SOFTWARE COMMUNICATIONS ARCHITECTURE SPECIFICATION

APPENDIX B

SCA APPLICATION ENVIRONMENT PROFILE

AMENDED / 22 October 2008

Version 2.2.2A <ICWG Approved>

Prepared by:

JTRS Standards
Joint Program Executive Office (JPEO) Joint Tactical Radio System (JTRS)
Space and Naval Warfare Systems Center San Diego
53560 Hull Street, San Diego CA 92152-5001

REVISION SUMMARY

Version	Revisions
1.0	Initial Release
1.1	no changes
2.0	no changes
2.1	no changes
2.2	no changes
2.2.1	Incorporate approved Change Proposals, SCA-CCM number 2
2.2.2	Incorporate Approved Change Proposal, SCA-CCM-264
2.2.2A <Final Draft>	Incorporate Changes Resulting from SCA AEP Amendment – Included 20 new operations within the profile, removed 10 operations and clarified the specification position regarding the incorporation of the Standard C Libraries.
2.2.2A <ICWG Approved>	Included memmove operation per JTRS Community discussion **ICWG Approved**

TABLE OF CONTENTS

Foreword

SCA 2.2.2 incorporated a number of modifications that were intended to reduce ambiguity within the specification, address implementation concerns of SCA developers and ensure that the specification maintained its commercial relevance.

Lessons learned after the SCA 2.2.2 publication necessitated a modification of the set of permissible RTOS capabilities for SCA compliant applications. The combination of a demonstrated need and commercial availability of several new OS functions resulted in the creation of this revised AEP. SCA Appendix B 2.2.2A contains new material and technical corrections to the content of JPEO JTRS Standards specification – SCA 2.2.2 Appendix B.

This AEP revision better aligns the specification with the emerging needs of waveform developers while balancing the challenges of SCA compliance faced by radio set providers. The design objectives of the amended AEP were focused on the following tenets:

1. Maximizing the commercial content of the OE – the newly incorporated operations are widely supported by commercially available products;

2. Preserving the minimalist nature of the waveform/OE interface – the revision does not expand the profile to full POSIX profile 53 compliance or a set of operations which would significantly impact the number of functions that the OE must provide;

3. Minimizing the amount of development required by JTR set developers –the OE is allowed to provide more required functionality, thus relieving the developer's implementation responsibilities.

This document also formalizes the SCA 2.2.2 position on the incorporation of the Standard C Libraries. The SCA 2.2.2 reference to the C Standard "C Standard: Programming languages – C, ISO/IEC 9899:1999 [C99]" contains a chapter that defines the Standard C Library and its incorporated header files. Earlier versions of Appendix B were focused on operations and silent on whether or not the standard C libraries or header files could be used. The document from which Appendix B was modeled, IEEE 1003.13 IEEE Standard for Information Technology — Standardized Application Environment Profile (AEP) — POSIX® Real-time and Embedded Application Support contains explicit language regarding the interpretation of symbols defined within header files. In summary, the specification supports the visibility of all symbols within a header file other than those governed by feature test macros associated with the POSIX® Real-time profiles.

The amended Appendix B accounts for the lack of clarity pertaining to the sanctioned usage of Standard C Library elements within the SCA documentation. The revised AEP identifies a subset of the Standard C Library header files that are required by the profile and explicitly permits the utilization of the symbols defined in those header files.

APPENDIX B SCA APPLICATION ENVIRONMENT PROFILE

B.1 SCOPE

This appendix defines the Application Environment Profile (AEP) for the SCA, based on Standardized Application Environment Profile - POSIX® Realtime Application Support (AEP), IEEE Std 1003.13-2003.

The application environment profile (AEP), is the SCA required profile referenced in sections 3.1.1, 3.2.1, and 3.3.1 of the main document. The SCA dictates that an Operating Environment provides the options and functions designated as mandatory within this AEP and constrains an application to only use those services.

B.2 CONVENTIONS

Within this appendix, the following abbreviations are used:

1. "MAN" indicates that the identified function or option is mandatory for the indicated profile
2. "NRQ" indicates that the identified function or option is not required for the indicated profile
3. "PRT" indicates that only a subset of the indicated option or unit of functionality is required. This designation will be followed by a note or cross-reference indicating which elements are required.

B.3 STANDARDS

The standards identified in Table B-1 are required in whole or in part by the SCA AEP application environment profile.

Table B-1: Required Standards

Standard	SCA AEP
C Standard (ISO/IEC 9899:1999)	PRT[1]
POSIX (ISO/IEC 9945:2003)	PRT[1]

B.4 CONSTRAINTS

The real-time profile defined in this appendix requires only specific Units of Functionality of the included standards. The absence of particular elements of these standards introduces constraints on the use of some of the features of particular functions. These constraints must be observed by an application that conforms to the profile when using each of the required functions.

® POSIX is a registered trademark of the Institute of Electrical and Electronics Engineers, Inc.

[1] Specific functions and options are identified in section B.4

An Ada AEP has not been explicitly defined. Any Ada application shall be restricted to using the equivalent Ada functionality, as defined in POSIX Ada language binding (ISO/IEC 14519:2001), designated as mandatory by the AEP or may use the C interface.

B.4.1 POSIX.1

The options, limits, and any other constraints on POSIX.1 shall be provided as described in Table B-2.

