.org

Greece

- cvsup.gr.FreeBSD.org
- cvsup2.gr.FreeBSD.org

Hungary

- cvsup.hu.FreeBSD.org

Iceland

- cvsup.is.FreeBSD.org

Indonesia

- cvsup.id.FreeBSD.org

Ireland

- cvsup.ie.FreeBSD.org
- cvsup2.ie.FreeBSD.org

Israel

- cvsup.il.FreeBSD.org

Italy

- cvsup.it.FreeBSD.org

Japan

- cvsup.jp.FreeBSD.org
- cvsup2.jp.FreeBSD.org
- cvsup3.jp.FreeBSD.org
- cvsup4.jp.FreeBSD.org
- cvsup5.jp.FreeBSD.org
- cvsup6.jp.FreeBSD.org

Korea

- cvsup.kr.FreeBSD.org
- cvsup2.kr.FreeBSD.org
- cvsup3.kr.FreeBSD.org

Kuwait

- cvsup1.kw.FreeBSD.org

Kyrgyzstan

- cvsup.kg.FreeBSD.org

Latvia

- cvsup.lv.FreeBSD.org
- cvsup2.lv.FreeBSD.org

Lithuania

- cvsup.lt.FreeBSD.org
- cvsup2.lt.FreeBSD.org
- cvsup3.lt.FreeBSD.org

Netherlands

- cvsup.nl.FreeBSD.org
- cvsup2.nl.FreeBSD.org
- cvsup3.nl.FreeBSD.org

New Zealand

- cvsup.nz.FreeBSD.org
- cvsup2.nz.FreeBSD.org

Norway

- cvsup.no.FreeBSD.org

Philippines

- cvsup1.ph.FreeBSD.org

Poland

- cvsup.pl.FreeBSD.org
- cvsup2.pl.FreeBSD.org
- cvsup3.pl.FreeBSD.org

Portugal

- cvsup.pt.FreeBSD.org
- cvsup2.pt.FreeBSD.org
- cvsup3.pt.FreeBSD.org

Romania

- cvsup.ro.FreeBSD.org
- cvsup1.ro.FreeBSD.org
- cvsup2.ro.FreeBSD.org
- cvsup3.ro.FreeBSD.org

Russia

- cvsup.ru.FreeBSD.org
- cvsup2.ru.FreeBSD.org
- cvsup3.ru.FreeBSD.org
- cvsup4.ru.FreeBSD.org
- cvsup5.ru.FreeBSD.org
- cvsup6.ru.FreeBSD.org
- cvsup7.ru.FreeBSD.org

San Marino

- cvsup.sm.FreeBSD.org

Singapore

- cvsup.sg.FreeBSD.org

Slovak Republic

- cvsup.sk.FreeBSD.org

Slovenia

- cvsup.si.FreeBSD.org
- cvsup2.si.FreeBSD.org

South Africa

- cvsup.za.FreeBSD.org
- cvsup2.za.FreeBSD.org

Spain

- cvsup.es.FreeBSD.org
- cvsup2.es.FreeBSD.org
- cvsup3.es.FreeBSD.org

Sweden

- cvsup.se.FreeBSD.org
- cvsup3.se.FreeBSD.org

Switzerland

- cvsup.ch.FreeBSD.org

Taiwan

- cvsup.tw.FreeBSD.org
- cvsup3.tw.FreeBSD.org
- cvsup4.tw.FreeBSD.org
- cvsup5.tw.FreeBSD.org
- cvsup6.tw.FreeBSD.org
- cvsup7.tw.FreeBSD.org
- cvsup8.tw.FreeBSD.org
- cvsup9.tw.FreeBSD.org
- cvsup10.tw.FreeBSD.org
- cvsup11.tw.FreeBSD.org
- cvsup12.tw.FreeBSD.org
- cvsup13.tw.FreeBSD.org
- cvsup14.tw.FreeBSD.org

Thailand

- cvsup.th.FreeBSD.org

Turkey

- cvsup.tr.FreeBSD.org
- cvsup2.tr.FreeBSD.org

Ukraine

- cvsup2.ua.FreeBSD.org

- cvsup3.ua.FreeBSD.org
- cvsup4.ua.FreeBSD.org
- cvsup5.ua.FreeBSD.org
- cvsup6.ua.FreeBSD.org
- cvsup7.ua.FreeBSD.org

United Kingdom

- cvsup.uk.FreeBSD.org
- cvsup2.uk.FreeBSD.org
- cvsup3.uk.FreeBSD.org
- cvsup4.uk.FreeBSD.org

USA

- cvsup1.us.FreeBSD.org
- cvsup2.us.FreeBSD.org
- cvsup3.us.FreeBSD.org
- cvsup4.us.FreeBSD.org
- cvsup5.us.FreeBSD.org
- cvsup6.us.FreeBSD.org
- cvsup7.us.FreeBSD.org
- cvsup8.us.FreeBSD.org
- cvsup9.us.FreeBSD.org
- cvsup10.us.FreeBSD.org
- cvsup11.us.FreeBSD.org
- cvsup12.us.FreeBSD.org
- cvsup13.us.FreeBSD.org
- cvsup14.us.FreeBSD.org
- cvsup15.us.FreeBSD.org
- cvsup16.us.FreeBSD.org
- cvsup18.us.FreeBSD.org

A.6 Using Portsnap

A.6.1 Portsnap 簡介

Portsnap is a system for securely distributing the FreeBSD ports tree. Approximately once an hour, a "snapshot" of the ports tree is generated, repackaged, and cryptographically signed. The resulting files are then distributed via HTTP.

Like **CVSup**, **Portsnap** uses a *pull* model of updating: The packaged and signed ports trees are placed on a web server which waits passively for clients to request files. Users must either run portsnap(8) manually to download updates or set up a cron(8) job to download updates automatically on a regular basis.

For technical reasons, **Portsnap** does not update the "live" ports tree in `/usr/ports/` directly; instead, it works via a compressed copy of the ports tree stored in `/var/db/portsnap/` by default. This compressed copy is then used to update the live ports tree.

> **Note:** If **Portsnap** is installed from the FreeBSD Ports Collection, then the default location for its compressed snapshot will be `/usr/local/portsnap/` instead of `/var/db/portsnap/`.

A.6.2 Installation

On FreeBSD 6.0 and more recent versions, **Portsnap** is contained in the FreeBSD base system. On older versions of FreeBSD, it can be installed using the `sysutils/portsnap` port.

A.6.3 Portsnap Configuration

Portsnap's operation is controlled by the `/etc/portsnap.conf` configuration file. For most users, the default configuration file will suffice; for more details, consult the portsnap.conf(5) manual page.

> **Note:** If **Portsnap** is installed from the FreeBSD Ports Collection, it will use the configuration file `/usr/local/etc/portsnap.conf` instead of `/etc/portsnap.conf`. This configuration file is not created when the port is installed, but a sample configuration file is distributed; to copy it into place, run the following command:
>
> ```
> # cd /usr/local/etc && cp portsnap.conf.sample portsnap.conf
> ```

A.6.4 Running Portsnap for the First Time

The first time portsnap(8) is run, it will need to download a compressed snapshot of the entire ports tree into `/var/db/portsnap/` (or `/usr/local/portsnap/` if **Portsnap** was installed from the Ports Collection). For the beginning of 2006 this is approximately a 41 MB download.

```
# portsnap fetch
```

Once the compressed snapshot has been downloaded, a "live" copy of the ports tree can be extracted into `/usr/ports/`. This is necessary even if a ports tree has already been created in that directory (e.g., by using **CVSup**), since it establishes a baseline from which `portsnap` can determine which parts of the ports tree need to be updated later.

```
# portsnap extract
```

> **Note:** In the default installation `/usr/ports` is not created. If you run FreeBSD 6.0-RELEASE, it should be created before `portsnap` is used. On more recent versions of FreeBSD or **Portsnap**, this operation will be done automatically at first use of the `portsnap` command.

A.6.5 Updating the Ports Tree

After an initial compressed snapshot of the ports tree has been downloaded and extracted into `/usr/ports/`, updating the ports tree consists of two steps: *fetch*ing updates to the compressed snapshot, and using them to *update* the live ports tree. These two steps can be specified to `portsnap` as a single command:

```
# portsnap fetch update
```

> **Note:** Some older versions of `portsnap` do not support this syntax; if it fails, try instead the following:
>
> ```
> # portsnap fetch
> # portsnap update
> ```

A.6.6 Running Portsnap from cron

In order to avoid problems with "flash crowds" accessing the **Portsnap** servers, `portsnap fetch` will not run from a cron(8) job. Instead, a special `portsnap cron` command exists, which waits for a random duration up to 3600 seconds before fetching updates.

In addition, it is strongly recommended that `portsnap update` not be run from a `cron` job, since it is liable to cause major problems if it happens to run at the same time as a port is being built or installed. However, it is safe to update the ports' `INDEX` files, and this can be done by passing the `-I` flag to `portsnap`. (Obviously, if `portsnap -I update` is run from `cron`, then it will be necessary to run `portsnap update` without the `-I` flag at a later time in order to update the rest of the tree.)

Adding the following line to `/etc/crontab` will cause `portsnap` to update its compressed snapshot and the `INDEX` files in `/usr/ports/`, and will send an email if any installed ports are out of date:

```
0 3 * * * root portsnap -I cron update && pkg_version -vIL=
```

> **Note:** If the system clock is not set to the local time zone, please replace `3` with a random value between 0 and 23, in order to spread the load on the **Portsnap** servers more evenly.

Note: Some older versions of `portsnap` do not support listing multiple commands (e.g., `cron update`) in the same invocation of `portsnap`. If the line above fails, try replacing `portsnap -I cron update` with `portsnap cron && portsnap -I update`.

A.7 CVS Tags

When obtaining or updating sources using **cvs** or **CVSup**, a revision tag must be specified. A revision tag refers to either a particular line of FreeBSD development, or a specific point in time. The first type are called “branch tags”, and the second type are called “release tags”.

A.7.1 Branch Tags

All of these, with the exception of `HEAD` (which is always a valid tag), only apply to the `src/` tree. The `ports/`, `doc/`, and `www/` trees are not branched.

HEAD

Symbolic name for the main line, or FreeBSD-CURRENT. Also the default when no revision is specified.

In **CVSup**, this tag is represented by a `.` (not punctuation, but a literal `.` character).

Note: In CVS, this is the default when no revision tag is specified. It is usually *not* a good idea to checkout or update to CURRENT sources on a STABLE machine, unless that is your intent.

RELENG_6

The line of development for FreeBSD-6.X, also known as FreeBSD 6-STABLE

RELENG_6_1

The release branch for FreeBSD-6.1, used only for security advisories and other critical fixes.

RELENG_6_0

The release branch for FreeBSD-6.0, used only for security advisories and other critical fixes.

RELENG_5

The line of development for FreeBSD-5.X, also known as FreeBSD 5-STABLE.

RELENG_5_5

The release branch for FreeBSD-5.5, used only for security advisories and other critical fixes.

RELENG_5_4

The release branch for FreeBSD-5.4, used only for security advisories and other critical fixes.

RELENG_5_3

The release branch for FreeBSD-5.3, used only for security advisories and other critical fixes.

RELENG_5_2

The release branch for FreeBSD-5.2 and FreeBSD-5.2.1, used only for security advisories and other critical fixes.

RELENG_5_1

The release branch for FreeBSD-5.1, used only for security advisories and other critical fixes.

RELENG_5_0

The release branch for FreeBSD-5.0, used only for security advisories and other critical fixes.

RELENG_4

The line of development for FreeBSD-4.X, also known as FreeBSD 4-STABLE.

RELENG_4_11

The release branch for FreeBSD-4.11, used only for security advisories and other critical fixes.

RELENG_4_10

The release branch for FreeBSD-4.10, used only for security advisories and other critical fixes.

RELENG_4_9

The release branch for FreeBSD-4.9, used only for security advisories and other critical fixes.

RELENG_4_8

The release branch for FreeBSD-4.8, used only for security advisories and other critical fixes.

RELENG_4_7

The release branch for FreeBSD-4.7, used only for security advisories and other critical fixes.

RELENG_4_6

The release branch for FreeBSD-4.6 and FreeBSD-4.6.2, used only for security advisories and other critical fixes.

RELENG_4_5

The release branch for FreeBSD-4.5, used only for security advisories and other critical fixes.

RELENG_4_4

The release branch for FreeBSD-4.4, used only for security advisories and other critical fixes.

RELENG_4_3

The release branch for FreeBSD-4.3, used only for security advisories and other critical fixes.

RELENG_3

The line of development for FreeBSD-3.X, also known as 3.X-STABLE.

RELENG_2_2

The line of development for FreeBSD-2.2.X, also known as 2.2-STABLE. This branch is mostly obsolete.

A.7.2 Release Tags

These tags refer to a specific point in time when a particular version of FreeBSD was released. The release engineering process is documented in more detail by the Release Engineering Information (http://www.FreeBSD.org/releng/) and Release Process (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/releng/release-proc.html) documents. The `src` tree uses tag names that start with `RELENG_` tags. The `ports` and `doc` trees use tags whose names begin with `RELEASE` tags. Finally, the `www` tree is not tagged with any special name for releases.

RELENG_6_1_0_RELEASE

FreeBSD 6.1

RELENG_6_0_0_RELEASE

FreeBSD 6.0

RELENG_5_5_0_RELEASE

FreeBSD 5.5

RELENG_5_4_0_RELEASE

FreeBSD 5.4

RELENG_4_11_0_RELEASE

FreeBSD 4.11

RELENG_5_3_0_RELEASE

FreeBSD 5.3

RELENG_4_10_0_RELEASE

FreeBSD 4.10

RELENG_5_2_1_RELEASE

FreeBSD 5.2.1

RELENG_5_2_0_RELEASE

FreeBSD 5.2

RELENG_4_9_0_RELEASE

FreeBSD 4.9

RELENG_5_1_0_RELEASE

FreeBSD 5.1

RELENG_4_8_0_RELEASE

FreeBSD 4.8

RELENG_5_0_0_RELEASE

FreeBSD 5.0

RELENG_4_7_0_RELEASE

FreeBSD 4.7

RELENG_4_6_2_RELEASE

FreeBSD 4.6.2

RELENG_4_6_1_RELEASE

FreeBSD 4.6.1

RELENG_4_6_0_RELEASE

FreeBSD 4.6

RELENG_4_5_0_RELEASE

FreeBSD 4.5

RELENG_4_4_0_RELEASE

FreeBSD 4.4

RELENG_4_3_0_RELEASE

FreeBSD 4.3

RELENG_4_2_0_RELEASE

FreeBSD 4.2

RELENG_4_1_1_RELEASE

FreeBSD 4.1.1

RELENG_4_1_0_RELEASE

FreeBSD 4.1

RELENG_4_0_0_RELEASE

FreeBSD 4.0

RELENG_3_5_0_RELEASE

FreeBSD-3.5

RELENG_3_4_0_RELEASE

FreeBSD-3.4

RELENG_3_3_0_RELEASE

FreeBSD-3.3

RELENG_3_2_0_RELEASE

FreeBSD-3.2

RELENG_3_1_0_RELEASE

FreeBSD-3.1

RELENG_3_0_0_RELEASE

FreeBSD-3.0

RELENG_2_2_8_RELEASE

FreeBSD-2.2.8

RELENG_2_2_7_RELEASE

FreeBSD-2.2.7

RELENG_2_2_6_RELEASE

FreeBSD-2.2.6

RELENG_2_2_5_RELEASE

FreeBSD-2.2.5

RELENG_2_2_2_RELEASE

FreeBSD-2.2.2

RELENG_2_2_1_RELEASE

FreeBSD-2.2.1

RELENG_2_2_0_RELEASE

FreeBSD-2.2.0

A.8 AFS Sites

AFS servers for FreeBSD are running at the following sites:

Sweden

The path to the files are: `/afs/stacken.kth.se/ftp/pub/FreeBSD/`

```
stacken.kth.se         # Stacken Computer Club, KTH, Sweden
```

```
130.237.234.43          #hot.stacken.kth.se
130.237.237.230         #fishburger.stacken.kth.se
130.237.234.3           #milko.stacken.kth.se
```

Maintainer <ftp@stacken.kth.se>

A.9 rsync Sites

The following sites make FreeBSD available through the rsync protocol. The **rsync** utility works in much the same way as the rcp(1) command, but has more options and uses the rsync remote-update protocol which transfers only the differences between two sets of files, thus greatly speeding up the synchronization over the network. This is most useful if you are a mirror site for the FreeBSD FTP server, or the CVS repository. The **rsync** suite is available for many operating systems, on FreeBSD, see the `net/rsync` port or use the package.

Czech Republic

rsync://ftp.cz.FreeBSD.org/

Available collections:

- ftp: A partial mirror of the FreeBSD FTP server.
- FreeBSD: A full mirror of the FreeBSD FTP server.

Germany

rsync://grappa.unix-ag.uni-kl.de/

Available collections:

- freebsd-cvs: The full FreeBSD CVS repository.

This machine also mirrors the CVS repositories of the NetBSD and the OpenBSD projects, among others.

Netherlands

rsync://ftp.nl.FreeBSD.org/

Available collections:

- vol/4/freebsd-core: A full mirror of the FreeBSD FTP server.

United Kingdom

rsync://rsync.mirror.ac.uk/

Available collections:

- ftp.FreeBSD.org: A full mirror of the FreeBSD FTP server.

United States of America

rsync://ftp-master.FreeBSD.org/

This server may only be used by FreeBSD primary mirror sites.

Available collections:

· FreeBSD: The master archive of the FreeBSD FTP server.

· acl: The FreeBSD master ACL list.

rsync://ftp13.FreeBSD.org/

Available collections:

· FreeBSD: A full mirror of the FreeBSD FTP server.

Appendix B. 參考文獻

雖然線上說明(manual pages)有提供FreeBSD 各個特定部分明確的說明，但它們卻難免有「小學而大遺」之憾，像是如何讓整個系統運作順暢。因此，身邊有UNIX 系統管理的好書以及好的使用手冊是不可或缺的。

B.1 FreeBSD 相關的書籍、雜誌

非英語的書籍、雜誌：

- FreeBSD 入門與應用(光碟豪華版) (http://jdli.tw.FreeBSD.org/publication/book/freebsd2/index.htm) (繁體中文)，博碩文化 (http://www.drmaster.com.tw/) ，1997。ISBN 9-578-39435-7
- FreeBSD 技術內幕(FreeBSD Unleashed 簡體中譯版)，機械工業出版社 (http://www.hzbook.com/) 。ISBN 7-111-10201-0
- FreeBSD 使用大全第一版(簡體中文)，機械工業出版社。ISBN 7-111-07482-3
- FreeBSD 使用大全第二版(簡體中文)，機械工業出版社。ISBN 7-111-10286-X
- FreeBSD Handbook 第二版(簡體中譯版)，人民郵電出版社 (http://www.ptpress.com.cn/) 。ISBN 7-115-10541-3
- FreeBSD 3.x Internet 高級服務器的架設與管理(簡體中文)，清華大學出版社 (http://www.tup.tsinghua.edu.cn/) 。ISBN 7-900625-66-6
- FreeBSD & Windows 集成組網實務(簡體中文)，中國鐵道出版社 (http://www.tdpress.com/) 。ISBN 7-113-03845-X
- FreeBSD 網站架設實務(簡體中文)，中國鐵道出版社。ISBN 7-113-03423-3
- FreeBSD for PC 98’ers (日文)，SHUWA SystemCo, LTD 。ISBN 4-87966-468-5 C3055 定價2900 日圓。
- FreeBSD (日文)，CUTT。ISBN 4-906391-22-2 C3055 定價2400 日圓。
- Complete Introduction to FreeBSD (http://www.shoeisha.com/book/Detail.asp?bid=650) (日文)，Shoeisha Co., Ltd (http://www.shoeisha.co.jp/)。ISBN 4-88135-473-6 定價3600 日圓。
- Personal UNIX Starter Kit FreeBSD (http://www.ascii.co.jp/pb/book1/shinkan/detail/1322785.html) (日文)，ASCII (http://www.ascii.co.jp/)。ISBN 4-7561-1733-3 定價3000 日圓。
- FreeBSD Handbook (日文譯版)，ASCII (http://www.ascii.co.jp/)。ISBN 4-7561-1580-2 定價3800 日圓。
- FreeBSD mit Methode (德文)，Computer und Literatur Verlag (http://www.cul.de)/Vertrieb Hanser，1998。ISBN 3-932311-31-0
- FreeBSD 4 - Installieren, Konfigurieren, Administrieren (http://www.cul.de/freebsd.html) (德文)，Computer und Literatur Verlag (http://www.cul.de)，2001。ISBN 3-932311-88-4
- FreeBSD 5 - Installieren, Konfigurieren, Administrieren (http://www.cul.de/freebsd.html) (德文)，Computer und Literatur Verlag (http://www.cul.de)，2003。ISBN 3-936546-06-1
- FreeBSD de Luxe (http://www.mitp.de/vmi/mitp/detail/pWert/1343/) (德文)，Verlag Modere Industrie (http://www.mitp.de)，2003。ISBN 3-8266-1343-0
- FreeBSD Install and Utilization Manual (http://www.pc.mycom.co.jp/FreeBSD/install-manual.html) (日文)，Mainichi Communications Inc. (http://www.pc.mycom.co.jp/) ，1998。ISBN 4-8399-0112-0

· Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo *Building Internet Server with FreeBSD (http://maxwell.itb.ac.id/)* (印尼文)，Elex Media Komputindo (http://www.elexmedia.co.id/)。

· FreeBSD 完全探索(Absolute BSD: The Ultimate Guide to FreeBSD 繁體中文譯版)，上奇 (http://www.grandtech.com.tw/)，2003。ISBN 986-7944-92-5

· FreeBSD 6.0架設管理與應用 (http://www.twbsd.org/cht/book/) (繁體中文)，博碩，2006。ISBN 9-575-27878-X

英文的書籍、雜誌：

· Absolute BSD: The Ultimate Guide to FreeBSD (http://www.AbsoluteBSD.com/)，No Starch Press (http://www.nostarch.com/)，2002。ISBN: 1886411743

· The Complete FreeBSD (http://www.freebsdmall.com/cgi-bin/fm/bsdcomp)，O'Reilly (http://www.oreilly.com/)，2003。ISBN: 0596005164

· The FreeBSD Corporate Networker's Guide (http://www.freebsd-corp-net-guide.com/)，Addison-Wesley (http://www.awl.com/aw/)，2000。ISBN: 0201704811

· FreeBSD: An Open-Source Operating System for Your Personal Computer (http://andrsn.stanford.edu/FreeBSD/introbook/)，The Bit Tree Press，2001。ISBN: 0971204500

· Teach Yourself FreeBSD in 24 Hours，Sams (http://www.samspublishing.com/)，2002。ISBN: 0672324245

· FreeBSD 6 Unleashed，Sams (http://www.samspublishing.com/)，2006。ISBN: 0672328755

· FreeBSD: The Complete Reference，McGrawHill (http://books.mcgraw-hill.com)，2003。ISBN: 0072224096

B.2 使用說明手冊

· Computer Systems Research Group, UC Berkeley. *4.4BSD User's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-075-9

· Computer Systems Research Group, UC Berkeley. *4.4BSD User's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-076-7

· *UNIX in a Nutshell*. O'Reilly & Associates, Inc., 1990. ISBN 093717520X

· Mui, Linda. *What You Need To Know When You Can't Find Your UNIX System Administrator*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-104-6

· Ohio State University (http://www.osu.edu/) 有撰寫UNIX 介紹的課程 (http://8help.osu.edu/wks/unix_course/index.html)，並提供HTML 或PostScript 兩種格式供人瀏覽。

UNIX 介紹的義大利文翻譯版 (http://www.FreeBSD.org/doc/it_IT.ISO8859-15/books/unix-introduction/index.html) ，同時本文件也是FreeBSD Italian Documentation Project 之一。

· Jpman Project, Japan FreeBSD Users Group (http://www.jp.FreeBSD.org/). FreeBSD User's Reference Manual (http://www.pc.mycom.co.jp/FreeBSD/urm.html) (日文翻譯)。Mainichi Communications Inc. (http://www.pc.mycom.co.jp/), 1998. ISBN4-8399-0088-4 P3800E.

