RTEMS POSIX 1003.1 Compliance Guide

Release 5.c5749d0-modified (1st November 2019)

CONTENTS

Copyrights and License

The authors have used their best efforts in preparing this material. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. No warranty of any kind, expressed or implied, with regard to the software or the material contained in this document is provided. No liability arising out of the application or use of any product described in this document is assumed. The authors reserve the right to revise this material and to make changes from time to time in the content hereof without obligation to notify anyone of such revision or changes.

The RTEMS Project is hosted at https://www.rtems.org. Any inquiries concerning RTEMS, its related support components, or its documentation should be directed to the RTEMS Project community.

RTEMS Online Resources

Home	https://www.rtems.org
Documentation	https://docs.rtems.org
Mailing Lists	https://lists.rtems.org
Bug Reporting	https://devel.rtems.org/wiki/Developer/Bug_Reporting
Git Repositories	https://git.rtems.org
Developers	https://devel.rtems.org

CHAPTER

ONE

PREFACE

RTEMS supports a variety of POSIX and BSD features including some POSIX methods that are now deemed obsolete and some methods for compatibility with GNU/Linux and FreeBSD. There are multiple POSIX standard versions as well as multiple efforts to tailor (e.g. profile) POSIX for embedded environments. They range in size from less than 200 required capabilities to the full POSIX standard which has over 1200 required capabilities. This document reports on the alignment of RTEMS with various standard versions and defined profiles.

RTEMS supports a number of POSIX process, user, and group oriented routines in what is referred to as a “SUSP” (Single-User, Single Process) manner. RTEMS supports a single process, multithreaded POSIX environment. In a pure world, there would be no reason to even include routines like `getpid()` when there can only be one process. But providing routines like `getpid()` and making them work in a sensible fashion for an embedded environment while not returning `ENOSYS` (for not implemented) makes it significantly easier to port code from a UNIX environment without modifying it.

In general, adding missing methods is always an open project for a volunteer. If considering addressing missing methods, please discuss this on mailing list. Some are properly implemented in the Newlib C Standard Library used by RTEMS. Others may require target architecture specific implementations. Still others may be impossible to implement without multiple processes or can only be implemented in a restricted fashion.

Missing methods required by the C99 standard or FACE Technical Standard Edition 3.0 General Purpose Profile are good candidates to add. Proposals to add missing methods from the C11 standard should be reviewed by RTEMS core developers to ensure the effort is well spent. There are rumors that some optional methods that are not being widely implemented will be removed in a future versino of the C Programming Language standard.

The next chapter in this document describes each of the standards with which the RTEMS alignment is tracked. Each subsequent chapter in this document presents the alignment of RTEMS with a specific standard version or defined profile. Each section with a chapter details the alignment of a specific header file relative to the chapter’s standard or profile. The implementation status of the items required by the standard are listed.

CHAPTER

TWO

STANDARDS

This chapter describes each of the standards which RTEMS tracks API alignment with. As a general rules, these standards are related to the POSIX or C programming language standards. Many are the result of domain specific efforts to define subsets or profiles or the full POSIX standard which are suitable for a specific domain. Each API set is considered a "profile" against which the full capability set of RTEMS is evaluated.

The RTEMS Complete Profile is the complete set of POSIX, BSD, and C programming language methods supported by RTEMS. This profile is independent of any standard and represents a union of multiple standards. For example, RTEMS supports BSD derived methods that are not in POSIX.

The IEEE Standard 1003.1 is the POSIX standard. Specifically, IEEE Standard 1003.1-2008 is the 2003 edition of the POSIX standard and IEEE Standard 1003.1-2008 is the 2008 edition. The 2008 is is an update from the 2003 edition. Each edition of the POSIX standard tends to add some methods, deprecate some methods, and obsolete (e.g. remove) other methods.

PSE51 through PSE54 are Open Group defined profiles of the 2003 edition of the POSIX standard. These profiles are:

- Profile 54 - Multipurpose
  - 1003.1-2003 Base Multi-process, Threads and File System
- Profile 53 - Dedicated
  - Multi-process, Threads and File System
- Profile 52 - Controller
  - Single Process, Threads, and File System
- Profile 51 - Minimal
  - Single Process, Threads, with No File System

The C99 Programming Language standard defines the Standard C Library. This library is largely included by reference in the POSIX standard.

The C11 Programming Language standard defines also defines an updated version of the Standard C Library. It deletes a few methods from the C99 version but adds many methods. A large portion of these methods are optional and not commonly implemented.

The Open Group FACE Consortium (https://www.opengroup.org/face) has defined four POSIX profiles targetting the avionics application domain. The FACE Technical Standard has been through multiple revisions and the POSIX API profiles are identical in Editions 1.0, 2.0, 2.1, and 2.1.1. In these editions, the profiles are as follows:

- Security - 163 APIs, single process, no FILE *
- Safety Basic - 246 APIs, single process, some FILE *
- Safety Extended - 335 APIs, multi-process, more FILE *
- General Purpose - 812 APIs, multi-process, much more

FACE Technical Standard Edition 3.0 adds the requirement for an operating system to support `clock_nanosleep()` in all profiles and defines one additional subcommand for the `posix_devctl()` methods.

RTEMS provides all of the methods required by the FACE Safety BASE profile and all of the methods in the Safety Extended profile which do not require multiple processes. Similarly, RTEMS provides most of the methods in the General Purpose profile which do not require multiple processes.

The Software Communications Architecture (SCA) specification targets the requirements for software-defined radios. This specification was originally developed in support of the Joint Tactical Radio System (JTRS) program in conjunction with the Object Management Group (OMG). This standard is now maintained by the Wireless Innovation Forum with support from the U.S. Navy Joint Tactical Network Center (JTNC). Some URLs of interest:

- SCA at Wireless Innovation Forum - http://www.wirelessinnovation.org/sca-based-standards-library
- JTRS - https://en.wikipedia.org/wiki/Joint_Tactical_Radio_System
- JTNC - http://www.public.navy.mil/jtnc/Pages/home.aspx

The SCA standard is hosted at the Wireless Innovation Forum with JTNC hosting supplemental information.

RTEMS includes all methods required by the SCA POSIX profiles.

CHAPTER

THREE

RTEMS COMPLETE PROFILE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

3.1 Summary

The follow table summarizes RTEMS supported methods for all tracked standards:

Supported	989
ENOSYS	19
Not supported	226

3.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio_cancel()
- aio_error()
- aio_fsync()
- aio_read()
- aio_return()
- aio_write()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio_suspend()
- lio_listio()

3.3 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

3.4 <assert.h>

The following methods and variables in `<assert.h>` are supported:

- `assert()`

3.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- cabs()
- cabsf()
- cabsl()
- cacos()
- cacosf()
- cacosh()
- cacoshf()
- cacoshl()
- cacosl()
- carg()
- cargf()
- cargl()
- casin()
- casinf()
- casinh()
- casinhf()
- casinhl()
- casinl()
- catan()
- catanf()
- catanh()
- catanhf()
- catanhl()
- catanl()
- ccos()
- ccosf()
- ccosh()
- ccoshf()
- ccoshl()
- ccosl()
- cexp()
- cexpf()

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

3.6 `<ctype.h>`

The following methods and variables in `<ctype.h>` are supported:

- `_tolower()`
- `_toupper()`
- `isalnum()`
- `isalnum_l()`
- `isalpha()`
- `isalpha_l()`
- `isascii()`
- `isblank()`
- `isblank_l()`
- `iscntrl()`
- `iscntrl_l()`
- `isdigit()`
- `isdigit_l()`
- `isgraph()`
- `isgraph_l()`
- `islower()`
- `islower_l()`
- `isprint()`
- `isprint_l()`
- `ispunct()`
- `ispunct_l()`
- `isspace()`
- `isspace_l()`
- `isupper()`
- `isupper_l()`
- `isxdigit()`
- `isxdigit_l()`
- `toascii()`
- `tolower()`
- `tolower_l()`
- `toupper()`
- `toupper_l()`

3.7 `<devctl.h>`

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

3.8 <dirent.h>

The following methods and variables in `<dirent.h>` are supported:

- `alphasort()`
- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `scandir()`
- `seekdir()`
- `telldir()`

The following methods and variables in `<dirent.h>` are not supported:

- `dirfd()`
- `fdopendir()`

3.9 `<dlfcn.h>`

The following methods and variables in `<dlfcn.h>` are supported:

- `dlclose()`
- `dlerror()`
- `dlopen()`
- `dlsym()`

3.10 <errno.h>

The following methods and variables in `<errno.h>` are supported:

- `errno`

3.11 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- openat()
- posix_fadvise()
- posix_fallocate()
- posix_openpt()

3.12 <fenv.h>

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

3.13 <fmtmsg.h>

The following methods and variables in <fmtmsg.h> are not supported:

- fmtmsg()

3.14 <fnmatch.h>

The following methods and variables in <fnmatch.h> are supported:

- fnmatch()

3.15 `<ftw.h>`

The following methods and variables in `<ftw.h>` are not supported:

- `ftw()`
- `nftw()`

3.16 `<glob.h>`

The following methods and variables in `<glob.h>` are supported:

- `glob()`
- `globfree()`

3.17 <grp.h>

The following methods and variables in <grp.h> are supported:

- endgrent()
- getgrent()
- getgrgid()
- getgrgid_r()
- getgrnam()
- getgrnam_r()
- setgrent()

3.18 <iconv.h>

The following methods and variables in <iconv.h> are supported:

- iconv()
- iconv_close()
- iconv_open()

3.19 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

3.20 <langinfo.h>

The following methods and variables in <langinfo.h> are supported:

- nl_langinfo()
- nl_langinfo_l()

3.21 <libgen.h>

The following methods and variables in `<libgen.h>` are supported:

- `basename()`
- `dirname()`

3.22 <locale.h>

The following methods and variables in `<locale.h>` are supported:

- `duplocale()`
- `freelocale()`
- `localeconv()`
- `newlocale()`
- `setlocale()`
- `uselocale()`

3.23 <math.h>

The following methods and variables in <math.h> are supported:

- acos()
- acosf()
- acosh()
- acoshf()
- acoshl()
- acosl()
- asin()
- asinf()
- asinh()
- asinhf()
- asinhl()
- asinl()
- atan()
- atan2()
- atan2f()
- atan2l()
- atanf()
- atanh()
- atanhf()
- atanhl()
- atanl()
- cbrt()
- cbrtf()
- cbrtl()
- ceil()
- ceilf()
- ceill()
- copysign()
- copysignf()
- copysignl()
- cos()
- cosf()

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- fmin()
- fminf()
- fminl()
- fmod()
- fmodf()
- fmodl()
- frexp()
- frexpf()
- frexpl()
- hypot()
- hypotf()
- hypotl()
- ilogb()
- ilogbf()
- ilogbl()
- isinf()
- isnan()
- ldexp()
- ldexpf()
- ldexpl()
- lgamma()
- lgammaf()
- lgammal()
- llrint()
- llrintf()
- llrintl()
- llround()
- llroundf()
- llroundl()
- log()
- log10()
- log10f()
- log10l()
- log1p()

- log1pf()
- log1pl()
- log2()
- log2f()
- log2l()
- logb()
- logbf()
- logbl()
- logf()
- logl()
- lrint()
- lrintf()
- lrintl()
- lround()
- lroundf()
- lroundl()
- modf()
- modff()
- modfl()
- nan()
- nanf()
- nanl()
- nearbyint()
- nearbyintf()
- nearbyintl()
- nextafter()
- nextafterf()
- nextafterl()
- nexttoward()
- nexttowardl()
- pow()
- powf()
- powl()
- remainder()

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalb()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()

- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `j0()`
- `j1()`
- `jn()`
- `nexttowardf()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

3.24 <monetary.h>

The following methods and variables in <monetary.h> are not supported:

- strfmon()
- strfmon_l()

3.25 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_close()
- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_timedreceive()
- mq_timedsend()
- mq_unlink()

3.26 <ndbm.h>

The following methods and variables in <ndbm.h> are not supported:

- dbm_clearerr()
- dbm_close()
- dbm_delete()
- dbm_error()
- dbm_fetch()
- dbm_firstkey()
- dbm_nextkey()
- dbm_open()
- dbm_store()

3.27 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if_freenameindex()
- if_indextoname()
- if_nameindex()
- if_nametoindex()

3.28 `<netdb.h>`

The following methods and variables in `<netdb.h>` are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostbyaddr()`
- `gethostbyname()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `h_errno`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

3.29 <nl_types.h>

The following methods and variables in <nl_types.h> are not supported:

- catclose()
- catgets()
- catopen()

3.30 `<poll.h>`

The following methods and variables in `<poll.h>` are not supported:

- `poll()`

3.31 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstackaddr()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstackaddr()
- pthread_attr_setstacksize()
- pthread_barrier_destroy()
- pthread_barrier_init()
- pthread_barrier_wait()
- pthread_barrierattr_destroy()
- pthread_barrierattr_getpshared()
- pthread_barrierattr_init()
- pthread_barrierattr_setpshared()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()

- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_getpshared()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_condattr_setpshared()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_getpshared()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()

- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_mutexattr_setpshared()
- pthread_mutexattr_settype()
- pthread_once()
- pthread_rwlock_destroy()
- pthread_rwlock_init()
- pthread_rwlock_rdlock()
- pthread_rwlock_timedrdlock()
- pthread_rwlock_timedwrlock()
- pthread_rwlock_tryrdlock()
- pthread_rwlock_trywrlock()
- pthread_rwlock_unlock()
- pthread_rwlock_wrlock()
- pthread_rwlockattr_destroy()
- pthread_rwlockattr_getpshared()
- pthread_rwlockattr_init()
- pthread_rwlockattr_setpshared()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()
- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()
- pthread_spin_destroy()
- pthread_spin_init()
- pthread_spin_lock()
- pthread_spin_trylock()
- pthread_spin_unlock()
- pthread_testcancel()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_atfork()
- pthread_getcpuclockid()

The following methods and variables in `<pthread.h>` are not supported:

- `pthread_mutex_consistent()`
- `pthread_mutexattr_getrobust()`
- `pthread_mutexattr_setrobust()`

3.32 `<pwd.h>`

The following methods and variables in `<pwd.h>` are supported:

- `endpwent()`
- `getpwent()`
- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`
- `setpwent()`

3.33 <regex.h>

The following methods and variables in `<regex.h>` are supported:

- `regcomp()`
- `regerror()`
- `regexec()`
- `regfree()`

3.34 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

3.35 <search.h>

The following methods and variables in <search.h> are supported:

- hcreate()
- hdestroy()
- hsearch()
- tdelete()
- tfind()
- tsearch()
- twalk()

The following methods and variables in <search.h> are not supported:

- insque()
- lfind()
- lsearch()
- remque()

3.36 <semaphore.h>

The following methods and variables in `<semaphore.h>` are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

3.37 <setjmp.h>

The following methods and variables in `<setjmp.h>` are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

The following methods and variables in `<setjmp.h>` are not supported:

- `_longjmp()`
- `_setjmp()`

3.38 <signal.h>

The following methods and variables in <signal.h> are supported:

- bsd_signal()
- kill()
- psignal()
- pthread_kill()
- pthread_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

The following methods and variables in <signal.h> are not supported:

- killpg()
- psiginfo()
- sigaltstack()
- sighold()
- sigignore()
- siginterrupt()
- sigpause()
- sigrelse()
- sigset()

3.39 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawn_file_actions_addclose()
- posix_spawn_file_actions_adddup2()
- posix_spawn_file_actions_addopen()
- posix_spawn_file_actions_destroy()
- posix_spawn_file_actions_init()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getpgroup()
- posix_spawnattr_getschedparam()
- posix_spawnattr_getschedpolicy()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setpgroup()
- posix_spawnattr_setschedparam()
- posix_spawnattr_setschedpolicy()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()
- posix_spawnp()

3.40 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va_arg()
- va_copy()
- va_end()
- va_start()

3.41 <stddef.h>

The following methods and variables in `<stddef.h>` are supported:

- `offsetof()`

3.42 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- dprintf()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fmemopen()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc_unlocked()
- getchar()

- getchar_unlocked()
- gets()
- open_memstream()
- perror()
- printf()
- putc()
- putc_unlocked()
- putchar()
- putchar_unlocked()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tempnam()
- tmpfile()
- tmpnam()
- ungetc()
- vdprintf()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

The following methods and variables in <stdio.h> are not supported:

- getdelim()
- getline()
- pclose()
- popen()
- renameat()

3.43 <stdlib.h>

The following methods and variables in `<stdlib.h>` are supported:

- `_Exit()`
- `a64l()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `drand48()`
- `ecvt()`
- `erand48()`
- `exit()`
- `fcvt()`
- `free()`
- `gcvt()`
- `getenv()`
- `getsubopt()`
- `jrand48()`
- `l64a()`
- `labs()`
- `lcong48()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `lrand48()`
- `malloc()`
- `mblen()`
- `mbstowcs()`

- `mbtowc()`
- `mkdtemp()`
- `mkstemp()`
- `mrand48()`
- `nrand48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `rand()`
- `rand_r()`
- `random()`
- `realloc()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`
- `ptsname()`

- `setkey()`
- `setstate()`
- `unlockpt()`

3.44 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memccpy()`
- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `stpcpy()`
- `stpncpy()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcoll_l()`
- `strcpy()`
- `strcspn()`
- `strdup()`
- `strerror()`
- `strerror_l()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strndup()`
- `strnlen()`
- `strpbrk()`
- `strrchr()`
- `strsignal()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`