Table B-2: POSIX.1 Option Requirements

Option	AEP
{_POSIX_ASYNCHRONOUS_IO}	MAN
{_POSIX_CHOWN_RESTRICTED}	NRQ
{_POSIX_CLOCK_SELECTION}	NRQ
{_POSIX_FSYNC}	PRT[2]
{_POSIX_MAPPED_FILES}	NRQ
{_POSIX_MEMLOCK_RANGE}	MAN
{_POSIX_MEMLOCK}	MAN
{_POSIX_MEMORY_PROTECTION}	NRQ
{_POSIX_MESSAGE_PASSING}	MAN
{_POSIX_MONOTONIC_CLOCK}	NRQ
{_POSIX_NO_TRUNC}	PRI
{_POSIX_PRIORITIZED_IO}	NRQ
{_POSIX_PRIORITY_SCHEDULING}	NRQ
{_POSIX_REALTIME_SIGNALS}	MAN
{_POSIX_SAVED_IDS}	NRQ
{_POSIX_SEMAPHORES}	MAN
{_POSIX_SHARED_MEMORY_OBJECTS}	NRQ
{_POSIX_SYNCHRONIZED_IO}	PRT[3]
{_POSIX_THREAD_ATTR_STACKADDR}	MAN
{_POSIX_THREAD_ATTR_STACKSIZE}	MAN
{_POSIX_THREAD_CPUTIME}	NRQ
{_POSIX_THREAD_PRIO_INHERIT}	MAN
{_POSIX_THREAD_PRIO_PROTECT}	MAN
{_POSIX_THREAD_PRIORITY_SCHEDULING}	MAN
{_POSIX_THREAD_PROCESS_SHARED}	NRQ

[2] fsync not required
[3] fdatasync not required

Option	AEP
{_POSIX_THREAD_SAFE_FUNCTIONS}	PRT[4]
{_POSIX_THREAD_SPORADIC_SERVER}	NRQ
{_POSIX_TIMEOUTS}	NRQ
{_POSIX_TIMERS}	MAN
{_POSIX_TRACE_EVENT_FILTER}	NRQ
{_POSIX_TRACE_LOG}	NRQ
{_POSIX_TRACE}	NRQ
{_POSIX_VDISABLE}	NRQ

NOTES:

- PRI - The primary file system shall generate an error for pathname components longer than NAME_MAX. The user is responsible for semantics of other file systems that may be mounted.
- Embedded processor C/C++ run-time libraries typically do not support stdio.h or iostream.h.
- Heavy weight processes are typically not supported in embedded operating systems. The mandatory POSIX.1b options can be implemented without the use of heavy weight signaling.

B.4.1.1 Single Process Function Behavior

The functions in Table B-3 shall behave as described in the applicable clauses of the referenced POSIX referenced POSIX specifications contained in Table B-1.

Table B-3: POSIX_SINGLE_PROCESS Functions

Function	AEP
confstr()	NRQ
environ	NRQ
errno	NRQ
getenv ( )	NRQ
setenv()	NRQ
sysconf ( )	NRQ
uname( )	NRQ
unsetenv()	NRQ

B.4.1.2 Multi Process Function Behavior

The functions listed in Table B-4 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

[4] See Table B-23: POSIX_THREAD_SAFE_FUNCTIONS Functions

Table B-4: POSIX_MULTI_PROCESS Functions

Function	AEP
_exit ( )	NRQ
_Exit()	NRQ
assert ( )	NRQ
atexit()	NRQ
clock()	NRQ
execl ( )	NRQ
execle ( )	NRQ
execlp ( )	NRQ
execv ( )	NRQ
execve ( )	NRQ
execvp ( )	NRQ
exit ( )	NRQ
fork( )	NRQ
getpgrp()	NRQ
getpid ( )	NRQ
getppid ( )	NRQ
setsid()	NRQ
sleep ( )	NRQ
times ( )	NRQ
wait( )	NRQ
waitpid ( )	NRQ

B.4.1.3 Job Control Function Behavior

The functions listed in Table B-5 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-5: POSIX_JOB_CONTROL Functions

Function	AEP
setpgid()	NRQ
tcgetpgrp()	NRQ
tcsetpgrp()	NRQ

B.4.1.4 Signals Function Behavior

Operating systems on embedded processors typically support neither signaling nor exception handling. POSIX does not define behaviors associated with divide by zero or overflow /

underflow. Signaling methods introduced as part of POSIX.1c are more consistent with the multi-threaded, single process model of a resource constrained processing environment.

The functions listed in Table B-6 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1, except for the following constraints:

1. An application that conforms to the AEP shall not result in abnormal termination of the process because this profile does not support multiple processes.
2. An application that conforms to the AEP shall not call the kill() function with a negative argument because this profile does not require process group functionality.

Table B-6: POSIX_SIGNALS Functions

Function	AEP
abort()	MAN
alarm()*	NRQ
kill()	MAN
pause()	MAN
raise()	MAN
sigaction()	MAN
sigaddset()	MAN
sigdelset()	MAN
sigemptyset()	MAN
sigfillset()	MAN
sigismember()	MAN
signal()	MAN
sigpending()	MAN
sigprocmask()	MAN
sigsupend()	MAN
sigwait()	MAN

NOTES:

- * Functionality provided through the POSIX timers
- abort() is used to support assert() which is widely supported.

B.4.1.5 Signal Jump Function Behavior

The functions listed in Table B-7 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-7: POSIX_SIGNAL_JUMP Functions

Function	AEP
siglongjmp()	NRQ
sigsetjmp()	NRQ

B.4.1.6 User Group Function Behavior

The functions listed in Table B-8 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-8: POSIX_USER_GROUPS Functions

Function	AEP
getegid()	NRQ
geteuid()	NRQ
getgid()	NRQ
getgroups()	NRQ
getlogin()	NRQ
getlogin_r()	NRQ
getuid()	NRQ
setegid()	NRQ
seteuid()	NRQ
setgid()	NRQ
setuid()	NRQ

B.4.1.7 File System Function Behavior

The functions listed in Table B-9 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-9: POSIX_FILE_SYSTEM Functions

Function	AEP
access()	MAN
chdir()	MAN
closedir()	MAN
creat()	MAN
fpathconf()	MAN
fstat()	MAN
getcwd()	MAN

Function	AEP
link()	MAN
mkdir()	MAN
opendir()	MAN
pathconf()	MAN
readdir()	MAN
readdir_r()	MAN
remove()	MAN
rename()	MAN
rewinddir()	MAN
rmdir()	MAN
stat()	MAN
tmpfile()	NRQ
tmpnam()	NRQ
unlink()	MAN
utime()	MAN

NOTE:

- POSIX file system not generally supported in embedded operating systems.

