A Portable Subset of the Standard C Library

Andrew Makhorin*

July, 2008

Abstract

This memorandum defines a subset of the standard C run-time library, which is absolutely portable between all modern 32-bit platforms, that has been proven by the practice.

Using this subset as a base for implementation of the GLPK package provides its high portability and allows avoiding many portability problems. (Note that not all library objects from the subset are actually used in GLPK.)

The subset conforms to the International Standard ISO/IEC 9899.[1]

Diagnostics `<assert.h>`

```
NDEBUG
void assert(expr);
```

Character handling `<ctype.h>`

Character testing functions

```
int isalnum(int c);
int isalpha(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);
```

---

*Department for Applied Informatics, Moscow Aviation Institute, Moscow, Russia. E-mail: `<mao@mai2.rcnet.ru>`, `<mao@gnu.org>`.

[1] See `<http://www.open-std.org/JTC1/SC22/WG14/>`.

Character case mapping functions

```
int tolower(int c);
int toupper(int c);
```

Errors `<errno.h>`

```
EDOM
ERANGE
errno
```

Characteristics of floating types `<errno.h>`

```
FLT_RADIX
FLT_MANT_DIG
FLT_MIN_EXP
FLT_MAX_EXP
FLT_DIG
FLT_MIN_10_EXP
FLT_MAX_10_EXP
FLT_EPSILON
FLT_MIN
FLT_MAX
DBL_MANT_DIG
DBL_MIN_EXP
DBL_MAX_EXP
DBL_DIG
DBL_MIN_10_EXP
DBL_MAX_10_EXP
DBL_EPSILON
DBL_MIN
DBL_MAX
```

Size of integer types `<limits.h>`

```
CHAR_BIT
SCHAR_MIN
SCHAR_MAX
UCHAR_MAX
CHAR_MIN
CHAR_MAX
SHRT_MIN
SHRT_MAX
```

```
USHRT_MAX
INT_MIN
INT_MAX
UINT_MAX
LONG_MIN
LONG_MAX
ULONG_MAX
```

Mathematics `<math.h>`

Trigonometric functions

```
double acos(double x);
double asin(double x);
double atan(double x);
double atan2(double y, double x);
double cos(double x);
double sin(double x);
double tan(double x);
```

Hyperbolic functions

```
double cosh(double x);
double sinh(double x);
double tanh(double x);
```

Exponential and logarithmic functions

```
double exp(double x);
double frexp(double value, int *exp);
double ldexp(double x, int exp);
double log(double x);
double log10(double x);
double modf(double value, double *iptr);
```

Power and absolute value functions

```
double fabs(double x);
double pow(double x, double y);
double sqrt(double x);
```

Nearest integer functions

```
double ceil(double x);
double floor(double x);
```

Remainder function

```
double fmod(double x, double y);
```

Non-local jumps <setjmp.h>

```
jmp_buf
int setjmp(jmp_buf env);
void longjmp(jmp_buf env, int val);
```

Signal handling <signal.h>

```
sig_atomic_t
SIG_DFL
SIG_IGN
SIG_ERR
SIGINT
SIGILL
SIGFPE
SIGSEGV
SIGTERM
SIGABRT
void (*signal(int sig, void (*func)(int)))(int);
int raise(int sig);
```

Variable arguments <stdarg.h>

```
va_list
type va_arg(va_list ap, type);
void va_end(va_list ap);
void va_start(va_list ap, parmN);
```

Common definitions <stddef.h>

```
ptrdiff_t
size_t
NULL
offsetof(type, member-designator)
```

Input/output `<stdio.h>`

```
size_t
va_list
FILE
fpos_t
NULL
_IOFBF
_IOLBF
_IONBF
BUFSIZ
EOF
FOPEN_MAX
FILENAME_MAX
L_tmpnam
SEEK_SET
SEEK_CUR
SEEK_END
TMP_MAX
stdin
stdout
stderr
```

Operations on files

```
int remove(const char *filename);
int rename(const char *old, const char *new);
FILE *tmpfile(void);
char *tmpnam(char *s);
```

File access functions

```
int fclose(FILE *stream);
int fflush(FILE *stream);
FILE *fopen(const char *filename, const char *mode);
FILE *freopen(const char *filename, const char *mode,
   FILE *stream);
void setbuf(FILE *stream, char *buf);
int setvbuf(FILE *stream, char *buf, int mode, size_t size);
```