· Edinburgh University (http://www.ed.ac.uk/) 為UNIX 新手所撰寫的Online Guide (http://unixhelp.ed.ac.uk/) 指引說明。

B.3 系統管理指南

- Albitz, Paul and Liu, Cricket. *DNS and BIND*, 4th Ed. O'Reilly & Associates, Inc., 2001. ISBN 1-59600-158-4
- Computer Systems Research Group, UC Berkeley. *4.4BSD System Manager's Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-080-5
- Costales, Brian, et al. *Sendmail*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-222-0
- Frisch, Æleen. *Essential System Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-127-5
- Hunt, Craig. *TCP/IP Network Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-322-7
- Nemeth, Evi. *UNIX System Administration Handbook*. 3rd Ed. Prentice Hall, 2000. ISBN 0-13-020601-6
- Stern, Hal *Managing NFS and NIS* O'Reilly & Associates, Inc., 1991. ISBN 0-937175-75-7
- Jpman Project, Japan FreeBSD Users Group (http://www.jp.FreeBSD.org/). FreeBSD System Administrator's Manual (http://www.pc.mycom.co.jp/FreeBSD/sam.html) (日文翻譯)。Mainichi Communications Inc. (http://www.pc.mycom.co.jp/), 1998. ISBN4-8399-0109-0 P3300E.
- Dreyfus, Emmanuel. Cahiers de l'Admin: BSD (http://www.eyrolles.com/Informatique/Livre/9782212114638/) 2nd Ed. (法文), Eyrolles, 2004. ISBN 2-212-11463-X

B.4 程式設計師指南

- Asente, Paul, Converse, Diana, and Swick, Ralph. *X Window System Toolkit*. Digital Press, 1998. ISBN 1-55558-178-1
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1
- Harbison, Samuel P. and Steele, Guy L. Jr. *C: A Reference Manual*. 4th ed. Prentice Hall, 1995. ISBN 0-13-326224-3
- Kernighan, Brian and Dennis M. Ritchie. *The C Programming Language*. 2nd Ed. PTR Prentice Hall, 1988. ISBN 0-13-110362-8
- Lehey, Greg. *Porting UNIX Software*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7
- Plauger, P. J. *The Standard C Library*. Prentice Hall, 1992. ISBN 0-13-131509-9
- Spinellis, Diomidis. *Code Reading: The Open Source Perspective* (http://www.spinellis.gr/codereading/). Addison-Wesley, 2003. ISBN 0-201-79940-5
- Spinellis, Diomidis. *Code Quality: The Open Source Perspective* (http://www.spinellis.gr/codequality/). Addison-Wesley, 2006. ISBN 0-321-16607-8
- Stevens, W. Richard and Stephen A. Rago. *Advanced Programming in the UNIX Environment*. 2nd Ed. Reading, Mass. : Addison-Wesley, 2005. ISBN 0-201-43307-9
- Stevens, W. Richard. *UNIX Network Programming*. 2nd Ed, PTR Prentice Hall, 1998. ISBN 0-13-490012-X
- Wells, Bill. "Writing Serial Drivers for UNIX". *Dr. Dobb's Journal*. 19(15), December 1994. pp68-71, 97-99.

B.5 深入作業系統

- Andleigh, Prabhat K. *UNIX System Architecture*. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William. “Porting UNIX to the 386”. *Dr. Dobb’s Journal*. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels and John Quarterman *The Design and Implementation of the 4.3BSD UNIX Operating System*. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1
- Leffler, Samuel J., Marshall Kirk McKusick, *The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book*. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. *The Design and Implementation of the 4.4BSD Operating System*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4

  (本書第二章的網路版 (http://www.FreeBSD.org/doc/zh_TW.Big5/books/design-44bsd/book.html) 是FreeBSD 文件計劃的一部份，以及第九章的部分可以在 這邊 (http://www.netapp.com/tech_library/nfsbook.html)找到)
- Marshall Kirk McKusick, George V. Neville-Neil *The Design and Implementation of the FreeBSD Operating System*. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9
- Schimmel, Curt. *Unix Systems for Modern Architectures*. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3
- Vahalia, Uresh. *UNIX Internals -- The New Frontiers*. Prentice Hall, 1996. ISBN 0-13-101908-2
- Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

B.6 資安領域的參考文獻

- Cheswick, William R. and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4
- Garfinkel, Simson and Gene Spafford. *Practical UNIX & Internet Security*. 2nd Ed. O’Reilly & Associates, Inc., 1996. ISBN 1-56592-148-8
- Garfinkel, Simson. *PGP Pretty Good Privacy* O’Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8

B.7 硬體方面的參考文獻

- Anderson, Don and Tom Shanley. *Pentium Processor System Architecture*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5

- Ferraro, Richard F. *Programmer's Guide to the EGA, VGA, and Super VGA Cards*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7
- Intel Corporation 通常會以PDF 格式在developer web site (http://developer.intel.com/) 網站放他們的CPU、晶片組、相關標準的規格書文件。
- Shanley, Tom. *80486 System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. *ISA System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. *PCI System Architecture*. 4th ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gilluwe, Frank. *The Undocumented PC*, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8
- Messmer, Hans-Peter. *The Indispensable PC Hardware Book*, 4th Ed. Reading, Mass: Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4

B.8 UNIX 歷史淵源

- Lion, John *Lion's Commentary on UNIX, 6th Ed. With Source Code*. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric S. *The New Hacker's Dictionary, 3rd edition*. MIT Press, 1996. ISBN 0-262-68092-0. Also known as the Jargon File (http://www.catb.org/~esr/jargon/html/index.html)
- Salus, Peter H. *A quarter century of UNIX*. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5
- Simon Garfinkel, Daniel Weise, Steven Strassmann. *The UNIX-HATERS Handbook*. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. Out of print, but available online (http://research.microsoft.com/~daniel/unix-haters.html).
- Don Libes, Sandy Ressler *Life with UNIX* ——special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7
- *BSD 族譜*：http://www.FreeBSD.org/cgi/cvsweb.cgi/src/share/misc/bsd-family-tree 或FreeBSD 機器內的`/usr/share/misc/bsd-family-tree`。
- *The BSD Release Announcements collection*. 1997. http://www.de.FreeBSD.org/de/ftp/releases/
- *Networked Computer Science Technical Reports Library*. http://www.ncstrl.org/
- *Old BSD releases from the Computer Systems Research group (CSRG)*. http://www.mckusick.com/csrg/: The 4CD set covers all BSD versions from 1BSD to 4.4BSD and 4.4BSD-Lite2 (but not 2.11BSD, unfortunately). The last disk also holds the final sources plus the SCCS files.

B.9 雜誌、期刊

- *The C/C++ Users Journal*. R&D Publications Inc. ISSN 1075-2838
- *Sys Admin ——The Journal for UNIX System Administrators* Miller Freeman, Inc., ISSN 1061-2688
- *freeX ——Das Magazin für Linux - BSD - UNIX* (德文) Computer- und Literaturverlag GmbH, ISSN 1436-7033

Appendix C. 網際網路上的資源

進展飛快的FreeBSD 使得現有的印刷、平面媒體跟不上它的最新進度！反而數位版本的資源，也許有時並不是最好，但通常是唯一一個跟上最新進展的方式。正由於FreeBSD 是來自於許多志工的努力，所以廣大的使用者群也通常扮演著“IT技術支援部門” 的角色。只要善用電子郵件和USENET 新聞群組就可以很快速地聯繫這些社群了。

以下簡介與FreeBSD 社群搭上線的主要方式。若你還知道其他這裡沒有列出的資源，請告知FreeBSD documentation project 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc)，以便我們更新。

C.1 郵遞論壇(Mailing Lists)

雖然，大部份的FreeBSD 開發人員都有看USENET 新聞群組，我們也不能保證我們永遠可以及時得知您的問題。尤其是，若您只在`comp.unix.bsd.freebsd.*` 其中群組內發問的話。建議您將問題發到適當的郵遞論壇(mailing list)上，不但可以同時讓FreeBSD 開發者和其他看倌們知道，通常也可以得到一個較好的(最起碼會比較快)的回應。

本文後面介紹的是各式不同的郵遞論壇、討論規則(charter)。*在訂閱任何list 之前，請先閱讀討論規則。* 訂閱這些list 的人們現在每天都會收到數百封FreeBSD 相關信件。而討論規則的制訂，則有助於提升彼此討論品質。否則，這些討論FreeBSD 的list 終將陷入溝通不良，而失去其原本美意。

對於該在哪個list 發問覺得很疑惑的時候，就來看看《如何在FreeBSD-questions mailing list 上得到正解》(http://www.FreeBSD.org/doc/zh_TW.Big5/articles/freebsd-questions) 一文吧

在發表問題、回覆到list 之前，請先讀過 FreeBSD Mailing Lists 常見問答集(FAQ) (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/mailing-list-faq) 一文以學會如何善用mailing list ，才能避免像是經常重複的月經文、戰文之類的事情發生。

全部的mailing lists 討論記錄都可以在 FreeBSD WWW主機 (http://www.FreeBSD.org/search/index.html) 上找到。它提供了很棒的關鍵字搜尋功能，可讓您找到常用問答集(FAQ) 。而在mailing lists 上發問之前，也請先搜尋是否已有解答。

C.1.1 List 一覽表

一般論壇： 以下的list 都是一般性質，而且大家可以自由地參與，我們也鼓勵大家訂閱：

List 名稱	目的
cvs-all (http://lists.FreeBSD.org/mailman/listinfo/cvs-all)	所有的FreeBSD 提交(commit)記錄
freebsd-advocacy (http://lists.FreeBSD.org/mailman/listinfo/freebsd-advocacy)	FreeBSD 惡魔福音電台
freebsd-announce (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce)	公布重要事件、計劃里程碑

List 名稱	目的
freebsd-arch (http://lists.FreeBSD.org/mailman/listinfo/freebsd-arch)	架構研發討論
freebsd-bugbusters (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugbusters)	FreeBSD 問題回報(PR)資料庫的維護議題與工具
freebsd-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugs)	Bug 回報
freebsd-chat (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat)	非技術交流的FreeBSD 社群聊天區
freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current)	討論FreeBSD-CURRENT 版本的FreeBSD
freebsd-isp (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp)	FreeBSD 的ISP 業者技術交流區
freebsd-jobs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-jobs)	FreeBSD 人力銀行
freebsd-policy (http://lists.FreeBSD.org/mailman/listinfo/freebsd-policy)	FreeBSD Core team 的policy 方針討論區。這裡文章不多，且只限core team 才可發言。
freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions)	使用問題及技術支援
freebsd-security-notifications (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications)	安全漏洞通知
freebsd-stable (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable)	討論FreeBSD-STABLE 版本的FreeBSD
freebsd-test (http://lists.FreeBSD.org/mailman/listinfo/freebsd-test)	論壇新手發表區，這裡可以讓你小試身手

技術論壇： 以下list 主是要以探討技術問題為主，在加入訂閱及討論之前，請務必仔細閱讀每個list 的版規(charter)，因為它們的討論內容都有很嚴謹的限制。

List 名稱	目的
freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi)	ACPI 以及電力管理議題

List 名稱	**目的**
freebsd-afs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-afs)	移植AFS 到FreeBSD
freebsd-aic7xxx (http://lists.FreeBSD.org/mailman/listinfo/aic7xxx)	研發Adaptec AIC 7xxx 的驅動程式
freebsd-alpha (http://lists.FreeBSD.org/mailman/listinfo/freebsd-alpha)	移植FreeBSD 到Alpha 系統架構
freebsd-amd64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-amd64)	移植FreeBSD 到AMD64 系統架構
freebsd-apache (http://lists.FreeBSD.org/mailman/listinfo/freebsd-apache)	探討**Apache** 相關的ports 議題
freebsd-arm (http://lists.FreeBSD.org/mailman/listinfo/freebsd-arm)	移植FreeBSD 到ARM® CPU 架構
freebsd-atm (http://lists.FreeBSD.org/mailman/listinfo/freebsd-atm)	在FreeBSD 上使用ATM 網路
freebsd-audit (http://lists.FreeBSD.org/mailman/listinfo/freebsd-audit)	Source code 的稽核(audit)計劃
freebsd-binup (http://lists.FreeBSD.org/mailman/listinfo/freebsd-binup)	研發binary 的升級方式
freebsd-bluetooth (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bluetooth)	在FreeBSD 中使用藍芽(Bluetooth)技術
freebsd-cluster (http://lists.FreeBSD.org/mailman/listinfo/freebsd-cluster)	把FreeBSD 在叢集架構環境(clustered environment)的運用
freebsd-cvsweb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-cvsweb)	CVSweb 的維護
freebsd-database (http://lists.FreeBSD.org/mailman/listinfo/freebsd-database)	討論各式資料庫在FreeBSD 的研發、運用
freebsd-doc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc)	撰寫FreeBSD 相關文件

List 名稱	目的
freebsd-drivers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-drivers)	撰寫FreeBSD 用的驅動程式
freebsd-eclipse (http://lists.FreeBSD.org/mailman/listinfo/freebsd-eclipse)	FreeBSD 的Eclipse IDE 工具愛用者交流
freebsd-emulation (http://lists.FreeBSD.org/mailman/listinfo/freebsd-emulation)	在FreeBSD 上模擬其他系統，如：Linux、MS-DOS、Windows
freebsd-firewire (http://lists.FreeBSD.org/mailman/listinfo/freebsd-firewire)	FreeBSD 的FireWire® (iLink, IEEE 1394) 方面技術交流
freebsd-fs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-fs)	檔案系統的探討、研發
freebsd-geom (http://lists.FreeBSD.org/mailman/listinfo/freebsd-geom)	GEOM 議題的探討、研發
freebsd-gnome (http://lists.FreeBSD.org/mailman/listinfo/freebsd-gnome)	移植**GNOME** 及**GNOME** 相關應用軟體
freebsd-hackers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers)	一般技術議題的探討
freebsd-hardware (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hardware)	FreeBSD 上的各式硬體問題交流
freebsd-i18n (http://lists.FreeBSD.org/mailman/listinfo/freebsd-i18n)	FreeBSD 的多語化(Internationalization)
freebsd-ia32 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia32)	FreeBSD 在IA-32 (Intel x86) 平台上的使用探討
freebsd-ia64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia64)	移植FreeBSD 到Intel 的未來IA64 平台架構
freebsd-ipfw (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ipfw)	對ipfw(IP firewall) 的技術探討
freebsd-isdn (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn)	ISDN 研發

List 名稱	目的
freebsd-java (http://lists.FreeBSD.org/mailman/listinfo/freebsd-java)	Java 程式開發者以及移植JDKs 到FreeBSD 上
freebsd-kde (https://mail.kde.org/mailman/listinfo/kde-freebsd)	移植**KDE** 以及**KDE** 相關應用程式
freebsd-lfs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-lfs)	移植LFS 到FreeBSD
freebsd-libh (http://lists.FreeBSD.org/mailman/listinfo/freebsd-libh)	新世代的安裝、打包套件機制
freebsd-mips (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips)	移植FreeBSD 到MIPS®
freebsd-mobile (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile)	關於各類mobile computing(比如：筆記型電腦)的研討
freebsd-mozilla (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mozilla)	把**Mozilla** 軟體移植到FreeBSD
freebsd-multimedia (http://lists.FreeBSD.org/mailman/listinfo/freebsd-multimedia)	各式影音運用、軟體
freebsd-new-bus (http://lists.FreeBSD.org/mailman/listinfo/freebsd-new-bus)	各式bus 架構的技術探討
freebsd-net (http://lists.FreeBSD.org/mailman/listinfo/freebsd-net)	網路運用探討與TCP/IP source code
freebsd-openoffice (http://lists.FreeBSD.org/mailman/listinfo/freebsd-openoffice)	移植**OpenOffice.org** 及**StarOffice** 到FreeBSD
freebsd-performance (http://lists.FreeBSD.org/mailman/listinfo/freebsd-performance)	在高效能/負荷環境下的效能調校(tuning)議題
freebsd-perl (http://lists.FreeBSD.org/mailman/listinfo/freebsd-perl)	探討Perl 相關ports 的維護
freebsd-pf (http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf)	pf(packet filter) 防火牆機制的探討

List 名稱	目的
freebsd-platforms (http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms)	討論著重於移植port 到非Intel 架構平台的議題
freebsd-ports (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports)	關於Ports Collection 的運用、探討
freebsd-ports-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs)	ports 相關的bugs/PRs
freebsd-ppc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ppc)	移植FreeBSD 到PowerPC® 系統架構
freebsd-proliant (http://lists.FreeBSD.org/mailman/listinfo/freebsd-proliant)	FreeBSD 在HP ProLiant 主機平台的使用交流
freebsd-python (http://lists.FreeBSD.org/mailman/listinfo/freebsd-python)	Python 在FreeBSD 上使用的各式議題。
freebsd-qa (http://lists.FreeBSD.org/mailman/listinfo/freebsd-qa)	QA(Quality Assurance)討論，通常會決定是否已經到達可以release 的程度
freebsd-rc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-rc)	`rc.d` 機制的討論、研發
freebsd-realtime (http://lists.FreeBSD.org/mailman/listinfo/freebsd-realtime)	FreeBSD 上realtime extensions 的研發
freebsd-scsi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-scsi)	SCSI 方面議題
freebsd-security (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security)	FreeBSD 安全漏洞議題
freebsd-small (http://lists.FreeBSD.org/mailman/listinfo/freebsd-small)	在嵌入式硬體環境的應用、探討
freebsd-smp (http://lists.FreeBSD.org/mailman/listinfo/freebsd-smp)	CPU 對稱多工處理(SMP, [A]Symmetric Multiprocessing)的應用、研討
freebsd-sparc64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-sparc64)	移植FreeBSD 到Sparc® 平台架構

List 名稱	目的
freebsd-standards (http://lists.FreeBSD.org/mailman/listinfo/freebsd-standards)	FreeBSD 與C99 及POSIX標準的相容議題
freebsd-threads (http://lists.FreeBSD.org/mailman/listinfo/freebsd-threads)	FreeBSD 上的Threading 運用、探討
freebsd-testing (http://lists.FreeBSD.org/mailman/listinfo/freebsd-testing)	FreeBSD 的效能與穩定性測試
freebsd-tokenring (http://lists.FreeBSD.org/mailman/listinfo/freebsd-tokenring)	FreeBSD 的Token Ring 支援的應用、探討
freebsd-usb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-usb)	FreeBSD 的USB 支援的應用、探討
freebsd-vuxml (http://lists.FreeBSD.org/mailman/listinfo/freebsd-vuxml)	VuXML 漏洞通報架構的探討
freebsd-x11 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-x11)	X11 在FreeBSD 的運用

有訂閱限制的論壇： 以下的lists 是針對一些特定要求的讀者而設，而且並不適合當成是一般的公開討論區。您最好先在某些技術討論區參與一段時間的討論後，再選擇訂閱這些有限制的論壇。因為如此一來，您可以了解到在這些討論區發言所須的禮儀。

List 名稱	目的
freebsd-hubs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hubs)	映射站台(mirror)的交流討論區
freebsd-user-groups (http://lists.FreeBSD.org/mailman/listinfo/freebsd-user-groups)	社群間的協調
freebsd-vendors (http://lists.FreeBSD.org/mailman/listinfo/freebsd-vendors)	Vendors pre-release coordination
freebsd-www (http://lists.FreeBSD.org/mailman/listinfo/freebsd-www)	www.FreeBSD.org (http://www.FreeBSD.org/index.html) 的管理維護

論壇的摘要版： 上述的各lists 都有摘要版(digest)，在訂閱list 之後，就可以先以自己帳號登入，然後個人訂閱選項那邊改為摘要版即可。

CVS lists: 以下的lists 是提供給想要看各個tree 的提交(commit)紀錄，請注意：他們是*唯讀(Read-Only)* 性質的list ，而且不能寄信給這。

List 名稱	Source 區域	本區簡介(source for)
cvs-all (http://lists.FreeBSD.org/mailman/listinfo/cvs-all)	`/usr/(CVSROOT|doc|ports|projects|src)`	全部的變更記錄(包括各類CVS commit)
cvs-doc (http://lists.FreeBSD.org/mailman/listinfo/cvs-doc)	`/usr/(doc|www)`	doc 及www trees 的所有變更紀錄
cvs-ports (http://lists.FreeBSD.org/mailman/listinfo/cvs-ports)	`/usr/ports`	ports tree 的所有變更紀錄
cvs-projects (http://lists.FreeBSD.org/mailman/listinfo/cvs-projects)	`/usr/projects`	projects tree 的所有變更紀錄
cvs-src (http://lists.FreeBSD.org/mailman/listinfo/cvs-src)	`/usr/src`	src tree 的所有變更紀錄

C.1.2 要如何訂閱

要訂閱list 的話，請以滑鼠按下上述list 名稱，或是到http://lists.FreeBSD.org/mailman/listinfo 然後即可挑選有興趣的list 來訂閱了，該網頁會指示你如何進行訂閱的步驟。

To actually post to a given list you simply send mail to <*listname*`@FreeBSD.org`>. It will then be redistributed to mailing list members world-wide.

To unsubscribe yourself from a list, click on the URL found at the bottom of every email received from the list. It is also possible to send an email to <*listname*`-unsubscribe@FreeBSD.org`> to unsubscribe yourself.

Again, we would like to request that you keep discussion in the technical mailing lists on a technical track. If you are only interested in important announcements then it is suggested that you join the FreeBSD announcements 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce), which is intended only for infrequent traffic.

C.1.3 List Charters

All FreeBSD mailing lists have certain basic rules which must be adhered to by anyone using them. Failure to comply with these guidelines will result in two (2) written warnings from the FreeBSD Postmaster <`postmaster@FreeBSD.org`>, after which, on a third offense, the poster will removed from all FreeBSD mailing

lists and filtered from further posting to them. We regret that such rules and measures are necessary at all, but today's Internet is a pretty harsh environment, it would seem, and many fail to appreciate just how fragile some of its mechanisms are.

Rules of the road:

- The topic of any posting should adhere to the basic charter of the list it is posted to, e.g. if the list is about technical issues then your posting should contain technical discussion. Ongoing irrelevant chatter or flaming only detracts from the value of the mailing list for everyone on it and will not be tolerated. For free-form discussion on no particular topic, the FreeBSD chat 郵遞論壇 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat) is freely available and should be used instead.
- No posting should be made to more than 2 mailing lists, and only to 2 when a clear and obvious need to post to both lists exists. For most lists, there is already a great deal of subscriber overlap and except for the most esoteric mixes (say "-stable & -scsi"), there really is no reason to post to more than one list at a time. If a message is sent to you in such a way that multiple mailing lists appear on the `Cc` line then the `Cc` line should also be trimmed before sending it out again. *You are still responsible for your own cross-postings, no matter who the originator might have been.*
- Personal attacks and profanity (in the context of an argument) are not allowed, and that includes users and developers alike. Gross breaches of netiquette, like excerpting or reposting private mail when permission to do so was not and would not be forthcoming, are frowned upon but not specifically enforced. *However*, there are also very few cases where such content would fit within the charter of a list and it would therefore probably rate a warning (or ban) on that basis alone.
- Advertising of non-FreeBSD related products or services is strictly prohibited and will result in an immediate ban if it is clear that the offender is advertising by spam.