B.4.1.8 File Attributes Function Behavior

The functions listed in Table B-10 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1, except for the following constraint:

1. An application that conforms to the AEP shall be guaranteed that the file mode creation mask for any object created by any process is S-IRWXU; that is, the object shall be fully accessible to the creator.

Table B-10: POSIX_FILE_ATTRIBUTES Functions

Function	AEP
chmod()	NRQ
chown()	NRQ
fchmod()	NRQ
fchown()	NRQ
umask()	NRQ

NOTE:

- POSIX file system not generally supported in embedded operating systems

B.4.1.9 File and Directory Management Function Behavior

The functions listed in Table B-11 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-11: POSIX_FD_MGMT Functions

Function	AEP
dup()	NRQ
dup2()	NRQ
fcntl()	NRQ
fgetpos()	NRQ
fseek()	MAN
fseeko()	MAN
fsetpos()	NRQ
ftell()	MAN
ftello()	MAN
ftruncate()	NRQ
lseek()	MAN
rewind()	MAN

NOTE:

- POSIX file system not generally supported in embedded operating systems.

B.4.1.10 Device I/O Function Behavior

The functions listed in Table B-12 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-12: POSIX_DEVICE_IO Functions

Function	AEP
clearerr()	MAN
close()	MAN
fclose()	MAN
fdopen()	MAN
feof()	MAN
ferror()	MAN
fflush()	MAN
fgetc()	MAN

Function	AEP
fgets()	MAN
fileno()	MAN
fopen()	MAN
fprintf()	MAN
fputc()	MAN
fputs()	MAN
fread()	MAN
freopen()	MAN

Function	AEP
fscanf()	MAN
fwrite()	MAN
getc()	MAN
getchar()	MAN
gets()	NRQ
open()	MAN
perror()	MAN
printf()	MAN
putc()	MAN
putchar()	MAN
puts()	NRQ

Function	AEP
read()	MAN
scanf()	NRQ
setbuf()	MAN
setvbuf()	MAN
ungetc()	MAN
vfprintf()	NRQ
vfscanf()	NRQ
vprintf()	NRQ
vscanf()	NRQ
write()	MAN

NOTE:

- POSIX streams not generally supported in embedded operating systems.

B.4.1.11 Device-Specific Function Behavior

The functions listed in Table B-13 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-13: POSIX_DEVICE_SPECIFIC Functions

Function	AEP
cfgetispeed()	NRQ
cfgetospeed()	NRQ
cfsetispeed()	NRQ
cfsetospeed()	NRQ
ctermid()	NRQ
isatty()	NRQ
tcdrain()	NRQ
tcflow()	NRQ
tcflush()	NRQ
tcgetattr()	NRQ
tcsendbreak()	NRQ
tcsetattr()	NRQ
ttyname()	NRQ
ttyname_r()	NRQ

B.4.1.12 System Database Function Behavior

The functions listed in Table B-14 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-14: POSIX_SYSTEM_DATABASE Functions

Function	AEP
getgrgid()	NRQ
getgrgid_r()	NRQ
getgrnam()	NRQ
getgrnam_r()	NRQ
getpwnam()	NRQ
getpwnam_r()	NRQ
getpwuid()	NRQ
getpwuid_r()	NRQ

B.4.1.13 Pipe Function Behavior

The function listed in Table B-15 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-15: POSIX_PIPE_Function

Function	AEP
pipe()	NRQ

B.4.1.14 FIFO Function Behavior

The function listed in Table B-16 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-16: POSIX_FIFO Function

Function	AEP
mkfifo()	NRQ

B.4.1.15 C Language-Specific Support Services Function Behavior

The functions listed in Table B-17 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-17: POSIX_C_LANG_SUPPORT Functions

Function	AEP
fesetenv()	NRQ
abs()	MAN
asctime()	MAN
asctime_r()	MAN
atof()	MAN
atoi()	MAN
atol()	MAN
atoll()	NRQ
bsearch()	MAN
calloc()	MAN
ctime()	MAN
ctime_r()	MAN
difftime()	NRQ
div()	NRQ
feclearexcept()	NRQ
fegetenv()	NRQ
fegetexceptflag()	NRQ
fegetround()	NRQ
feholdexcept()	NRQ
feraiseexcept()	NRQ
fesetexceptflag()	NRQ
fesetround()	NRQ
fetestexcept()	NRQ
feupdateenv()	NRQ
free()	MAN
gmtime()	MAN
gmtime_r()	MAN
imaxabs()	NRQ
imaxdiv()	NRQ
isalnum()	MAN
isalpha()	MAN
isblank()	MAN
iscntrl()	MAN

Function	AEP
isdigit()	MAN
isgraph()	MAN
islower()	MAN
isprint()	MAN
ispunct()	MAN
isspace()	MAN
isupper()	MAN
isxdigit()	MAN
labs()	MAN
ldiv()	NRQ
llabs()	NRQ
lldiv()	NRQ
localeconv()	NRQ
localtime()	MAN
localtime_r()	MAN
malloc()	MAN
memchr()	MAN
memcmp()	MAN
memcpy()	MAN
memmove()	MAN
memset()	MAN
mktime()	MAN
qsort()	MAN
rand()	MAN
rand_r()	MAN
realloc()	MAN
setlocale ( )	MAN
snprintf()	MAN
sprintf()	NRQ
srand()	MAN
sscanf()	MAN
strcat()	NRQ
strchr()	MAN