- strxfrm()
- strxfrm_l()

3.45 <strings.h>

The following methods and variables in `<strings.h>` are supported:

- `bcmp()`
- `bcopy()`
- `ffs()`
- `index()`
- `rindex()`
- `strcasecmp()`
- `strcasecmp_l()`
- `strncasecmp()`
- `strncasecmp_l()`

3.46 `<stropts.h>`

The following methods and variables in `<stropts.h>` are supported:

- `ioctl()`

The following methods and variables in `<stropts.h>` are not supported:

- `fattach()`
- `fdetach()`
- `getmsg()`
- `getpmsg()`
- `isastream()`
- `putmsg()`
- `putpmsg()`

3.47 <sys/ipc.h>

The following methods and variables in <sys/ipc.h> are not supported:

- ftok()

3.48 <sys/mman.h>

The following methods and variables in `<sys/mman.h>` are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `posix_madvise()`
- `shm_open()`
- `shm_unlink()`

The following methods and variables in `<sys/mman.h>` are not supported:

- `posix_mem_offset()`
- `posix_typed_mem_get_info()`
- `posix_typed_mem_open()`

3.49 <sys/msg.h>

The following methods and variables in `<sys/msg.h>` are not supported:

- `msgctl()`
- `msgget()`
- `msgrcv()`
- `msgsnd()`

3.50 <sys/resource.h>

The following methods and variables in <sys/resource.h> are supported:

- getrusage()

The following methods and variables in <sys/resource.h> are not supported:

- getpriority()
- getrlimit()
- setpriority()
- setrlimit()

3.51 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

The following methods and variables in `<sys/select.h>` are not supported:

- `pselect()`

3.52 <sys/sem.h>

The following methods and variables in `<sys/sem.h>` are not supported:

- `semctl()`
- `semget()`
- `semop()`

3.53 <sys/shm.h>

The following methods and variables in <sys/shm.h> are not supported:

- shmat()
- shmctl()
- shmdt()
- shmget()

3.54 <sys/socket.h>

The following methods and variables in `<sys/socket.h>` are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in `<sys/socket.h>` are not supported:

- `sockatmark()`

3.55 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- mknod()
- stat()
- umask()

The following methods and variables in <sys/stat.h> are not supported:

- fchmodat()
- fstatat()
- futimens()
- mkdirat()
- mkfifoat()
- mknodat()
- utimensat()

3.56 `<sys/statvfs.h>`

The following methods and variables in `<sys/statvfs.h>` are supported:

- `statvfs()`

The following methods and variables in `<sys/statvfs.h>` are not supported:

- `fstatvfs()`

3.57 <sys/time.h>

The following methods and variables in `<sys/time.h>` are supported:

- `gettimeofday()`
- `utimes()`

The following methods in `<sys/time.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `getitimer()`
- `setitimer()`

3.58 <sys/times.h>

The following methods and variables in `<sys/times.h>` are supported:

- `times()`

3.59 <sys/uio.h>

The following methods and variables in `<sys/uio.h>` are supported:

- `readv()`
- `writev()`

3.60 `<sys/utsname.h>`

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

3.61 <sys/wait.h>

The following methods and variables in `<sys/wait.h>` are supported:

- `wait()`
- `waitpid()`

The following methods and variables in `<sys/wait.h>` are not supported:

- `waitid()`

3.62 <syslog.h>

The following methods and variables in <syslog.h> are not supported:

- closelog()
- openlog()
- setlogmask()
- syslog()

3.63 <termios.h>

The following methods and variables in `<termios.h>` are supported:

- `cfgetispeed()`
- `cfgetospeed()`
- `cfsetispeed()`
- `cfsetospeed()`
- `tcdrain()`
- `tcflow()`
- `tcflush()`
- `tcgetattr()`
- `tcsendbreak()`
- `tcsetattr()`

The following methods and variables in `<termios.h>` are not supported:

- `tcgetsid()`

3.64 <threads.h>

The following methods and variables in <threads.h> are supported:

- call_once()
- cnd_broadcast()
- cnd_destroy()
- cnd_init()
- cnd_signal()
- cnd_timedwait()
- cnd_wait()
- mtx_destroy()
- mtx_init()
- mtx_lock()
- mtx_timedlock()
- mtx_trylock()
- mtx_unlock()
- thrd_create()
- thrd_current()
- thrd_detach()
- thrd_equal()
- thrd_exit()
- thrd_join()
- thrd_sleep()
- thrd_yield()
- tss_create()
- tss_delete()
- tss_get()
- tss_set()

3.65 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime_r()
- clock()
- clock_getres()
- clock_gettime()
- clock_nanosleep()
- clock_settime()
- ctime()
- ctime_r()
- difftime()
- gmtime()
- gmtime_r()
- localtime()
- localtime_r()
- mktime()
- nanosleep()
- strftime()
- strftime_l()
- strptime()
- time()
- timer_create()
- timer_delete()
- timer_getoverrun()
- timer_gettime()
- timer_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- `daylight`
- `getdate()`
- `getdate_err`

3.66 <trace.h>

The following methods and variables in <trace.h> are not supported:

- posix_trace_attr_destroy()
- posix_trace_attr_getclockres()
- posix_trace_attr_getcreatetime()
- posix_trace_attr_getgenversion()
- posix_trace_attr_getinherited()
- posix_trace_attr_getlogfullpolicy()
- posix_trace_attr_getlogsize()
- posix_trace_attr_getmaxdatasize()
- posix_trace_attr_getmaxsystemeventsize()
- posix_trace_attr_getmaxusereventsize()
- posix_trace_attr_getname()
- posix_trace_attr_getstreamfullpolicy()
- posix_trace_attr_getstreamsize()
- posix_trace_attr_init()
- posix_trace_attr_setinherited()
- posix_trace_attr_setlogfullpolicy()
- posix_trace_attr_setlogsize()
- posix_trace_attr_setmaxdatasize()
- posix_trace_attr_setname()
- posix_trace_attr_setstreamfullpolicy()
- posix_trace_attr_setstreamsize()
- posix_trace_clear()
- posix_trace_close()
- posix_trace_create()
- posix_trace_create_withlog()
- posix_trace_event()
- posix_trace_eventid_equal()
- posix_trace_eventid_get_name()
- posix_trace_eventid_open()
- posix_trace_eventset_add()
- posix_trace_eventset_del()
- posix_trace_eventset_empty()

- posix_trace_eventset_fill()
- posix_trace_eventset_ismember()
- posix_trace_eventtypelist_getnext_id()
- posix_trace_eventtypelist_rewind()
- posix_trace_flush()
- posix_trace_get_attr()
- posix_trace_get_filter()
- posix_trace_get_status()
- posix_trace_getnext_event()
- posix_trace_open()
- posix_trace_rewind()
- posix_trace_set_filter()
- posix_trace_shutdown()
- posix_trace_start()
- posix_trace_stop()
- posix_trace_timedgetnext_event()
- posix_trace_trid_eventid_open()
- posix_trace_trygetnext_event()

3.67 <ulimit.h>

The following methods and variables in <ulimit.h> are not supported:

- ulimit()

3.68 `<unistd.h>`

The following methods and variables in `<unistd.h>` are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchdir()`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `lchown()`
- `link()`
- `lseek()`

- optarg
- opterr
- optind
- optopt
- pathconf()
- pause()
- pipe()
- pread()
- pwrite()
- read()
- readlink()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setpgid()
- setsid()
- setuid()
- sleep()
- swab()
- symlink()
- sync()
- sysconf()
- tcgetpgrp()
- tcsetpgrp()
- truncate()
- ttyname()
- ttyname_r()
- ualarm()
- unlink()
- usleep()
- write()

The following methods in <unistd.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- execl()

- `execle()`
- `execlp()`
- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `faccessat()`
- `fchownat()`
- `fexecve()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `linkat()`
- `lockf()`
- `nice()`
- `readlinkat()`
- `setpgrp()`
- `setregid()`
- `setreuid()`
- `symlinkat()`
- `unlinkat()`

3.69 <utime.h>

The following methods and variables in <utime.h> are supported:

- utime()

3.70 <utmpx.h>

The following methods and variables in <utmpx.h> are not supported:

- endutxent()
- getutxent()
- getutxid()
- getutxline()
- pututxline()
- setutxent()

3.71 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsnrtowcs()
- mbsrtowcs()
- open_wmemstream()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcpcpy()
- wcpncpy()
- wcrtomb()
- wcscasecmp()
- wcscasecmp_l()

- `wcscat()`
- `wcschr()`
- `wcscmp()`
- `wcscoll()`
- `wcscoll_l()`
- `wcscpy()`
- `wcscspn()`
- `wcsdup()`
- `wcsftime()`
- `wcslen()`
- `wcsncasecmp()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcsnlen()`
- `wcsnrtombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wcsxfrm_l()`
- `wctob()`
- `wcwidth()`

- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

The following methods and variables in <wchar.h> are not supported:

- wcsncasemcp_l()

3.72 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- iswalnum()
- iswalnum_l()
- iswalpha()
- iswalpha_l()
- iswblank()
- iswblank_l()
- iswcntrl()
- iswcntrl_l()
- iswctype()
- iswctype_l()
- iswdigit()
- iswdigit_l()
- iswgraph()
- iswgraph_l()
- iswlower()
- iswlower_l()
- iswprint()
- iswprint_l()
- iswpunct()
- iswpunct_l()
- iswspace()
- iswspace_l()
- iswupper()
- iswupper_l()
- iswxdigit()
- iswxdigit_l()
- towctrans()
- towctrans_l()
- towlower()
- towlower_l()
- towupper()
- towupper_l()

- wctrans()
- wctrans_l()
- wctype()
- wctype_l()

3.73 <wordexp.h>

The following methods and variables in `<wordexp.h>` are not supported:

- `wordexp()`
- `wordfree()`

CHAPTER

FOUR

POSIX-2008

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

4.1 Summary

The follow table summarizes alignment with the POSIX-2008 standard:

Supported	949
ENOSYS	19
Not supported	226

4.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio_cancel()
- aio_error()
- aio_fsync()
- aio_read()
- aio_return()
- aio_write()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio_suspend()
- lio_listio()

4.3 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

4.4 <assert.h>

The following methods and variables in `<assert.h>` are supported:

- `assert()`

4.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- cabs()
- cabsf()
- cabsl()
- cacos()
- cacosf()
- cacosh()
- cacoshf()
- cacoshl()
- cacosl()
- carg()
- cargf()
- cargl()
- casin()
- casinf()
- casinh()
- casinhf()
- casinhl()
- casinl()
- catan()
- catanf()
- catanh()
- catanhf()
- catanhl()
- catanl()
- ccos()
- ccosf()
- ccosh()
- ccoshf()
- ccoshl()
- ccosl()
- cexp()
- cexpf()

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

4.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- _tolower()
- _toupper()
- isalnum()
- isalnum_l()
- isalpha()
- isalpha_l()
- isascii()
- isblank()
- isblank_l()
- iscntrl()
- iscntrl_l()
- isdigit()
- isdigit_l()
- isgraph()
- isgraph_l()
- islower()
- islower_l()
- isprint()
- isprint_l()
- ispunct()
- ispunct_l()
- isspace()
- isspace_l()
- isupper()
- isupper_l()
- isxdigit()
- isxdigit_l()
- toascii()
- tolower()
- tolower_l()
- toupper()
- toupper_l()

4.7 `<dirent.h>`

The following methods and variables in `<dirent.h>` are supported:

- `alphasort()`
- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `scandir()`
- `seekdir()`
- `telldir()`

The following methods and variables in `<dirent.h>` are not supported:

- `dirfd()`
- `fdopendir()`

4.8 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- dlclose()
- dlerror()
- dlopen()
- dlsym()

4.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

4.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- openat()
- posix_fadvise()
- posix_fallocate()
- posix_openpt()

4.11 <fenv.h>

The following methods and variables in <fenv.h> are not supported:

- feclearexcept()
- fegetenv()
- fegetexceptflag()
- fegetround()
- feholdexcept()
- feraiseexcept()
- fesetenv()
- fesetexceptflag()
- fesetround()
- fetestexcept()
- feupdateenv()

4.12 <fmtmsg.h>

The following methods and variables in <fmtmsg.h> are not supported:

- fmtmsg()

4.13 <fnmatch.h>

The following methods and variables in `<fnmatch.h>` are supported:

- `fnmatch()`

4.14 `<ftw.h>`

The following methods and variables in `<ftw.h>` are not supported:

- `ftw()`
- `nftw()`

4.15 <glob.h>

The following methods and variables in `<glob.h>` are supported:

- `glob()`
- `globfree()`

4.16 <grp.h>

The following methods and variables in <grp.h> are supported:

- endgrent()
- getgrent()
- getgrgid()
- getgrgid_r()
- getgrnam()
- getgrnam_r()
- setgrent()

4.17 <iconv.h>

The following methods and variables in `<iconv.h>` are supported:

- `iconv()`
- `iconv_close()`
- `iconv_open()`

4.18 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

4.19 `<langinfo.h>`

The following methods and variables in `<langinfo.h>` are supported:

- `nl_langinfo()`
- `nl_langinfo_l()`

4.20 <libgen.h>

The following methods and variables in <libgen.h> are supported:

- basename()
- dirname()

4.21 `<locale.h>`

The following methods and variables in `<locale.h>` are supported:

- `duplocale()`
- `freelocale()`
- `localeconv()`
- `newlocale()`
- `setlocale()`
- `uselocale()`

4.22 <math.h>

The following methods and variables in <math.h> are supported:

- acos()
- acosf()
- acosh()
- acoshf()
- acoshl()
- acosl()
- asin()
- asinf()
- asinh()
- asinhf()
- asinhl()
- asinl()
- atan()
- atan2()
- atan2f()
- atan2l()
- atanf()
- atanh()
- atanhf()
- atanhl()
- atanl()
- cbrt()
- cbrtf()
- cbrtl()
- ceil()
- ceilf()
- ceill()
- copysign()
- copysignf()
- copysignl()
- cos()
- cosf()

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`
- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`

- log1pf()
- log1pl()
- log2()
- log2f()
- log2l()
- logb()
- logbf()
- logbl()
- logf()
- logl()
- lrint()
- lrintf()
- lrintl()
- lround()
- lroundf()
- lroundl()
- modf()
- modff()
- modfl()
- nan()
- nanf()
- nanl()
- nearbyint()
- nearbyintf()
- nearbyintl()
- nextafter()
- nextafterf()
- nextafterl()
- nexttoward()
- nexttowardl()
- pow()
- powf()
- powl()
- remainder()

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `j0()`
- `j1()`
- `jn()`
- `nexttowardf()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

4.23 <monetary.h>

The following methods and variables in `<monetary.h>` are not supported:

- `strfmon()`
- `strfmon_l()`

4.24 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_close()
- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_timedreceive()
- mq_timedsend()
- mq_unlink()

4.25 <ndbm.h>

The following methods and variables in `<ndbm.h>` are not supported:

- `dbm_clearerr()`
- `dbm_close()`
- `dbm_delete()`
- `dbm_error()`
- `dbm_fetch()`
- `dbm_firstkey()`
- `dbm_nextkey()`
- `dbm_open()`
- `dbm_store()`

4.26 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if_freenameindex()
- if_indextoname()
- if_nameindex()
- if_nametoindex()

4.27 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- endhostent()
- endnetent()
- endprotoent()
- endservent()
- freeaddrinfo()
- gai_strerror()
- getaddrinfo()
- gethostent()
- getnameinfo()
- getnetbyaddr()
- getnetbyname()
- getnetent()
- getprotobyname()
- getprotobynumber()
- getprotoent()
- getservbyname()
- getservbyport()
- getservent()
- sethostent()
- setnetent()
- setprotoent()
- setservent()

4.28 <nl_types.h>

The following methods and variables in `<nl_types.h>` are not supported:

- `catclose()`
- `catgets()`
- `catopen()`

4.29 <poll.h>

The following methods and variables in <poll.h> are not supported:

- poll()

4.30 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstackaddr()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstackaddr()
- pthread_attr_setstacksize()
- pthread_barrier_destroy()
- pthread_barrier_init()
- pthread_barrier_wait()
- pthread_barrierattr_destroy()
- pthread_barrierattr_getpshared()
- pthread_barrierattr_init()
- pthread_barrierattr_setpshared()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()

- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_getpshared()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_condattr_setpshared()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_getpshared()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()

- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_getpshared()`
- `pthread_rwlockattr_init()`
- `pthread_rwlockattr_setpshared()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_spin_destroy()`
- `pthread_spin_init()`
- `pthread_spin_lock()`
- `pthread_spin_trylock()`
- `pthread_spin_unlock()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

The following methods and variables in `<pthread.h>` are not supported:

- `pthread_mutex_consistent()`
- `pthread_mutexattr_getrobust()`
- `pthread_mutexattr_setrobust()`

4.31 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- endpwent()
- getpwent()
- getpwnam()
- getpwnam_r()
- getpwuid()
- getpwuid_r()
- setpwent()

4.32 `<regex.h>`

The following methods and variables in `<regex.h>` are supported:

- `regcomp()`
- `regerror()`
- `regexec()`
- `regfree()`

4.33 <sched.h>

The following methods and variables in `<sched.h>` are supported:

- `sched_get_priority_max()`
- `sched_get_priority_min()`
- `sched_rr_get_interval()`
- `sched_yield()`

The following methods in `<sched.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `sched_getparam()`
- `sched_getscheduler()`
- `sched_setparam()`
- `sched_setscheduler()`