Individual list charters:

freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi)

ACPI and power management development

freebsd-afs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-afs)

Andrew File System

This list is for discussion on porting and using AFS from CMU/Transarc

freebsd-announce (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce)

Important events / milestones

This is the mailing list for people interested only in occasional announcements of significant FreeBSD events. This includes announcements about snapshots and other releases. It contains announcements of new FreeBSD capabilities. It may contain calls for volunteers etc. This is a low volume, strictly moderated mailing list.

freebsd-arch (http://lists.FreeBSD.org/mailman/listinfo/freebsd-arch)

Architecture and design discussions

This list is for discussion of the FreeBSD architecture. Messages will mostly be kept strictly technical in nature. Examples of suitable topics are:

- How to re-vamp the build system to have several customized builds running at the same time.

- What needs to be fixed with VFS to make Heidemann layers work.
- How do we change the device driver interface to be able to use the same drivers cleanly on many buses and architectures.
- How to write a network driver.

freebsd-audit (http://lists.FreeBSD.org/mailman/listinfo/freebsd-audit)

Source code audit project

This is the mailing list for the FreeBSD source code audit project. Although this was originally intended for security-related changes, its charter has been expanded to review any code changes.

This list is very heavy on patches, and is probably of no interest to the average FreeBSD user. Security discussions not related to a particular code change are held on freebsd-security. Conversely, all developers are encouraged to send their patches here for review, especially if they touch a part of the system where a bug may adversely affect the integrity of the system.

freebsd-binup (http://lists.FreeBSD.org/mailman/listinfo/freebsd-binup)

FreeBSD Binary Update Project

This list exists to provide discussion for the binary update system, or **binup**. Design issues, implementation details, patches, bug reports, status reports, feature requests, commit logs, and all other things related to **binup** are fair game.

freebsd-bluetooth (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bluetooth)

Bluetooth in FreeBSD

This is the forum where FreeBSD's Bluetooth users congregate. Design issues, implementation details, patches, bug reports, status reports, feature requests, and all matters related to Bluetooth are fair game.

freebsd-bugbusters (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugbusters)

Coordination of the Problem Report handling effort

The purpose of this list is to serve as a coordination and discussion forum for the Bugmeister, his Bugbusters, and any other parties who have a genuine interest in the PR database. This list is not for discussions about specific bugs, patches or PRs.

freebsd-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugs)

Bug reports

This is the mailing list for reporting bugs in FreeBSD. Whenever possible, bugs should be submitted using the send-pr(1) command or the WEB interface (http://www.FreeBSD.org/send-pr.html) to it.

freebsd-chat (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat)

Non technical items related to the FreeBSD community

This list contains the overflow from the other lists about non-technical, social information. It includes discussion about whether Jordan looks like a toon ferret or not, whether or not to type in capitals, who is drinking too much coffee, where the best beer is brewed, who is brewing beer in their basement, and so on.

Occasional announcements of important events (such as upcoming parties, weddings, births, new jobs, etc) can be made to the technical lists, but the follow ups should be directed to this -chat list.

freebsd-core

FreeBSD core team

This is an internal mailing list for use by the core members. Messages can be sent to it when a serious FreeBSD-related matter requires arbitration or high-level scrutiny.

freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current)

Discussions about the use of FreeBSD-CURRENT

This is the mailing list for users of FreeBSD-CURRENT. It includes warnings about new features coming out in -CURRENT that will affect the users, and instructions on steps that must be taken to remain -CURRENT. Anyone running "CURRENT" must subscribe to this list. This is a technical mailing list for which strictly technical content is expected.

freebsd-cvsweb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-cvsweb)

FreeBSD CVSweb Project

Technical discussions about use, development and maintenance of FreeBSD-CVSweb.

freebsd-doc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc)

Documentation project

This mailing list is for the discussion of issues and projects related to the creation of documentation for FreeBSD. The members of this mailing list are collectively referred to as "The FreeBSD Documentation Project". It is an open list; feel free to join and contribute!

freebsd-drivers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-drivers)

Writing device drivers for FreeBSD

This is a forum for technical discussions related to device drivers on FreeBSD. It is primarily a place for device driver writers to ask questions about how to write device drivers using the APIs in the FreeBSD kernel.

freebsd-eclipse (http://lists.FreeBSD.org/mailman/listinfo/freebsd-eclipse)

FreeBSD users of Eclipse IDE, tools, rich client applications and ports.

The intention of this list is to provide mutual support for everything to do with choosing, installing, using, developing and maintaining the Eclipse IDE, tools, rich client applications on the FreeBSD platform and assisting with the porting of Eclipse IDE and plugins to the FreeBSD environment.

The intention is also to facilitate exchange of information between the Eclipse community and the FreeBSD community to the mutual benefit of both.

Although this list is focused primarily on the needs of Eclipse users it will also provide a forum for those who would like to develop FreeBSD specific applications using the Eclipse framework.

freebsd-emulation (http://lists.FreeBSD.org/mailman/listinfo/freebsd-emulation)

Emulation of other systems such as Linux/MS-DOS/Windows

This is a forum for technical discussions related to running programs written for other operating systems on FreeBSD.

freebsd-firewire (http://lists.FreeBSD.org/mailman/listinfo/freebsd-firewire)

FireWire (iLink, IEEE 1394)

This is a mailing list for discussion of the design and implementation of a FireWire (aka IEEE 1394 aka iLink) subsystem for FreeBSD. Relevant topics specifically include the standards, bus devices and their protocols, adapter boards/cards/chips sets, and the architecture and implementation of code for their proper support.

freebsd-fs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-fs)

File systems

Discussions concerning FreeBSD file systems. This is a technical mailing list for which strictly technical content is expected.

freebsd-geom (http://lists.FreeBSD.org/mailman/listinfo/freebsd-geom)

GEOM

Discussions specific to GEOM and related implementations. This is a technical mailing list for which strictly technical content is expected.

freebsd-gnome (http://lists.FreeBSD.org/mailman/listinfo/freebsd-gnome)

GNOME

Discussions concerning The **GNOME** Desktop Environment for FreeBSD systems. This is a technical mailing list for which strictly technical content is expected.

freebsd-ipfw (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ipfw)

IP Firewall

This is the forum for technical discussions concerning the redesign of the IP firewall code in FreeBSD. This is a technical mailing list for which strictly technical content is expected.

freebsd-ia64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia64)

Porting FreeBSD to IA64

This is a technical mailing list for individuals actively working on porting FreeBSD to the IA-64 platform from Intel, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.

freebsd-isdn (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn)

ISDN Communications

This is the mailing list for people discussing the development of ISDN support for FreeBSD.

freebsd-java (http://lists.FreeBSD.org/mailman/listinfo/freebsd-java)

Java Development

This is the mailing list for people discussing the development of significant Java applications for FreeBSD and the porting and maintenance of JDKs.

freebsd-jobs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-jobs)

Jobs offered and sought

This is a forum for posting employment notices and resumes specifically related to FreeBSD, e.g. if you are seeking FreeBSD-related employment or have a job involving FreeBSD to advertise then this is the right place. This is *not* a mailing list for general employment issues since adequate forums for that already exist elsewhere.

Note that this list, like other `FreeBSD.org` mailing lists, is distributed worldwide. Thus, you need to be clear about location and the extent to which telecommuting or assistance with relocation is available.

Email should use open formats only ——preferably plain text, but basic Portable Document Format (PDF), HTML, and a few others are acceptable to many readers. Closed formats such as Microsoft Word (`.doc`) will be rejected by the mailing list server.

freebsd-kde (https://mail.kde.org/mailman/listinfo/kde-freebsd)

KDE

Discussions concerning **KDE** on FreeBSD systems. This is a technical mailing list for which strictly technical content is expected.

freebsd-hackers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers)

Technical discussions

This is a forum for technical discussions related to FreeBSD. This is the primary technical mailing list. It is for individuals actively working on FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome. This is a technical mailing list for which strictly technical content is expected.

freebsd-hardware (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hardware)

General discussion of FreeBSD hardware

General discussion about the types of hardware that FreeBSD runs on, various problems and suggestions concerning what to buy or avoid.

freebsd-hubs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hubs)

Mirror sites

Announcements and discussion for people who run FreeBSD mirror sites.

freebsd-isp (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp)

Issues for Internet Service Providers

This mailing list is for discussing topics relevant to Internet Service Providers (ISPs) using FreeBSD. This is a technical mailing list for which strictly technical content is expected.

freebsd-openoffice (http://lists.FreeBSD.org/mailman/listinfo/freebsd-openoffice)

OpenOffice.org

Discussions concerning the porting and maintenance of **OpenOffice.org** and **StarOffice**.

freebsd-performance (http://lists.FreeBSD.org/mailman/listinfo/freebsd-performance)

Discussions about tuning or speeding up FreeBSD

This mailing list exists to provide a place for hackers, administrators, and/or concerned parties to discuss performance related topics pertaining to FreeBSD. Acceptable topics includes talking about FreeBSD installations that are either under high load, are experiencing performance problems, or are pushing the limits of FreeBSD. Concerned parties that are willing to work toward improving the performance of FreeBSD are highly encouraged to subscribe to this list. This is a highly technical list ideally suited for experienced FreeBSD users, hackers, or administrators interested in keeping FreeBSD fast, robust, and scalable. This list is not a question-and-answer list that replaces reading through documentation, but it is a place to make contributions or inquire about unanswered performance related topics.

freebsd-pf (http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf)

Discussion and questions about the packet filter firewall system

Discussion concerning the packet filter (pf) firewall system in terms of FreeBSD. Technical discussion and user questions are both welcome. This list is also a place to discuss the ALTQ QoS framework.

freebsd-platforms (http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms)

Porting to Non Intel platforms

Cross-platform FreeBSD issues, general discussion and proposals for non Intel FreeBSD ports. This is a technical mailing list for which strictly technical content is expected.

freebsd-policy (http://lists.FreeBSD.org/mailman/listinfo/freebsd-policy)

Core team policy decisions

This is a low volume, read-only mailing list for FreeBSD Core Team Policy decisions.

freebsd-ports (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports)

Discussion of "ports"

Discussions concerning FreeBSD's "ports collection" (`/usr/ports`), ports infrastructure, and general ports coordination efforts. This is a technical mailing list for which strictly technical content is expected.

freebsd-ports-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs)

Discussion of "ports" bugs

Discussions concerning problem reports for FreeBSD's "ports collection" (`/usr/ports`), proposed ports, or modifications to ports. This is a technical mailing list for which strictly technical content is expected.

freebsd-proliant (http://lists.FreeBSD.org/mailman/listinfo/freebsd-proliant)

Technical discussion of FreeBSD on HP ProLiant server platforms

This mailing list is to be used for the technical discussion of the usage of FreeBSD on HP ProLiant servers, including the discussion of ProLiant-specific drivers, management software, configuration tools, and BIOS updates. As such, this is the primary place to discuss the hpasmd, hpasmcli, and hpacucli modules.

freebsd-python (http://lists.FreeBSD.org/mailman/listinfo/freebsd-python)

Python on FreeBSD

This is a list for discussions related to improving Python-support on FreeBSD. This is a technical mailing list. It is for individuals working on porting Python, its 3rd party modules and **Zope** stuff to FreeBSD. Individuals interested in following the technical discussion are also welcome.

freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions)

User questions

This is the mailing list for questions about FreeBSD. You should not send "how to" questions to the technical lists unless you consider the question to be pretty technical.

freebsd-scsi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-scsi)

SCSI subsystem

This is the mailing list for people working on the SCSI subsystem for FreeBSD. This is a technical mailing list for which strictly technical content is expected.

freebsd-security (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security)

Security issues

FreeBSD computer security issues (DES, Kerberos, known security holes and fixes, etc). This is a technical mailing list for which strictly technical discussion is expected. Note that this is not a question-and-answer list, but that contributions (BOTH question AND answer) to the FAQ are welcome.

freebsd-security-notifications (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications)

Security Notifications

Notifications of FreeBSD security problems and fixes. This is not a discussion list. The discussion list is FreeBSD-security.

freebsd-small (http://lists.FreeBSD.org/mailman/listinfo/freebsd-small)

Using FreeBSD in embedded applications

This list discusses topics related to unusually small and embedded FreeBSD installations. This is a technical mailing list for which strictly technical content is expected.

freebsd-stable (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable)

Discussions about the use of FreeBSD-STABLE

This is the mailing list for users of FreeBSD-STABLE. It includes warnings about new features coming out in -STABLE that will affect the users, and instructions on steps that must be taken to remain -STABLE. Anyone running "STABLE" should subscribe to this list. This is a technical mailing list for which strictly technical content is expected.

freebsd-standards (http://lists.FreeBSD.org/mailman/listinfo/freebsd-standards)

C99 & POSIX Conformance

This is a forum for technical discussions related to FreeBSD Conformance to the C99 and the POSIX standards.

freebsd-usb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-usb)

Discussing FreeBSD support for USB

This is a mailing list for technical discussions related to FreeBSD support for USB.

freebsd-user-groups (http://lists.FreeBSD.org/mailman/listinfo/freebsd-user-groups)

User Group Coordination List

This is the mailing list for the coordinators from each of the local area Users Groups to discuss matters with each other and a designated individual from the Core Team. This mail list should be limited to meeting synopsis and coordination of projects that span User Groups.

freebsd-vendors (http://lists.FreeBSD.org/mailman/listinfo/freebsd-vendors)

Vendors

Coordination discussions between The FreeBSD Project and Vendors of software and hardware for FreeBSD.

C.1.4 Filtering on the Mailing Lists

The FreeBSD mailing lists are filtered in multiple ways to avoid the distribution of spam, viruses, and other unwanted emails. The filtering actions described in this section do not include all those used to protect the mailing lists.

Only certain types of attachments are allowed on the mailing lists. All attachments with a MIME content type not found in the list below will be stripped before an email is distributed on the mailing lists.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch

Note: Some of the mailing lists might allow attachments of other MIME content types, but the above list should be applicable for most of the mailing lists.

If an email contains both an HTML and a plain text version, the HTML version will be removed. If an email contains only an HTML version, it will be converted to plain text.

C.2 Usenet Newsgroups

In addition to two FreeBSD specific newsgroups, there are many others in which FreeBSD is discussed or are otherwise relevant to FreeBSD users. Keyword searchable archives

(http://minnie.tuhs.org/BSD-info/bsdnews_search.html) are available for some of these newsgroups from courtesy of Warren Toomey <wkt@cs.adfa.edu.au>.

C.2.1 BSD Specific Newsgroups

- comp.unix.bsd.freebsd.announce (news:comp.unix.bsd.freebsd.announce)
- comp.unix.bsd.freebsd.misc (news:comp.unix.bsd.freebsd.misc)
- de.comp.os.unix.bsd (news:de.comp.os.unix.bsd) (German)
- fr.comp.os.bsd (news:fr.comp.os.bsd) (French)
- it.comp.os.freebsd (news:it.comp.os.freebsd) (Italian)

C.2.2 Other UNIX Newsgroups of Interest

- comp.unix (news:comp.unix)
- comp.unix.questions (news:comp.unix.questions)
- comp.unix.admin (news:comp.unix.admin)
- comp.unix.programmer (news:comp.unix.programmer)
- comp.unix.shell (news:comp.unix.shell)
- comp.unix.user-friendly (news:comp.unix.user-friendly)
- comp.security.unix (news:comp.security.unix)
- comp.sources.unix (news:comp.sources.unix)
- comp.unix.advocacy (news:comp.unix.advocacy)
- comp.unix.misc (news:comp.unix.misc)
- comp.bugs.4bsd (news:comp.bugs.4bsd)
- comp.bugs.4bsd.ucb-fixes (news:comp.bugs.4bsd.ucb-fixes)
- comp.unix.bsd (news:comp.unix.bsd)

C.2.3 X Window System

- comp.windows.x.i386unix (news:comp.windows.x.i386unix)
- comp.windows.x (news:comp.windows.x)
- comp.windows.x.apps (news:comp.windows.x.apps)
- comp.windows.x.announce (news:comp.windows.x.announce)
- comp.windows.x.intrinsics (news:comp.windows.x.intrinsics)
- comp.windows.x.motif (news:comp.windows.x.motif)

- comp.windows.x.pex (news:comp.windows.x.pex)
- comp.emulators.ms-windows.wine (news:comp.emulators.ms-windows.wine)

C.3 World Wide Web Servers

Central Servers, Argentina, Armenia, Australia, Austria, Belgium, Brazil, Bulgaria, Canada, China, Costa Rica, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hong Kong, Hungary, Iceland, Indonesia, Ireland, Italy, Japan, Korea, Kuwait, Kyrgyzstan, Latvia, Lithuania, Netherlands, New Zealand, Norway, Philippines, Poland, Portugal, Romania, Russia, San Marino, Singapore, Slovak Republic, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Thailand, Turkey, Ukraine, United Kingdom, USA.

(as of 2009/05/21 21:02:58 UTC)

.

Central Servers

- http://www.FreeBSD.org/

.

Argentina

- http://www.ar.FreeBSD.org/

.

Armenia

- http://www1.am.FreeBSD.org/ (IPv6)

.

Australia

- http://www.au.FreeBSD.org/
- http://www2.au.FreeBSD.org/

.

Austria

- http://www.at.FreeBSD.org/
- http://www2.at.FreeBSD.org/ (IPv6)

.

Belgium

- http://freebsd.unixtech.be/

.

Brazil

- http://www.br.FreeBSD.org/
- http://www2.br.FreeBSD.org/www.freebsd.org/
- http://www3.br.FreeBSD.org/

.

Bulgaria

- http://www.bg.FreeBSD.org/
- http://www2.bg.FreeBSD.org/

.

Canada

- http://www.ca.FreeBSD.org/
- http://www2.ca.FreeBSD.org/

.

China

- http://www.cn.FreeBSD.org/

.

Costa Rica

- http://www1.cr.FreeBSD.org/

.

Czech Republic

- http://www.cz.FreeBSD.org/

.

Denmark

- http://www.dk.FreeBSD.org/ (IPv6)
- http://www3.dk.FreeBSD.org/

- Estonia
  - http://www.ee.FreeBSD.org/
- Finland
  - http://www.fi.FreeBSD.org/
  - http://www2.fi.FreeBSD.org/
- France
  - http://www.fr.FreeBSD.org/
  - http://www1.fr.FreeBSD.org/
- Germany
  - http://www.de.FreeBSD.org/
- Greece
  - http://www.gr.FreeBSD.org/
- Hong Kong
  - http://www.hk.FreeBSD.org/
- Hungary
  - http://www.hu.FreeBSD.org/
  - http://www2.hu.FreeBSD.org/
- Iceland
  - http://www.is.FreeBSD.org/

- Indonesia
  - http://www.id.FreeBSD.org/

- Ireland
  - http://www.ie.FreeBSD.org/
  - http://www2.ie.FreeBSD.org/

- Italy
  - http://www.it.FreeBSD.org/
  - http://www.gufi.org/mirrors/www.freebsd.org/data/

- Japan
  - http://www.jp.FreeBSD.org/www.FreeBSD.org/ (IPv6)

- Korea
  - http://www.kr.FreeBSD.org/
  - http://www2.kr.FreeBSD.org/

- Kuwait
  - http://www.kw.FreeBSD.org/

- Kyrgyzstan
  - http://www.kg.FreeBSD.org/

- Latvia
  - http://www.lv.FreeBSD.org/
  - http://www2.lv.FreeBSD.org/

.

Lithuania

- http://www.lt.FreeBSD.org/

.

Netherlands

- http://www.nl.FreeBSD.org/
- http://www2.nl.FreeBSD.org/

.

New Zealand

- http://www.nz.FreeBSD.org/

.

Norway

- http://www.no.FreeBSD.org/

.

Philippines

- http://www.FreeBSD.org.ph/

.

Poland

- http://www.pl.FreeBSD.org/
- http://www2.pl.FreeBSD.org/

.

Portugal

- http://www.pt.FreeBSD.org/
- http://www1.pt.FreeBSD.org/
- http://www4.pt.FreeBSD.org/
- http://www5.pt.FreeBSD.org/

.

Romania

- http://www.ro.FreeBSD.org/
- http://www1.ro.FreeBSD.org/
- http://www2.ro.FreeBSD.org/
- http://www3.ro.FreeBSD.org/

.

Russia

- http://www.ru.FreeBSD.org/
- http://www2.ru.FreeBSD.org/
- http://www3.ru.FreeBSD.org/
- http://www4.ru.FreeBSD.org/
- http://www5.ru.FreeBSD.org/

.

San Marino

- http://www.sm.FreeBSD.org/

.

Singapore

- http://www2.sg.FreeBSD.org/

.

Slovak Republic

- http://www.sk.FreeBSD.org/

.

Slovenia

- http://www.si.FreeBSD.org/
- http://www2.si.FreeBSD.org/

.

South Africa

- http://www.za.FreeBSD.org/
- http://www2.za.FreeBSD.org/

.

Spain

- http://www.es.FreeBSD.org/
- http://www2.es.FreeBSD.org/
- http://www3.es.FreeBSD.org/

.

Sweden

- http://www.se.FreeBSD.org/
- http://www2.se.FreeBSD.org/

.

Switzerland

- http://www.ch.FreeBSD.org/
- http://www2.ch.FreeBSD.org/

.

Taiwan

- http://www.tw.FreeBSD.org/ (IPv6)
- http://www2.tw.FreeBSD.org/
- http://www3.tw.FreeBSD.org/
- http://www4.tw.FreeBSD.org/
- http://www5.tw.FreeBSD.org/ (IPv6)
- http://www6.tw.FreeBSD.org/
- http://www7.tw.FreeBSD.org/

.

Thailand

- http://www.th.FreeBSD.org/

.

Turkey

- http://www.tr.FreeBSD.org/
- http://www2.tr.FreeBSD.org/

- http://www3.tr.FreeBSD.org/ (IPv6)

.

Ukraine

- http://www.ua.FreeBSD.org/
- http://www2.ua.FreeBSD.org/
- http://www5.ua.FreeBSD.org/
- http://www4.ua.FreeBSD.org/

.

United Kingdom

- http://www1.uk.FreeBSD.org/
- http://www3.uk.FreeBSD.org/

.

USA

- http://www2.us.FreeBSD.org/
- http://www4.us.FreeBSD.org/ (IPv6)
- http://www5.us.FreeBSD.org/ (IPv6)

C.4 Email Addresses

The following user groups provide FreeBSD related email addresses for their members. The listed administrator reserves the right to revoke the address if it is abused in any way.

Domain	Facilities	User Group	Administrator
ukug.uk.FreeBSD.org	Forwarding only	<freebsd-users@uk.FreeBSD.org>	Lee Johnston <lee@uk.FreeBSD.org>

C.5 Shell Accounts

The following user groups provide shell accounts for people who are actively supporting the FreeBSD project. The listed administrator reserves the right to cancel the account if it is abused in any way.

Host	Access	Facilities	Administrator

Host	Access	Facilities	Administrator
dogma.freebsd-uk.eu.org	Telnet/FTP/SSH	Email, Web space, Anonymous FTP	Lee Johnston <lee@uk.FreeBSD.org>

Appendix D. PGP Keys

In case you need to verify a signature or send encrypted email to one of the officers or developers a number of keys are provided here for your convenience. A complete keyring of `FreeBSD.org` users is available for download from http://www.FreeBSD.org/doc/pgpkeyring.txt.