Function	AEP
strcmp()	MAN
strcoll()	MAN
strcpy()	NRQ
strcspn()	MAN
strerror()	MAN
strerror_r()	MAN
strftime()	MAN
strlen()	MAN
strncat()	MAN
strncmp()	MAN
strncpy()	MAN
strpbrk()	MAN
strrchr()	MAN
strspn()	MAN
strstr()	MAN
strtod()	MAN
strtof()	NRQ
strtoimax()	NRQ
strtok()	MAN
strtok_r()	MAN

Function	AEP
strtol()	MAN
strtold()	NRQ
strtoll()	NRQ
strtoul()	MAN
strtoull()	NRQ
strtoumax()	NRQ
strxfrm()	MAN
time()	MAN
tolower()	MAN
toupper()	MAN
tzname,	NRQ
tzset()	NRQ
va_arg()	MAN
va_copy()	NRQ
va_end()	MAN
va_start()	MAN
vsnprintf()	MAN
vsprintf()	NRQ
vsscanf()	NRQ

NOTE:

- Support for dynamic memory allocation is essential to re-entrant object-oriented design.

B.4.1.16 C Language-Specific Mathematical Function Behavior

The functions listed in Table B-18 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-18: POSIX_C_LANG_MATH Functions

Function	AEP
acos()	MAN
acosf()	NRQ
acosh()	NRQ
acoshf()	NRQ
acoshl()	NRQ
acosl()	NRQ

Function	AEP
asin()	MAN
asinf()	NRQ
asinh()	NRQ
asinhf()	NRQ
asinhl()	NRQ
asinl()	NRQ

Function	AEP
atan()	MAN
atan2()	MAN
atan2f()	NRQ
atan2l()	NRQ
atanf()	NRQ
atanh()	NRQ
atanhf()	NRQ
atanhl()	NRQ
atanl()	NRQ
cabs()	NRQ
cabsf()	NRQ
cabsl()	NRQ
cacos()	NRQ
cacosf()	NRQ
cacosh()	NRQ
cacoshf()	NRQ
cacoshl()	NRQ
cacosl()	NRQ
carg()	NRQ
cargf()	NRQ
cargl()	NRQ
casin()	NRQ
casinf()	NRQ
casinh()	NRQ
casinhf()	NRQ
casinhl()	NRQ
casinl()	NRQ
catan()	NRQ
catanf()	NRQ
catanh()	NRQ
catanhf()	NRQ
catanhl()	NRQ
catanl()	NRQ
cbrt()	NRQ
cbrtf()	NRQ

Function	AEP
cbrtl()	NRQ
ccos()	NRQ
ccosf()	NRQ
Ccosh()	NRQ
ccoshf()	NRQ
ccoshl()	NRQ
ccosl()	NRQ
ceil()	MAN
ceilf()	NRQ
ceill()	NRQ
cexp()	NRQ
cexpf()	NRQ
cexpl()	NRQ
Cimag()	NRQ
cimagf()	NRQ
cimagl()	NRQ
clog()	NRQ
clogf()	NRQ
clogl()	NRQ
conj()	NRQ
conjf()	NRQ
conjl()	NRQ
copysign()	NRQ
copysignf()	NRQ
copysignl()	NRQ
cos()	MAN
cosf()	NRQ
cosh()	MAN
coshf()	NRQ
coshl()	NRQ
cosl()	NRQ
cpow()	NRQ
cpowf()	NRQ
cpowl()	NRQ
cproj()	NRQ

Function	AEP
cprojf()	NRQ
cprojl()	NRQ
creal()	NRQ
crealf()	NRQ
creall()	NRQ
csin()	NRQ
csinf()	NRQ
csinh()	NRQ
csinhf()	NRQ
csinhl()	NRQ
csinl()	NRQ
csqrt()	NRQ
csqrtf()	NRQ
csqrtl()	NRQ
ctan()	NRQ
ctanf()	NRQ
ctanh()	NRQ
ctanhf()	NRQ
ctanhl()	NRQ
ctanl()	NRQ
erf()	NRQ
erfc()	NRQ
erfcf()	NRQ
erfcl()	NRQ
erff()	NRQ
erfl()	NRQ
exp()	MAN
exp2()	NRQ
exp2f()	NRQ
exp2l()	NRQ
expf()	NRQ
expl()	NRQ
expm1()	NRQ
expm1f()	NRQ
expm1l()	NRQ

Function	AEP
fabs()	MAN
fabsf()	NRQ
fabsl()	NRQ
fdim()	NRQ
fdimf()	NRQ
fdiml()	NRQ
floor()	MAN
floorf()	NRQ
floorl()	NRQ
fma()	NRQ
fmaf()	NRQ
fmal()	NRQ
fmax()	NRQ
fmaxf()	NRQ
fmaxl()	NRQ
fmin()	NRQ
fminf()	NRQ
fminl()	NRQ
fmod()	MAN
fmodf()	NRQ
fmodl()	NRQ
fpclassify()	NRQ
frexp()	MAN
frexpf()	NRQ
frexpl()	NRQ
hypot()	NRQ
hypot()	NRQ
hypotl()	NRQ
ilogb()	NRQ
ilogbf()	NRQ
ilogbl()	NRQ
isfinite()	NRQ
isgreater()	NRQ
isgreaterequal()	NRQ
isinf()	NRQ