4.34 <search.h>

The following methods and variables in <search.h> are supported:

- hcreate()
- hdestroy()
- hsearch()
- tdelete()
- tfind()
- tsearch()
- twalk()

The following methods and variables in <search.h> are not supported:

- insque()
- lfind()
- lsearch()
- remque()

4.35 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

4.36 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

The following methods and variables in <setjmp.h> are not supported:

- _longjmp()
- _setjmp()

4.37 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- psignal()
- pthread_kill()
- pthread_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

The following methods and variables in <signal.h> are not supported:

- killpg()
- psiginfo()
- sigaltstack()
- sighold()
- sigignore()
- siginterrupt()
- sigpause()
- sigrelse()
- sigset()

4.38 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawn_file_actions_addclose()
- posix_spawn_file_actions_adddup2()
- posix_spawn_file_actions_addopen()
- posix_spawn_file_actions_destroy()
- posix_spawn_file_actions_init()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getpgroup()
- posix_spawnattr_getschedparam()
- posix_spawnattr_getschedpolicy()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setpgroup()
- posix_spawnattr_setschedparam()
- posix_spawnattr_setschedpolicy()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()
- posix_spawnp()

4.39 <stdarg.h>

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

4.40 `<stddef.h>`

The following methods and variables in `<stddef.h>` are supported:

- `offsetof()`

4.41 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- dprintf()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fmemopen()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc_unlocked()
- getchar()

- getchar_unlocked()
- gets()
- open_memstream()
- perror()
- printf()
- putc()
- putc_unlocked()
- putchar()
- putchar_unlocked()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tempnam()
- tmpfile()
- tmpnam()
- ungetc()
- vdprintf()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

The following methods and variables in `<stdio.h>` are not supported:

- `getdelim()`
- `getline()`
- `pclose()`
- `popen()`
- `renameat()`

4.42 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- _Exit()
- a64l()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- drand48()
- erand48()
- exit()
- free()
- getenv()
- getsubopt()
- jrand48()
- l64a()
- labs()
- lcong48()
- ldiv()
- llabs()
- lldiv()
- lrand48()
- malloc()
- mblen()
- mbstowcs()
- mbtowc()
- mkdtemp()
- mkstemp()

- `mrand48()`
- `nrand48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `rand()`
- `rand_r()`
- `random()`
- `realloc()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`
- `ptsname()`
- `setkey()`
- `setstate()`
- `unlockpt()`

4.43 `<string.h>`

The following methods and variables in `<string.h>` are supported:

- `memccpy()`
- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `stpcpy()`
- `stpncpy()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcoll_l()`
- `strcpy()`
- `strcspn()`
- `strdup()`
- `strerror()`
- `strerror_l()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strndup()`
- `strnlen()`
- `strpbrk()`
- `strrchr()`
- `strsignal()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`

- `strxfrm()`
- `strxfrm_l()`

4.44 <strings.h>

The following methods and variables in <strings.h> are supported:

- ffs()
- strcasecmp()
- strcasecmp_l()
- strncasecmp()
- strncasecmp_l()

4.45 <stropts.h>

The following methods and variables in <stropts.h> are supported:

- ioctl()

The following methods and variables in <stropts.h> are not supported:

- fattach()
- fdetach()
- getmsg()
- getpmsg()
- isastream()
- putmsg()
- putpmsg()

4.46 <sys/ipc.h>

The following methods and variables in `<sys/ipc.h>` are not supported:

- `ftok()`

4.47 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mlock()
- mlockall()
- mmap()
- mprotect()
- msync()
- munlock()
- munlockall()
- munmap()
- posix_madvise()
- shm_open()
- shm_unlink()

The following methods and variables in <sys/mman.h> are not supported:

- posix_mem_offset()
- posix_typed_mem_get_info()
- posix_typed_mem_open()

4.48 <sys/msg.h>

The following methods and variables in `<sys/msg.h>` are not supported:

- `msgctl()`
- `msgget()`
- `msgrcv()`
- `msgsnd()`

4.49 <sys/resource.h>

The following methods and variables in `<sys/resource.h>` are supported:

- `getrusage()`

The following methods and variables in `<sys/resource.h>` are not supported:

- `getpriority()`
- `getrlimit()`
- `setpriority()`
- `setrlimit()`

4.50 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD_CLR()
- FD_ISSET()
- FD_SET()
- FD_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

4.51 <sys/sem.h>

The following methods and variables in `<sys/sem.h>` are not supported:

- `semctl()`
- `semget()`
- `semop()`

4.52 <sys/shm.h>

The following methods and variables in `<sys/shm.h>` are not supported:

- `shmat()`
- `shmctl()`
- `shmdt()`
- `shmget()`

4.53 <sys/socket.h>

The following methods and variables in `<sys/socket.h>` are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in `<sys/socket.h>` are not supported:

- `sockatmark()`

4.54 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- mknod()
- stat()
- umask()

The following methods and variables in <sys/stat.h> are not supported:

- fchmodat()
- fstatat()
- futimens()
- mkdirat()
- mkfifoat()
- mknodat()
- utimensat()

4.55 <sys/statvfs.h>

The following methods and variables in <sys/statvfs.h> are supported:

- statvfs()

The following methods and variables in <sys/statvfs.h> are not supported:

- fstatvfs()

4.56 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- gettimeofday()
- utimes()

The following methods in <sys/time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- getitimer()
- setitimer()

4.57 <sys/times.h>

The following methods and variables in `<sys/times.h>` are supported:

- `times()`

4.58 <sys/uio.h>

The following methods and variables in <sys/uio.h> are supported:

- readv()
- writev()

4.59 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

4.60 <sys/wait.h>

The following methods and variables in `<sys/wait.h>` are supported:

- `wait()`
- `waitpid()`

The following methods and variables in `<sys/wait.h>` are not supported:

- `waitid()`

4.61 <syslog.h>

The following methods and variables in `<syslog.h>` are not supported:

- `closelog()`
- `openlog()`
- `setlogmask()`
- `syslog()`

4.62 <termios.h>

The following methods and variables in `<termios.h>` are supported:

- `cfgetispeed()`
- `cfgetospeed()`
- `cfsetispeed()`
- `cfsetospeed()`
- `tcdrain()`
- `tcflow()`
- `tcflush()`
- `tcgetattr()`
- `tcsendbreak()`
- `tcsetattr()`

The following methods and variables in `<termios.h>` are not supported:

- `tcgetsid()`

4.63 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime_r()
- clock()
- clock_getres()
- clock_gettime()
- clock_nanosleep()
- clock_settime()
- ctime()
- ctime_r()
- difftime()
- gmtime()
- gmtime_r()
- localtime()
- localtime_r()
- mktime()
- nanosleep()
- strftime()
- strftime_l()
- strptime()
- time()
- timer_create()
- timer_delete()
- timer_getoverrun()
- timer_gettime()
- timer_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- `daylight`
- `getdate()`
- `getdate_err`

4.64 <trace.h>

The following methods and variables in <trace.h> are not supported:

- posix_trace_attr_destroy()
- posix_trace_attr_getclockres()
- posix_trace_attr_getcreatetime()
- posix_trace_attr_getgenversion()
- posix_trace_attr_getinherited()
- posix_trace_attr_getlogfullpolicy()
- posix_trace_attr_getlogsize()
- posix_trace_attr_getmaxdatasize()
- posix_trace_attr_getmaxsystemeventsize()
- posix_trace_attr_getmaxusereventsize()
- posix_trace_attr_getname()
- posix_trace_attr_getstreamfullpolicy()
- posix_trace_attr_getstreamsize()
- posix_trace_attr_init()
- posix_trace_attr_setinherited()
- posix_trace_attr_setlogfullpolicy()
- posix_trace_attr_setlogsize()
- posix_trace_attr_setmaxdatasize()
- posix_trace_attr_setname()
- posix_trace_attr_setstreamfullpolicy()
- posix_trace_attr_setstreamsize()
- posix_trace_clear()
- posix_trace_close()
- posix_trace_create()
- posix_trace_create_withlog()
- posix_trace_event()
- posix_trace_eventid_equal()
- posix_trace_eventid_get_name()
- posix_trace_eventid_open()
- posix_trace_eventset_add()
- posix_trace_eventset_del()
- posix_trace_eventset_empty()

- posix_trace_eventset_fill()
- posix_trace_eventset_ismember()
- posix_trace_eventtypelist_getnext_id()
- posix_trace_eventtypelist_rewind()
- posix_trace_flush()
- posix_trace_get_attr()
- posix_trace_get_filter()
- posix_trace_get_status()
- posix_trace_getnext_event()
- posix_trace_open()
- posix_trace_rewind()
- posix_trace_set_filter()
- posix_trace_shutdown()
- posix_trace_start()
- posix_trace_stop()
- posix_trace_timedgetnext_event()
- posix_trace_trid_eventid_open()
- posix_trace_trygetnext_event()

4.65 <ulimit.h>

The following methods and variables in <ulimit.h> are not supported:

- ulimit()

4.66 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- _exit()
- access()
- alarm()
- chdir()
- chown()
- close()
- dup()
- dup2()
- environ
- fchdir()
- fchown()
- fdatasync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getlogin()
- getlogin_r()
- getopt()
- getpgrp()
- getpid()
- getppid()
- getuid()
- isatty()
- lchown()
- link()
- lseek()

- `optarg`
- `opterr`
- `optind`
- `optopt`
- `pathconf()`
- `pause()`
- `pipe()`
- `pread()`
- `pwrite()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `swab()`
- `symlink()`
- `sync()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `truncate()`
- `ttyname()`
- `ttyname_r()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execlp()`

- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `faccessat()`
- `fchownat()`
- `fexecve()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `linkat()`
- `lockf()`
- `nice()`
- `readlinkat()`
- `setpgrp()`
- `setregid()`
- `setreuid()`
- `symlinkat()`
- `unlinkat()`

4.67 <utime.h>

The following methods and variables in `<utime.h>` are supported:

- `utime()`

4.68 <utmpx.h>

The following methods and variables in `<utmpx.h>` are not supported:

- `endutxent()`
- `getutxent()`
- `getutxid()`
- `getutxline()`
- `pututxline()`
- `setutxent()`

4.69 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsnrtowcs()
- mbsrtowcs()
- open_wmemstream()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcpcpy()
- wcpncpy()
- wcrtomb()
- wcscasecmp()
- wcscasecmp_l()

- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscoll_l()
- wcscpy()
- wcscspn()
- wcsdup()
- wcsftime()
- wcslen()
- wcsncasecmp()
- wcsncat()
- wcsncmp()
- wcsncpy()
- wcsnlen()
- wcsnrtombs()
- wcspbrk()
- wcsrchr()
- wcsrtombs()
- wcsspn()
- wcsstr()
- wcstod()
- wcstof()
- wcstok()
- wcstol()
- wcstold()
- wcstoll()
- wcstoul()
- wcstoull()
- wcswidth()
- wcsxfrm()
- wcsxfrm_l()
- wctob()
- wcwidth()

- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

The following methods and variables in `<wchar.h>` are not supported:

- `wcsncasemcp_l()`

4.70 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- iswalnum()
- iswalnum_l()
- iswalpha()
- iswalpha_l()
- iswblank()
- iswblank_l()
- iswcntrl()
- iswcntrl_l()
- iswctype()
- iswctype_l()
- iswdigit()
- iswdigit_l()
- iswgraph()
- iswgraph_l()
- iswlower()
- iswlower_l()
- iswprint()
- iswprint_l()
- iswpunct()
- iswpunct_l()
- iswspace()
- iswspace_l()
- iswupper()
- iswupper_l()
- iswxdigit()
- iswxdigit_l()
- towctrans()
- towctrans_l()
- towlower()
- towlower_l()
- towupper()
- towupper_l()

- `wctrans()`
- `wctrans_l()`
- `wctype()`
- `wctype_l()`

4.71 <wordexp.h>

The following methods and variables in `<wordexp.h>` are not supported:

- `wordexp()`
- `wordfree()`

CHAPTER

FIVE

POSIX-2003

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

5.1 Summary

The follow table summarizes alignment with the POSIX-2003 standard:

Supported	897
ENOSYS	19
Not supported	199

5.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio_cancel()
- aio_error()
- aio_fsync()
- aio_read()
- aio_return()
- aio_write()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio_suspend()
- lio_listio()

5.3 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

5.4 `<assert.h>`

The following methods and variables in `<assert.h>` are supported:

- `assert()`

5.5 <complex.h>

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- cexpl()
- cimag()
- cimagf()
- cimagl()
- clog()
- clogf()
- clogl()
- conj()
- conjf()
- conjl()
- cpow()
- cpowf()
- cpowl()
- cproj()
- cprojf()
- cprojl()
- creal()
- crealf()
- creall()
- csin()
- csinf()
- csinh()
- csinhf()
- csinhl()
- csinl()
- csqrt()
- csqrtf()
- csqrtl()
- ctan()
- ctanf()
- ctanh()
- ctanhf()
- ctanhl()
- ctanl()

5.6 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `_tolower()`
- `_toupper()`
- `isalnum()`
- `isalpha()`
- `isascii()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `islower_l()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `toascii()`
- `tolower()`
- `toupper()`

5.7 <dirent.h>

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `seekdir()`

5.8 <dlfcn.h>

The following methods and variables in `<dlfcn.h>` are supported:

- `dlclose()`
- `dlerror()`
- `dlopen()`
- `dlsym()`

5.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

5.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- posix_fadvise()
- posix_fallocate()
- posix_openpt()

5.11 <fenv.h>

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

5.12 <fmtmsg.h>

The following methods and variables in `<fmtmsg.h>` are not supported:

- `fmtmsg()`

5.13 <fnmatch.h>

The following methods and variables in `<fnmatch.h>` are supported:

- `fnmatch()`

5.14 <ftw.h>

The following methods and variables in `<ftw.h>` are not supported:

- `ftw()`
- `nftw()`

5.15 `<glob.h>`

The following methods and variables in `<glob.h>` are supported:

- `glob()`
- `globfree()`

5.16 <grp.h>

The following methods and variables in <grp.h> are supported:

- endgrent()
- getgrent()
- getgrgid()
- getgrgid_r()
- getgrnam()
- getgrnam_r()
- setgrent()

5.17 `<iconv.h>`

The following methods and variables in `<iconv.h>` are supported:

- `iconv()`
- `iconv_close()`
- `iconv_open()`

5.18 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

5.19 <langinfo.h>

The following methods and variables in `<langinfo.h>` are supported:

- `nl_langinfo()`

5.20 <libgen.h>

The following methods and variables in <libgen.h> are supported:

- basename()
- dirname()

5.21 `<locale.h>`

The following methods and variables in `<locale.h>` are supported:

- `localeconv()`
- `setlocale()`

5.22 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinhl()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfcl()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`
- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`

- log1pf()
- log1pl()
- log2()
- log2f()
- log2l()
- logb()
- logbf()
- logbl()
- logf()
- logl()
- lrint()
- lrintf()
- lrintl()
- lround()
- lroundf()
- lroundl()
- modf()
- modff()
- modfl()
- nan()
- nanf()
- nanl()
- nearbyint()
- nearbyintf()
- nearbyintl()
- nextafter()
- nextafterf()
- nextafterl()
- nexttoward()
- nexttowardl()
- pow()
- powf()
- powl()
- remainder()

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalb()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()

- tgammaf()
- tgammal()
- trunc()
- truncf()
- truncl()

The following methods and variables in <math.h> are not supported:

- fpclassify()
- isfinite()
- isgreater()
- isgreaterequal()
- isless()
- islessequal()
- islessgreater()
- isnormal()
- isunordered()
- j0()
- j1()
- jn()
- nexttowardf()
- signbit()
- signgam
- y0()
- y1()
- yn()

5.23 <monetary.h>

The following methods and variables in <monetary.h> are not supported:

- strfmon()

5.24 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_close()
- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_timedreceive()
- mq_timedsend()
- mq_unlink()

5.25 `<ndbm.h>`

The following methods and variables in `<ndbm.h>` are not supported:

- `dbm_clearerr()`
- `dbm_close()`
- `dbm_delete()`
- `dbm_error()`
- `dbm_fetch()`
- `dbm_firstkey()`
- `dbm_nextkey()`
- `dbm_open()`
- `dbm_store()`

5.26 `<net/if.h>`

The following methods and variables in `<net/if.h>` are supported:

- `if_freenameindex()`
- `if_indextoname()`
- `if_nameindex()`
- `if_nametoindex()`

5.27 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- endhostent()
- endnetent()
- endprotoent()
- endservent()
- freeaddrinfo()
- gai_strerror()
- getaddrinfo()
- gethostbyaddr()
- gethostbyname()
- gethostent()
- getnameinfo()
- getnetbyaddr()
- getnetbyname()
- getnetent()
- getprotobyname()
- getprotobynumber()
- getprotoent()
- getservbyname()
- getservbyport()
- getservent()
- h_errno
- sethostent()
- setnetent()
- setprotoent()
- setservent()

5.28 `<nl_types.h>`

The following methods and variables in `<nl_types.h>` are not supported:

- `catclose()`
- `catgets()`
- `catopen()`

5.29 `<poll.h>`

The following methods and variables in `<poll.h>` are not supported:

- `poll()`

5.30 <pthread.h>

The following methods and variables in `<pthread.h>` are supported:

- `pthread_attr_destroy()`
- `pthread_attr_getdetachstate()`
- `pthread_attr_getguardsize()`
- `pthread_attr_getinheritsched()`
- `pthread_attr_getschedparam()`
- `pthread_attr_getschedpolicy()`
- `pthread_attr_getscope()`
- `pthread_attr_getstack()`
- `pthread_attr_getstackaddr()`
- `pthread_attr_getstacksize()`
- `pthread_attr_init()`
- `pthread_attr_setdetachstate()`
- `pthread_attr_setguardsize()`
- `pthread_attr_setinheritsched()`
- `pthread_attr_setschedparam()`
- `pthread_attr_setschedpolicy()`
- `pthread_attr_setscope()`
- `pthread_attr_setstack()`
- `pthread_attr_setstackaddr()`
- `pthread_attr_setstacksize()`
- `pthread_barrier_destroy()`
- `pthread_barrier_init()`
- `pthread_barrier_wait()`
- `pthread_barrierattr_destroy()`
- `pthread_barrierattr_getpshared()`
- `pthread_barrierattr_init()`
- `pthread_barrierattr_setpshared()`
- `pthread_cancel()`
- `pthread_cleanup_pop()`
- `pthread_cleanup_push()`
- `pthread_cond_broadcast()`
- `pthread_cond_destroy()`

- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_getpshared()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_condattr_setpshared()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_getpshared()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()

- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_mutexattr_setpshared()
- pthread_mutexattr_settype()
- pthread_once()
- pthread_rwlock_destroy()
- pthread_rwlock_init()
- pthread_rwlock_rdlock()
- pthread_rwlock_timedrdlock()
- pthread_rwlock_timedwrlock()
- pthread_rwlock_tryrdlock()
- pthread_rwlock_trywrlock()
- pthread_rwlock_unlock()
- pthread_rwlock_wrlock()
- pthread_rwlockattr_destroy()
- pthread_rwlockattr_getpshared()
- pthread_rwlockattr_init()
- pthread_rwlockattr_setpshared()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()
- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()
- pthread_spin_destroy()
- pthread_spin_init()
- pthread_spin_lock()
- pthread_spin_trylock()
- pthread_spin_unlock()
- pthread_testcancel()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_atfork()
- pthread_getcpuclockid()

5.31 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- endpwent()
- getpwent()
- getpwnam()
- getpwnam_r()
- getpwuid()
- getpwuid_r()
- setpwent()

5.32 <regex.h>

The following methods and variables in `<regex.h>` are supported:

- `regcomp()`
- `regerror()`
- `regexec()`
- `regfree()`

5.33 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

5.34 <search.h>

The following methods and variables in <search.h> are supported:

- hcreate()
- hdestroy()
- hsearch()
- tdelete()
- tfind()
- tsearch()
- twalk()

The following methods and variables in <search.h> are not supported:

- insque()
- lfind()
- lsearch()
- remque()

5.35 <semaphore.h>

The following methods and variables in `<semaphore.h>` are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

5.36 <setjmp.h>

The following methods and variables in `<setjmp.h>` are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

The following methods and variables in `<setjmp.h>` are not supported:

- `_longjmp()`
- `_setjmp()`

5.37 <signal.h>

The following methods and variables in <signal.h> are supported:

- bsd_signal()
- kill()
- pthread_kill()
- pthread_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

The following methods and variables in <signal.h> are not supported:

- killpg()
- sighold()
- sigignore()
- siginterrupt()
- sigpause()
- sigrelse()
- sigset()

5.38 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawn_file_actions_addclose()
- posix_spawn_file_actions_adddup2()
- posix_spawn_file_actions_addopen()
- posix_spawn_file_actions_destroy()
- posix_spawn_file_actions_init()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getpgroup()
- posix_spawnattr_getschedparam()
- posix_spawnattr_getschedpolicy()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setpgroup()
- posix_spawnattr_setschedparam()
- posix_spawnattr_setschedpolicy()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()
- posix_spawnp()

5.39 <stdarg.h>

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

5.40 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- offsetof()

5.41 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc_unlocked()
- getchar()
- getchar_unlocked()
- gets()

- perror()
- printf()
- putc()
- putc_unlocked()
- putchar()
- putchar_unlocked()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tempnam()
- tmpfile()
- tmpnam()
- ungetc()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

The following methods and variables in <stdio.h> are not supported:

- pclose()
- popen()

5.42 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- _Exit()
- a64l()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- drand48()
- ecvt()
- erand48()
- exit()
- fcvt()
- free()
- gcvt()
- getenv()
- getsubopt()
- jrand48()
- l64a()
- labs()
- lcong48()
- ldiv()
- llabs()
- lldiv()
- lrand48()
- malloc()
- mblen()
- mbstowcs()

- `mbtowc()`
- `mkstemp()`
- `mrand48()`
- `nrand48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `rand()`
- `rand_r()`
- `random()`
- `realloc()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`
- `ptsname()`
- `setkey()`

- `setstate()`
- `unlockpt()`

5.43 `<string.h>`

The following methods and variables in `<string.h>` are supported:

- `memccpy()`
- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strdup()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

5.44 <strings.h>

The following methods and variables in <strings.h> are supported:

- bcmp()
- bcopy()
- ffs()
- index()
- rindex()
- strcasecmp()
- strncasecmp()

5.45 <stropts.h>

The following methods and variables in `<stropts.h>` are supported:

- `ioctl()`

The following methods and variables in `<stropts.h>` are not supported:

- `fattach()`
- `fdetach()`
- `getmsg()`
- `getpmsg()`
- `isastream()`
- `putmsg()`
- `putpmsg()`

5.46 <sys/ipc.h>

The following methods and variables in `<sys/ipc.h>` are not supported:

- `ftok()`

5.47 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mlock()
- mlockall()
- mmap()
- mprotect()
- msync()
- munlock()
- munlockall()
- munmap()
- posix_madvise()
- shm_open()
- shm_unlink()

The following methods and variables in <sys/mman.h> are not supported:

- posix_mem_offset()
- posix_typed_mem_get_info()
- posix_typed_mem_open()

5.48 <sys/msg.h>

The following methods and variables in <sys/msg.h> are not supported:

- msgctl()
- msgget()
- msgrcv()
- msgsnd()

5.49 <sys/resource.h>

The following methods and variables in `<sys/resource.h>` are supported:

- `getrusage()`

The following methods and variables in `<sys/resource.h>` are not supported:

- `getpriority()`
- `getrlimit()`
- `setpriority()`
- `setrlimit()`

5.50 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD_CLR()
- FD_ISSET()
- FD_SET()
- FD_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

5.51 <sys/sem.h>

The following methods and variables in <sys/sem.h> are not supported:

- semctl()
- semget()
- semop()

5.52 <sys/shm.h>

The following methods and variables in `<sys/shm.h>` are not supported:

- `shmat()`
- `shmctl()`
- `shmdt()`
- `shmget()`

5.53 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- recvmsg()
- send()
- sendmsg()
- sendto()
- setsockopt()
- shutdown()
- socket()
- socketpair()

The following methods and variables in <sys/socket.h> are not supported:

- sockatmark()

5.54 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- mknod()
- stat()
- umask()

5.55 <sys/statvfs.h>

The following methods and variables in `<sys/statvfs.h>` are supported:

- `statvfs()`

The following methods and variables in `<sys/statvfs.h>` are not supported:

- `fstatvfs()`

5.56 <sys/time.h>

The following methods and variables in `<sys/time.h>` are supported:

- `gettimeofday()`
- `utimes()`

The following methods in `<sys/time.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `getitimer()`
- `setitimer()`

5.57 <sys/times.h>

The following methods and variables in `<sys/times.h>` are supported:

- `times()`

5.58 <sys/uio.h>

The following methods and variables in `<sys/uio.h>` are supported:

- `readv()`
- `writev()`

5.59 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

5.60 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()
- waitpid()

The following methods and variables in <sys/wait.h> are not supported:

- waitid()

5.61 `<syslog.h>`

The following methods and variables in `<syslog.h>` are not supported:

- `closelog()`
- `openlog()`
- `setlogmask()`
- `syslog()`

5.62 <termios.h>

The following methods and variables in <termios.h> are supported:

- cfgetispeed()
- cfgetospeed()
- cfsetispeed()
- cfsetospeed()
- tcdrain()
- tcflow()
- tcflush()
- tcgetattr()
- tcsendbreak()
- tcsetattr()

The following methods and variables in <termios.h> are not supported:

- tcgetsid()

5.63 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime_r()
- clock()
- clock_getres()
- clock_gettime()
- clock_nanosleep()
- clock_settime()
- ctime()
- ctime_r()
- difftime()
- gmtime()
- gmtime_r()
- localtime()
- localtime_r()
- mktime()
- nanosleep()
- strftime()
- strptime()
- time()
- timer_create()
- timer_delete()
- timer_getoverrun()
- timer_gettime()
- timer_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- daylight

- `getdate()`
- `getdate_err`

5.64 `<trace.h>`

The following methods and variables in `<trace.h>` are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsize()`
- `posix_trace_attr_getmaxusereventsize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- posix_trace_eventset_fill()
- posix_trace_eventset_ismember()
- posix_trace_eventtypelist_getnext_id()
- posix_trace_eventtypelist_rewind()
- posix_trace_flush()
- posix_trace_get_attr()
- posix_trace_get_filter()
- posix_trace_get_status()
- posix_trace_getnext_event()
- posix_trace_open()
- posix_trace_rewind()
- posix_trace_set_filter()
- posix_trace_shutdown()
- posix_trace_start()
- posix_trace_stop()
- posix_trace_timedgetnext_event()
- posix_trace_trid_eventid_open()
- posix_trace_trygetnext_event()

5.65 `<ulimit.h>`

The following methods and variables in `<ulimit.h>` are not supported:

- `ulimit()`

5.66 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- _exit()
- access()
- alarm()
- chdir()
- chown()
- close()
- dup()
- dup2()
- environ
- fchdir()
- fchown()
- fdatasync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getlogin()
- getlogin_r()
- getopt()
- getpgrp()
- getpid()
- getppid()
- getuid()
- isatty()
- lchown()
- link()
- lseek()

- `optarg`
- `opterr`
- `optind`
- `optopt`
- `pathconf()`
- `pause()`
- `pipe()`
- `pread()`
- `pwrite()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `swab()`
- `symlink()`
- `sync()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `truncate()`
- `ttyname()`
- `ttyname_r()`
- `ualarm()`
- `unlink()`
- `usleep()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`

- `execle()`
- `execlp()`
- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `lockf()`
- `nice()`
- `setpgrp()`
- `setregid()`
- `setreuid()`

5.67 <utime.h>

The following methods and variables in `<utime.h>` are supported:

- `utime()`

5.68 <utmpx.h>

The following methods and variables in <utmpx.h> are not supported:

- endutxent()
- getutxent()
- getutxid()
- getutxline()
- pututxline()
- setutxent()

5.69 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcrtomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcscspn()

- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcsnlen()`
- `wcsnrtombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wctob()`
- `wcwidth()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

5.70 `<wctype.h>`

The following methods and variables in `<wctype.h>` are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

5.71 <wordexp.h>

The following methods and variables in `<wordexp.h>` are not supported:

- `wordexp()`
- `wordfree()`

CHAPTER

SIX

POSIX PSE51 - MINIMAL

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

6.1 Summary

The follow table summarizes alignment with the POSIX PSE51 - Minimal standard:

Supported	270
ENOSYS	2
Not supported	12

6.2 `<ctype.h>`

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

6.3 <errno.h>

The following methods and variables in `<errno.h>` are supported:

- `errno`

6.4 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- open()

6.5 <fenv.h>

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

6.6 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

6.7 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

6.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstackaddr()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstackaddr()
- pthread_attr_setstacksize()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_init()

- pthread_condattr_setclock()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_mutexattr_settype()
- pthread_once()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()
- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()

- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

6.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()

6.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

6.11 <setjmp.h>

The following methods and variables in `<setjmp.h>` are supported:

- `longjmp()`
- `setjmp()`

6.12 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread_kill()
- pthread_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

6.13 <stdarg.h>

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

6.14 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `getc()`
- `getc_unlocked()`
- `getchar()`
- `getchar_unlocked()`
- `gets()`
- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`

- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- ungetc()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

6.15 <stdlib.h>

The following methods and variables in `<stdlib.h>` are supported:

- `abort()`
- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `qsort()`
- `rand()`
- `rand_r()`
- `realloc()`
- `setenv()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`

6.16 `<string.h>`

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

6.17 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mlock()
- mlockall()
- mmap()
- munlock()
- munmap()
- shm_open()
- shm_unlink()

6.18 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

6.19 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `asctime()`
- `asctime_r()`
- `clock_getres()`
- `clock_gettime()`
- `clock_nanosleep()`
- `clock_settime()`
- `ctime()`
- `ctime_r()`
- `difftime()`
- `gmtime()`
- `gmtime_r()`
- `localtime()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `strftime()`
- `time()`
- `timer_create()`
- `timer_delete()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`
- `tzname`
- `tzset()`

6.20 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- alarm()
- close()
- environ
- fdatasync()
- fsync()
- pause()
- read()
- sysconf()
- write()

The following methods and variables in <unistd.h> are not supported:

- confstr()

CHAPTER

SEVEN

POSIX PSE52 - REAL-TIME CONTROLLER

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

7.1 Summary

The follow table summarizes alignment with the POSIX PSE52 - Real-Time Controller standard:

Supported	553
ENOSYS	2
Not supported	73

7.2 `<complex.h>`

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

7.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

7.4 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- closedir()
- opendir()
- readdir()
- readdir_r()
- rewinddir()

7.5 <errno.h>

The following methods and variables in `<errno.h>` are supported:

- `errno`

7.6 <fcntl.h>

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `fcntl()`
- `open()`

7.7 <fenv.h>

The following methods and variables in <fenv.h> are not supported:

- feclearexcept()
- fegetenv()
- fegetexceptflag()
- fegetround()
- feholdexcept()
- feraiseexcept()
- fesetenv()
- fesetexceptflag()
- fesetround()
- fetestexcept()
- feupdateenv()

7.8 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

7.9 <locale.h>

The following methods and variables in <locale.h> are supported:

- localeconv()
- setlocale()

7.10 <math.h>

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinhl()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- fmin()
- fminf()
- fminl()
- fmod()
- fmodf()
- fmodl()
- frexp()
- frexpf()
- frexpl()
- hypot()
- hypotf()
- hypotl()
- ilogb()
- ilogbf()
- ilogbl()
- isinf()
- isnan()
- ldexp()
- ldexpf()
- ldexpl()
- lgamma()
- lgammaf()
- lgammal()
- llrint()
- llrintf()
- llrintl()
- llround()
- llroundf()
- llroundl()
- log()
- log10()
- log10f()
- log10l()
- log1p()

- log1pf()
- log1pl()
- log2()
- log2f()
- log2l()
- logb()
- logbf()
- logbl()
- logf()
- logl()
- lrint()
- lrintf()
- lrintl()
- lround()
- lroundf()
- lroundl()
- modf()
- modff()
- modfl()
- nan()
- nanf()
- nanl()
- nearbyint()
- nearbyintf()
- nearbyintl()
- nextafter()
- nextafterf()
- nextafterl()
- nexttoward()
- nexttowardl()
- pow()
- powf()
- powl()
- remainder()

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- tgammal()
- trunc()
- truncf()
- truncl()

The following methods and variables in <math.h> are not supported:

- fpclassify()
- isfinite()
- isgreater()
- isgreaterequal()
- isless()
- islessequal()
- islessgreater()
- isnormal()
- isunordered()
- nexttowardf()
- signbit()

7.11 `<mqueue.h>`

The following methods and variables in `<mqueue.h>` are supported:

- `mq_close()`
- `mq_getattr()`
- `mq_notify()`
- `mq_open()`
- `mq_receive()`
- `mq_send()`
- `mq_setattr()`
- `mq_timedreceive()`
- `mq_timedsend()`
- `mq_unlink()`

7.12 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstackaddr()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstackaddr()
- pthread_attr_setstacksize()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_init()

- `pthread_condattr_setclock()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

- pthread_testcancel()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_atfork()
- pthread_getcpuclockid()

7.13 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()

7.14 <semaphore.h>

The following methods and variables in `<semaphore.h>` are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

7.15 <setjmp.h>

The following methods and variables in `<setjmp.h>` are supported:

- `longjmp()`
- `setjmp()`

7.16 <signal.h>

The following methods and variables in `<signal.h>` are supported:

- `kill()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `signal()`
- `sigpending()`
- `sigprocmask()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

7.17 <stdarg.h>

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

7.18 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fseeko()`
- `fsetpos()`
- `ftell()`
- `ftello()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `getc()`
- `getc_unlocked()`
- `getchar()`
- `getchar_unlocked()`
- `gets()`
- `perror()`

- printf()
- putc()
- putc_unlocked()
- putchar()
- putchar_unlocked()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tmpfile()
- tmpnam()
- ungetc()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

7.19 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- qsort()
- rand()
- rand_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

7.20 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok_r()
- strxfrm()

7.21 `<sys/mman.h>`

The following methods and variables in `<sys/mman.h>` are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `msync()`
- `munlock()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

7.22 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `fstat()`
- `mkdir()`
- `stat()`

7.23 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- uname()

7.24 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime_r()
- clock_getres()
- clock_gettime()
- clock_nanosleep()
- clock_settime()
- ctime()
- ctime_r()
- difftime()
- gmtime()
- gmtime_r()
- localtime()
- localtime_r()
- mktime()
- nanosleep()
- strftime()
- time()
- timer_create()
- timer_delete()
- timer_getoverrun()
- timer_gettime()
- timer_settime()
- tzname
- tzset()