D.1 Officers

D.1.1 Security Officer Team <security-officer@FreeBSD.org>

```
pub  1024D/CA6CDFB2 2002-08-27 FreeBSD Security Officer <security-officer@FreeBSD.org>
     Key fingerprint = C374 0FC5 69A6 FBB1 4AED  B131 15D6 8804 CA6C DFB2
sub  2048g/A3071809 2002-08-27
```

D.1.2 Core Team Secretary <core-secretary@FreeBSD.org>

```
pub  1024R/FF8AE305 2002-01-08 core-secretary@FreeBSD.org
     Key fingerprint = CE EF 8A 48 70 00 B5 A9  55 69 DE 87 E3 9A E1 CD
```

D.1.3 Ports Management Team Secretary <portmgr-secretary@FreeBSD.org>

```
pub   1024D/7414629C 2005-11-30
      Key fingerprint = D50C BA61 8DC6 C42E 4C05  BF9A 79F6 E071 7414 629C
uid                  FreeBSD portmgr secretary <portmgr-secretary@FreeBSD.org>
sub   2048g/80B696E6 2005-11-30
```

D.2 Core Team Members

D.2.1 John Baldwin <jhb@FreeBSD.org>

```
pub  1024R/C10A874D 1999-01-13 John Baldwin <jbaldwin@weather.com>
     Key fingerprint = 43 33 1D 37 72 B1 EF 5B  9B 5F 39 F8 BD C1 7C B5
uid                            John Baldwin <john@baldwin.cx>
uid                            John Baldwin <jhb@FreeBSD.org>
uid                            John Baldwin <jobaldwi@vt.edu>
```

D.2.2 Jun Kuriyama <kuriyama@FreeBSD.org>

```
pub  1024D/FE3B59CD 1998-11-23 Jun Kuriyama <kuriyama@imgsrc.co.jp>
     Key fingerprint = 5219 55CE AC84 C296 3A3B  B076 EE3C 4DBB FE3B 59CD
```

```
uid                           Jun Kuriyama <kuriyama@FreeBSD.org>
uid                           Jun Kuriyama <kuriyama@jp.FreeBSD.org>
sub  2048g/1CF20D27 1998-11-23
```

D.2.3 Warner Losh <imp@FreeBSD.org>

```
pub   1024D/1EF6D8A7 2006-08-15
      Key fingerprint = AEC9 99C1 3212 1A86 93A6  A96B DB9F 6F12 1EF6 D8A7
uid                  M. Warner Losh <imp@bsdimp.com>
sub   4096g/34FC5B17 2006-08-15
```

D.2.4 Wes Peters <wes@FreeBSD.org>

```
pub  1024D/AD10C6C4 2000-10-19 Wes Peters <wes@freebsd.org>
     Key fingerprint = 76C4 753C 83FF D982 C57D  3A2A 7387 E292 AD10 C6C4
uid                           Wes Peters <wes@softweyr.com>
sub  1024g/237A5EF9 2000-10-19
```

D.2.5 Murray Stokely <murray@FreeBSD.org>

```
pub  1024D/0E451F7D 2001-02-12 Murray Stokely <murray@freebsd.org>
     Key fingerprint = E2CA 411D DD44 53FD BB4B  3CB5 B4D7 10A2 0E45 1F7D
sub  1024g/965A770C 2001-02-12
```

D.2.6 Peter Wemm <peter@FreeBSD.org>

```
pub  1024D/7277717F 2003-12-14 Peter Wemm <peter@wemm.org>
     Key fingerprint = 622B 2282 E92B 3BAB 57D1  A417 1512 AE52 7277 717F
uid                           Peter Wemm <peter@FreeBSD.ORG>
sub  1024g/8B40D9D1 2003-12-14
pub  1024R/D89CE319 1995-04-02 Peter Wemm <peter@netplex.com.au>
     Key fingerprint = 47 05 04 CA 4C EE F8 93  F6 DB 02 92 6D F5 58 8A
uid                           Peter Wemm <peter@perth.dialix.oz.au>
uid                           Peter Wemm <peter@haywire.dialix.com>
```

D.3 Developers

D.3.1 Ariff Abdullah <ariff@FreeBSD.org>

```
pub   1024D/C5304CDA 2005-10-01
      Key fingerprint = 5C7C 6BF4 8293 DE76 27D9  FD57 96BF 9D78 C530 4CDA
uid                  Ariff Abdullah <skywizard@MyBSD.org.my>
```

```
uid                  Ariff Abdullah <ariff@MyBSD.org.my>
uid                  Ariff Abdullah <ariff@FreeBSD.org>
sub   2048g/8958C1D3 2005-10-01
```

D.3.2 Will Andrews <will@FreeBSD.org>

```
pub  1024D/F81672C5 2000-05-22 Will Andrews (Key for official matters) <will@FreeBSD.org>
     Key fingerprint = 661F BBF7 9F5D 3D02 C862  5F6C 178E E274 F816 72C5
uid                            Will Andrews <will@physics.purdue.edu>
uid                            Will Andrews <will@puck.firepipe.net>
uid                            Will Andrews <will@c-60.org>
uid                            Will Andrews <will@csociety.org>
uid                            Will Andrews <will@csociety.ecn.purdue.edu>
uid                            Will Andrews <will@telperion.openpackages.org>
sub  1024g/55472804 2000-05-22
```

D.3.3 Eric Anholt <anholt@FreeBSD.org>

```
pub   1024D/6CF0EAF7 2003-09-08
      Key fingerprint = 76FE 2475 820B B75F DCA4  0F3E 1D47 6F60 6CF0 EAF7
uid                  Eric Anholt <eta@lclark.edu>
uid                  Eric Anholt <anholt@FreeBSD.org>
sub   1024g/80B404C1 2003-09-08
```

D.3.4 Mathieu Arnold <mat@FreeBSD.org>

```
pub   1024D/FE6D850F 2005-04-25
      Key fingerprint = 2771 11F4 0A7E 73F9 ADDD  A542 26A4 7C6A FE6D 850F
uid                  Mathieu Arnold <mat@FreeBSD.org>
uid                  Mathieu Arnold <mat@mat.cc>
uid                  Mathieu Arnold <mat@cpan.org>
uid                  Mathieu Arnold <m@absolight.fr>
uid                  Mathieu Arnold <m@absolight.net>
uid                  Mathieu Arnold <mat@club-internet.fr>
uid                  Mathieu Arnold <marnold@april.org>
uid                  Mathieu Arnold <paypal@mat.cc>
sub   2048g/EAD18BD9 2005-04-25
```

D.3.5 Satoshi Asami <asami@FreeBSD.org>

```
pub  1024R/1E08D889 1997-07-23 Satoshi Asami <asami@cs.berkeley.edu>
     Key fingerprint = EB 3C 68 9E FB 6C EB 3F  DB 2E 0F 10 8F CE 79 CA
uid                            Satoshi Asami <asami@FreeBSD.ORG>
```

D.3.6 Simon Barner <barner@FreeBSD.org>

```
pub   1024D/EBADA82A 2000-11-10
      Key fingerprint = 67D1 3562 9A2F 3177 E46A  35ED 0A49 FEFD EBAD A82A
uid                  Simon Barner <barner@FreeBSD.org>
uid                  Simon Barner <barner@in.tum.de>
uid                  Simon Barner <barner@informatik.tu-muenchen.de>
uid                  Simon Barner <barner@gmx.de>
sub   2048g/F63052DE 2000-11-10
```

D.3.7 Doug Barton <DougB@FreeBSD.org>

```
pub   1024D/D5B2F0FB 2003-01-16
      Key fingerprint = 9DD1 E44C 8660 ADA6 580F  83B6 C886 A42B D5B2 F0FB
      uid                  Doug Barton <DougB@DougBarton.us>
      uid                  Doug Barton <DougB@DougBarton.net>
      uid                  Doug Barton <DougB@FreeBSD.org>
      sub   4096g/2DBB3F89 2003-01-16
```

D.3.8 Vitaly Bogdanov <bvs@FreeBSD.org>

```
pub  1024D/B32017F7 2005-10-02 Vitaly Bogdanov <gad@gad.glazov.net>
     Key fingerprint = 402E B8E4 53CB 22FF BE62  AE35 A0BF B077 B320 17F7
uid                            Vitaly Bogdanov <bvs@freebsd.org>
sub  1024g/0E88C62E 2005-10-02
```

D.3.9 Anton Berezin <tobez@FreeBSD.org>

```
pub  1024D/7A7BA3C0 2000-05-25 Anton Berezin <tobez@catpipe.net>
     Key fingerprint = CDD8 560C 174B D8E5 0323  83CE 22CA 584C 7A7B A3C0
uid                            Anton Berezin <tobez@tobez.org>
uid                            Anton Berezin <tobez@FreeBSD.org>
sub  1024g/ADC71E87 2000-05-25
```

D.3.10 Damien Bergamini <damien@FreeBSD.org>

```
pub   2048R/D129F093 2005-03-02
      Key fingerprint = D3AB 28C3 1A4A E219 3145  54FE 220A 7486 D129 F093
uid                  Damien Bergamini <damien.bergamini@free.fr>
uid                  Damien Bergamini <damien@FreeBSD.org>
sub   2048R/9FBA73A4 2005-03-02
```

D.3.11 Tim Bishop <tdb@FreeBSD.org>

```
pub   1024D/5AE7D984 2000-10-07
      Key fingerprint = 1453 086E 9376 1A50 ECF6  AE05 7DCE D659 5AE7 D984
uid                  Tim Bishop <tim@bishnet.net>
uid                  Tim Bishop <T.D.Bishop@kent.ac.uk>
uid                  Tim Bishop <tdb@i-scream.org>
uid                  Tim Bishop <tdb@FreeBSD.org>
sub   4096g/7F886031 2000-10-07
```

D.3.12 Martin Blapp <mbr@FreeBSD.org>

```
pub  1024D/D300551E 2001-12-20 Martin Blapp <mb@imp.ch>
     Key fingerprint = B434 53FC C87C FE7B 0A18  B84C 8686 EF22 D300 551E
sub  1024g/998281C8 2001-12-20
```

D.3.13 Roman Bogorodskiy <novel@FreeBSD.org>

```
pub   1024R/1DAACA46 2004-05-25 [expires: 2009-04-26]
      Key fingerprint = AC27 CF29 5E51 E53F 8C8D  DB90 8074 5B38 1DAA CA46
uid                  Roman Bogorodskiy <novel@FreeBSD.org>
uid                  Roman Bogorodskiy <bogorodskiy@gmail.com>
uid                  Roman Bogorodskiy <bogorodskiy@inbox.ru>
uid                  Roman Bogorodskiy <novel@clublife.ru>
```

D.3.14 Hartmut Brandt <harti@FreeBSD.org>

```
pub  1024D/5920099F 2003-01-29 Hartmut Brandt <brandt@fokus.fraunhofer.de>
     Key fingerprint = F60D 09A0 76B7 31EE 794B  BB91 082F 291D 5920 099F
uid                            Hartmut Brandt <harti@freebsd.org>
sub  1024g/21D30205 2003-01-29
```

D.3.15 Oliver Braun <obraun@FreeBSD.org>

```
pub  1024D/EF25B1BA 2001-05-06 Oliver Braun <obraun@unsane.org>
     Key fingerprint = 6A3B 042A 732E 17E4 B6E7  3EAF C0B1 6B7D EF25 B1BA
uid                            Oliver Braun <obraun@obraun.net>
uid                            Oliver Braun <obraun@freebsd.org>
uid                            Oliver Braun <obraun@haskell.org>
sub  1024g/09D28582 2001-05-06
```

D.3.16 Jonathan M. Bresler <jmb@FreeBSD.org>

```
pub  1024R/97E638DD 1996-06-05 Jonathan M. Bresler <jmb@Bresler.org>
     Key fingerprint = 31 57 41 56 06 C1 40 13  C5 1C E3 E5 DC 62 0E FB
uid                            Jonathan M. Bresler <jmb@FreeBSD.ORG>
uid                            Jonathan M. Bresler
uid                            Jonathan M. Bresler <Jonathan.Bresler@USi.net>
uid                            Jonathan M. Bresler <jmb@Frb.GOV>
```

D.3.17 Christian Brüffer <brueffer@FreeBSD.org>

```
pub  1024D/A0ED982D 2002-10-14 Christian Brueffer <chris@unixpages.org>
     Key fingerprint = A5C8 2099 19FF AACA F41B  B29B 6C76 178C A0ED 982D
uid                            Christian Brueffer <brueffer@hitnet.rwth-aachen.de>
uid                            Christian Brueffer <brueffer@FreeBSD.org>
sub  4096g/1DCC100F 2002-10-14
```

D.3.18 Markus Brüffer <markus@FreeBSD.org>

```
pub   1024D/78F8A8D4 2002-10-21
      Key fingerprint = 3F9B EBE8 F290 E5CC 1447  8760 D48D 1072 78F8 A8D4
uid                  Markus Brueffer <markus@brueffer.de>
uid                  Markus Brueffer <buff@hitnet.rwth-aachen.de>
uid                  Markus Brueffer <mbrueffer@mi.rwth-aachen.de>
uid                  Markus Brueffer <markus@FreeBSD.org>
sub   4096g/B7E5C7B6 2002-10-21
```

D.3.19 Wilko Bulte <wilko@FreeBSD.org>

```
pub   1024D/186B8DBD 2006-07-29
      Key fingerprint = 07C2 6CB3 9C18 D290 6C5F  8879 CF83 EC86 186B 8DBD
uid                  Wilko Bulte (wilko@FreeBSD.org) <wilko@FreeBSD.org>
sub   2048g/1C4683F1 2006-07-29
```

D.3.20 Oleg Bulyzhin <oleg@FreeBSD.org>

```
pub   1024D/78CE105F 2004-02-06
      Key fingerprint = 98CC 3E66 26DE 50A8 DBC4  EB27 AF22 DCEF 78CE 105F
uid                  Oleg Bulyzhin <oleg@FreeBSD.org>
uid                  Oleg Bulyzhin <oleg@rinet.ru>
sub   1024g/F747C159 2004-02-06
```

D.3.21 Jesus R. Camou <jcamou@FreeBSD.org>

```
pub   1024D/C2161947 2005-03-01
      Key fingerprint = 274C B265 48EC 42AE A2CA  47D9 7D98 588A C216 1947
uid                  Jesus R. Camou <jcamou@FreeBSD.org>
sub   2048g/F8D2A8DF 2005-03-01
```

D.3.22 Hye-Shik Chang <perky@FreeBSD.org>

```
pub  1024D/CFDB4BA4 1999-04-23 Hye-Shik Chang <perky@FreeBSD.org>
     Key fingerprint = 09D9 57D6 58BA 44DD CAEC  71CD 0D65 2C59 CFDB 4BA4
uid                           Hye-Shik Chang <hyeshik@gmail.com>
sub  1024g/A94A8ED1 1999-04-23
```

D.3.23 Jonathan Chen <jon@FreeBSD.org>

```
pub  1024D/2539468B 1999-10-11 Jonathan Chen <jon@spock.org>
     Key fingerprint = EE31 CDA1 A105 C8C9 5365  3DB5 C2FC 86AA 2539 468B
uid                           Jonathan Chen <jon@freebsd.org>
uid                           Jonathan Chen <chenj@rpi.edu>
uid                           Jonathan Chen <spock@acm.rpi.edu>
uid                           Jonathan Chen <jon@cs.rpi.edu>
sub  3072g/B81EF1DB 1999-10-11
```

D.3.24 陳洛祁<luoqi@FreeBSD.org>

```
pub  1024D/2926F3BE 2002-02-22 Luoqi Chen <luoqi@FreeBSD.org>
     Key fingerprint = B470 A815 5917 D9F4 37F3  CE2A 4D75 3BD1 2926 F3BE
uid                           Luoqi Chen <luoqi@bricore.com>
uid                           Luoqi Chen <lchen@onetta.com>
sub  1024g/5446EB72 2002-02-22
```

D.3.25 Andrey A. Chernov <ache@FreeBSD.org>

```
pub   1024D/964474DD 2006-12-26
      Key fingerprint = 0F63 1B61 D76D AA23 1591  EA09 560E 582B 9644 74DD
uid                   Andrey Chernov <ache@freebsd.org>
uid                   [jpeg image of size 4092]
sub   2048g/08331894 2006-12-26
```

D.3.26 Sean Chittenden <seanc@FreeBSD.org>

```
pub  1024D/EE278A28 2004-02-08 Sean Chittenden <sean@chittenden.org>
     Key fingerprint = E41F F441 7E91 6CBA 1844  65CF B939 3C78 EE27 8A28
sub  2048g/55321853 2004-02-08
```

D.3.27 Junho CHOI <cjh@FreeBSD.org>

```
pub  1024D/E60260F5 2002-10-14 CHOI Junho (Work) <cjh@wdb.co.kr>
     Key fingerprint = 1369 7374 A45F F41A F3C0  07E3 4A01 C020 E602 60F5
uid                            CHOI Junho (Personal) <cjh@kr.FreeBSD.org>
uid                            CHOI Junho (FreeBSD) <cjh@FreeBSD.org>
sub  1024g/04A4FDD8 2002-10-14
```

D.3.28 Crist J. Clark <cjc@FreeBSD.org>

```
pub  1024D/FE886AD3 2002-01-25 Crist J. Clark <cjclark@jhu.edu>
     Key fingerprint = F04E CCD7 3834 72C2 707F  0A8F 259F 8F4B FE88 6AD3
uid                            Crist J. Clark <cjclark@alum.mit.edu>
uid                            Crist J. Clark <cjc@freebsd.org>
sub  1024g/9B6BAB99 2002-01-25
```

D.3.29 Joe Marcus Clarke <marcus@FreeBSD.org>

```
pub  1024D/FE14CF87 2002-03-04 Joe Marcus Clarke (FreeBSD committer address) <marcus@FreeBSD.org>
     Key fingerprint = CC89 6407 73CC 0286 28E4  AFB9 6F68 8F8A FE14 CF87
uid                            Joe Marcus Clarke <marcus@marcuscom.com>
sub  1024g/B9ACE4D2 2002-03-04
```

D.3.30 Nik Clayton <nik@FreeBSD.org>

```
pub  1024D/2C37E375 2000-11-09 Nik Clayton <nik@freebsd.org>
     Key fingerprint = 15B8 3FFC DDB4 34B0 AA5F  94B7 93A8 0764 2C37 E375
uid                            Nik Clayton <nik@slashdot.org>
uid                            Nik Clayton <nik@crf-consulting.co.uk>
uid                            Nik Clayton <nik@ngo.org.uk>
uid                            Nik Clayton <nik@bsdi.com>
sub  1024g/769E298A 2000-11-09
```

D.3.31 Aaron Dalton <aaron@FreeBSD.org>

```
pub   1024D/8811D2A4 2006-06-21 [expires: 2011-06-20]
      Key fingerprint = 8DE0 3CBB 3692 992F 53EF  ACC7 BE56 0A4D 8811 D2A4
uid                  Aaron Dalton <aaron@freebsd.org>
```

```
sub   2048g/304EE8E5 2006-06-21 [expires: 2011-06-20]
```

D.3.32 Ceri Davies <ceri@FreeBSD.org>

```
pub   1024D/34B7245F 2002-03-08
      Key fingerprint = 9C88 EB05 A908 1058 A4AE  9959 A1C7 DCC1 34B7 245F
uid                  Ceri Davies <ceri@submonkey.net>
uid                  Ceri Davies <ceri@FreeBSD.org>
uid                  Ceri Davies <ceri@opensolaris.org>
sub   1024g/0C482CBC 2002-03-08
```

D.3.33 Brad Davis <brd@FreeBSD.org>

```
pub   1024D/ED0A754D 2005-05-14 [expires: 2014-02-21]
      Key fingerprint = 5DFD D1A6 BEEE A6D4 B3F5  4236 D362 3291 ED0A 754D
uid                  Brad Davis <so14k@so14k.com>
uid                  Brad Davis <brd@FreeBSD.org>
sub   2048g/1F29D404 2005-05-14 [expires: 2014-02-21]
```

D.3.34 Brooks Davis <brooks@FreeBSD.org>

```
pub  1024D/F2381AD4 2001-02-10 Brooks Davis (The Aerospace Corporation) <brooks@aero.org>
     Key fingerprint = 655D 519C 26A7 82E7 2529  9BF0 5D8E 8BE9 F238 1AD4
uid                            Brooks Davis <brooks@one-eyed-alien.net>
uid                            Brooks Davis <brooks@FreeBSD.org>
uid                            Brooks Davis <brooks@aero.org>
sub  2048g/CFDACA7A 2003-01-25 [expires: 2008-01-24]
sub  1024g/42921194 2001-02-10 [expires: 2009-02-08]
```

D.3.35 George V. Neville-Neil <gnn@FreeBSD.org>

```
pub   1024D/440A33D2 2002-09-17
      Key fingerprint = AF66 410F CC8D 1FC9 17DB  6225 61D8 76C1 440A 33D2
uid                  George V. Neville-Neil <gnn@freebsd.org>
uid                  George V. Neville-Neil <gnn@neville-neil.com>
sub   2048g/95A74F6E 2002-09-17
```

D.3.36 Pawel Jakub Dawidek <pjd@FreeBSD.org>

```
pub  1024D/B1293F34 2004-02-02 Pawel Jakub Dawidek <Pawel@Dawidek.net>
     Key fingerprint = A3A3 5B4D 9CF9 2312 0783  1B1D 168A EF5D B129 3F34
uid                            Pawel Jakub Dawidek <pjd@FreeBSD.org>
uid                            Pawel Jakub Dawidek <pjd@FreeBSD.pl>
sub  2048g/3EEC50A7 2004-02-02 [expires: 2006-02-01]
```

D.3.37 Brian S. Dean <bsd@FreeBSD.org>

```
pub  1024D/723BDEE9 2002-01-23 Brian S. Dean <bsd@FreeBSD.org>
     Key fingerprint = EF49 7ABE 47ED 91B3 FC3D  7EA5 4D90 2FF7 723B DEE9
sub  1024g/4B02F876 2002-01-23
```

D.3.38 Alexey Dokuchaev <danfe@FreeBSD.org>

```
pub  1024D/3C060B44 2004-08-23 Alexey Dokuchaev <danfe@FreeBSD.org>
     Key fingerprint = D970 08A4 922C 8D63 0C19  8D27 F421 76EE 3C06 0B44
sub  1024g/70BAE967 2004-08-23
```

D.3.39 Dima Dorfman <dd@FreeBSD.org>

```
pub   1024D/69FAE582 2001-09-04
      Key fingerprint = B340 8338 7DA3 4D61 7632  098E 0730 055B 69FA E582
uid                  Dima Dorfman <dima@trit.org>
uid                  Dima Dorfman <dima@unixfreak.org>
uid                  Dima Dorfman <dd@freebsd.org>
sub   2048g/65AF3B89 2003-08-19 [expires: 2005-08-18]
sub   2048g/8DB0CF2C 2005-05-29 [expires: 2007-05-29]
```

D.3.40 Bruno Ducrot <bruno@FreeBSD.org>

```
pub   1024D/7F463187 2000-12-29
      Key fingerprint = 7B79 E1D6 F5A1 6614 792F  D906 899B 4D28 7F46 3187
uid                  Ducrot Bruno (Poup Master) <ducrot@poupinou.org>
sub   1024g/40282874 2000-12-29
```

D.3.41 Alex Dupre <ale@FreeBSD.org>

```
pub  1024D/CE5F554D 1999-06-27 Alex Dupre <sysadmin@alexdupre.com>
     Key fingerprint = DE23 02EA 5927 D5A9 D793  2BA2 8115 E9D8 CE5F 554D
uid                           Alex Dupre <ale@FreeBSD.org>
uid                           [jpeg image of size 5544]
uid                           Alex Dupre <ICQ:5431856>
sub  2048g/FD5E2D21 1999-06-27
```