Function	AEP
isless()	NRQ
islessequal()	NRQ
islessgreater()	NRQ
isnan()	NRQ
isnormal()	NRQ
isunordered()	NRQ
ldexp()	MAN
ldexpf()	NRQ
ldexpl()	NRQ
lgamma()	NRQ
lgammaf()	NRQ
lgammal()	NRQ
llrint()	NRQ
llrintf()	NRQ
llrintl()	NRQ
llround()	NRQ
llroundf()	NRQ
llroundl()	NRQ
log()	MAN
log10()	MAN
log10f()	NRQ
log10l()	NRQ
log1p()	NRQ
log1pf()	NRQ
log1pl()	NRQ
log2()	NRQ
log2f()	NRQ
log2l()	NRQ
logb()	NRQ
logbf()	NRQ
logbl()	NRQ
logf()	NRQ
logl()	NRQ
lrint()	NRQ
lrintf()	NRQ

Function	AEP
lrintl()	NRQ
lround()	NRQ
lroundf()	NRQ
lroundl()	NRQ
modf()	MAN
modff()	NRQ
modfl()	NRQ
nan()	NRQ
nanf()	NRQ
nanl()	NRQ
nearbyint()	NRQ
nearbyintf()	NRQ
nearbyintl()	NRQ
nextafter()	NRQ
nextafterf()	NRQ
nextafterl()	NRQ
nexttoward()	NRQ
nexttowardf()	NRQ
nexttowardl()	NRQ
pow()	MAN
powf()	NRQ
powl()	NRQ
remainder()	NRQ
remainderf()	NRQ
remainderl()	NRQ
remquo()	NRQ
remquof()	NRQ
remquol()	NRQ
rint()	NRQ
rintf()	NRQ
rintl()	NRQ
round()	NRQ
roundf()	NRQ
roundl()	NRQ
scalbln()	NRQ

Function	AEP
scalblnf()	NRQ
scalblnl()	NRQ
scalbn()	NRQ
scalbnf()	NRQ
scalbnl()	NRQ
signbit()	NRQ
sin()	MAN
sinf()	NRQ
sinh()	MAN
sinhf()	NRQ
sinhl()	NRQ
sinl()	NRQ
sqrt()	MAN
sqrtf()	NRQ

Function	AEP
sqrtl()	NRQ
tan()	MAN
tanf()	NRQ
tanh()	MAN
tanhf()	NRQ
tanhl()	NRQ
tanl()	NRQ
tgamma()	NRQ
tgammaf()	NRQ
tgammal()	NRQ
trunc()	NRQ
truncf()	NRQ
truncl()	NRQ

B.4.1.17 C Language-Specific Non-local Jump Function Behavior.

The functions listed in Table B-19 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-19: POSIX_C_LANG_JUMP Functions

Function	AEP
longjmp()	NRQ
setjmp()	NRQ

NOTE:

- This is a form of context switch used to support a non-local exit.

B.4.1.18 POSIX Semaphore Function Behavior

The functions listed in Table B-20 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-20. POSIX_SEMAPHORES Functions

Function	AEP
sem_close()	MAN
sem_destroy()	MAN
sem_getvalue()	MAN
sem_init()	MAN

Function	AEP
sem_open()	MAN
sem_post()	MAN
sem_trywait()	MAN
sem_unlink()	MAN
sem_wait()	MAN

B.4.1.19 POSIX Timer Function Behavior

The functions listed in Table B-21 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-21. POSIX_TIMERS Functions

Function	AEP
clock_getres()	MAN
clock_gettime()	MAN
clock_settime()	MAN
nanosleep()	MAN
timer_create()	MAN
timer_delete()	MAN
timer_getoverrun()	MAN
timer_gettime()	MAN
timer_settime()	MAN

B.4.1.20 POSIX Threading Function Behavior

The functions listed in Table B-22 shall behave as described in the applicable clauses of the referenced POSIX specifications contained in Table B-1.

Table B-22. POSIX_THREADS_BASE Functions

Function	AEP
pthread_atfork()	NRQ
pthread_attr_xxx()	MAN
pthread_cancel()	MAN
pthread_cleanup_xxx()	MAN
pthread_cond_xxx()	MAN
pthread_condattr_xxx()	MAN
pthread_create()	MAN
pthread_detach()	MAN

Function	AEP
pthread_equal()	MAN
pthread_exit()	MAN
pthread_getschedparam()	MAN
pthread_getspecific()	MAN
pthread_join()	MAN
pthread_key_xxx()	MAN
pthread_kill()	MAN
pthread_mutex_xxx()	MAN
pthread_mutexattr_xxx()	MAN
pthread_once()	MAN
pthread_self()	MAN
pthread_setcancelstate()	MAN
pthread_setcanceltype()	MAN
pthread_setschedparam()	MAN
pthread_setspecific()	MAN
pthread_sigmask()	MAN
pthread_testcancel()	MAN

B.4.1.21 POSIX Thread Safe Option Requirements Behavior

The function listed in Table B-23shall behave as described in the referenced clause.

Table B-23: POSIX_THREAD_SAFE_FUNCTIONS Functions

Function	AEP
asctime_r()	MAN
ctime_r()	MAN
flockfile()	NRQ
ftrylockfile()	NRQ
funlockfile()	NRQ
getc_unlocked()	NRQ
getchar_unlocked()	NRQ
getgrgid_r()	NRQ
getgrnam_r()	NRQ
getlogin_r()	NRQ
getpwnam_r()	NRQ

Function	AEP
getpwuid_r()	NRQ
gmtime_r()	MAN
localtime_r()	MAN
putc_unlocked()	NRQ
putchar_unlocked()	NRQ
rand_r()	MAN
readdir_r()	MAN
strerror_r()	MAN
strtok_r()	MAN
ttyname_r()	NRQ

B.4.1.22 XSI Thread Mutex Ext Option Requirements Behavior

The function listed in Table B-24 shall behave as described in the referenced clause.

Table B-24 XSI_THREAD_MUTEX_EXT Functions

Function	AEP
pthread_mutexattr_gettype()	MAN
pthread_mutexattr_settype()	MAN

B.5 POSIX STANDARD C LIBRARY HEADER FILES

The Standard C Library header files listed in Table B-25 shall be included within the AEP as described in the referenced clause. All symbols (other than operations) included within the header files with a MAN or PRT designation are considered elements of the profile.