Formatted input/output functions

```
int fprintf(FILE *stream, const char *format, ...);
int fscanf(FILE *stream, const char *format, ...);
int printf(const char *format, ...);
int scanf(const char *format, ...);
int sprintf(char *s, const char *format, ...);
int sscanf(const char *s, const char *format, ...);
int vfprintf(FILE *stream, const char *format, va_list arg);
int vprintf(const char *format, va_list arg);
int vsprintf(char *s, const char *format, va_list arg);
```

Character input/output functions

```
int fgetc(FILE *stream);
char *fgets(char *s, int n, FILE *stream);
int fputc(int c, FILE *stream);
int fputs(const char *s, FILE *stream);
int getc(FILE *stream);
int getchar(void);
char *gets(char *s);
int putc(int c, FILE *stream);
int putchar(int c);
int puts(const char *s);
int ungetc(int c, FILE *stream);
```

Direct input/output functions

```
size_t fread(void *ptr, size_t size, size_t nmemb,
   FILE *stream);
size_t fwrite(const void *ptr, size_t size, size_t nmemb,
   FILE *stream);
```

File positioning functions

```
int fgetpos(FILE *stream, fpos_t *pos);
int fseek(FILE *stream, long int offset, int whence);
int fsetpos(FILE *stream, const fpos_t *pos);
long int ftell(FILE *stream);
void rewind(FILE *stream);
```

Error handling functions

```
void clearerr(FILE *stream);
int feof(FILE *stream);
int ferror(FILE *stream);
void perror(const char *s);
```

General utilities `<stdlib.h>`

```
size_t
div_t
ldiv_t
NULL
EXIT_SUCCESS
EXIT_FAILURE
RAND_MAX
```

String conversion functions

```
double atof(const char *nptr);
int atoi(const char *nptr);
long int atol(const char *nptr);
double strtod(const char *nptr, char **endptr);
long int strtol(const char *nptr, char **endptr, int base);
unsigned long int strtoul(const char *nptr, char **endptr,
   int base);
```

Pseudo-random sequence generation functions

```
int rand(void);
void srand(unsigned int seed);
```

Memory management functions

```
void *calloc(size_t nmemb, size_t size);
void free(void *ptr);
void *malloc(size_t size);
void *realloc(void *ptr, size_t size);
```

Communication with the environment

```
void abort(void);
int atexit(void (*func)(void));
void exit(int status);
char *getenv(const char *name);
int system(const char *string);
```

Searching and sorting utilities

```
void *bsearch(const void *key, const void *base,
   size_t nmemb, size_t size,
   int (*compar)(const void *, const void *));
```

```
void qsort(void *base, size_t nmemb, size_t size,
   int (*compar)(const void *, const void *));
```

Integer arithmetic functions

```
int abs(int j);
long int labs(long int j);
div_t div(int numer, int denom);
ldiv_t ldiv(long int numer, long int denom);
```

String handling `<string.h>`

```
size_t
NULL
```

Copying functions

```
void *memcpy(void *s1, const void *s2, size_t n);
void *memmove(void *s1, const void *s2, size_t n);
char *strcpy(char *s1, const char *s2);
char *strncpy(char *s1, const char *s2, size_t n);
```

Concatenation functions

```
char *strcat(char *s1, const char *s2);
char *strncat(char *s1, const char *s2, size_t n);
```

Comparison functions

```
int memcmp(const void *s1, const void *s2, size_t n);
int strcmp(const char *s1, const char *s2);
int strncmp(const char *s1, const char *s2, size_t n);
```

Search functions

```
void *memchr(const void *s, int c, size_t n);
char *strchr(const char *s, int c);
size_t strcspn(const char *s1, const char *s2);
char *strpbrk(const char *s1, const char *s2);
char *strrchr(const char *s, int c);
size_t strspn(const char *s1, const char *s2);
char *strstr(const char *s1, const char *s2);
char *strtok(char *s1, const char *s2);
```

Miscellaneous functions

```
void *memset(void *s, int c, size_t n);
char *strerror(int errnum);
size_t strlen(const char *s);
```

Date and time `<time.h>`

```
NULL
CLOCKS_PER_SEC
clock_t
time_t
struct tm
```

Time manipulation functions

```
clock_t clock(void);
double difftime(time_t time1, time_t time0);
time_t mktime(struct tm *timeptr);
time_t time(time_t *timer);
```

Time conversion functions

```
char *asctime(const struct tm *timeptr);
char *ctime(const time_t *timer);
struct tm *gmtime(const time_t *timer);
struct tm *localtime(const time_t *timer);
```