D.3.42 Peter Edwards <peadar@FreeBSD.org>

```
pub  1024D/D80B4B3F 2004-03-01 Peter Edwards <peadar@FreeBSD.org>
     Key fingerprint = 7A8A 9756 903E BEF2 4D9E  3C94 EE52 52F7 D80B 4B3F
uid                           Peter Edwards <pmedwards@eircom.net>
```

D.3.43 Josef El-Rayes <josef@FreeBSD.org>

```
pub  2048R/A79DB53C 2004-01-04 Josef El-Rayes <josef@FreeBSD.org>
     Key fingerprint = 58EB F5B7 2AB9 37FE 33C8  716B 59C5 22D9 A79D B53C
uid                            Josef El-Rayes <josef@daemon.li>
```

D.3.44 Udo Erdelhoff <ue@FreeBSD.org>

```
pub  1024R/E74FA871 1994-07-19 Udo Erdelhoff <uer@de.uu.net>
     Key fingerprint = 8C B1 80 CA 2C 52 73 81  FB A7 B4 03 C5 32 C8 67
uid                            Udo Erdelhoff <ue@nathan.ruhr.de>
uid                            Udo Erdelhoff <ue@freebsd.org>
uid                            Udo Erdelhoff <uerdelho@eu.uu.net>
uid                            Udo Erdelhoff <uerdelho@uu.net>
```

D.3.45 Ruslan Ermilov <ru@FreeBSD.org>

```
pub  1024D/996E145E 2004-06-02 Ruslan Ermilov (FreeBSD) <ru@FreeBSD.org>
     Key fingerprint = 274E D201 71ED 11F6 9CCB  0194 A917 E9CC 996E 145E
uid                            Ruslan Ermilov (FreeBSD Ukraine) <ru@FreeBSD.org.ua>
uid                            Ruslan Ermilov (IPNet) <ru@ip.net.ua>
sub  1024g/557E3390 2004-06-02 [expires: 2007-06-02]
```

D.3.46 Lukas Ertl <le@FreeBSD.org>

```
pub  1024D/F10D06CB 2000-11-23 Lukas Ertl <le@FreeBSD.org>
     Key fingerprint = 20CD C5B3 3A1D 974E 065A  B524 5588 79A9 F10D 06CB
uid                            Lukas Ertl <a9404849@unet.univie.ac.at>
uid                            Lukas Ertl <l.ertl@univie.ac.at>
uid                            Lukas Ertl <le@univie.ac.at>
sub  1024g/5960CE8E 2000-11-23
```

D.3.47 Stefan Farfeleder <stefanf@FreeBSD.org>

```
pub  1024D/8BEFD15F 2004-03-14 Stefan Farfeleder <stefan@fafoe.narf.at>
     Key fingerprint = 4220 FE60 A4A1 A490 5213  27A6 319F 8B28 8BEF D15F
uid                            Stefan Farfeleder <stefanf@complang.tuwien.ac.at>
uid                            Stefan Farfeleder <stefanf@FreeBSD.org>
uid                            Stefan Farfeleder <stefanf@ten15.org>
sub  2048g/418753E9 2004-03-14 [expires: 2007-03-14]
```

D.3.48 Chris D. Faulhaber <jedgar@FreeBSD.org>

```
pub  1024D/FE817A50 2000-12-20 Chris D. Faulhaber <jedgar@FreeBSD.org>
     Key fingerprint = A47D A838 9216 F921 A456  54FF 39B6 86E0 FE81 7A50
uid                            Chris D. Faulhaber <jedgar@fxp.org>
sub  2048g/93452698 2000-12-20
```

D.3.49 Brian F. Feldman <green@FreeBSD.org>

```
pub  1024D/41C13DE3 2000-01-11 Brian Fundakowski Feldman <green@FreeBSD.org>
     Key fingerprint = 6A32 733A 1BF6 E07B 5B8D  AE14 CC9D DCA2 41C1 3DE3
sub  1024g/A98B9FCC 2000-01-11 [expires: 2001-01-10]

pub  1024D/773905D6 2000-09-02 Brian Fundakowski Feldman <green@FreeBSD.org>
     Key fingerprint = FE23 7481 91EA 5E58 45EA  6A01 B552 B043 7739 05D6
sub  2048g/D2009B98 2000-09-02
```

D.3.50 Mário Sérgio Fujikawa Ferreira <lioux@FreeBSD.org>

```
pub   1024D/75A63712 2006-02-23 [expires: 2007-02-23]
      Key fingerprint = 42F2 2F74 8EF9 5296 898F  C981 E9CF 463B 75A6 3712
uid                  Mario Sergio Fujikawa Ferreira (lioux) <lioux@FreeBSD.org>
uid                  Mario Sergio Fujikawa Ferreira <lioux@uol.com.br>
sub   4096g/BB7D80F2 2006-02-23 [expires: 2007-02-23]
```

D.3.51 Tony Finch <fanf@FreeBSD.org>

```
pub  1024D/84C71B6E 2002-05-03 Tony Finch <dot@dotat.at>
     Key fingerprint = 199C F25B 2679 6D04 63C5  2159 FFC0 F14C 84C7 1B6E
uid                            Tony Finch <fanf@FreeBSD.org>
uid                            Tony Finch <fanf@apache.org>
uid                            Tony Finch <fanf2@cam.ac.uk>
sub  2048g/FD101E8B 2002-05-03
```

D.3.52 Marc Fonvieille <blackend@FreeBSD.org>

```
pub  1024D/4F8E74E8 2004-12-25 Marc Fonvieille <blackend@FreeBSD.org>
     Key fingerprint = 55D3 4883 4A04 828A A139  A5CF CD0F 51C0 4F8E 74E8
uid                            Marc Fonvieille <marc@blackend.org>
uid                            Marc Fonvieille <marc@freebsd-fr.org>
sub  1024g/37AD4E7D 2004-12-25
```

D.3.53 Pete Fritchman <petef@FreeBSD.org>

```
pub  1024D/74B91CFD 2001-01-30 Pete Fritchman <petef@FreeBSD.org>
     Key fingerprint = 9A9F 8A13 DB0D 7777 8D8E  1CB2 C5C9 A08F 74B9 1CFD
uid                            Pete Fritchman <petef@databits.net>
uid                            Pete Fritchman <petef@csh.rit.edu>
sub  1024g/0C02AF0C 2001-01-30
```

D.3.54 Bill Fumerola <billf@FreeBSD.org>

```
pub  1024D/7F868268 2000-12-07 Bill Fumerola (FreeBSD Developer) <billf@FreeBSD.org>
     Key fingerprint = 5B2D 908E 4C2B F253 DAEB  FC01 8436 B70B 7F86 8268
uid                            Bill Fumerola (Security Yahoo) <fumerola@yahoo-inc.com>
sub  1024g/43980DA9 2000-12-07
```

D.3.55 Sebastien Gioria <gioria@FreeBSD.org>

```
pub  1024D/7C8DA4F4 2002-02-09 Sebastien Gioria <eagle@freebsd-fr.org>
     Key fingerprint = 41F4 4885 7C23 6ED3 CC24  97AA 6DDD B426 7C8D A4F4
uid                            Sebastien Gioria <gioria@FreeBSD.ORG>
uid                            Sebastien Gioria <gioria@Francenet.fr>
uid                            Sebastien Gioria <gioria@fluxus.net>
sub  4096g/F147E4D3 2002-02-09
```

D.3.56 Marcus Alves Grando <mnag@FreeBSD.org>

```
pub   1024D/CDCC273F 2005-09-15 [expires: 2010-09-14]
      Key fingerprint = 57F9 DEC1 5BBF 06DE 44A5  9A4A 8BEE 5F3A CDCC 273F
uid                  Marcus Alves Grando <marcus@sbh.eng.br>
uid                  Marcus Alves Grando <marcus@corp.grupos.com.br>
uid                  Marcus Alves Grando <mnag@FreeBSD.org>
sub   2048g/698AC00C 2005-09-15 [expires: 2010-09-14]
```

D.3.57 John-Mark Gurney <jmg@FreeBSD.org>

```
pub  1024R/3F9951F5 1997-02-11 John-Mark Gurney <johnmark@gladstone.uoregon.edu>
     Key fingerprint = B7 EC EF F8 AE ED A7 31  96 7A 22 B3 D8 56 36 F4
uid                            John-Mark Gurney <gurney_j@efn.org>
uid                            John-Mark Gurney <jmg@cs.uoregon.edu>
uid                            John-Mark Gurney <gurney_j@resnet.uoregon.edu>
```

D.3.58 Daniel Harris <dannyboy@FreeBSD.org>

```
pub  1024D/84D0D7E7 2001-01-15 Daniel Harris <dannyboy@worksforfood.com>
     Key fingerprint = 3C61 B8A1 3F09 D194 3259  7173 6C63 DA04 84D0 D7E7
uid                           Daniel Harris <dannyboy@freebsd.org>
uid                           Daniel Harris <dh@askdh.com>
uid                           Daniel Harris <dh@wordassault.com>
sub  1024g/9DF0231A 2001-01-15
```

D.3.59 Daniel Hartmeier <dhartmei@FreeBSD.org>

```
pub  1024R/6A3A7409 1994-08-15 Daniel Hartmeier <dhartmei@freebsd.org>
     Key fingerprint = 13 7E 9A F3 36 82 09 FE  FD 57 B8 5C 2B 81 7E 1F
```

D.3.60 John Hay <jhay@FreeBSD.org>

```
pub  2048R/A9275B93 2000-05-10 John Hay <jhay@icomtek.csir.co.za>
     Key fingerprint = E7 95 F4 B9 D4 A7 49 6A  83 B9 77 49 28 9E 37 70
uid                           John Hay <jhay@mikom.csir.co.za>
uid                           Thawte Freemail Member <jhay@mikom.csir.co.za>
uid                           John Hay <jhay@csir.co.za>
uid                           John Hay <jhay@FreeBSD.ORG>
```

D.3.61 Sheldon Hearn <sheldonh@FreeBSD.org>

```
pub  1024D/74A06ACD 2002-06-20 Sheldon Hearn <sheldonh@starjuice.net>
     Key fingerprint = 01A3 EF91 9C5A 3633 4E01  8085 A462 57F1 74A0 6ACD
sub  1536g/C42F8AC8 2002-06-20
```

D.3.62 Mike Heffner <mikeh@FreeBSD.org>

```
pub  1024D/CDECBF99 2001-02-02 Michael Heffner <mheffner@novacoxmail.com>
     Key fingerprint = AFAB CCEB 68C7 573F 5110  9285 1689 1942 CDEC BF99
uid                           Michael Heffner <mheffner@vt.edu>
uid                           Michael Heffner <mikeh@FreeBSD.org>
uid                           Michael Heffner <spock@techfour.net>
uid                           Michael Heffner (ACM sysadmin) <mheffner@acm.vt.edu>
sub  1024g/3FE83FB5 2001-02-02
```

D.3.63 Martin Heinen <mheinen@FreeBSD.org>

```
pub  1024D/116C5C85 2002-06-17 Martin Heinen <mheinen@freebsd.org>
     Key fingerprint = C898 3FCD EEA0 17ED BEA9  564D E5A6 AFF2 116C 5C85
uid                           Martin Heinen <martin@sumuk.de>
```

```
sub  1024g/EA67506B 2002-06-17
```

D.3.64 Niels Heinen <niels@FreeBSD.org>

```
pub  1024D/5FE39B80 2004-12-06 Niels Heinen <niels.heinen@ubizen.com>
     Key fingerprint = 75D8 4100 CF5B 3280 543F  930C 613E 71AA 5FE3 9B80
uid                            Niels Heinen <niels@defaced.be>
uid                            Niels Heinen <niels@heinen.ws>
uid                            Niels Heinen <niels@FreeBSD.org>
sub  2048g/057F4DA7 2004-12-06
```

D.3.65 Guy Helmer <ghelmer@FreeBSD.org>

```
pub  1024R/35F4ED2D 1997-01-26 Guy G. Helmer <ghelmer@freebsd.org>
     Key fingerprint = A2 59 4B 92 02 5B 9E B1  B9 4E 2E 03 29 D5 DC 3A
uid                            Guy G. Helmer <ghelmer@cs.iastate.edu>
uid                            Guy G. Helmer <ghelmer@palisadesys.com>
```

D.3.66 Maxime Henrion <mux@FreeBSD.org>

```
pub  1024D/881D4806 2003-01-09 Maxime Henrion <mux@FreeBSD.org>
     Key fingerprint = 81F1 BE2D 12F1 184A 77E4  ACD0 5563 7614 881D 4806
sub  2048g/D0B510C0 2003-01-09
```

D.3.67 Michael L. Hostbaek <mich@FreeBSD.org>

```
pub  1024D/0F55F6BE 2001-08-07 Michael L. Hostbaek <mich@freebsdcluster.org>
     Key fingerprint = 4D62 9396 B19F 38D3 5C99  1663 7B0A 5212 0F55 F6BE
uid                            Michael L. Hostbaek <mich@freebsdcluster.dk>
uid                            Michael L. Hostbaek <mich@icommerce-france.com>
uid                            Micahel L. Hostbaek <mich@freebsd.dk>
uid                            Michael L. Hostbaek <mich@the-lab.org>
uid                            Michael L. Hostbaek <mich@freebsd.org>
sub  1024g/8BE4E30F 2001-08-07
```

D.3.68 胡庭豪<foxfair@FreeBSD.org>

```
pub  1024D/4E9BCA59 2003-09-01 Foxfair Hu <foxfair@FreeBSD.org>
     Key fingerprint = 280C A846 CA1B CAC9 DDCF  F4CB D553 4BD5 4E9B CA59
uid                            Foxfair Hu <foxfair@drago.fomokka.net>
uid                            Howard Hu <howardhu@yahoo-inc.com>
sub  1024g/3356D8C1 2003-09-01
```

D.3.69 Jordan K. Hubbard <jkh@FreeBSD.org>

```
pub  1024R/8E542D5D 1996-04-04 Jordan K. Hubbard <jkh@FreeBSD.org>
     Key fingerprint = 3C F2 27 7E 4A 6C 09 0A  4B C9 47 CD 4F 4D 0B 20
```

D.3.70 Michael Johnson <ahze@FreeBSD.org>

```
pub  1024D/3C046FD6 2004-10-29 Michael Johnson (FreeBSD key) <ahze@FreeBSD.org>
     Key fingerprint = 363C 6ABA ED24 C23B 5F0C  3AB4 9F8B AA7D 3C04 6FD6
uid                            Michael Johnson (pgp key) <ahze@ahze.net>
sub  2048g/FA334AE3 2004-10-29
```

D.3.71 Trevor Johnson <trevor@FreeBSD.org>

```
pub  1024D/3A3EA137 2000-04-20 Trevor Johnson <trevor@jpj.net>
     Key fingerprint = 7ED1 5A92 76C1 FFCB E5E3  A998 F037 5A0B 3A3E A137
sub  1024g/46C24F1E 2000-04-20
```

D.3.72 Poul-Henning Kamp <phk@FreeBSD.org>

```
pub  1024R/0358FCBD 1995-08-01 Poul-Henning Kamp <phk@FreeBSD.org>
     Key fingerprint = A3 F3 88 28 2F 9B 99 A2  49 F4 E2 FA 5A 78 8B 3E
```

D.3.73 Josef Karthauser <joe@FreeBSD.org>

```
pub  1024D/E6B15016 2000-10-19 Josef Karthauser <joe@FreeBSD.org>
     Key fingerprint = 7266 8EAF 82C2 D439 5642  AC26 5D52 1C8C E6B1 5016
uid                            Josef Karthauser <joe@tao.org.uk>
uid                            Josef Karthauser <joe@uk.FreeBSD.org>
uid                            [revoked] Josef Karthauser <josef@bsdi.com>
uid                            [revoked] Josef Karthauser <joe@pavilion.net>
sub  2048g/1178B692 2000-10-19
```

D.3.74 Vinod Kashyap <vkashyap@FreeBSD.org>

```
pub  1024R/04FCCDD3 2004-02-19 Vinod Kashyap (gnupg key) <vkashyap@freebsd.org>
     Key fingerprint = 9B83 0B55 604F E491 B7D2  759D DF92 DAA0 04FC CDD3
```

D.3.75 Kris Kennaway <kris@FreeBSD.org>

```
pub  1024D/68E840A5 2000-01-14 Kris Kennaway <kris@citusc.usc.edu>
     Key fingerprint = E65D 0E7D 7E16 B212 1BD6  39EE 5ABC B405 68E8 40A5
uid                            Kris Kennaway <kris@FreeBSD.org>
```

```
uid                          Kris Kennaway <kris@obsecurity.org>
sub  2048g/03A41C45 2000-01-14 [expires: 2006-01-14]
```

D.3.76 Giorgos Keramidas <keramida@FreeBSD.org>

```
pub   1024D/318603B6 2001-09-21
      Key fingerprint = C1EB 0653 DB8B A557 3829  00F9 D60F 941A 3186 03B6
uid                  Giorgos Keramidas <keramida@FreeBSD.org>
uid                  Giorgos Keramidas <keramida@ceid.upatras.gr>
uid                  Giorgos Keramidas <keramida@hellug.gr>
uid                  Giorgos Keramidas <keramida@linux.gr>
sub   1024g/50FDBAD1 2001-09-21
```

D.3.77 Max Khon <fjoe@FreeBSD.org>

```
pub  1024D/414420F4 2003-04-29 Max Khon <fjoe@freebsd.org>
     Key fingerprint = CE1F 29CA A6BF 2F26 13E8  1B61 62AE 6B8F 4144 20F4
uid                          Max Khon <fjoe@iclub.nsu.ru>
sub  1024g/6585039B 2003-04-29
```

D.3.78 Jung-uk Kim <jkim@FreeBSD.org>

```
pub   1024D/BF6A9D53 2004-04-07
      Key fingerprint = F841 0339 93EF D27D 32AD  3261 9A56 B2D5 BF6A 9D53
uid                  Jung-uk Kim <jkim@FreeBSD.org>
uid                  Jung-uk Kim <jkim@niksun.com>
sub   4096g/B01CA5A0 2004-04-07
```

D.3.79 Andreas Klemm <andreas@FreeBSD.org>

```
pub  1024D/6C6F6CBA 2001-01-06 Andreas Klemm <andreas.klemm@eu.didata.com>
     Key fingerprint = F028 D51A 0D42 DD67 4109  19A3 777A 3E94 6C6F 6CBA
uid                          Andreas Klemm <andreas@klemm.gtn.com>
uid                          Andreas Klemm <andreas@FreeBSD.org>
uid                          Andreas Klemm <andreas@apsfilter.org>
sub  2048g/FE23F866 2001-01-06
```

D.3.80 Johann Kois <jkois@FreeBSD.org>

```
pub  1024D/DD61C2D8 2004-06-27 Johann Kois <J.Kois@web.de>
     Key fingerprint = 8B70 03DB 3C45 E71D 0ED4  4825 FEB0 EBEF DD61 C2D8
uid                          Johann Kois <jkois@freebsd.org>
sub  1024g/568307CB 2004-06-27
```

D.3.81 Sergei Kolobov <sergei@FreeBSD.org>

```
pub  1024D/3BA53401 2003-10-10 Sergei Kolobov <sergei@FreeBSD.org>
     Key fingerprint = A2F4 5F34 0586 CC9C 493A  347C 14EC 6E69 3BA5 3401
uid                            Sergei Kolobov <sergei@kolobov.com>
sub  2048g/F8243671 2003-10-10
```

D.3.82 Maxim Konovalov <maxim@FreeBSD.org>

```
pub  1024D/2C172083 2002-05-21 Maxim Konovalov <maxim@FreeBSD.org>
     Key fingerprint = 6550 6C02 EFC2 50F1 B7A3  D694 ECF0 E90B 2C17 2083
uid                            Maxim Konovalov <maxim@macomnet.ru>
sub  1024g/F305DDCA 2002-05-21
```

D.3.83 Joseph Koshy <jkoshy@FreeBSD.org>

```
pub  1024D/D93798B6 2001-12-21 Joseph Koshy (FreeBSD) <jkoshy@freebsd.org>
     Key fingerprint = 0DE3 62F3 EF24 939F 62AA  2E3D ABB8 6ED3 D937 98B6
sub  1024g/43FD68E9 2001-12-21
```

D.3.84 Roman Kurakin <rik@FreeBSD.org>

```
pub   1024D/C8550F4C 2005-12-16 [expires: 2008-12-15]
      Key fingerprint = 25BB 789A 6E07 E654 8E59  0FA9 42B1 937C C855 0F4C
uid                  Roman Kurakin <rik@FreeBSD.org>
sub   2048g/D15F2AB6 2005-12-16 [expires: 2008-12-15]
```

D.3.85 Hideyuki KURASHINA <rushani@FreeBSD.org>

```
pub  1024D/439ADC57 2002-03-22 Hideyuki KURASHINA <rushani@bl.mmtr.or.jp>
     Key fingerprint = A052 6F98 6146 6FE3 91E2  DA6B F2FA 2088 439A DC57
uid                            Hideyuki KURASHINA <rushani@FreeBSD.org>
uid                            Hideyuki KURASHINA <rushani@jp.FreeBSD.org>
sub  1024g/64764D16 2002-03-22
```

D.3.86 Clement Laforet <clement@FreeBSD.org>

```
pub  1024D/0723BA1D 2003-12-13 Clement Laforet (FreeBSD committer address) <clement@FreeBSD.org>
     Key fingerprint = 3638 4B14 8463 A67B DC7E  641C B118 5F8F 0723 BA1D
uid                            Clement Laforet <sheepkiller@cultdeadsheep.org>
uid                            Clement Laforet <clement.laforet@cotds.org>
sub  2048g/23D57658 2003-12-13
```

D.3.87 Max Laier <mlaier@FreeBSD.org>

```
pub   1024D/3EB6046D 2004-02-09
      Key fingerprint = 917E 7F25 E90F 77A4 F746  2E8D 5F2C 84A1 3EB6 046D
uid                  Max Laier <max@love2party.net>
uid                  Max Laier <max.laier@ira.uka.de>
uid                  Max Laier <mlaier@freebsd.org>
uid                  Max Laier <max.laier@tm.uka.de>
sub   4096g/EDD08B9B 2005-06-28
```

D.3.88 Erwin Lansing <erwin@FreeBSD.org>

```
pub   1024D/15256990 1998-07-03
      Key fingerprint = FB58 9797 299A F18E 2D3E  73D6 AB2F 5A5B 1525 6990
uid                  Erwin Lansing <erwin@lansing.dk>
uid                  Erwin Lansing <erwin@FreeBSD.org>
uid                  Erwin Lansing <erwin@droso.dk>
uid                  Erwin Lansing <erwin@droso.org>
uid                  Erwin Lansing <erwin@aauug.dk>
sub   2048g/7C64013D 1998-07-03
```

D.3.89 Sam Lawrance <lawrance@FreeBSD.org>

```
pub   1024D/32708C59 2003-08-14
      Key fingerprint = 1056 2A02 5247 64D4 538D  6975 8851 7134 3270 8C59
uid                  Sam Lawrance <lawrance@FreeBSD.org>
uid                  Sam Lawrance <boris@brooknet.com.au>
sub   2048g/0F9CCF92 2003-08-14
```

D.3.90 李彦明<leeym@FreeBSD.org>

```
pub   1024D/93FA8BD6 2007-05-21
      Key fingerprint = DEC4 6E7F 69C0 4AC3 21ED  EE65 6C0E 9257 93FA 8BD6
uid                  Yen-Ming Lee <leeym@leeym.com>
sub   2048g/899A3931 2007-05-21
```

D.3.91 Sam Leffler <sam@FreeBSD.org>

```
pub   1024D/BD147743 2005-03-28
      Key fingerprint = F618 F2FC 176B D201 D91C  67C6 2E33 A957 BD14 7743
uid                  Samuel J. Leffler <sam@freebsd.org>
sub   2048g/8BA91D05 2005-03-28
```