Table B-25 POSIX Standard C Library Header Files

Header File	AEP
assert.h	NRQ
complex.h	NRQ
ctype.h	MAN
errno.h	MAN
fenv.h	NRQ
float.h	NRQ
inttypes.h	NRQ
iso646.h	NRQ
limits.h	MAN
locale.h	PRT
math.h	PRT
setjmp.h	NRQ

Header File	AEP
signal.h	MAN
stdarg.h	PRT
stdbool.h	NRQ
stddef.h	NRQ
stdint.h	NRQ
stdio.h	PRT
stdlib.h	PRT
string.h	PRT
tgmath.h	NRQ
time.h	PRT
wchar.h	NRQ
wctype.h	NRQ

The remainder of this section is non-normative and provides a detailed view of the Standard C Library Functions that are included within the AEP relative to the header file in which they are defined.

B.5.1 Diagnostics **<assert.h>**

Table B-26: <assert.h> Functions

Function	AEP
`void assert(`*scalar* `expression);`	NRQ

B.5.2 Complex **<complex.h>**

Table B-27: <complex.h> Functions

Function	AEP
`double complex cacos(double complex z);`	NRQ
`float complex cacosf(float complex z);`	NRQ
`long double complex cacosl(long double complex z);`	NRQ
`double complex casin(double complex z);`	NRQ
`float complex casinf(float complex z);`	NRQ
`long double complex casinl(long double complex z);`	NRQ
`double complex catan(double complex z);`	NRQ
`float complex catanf(float complex z);`	NRQ
`long double complex catanl(long double complex z);`	NRQ
`double complex ccos(double complex z);`	NRQ
`float complex ccosf(float complex z);`	NRQ
`long double complex ccosl(long double complex z);`	NRQ
`double complex csin(double complex z);`	NRQ
`float complex csinf(float complex z);`	NRQ
`long double complex csinl(long double complex z);`	NRQ
`double complex ctan(double complex z);`	NRQ
`float complex ctanf(float complex z);`	NRQ
`long double complex ctanl(long double complex z);`	NRQ
`double complex cacosh(double complex z);`	NRQ
`float complex cacoshf(float complex z);`	NRQ
`long double complex cacoshl(long double complex z);`	NRQ
`double complex casinh(double complex z);`	NRQ
`float complex casinhf(float complex z);`	NRQ
`long double complex casinhl(long double complex z);`	NRQ
`double complex catanh(double complex z);`	NRQ

Function	AEP
`float complex catanhf(float complex z);`	NRQ
`long double complex catanhl(long double complex z);`	NRQ
`double complex ccosh(double complex z);`	NRQ
`float complex ccoshf(float complex z);`	NRQ
`long double complex ccoshl(long double complex z);`	NRQ
`double complex csinh(double complex z);`	NRQ
`float complex csinhf(float complex z);`	NRQ
`long double complex csinhl(long double complex z);`	NRQ
`double complex ctanh(double complex z);`	NRQ
`float complex ctanhf(float complex z);`	NRQ
`long double complex ctanhl(long double complex z);`	NRQ
`double complex cexp(double complex z);`	NRQ
`float complex cexpf(float complex z);`	NRQ
`long double complex cexpl(long double complex z);`	NRQ
`double complex clog(double complex z);`	NRQ
`float complex clogf(float complex z);`	NRQ
`long double complex clogl(long double complex z);`	NRQ
`double cabs(double complex z);`	NRQ
`float cabsf(float complex z);`	NRQ
`long double cabsl(long double complex z);`	NRQ
`double complex cpow(double complex x, double complex y);`	NRQ
`float complex cpowf(float complex x, float complex y);`	NRQ
`long double complex cpowl(long double complex x, long double complex y);`	NRQ
`double complex csqrt(double complex z);`	NRQ
`float complex csqrtf(float complex z);`	NRQ
`long double complex csqrtl(long double complex z);`	NRQ
`double carg(double complex z);`	NRQ
`float cargf(float complex z);`	NRQ
`long double cargl(long double complex z);`	NRQ
`double cimag(double complex z);`	NRQ
`float cimagf(float complex z);`	NRQ
`long double cimagl(long double complex z);`	NRQ
`double complex conj(double complex z);`	NRQ

Function	AEP
`float complex conjf(float complex z);`	NRQ
`long double complex conjl(long double complex z);`	NRQ
`double complex cproj(double complex z);`	NRQ
`float complex cprojf(float complex z);`	NRQ
`long double complex cprojl(long double complex z);`	NRQ
`double creal(double complex z);`	NRQ
`float crealf(float complex z);`	NRQ
`long double creall(long double complex z);`	NRQ

B.5.3 Character handling `<ctype.h>`

Table B-28: `<ctype.h>` Functions

Function	AEP
`int isalnum(int c);`	MAN
`int isalpha(int c);`	MAN
`int isblank(int c);`	MAN
`int iscntrl(int c);`	MAN
`int isdigit(int c);`	MAN
`int isgraph(int c);`	MAN
`int islower(int c);`	MAN
`int isprint(int c);`	MAN
`int ispunct(int c);`	MAN
`int isspace(int c);`	MAN
`int isupper(int c);`	MAN
`int isxdigit(int c);`	MAN
`int tolower(int c);`	MAN
`int toupper(int c);`	MAN