7.25 `<trace.h>`

The following methods and variables in `<trace.h>` are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsize()`
- `posix_trace_attr_getmaxusereventsize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- posix_trace_eventset_fill()
- posix_trace_eventset_ismember()
- posix_trace_eventtypelist_getnext_id()
- posix_trace_eventtypelist_rewind()
- posix_trace_flush()
- posix_trace_get_attr()
- posix_trace_get_filter()
- posix_trace_get_status()
- posix_trace_getnext_event()
- posix_trace_open()
- posix_trace_rewind()
- posix_trace_set_filter()
- posix_trace_shutdown()
- posix_trace_start()
- posix_trace_stop()
- posix_trace_timedgetnext_event()
- posix_trace_trid_eventid_open()
- posix_trace_trygetnext_event()

7.26 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- close()
- dup()
- dup2()
- environ
- fdatasync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- link()
- lseek()
- pathconf()
- pause()
- read()
- rmdir()
- sysconf()
- unlink()
- write()

The following methods and variables in <unistd.h> are not supported:

- confstr()

7.27 <utime.h>

The following methods and variables in `<utime.h>` are supported:

- `utime()`

CHAPTER

EIGHT

POSIX PSE53 - DEDICATED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

8.1 Summary

The follow table summarizes alignment with the POSIX PSE53 - Dedicated standard:

Supported	639
ENOSYS	16
Not supported	96

8.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio_cancel()
- aio_error()
- aio_fsync()
- aio_read()
- aio_return()
- aio_write()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio_suspend()
- lio_listio()

8.3 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

8.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- assert()

8.5 `<complex.h>`

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- cexpl()
- cimag()
- cimagf()
- cimagl()
- clog()
- clogf()
- clogl()
- conj()
- conjf()
- conjl()
- cpow()
- cpowf()
- cpowl()
- cproj()
- cprojf()
- cprojl()
- creal()
- crealf()
- creall()
- csin()
- csinf()
- csinh()
- csinhf()
- csinhl()
- csinl()
- csqrt()
- csqrtf()
- csqrtl()
- ctan()
- ctanf()
- ctanh()
- ctanhf()
- ctanhl()
- ctanl()

8.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

8.7 `<dirent.h>`

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

8.8 <errno.h>

The following methods and variables in `<errno.h>` are supported:

- `errno`

8.9 <fcntl.h>

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `fcntl()`
- `open()`

8.10 <fenv.h>

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

8.11 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

8.12 <locale.h>

The following methods and variables in `<locale.h>` are supported:

- `localeconv()`
- `setlocale()`

8.13 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinhl()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfcl()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`
- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`

- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`
- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`
- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanhl()`
- `tanl()`
- `tgamma()`
- `tgammaf()`

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

8.14 <mqueue.h>

The following methods and variables in `<mqueue.h>` are supported:

- `mq_close()`
- `mq_getattr()`
- `mq_notify()`
- `mq_open()`
- `mq_receive()`
- `mq_send()`
- `mq_setattr()`
- `mq_timedreceive()`
- `mq_timedsend()`
- `mq_unlink()`

8.15 <net/if.h>

The following methods and variables in `<net/if.h>` are supported:

- `if_freenameindex()`
- `if_indextoname()`
- `if_nameindex()`
- `if_nametoindex()`

8.16 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- endhostent()
- endnetent()
- endprotoent()
- endservent()
- freeaddrinfo()
- gai_strerror()
- getaddrinfo()
- gethostent()
- getnameinfo()
- getnetbyaddr()
- getnetbyname()
- getnetent()
- getprotobyname()
- getprotobynumber()
- getprotoent()
- getservbyname()
- getservbyport()
- getservent()
- sethostent()
- setnetent()
- setprotoent()
- setservent()

8.17 `<pthread.h>`

The following methods and variables in `<pthread.h>` are supported:

- `pthread_attr_destroy()`
- `pthread_attr_getdetachstate()`
- `pthread_attr_getguardsize()`
- `pthread_attr_getinheritsched()`
- `pthread_attr_getschedparam()`
- `pthread_attr_getschedpolicy()`
- `pthread_attr_getscope()`
- `pthread_attr_getstack()`
- `pthread_attr_getstackaddr()`
- `pthread_attr_getstacksize()`
- `pthread_attr_init()`
- `pthread_attr_setdetachstate()`
- `pthread_attr_setguardsize()`
- `pthread_attr_setinheritsched()`
- `pthread_attr_setschedparam()`
- `pthread_attr_setschedpolicy()`
- `pthread_attr_setscope()`
- `pthread_attr_setstack()`
- `pthread_attr_setstackaddr()`
- `pthread_attr_setstacksize()`
- `pthread_cancel()`
- `pthread_cleanup_pop()`
- `pthread_cleanup_push()`
- `pthread_cond_broadcast()`
- `pthread_cond_destroy()`
- `pthread_cond_init()`
- `pthread_cond_signal()`
- `pthread_cond_timedwait()`
- `pthread_cond_wait()`
- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`

- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_condattr_setpshared()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_getpshared()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_mutexattr_setpshared()
- pthread_mutexattr_settype()
- pthread_once()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()

- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()
- pthread_testcancel()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_atfork()
- pthread_getcpuclockid()

8.18 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

8.19 <semaphore.h>

The following methods and variables in `<semaphore.h>` are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

8.20 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

8.21 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread_kill()
- pthread_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

8.22 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawn_file_actions_addclose()
- posix_spawn_file_actions_adddup2()
- posix_spawn_file_actions_addopen()
- posix_spawn_file_actions_destroy()
- posix_spawn_file_actions_init()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getpgroup()
- posix_spawnattr_getschedparam()
- posix_spawnattr_getschedpolicy()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setpgroup()
- posix_spawnattr_setschedparam()
- posix_spawnattr_setschedpolicy()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()
- posix_spawnp()

8.23 <stdarg.h>

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

8.24 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fseeko()`
- `fsetpos()`
- `ftell()`
- `ftello()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `getc()`
- `getc_unlocked()`
- `getchar()`
- `getchar_unlocked()`
- `gets()`
- `perror()`

- printf()
- putc()
- putc_unlocked()
- putchar()
- putchar_unlocked()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tmpfile()
- tmpnam()
- ungetc()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

8.25 <stdlib.h>

The following methods and variables in `<stdlib.h>` are supported:

- `_Exit()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `qsort()`
- `rand()`
- `rand_r()`
- `realloc()`
- `setenv()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`

- `unsetenv()`

8.26 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok_r()
- strxfrm()

8.27 <sys/mman.h>

The following methods and variables in `<sys/mman.h>` are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

8.28 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD_CLR()
- FD_ISSET()
- FD_SET()
- FD_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

8.29 <sys/socket.h>

The following methods and variables in `<sys/socket.h>` are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in `<sys/socket.h>` are not supported:

- `sockatmark()`

8.30 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `fstat()`
- `mkdir()`
- `stat()`

8.31 <sys/time.h>

The following methods and variables in `<sys/time.h>` are supported:

- `utimes()`

8.32 <sys/times.h>

The following methods and variables in <sys/times.h> are supported:

- times()

8.33 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- uname()

8.34 <sys/wait.h>

The following methods and variables in `<sys/wait.h>` are supported:

- `wait()`

8.35 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime_r()
- clock()
- clock_getres()
- clock_gettime()
- clock_nanosleep()
- clock_settime()
- ctime()
- ctime_r()
- difftime()
- gmtime()
- gmtime_r()
- localtime()
- localtime_r()
- mktime()
- nanosleep()
- strftime()
- time()
- timer_create()
- timer_delete()
- timer_getoverrun()
- timer_gettime()
- timer_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock_getcpuclockid()

8.36 <trace.h>

The following methods and variables in `<trace.h>` are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsize()`
- `posix_trace_attr_getmaxusereventsize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- posix_trace_eventset_fill()
- posix_trace_eventset_ismember()
- posix_trace_eventtypelist_getnext_id()
- posix_trace_eventtypelist_rewind()
- posix_trace_flush()
- posix_trace_get_attr()
- posix_trace_get_filter()
- posix_trace_get_status()
- posix_trace_getnext_event()
- posix_trace_open()
- posix_trace_rewind()
- posix_trace_set_filter()
- posix_trace_shutdown()
- posix_trace_start()
- posix_trace_stop()
- posix_trace_timedgetnext_event()
- posix_trace_trid_eventid_open()
- posix_trace_trygetnext_event()

8.37 <unistd.h>

The following methods and variables in `<unistd.h>` are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `gethostname()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`
- `setsid()`
- `sleep()`
- `sysconf()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`

- `execle()`
- `execlp()`
- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`

8.38 <utime.h>

The following methods and variables in `<utime.h>` are supported:

- `utime()`

CHAPTER

NINE

POSIX PSE54 - MULTIPURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

9.1 Summary

The follow table summarizes alignment with the POSIX PSE54 - Multipurpose standard:

Supported	783
ENOSYS	17
Not supported	106

9.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio_cancel()
- aio_error()
- aio_fsync()
- aio_read()
- aio_return()
- aio_write()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio_suspend()
- lio_listio()

9.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- htonl()
- htons()
- inet_addr()
- inet_ntoa()
- inet_ntop()
- inet_pton()
- ntohl()
- ntohs()

9.4 <assert.h>

The following methods and variables in `<assert.h>` are supported:

- `assert()`

9.5 <complex.h>

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- cexpl()
- cimag()
- cimagf()
- cimagl()
- clog()
- clogf()
- clogl()
- conj()
- conjf()
- conjl()
- cpow()
- cpowf()
- cpowl()
- cproj()
- cprojf()
- cprojl()
- creal()
- crealf()
- creall()
- csin()
- csinf()
- csinh()
- csinhf()
- csinhl()
- csinl()
- csqrt()
- csqrtf()
- csqrtl()
- ctan()
- ctanf()
- ctanh()
- ctanhf()
- ctanhl()
- ctanl()

9.6 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

9.7 `<dirent.h>`

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

9.8 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- dlclose()
- dlerror()
- dlopen()
- dlsym()

9.9 <errno.h>

The following methods and variables in `<errno.h>` are supported:

- `errno`

9.10 <fcntl.h>

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `fcntl()`
- `open()`

The following methods and variables in `<fcntl.h>` are not supported:

- `posix_fadvise()`
- `posix_fallocate()`

9.11 `<fenv.h>`

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

9.12 <fnmatch.h>

The following methods and variables in `<fnmatch.h>` are supported:

- `fnmatch()`

9.13 `<glob.h>`

The following methods and variables in `<glob.h>` are supported:

- `glob()`
- `globfree()`

9.14 <grp.h>

The following methods and variables in <grp.h> are supported:

- getgrgid()
- getgrgid_r()
- getgrnam()
- getgrnam_r()

9.15 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

9.16 <locale.h>

The following methods and variables in <locale.h> are supported:

- localeconv()
- setlocale()

9.17 <math.h>

The following methods and variables in <math.h> are supported:

- acos()
- acosf()
- acosh()
- acoshf()
- acoshl()
- acosl()
- asin()
- asinf()
- asinh()
- asinhf()
- asinhl()
- asinl()
- atan()
- atan2()
- atan2f()
- atan2l()
- atanf()
- atanh()
- atanhf()
- atanhl()
- atanl()
- cbrt()
- cbrtf()
- cbrtl()
- ceil()
- ceilf()
- ceill()
- copysign()
- copysignf()
- copysignl()
- cos()
- cosf()

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfcl()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- fmin()
- fminf()
- fminl()
- fmod()
- fmodf()
- fmodl()
- frexp()
- frexpf()
- frexpl()
- hypot()
- hypotf()
- hypotl()
- ilogb()
- ilogbf()
- ilogbl()
- isinf()
- isnan()
- ldexp()
- ldexpf()
- ldexpl()
- lgamma()
- lgammaf()
- lgammal()
- llrint()
- llrintf()
- llrintl()
- llround()
- llroundf()
- llroundl()
- log()
- log10()
- log10f()
- log10l()
- log1p()

- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`
- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`
- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanhl()`
- `tanl()`
- `tgamma()`
- `tgammaf()`

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

9.18 `<mqueue.h>`

The following methods and variables in `<mqueue.h>` are supported:

- `mq_close()`
- `mq_getattr()`
- `mq_notify()`
- `mq_open()`
- `mq_receive()`
- `mq_send()`
- `mq_setattr()`
- `mq_timedreceive()`
- `mq_timedsend()`
- `mq_unlink()`

9.19 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if_freenameindex()
- if_indextoname()
- if_nameindex()
- if_nametoindex()

9.20 `<netdb.h>`

The following methods and variables in `<netdb.h>` are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

9.21 `<pthread.h>`

The following methods and variables in `<pthread.h>` are supported:

- `pthread_attr_destroy()`
- `pthread_attr_getdetachstate()`
- `pthread_attr_getguardsize()`
- `pthread_attr_getinheritsched()`
- `pthread_attr_getschedparam()`
- `pthread_attr_getschedpolicy()`
- `pthread_attr_getscope()`
- `pthread_attr_getstack()`
- `pthread_attr_getstackaddr()`
- `pthread_attr_getstacksize()`
- `pthread_attr_init()`
- `pthread_attr_setdetachstate()`
- `pthread_attr_setguardsize()`
- `pthread_attr_setinheritsched()`
- `pthread_attr_setschedparam()`
- `pthread_attr_setschedpolicy()`
- `pthread_attr_setscope()`
- `pthread_attr_setstack()`
- `pthread_attr_setstackaddr()`
- `pthread_attr_setstacksize()`
- `pthread_cancel()`
- `pthread_cleanup_pop()`
- `pthread_cleanup_push()`
- `pthread_cond_broadcast()`
- `pthread_cond_destroy()`
- `pthread_cond_init()`
- `pthread_cond_signal()`
- `pthread_cond_timedwait()`
- `pthread_cond_wait()`
- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`

- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`

- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()
- pthread_testcancel()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_atfork()
- pthread_getcpuclockid()

9.22 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- getpwnam()
- getpwnam_r()
- getpwuid()
- getpwuid_r()

9.23 `<regex.h>`

The following methods and variables in `<regex.h>` are supported:

- `regcomp()`
- `regerror()`
- `regexec()`
- `regfree()`

9.24 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

9.25 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

9.26 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

9.27 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread_kill()
- pthread_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

9.28 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawn_file_actions_addclose()
- posix_spawn_file_actions_adddup2()
- posix_spawn_file_actions_addopen()
- posix_spawn_file_actions_destroy()
- posix_spawn_file_actions_init()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getpgroup()
- posix_spawnattr_getschedparam()
- posix_spawnattr_getschedpolicy()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setpgroup()
- posix_spawnattr_setschedparam()
- posix_spawnattr_setschedpolicy()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()
- posix_spawnp()

9.29 <stdarg.h>

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

9.30 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `ctermid()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fseeko()`
- `fsetpos()`
- `ftell()`
- `ftello()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `getc()`
- `getc_unlocked()`
- `getchar()`
- `getchar_unlocked()`
- `gets()`

- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

The following methods and variables in <`stdio.h`> are not supported:

- `pclose()`
- `popen()`

9.31 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- _Exit()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbstowcs()
- mbtowc()
- posix_memalign()
- qsort()
- rand()
- rand_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()

- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `system()`

9.32 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

9.33 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mlock()
- mlockall()
- mmap()
- mprotect()
- msync()
- munlock()
- munmap()
- posix_madvise()
- shm_open()
- shm_unlink()

9.34 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

The following methods and variables in `<sys/select.h>` are not supported:

- `pselect()`

9.35 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- recvmsg()
- send()
- sendmsg()
- sendto()
- setsockopt()
- shutdown()
- socket()
- socketpair()

The following methods and variables in <sys/socket.h> are not supported:

- sockatmark()

9.36 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `chmod()`
- `fchmod()`
- `fstat()`
- `lstat()`
- `mkdir()`
- `mkfifo()`
- `stat()`
- `umask()`

9.37 <sys/time.h>

The following methods and variables in `<sys/time.h>` are supported:

- `utimes()`

9.38 <sys/times.h>

The following methods and variables in `<sys/times.h>` are supported:

- `times()`

9.39 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

9.40 `<sys/wait.h>`

The following methods and variables in `<sys/wait.h>` are supported:

- `wait()`

9.41 <syslog.h>

The following methods and variables in `<syslog.h>` are not supported:

- `closelog()`
- `openlog()`
- `setlogmask()`
- `syslog()`

9.42 `<termios.h>`

The following methods and variables in `<termios.h>` are supported:

- `cfgetispeed()`
- `cfgetospeed()`
- `cfsetispeed()`
- `cfsetospeed()`
- `tcdrain()`
- `tcflow()`
- `tcflush()`
- `tcgetattr()`
- `tcsendbreak()`
- `tcsetattr()`

9.43 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime_r()
- clock()
- clock_getres()
- clock_gettime()
- clock_nanosleep()
- clock_settime()
- ctime()
- ctime_r()
- difftime()
- gmtime()
- gmtime_r()
- localtime()
- localtime_r()
- mktime()
- nanosleep()
- strftime()
- time()
- timer_create()
- timer_delete()
- timer_getoverrun()
- timer_gettime()
- timer_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock_getcpuclockid()

9.44 <trace.h>

The following methods and variables in <trace.h> are not supported:

- posix_trace_attr_destroy()
- posix_trace_attr_getclockres()
- posix_trace_attr_getcreatetime()
- posix_trace_attr_getgenversion()
- posix_trace_attr_getinherited()
- posix_trace_attr_getlogfullpolicy()
- posix_trace_attr_getlogsize()
- posix_trace_attr_getmaxdatasize()
- posix_trace_attr_getmaxsystemeventsize()
- posix_trace_attr_getmaxusereventsize()
- posix_trace_attr_getname()
- posix_trace_attr_getstreamfullpolicy()
- posix_trace_attr_getstreamsize()
- posix_trace_attr_init()
- posix_trace_attr_setinherited()
- posix_trace_attr_setlogfullpolicy()
- posix_trace_attr_setlogsize()
- posix_trace_attr_setmaxdatasize()
- posix_trace_attr_setname()
- posix_trace_attr_setstreamfullpolicy()
- posix_trace_attr_setstreamsize()
- posix_trace_clear()
- posix_trace_close()
- posix_trace_create()
- posix_trace_create_withlog()
- posix_trace_event()
- posix_trace_eventid_equal()
- posix_trace_eventid_get_name()
- posix_trace_eventid_open()
- posix_trace_eventset_add()
- posix_trace_eventset_del()
- posix_trace_eventset_empty()

- posix_trace_eventset_fill()
- posix_trace_eventset_ismember()
- posix_trace_eventtypelist_getnext_id()
- posix_trace_eventtypelist_rewind()
- posix_trace_flush()
- posix_trace_get_attr()
- posix_trace_get_filter()
- posix_trace_get_status()
- posix_trace_getnext_event()
- posix_trace_open()
- posix_trace_rewind()
- posix_trace_set_filter()
- posix_trace_shutdown()
- posix_trace_start()
- posix_trace_stop()
- posix_trace_timedgetnext_event()
- posix_trace_trid_eventid_open()
- posix_trace_trygetnext_event()

9.45 <unistd.h>

The following methods and variables in `<unistd.h>` are supported:

- `_exit()`
- `access()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `link()`
- `lseek()`
- `optarg`
- `opterr`
- `optind`

- optopt
- pathconf()
- pipe()
- read()
- readlink()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setpgid()
- setsid()
- setuid()
- sleep()
- symlink()
- sysconf()
- tcgetpgrp()
- tcsetpgrp()
- ttyname()
- ttyname_r()
- unlink()
- write()

The following methods in <unistd.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- execl()
- execle()
- execlp()
- execv()
- execve()
- execvp()
- fork()

The following methods and variables in <unistd.h> are not supported:

- confstr()

9.46 <utime.h>

The following methods and variables in `<utime.h>` are supported:

- `utime()`

9.47 <wchar.h>

The following methods and variables in `<wchar.h>` are supported:

- `btowc()`
- `fgetwc()`
- `fgetws()`
- `fputwc()`
- `fputws()`
- `fwide()`
- `fwprintf()`
- `fwscanf()`
- `getwc()`
- `getwchar()`
- `mbrlen()`
- `mbrtowc()`
- `mbsinit()`
- `mbsrtowcs()`
- `putwc()`
- `putwchar()`
- `swprintf()`
- `swscanf()`
- `ungetwc()`
- `vfwprintf()`
- `vfwscanf()`
- `vswprintf()`
- `vswscanf()`
- `vwprintf()`
- `vwscanf()`
- `wcrtomb()`
- `wcscat()`
- `wcschr()`
- `wcscmp()`
- `wcscoll()`
- `wcscpy()`
- `wcscspn()`

- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

9.48 <wctype.h>

The following methods and variables in `<wctype.h>` are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

9.49 <wordexp.h>

The following methods and variables in <wordexp.h> are not supported:

- wordexp()
- wordfree()

CHAPTER

TEN

C99 STANDARD LIBRARY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

10.1 Summary

The follow table summarizes alignment with the C99 Standard Library standard:

Supported	463
ENOSYS	0
Not supported	22

10.2 <assert.h>

The following methods and variables in <assert.h> are supported:

- assert()

10.3 <complex.h>

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- cexpl()
- cimag()
- cimagf()
- cimagl()
- clog()
- clogf()
- clogl()
- conj()
- conjf()
- conjl()
- cpow()
- cpowf()
- cpowl()
- cproj()
- cprojf()
- cprojl()
- creal()
- crealf()
- creall()
- csin()
- csinf()
- csinh()
- csinhf()
- csinhl()
- csinl()
- csqrt()
- csqrtf()
- csqrtl()
- ctan()
- ctanf()
- ctanh()
- ctanhf()
- ctanhl()
- ctanl()

10.4 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

10.5 `<errno.h>`

The following methods and variables in `<errno.h>` are supported:

- `errno`

10.6 <fenv.h>

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

10.7 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

10.8 <locale.h>

The following methods and variables in `<locale.h>` are supported:

- `localeconv()`
- `setlocale()`

10.9 <math.h>

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinhl()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfcl()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`
- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`

- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`
- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`
- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanhl()`
- `tanl()`
- `tgamma()`
- `tgammaf()`

- tgammal()
- trunc()
- truncf()
- truncl()

The following methods and variables in `<math.h>` are not supported:

- fpclassify()
- isfinite()
- isgreater()
- isgreaterequal()
- isless()
- islessequal()
- islessgreater()
- isnormal()
- isunordered()
- nexttowardf()
- signbit()

10.10 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()

10.11 `<signal.h>`

The following methods and variables in `<signal.h>` are supported:

- `raise()`
- `signal()`

10.12 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va_arg()
- va_copy()
- va_end()
- va_start()

10.13 <stddef.h>

The following methods and variables in `<stddef.h>` are supported:

- `offsetof()`

10.14 `<stdio.h>`

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fileno()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fsetpos()`
- `ftell()`
- `fwrite()`
- `getc()`
- `getchar()`
- `gets()`
- `perror()`
- `printf()`
- `putc()`
- `putchar()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`

- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tmpfile()
- tmpnam()
- ungetc()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

10.15 <stdlib.h>

The following methods and variables in `<stdlib.h>` are supported:

- `_Exit()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `qsort()`
- `rand()`
- `realloc()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`

- `wcstombs()`
- `wctomb()`

10.16 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strxfrm()`

10.17 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `asctime()`
- `clock()`
- `ctime()`
- `difftime()`
- `gmtime()`
- `localtime()`
- `mktime()`
- `strftime()`
- `time()`

10.18 `<wchar.h>`

The following methods and variables in `<wchar.h>` are supported:

- `btowc()`
- `fgetwc()`
- `fgetws()`
- `fputwc()`
- `fputws()`
- `fwide()`
- `fwprintf()`
- `fwscanf()`
- `getwc()`
- `getwchar()`
- `mbrlen()`
- `mbrtowc()`
- `mbsinit()`
- `mbsrtowcs()`
- `putwc()`
- `putwchar()`
- `swprintf()`
- `swscanf()`
- `ungetwc()`
- `vfwprintf()`
- `vfwscanf()`
- `vswprintf()`
- `vswscanf()`
- `vwprintf()`
- `vwscanf()`
- `wcrtomb()`
- `wcscat()`
- `wcschr()`
- `wcscmp()`
- `wcscoll()`
- `wcscpy()`
- `wcscspn()`

- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

10.19 `<wctype.h>`

The following methods and variables in `<wctype.h>` are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

CHAPTER

ELEVEN

C11 STANDARD LIBRARY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

11.1 Summary

The follow table summarizes alignment with the C11 Standard Library standard:

Supported	486
ENOSYS	0
Not supported	22

11.2 <assert.h>

The following methods and variables in <assert.h> are supported:

- assert()

11.3 <complex.h>

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- cexpl()
- cimag()
- cimagf()
- cimagl()
- clog()
- clogf()
- clogl()
- conj()
- conjf()
- conjl()
- cpow()
- cpowf()
- cpowl()
- cproj()
- cprojf()
- cprojl()
- creal()
- crealf()
- creall()
- csin()
- csinf()
- csinh()
- csinhf()
- csinhl()
- csinl()
- csqrt()
- csqrtf()
- csqrtl()
- ctan()
- ctanf()
- ctanh()
- ctanhf()
- ctanhl()
- ctanl()

11.4 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

11.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

11.6 <fenv.h>

The following methods and variables in <fenv.h> are not supported:

- feclearexcept()
- fegetenv()
- fegetexceptflag()
- fegetround()
- feholdexcept()
- feraiseexcept()
- fesetenv()
- fesetexceptflag()
- fesetround()
- fetestexcept()
- feupdateenv()

11.7 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- imaxabs()
- imaxdiv()
- strtoimax()
- strtoumax()
- wcstoimax()
- wcstoumax()

11.8 <locale.h>

The following methods and variables in `<locale.h>` are supported:

- `localeconv()`
- `setlocale()`

11.9 <math.h>

The following methods and variables in <math.h> are supported:

- acos()
- acosf()
- acosh()
- acoshf()
- acoshl()
- acosl()
- asin()
- asinf()
- asinh()
- asinhf()
- asinhl()
- asinl()
- atan()
- atan2()
- atan2f()
- atan2l()
- atanf()
- atanh()
- atanhf()
- atanhl()
- atanl()
- cbrt()
- cbrtf()
- cbrtl()
- ceil()
- ceilf()
- ceill()
- copysign()
- copysignf()
- copysignl()
- cos()
- cosf()

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfcl()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- fmin()
- fminf()
- fminl()
- fmod()
- fmodf()
- fmodl()
- frexp()
- frexpf()
- frexpl()
- hypot()
- hypotf()
- hypotl()
- ilogb()
- ilogbf()
- ilogbl()
- isinf()
- isnan()
- ldexp()
- ldexpf()
- ldexpl()
- lgamma()
- lgammaf()
- lgammal()
- llrint()
- llrintf()
- llrintl()
- llround()
- llroundf()
- llroundl()
- log()
- log10()
- log10f()
- log10l()
- log1p()

- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`
- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

11.10 `<setjmp.h>`

The following methods and variables in `<setjmp.h>` are supported:

- `longjmp()`
- `setjmp()`

11.11 <signal.h>

The following methods and variables in `<signal.h>` are supported:

- `raise()`
- `signal()`

11.12 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va_arg()
- va_copy()
- va_end()
- va_start()

11.13 <stddef.h>

The following methods and variables in `<stddef.h>` are supported:

- `offsetof()`

11.14 `<stdio.h>`

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fsetpos()`
- `ftell()`
- `fwrite()`
- `getc()`
- `getchar()`
- `perror()`
- `printf()`
- `putc()`
- `putchar()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`

- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

11.15 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- _Exit()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbstowcs()
- mbtowc()
- qsort()
- rand()
- realloc()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()

- `wcstombs()`
- `wctomb()`

11.16 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strxfrm()`

11.17 <threads.h>

The following methods and variables in `<threads.h>` are supported:

- `call_once()`
- `cnd_broadcast()`
- `cnd_destroy()`
- `cnd_init()`
- `cnd_signal()`
- `cnd_timedwait()`
- `cnd_wait()`
- `mtx_destroy()`
- `mtx_init()`
- `mtx_lock()`
- `mtx_timedlock()`
- `mtx_trylock()`
- `mtx_unlock()`
- `thrd_create()`
- `thrd_current()`
- `thrd_detach()`
- `thrd_equal()`
- `thrd_exit()`
- `thrd_join()`
- `thrd_sleep()`
- `thrd_yield()`
- `tss_create()`
- `tss_delete()`
- `tss_get()`
- `tss_set()`

11.18 <time.h>

The following methods and variables in `<time.h>` are supported:

- `asctime()`
- `clock()`
- `ctime()`
- `difftime()`
- `gmtime()`
- `localtime()`
- `mktime()`
- `strftime()`
- `time()`

11.19 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcrtomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcscspn()

- wcsftime()
- wcslen()
- wcsncat()
- wcsncmp()
- wcsncpy()
- wcspbrk()
- wcsrchr()
- wcsrtombs()
- wcsspn()
- wcsstr()
- wcstod()
- wcstof()
- wcstok()
- wcstol()
- wcstold()
- wcstoll()
- wcstoul()
- wcstoull()
- wcsxfrm()
- wctob()
- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

11.20 <wctype.h>

The following methods and variables in `<wctype.h>` are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

CHAPTER

TWELVE

FACE 2.1 SECURITY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

12.1 Summary

The follow table summarizes alignment with the FACE 2.1 Security standard:

Supported	162
ENOSYS	1
Not supported	0

12.2 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

12.3 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

12.4 `<devctl.h>`

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

12.5 `<errno.h>`

The following methods and variables in `<errno.h>` are supported:

- `errno`

12.6 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

12.7 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

12.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_init()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_create()
- pthread_equal()
- pthread_getschedparam()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_once()
- pthread_self()
- pthread_setschedparam()
- pthread_setschedprio()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- `pthread_getcpuclockid()`

12.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_yield()

12.10 <semaphore.h>

The following methods and variables in `<semaphore.h>` are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

12.11 <signal.h>

The following methods and variables in `<signal.h>` are supported:

- `pthread_sigmask()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `sigpending()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

12.12 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abs()
- atof()
- atoi()
- atol()
- div()
- labs()
- ldiv()
- malloc()
- rand_r()
- strtod()
- strtol()
- strtoul()

12.13 `<string.h>`

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

12.14 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mmap()
- shm_open()

12.15 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

12.16 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `stat()`

12.17 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `clock_getres()`
- `clock_gettime()`
- `clock_settime()`
- `nanosleep()`
- `timer_create()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`

12.18 <unistd.h>

The following methods and variables in `<unistd.h>` are supported:

- `alarm()`
- `ftruncate()`
- `pause()`

CHAPTER

THIRTEEN

FACE 2.1 SAFETY BASE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

13.1 Summary

The follow table summarizes alignment with the FACE 2.1 Safety Base standard:

Supported	245
ENOSYS	1
Not supported	0

13.2 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

13.3 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

13.4 `<devctl.h>`

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

13.5 <dirent.h>

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

13.6 <errno.h>

The following methods and variables in `<errno.h>` are supported:

- `errno`

13.7 <fcntl.h>

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `open()`

13.8 <math.h>

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

13.9 `<mqueue.h>`

The following methods and variables in `<mqueue.h>` are supported:

- `mq_getattr()`
- `mq_notify()`
- `mq_open()`
- `mq_receive()`
- `mq_send()`
- `mq_setattr()`
- `mq_timedreceive()`
- `mq_timedsend()`

13.10 `<netdb.h>`

The following methods and variables in `<netdb.h>` are supported:

- `freeaddrinfo()`
- `getaddrinfo()`
- `getnameinfo()`

13.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstacksize()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_create()
- pthread_equal()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_key_create()
- pthread_mutex_init()
- pthread_mutex_lock()

- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_getcpuclockid()`

13.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- `sched_get_priority_max()`
- `sched_get_priority_min()`
- `sched_yield()`

13.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_wait()

13.14 `<signal.h>`

The following methods and variables in `<signal.h>` are supported:

- `pthread_sigmask()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `sigpending()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

13.15 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgets()`
- `fileno()`
- `fopen()`
- `fprintf()`
- `fread()`
- `freopen()`
- `fseek()`
- `fseeko()`
- `ftell()`
- `ftello()`
- `fwrite()`
- `remove()`
- `rename()`
- `snprintf()`

13.16 `<stdlib.h>`

The following methods and variables in `<stdlib.h>` are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `calloc()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

13.17 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

13.18 <sys/mman.h>

The following methods and variables in `<sys/mman.h>` are supported:

- `mmap()`
- `shm_open()`

13.19 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

13.20 <sys/socket.h>

The following methods and variables in `<sys/socket.h>` are supported:

- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`

13.21 `<sys/stat.h>`

The following methods and variables in `<sys/stat.h>` are supported:

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

13.22 <time.h>

The following methods and variables in `<time.h>` are supported:

- `asctime_r()`
- `clock_getres()`
- `clock_gettime()`
- `clock_settime()`
- `ctime_r()`
- `difftime()`
- `gmtime_r()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `time()`
- `timer_create()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`
- `tzname`
- `tzset()`

13.23 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- close()
- fsync()
- ftruncate()
- getcwd()
- gethostname()
- link()
- lseek()
- pause()
- read()
- rmdir()
- unlink()
- write()

CHAPTER

FOURTEEN

FACE 2.1 SAFETY EXTENDED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

14.1 Summary

The follow table summarizes alignment with the FACE 2.1 Safety Extended standard:

Supported	315
ENOSYS	11
Not supported	9

14.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- htonl()
- htons()
- inet_ntop()
- inet_pton()
- ntohl()
- ntohs()

14.3 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

14.4 `<devctl.h>`

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

14.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- closedir()
- opendir()
- readdir()
- readdir_r()
- rewinddir()

14.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

14.7 `<fcntl.h>`

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `fcntl()`
- `open()`

14.8 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

14.9 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_timedreceive()
- mq_timedsend()

14.10 `<netdb.h>`

The following methods and variables in `<netdb.h>` are supported:

- `freeaddrinfo()`
- `getaddrinfo()`
- `getnameinfo()`

14.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstacksize()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_create()

- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_once()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()
- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_atfork()
- pthread_getcpuclockid()

14.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

14.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

14.14 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- siglongjmp()
- sigsetjmp()

14.15 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread_kill()
- pthread_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

14.16 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()

14.17 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va_arg()
- va_end()
- va_start()

14.18 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- remove()
- rename()
- snprintf()
- sscanf()
- vfprintf()
- vsnprintf()

14.19 <stdlib.h>

The following methods and variables in `<stdlib.h>` are supported:

- `_Exit()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `realloc()`
- `strtod()`
- `strtol()`
- `strtoul()`

14.20 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok_r()