D.3.92 Jean-Yves Lefort <jylefort@FreeBSD.org>

```
pub   1024D/A3B8006A 2002-09-07
      Key fingerprint = CC99 D1B0 8E44 293D 32F7  D92E CB30 FB51 A3B8 006A
uid                  Jean-Yves Lefort <jylefort@FreeBSD.org>
uid                  Jean-Yves Lefort <jylefort@brutele.be>
sub   4096g/C9271AFC 2002-09-07
```

D.3.93 Alexander Leidinger <netchild@FreeBSD.org>

```
pub   1024D/72077137 2002-01-31
      Key fingerprint = AA3A 8F69 B214 6BBD 5E73  C9A0 C604 3C56 7207 7137
uid                  Alexander Leidinger <netchild@FreeBSD.org>
uid                  [jpeg image of size 19667]
sub   2048g/8C9828D3 2002-01-31
```

D.3.94 Dejan Lesjak <lesi@FreeBSD.org>

```
pub  1024D/96C5221F 2004-08-18 Dejan Lesjak <lesi@FreeBSD.org>
     Key fingerprint = 2C5C 02EA 1060 1D6D 9982  38C0 1DA7 DBC4 96C5 221F
uid                           Dejan Lesjak <dejan.lesjak@ijs.si>
sub  1024g/E0A69278 2004-08-18
```

D.3.95 Greg Lewis <glewis@FreeBSD.org>

```
pub  1024D/1BB6D9E0 2002-03-05 Greg Lewis (FreeBSD) <glewis@FreeBSD.org>
     Key fingerprint = 2410 DA6D 5A3C D801 65FE  C8DB DEEA 9923 1BB6 D9E0
uid                           Greg Lewis <glewis@eyesbeyond.com>
sub  2048g/45E67D60 2002-03-05
```

D.3.96 李鑫<delphij@FreeBSD.org>

```
pub   1024D/CAEEB8C0 2004-01-28
      Key fingerprint = 43B8 B703 B8DD 0231 B333  DC28 39FB 93A0 CAEE B8C0
uid                  Xin LI <delphij@FreeBSD.org>
uid                  Xin LI <delphij@frontfree.net>
uid                  Xin LI <delphij@delphij.net>
uid                  Xin LI <delphij@geekcn.org>

pub   1024D/42EA8A4B 2006-01-27 [expired: 2008-01-01]
      Key fingerprint = F19C 2616 FA97 9C13 2581  C6F3 85C5 1CCE 42EA 8A4B
uid                  Xin LI <delphij@geekcn.org>
uid                  Xin LI <delphij@FreeBSD.org>
uid                  Xin LI <delphij@delphij.net>

pub   1024D/18EDEBA0 2008-01-02 [expires: 2010-01-02]
```

```
      Key fingerprint = 79A6 CF42 F917 DDCA F1C2  C926 8BEB DB04 18ED EBA0
uid                  Xin LI <delphij@geekcn.org>
uid                  Xin LI <delphij@FreeBSD.org>
uid                  Xin LI <delphij@delphij.net>
sub   4096g/8ED8F128 2008-01-02 [expires: 2010-01-02]
```

D.3.97 梁泰華<avatar@FreeBSD.org>

```
pub  1024R/F4013AB1 1998-05-13 Tai-hwa Liang <avatar@FreeBSD.org>
     Key fingerprint = 5B 05 1D 37 7F 35 31 4E  5D 38 BD 07 10 32 B9 D0
uid                            Tai-hwa Liang <avatar@mmlab.cse.yzu.edu.tw>
```

D.3.98 廖英傑<ijliao@FreeBSD.org>

```
pub  1024D/11C02382 2001-01-09 Ying-Chieh Liao <ijliao@CCCA.NCTU.edu.tw>
     Key fingerprint = 4E98 55CC 2866 7A90 EFD7  9DA5 ACC6 0165 11C0 2382
uid                            Ying-Chieh Liao <ijliao@FreeBSD.org>
uid                            Ying-Chieh Liao <ijliao@csie.nctu.edu.tw>
uid                            Ying-Chieh Liao <ijliao@dragon2.net>
uid                            Ying-Chieh Liao <ijliao@tw.FreeBSD.org>
sub  4096g/C1E16E89 2001-01-09
```

D.3.99 林東毅<clive@FreeBSD.org>

```
pub  1024D/A008C03E 2001-07-30 Clive Lin <clive@tongi.org>
     Key fingerprint = FA3F 20B6 A77A 6CEC 1856  09B0 7455 2805 A008 C03E
uid                            Clive Lin <clive@CirX.ORG>
uid                            Clive Lin <clive@FreeBSD.org>
sub  1024g/03C2DC87 2001-07-30 [expires: 2005-08-25]
```

D.3.100 宋政隆<clsung@FreeBSD.org>

```
pub  1024D/956E8BC1 2003-09-12 Cheng-Lung Sung <clsung@FreeBSD.org>
     Key fingerprint = E0BC 57F9 F44B 46C6 DB53  8462 F807 89F3 956E 8BC1
uid                            Cheng-Lung Sung (Software Engineer) <clsung@dragon2.net>
uid                            Cheng-Lung Sung (Alumnus of CSIE, NCTU, Taiwan) <clsung@sungsung.c
uid                            Cheng-Lung Sung (AlanSung) <clsung@tiger2.net>
uid                            Cheng-Lung Sung (FreeBSD@Taiwan) <clsung@freebsd.csie.nctu.edu.tw>
uid                            Cheng-Lung Sung (Ph.D. Student of NTU.EECS) <d92921016@ntu.edu.tw>
uid                            Cheng-Lung Sung (FreeBSD Freshman) <clsung@tw.freebsd.org>
uid                            Cheng-Lung Sung (ports committer) <clsung@FreeBSD.org>
sub  1024g/1FB800C2 2003-09-12
```

D.3.101 Tilman Linneweh <arved@FreeBSD.org>

```
pub   1024D/807AC53A 2002-06-03 [expires: 2009-06-15]
      Key fingerprint = A92F 344F 31A8 B8DE DDFA  7FB4 7C22 C39F 807A C53A
uid                  Tilman Linneweh <e0025974@student.tuwien.ac.at>
uid                  Tilman Linneweh <arved@arved.at>
uid                  Tilman Linneweh <arved@FreeBSD.org>
uid                  Tilman Linneweh <arved@inso.tuwien.ac.at>
sub   1024g/FA351986 2002-06-03 [expires: 2009-06-15]
```

D.3.102 Remko Lodder <remko@FreeBSD.org>

```
pub   1024D/8F494B77 2004-09-03 [expires: 2009-06-25]
      Key fingerprint = 575D 8AD6 8646 E6D2 1226  0A8C D2A9 0DFF 8F49 4B77
uid                  Remko Lodder (Remko Lodder) <remko@elvandar.org>
uid                  Remko Lodder (my FreeBSD.org uid) <remko@FreeBSD.org>
sub   2048g/6BF55109 2006-02-25 [expires: 2008-02-25]
```

D.3.103 Scott Long <scottl@FreeBSD.org>

```
pub  1024D/017C5EBF 2003-01-18 Scott A. Long (This is my official FreeBSD key) <scottl@freebsd.org
     Key fingerprint = 34EA BD06 44F7 F8C3 22BC  B52C 1D3A F6D1 017C 5EBF
sub  1024g/F61C8F91 2003-01-18
```

D.3.104 Pav Lucistnik <pav@FreeBSD.org>

```
pub  1024D/C14EB282 2003-08-25 Pav Lucistnik <pav@FreeBSD.org>
     Key fingerprint = 2622 B7E3 7DA5 5C53 2079  855B 9ED7 583F C14E B282
uid                           Pav Lucistnik <pav@oook.cz>
sub  1024g/7287A947 2003-08-25
```

D.3.105 馬源浩<bmah@FreeBSD.org>

```
pub   1024D/5BA052C3 1997-12-08
      Key fingerprint = F829 B805 207D 14C7 7197  7832 D8CA 3171 5BA0 52C3
uid                  Bruce A. Mah <bmah@acm.org>
uid                  Bruce A. Mah <bmah@ca.sandia.gov>
uid                  Bruce A. Mah <bmah@ieee.org>
uid                  Bruce A. Mah <bmah@cisco.com>
uid                  Bruce A. Mah <bmah@employees.org>
uid                  Bruce A. Mah <bmah@freebsd.org>
uid                  Bruce A. Mah <bmah@packetdesign.com>
uid                  Bruce A. Mah <bmah@kitchenlab.org>
sub   2048g/B4E60EA1 1997-12-08
```

D.3.106 Mike Makonnen <mtm@FreeBSD.org>

```
pub  1024D/7CD41F55 2004-02-06 Michael Telahun Makonnen <mtm@FreeBSD.Org>
     Key fingerprint = AC7B 5672 2D11 F4D0 EBF8  5279 5359 2B82 7CD4 1F55
uid                            Michael Telahun Makonnen <mtm@tmsa-inc.com>
uid                            Mike Makonnen <mtm@identd.net>
uid                            Michael Telahun Makonnen <mtm@acs-et.com>
sub  2048g/E7DC936B 2004-02-06
```

D.3.107 David Malone <dwmalone@FreeBSD.org>

```
pub   512/40378991 1994/04/21 David Malone <dwmalone@maths.tcd.ie>
            Key fingerprint = 86 A7 F4 86 39 2C 47 2C  C1 C2 35 78 8E 2F B8 F5
```

D.3.108 Sergey Matveychuk <sem@FreeBSD.org>

```
pub   1024D/B71F605D 1999-10-13
      Key fingerprint = 4704 F374 DB28 BEC6 51C8  1322 4DC9 4BD8 B71F 605D
uid                  Sergey Matveychuk <sem@FreeBSD.org>
uid                  Sergey Matveychuk <sem@ciam.ru>
uid                  Sergey Matveychuk <sem@core.inec.ru>
sub   2048g/DEAF9D91 1999-10-13
```

D.3.109 Emanuel Haupt <ehaupt@FreeBSD.org>

```
pub   1024D/90215DB9 2007-02-06 [expires: 2008-02-06]
      Key fingerprint = 741B C70F 100B F360 0B52  E92D 5F01 7A86 9021 5DB9
uid                  Emanuel Haupt <ehaupt@FreeBSD.org>
uid                  Emanuel Haupt <ehaupt@critical.ch>
sub   2048g/6DD0929C 2007-02-06 [expires: 2008-02-06]
```

D.3.110 Koop Mast <kwm@FreeBSD.org>

```
pub  1024D/F95426DA 2004-09-10 Koop Mast <kwm@rainbow-runner.nl>
     Key fingerprint = C66F 1835 0548 3440 8576  0FFE 6879 B7CD F954 26DA
uid                            Koop Mast <kwm@FreeBSD.org>
sub  1024g/A782EEDD 2004-09-10
```

D.3.111 Makoto Matsushita <matusita@FreeBSD.org>

```
pub   1024D/20544576 1999-04-18
      Key fingerprint = 71B6 13BF B262 2DD8 2B7C  6CD0 EB2D 4147 2054 4576
uid                  Makoto Matsushita <matusita@matatabi.or.jp>
uid                  Makoto Matsushita <matusita@FreeBSD.org>
```

```
uid                  Makoto Matsushita <matusita@jp.FreeBSD.ORG>
uid                  Makoto Matsushita <matusita@ist.osaka-u.ac.jp>
sub   1024g/F1F3C94D 1999-04-18
```

D.3.112 Tom McLaughlin <tmclaugh@FreeBSD.org>

```
pub   1024D/E2F7B3D8 2005-05-24
      Key fingerprint = 7692 B222 8D23 CF94 1993  0138 E339 E225 E2F7 B3D8
uid                  Tom McLaughlin (Personal email address) <tmclaugh@sdf.lonestar.org>
uid                  Tom McLaughlin (Work email address) <tmclaughlin@meditech.com>
uid                  Tom McLaughlin (FreeBSD email address) <tmclaugh@FreeBSD.org>
sub   2048g/16838F62 2005-05-24
```

D.3.113 Kenneth D. Merry <ken@FreeBSD.org>

```
pub  1024D/54C745B5 2000-05-15 Kenneth D. Merry <ken@FreeBSD.org>
     Key fingerprint = D25E EBC5 F17A 9E52 84B4  BF14 9248 F0DA 54C7 45B5
uid                            Kenneth D. Merry <ken@kdm.org>
sub  2048g/89D0F797 2000-05-15

pub  1024R/2FA0A505 1995-10-30 Kenneth D. Merry <ken@plutotech.com>
     Key fingerprint = FD FA 85 85 95 C4 8E E8  98 1A CA 18 56 F0 00 1F
```

D.3.114 Dirk Meyer <dinoex@FreeBSD.org>

```
pub  1024R/331CDA5D 1995-06-04 Dirk Meyer <dinoex@FreeBSD.org>
     Key fingerprint = 44 16 EC 0A D3 3A 4F 28  8A 8A 47 93 F1 CF 2F 12
uid                            Dirk Meyer <dirk.meyer@dinoex.sub.org>
uid                            Dirk Meyer <dirk.meyer@guug.de>
```

D.3.115 Yoshiro Sanpei MIHIRA <sanpei@FreeBSD.org>

```
pub  1024R/391C5D69 1996-11-21 sanpei@SEAPLE.ICC.NE.JP
     Key fingerprint = EC 04 30 24 B0 6C 1E 63  5F 5D 25 59 3E 83 64 51
uid                            MIHIRA Yoshiro <sanpei@sanpei.org>
uid                            Yoshiro MIHIRA <sanpei@FreeBSD.org>
uid                            MIHIRA Yoshiro <sanpei@yy.cs.keio.ac.jp>
uid                            MIHIRA Yoshiro <sanpei@cc.keio.ac.jp>
uid                            MIHIRA Yoshiro <sanpei@educ.cc.keio.ac.jp>
uid                            MIHIRA Yoshiro <sanpei@st.keio.ac.jp>
```

D.3.116 Marcel Moolenaar <marcel@FreeBSD.org>

```
pub  1024D/61EE89F6 2002-02-09 Marcel Moolenaar <marcel@xcllnt.net>
     Key fingerprint = 68BB E2B7 49AA FF69 CA3A  DF71 A605 A52D 61EE 89F6
sub  1024g/6EAAB456 2002-02-09
```

D.3.117 Dmitry Morozovsky <marck@FreeBSD.org>

```
pub   1024D/6B691B03 2001-07-20
      Key fingerprint = 39AC E336 F03D C0F8 5305  B725 85D4 5045 6B69 1B03
uid                  Dmitry Morozovsky <marck@rinet.ru>
uid                  Dmitry Morozovsky <marck@FreeBSD.org>
sub   2048g/44D656F8 2001-07-20
```

D.3.118 Thomas Möstl <tmm@FreeBSD.org>

```
pub  1024D/419C776C 2000-11-28 Thomas Moestl <tmm@FreeBSD.org>
     Key fingerprint = 1C97 A604 2BD0 E492 51D0  9C0F 1FE6 4F1D 419C 776C
uid                            Thomas Moestl <tmoestl@gmx.net>
uid                            Thomas Moestl <t.moestl@tu-bs.de>
sub  2048g/ECE63CE6 2000-11-28
```

D.3.119 Rich Murphey <rich@FreeBSD.org>

```
pub  1024R/583443A9 1995-03-31 Rich Murphey <rich@lamprey.utmb.edu>
     Key fingerprint = AF A0 60 C4 84 D6 0C 73  D1 EF C0 E9 9D 21 DB E4
```

D.3.120 Akinori MUSHA <knu@FreeBSD.org>

```
pub  1024D/9FD9E1EE 2000-03-21 Akinori MUSHA <knu@and.or.jp>
     Key fingerprint = 081D 099C 1705 861D 4B70  B04A 920B EFC7 9FD9 E1EE
uid                            Akinori MUSHA <knu@FreeBSD.org>
uid                            Akinori MUSHA <knu@idaemons.org>
uid                            Akinori MUSHA <knu@ruby-lang.org>
sub  1024g/71BA9D45 2000-03-21
```

D.3.121 Masafumi NAKANE <max@FreeBSD.org>

```
pub  1024D/CE356B59 2000-02-19 Masafumi NAKANE <max@wide.ad.jp>
     Key fingerprint = EB40 BCAB 4CE5 0764 9942  378C 9596 159E CE35 6B59
uid                            Masafumi NAKANE <max@FreeBSD.org>
uid                            Masafumi NAKANE <max@accessibility.org>
uid                            Masafumi NAKANE <kd5pdi@qsl.net>
sub  1024g/FA9BD48B 2000-02-19
```

D.3.122 Yoichi NAKAYAMA <yoichi@FreeBSD.org>

```
pub  1024D/E0788E46 2000-12-28 Yoichi NAKAYAMA <yoichi@assist.media.nagoya-u.ac.jp>
     Key fingerprint = 1550 2662 46B3 096C 0460  BC03 800D 0C8A E078 8E46
uid                            Yoichi NAKAYAMA <yoichi@eken.phys.nagoya-u.ac.jp>
uid                            Yoichi NAKAYAMA <yoichi@FreeBSD.org>
sub  1024g/B987A394 2000-12-28
```

D.3.123 Alexander Nedotsukov <bland@FreeBSD.org>

```
pub  1024D/D004116C 2003-08-14 Alexander Nedotsukov <bland@FreeBSD.org>
     Key fingerprint = 35E2 5020 55FC 2071 4ADD  1A4A 86B6 8A5D D004 116C
sub  1024g/1CCA8D46 2003-08-14
```

D.3.124 Simon L. Nielsen <simon@FreeBSD.org>

```
pub   1024D/FF7490AB 2007-01-14
      Key fingerprint = 4E92 BA8D E45E 85E2 0380  B264 049C 7480 FF74 90AB
uid                  Simon L. Nielsen <simon@FreeBSD.org>
uid                  Simon L. Nielsen <simon@nitro.dk>
sub   2048g/E3F5A76E 2007-01-14
```

D.3.125 Anders Nordby <anders@FreeBSD.org>

```
pub  1024D/00835956 2000-08-13 Anders Nordby <anders@fix.no>
     Key fingerprint = 1E0F C53C D8DF 6A8F EAAD  19C5 D12A BC9F 0083 5956
uid                            Anders Nordby <anders@FreeBSD.org>
sub  2048g/4B160901 2000-08-13
```

D.3.126 David O'Brien <obrien@FreeBSD.org>

```
pub  1024R/34F9F9D5 1995-04-23 David E. O'Brien <defunct - obrien@Sea.Legent.com>
     Key fingerprint = B7 4D 3E E9 11 39 5F A3  90 76 5D 69 58 D9 98 7A
uid                            David E. O'Brien <obrien@NUXI.com>
uid                            deobrien@ucdavis.edu
uid                            David E. O'Brien <whois Do38>
uid                            David E. O'Brien <obrien@FreeBSD.org>
uid                            David E. O'Brien <dobrien@seas.gwu.edu>
uid                            David E. O'Brien <obrien@cs.ucdavis.edu>
uid                            David E. O'Brien <defunct - obrien@media.sra.com>
uid                            David E. O'Brien <obrien@elsewhere.roanoke.va.us>
uid                            David E. O'Brien <obrien@Nuxi.com>

pub  1024D/7F9A9BA2 1998-06-10 "David E. O'Brien" <obrien@cs.ucdavis.edu>
     Key fingerprint = 02FD 495F D03C 9AF2 5DB7  F496 6FC8 DABD 7F9A 9BA2
uid                            "David E. O'Brien" <obrien@NUXI.com>
```

```
uid                           "David E. O'Brien" <obrien@FreeBSD.org>
sub  3072g/BA32C20D 1998-06-10
```

D.3.127 Philip Paeps <philip@FreeBSD.org>

```
pub   4096R/C5D34D05 2006-10-22
      Key fingerprint = 356B AE02 4763 F739 2FA2  E438 2649 E628 C5D3 4D05
uid                  Philip Paeps <philip@paeps.cx>
uid                  Philip Paeps <philip@nixsys.be>
uid                  Philip Paeps <philip@fosdem.org>
uid                  Philip Paeps <philip@freebsd.org>
uid                  Philip Paeps <philip@pub.telenet.be>
sub   1024D/035EFC58 2006-10-22 [expires: 2008-10-21]
sub   2048g/6E5FD7D6 2006-10-22 [expires: 2008-11-17]
```

D.3.128 Hiten Pandya <hmp@FreeBSD.org>

```
pub  1024D/938CACA8 2004-02-13 Hiten Pandya (FreeBSD) <hmp@FreeBSD.org>
     Key fingerprint = 84EB C75E C75A 50ED 304E  E446 D974 7842 938C ACA8
uid                           Hiten Pandya <hmp@backplane.com>
sub  2048g/783874B5 2004-02-13
```

D.3.129 Mark Peek <mp@FreeBSD.org>

```
pub  1024D/330D4D01 2002-01-27 Mark Peek <mp@FreeBSD.org>
     Key fingerprint = 510C 96EE B4FB 1B0A 2CF8  A0AF 74B0 0B0E 330D 4D01
sub  1024g/9C6CAC09 2002-01-27
```

D.3.130 Peter Pentchev <roam@FreeBSD.org>

```
pub   1024D/16194553 2002-02-01
      Key fingerprint = FDBA FD79 C26F 3C51 C95E  DF9E ED18 B68D 1619 4553
uid                  Peter Pentchev <roam@ringlet.net>
uid                  Peter Pentchev <roam@cnsys.bg>
uid                  Peter Pentchev <roam@sbnd.net>
uid                  Peter Pentchev <roam@online.bg>
uid                  Peter Pentchev <roam@orbitel.bg>
uid                  Peter Pentchev <roam@FreeBSD.org>
uid                  Peter Pentchev <roam@techlab.office1.bg>
sub   1024g/7074473C 2002-02-01
```

D.3.131 Denis Peplin <den@FreeBSD.org>

```
pub  1024D/485DDDF5 2003-09-11 Denis Peplin <den@FreeBSD.org>
     Key fingerprint = 495D 158C 8EC9 C2C1 80F5  EA96 6F72 7C1C 485D DDF5
sub  1024g/E70BA158 2003-09-11
```

D.3.132 Gerald Pfeifer <gerald@FreeBSD.org>

```
pub  1024D/745C015A 1999-11-09 Gerald Pfeifer <gerald@pfeifer.com>
     Key fingerprint = B215 C163 3BCA 0477 615F  1B35 A5B3 A004 745C 015A
uid                            Gerald Pfeifer <Gerald.Pfeifer@vibe.at>
uid                            Gerald Pfeifer <pfeifer@dbai.tuwien.ac.at>
uid                            Gerald Pfeifer <gerald@pfeifer.at>
uid                            Gerald Pfeifer <gerald@FreeBSD.org>
sub  1536g/F0156927 1999-11-09
```

D.3.133 John Polstra <jdp@FreeBSD.org>

```
pub  1024R/BFBCF449 1997-02-14 John D. Polstra <jdp@polstra.com>
     Key fingerprint = 54 3A 90 59 6B A4 9D 61  BF 1D 03 09 35 8D F6 0D
```

D.3.134 Kirill Ponomarew <krion@FreeBSD.org>

```
pub   1024D/AEB426E5 2002-04-07
      Key fingerprint = 58E7 B953 57A2 D9DD 4960  2A2D 402D 46E9 AEB4 26E5
uid                  Kirill Ponomarew <krion@voodoo.bawue.com>
uid                  Kirill Ponomarew <krion@guug.de>
uid                  Kirill Ponomarew <krion@FreeBSD.org>
sub   1024D/05AC7CA0 2006-01-30 [expires: 2008-01-30]
sub   2048g/C3EE5537 2006-01-30 [expires: 2008-01-30]
```