B.5.4 Floating-point environment `<fenv.h>`

Table B-29: `<fenv.h>` Functions

Function	AEP
`void feclearexcept(int excepts);`	NRQ

Function	AEP
`void fegetexceptflag(fexcept_t *flagp, int excepts);`	NRQ
`void feraiseexcept(int excepts);`	NRQ
`void fesetexceptflag(const fexcept_t *flagp, int excepts);`	NRQ
`int fetestexcept(int excepts);`	NRQ
`int fegetround(void);`	NRQ
`int fesetround(int round);`	NRQ
`void fegetenv(fenv_t *envp);`	NRQ
`int feholdexcept(fenv_t *envp);`	NRQ
`void fesetenv(const fenv_t *envp);`	NRQ
`void feupdateenv(const fenv_t *envp);`	NRQ

B.5.5 Format conversion of integer types **<inttypes.h>**

Table B-30: <inttypes.h> Functions

Function	AEP
`intmax_t imaxabs(intmax_t j);`	NRQ
`imaxdiv_t imaxdiv(intmax_t numer, intmax_t denom);`	NRQ
`intmax_t strtoimax(const char * restrict nptr, char ** restrict endptr, int base);`	NRQ
`uintmax_t strtoumax(const char * restrict nptr, char ** restrict endptr, int base);`	NRQ
`intmax_t wcstoimax(const wchar_t * restrict nptr, wchar_t ** restrict endptr, int base);`	NRQ
`uintmax_t wcstoumax(const wchar_t * restrict nptr, wchar_t ** restrict endptr, int base);`	NRQ

B.5.6 Localization **<locale.h>**

Table B-25: <locale.h> Functions

Function	AEP
`char *setlocale(int category, const char *locale);`	MAN
`struct lconv *localeconv(void);`	NRQ

B.5.7 Mathematics **<math.h>**

The Macros and definitions introduced by C99 and IEEE TR 19768 are not required by the profile.

Table B-26: <math.h> Functions

Function	AEP
`double acos(double x);`	MAN
`long double acosl(long double x);`	NRQ
`double asin(double x);`	MAN
`long double asinl(long double x);`	NRQ
`double atan(double x);`	MAN
`long double atanl(long double x);`	NRQ
`double atan2(double y, double x);`	MAN
`long double atan2l(long double y, long double x);`	NRQ
`double cos(double x);`	MAN
`long double cosl(long double x);`	NRQ
`double sin(double x);`	MAN
`long double sinl(long double x);`	NRQ
`double tan(double x);`	MAN
`long double tanl(long double x);`	NRQ
`double acosh(double x);`	NRQ
`long double acoshl(long double x);`	NRQ
`double asinh(double x);`	NRQ
`long double asinhl(long double x);`	NRQ
`double atanh(double x);`	NRQ
`long double atanhl(long double x);`	NRQ
`double cosh(double x);`	MAN
`long double coshl(long double x);`	NRQ
`double sinh(double x);`	MAN
`long double sinhl(long double x);`	NRQ
`double tanh(double x);`	MAN
`long double tanhl(long double x);`	NRQ
`double exp(double x);`	MAN
`long double expl(long double x);`	NRQ
`double exp2(double x);`	NRQ
`long double exp2l(long double x);`	NRQ
`double expm1(double x);`	NRQ
`long double expm1l(long double x);`	NRQ

Function	AEP
`double frexp(double value, int *exp);`	MAN
`long double frexpl(long double value, int *exp);`	NRQ
`int ilogb(double x);`	NRQ
`int ilogbl(long double x);`	NRQ
`double ldexp(double x, int exp);`	MAN
`long double ldexpl(long double x, int exp);`	NRQ
`double log(double x);`	MAN
`long double logl(long double x);`	NRQ
`double log10(double x);`	MAN
`long double log10l(long double x);`	NRQ
`double log1p(double x);`	NRQ
`long double log1pl(long double x);`	NRQ
`double log2(double x);`	NRQ
`long double log2l(long double x);`	NRQ
`double logb(double x);`	NRQ
`long double logbl(long double x);`	NRQ
`double modf(double value, double *iptr);`	MAN
`long double modfl(long double value, long double *iptr);`	NRQ
`double scalbn(double x, int n);`	NRQ
`long double scalbnl(long double x, int n);`	NRQ
`double scalbln(double x, long int n);`	NRQ
`long double scalblnl(long double x, long int n);`	NRQ
`double cbrt(double x);`	NRQ
`long double cbrtl(long double x);`	NRQ
`double fabs(double x);`	MAN
`long double fabsl(long double x);`	NRQ
`double hypot(double x, double y);`	NRQ
`long double hypotl(long double x, long double y);`	NRQ
`double pow(double x, double y);`	MAN
`long double powl(long double x, long double y);`	NRQ
`double sqrt(double x);`	MAN
`long double sqrtl(long double x);`	NRQ

Function	AEP
`double erf(double x);`	NRQ
`long double erfl(long double x);`	NRQ
`double erfc(double x);`	NRQ
`long double erfcl(long double x);`	NRQ
`double lgamma(double x);`	NRQ
`long double lgammal(long double x);`	NRQ
`double tgamma(double x);`	NRQ
`long double tgammal(long double x);`	NRQ
`double ceil(double x);`	MAN
`long double ceill(long double x);`	NRQ
`double floor(double x);`	MAN
`long double floorl(long double x);`	NRQ
`double nearbyint(double x);`	NRQ
`long double nearbyintl(long double x);`	NRQ
`double rint(double x);`	NRQ
`long double rintl(long double x);`	NRQ
`long int lrint(double x);`	NRQ
`long int lrintl(long double x);`	NRQ
`long long int llrint(double x);`	NRQ
`long long int llrintl(long double x);`	NRQ
`double round(double x);`	NRQ
`long double roundl(long double x);`	NRQ
`long int lround(double x);`	NRQ
`long int lroundl(long double x);`	NRQ
`long long int llround(double x);`	NRQ
`long long int llroundl(long double x);`	NRQ
`double trunc(double x);`	NRQ
`long double truncl(long double x);`	NRQ
`double fmod(double x, double y);`	MAN
`long double fmodl(long double x, long double y);`	NRQ
`double remainder(double x, double y);`	NRQ
`long double remainderl(long double x, long double y);`	NRQ