14.21 `<sys/mman.h>`

The following methods and variables in `<sys/mman.h>` are supported:

- `mmap()`
- `shm_open()`

14.22 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

14.23 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

14.24 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `chmod()`
- `fstat()`
- `lstat()`
- `mkdir()`
- `mkfifo()`
- `stat()`
- `umask()`

14.25 `<sys/times.h>`

The following methods and variables in `<sys/times.h>` are supported:

- `times()`

14.26 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

14.27 <sys/wait.h>

The following methods and variables in `<sys/wait.h>` are supported:

- `waitpid()`

14.28 <time.h>

The following methods and variables in `<time.h>` are supported:

- `asctime_r()`
- `clock()`
- `clock_getres()`
- `clock_gettime()`
- `clock_settime()`
- `ctime_r()`
- `difftime()`
- `gmtime_r()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `strftime()`
- `time()`
- `timer_create()`
- `timer_delete()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`
- `tzname`
- `tzset()`

14.29 <unistd.h>

The following methods and variables in `<unistd.h>` are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup2()`
- `environ`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `link()`
- `lseek()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setuid()`
- `sleep()`
- `sysconf()`

- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`

CHAPTER

FIFTEEN

FACE 2.1 GENERAL PURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

15.1 Summary

The follow table summarizes alignment with the FACE 2.1 General Purpose standard:

Supported	752
ENOSYS	14
Not supported	46

15.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio_cancel()
- aio_error()
- aio_fsync()
- aio_read()
- aio_return()
- aio_write()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio_suspend()
- lio_listio()

15.3 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

15.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- assert()

15.5 `<complex.h>`

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- cexpl()
- cimag()
- cimagf()
- cimagl()
- clog()
- clogf()
- clogl()
- conj()
- conjf()
- conjl()
- cpow()
- cpowf()
- cpowl()
- cproj()
- cprojf()
- cprojl()
- creal()
- crealf()
- creall()
- csin()
- csinf()
- csinh()
- csinhf()
- csinhl()
- csinl()
- csqrt()
- csqrtf()
- csqrtl()
- ctan()
- ctanf()
- ctanh()
- ctanhf()
- ctanhl()
- ctanl()

15.6 `<ctype.h>`

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

15.7 `<devctl.h>`

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

15.8 <dirent.h>

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

15.9 `<errno.h>`

The following methods and variables in `<errno.h>` are supported:

- `errno`

15.10 `<fcntl.h>`

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `fcntl()`
- `open()`

15.11 `<fenv.h>`

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

15.12 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

15.13 `<locale.h>`

The following methods and variables in `<locale.h>` are supported:

- `localeconv()`
- `setlocale()`

15.14 <math.h>

The following methods and variables in <math.h> are supported:

- acos()
- acosf()
- acosh()
- acoshf()
- acoshl()
- acosl()
- asin()
- asinf()
- asinh()
- asinhf()
- asinhl()
- asinl()
- atan()
- atan2()
- atan2f()
- atan2l()
- atanf()
- atanh()
- atanhf()
- atanhl()
- atanl()
- cbrt()
- cbrtf()
- cbrtl()
- ceil()
- ceilf()
- ceill()
- copysign()
- copysignf()
- copysignl()
- cos()
- cosf()

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfcl()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`
- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`

- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`
- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

15.15 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_close()
- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_timedreceive()
- mq_timedsend()
- mq_unlink()

15.16 `<net/if.h>`

The following methods and variables in `<net/if.h>` are supported:

- `if_freenameindex()`
- `if_indextoname()`
- `if_nameindex()`
- `if_nametoindex()`

15.17 <netdb.h>

The following methods and variables in `<netdb.h>` are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

15.18 <pthread.h>

The following methods and variables in `<pthread.h>` are supported:

- `pthread_attr_destroy()`
- `pthread_attr_getdetachstate()`
- `pthread_attr_getguardsize()`
- `pthread_attr_getinheritsched()`
- `pthread_attr_getschedparam()`
- `pthread_attr_getschedpolicy()`
- `pthread_attr_getscope()`
- `pthread_attr_getstack()`
- `pthread_attr_getstacksize()`
- `pthread_attr_init()`
- `pthread_attr_setdetachstate()`
- `pthread_attr_setguardsize()`
- `pthread_attr_setinheritsched()`
- `pthread_attr_setschedparam()`
- `pthread_attr_setschedpolicy()`
- `pthread_attr_setscope()`
- `pthread_attr_setstack()`
- `pthread_attr_setstacksize()`
- `pthread_barrier_destroy()`
- `pthread_barrier_init()`
- `pthread_barrier_wait()`
- `pthread_barrierattr_destroy()`
- `pthread_barrierattr_init()`
- `pthread_cancel()`
- `pthread_cleanup_pop()`
- `pthread_cleanup_push()`
- `pthread_cond_broadcast()`
- `pthread_cond_destroy()`
- `pthread_cond_init()`
- `pthread_cond_signal()`
- `pthread_cond_timedwait()`
- `pthread_cond_wait()`

- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_getpshared()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_condattr_setpshared()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_getpshared()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_mutexattr_setpshared()
- pthread_mutexattr_settype()

- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

15.19 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

15.20 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

15.21 <setjmp.h>

The following methods and variables in `<setjmp.h>` are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

15.22 `<signal.h>`

The following methods and variables in `<signal.h>` are supported:

- `kill()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `signal()`
- `sigpending()`
- `sigprocmask()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

15.23 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawn_file_actions_addclose()
- posix_spawn_file_actions_adddup2()
- posix_spawn_file_actions_addopen()
- posix_spawn_file_actions_destroy()
- posix_spawn_file_actions_init()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getpgroup()
- posix_spawnattr_getschedparam()
- posix_spawnattr_getschedpolicy()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setpgroup()
- posix_spawnattr_setschedparam()
- posix_spawnattr_setschedpolicy()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()
- posix_spawnp()

15.24 `<stdarg.h>`

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

15.25 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fseeko()`
- `fsetpos()`
- `ftell()`
- `ftello()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `getc()`
- `getc_unlocked()`
- `getchar()`
- `getchar_unlocked()`
- `perror()`
- `printf()`

- putc()
- putc_unlocked()
- putchar()
- putchar_unlocked()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tmpfile()
- ungetc()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

15.26 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- _Exit()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbstowcs()
- mbtowc()
- qsort()
- rand()
- rand_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()

- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

15.27 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

15.28 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mlock()
- mlockall()
- mmap()
- mprotect()
- msync()
- munlock()
- munlockall()
- munmap()
- shm_open()
- shm_unlink()

15.29 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

The following methods and variables in `<sys/select.h>` are not supported:

- `pselect()`

15.30 <sys/socket.h>

The following methods and variables in `<sys/socket.h>` are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in `<sys/socket.h>` are not supported:

- `sockatmark()`

15.31 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `chmod()`
- `fchmod()`
- `fstat()`
- `lstat()`
- `mkdir()`
- `mkfifo()`
- `stat()`
- `umask()`

15.32 <sys/times.h>

The following methods and variables in <sys/times.h> are supported:

- times()

15.33 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

15.34 `<sys/wait.h>`

The following methods and variables in `<sys/wait.h>` are supported:

- `wait()`
- `waitpid()`

15.35 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `asctime_r()`
- `clock()`
- `clock_getres()`
- `clock_gettime()`
- `clock_nanosleep()`
- `clock_settime()`
- `ctime_r()`
- `difftime()`
- `gmtime()`
- `gmtime_r()`
- `localtime()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `strftime()`
- `time()`
- `timer_create()`
- `timer_delete()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`
- `tzname`
- `tzset()`

The following methods in `<time.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `clock_getcpuclockid()`

15.36 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- _exit()
- access()
- alarm()
- chdir()
- chown()
- close()
- dup()
- dup2()
- environ
- fchown()
- fdatasync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getlogin()
- getpgrp()
- getpid()
- getppid()
- getuid()
- link()
- lseek()
- pathconf()
- pause()
- pipe()
- read()
- rmdir()

- `setegid()`
- `seteuid()`
- `setgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `sysconf()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`

15.37 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcrtomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcscspn()

- wcsftime()
- wcslen()
- wcsncat()
- wcsncmp()
- wcsncpy()
- wcspbrk()
- wcsrchr()
- wcsrtombs()
- wcsspn()
- wcsstr()
- wcstod()
- wcstof()
- wcstok()
- wcstol()
- wcstold()
- wcstoll()
- wcstoul()
- wcstoull()
- wcsxfrm()
- wctob()
- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

15.38 `<wctype.h>`

The following methods and variables in `<wctype.h>` are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

CHAPTER

SIXTEEN

FACE 3.0 SECURITY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

16.1 Summary

The follow table summarizes alignment with the FACE 3.0 Security standard:

Supported	163
ENOSYS	1
Not supported	0

16.2 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

16.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

16.4 <devctl.h>

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

16.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

16.6 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

16.7 <netdb.h>

The following methods and variables in `<netdb.h>` are supported:

- `freeaddrinfo()`
- `getaddrinfo()`
- `getnameinfo()`

16.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_init()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_create()
- pthread_equal()
- pthread_getschedparam()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_once()
- pthread_self()
- pthread_setschedparam()
- pthread_setschedprio()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_getcpuclockid()

16.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_yield()

16.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_wait()

16.11 `<signal.h>`

The following methods and variables in `<signal.h>` are supported:

- `pthread_sigmask()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `sigpending()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

16.12 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abs()
- atof()
- atoi()
- atol()
- div()
- labs()
- ldiv()
- malloc()
- rand_r()
- strtod()
- strtol()
- strtoul()

16.13 `<string.h>`

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

16.14 <sys/mman.h>

The following methods and variables in `<sys/mman.h>` are supported:

- `mmap()`
- `shm_open()`

16.15 `<sys/socket.h>`

The following methods and variables in `<sys/socket.h>` are supported:

- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`

16.16 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `stat()`

16.17 <time.h>

The following methods and variables in <time.h> are supported:

- clock_getres()
- clock_gettime()
- clock_nanosleep()
- clock_settime()
- nanosleep()
- timer_create()
- timer_getoverrun()
- timer_gettime()
- timer_settime()

16.18 <unistd.h>

The following methods and variables in `<unistd.h>` are supported:

- `alarm()`
- `ftruncate()`
- `pause()`

CHAPTER

SEVENTEEN

FACE 3.0 SAFETY BASE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

17.1 Summary

The follow table summarizes alignment with the FACE 3.0 Safety Base standard:

Supported	246
ENOSYS	1
Not supported	0

17.2 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

17.3 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

17.4 <devctl.h>

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

17.5 <dirent.h>

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

17.6 <errno.h>

The following methods and variables in `<errno.h>` are supported:

- `errno`

17.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- open()

17.8 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

17.9 `<mqueue.h>`

The following methods and variables in `<mqueue.h>` are supported:

- `mq_getattr()`
- `mq_notify()`
- `mq_open()`
- `mq_receive()`
- `mq_send()`
- `mq_setattr()`
- `mq_timedreceive()`
- `mq_timedsend()`

17.10 `<netdb.h>`

The following methods and variables in `<netdb.h>` are supported:

- `freeaddrinfo()`
- `getaddrinfo()`
- `getnameinfo()`

17.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstacksize()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_create()
- pthread_equal()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_key_create()
- pthread_mutex_init()
- pthread_mutex_lock()

- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_once()
- pthread_self()
- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_getcpuclockid()

17.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_yield()

17.13 <semaphore.h>

The following methods and variables in `<semaphore.h>` are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

17.14 <signal.h>

The following methods and variables in `<signal.h>` are supported:

- `pthread_sigmask()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `sigpending()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

17.15 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgets()`
- `fileno()`
- `fopen()`
- `fprintf()`
- `fread()`
- `freopen()`
- `fseek()`
- `fseeko()`
- `ftell()`
- `ftello()`
- `fwrite()`
- `remove()`
- `rename()`
- `snprintf()`

17.16 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abs()
- atof()
- atoi()
- atol()
- calloc()
- div()
- labs()
- ldiv()
- malloc()
- rand_r()
- strtod()
- strtol()
- strtoul()

17.17 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

17.18 <sys/mman.h>

The following methods and variables in `<sys/mman.h>` are supported:

- `mmap()`
- `shm_open()`

17.19 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

17.20 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

17.21 `<sys/stat.h>`

The following methods and variables in `<sys/stat.h>` are supported:

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

17.22 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `asctime_r()`
- `clock_getres()`
- `clock_gettime()`
- `clock_nanosleep()`
- `clock_settime()`
- `ctime_r()`
- `difftime()`
- `gmtime_r()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `time()`
- `timer_create()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`
- `tzname`
- `tzset()`

17.23 <unistd.h>

The following methods and variables in `<unistd.h>` are supported:

- `access()`
- `alarm()`
- `chdir()`
- `close()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `gethostname()`
- `link()`
- `lseek()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`

CHAPTER

EIGHTEEN

FACE 3.0 SAFETY EXTENDED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

18.1 Summary

The follow table summarizes alignment with the FACE 3.0 Safety Extended standard:

Supported	316
ENOSYS	11
Not supported	9

18.2 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

18.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

18.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix_devctl()

18.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- closedir()
- opendir()
- readdir()
- readdir_r()
- rewinddir()

18.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

18.7 <fcntl.h>

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `fcntl()`
- `open()`

18.8 <math.h>

The following methods and variables in <math.h> are supported:

- acos()
- acosh()
- asin()
- asinh()
- atan()
- atan2()
- atanh()
- ceil()
- cos()
- cosh()
- exp()
- fabs()
- floor()
- fmod()
- frexp()
- isinf()
- isnan()
- ldexp()
- log()
- log10()
- modf()
- pow()
- sin()
- sinh()
- sqrt()
- tan()
- tanh()

18.9 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_timedreceive()
- mq_timedsend()

18.10 `<netdb.h>`

The following methods and variables in `<netdb.h>` are supported:

- `freeaddrinfo()`
- `getaddrinfo()`
- `getnameinfo()`

18.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstacksize()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_create()

- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_once()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()
- pthread_setconcurrency()
- pthread_setschedparam()
- pthread_setschedprio()
- pthread_setspecific()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- pthread_atfork()
- pthread_getcpuclockid()

18.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

18.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

18.14 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- siglongjmp()
- sigsetjmp()

18.15 `<signal.h>`

The following methods and variables in `<signal.h>` are supported:

- `kill()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `sigpending()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

18.16 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix_spawn()
- posix_spawnattr_destroy()
- posix_spawnattr_getflags()
- posix_spawnattr_getsigdefault()
- posix_spawnattr_getsigmask()
- posix_spawnattr_init()
- posix_spawnattr_setflags()
- posix_spawnattr_setsigdefault()
- posix_spawnattr_setsigmask()

18.17 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va_arg()
- va_end()
- va_start()

18.18 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- remove()
- rename()
- snprintf()
- sscanf()
- vfprintf()
- vsnprintf()

18.19 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- _Exit()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- malloc()
- rand_r()
- realloc()
- strtod()
- strtol()
- strtoul()

18.20 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

18.21 `<sys/mman.h>`

The following methods and variables in `<sys/mman.h>` are supported:

- `mmap()`
- `shm_open()`

18.22 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

18.23 <sys/socket.h>

The following methods and variables in `<sys/socket.h>` are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`

18.24 <sys/stat.h>

The following methods and variables in `<sys/stat.h>` are supported:

- `chmod()`
- `fstat()`
- `lstat()`
- `mkdir()`
- `mkfifo()`
- `stat()`
- `umask()`

18.25 <sys/times.h>

The following methods and variables in `<sys/times.h>` are supported:

- `times()`

18.26 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

18.27 `<sys/wait.h>`

The following methods and variables in `<sys/wait.h>` are supported:

- `waitpid()`

18.28 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `asctime_r()`
- `clock()`
- `clock_getres()`
- `clock_gettime()`
- `clock_nanosleep()`
- `clock_settime()`
- `ctime_r()`
- `difftime()`
- `gmtime_r()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `strftime()`
- `time()`
- `timer_create()`
- `timer_delete()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`
- `tzname`
- `tzset()`

18.29 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- _exit()
- access()
- alarm()
- chdir()
- chown()
- close()
- dup2()
- environ
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getpgrp()
- getpid()
- getppid()
- getuid()
- link()
- lseek()
- pause()
- pipe()
- read()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setuid()
- sleep()
- sysconf()

- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`

CHAPTER

NINETEEN

FACE 3.0 GENERAL PURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

19.1 Summary

The follow table summarizes alignment with the FACE 3.0 General Purpose standard:

Supported	752
ENOSYS	14
Not supported	46

19.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio_cancel()
- aio_error()
- aio_fsync()
- aio_read()
- aio_return()
- aio_write()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio_suspend()
- lio_listio()

19.3 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

19.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- assert()

19.5 `<complex.h>`

The following methods and variables in `<complex.h>` are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- cexpl()
- cimag()
- cimagf()
- cimagl()
- clog()
- clogf()
- clogl()
- conj()
- conjf()
- conjl()
- cpow()
- cpowf()
- cpowl()
- cproj()
- cprojf()
- cprojl()
- creal()
- crealf()
- creall()
- csin()
- csinf()
- csinh()
- csinhf()
- csinhl()
- csinl()
- csqrt()
- csqrtf()
- csqrtl()
- ctan()
- ctanf()
- ctanh()
- ctanhf()
- ctanhl()
- ctanl()

19.6 <ctype.h>

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

19.7 `<devctl.h>`

The following methods and variables in `<devctl.h>` are supported:

- `posix_devctl()`

19.8 <dirent.h>

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

19.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

19.10 <fcntl.h>

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `fcntl()`
- `open()`

19.11 <fenv.h>

The following methods and variables in `<fenv.h>` are not supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

19.12 <inttypes.h>

The following methods and variables in `<inttypes.h>` are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

19.13 `<locale.h>`

The following methods and variables in `<locale.h>` are supported:

- `localeconv()`
- `setlocale()`

19.14 <math.h>

The following methods and variables in <math.h> are supported:

- acos()
- acosf()
- acosh()
- acoshf()
- acoshl()
- acosl()
- asin()
- asinf()
- asinh()
- asinhf()
- asinhl()
- asinl()
- atan()
- atan2()
- atan2f()
- atan2l()
- atanf()
- atanh()
- atanhf()
- atanhl()
- atanl()
- cbrt()
- cbrtf()
- cbrtl()
- ceil()
- ceilf()
- ceill()
- copysign()
- copysignf()
- copysignl()
- cos()
- cosf()

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfcl()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- fmin()
- fminf()
- fminl()
- fmod()
- fmodf()
- fmodl()
- frexp()
- frexpf()
- frexpl()
- hypot()
- hypotf()
- hypotl()
- ilogb()
- ilogbf()
- ilogbl()
- isinf()
- isnan()
- ldexp()
- ldexpf()
- ldexpl()
- lgamma()
- lgammaf()
- lgammal()
- llrint()
- llrintf()
- llrintl()
- llround()
- llroundf()
- llroundl()
- log()
- log10()
- log10f()
- log10l()
- log1p()

- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`
- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

19.15 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_close()
- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_timedreceive()
- mq_timedsend()
- mq_unlink()

19.16 `<net/if.h>`

The following methods and variables in `<net/if.h>` are supported:

- `if_freenameindex()`
- `if_indextoname()`
- `if_nameindex()`
- `if_nametoindex()`

19.17 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- endhostent()
- endnetent()
- endprotoent()
- endservent()
- freeaddrinfo()
- gai_strerror()
- getaddrinfo()
- gethostent()
- getnameinfo()
- getnetbyaddr()
- getnetbyname()
- getnetent()
- getprotobyname()
- getprotobynumber()
- getprotoent()
- getservbyname()
- getservbyport()
- getservent()
- sethostent()
- setnetent()
- setprotoent()
- setservent()

19.18 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstacksize()
- pthread_barrier_destroy()
- pthread_barrier_init()
- pthread_barrier_wait()
- pthread_barrierattr_destroy()
- pthread_barrierattr_init()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()

- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_getpshared()
- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_condattr_setpshared()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getconcurrency()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_getpshared()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_mutexattr_setpshared()
- pthread_mutexattr_settype()

- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

19.19 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched_get_priority_max()
- sched_get_priority_min()
- sched_rr_get_interval()
- sched_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched_getparam()
- sched_getscheduler()
- sched_setparam()
- sched_setscheduler()

19.20 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

19.21 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

19.22 `<signal.h>`

The following methods and variables in `<signal.h>` are supported:

- `kill()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `signal()`
- `sigpending()`
- `sigprocmask()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

19.23 <spawn.h>

The following methods and variables in `<spawn.h>` are not supported:

- `posix_spawn()`
- `posix_spawn_file_actions_addclose()`
- `posix_spawn_file_actions_adddup2()`
- `posix_spawn_file_actions_addopen()`
- `posix_spawn_file_actions_destroy()`
- `posix_spawn_file_actions_init()`
- `posix_spawnattr_destroy()`
- `posix_spawnattr_getflags()`
- `posix_spawnattr_getpgroup()`
- `posix_spawnattr_getschedparam()`
- `posix_spawnattr_getschedpolicy()`
- `posix_spawnattr_getsigdefault()`
- `posix_spawnattr_getsigmask()`
- `posix_spawnattr_init()`
- `posix_spawnattr_setflags()`
- `posix_spawnattr_setpgroup()`
- `posix_spawnattr_setschedparam()`
- `posix_spawnattr_setschedpolicy()`
- `posix_spawnattr_setsigdefault()`
- `posix_spawnattr_setsigmask()`
- `posix_spawnp()`

19.24 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va_arg()
- va_copy()
- va_end()
- va_start()

19.25 <stdio.h>

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fseeko()`
- `fsetpos()`
- `ftell()`
- `ftello()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `getc()`
- `getc_unlocked()`
- `getchar()`
- `getchar_unlocked()`
- `perror()`
- `printf()`

- putc()
- putc_unlocked()
- putchar()
- putchar_unlocked()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()
- sscanf()
- stderr
- stdin
- stdout
- tmpfile()
- ungetc()
- vfprintf()
- vfscanf()
- vprintf()
- vscanf()
- vsnprintf()
- vsprintf()
- vsscanf()

19.26 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- _Exit()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbstowcs()
- mbtowc()
- qsort()
- rand()
- rand_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()

- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

19.27 <string.h>

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

19.28 <sys/mman.h>

The following methods and variables in `<sys/mman.h>` are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

19.29 <sys/select.h>

The following methods and variables in `<sys/select.h>` are supported:

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

The following methods and variables in `<sys/select.h>` are not supported:

- `pselect()`

19.30 `<sys/socket.h>`

The following methods and variables in `<sys/socket.h>` are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in `<sys/socket.h>` are not supported:

- `sockatmark()`

19.31 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

19.32 <sys/times.h>

The following methods and variables in <sys/times.h> are supported:

- times()

19.33 <sys/utsname.h>

The following methods and variables in `<sys/utsname.h>` are supported:

- `uname()`

19.34 <sys/wait.h>

The following methods and variables in `<sys/wait.h>` are supported:

- `wait()`
- `waitpid()`

19.35 <time.h>

The following methods and variables in `<time.h>` are supported:

- `asctime_r()`
- `clock()`
- `clock_getres()`
- `clock_gettime()`
- `clock_nanosleep()`
- `clock_settime()`
- `ctime_r()`
- `difftime()`
- `gmtime()`
- `gmtime_r()`
- `localtime()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `strftime()`
- `time()`
- `timer_create()`
- `timer_delete()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`
- `tzname`
- `tzset()`

The following methods in `<time.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `clock_getcpuclockid()`

19.36 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- _exit()
- access()
- alarm()
- chdir()
- chown()
- close()
- dup()
- dup2()
- environ
- fchown()
- fdatasync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getlogin()
- getpgrp()
- getpid()
- getppid()
- getuid()
- link()
- lseek()
- pathconf()
- pause()
- pipe()
- read()
- rmdir()

- setegid()
- seteuid()
- setgid()
- setsid()
- setuid()
- sleep()
- sysconf()
- unlink()
- write()

The following methods in <unistd.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- execl()
- execle()
- execv()
- execve()
- fork()

The following methods and variables in <unistd.h> are not supported:

- confstr()

19.37 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcrtomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcscspn()

- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

19.38 `<wctype.h>`

The following methods and variables in `<wctype.h>` are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

CHAPTER

TWENTY

SCA 2.2.2 AEP

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

20.1 Summary

The follow table summarizes alignment with the SCA 2.2.2 AEP standard:

Supported	244
ENOSYS	0
Not supported	0

20.2 `<ctype.h>`

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

20.3 <dirent.h>

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

20.4 `<fcntl.h>`

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `open()`

20.5 `<locale.h>`

The following methods and variables in `<locale.h>` are supported:

- `setlocale()`

20.6 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `asin()`
- `atan()`
- `atan2()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

20.7 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getguardsize()
- pthread_attr_getinheritsched()
- pthread_attr_getschedparam()
- pthread_attr_getschedpolicy()
- pthread_attr_getscope()
- pthread_attr_getstack()
- pthread_attr_getstackaddr()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setdetachstate()
- pthread_attr_setguardsize()
- pthread_attr_setinheritsched()
- pthread_attr_setschedparam()
- pthread_attr_setschedpolicy()
- pthread_attr_setscope()
- pthread_attr_setstack()
- pthread_attr_setstackaddr()
- pthread_attr_setstacksize()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_getclock()
- pthread_condattr_getpshared()

- pthread_condattr_init()
- pthread_condattr_setclock()
- pthread_condattr_setpshared()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_getprioceiling()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_setprioceiling()
- pthread_mutex_timedlock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_destroy()
- pthread_mutexattr_getprioceiling()
- pthread_mutexattr_getprotocol()
- pthread_mutexattr_getpshared()
- pthread_mutexattr_gettype()
- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_setprotocol()
- pthread_mutexattr_setpshared()
- pthread_mutexattr_settype()
- pthread_once()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()

- `pthread_setschedparam()`
- `pthread_setspecific()`
- `pthread_testcancel()`

20.8 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_trywait()
- sem_unlink()
- sem_wait()

20.9 <setjmp.h>

The following methods and variables in `<setjmp.h>` are supported:

- `longjmp()`
- `setjmp()`

20.10 <signal.h>

The following methods and variables in `<signal.h>` are supported:

- `kill()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `signal()`
- `sigpending()`
- `sigprocmask()`
- `sigsuspend()`
- `sigwait()`

20.11 `<stdio.h>`

The following methods and variables in `<stdio.h>` are supported:

- `clearerr()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgets()`
- `fileno()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fseeko()`
- `ftell()`
- `ftello()`
- `fwrite()`
- `getc()`
- `getchar()`
- `gets()`
- `perror()`
- `printf()`
- `putc()`
- `putchar()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`

- scanf()
- setbuf()
- setvbuf()
- sprintf()
- sscanf()
- tmpfile()
- tmpnam()
- ungetc()

20.12 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- free()
- malloc()
- qsort()
- rand()
- rand_r()
- realloc()
- srand()

20.13 `<string.h>`

The following methods and variables in `<string.h>` are supported:

- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`

20.14 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- fstat()
- mkdir()
- stat()

20.15 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `asctime()`
- `asctime_r()`
- `clock_getres()`
- `clock_gettime()`
- `clock_settime()`
- `ctime()`
- `ctime_r()`
- `gmtime()`
- `gmtime_r()`
- `localtime()`
- `localtime_r()`
- `mktime()`
- `nanosleep()`
- `strftime()`
- `time()`
- `timer_create()`
- `timer_delete()`
- `timer_getoverrun()`
- `timer_gettime()`
- `timer_settime()`

20.16 <unistd.h>

The following methods and variables in `<unistd.h>` are supported:

- `access()`
- `chdir()`
- `close()`
- `fpathconf()`
- `getcwd()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`

20.17 <utime.h>

The following methods and variables in <utime.h> are supported:

- utime()

CHAPTER

TWENTYONE

SCA 4.1 ULTRA LIGHTWEIGHT AEP

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

21.1 Summary

The follow table summarizes alignment with the SCA 4.1 Ultra Lightweight AEP standard:

Supported	22
ENOSYS	0
Not supported	0

21.2 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `exp()`
- `exp2()`

21.3 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_open()
- mq_receive()
- mq_send()

21.4 `<pthread.h>`

The following methods and variables in `<pthread.h>` are supported:

- `pthread_attr_getstacksize()`
- `pthread_attr_init()`
- `pthread_attr_setinheritsched()`
- `pthread_create()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_settype()`
- `pthread_self()`

21.5 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_init()
- sem_post()
- sem_wait()

21.6 <time.h>

The following methods and variables in <time.h> are supported:

- clock_getres()
- clock_gettime()
- timer_create()
- timer_settime()

CHAPTER

TWENTYTWO

SCA 4.1 LIGHTWEIGHT AEP

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

22.1 Summary

The follow table summarizes alignment with the SCA 4.1 Lightweight AEP standard:

Supported	111
ENOSYS	0
Not supported	0

22.2 `<ctype.h>`

The following methods and variables in `<ctype.h>` are supported:

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

22.3 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- open()

22.4 `<math.h>`

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

22.5 `<mqueue.h>`

The following methods and variables in `<mqueue.h>` are supported:

- `mq_open()`
- `mq_receive()`
- `mq_send()`

22.6 `<pthread.h>`

The following methods and variables in `<pthread.h>` are supported:

- `pthread_attr_destroy()`
- `pthread_attr_getschedparam()`
- `pthread_attr_getstacksize()`
- `pthread_attr_init()`
- `pthread_attr_setinheritsched()`
- `pthread_cond_broadcast()`
- `pthread_cond_destroy()`
- `pthread_cond_init()`
- `pthread_cond_signal()`
- `pthread_cond_wait()`
- `pthread_create()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_settype()`
- `pthread_self()`

22.7 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_getvalue()
- sem_init()
- sem_post()
- sem_wait()

22.8 `<stdio.h>`

The following methods and variables in `<stdio.h>` are supported:

- `sscanf()`

22.9 <stdlib.h>

The following methods and variables in `<stdlib.h>` are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `free()`
- `malloc()`
- `qsort()`
- `rand()`
- `realloc()`
- `srand()`

22.10 `<string.h>`

The following methods and variables in `<string.h>` are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strchr()`
- `strcmp()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`

22.11 `<time.h>`

The following methods and variables in `<time.h>` are supported:

- `clock_getres()`
- `clock_gettime()`
- `gmtime()`
- `localtime()`
- `mktime()`
- `strftime()`
- `time()`
- `timer_create()`
- `timer_gettime()`
- `timer_settime()`

22.12 `<unistd.h>`

The following methods and variables in `<unistd.h>` are supported:

- `close()`
- `read()`
- `write()`

CHAPTER

TWENTYTHREE

SCA 4.1 [FULL] AEP

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

23.1 Summary

The follow table summarizes alignment with the SCA 4.1 [Full] AEP standard:

Supported	256
ENOSYS	0
Not supported	0

23.2 <arpa/inet.h>

The following methods and variables in `<arpa/inet.h>` are supported:

- `htonl()`
- `htons()`
- `ntohl()`
- `ntohs()`

23.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

23.4 `<dirent.h>`

The following methods and variables in `<dirent.h>` are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

23.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

23.6 `<fcntl.h>`

The following methods and variables in `<fcntl.h>` are supported:

- `creat()`
- `open()`

23.7 <math.h>

The following methods and variables in `<math.h>` are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

23.8 <mqueue.h>

The following methods and variables in <mqueue.h> are supported:

- mq_close()
- mq_getattr()
- mq_notify()
- mq_open()
- mq_receive()
- mq_send()
- mq_setattr()
- mq_unlink()

23.9 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread_attr_destroy()
- pthread_attr_getdetachstate()
- pthread_attr_getschedparam()
- pthread_attr_getstacksize()
- pthread_attr_init()
- pthread_attr_setinheritsched()
- pthread_cancel()
- pthread_cleanup_pop()
- pthread_cleanup_push()
- pthread_cond_broadcast()
- pthread_cond_destroy()
- pthread_cond_init()
- pthread_cond_signal()
- pthread_cond_timedwait()
- pthread_cond_wait()
- pthread_condattr_destroy()
- pthread_condattr_init()
- pthread_create()
- pthread_detach()
- pthread_equal()
- pthread_exit()
- pthread_getschedparam()
- pthread_getspecific()
- pthread_join()
- pthread_key_create()
- pthread_key_delete()
- pthread_mutex_destroy()
- pthread_mutex_init()
- pthread_mutex_lock()
- pthread_mutex_trylock()
- pthread_mutex_unlock()
- pthread_mutexattr_getpshared()

- pthread_mutexattr_init()
- pthread_mutexattr_setprioceiling()
- pthread_mutexattr_settype()
- pthread_once()
- pthread_self()
- pthread_setcancelstate()
- pthread_setcanceltype()
- pthread_setschedparam()
- pthread_setspecific()
- pthread_testcancel()

23.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- sem_close()
- sem_destroy()
- sem_getvalue()
- sem_init()
- sem_open()
- sem_post()
- sem_timedwait()
- sem_trywait()
- sem_unlink()
- sem_wait()

23.11 <signal.h>

The following methods and variables in `<signal.h>` are supported:

- `kill()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `signal()`
- `sigpending()`
- `sigprocmask()`
- `sigsuspend()`
- `sigwait()`

23.12 <stdarg.h>

The following methods and variables in `<stdarg.h>` are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

23.13 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- getc()
- getchar()
- perror()
- printf()
- putc()
- putchar()
- remove()
- rename()
- rewind()
- setbuf()
- setvbuf()

- snprintf()
- sscanf()
- ungetc()
- vsnprintf()

23.14 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- free()
- labs()
- malloc()
- qsort()
- rand()
- rand_r()
- realloc()
- srand()
- strtod()
- strtol()
- strtoul()

23.15 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strchr()
- strcmp()
- strcoll()
- strcspn()
- strerror()
- strerror_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok_r()
- strxfrm()

23.16 `<sys/select.h>`

The following methods and variables in `<sys/select.h>` are supported:

- `select()`

23.17 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getsockopt()
- listen()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()

23.18 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- fstat()
- mkdir()
- stat()

23.19 <time.h>

The following methods and variables in <time.h> are supported:

- asctime_r()
- clock_getres()
- clock_gettime()
- clock_settime()
- ctime_r()
- gmtime()
- gmtime_r()
- localtime()
- localtime_r()
- mktime()
- nanosleep()
- strftime()
- time()
- timer_create()
- timer_delete()
- timer_getoverrun()
- timer_gettime()
- timer_settime()

23.20 `<unistd.h>`

The following methods and variables in `<unistd.h>` are supported:

- `access()`
- `chdir()`
- `close()`
- `fpathconf()`
- `getcwd()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`

CHAPTER

TWENTYFOUR

GLOSSARY

POSIX

Portable Operating System Interface is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems.

INDEX