D.3.135 Mark Pulford <markp@FreeBSD.org>

```
pub  1024D/182C368F 2000-05-10 Mark Pulford <markp@FreeBSD.org>
     Key fingerprint = 58C9 C9BF C758 D8D4 7022  8EF5 559F 7F7B 182C 368F
uid                            Mark Pulford <mark@kyne.com.au>
sub  2048g/380573E8 2000-05-10
```

D.3.136 Thomas Quinot <thomas@FreeBSD.org>

```
pub  1024D/393D2469 1999-09-23 Thomas Quinot <thomas@cuivre.fr.eu.org>
 Empreinte de la cl☸= 4737 A0AD E596 6D30 4356  29B8 004D 54B8 393D 2469
uid                            Thomas Quinot <thomas@debian.org>
uid                            Thomas Quinot <thomas@FreeBSD.org>
```

```
sub  1024g/8DE13BB2 1999-09-23
```

D.3.137 Herve Quiroz <hq@FreeBSD.org>

```
pub  1024D/85AC8A80 2004-07-22 Herve Quiroz <hq@FreeBSD.org>
     Key fingerprint = 14F5 BC56 D736 102D 41AF  A07B 1D97 CE6C 85AC 8A80
uid                            Herve Quiroz <herve.quiroz@esil.univ-mrs.fr>
sub  1024g/8ECCAFED 2004-07-22
```

D.3.138 Doug Rabson <dfr@FreeBSD.org>

```
pub   1024D/59F57821 2004-02-07
      Key fingerprint = 9451 C4FE 1A7E 117B B95F  1F8F B123 456E 59F5 7821
uid                  Doug Rabson <dfr@nlsystems.com>
sub   1024g/6207AA32 2004-02-07
```

D.3.139 Jim Rees <rees@FreeBSD.org>

```
pub   512/B623C791 1995/02/21 Jim Rees <rees@umich.edu>
          Key fingerprint =  02 5F 1B 15 B4 6E F1 3E  F1 C5 E0 1D EA CC 17 88
```

D.3.140 Tom Rhodes <trhodes@FreeBSD.org>

```
pub   1024D/FB7D88E1 2008-05-07
      Key fingerprint = 8279 3100 2DF2 F00E 7FDD  AC2C 5776 23AB FB7D 88E1
uid                  Tom Rhodes (trhodes) <trhodes@FreeBSD.org>
sub   4096g/7B0CD79F 2008-05-07
```

D.3.141 Benno Rice <benno@FreeBSD.org>

```
pub  1024D/87C59909 2002-01-16 Benno Rice <benno@FreeBSD.org>
     Key fingerprint = CE27 DADA 08E3 FAA3 88F1  5B31 5E34 705A 87C5 9909
uid                            Benno Rice <benno@jeamland.net>
sub  1024g/4F7C2BAD 2002-01-16 [expires: 2007-01-15]
```

D.3.142 Ollivier Robert <roberto@FreeBSD.org>

```
pub   1024D/7DCAE9D3 1997-08-21
      Key fingerprint = 2945 61E7 D4E5 1D32 C100  DBEC A04F FB1B 7DCA E9D3
uid                  Ollivier Robert <roberto@keltia.freenix.fr>
uid                  Ollivier Robert <roberto@FreeBSD.org>
sub   2048g/C267084D 1997-08-21
```

D.3.143 Craig Rodrigues <rodrigc@FreeBSD.org>

```
pub   1024D/3998479D 2005-05-20
      Key fingerprint = F01F EBE6 F5C8 6DC2 954F  098F D20A 8A2A 3998 479D
uid                  Craig Rodrigues <rodrigc@freebsd.org>
uid                  Craig Rodrigues <rodrigc@crodrigues.org>
sub   2048g/AA77E09B 2005-05-20
```

D.3.144 Guido van Rooij <guido@FreeBSD.org>

```
pub  1024R/599F323D 1996-05-18 Guido van Rooij <guido@gvr.org>
     Key fingerprint = 16 79 09 F3 C0 E4 28 A7  32 62 FA F6 60 31 C0 ED
uid                           Guido van Rooij <guido@gvr.win.tue.nl>

pub  1024D/A95102C1 2000-10-25 Guido van Rooij <guido@madison-gurkha.nl>
     Key fingerprint = 5B3E 51B7 0E7A D170 0574  1E51 2471 117F A951 02C1
uid                           Guido van Rooij <guido@madison-gurkha.com>
sub  1024g/A5F20553 2000-10-25
```

D.3.145 Niklas Saers <niklas@FreeBSD.org>

```
pub  1024D/C822A476 2004-03-09 Niklas Saers <niklas@saers.com>
     Key fingerprint = C41E F734 AF0E 3D21 7499  9EB1 9A31 2E7E C822 A476
sub  1024g/81E2FF36 2004-03-09
```

D.3.146 Mark Santcroos <marks@FreeBSD.org>

```
pub   1024D/DBE7EB8E 2005-03-08
      Key fingerprint = C0F0 44F3 3F15 520F 6E32  186B BE0A BA42 DBE7 EB8E
uid                  Mark Santcroos <marks@ripe.net>
uid                  Mark Santcroos <mark@santcroos.net>
uid                  Mark Santcroos <marks@freebsd.org>
sub   2048g/FFF80F85 2005-03-08
```

D.3.147 Hiroki Sato <hrs@FreeBSD.org>

```
pub   1024D/2793CF2D 2001-06-12
      Key fingerprint = BDB3 443F A5DD B3D0 A530  FFD7 4F2C D3D8 2793 CF2D
uid                  Hiroki Sato <hrs@allbsd.org>
uid                  Hiroki Sato <hrs@eos.ocn.ne.jp>
uid                  Hiroki Sato <hrs@ring.gr.jp>
uid                  Hiroki Sato <hrs@FreeBSD.org>
uid                  Hiroki Sato <hrs@jp.FreeBSD.org>
uid                  Hiroki Sato <hrs@vlsi.ee.noda.tus.ac.jp>
uid                  Hiroki Sato <hrs@jp.NetBSD.org>
uid                  Hiroki Sato <hrs@NetBSD.org>
```

```
sub   1024g/8CD251FF 2001-06-12
```

D.3.148 Wolfram Schneider <wosch@FreeBSD.org>

```
Type Bits/KeyID    Date       User ID
pub  1024/2B7181AD 1997/08/09 Wolfram Schneider <wosch@FreeBSD.org>
            Key fingerprint = CA 16 91 D9 75 33 F1 07  1B F0 B4 9F 3E 95 B6 09
```

D.3.149 David Schultz <das@FreeBSD.org>

```
pub  1024D/BE848B57 2001-07-19 David Schultz <das@FreeBSD.ORG>
     Key fingerprint = 0C12 797B A9CB 19D9 FDAF  2A39 2D76 A2DB BE84 8B57
uid  David Schultz <dschultz@uclink.Berkeley.EDU>
uid  David Schultz <das@FreeBSD.ORG>
sub  2048g/69206E8E 2001-07-19
```

D.3.150 Jens Schweikhardt <schweikh@FreeBSD.org>

```
pub  1024D/0FF231FD 2002-01-27 Jens Schweikhardt <schweikh@FreeBSD.org>
     Key fingerprint = 3F35 E705 F02F 35A1 A23E  330E 16FE EA33 0FF2 31FD
uid                            Jens Schweikhardt <schweikh@schweikhardt.net>
sub  1024g/6E93CACC 2002-01-27 [expires: 2005-01-26]
```

D.3.151 Gregory Neil Shapiro <gshapiro@FreeBSD.org>

```
pub  1024R/4FBE2ADD 2000-10-13 Gregory Neil Shapiro <gshapiro@gshapiro.net>
     Key fingerprint = 56 D5 FF A7 A6 54 A6 B5  59 10 00 B9 5F 5F 20 09
uid                            Gregory Neil Shapiro <gshapiro@FreeBSD.org>

pub  1024D/F76A9BF5 2001-11-14 Gregory Neil Shapiro <gshapiro@FreeBSD.org>
     Key fingerprint = 3B5E DAF1 4B04 97BA EE20  F841 21F9 C5BC F76A 9BF5
uid                            Gregory Neil Shapiro <gshapiro@gshapiro.net>
sub  2048g/935657DC 2001-11-14

pub  1024D/FCE56561 2000-10-14 Gregory Neil Shapiro <gshapiro@FreeBSD.org>
     Key fingerprint = 42C4 A87A FD85 C34F E77F  5EA1 88E1 7B1D FCE5 6561
uid                            Gregory Neil Shapiro <gshapiro@gshapiro.net>
sub  1024g/285DC8A0 2000-10-14 [expires: 2001-10-14]
```

D.3.152 Arun Sharma <arun@FreeBSD.org>

```
pub  1024D/7D112181 2003-03-06 Arun Sharma <arun@sharma-home.net>
     Key fingerprint = A074 41D6 8537 C7D5 070E  0F78 0247 1AE2 7D11 2181
uid                            Arun Sharma <arun@freebsd.org>
```

```
uid                           Arun Sharma <arun.sharma@intel.com>
sub  1024g/ACAD98DA 2003-03-06 [expires: 2005-03-05]
```

D.3.153 Norikatsu Shigemura <nork@FreeBSD.org>

```
pub   1024D/7104EA4E 2005-02-14
      Key fingerprint = 9580 60A3 B58A 0864 79CB  779A 6FAE 229B 7104 EA4E
uid                  Norikatsu Shigemura <nork@cityfujisawa.ne.jp>
uid                  Norikatsu Shigemura <nork@ninth-nine.com>
uid                  Norikatsu Shigemura <nork@FreeBSD.org>
sub   4096g/EF56997E 2005-02-14
```

D.3.154 徐三泰<vanilla@FreeBSD.org>

```
pub  1024D/ACE75853 2001-11-20 Vanilla I. Shu <vanilla@FreeBSD.org>
     Key fingerprint = 290F 9DB8 42A3 6257 5D9A  5585 B25A 909E ACE7 5853
sub  1024g/CE695D0E 2001-11-20
```

D.3.155 Dmitry Sivachenko <demon@FreeBSD.org>

```
pub  1024D/13D5DF80 2002-03-18 Dmitry Sivachenko <mitya@cavia.pp.ru>
     Key fingerprint = 72A9 12C9 BB02 46D4 4B13  E5FE 1194 9963 13D5 DF80
uid                            Dmitry S. Sivachenko <demon@FreeBSD.org>
sub  1024g/060F6DBD 2002-03-18
```

D.3.156 Jesper Skriver <jesper@FreeBSD.org>

```
pub  1024D/F9561C31 2001-03-09 Jesper Skriver <jesper@FreeBSD.org>
     Key fingerprint = 6B88 9CE8 66E9 E631 C9C5  5EB4 22AB F0EC F956 1C31
uid                            Jesper Skriver <jesper@skriver.dk>
uid                            Jesper Skriver <jesper@wheel.dk>
sub  1024g/777C378C 2001-03-09
```

D.3.157 Ville Skyttä <scop@FreeBSD.org>

```
pub  1024D/BCD241CB 2002-04-07 Ville Skytt☸<ville.skytta@iki.fi>
     Key fingerprint = 4E0D EBAB 3106 F1FA 3FA9  B875 D98C D635 BCD2 41CB
uid                            Ville Skytt☸<ville.skytta@xemacs.org>
uid                            Ville Skytt☸<scop@FreeBSD.org>
sub  2048g/9426F4D1 2002-04-07
```

D.3.158 Andrey Slusar <anray@FreeBSD.org>

```
pub   1024D/AE7B5418 2005-12-12
      Key fingerprint = DE70 C24B 55A0 4A06 68A1  D425 3C59 9A9B AE7B 5418
uid                  Andrey Slusar <anray@ext.by>
uid                  Andrey Slusar <anrays@gmail.com>
uid                  Andrey Slusar <anray@FreeBSD.org>
sub   2048g/7D0EB77D 2005-12-12
```

D.3.159 Gleb Smirnoff <glebius@FreeBSD.org>

```
pub   1024D/1949DC80 2003-08-25
      Key fingerprint = 872C E14A 2F03 A3E8 D882  026E 5DE4 D7FE 1949 DC80
uid                  Gleb Smirnoff <glebius@FreeBSD.org>
uid                  Gleb Smirnoff <glebius@cell.sick.ru>
uid                  Gleb Smirnoff <glebius@bestcom.ru>
uid                  Gleb Smirnoff <glebius@rambler-co.ru>
uid                  Gleb Smirnoff <glebius@freebsd.org>
uid                  Gleb Smirnoff <glebius@freebsd.int.ru>
sub   1024g/A05118BD 2003-08-25
```

D.3.160 Ken Smith <kensmith@FreeBSD.org>

```
pub  1024D/29AEA7F6 2003-12-02 Ken Smith <kensmith@cse.buffalo.edu>
     Key fingerprint = 4AB7 D302 0753 8215 31E7  F1AD FC6D 7855 29AE A7F6
uid                            Ken Smith <kensmith@freebsd.org>
sub  1024g/0D509C6C 2003-12-02
```

D.3.161 Ben Smithurst <ben@FreeBSD.org>

```
pub  1024D/2CEF442C 2001-07-11 Ben Smithurst <ben@LSRfm.com>
     Key fingerprint = 355D 0FFF B83A 90A9 D648  E409 6CFC C9FB 2CEF 442C
uid                            Ben Smithurst <ben@vinosystems.com>
uid                            Ben Smithurst <ben@smithurst.org>
uid                            Ben Smithurst <ben@FreeBSD.org>
uid                            Ben Smithurst <csxbcs@comp.leeds.ac.uk>
uid                            Ben Smithurst <ben@scientia.demon.co.uk>
sub  1024g/347071FF 2001-07-11
```

D.3.162 Dag-Erling C. Smørgrav <des@FreeBSD.org>

```
pub   1024D/64EBE220 2006-11-11 [expires: 2009-11-10]
      Key fingerprint = 3A1C 8E68 952C 3305 6984  6486 30D4 3A6E 64EB E220
uid                  Dag-Erling Sm鵜grav <des@des.no>
uid                  Dag-Erling Sm鵜grav <des@freebsd.org>
uid                  [jpeg image of size 3315]
```

D.3.163 Maxim Sobolev <sobomax@FreeBSD.org>

```
pub  1024D/888205AF 2001-11-21 Maxim Sobolev <sobomax@FreeBSD.org>
     Key fingerprint = 85C9 DCB0 6828 087C C977  3034 A0DB B9B7 8882 05AF
uid                            Maxim Sobolev <sobomax@mail.ru>
uid                            Maxim Sobolev <sobomax@altavista.net>
uid                            Maxim Sobolev <vegacap@i.com.ua>

pub  1024D/468EE6D8 2003-03-21 Maxim Sobolev <sobomax@portaone.com>
     Key fingerprint = 711B D315 3360 A58F 9A0E  89DB 6D40 2558 468E E6D8
uid                            Maxim Sobolev <sobomax@FreeBSD.org>
uid                            Maxim Sobolev <sobomax@mail.ru>
uid                            Maxim Sobolev <vegacap@i.com.ua>

pub  1024D/6BEC980A 2004-02-13 Maxim Sobolev <sobomax@portaone.com>
     Key fingerprint = 09D5 47B4 8D23 626F B643  76EB DFEE 3794 6BEC 980A
uid                            Maxim Sobolev <sobomax@FreeBSD.org>
uid                            Maksym Sobolyev (It’s how they call me in official documents. Pret
uid                            Maksym Sobolyev (It’s how they call me in official documents. Pret
sub  2048g/16D049AB 2004-02-13 [expires: 2005-02-12]
```

D.3.164 Brian Somers <brian@FreeBSD.org>

```
pub  1024R/666A7421 1997-04-30 Brian Somers <brian@freebsd-services.com>
     Key fingerprint = 2D 91 BD C2 94 2C 46 8F  8F 09 C4 FC AD 12 3B 21
uid                            Brian Somers <brian@awfulhak.org>
uid                            Brian Somers <brian@FreeBSD.org>
uid                            Brian Somers <brian@OpenBSD.org>
uid                            Brian Somers <brian@uk.FreeBSD.org>
uid                            Brian Somers <brian@uk.OpenBSD.org>
```

D.3.165 Nicolas Souchu <nsouch@FreeBSD.org>

```
pub  1024D/C744F18B 2002-02-13 Nicholas Souchu <nsouch@freebsd.org>
     Key fingerprint = 992A 144F AC0F 40BA 55AE  DE6D 752D 0A6C C744 F18B
sub  1024g/90BD3231 2002-02-13
```

D.3.166 Suleiman Souhlal <ssouhlal@FreeBSD.org>

```
pub  1024D/2EA50469 2004-07-24 Suleiman Souhlal <ssouhlal@FreeBSD.org>
     Key fingerprint = DACF 89DB 54C7 DA1D 37AF  9A94 EB55 E272 2EA5 0469
sub  2048g/0CDCC535 2004-07-24
```

D.3.167 Vsevolod Stakhov <vsevolod@FreeBSD.org>

```
pub   1024D/213D0033 2005-03-14 [expires: 2008-03-13]
      Key fingerprint = B852 0010 761E 944A C76D  D447 A25D C12C 213D 0033
uid                  Vsevolod Stakhov <vsevolod@FreeBSD.org>
uid                  Vsevolod Stakhov <cebka@jet.msk.su>
uid                  Vsevolod Stakhov <vsevolod@highsecure.ru>
sub   2048g/786F2187 2005-03-14 [expires: 2008-03-13]
```

D.3.168 Volker Stolz <vs@FreeBSD.org>

```
pub  1024R/3FD1B6B5 1998-06-16 Volker Stolz <vs@freebsd.org>
     Key fingerprint = 69 6F BD A0 2E FE 19 66  CF B9 68 6E 41 7D F9 B9
uid                            Volker Stolz <stolz@i2.informatik.rwth-aachen.de> (LSK)
uid                            Volker Stolz <vs@foldr.org>
```

D.3.169 Gregory Sutter <gsutter@FreeBSD.org>

```
pub  1024D/845DFEDD 2000-10-10 Gregory S. Sutter <gsutter@zer0.org>
     Key fingerprint = D161 E4EA 4BFA 2427 F3F9  5B1F 2015 31D5 845D FEDD
uid                            Gregory S. Sutter <gsutter@freebsd.org>
uid                            Gregory S. Sutter <gsutter@daemonnews.org>
uid                            Gregory S. Sutter <gsutter@pobox.com>
sub  2048g/0A37BBCE 2000-10-10
```

D.3.170 Koichi Suzuki <metal@FreeBSD.org>

```
pub  1024D/AE562682 2004-05-23 SUZUKI Koichi <metal@FreeBSD.org>
     Key fingerprint = 92B9 A202 B5AB 8CB6 89FC  6DD1 5737 C702 AE56 2682
sub  4096g/730E604B 2004-05-23
```

D.3.171 Gary W. Swearingen <garys@FreeBSD.org>

```
pub   1024D/FAA48AD5 2005-08-22 [expires: 2007-08-22]
      Key fingerprint = 8292 CC3E 81B5 E54F E3DD  F987 FA52 E643 FAA4 8AD5
uid                  Gary W. Swearingen <garys@freebsd.org>
sub   2048g/E34C3CA0 2005-08-22 [expires: 2007-08-22]
```

D.3.172 Yoshihiro Takahashi <nyan@FreeBSD.org>

```
pub  1024D/8394B81F 2001-10-15 Yoshihiro TAKAHASHI <nyan@jp.FreeBSD.org>
     Key fingerprint = D4FA D8CA 2AED FCF4 90A3  3569 8666 0500 8394 B81F
uid                            Yoshihiro TAKAHASHI <nyan@furiru.org>
uid                            Yoshihiro TAKAHASHI <nyan@FreeBSD.org>
```

```
sub  1024g/B796F020 2001-10-15
```

D.3.173 Mikhail Teterin <mi@FreeBSD.org>

```
pub  1024R/3FC71479 1995-09-08 Mikhail Teterin <mi@aldan.star89.galstar.com>
     Key fingerprint = 5F 15 EA 78 A5 40 6A 0F  14 D7 D9 EA 6E 2B DA A4
```

D.3.174 Gordon Tetlow <gordon@FreeBSD.org>

```
pub  1024D/357D65FB 2002-05-14 Gordon Tetlow <gordont@gnf.org>
     Key fingerprint = 34EF AD12 10AF 560E C3AE  CE55 46ED ADF4 357D 65FB
uid                            Gordon Tetlow <gordon@FreeBSD.org>
sub  1024g/243694AB 2002-05-14
```

D.3.175 Lars Thegler <lth@FreeBSD.org>

```
pub  1024D/56B0CA08 2004-05-31 Lars Thegler <lth@FreeBSD.org>
     Key fingerprint = ABAE F98C EA78 1C8D 6FDD  CB27 1CA9 5A63 56B0 CA08
uid                            Lars Thegler <lars@thegler.dk>
sub  1024g/E8C58EF3 2004-05-31
```

D.3.176 Thierry Thomas <thierry@FreeBSD.org>

```
pub   1024D/C71405A2 1997-10-11
      Key fingerprint = 3BB8 F358 C2F1 776C 65C9  AE51 73DE 698C C714 05A2
uid                  Thierry Thomas <thierry@pompo.net>
uid                  Thierry Thomas <tthomas@mail.dotcom.fr>
uid                  Thierry Thomas (FreeBSD committer) <thierry@FreeBSD.org>
sub   1024R/C5529925 2003-11-26
sub   2048g/05CF3992 2008-02-05
```

D.3.177 Andrew Thompson <thompsa@FreeBSD.org>

```
pub   1024D/BC6B839B 2005-05-05
      Key fingerprint = DE74 3F49 B97C A170 C8F1  8423 CAB6 9D57 BC6B 839B
uid                  Andrew Thompson <thompsa@freebsd.org>
uid                  Andrew Thompson <andy@fud.org.nz>
sub   2048g/92E370FB 2005-05-05
```

D.3.178 Florent Thoumie <flz@FreeBSD.org>

```
pub   1024D/5147DCF4 2004-12-04
      Key fingerprint = D203 AF5F F31A 63E2 BFD5  742B 3311 246D 5147 DCF4
uid                  Florent Thoumie (FreeBSD committer address) <flz@FreeBSD.org>
uid                  Florent Thoumie (flz) <florent@thoumie.net>
uid                  Florent Thoumie (flz) <flz@xbsd.org>
uid                  [jpeg image of size 1796]
sub   2048g/15D930B9 2004-12-04
```

D.3.179 Hajimu UMEMOTO <ume@FreeBSD.org>

```
pub   1024D/BF9071FE 2005-03-17
      Key fingerprint = 1F00 0B9E 2164 70FC 6DC5  BF5F 04E9 F086 BF90 71FE
uid                  Hajimu UMEMOTO <ume@mahoroba.org>
uid                  Hajimu UMEMOTO <ume@FreeBSD.org>
uid                  Hajimu UMEMOTO <ume@jp.FreeBSD.org>
sub   2048g/748DB3B0 2005-03-17
```

D.3.180 Stephan Uphoff <ups@FreeBSD.org>

```
pub  2048R/D684B04A 2004-10-06 Stephan Uphoff <ups@freebsd.org>
     Key fingerprint = B5D2 04AE CA8F 7055 7474  3C85 F908 7F55 D684 B04A
uid                            Stephan Uphoff <ups@tree.com>
sub  2048R/A15F921B 2004-10-06
```

D.3.181 Jacques Vidrine <nectar@FreeBSD.org>

```
pub  2048R/33C1627B 2001-07-05 Jacques A. Vidrine <nectar@celabo.org>
     Key fingerprint = CB CE 7D A0 6E 01 DC 61  E5 91 0A BE 79 17 D3 82
uid                            Jacques A. Vidrine <jvidrine@verio.net>
uid                            Jacques A. Vidrine <n@nectar.com>
uid                            Jacques A. Vidrine <jacques@vidrine.cc>
uid                            Jacques A. Vidrine <nectar@FreeBSD.org>
uid                            Jacques A. Vidrine <n@nectar.cc>

pub  1024D/1606DB95 2001-07-05 Jacques A. Vidrine <nectar@celabo.org>
     Key fingerprint = 46BC EA5B F70A CC81 5332  0832 8C32 8CFF 1606 DB95
uid                            Jacques A. Vidrine <jvidrine@verio.net>
uid                            Jacques A. Vidrine <n@nectar.com>
uid                            Jacques A. Vidrine <jacques@vidrine.cc>
uid                            Jacques A. Vidrine <nectar@FreeBSD.org>
uid                            Jacques A. Vidrine <n@nectar.cc>
sub  2048g/57EDEA6F 2001-07-05
```

D.3.182 Adam Weinberger <adamw@FreeBSD.org>

```
pub  1024D/42C743FD 2002-10-12 Adam Weinberger <adam@vectors.cx>
     Key fingerprint = A980 3F2E 80A8 9619 9D1C  82E8 A3C2 8CD9 42C7 43FD
sub  1024g/15D67628 2002-10-12
```

D.3.183 Nate Williams <nate@FreeBSD.org>

```
pub  1024D/C2AC6BA4 2002-01-28 Nate Williams (FreeBSD) <nate@FreeBSD.org>
     Key fingerprint = 8EE8 5E72 8A94 51FA EA68  E001 FFF9 8AA9 C2AC 6BA4
sub  1024g/03EE46D2 2002-01-28
```

D.3.184 Garrett Wollman <wollman@FreeBSD.org>

```
pub  1024D/0B92FAEA 2000-01-20 Garrett Wollman <wollman@FreeBSD.org>
     Key fingerprint = 4627 19AF 4649 31BF DE2E  3C66 3ECF 741B 0B92 FAEA
sub  1024g/90D5EBC2 2000-01-20
```

D.3.185 Jörg Wunsch <joerg@FreeBSD.org>

```
pub  1024D/69A85873 2001-12-11 Joerg Wunsch <j@uriah.heep.sax.de>
     Key fingerprint = 5E84 F980 C3CA FD4B B584  1070 F48C A81B 69A8 5873
pub  1024D/69A85873 2001-12-11 Joerg Wunsch <j@uriah.heep.sax.de>
uid                            Joerg Wunsch <joerg_wunsch@interface-systems.de>
uid                            Joerg Wunsch <joerg@FreeBSD.org>
uid                            Joerg Wunsch <j@ida.interface-business.de>
sub  1024g/21DC9924 2001-12-11
```

D.3.186 Maksim Yevmenkin <emax@FreeBSD.org>

```
pub  1024D/F050D2DD 2003-10-01 Maksim Yevmenkin <m_evmenkin@yahoo.com>
     Key fingerprint = 8F3F D359 E318 5641 8C81  34AD 791D 53F5 F050 D2DD
```

D.3.187 Bjoern A. Zeeb <bz@FreeBSD.org>

```
pub   1024D/3CCF1842 2007-02-20
      Key fingerprint = 1400 3F19 8FEF A3E7 7207  EE8D 2B58 B8F8 3CCF 1842
uid                  Bjoern A. Zeeb <bz@zabbadoz.net>
uid                  Bjoern A. Zeeb <bzeeb@zabbadoz.net>
uid                  Bjoern A. Zeeb <bz@FreeBSD.org>
uid                  Bjoern A. Zeeb <bzeeb-lists@lists.zabbadoz.net>
sub   4096g/F36BDC5D 2007-02-20
```

D.3.188 Alexey Zelkin <phantom@FreeBSD.org>

```
pub  1024D/9196B7D9 2002-01-28 Alexey Zelkin <phantom@FreeBSD.org>
     Key fingerprint = 4465 F2A4 28C1 C2E4 BB95  1EA0 C70D 4964 9196 B7D9
sub  1024g/E590ABA4 2002-01-28
```

FreeBSD Glossary

This glossary contains terms and acronyms used within the FreeBSD community and documentation.

A

ACL

See: Access Control List

ACPI

See: Advanced Configuration and Power Interface

AMD

See: Automatic Mount Daemon

AML

See: ACPI Machine Language

API

See: Application Programming Interface

APIC

See: Advanced Programmable Interrupt Controller

APM

See: Advanced Power Management

APOP

See: Authenticated Post Office Protocol

ASL

See: ACPI Source Language

ATA

See: Advanced Technology Attachment

ATM

See: Asynchronous Transfer Mode

ACPI Machine Language

Pseudocode, interpreted by a virtual machine within an ACPI-compliant operating system, providing a layer between the underlying hardware and the documented interface presented to the OS.

ACPI Source Language

The programming language AML is written in.

Access Control List

A list of permissions attached to an object, usually either a file or a network device.

Advanced Configuration and Power Interface

A specification which provides an abstraction of the interface the hardware presents to the operating system, so that the operating system should need to know nothing about the underlying hardware to make the most of it. ACPI evolves and supercedes the functionality provided previously by APM, PNPBIOS and other technologies, and provides facilities for controlling power consumption, machine suspension, device enabling and disabling, etc.

Application Programming Interface

A set of procedures, protocols and tools that specify the canonical interaction of one or more program parts; how, when and why they do work together, and what data they share or operate on.

Advanced Power Management

An API enabling the operating system to work in conjunction with the BIOS in order to achieve power management. APM has been superseded by the much more generic and powerful ACPI specification for most applications.

Advanced Programmable Interrupt Controller

Advanced Technology Attachment

Asynchronous Transfer Mode

Authenticated Post Office Protocol

Automatic Mount Daemon

A daemon that automatically mounts a filesystem when a file or directory within that filesystem is accessed.

B

BAR

See: Base Address Register

BIND

See: Berkeley Internet Name Domain

BIOS

See: Basic Input/Output System

BSD

See: Berkeley Software Distribution

Base Address Register

The registers that determine which address range a PCI device will respond to.

Basic Input/Output System

The definition of BIOS depends a bit on the context. Some people refer to it as the ROM chip with a basic set of routines to provide an interface between software and hardware. Others refer to it as the set of routines contained in the chip that help in bootstrapping the system. Some might also refer to it as the screen used to configure the boostrapping process. The BIOS is PC-specific but other systems have something similar.

Berkeley Internet Name Domain

An implementation of the DNS protocols.

Berkeley Software Distribution

This is the name that the Computer Systems Research Group (CSRG) at The University of California at Berkeley (http://www.berkeley.edu) gave to their improvements and modifications to AT&T's 32V UNIX. FreeBSD is a descendant of the CSRG work.

Bikeshed Building

A phenomenon whereby many people will give an opinion on an uncomplicated topic, whilst a complex topic receives little or no discussion. See the FAQ (http://www.FreeBSD.org/doc/zh_TW.Big5/books/faq/misc.html#BIKESHED-PAINTING) for the origin of the term.

C

CD

See: Carrier Detect

CHAP

See: Challenge Handshake Authentication Protocol

CLIP

See: Classical IP over ATM

COFF

See: Common Object File Format

CPU

See: Central Processing Unit

CTS

See: Clear To Send

CVS

See: Concurrent Versions System

Carrier Detect

An RS232C signal indicating that a carrier has been detected.

Central Processing Unit

Also known as the processor. This is the brain of the computer where all calculations take place. There are a number of different architectures with different instruction sets. Among the more well-known are the Intel-x86 and derivatives, Sun SPARC, PowerPC, and Alpha.

Challenge Handshake Authentication Protocol

A method of authenticating a user, based on a secret shared between client and server.

Classical IP over ATM

Clear To Send

An RS232C signal giving the remote system permission to send data.

See Also: Request To Send.

Common Object File Format

Concurrent Versions System

A version control system, providing a method of working with and keeping track of many different revisions of files. CVS provides the ability to extract, merge and revert individual changes or sets of changes, and offers the ability to keep track of which changes were made, by who and for what reason.

D

DAC

See: Discretionary Access Control

DDB

See: Debugger

DES

See: Data Encryption Standard

DHCP

See: Dynamic Host Configuration Protocol

DNS

See: Domain Name System

DSDT

See: Differentiated System Description Table

DSR

See: Data Set Ready

DTR

See: Data Terminal Ready

DVMRP

See: Distance-Vector Multicast Routing Protocol

Discretionary Access Control

Data Encryption Standard

A method of encrypting information, traditionally used as the method of encryption for UNIX passwords and the crypt(3) function.

Data Set Ready

An RS232C signal sent from the modem to the computer or terminal indicating a readiness to send and receive data.

See Also: Data Terminal Ready.

Data Terminal Ready

An RS232C signal sent from the computer or terminal to the modem indicating a readiness to send and receive data.

Debugger

An interactive in-kernel facility for examining the status of a system, often used after a system has crashed to establish the events surrounding the failure.

Differentiated System Description Table

An ACPI table, supplying basic configuration information about the base system.

Distance-Vector Multicast Routing Protocol

Domain Name System

The system that converts humanly readable hostnames (i.e., mail.example.net) to Internet addresses and vice versa.

Dynamic Host Configuration Protocol

A protocol that dynamically assigns IP addresses to a computer (host) when it requests one from the server. The address assignment is called a “lease”.

E

ECOFF

See: Extended COFF

ELF

See: Executable and Linking Format

ESP

See: Encapsulated Security Payload

Encapsulated Security Payload

Executable and Linking Format

Extended COFF

F

FADT

See: Fixed ACPI Description Table

FAT

See: File Allocation Table

FAT16

See: File Allocation Table (16-bit)

FTP

See: File Transfer Protocol

File Allocation Table

File Allocation Table (16-bit)

File Transfer Protocol

A member of the family of high-level protocols implemented on top of TCP which can be used to transfer files over a TCP/IP network.

Fixed ACPI Description Table

G

GUI

See: Graphical User Interface

Giant

The name of a mutual exclusion mechanism (a `sleep mutex`) that protects a large set of kernel resources. Although a simple locking mechanism was adequate in the days where a machine might have only a few dozen processes, one networking card, and certainly only one processor, in current times it is an unacceptable performance bottleneck. FreeBSD developers are actively working to replace it with locks that protect individual resources, which will allow a much greater degree of parallelism for both single-processor and multi-processor machines.

Graphical User Interface

A system where the user and computer interact with graphics.

H

HTML

See: HyperText Markup Language

HUP

See: HangUp

HangUp

HyperText Markup Language

The markup language used to create web pages.

I

I/O

See: Input/Output

IASL

See: Intel's ASL compiler

IMAP

See: Internet Message Access Protocol

IP

See: Internet Protocol

IPFW

See: IP Firewall

IPP

See: Internet Printing Protocol

IPv4

See: IP Version 4

IPv6

See: IP Version 6

ISP

See: Internet Service Provider

IP Firewall

IP Version 4

The IP protocol version 4, which uses 32 bits for addressing. This version is still the most widely used, but it is slowly being replaced with IPv6.

See Also: IP Version 6.

IP Version 6

The new IP protocol. Invented because the address space in IPv4 is running out. Uses 128 bits for addressing.

Input/Output

Intel's ASL compiler

Intel's compiler for converting ASL into AML.

Internet Message Access Protocol

A protocol for accessing email messages on a mail server, characterised by the messages usually being kept on the server as opposed to being downloaded to the mail reader client.

See Also: Post Office Protocol Version 3.

Internet Printing Protocol

Internet Protocol

The packet transmitting protocol that is the basic protocol on the Internet. Originally developed at the U.S. Department of Defense and an extremly important part of the TCP/IP stack. Without the Internet Protocol, the Internet would not have become what it is today. For more information, see RFC 791 (ftp://ftp.rfc-editor.org/in-notes/rfc791.txt).

Internet Service Provider

A company that provides access to the Internet.

K

KAME

Japanese for "turtle", the term KAME is used in computing circles to refer to the KAME Project (http://www.kame.net/), who work on an implementation of IPv6.

KDC

See: Key Distribution Center

KLD

See: Kernel ld(1)

KSE

See: Kernel Scheduler Entities

KVA

See: Kernel Virtual Address

Kbps

See: Kilo Bits Per Second

Kernel ld(1)

A method of dynamically loading functionality into a FreeBSD kernel without rebooting the system.

Kernel Scheduler Entities

A kernel-supported threading system. See the project home page (http://www.FreeBSD.org/kse) for further details.

Kernel Virtual Address

Key Distribution Center

Kilo Bits Per Second

Used to measure bandwith (how much data can pass a given point at a specified amount of time). Alternates to the Kilo prefix include Mega, Giga, Tera, and so forth.

L

LAN

See: Local Area Network

LOR

See: Lock Order Reversal

LPD

See: Line Printer Daemon

Line Printer Daemon

Local Area Network

A network used on a local area, e.g. office, home, or so forth.

Lock Order Reversal

The FreeBSD kernel uses a number of resource locks to arbitrate contention for those resources. A run-time lock diagnostic system found in FreeBSD-CURRENT kernels (but removed for releases), called witness(4), detects the potential for deadlocks due to locking errors. (witness(4) is actually slightly conservative, so it is possible to get false positives.) A true positive report indicates that "if you were unlucky, a deadlock would have happened here".

True positive LORs tend to get fixed quickly, so check http://lists.FreeBSD.org/mailman/listinfo/freebsd-current and the LORs Seen (http://sources.zabbadoz.net/freebsd/lor.html) page before posting to the mailing lists.

M

MAC

See: Mandatory Access Control

MADT

See: Multiple APIC Description Table

MFC

See: Merge From Current

MFP4

See: Merge From Perforce

MFS

See: Merge From Stable

MIT

See: Massachusetts Institute of Technology

MLS

See: Multi-Level Security

MOTD

See: Message Of The Day

MTA

See: Mail Transfer Agent

MUA

See: Mail User Agent

Mail Transfer Agent

An application used to transfer email. An MTA has traditionally been part of the BSD base system. Today Sendmail is included in the base system, but there are many other MTAs, such as postfix, qmail and Exim.

Mail User Agent

An application used by users to display and write email.

Mandatory Access Control

Massachusetts Institute of Technology

Merge From Current

To merge functionality or a patch from the -CURRENT branch to another, most often -STABLE.

Merge From Perforce

To merge functionality or a patch from the Perforce repository to the -CURRENT branch.

See Also: Perforce.

Merge From Stable

In the normal course of FreeBSD development, a change will be committed to the -CURRENT branch for testing before being merged to -STABLE. On rare occasions, a change will go into -STABLE first and then be merged to -CURRENT.

This term is also used when a patch is merged from -STABLE to a security branch.

See Also: Merge From Current.

Message Of The Day

A message, usually shown on login, often used to distribute information to users of the system.

Multi-Level Security

Multiple APIC Description Table

N

NAT

See: Network Address Translation

NDISulator

See: Project Evil

NFS

See: Network File System

NTFS

See: New Technology File System

NTP

See: Network Time Protocol

Network Address Translation

A technique where IP packets are rewritten on the way through a gateway, enabling many machines behind the gateway to effectively share a single IP address.

Network File System

New Technology File System

A filesystem developed by Microsoft and available in its "New Technology" operating systems, such as Windows 2000, Windows NT and Windows XP.

Network Time Protocol

A means of synchronizing clocks over a network.

O

OBE

See: Overtaken By Events

ODMR

See: On-Demand Mail Relay

OS

See: Operating System

On-Demand Mail Relay

Operating System

A set of programs, libraries and tools that provide access to the hardware resources of a computer. Operating systems range today from simplistic designs that support only one program running at a time, accessing only one device to fully multi-user, multi-tasking and multi-process systems that can serve thousands of users simultaneously, each of them running dozens of different applications.

Overtaken By Events

Indicates a suggested change (such as a Problem Report or a feature request) which is no longer relevant or applicable due to such things as later changes to FreeBSD, changes in networking standards, the affected hardware having since become obsolete, and so forth.

P

p4

See: Perforce

PAE

See: Physical Address Extensions

PAM

See: Pluggable Authentication Modules

PAP

See: Password Authentication Protocol

PC

See: Personal Computer

PCNSFD

See: Personal Computer Network File System Daemon

PDF

See: Portable Document Format

PID

See: Process ID

POLA

See: Principle Of Least Astonishment

POP

See: Post Office Protocol

POP3

See: Post Office Protocol Version 3

PPD

See: PostScript Printer Description

PPP

See: Point-to-Point Protocol

PPPoA

See: PPP over ATM

PPPoE

See: PPP over Ethernet

PPP over ATM

PPP over Ethernet

PR

See: Problem Report

PXE

See: Preboot eXecution Environment

Password Authentication Protocol

Perforce

A source code control product made by Perforce Software (http://www.perforce.com/) which is more advanced than CVS. Although not open source, its use is free of charge to open-source projects such as FreeBSD.

Some FreeBSD developers use a Perforce repository as a staging area for code that is considered too experimental for the -CURRENT branch.

Personal Computer

Personal Computer Network File System Daemon

Physical Address Extensions

A method of enabling access to up to 64 GB of RAM on systems which only physically have a 32-bit wide address space (and would therefore be limited to 4 GB without PAE).

Pluggable Authentication Modules

Point-to-Point Protocol

Pointy Hat

A mythical piece of headgear, much like a `dunce cap`, awarded to any FreeBSD committer who breaks the build, makes revision numbers go backwards, or creates any other kind of havoc in the source base. Any committer worth his or her salt will soon accumulate a large collection. The usage is (almost always?) humorous.

Portable Document Format

Post Office Protocol

See Also: Post Office Protocol Version 3.

Post Office Protocol Version 3

A protocol for accessing email messages on a mail server, characterised by the messages usually being downloaded from the server to the client, as opposed to remaining on the server.

See Also: Internet Message Access Protocol.

PostScript Printer Description

Preboot eXecution Environment

Principle Of Least Astonishment

As FreeBSD evolves, changes visible to the user should be kept as unsurprising as possible. For example, arbitrarily rearranging system startup variables in `/etc/defaults/rc.conf` violates POLA. Developers consider POLA when contemplating user-visible system changes.

Problem Report

A description of some kind of problem that has been found in either the FreeBSD source or documentation. See Writing FreeBSD Problem Reports (http://www.FreeBSD.org/doc/zh_TW.Big5/articles/problem-reports/index.html).

Process ID

A number, unique to a particular process on a system, which identifies it and allows actions to be taken against it.

Project Evil

The working title for the NDISulator, written by Bill Paul, who named it referring to how awful it is (from a philosophical standpoint) to need to have something like this in the first place. The NDISulator is a special compatibility module to allow Microsoft Windows™ NDIS miniport network drivers to be used with FreeBSD/i386. This is usually the only way to use cards where the driver is closed-source. See `src/sys/compat/ndis/subr_ndis.c`.

R

RA

See: Router Advertisement

RAID

See: Redundant Array of Inexpensive Disks

RAM

See: Random Access Memory

RD

See: Received Data

RFC

See: Request For Comments

RISC

See: Reduced Instruction Set Computer

RPC

See: Remote Procedure Call

RS232C

See: Recommended Standard 232C

RTS

See: Request To Send

Random Access Memory

Received Data

An RS232C pin or wire that data is recieved on.

See Also: Transmitted Data.

Recommended Standard 232C

A standard for communications between serial devices.

Reduced Instruction Set Computer

An approach to processor design where the operations the hardware can perform are simplified but made as general purpose as possible. This can lead to lower power consumption, fewer transistors and in some cases, better performance and increased code density. Examples of RISC processors include the Alpha, Sparc, ARM and PowerPC.

Redundant Array of Inexpensive Disks

Remote Procedure Call

repocopy

See: Repository Copy

Repository Copy

A direct copying of files within the CVS repository.

Without a repocopy, if a file needed to be copied or moved to another place in the repository, the committer would run `cvs add` to put the file in its new location, and then `cvs rm` on the old file if the old copy was being removed.

The disadvantage of this method is that the history (i.e. the entries in the CVS logs) of the file would not be copied to the new location. As the FreeBSD Project considers this history very useful, a repository copy is often used instead. This is a process where one of the repository meisters will copy the files directly within the repository, rather than using the cvs(1) program.

Request For Comments

A set of documents defining Internet standards, protocols, and so forth. See www.rfc-editor.org (http://www.rfc-editor.org/).

Also used as a general term when someone has a suggested change and wants feedback.

Request To Send

An RS232C signal requesting that the remote system commences transmission of data.

See Also: Clear To Send.

Router Advertisement

S

SCI

See: System Control Interrupt

SCSI

See: Small Computer System Interface

SG

See: Signal Ground

SMB

See: Server Message Block

SMP

See: Symmetric MultiProcessor

SMTP

See: Simple Mail Transfer Protocol

SMTP AUTH

See: SMTP Authentication

SSH

See: Secure Shell

STR

See: Suspend To RAM

SVN

See: Subversion

SMTP Authentication

Server Message Block

Signal Ground

An RS232 pin or wire that is the ground reference for the signal.

Simple Mail Transfer Protocol

Secure Shell

Small Computer System Interface

Subversion

Subversion is a version control system, similar to CVS, but with an expanded feature list.

See Also: Concurrent Versions System.

Suspend To RAM

Symmetric MultiProcessor

System Control Interrupt

T

TCP

See: Transmission Control Protocol

TCP/IP

See: Transmission Control Protocol/Internet Protocol

TD

See: Transmitted Data

TFTP

See: Trivial FTP

TGT

See: Ticket-Granting Ticket

TSC

See: Time Stamp Counter

Ticket-Granting Ticket

Time Stamp Counter

A profiling counter internal to modern Pentium processors that counts core frequency clock ticks.

Transmission Control Protocol

A protocol that sits on top of (e.g.) the IP protocol and guarantees that packets are delivered in a reliable, ordered, fashion.

Transmission Control Protocol/Internet Protocol

The term for the combination of the TCP protocol running over the IP protocol. Much of the Internet runs over TCP/IP.

Transmitted Data

An RS232C pin or wire that data is transmitted on.

See Also: Received Data.

Trivial FTP

U

UDP

See: User Datagram Protocol

UFS1

See: Unix File System Version 1

UFS2

See: Unix File System Version 2

UID

See: User ID

URL

See: Uniform Resource Locator

USB

See: Universal Serial Bus

Uniform Resource Locator

A method of locating a resource, such as a document on the Internet and a means to identify that resource.

Unix File System Version 1

The original UNIX file system, sometimes called the Berkeley Fast File System.

Unix File System Version 2

An extension to UFS1, introduced in FreeBSD 5-CURRENT. UFS2 adds 64 bit block pointers (breaking the 1T barrier), support for extended file storage and other features.

Universal Serial Bus

A hardware standard used to connect a wide variety of computer peripherals to a universal interface.

User ID

A unique number assigned to each user of a computer, by which the resources and permissions assigned to that user can be identified.

User Datagram Protocol

A simple, unreliable datagram protocol which is used for exchanging data on a TCP/IP network. UDP does not provide error checking and correction like TCP.

V

VPN

See: Virtual Private Network

Virtual Private Network

A method of using a public telecommunication such as the Internet, to provide remote access to a localized network, such as a corporate LAN.

Colophon

This book is the combined work of hundreds of contributors to “The FreeBSD Documentation Project”. The text is authored in SGML according to the DocBook DTD and is formatted from SGML into many different presentation formats using **Jade**, an open source DSSSL engine. Norm Walsh’s DSSSL stylesheets were used with an additional customization layer to provide the presentation instructions for **Jade**. The printed version of this document would not be possible without Donald Knuth’s TEX typesetting language, Leslie Lamport’s LATEX, or Sebastian Rahtz’s **JadeTeX** macro package.