Function	AEP
`double remquo(double x, double y, int *quo);`	NRQ
`long double remquol(long double x, long double y,int *quo);`	NRQ
`double copysign(double x, double y);`	NRQ
`long double copysignl(long double x, long double y);`	NRQ
`double nan(const char *tagp);`	NRQ
`long double nanl(const char *tagp);`	NRQ
`double nextafter(double x, double y);`	NRQ
`long double nextafterl(long double x, long double y);`	NRQ
`double nexttoward(double x, long double y);`	NRQ
`long double nexttowardl(long double x, long double y);`	NRQ
`double fdim(double x, double y);`	NRQ
`long double fdiml(long double x, long double y);`	NRQ
`double fmax(double x, double y);`	NRQ
`long double fmaxl(long double x, long double y);`	NRQ
`double fmin(double x, double y);`	NRQ
`long double fminl(long double x, long double y);`	NRQ
`double fma(double x, double y, double z);`	NRQ
`long double fmal(long double x, long double y,long double z);`	NRQ
`float acosf(float x);`	NRQ
`float asinf(float x);`	NRQ
`float atanf(float x);`	NRQ
`float atan2f(float y, float x);`	NRQ
`float cosf(float x);`	NRQ
`float sinf(float x);`	NRQ
`float tanf(float x);`	NRQ
`float acoshf(float x);`	NRQ
`float asinhf(float x);`	NRQ
`float atanhf(float x);`	NRQ
`float coshf(float x);`	NRQ
`float sinhf(float x);`	NRQ
`float tanhf(float x);`	NRQ

Function	AEP
`float expf(float x);`	NRQ
`float exp2f(float x);`	NRQ
`float expm1f(float x);`	NRQ
`float frexpf(float value, int *exp);`	NRQ
`int ilogbf(float x);`	NRQ
`float ldexpf(float x, int exp);`	NRQ
`float logf(float x);`	NRQ
`float log10f(float x);`	NRQ
`float log1pf(float x);`	NRQ
`float log2f(float x);`	NRQ
`float logbf(float x);`	NRQ
`float modff(float value, float *iptr);`	NRQ
`float scalbnf(float x, int n);`	NRQ
`float scalblnf(float x, long int n);`	NRQ
`float cbrtf(float x);`	NRQ
`float fabsf(float x);`	NRQ
`float hypotf(float x, float y);`	NRQ
`float powf(float x, float y);`	NRQ
`float sqrtf(float x);`	NRQ
`float erff(float x);`	NRQ
`float erfcf(float x);`	NRQ
`float lgammaf(float x);`	NRQ
`float tgammaf(float x);`	NRQ
`float ceilf(float x);`	NRQ
`float floorf(float x);`	NRQ
`float nearbyintf(float x);`	NRQ
`float rintf(float x);`	NRQ
`long int lrintf(float x);`	NRQ
`long long int llrintf(float x);`	NRQ
`float roundf(float x);`	NRQ
`long int lroundf(float x);`	NRQ
`long long int llroundf(float x);`	NRQ

Function	AEP
`float truncf(float x);`	NRQ
`float fmodf(float x, float y);`	NRQ
`float remainderf(float x, float y);`	NRQ
`float remquof(float x, float y, int *quo);`	NRQ
`float copysignf(float x, float y);`	NRQ
`float nanf(const char *tagp);`	NRQ
`float nextafterf(float x, float y);`	NRQ
`float nexttowardf(float x, long double y);`	NRQ
`float fdimf(float x, float y);`	NRQ
`float fmaxf(float x, float y);`	NRQ
`float fminf(float x, float y);`	NRQ
`float fmaf(float x, float y, float z);`	NRQ

B.5.8 Nonlocal jumps **<setjmp.h>**

Table B-27: <setjmp.h> Functions

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

B.5.9 Signal handling **<signal.h>**

Table B-28: <signal.h> Functions

Function	AEP
`void (*signal(int sig, void (*func)(int)))(int);`	MAN
`int raise(int sig);`	MAN

B.5.10 Variable arguments **<stdarg.h>**

Table B-29: <stdarg.h> Functions

Function	AEP
`void va_arg(va_list ap, type);`	MAN
`void va_copy(va_list dest, va_list src);`	NRQ
`void va_end(va_list ap);`	MAN

Function	AEP
`void va_start(va_list ap, parmN);`	MAN

B.5.11 Input/output `<stdio.h>`

Table B-30: `<stdio.h>` Functions

Function	AEP
`int remove(const char *filename);`	MAN
`int rename(const char *old, const char *new);`	MAN
`int fclose(FILE *stream);`	MAN
`int fflush(FILE *stream);`	MAN
`FILE *fopen(const char * restrict filename, const char * restrict mode);`	MAN
`FILE *freopen(const char * restrict filename, const char * restrict mode,FILE * restrict stream);`	MAN
`void setbuf(FILE * restrict stream, char * restrict buf);`	MAN
`int setvbuf(FILE * restrict stream, char * restrict buf,int mode, size_t size);`	MAN
`int fprintf(FILE * restrict stream, const char * restrict format, ...);`	MAN
`int fscanf(FILE * restrict stream, const char * restrict format, ...);`	MAN
`int printf(const char * restrict format, ...);`	MAN
`int snprintf(char * restrict s, size_t n, const char * restrict format, ...);`	MAN
`int sscanf(const char * restrict s, const char * restrict format, ...);`	MAN
`int fgetc(FILE *stream);`	MAN
`char *fgets(char * restrict s, int n, FILE * restrict stream);`	MAN
`int fputc(int c, FILE *stream);`	MAN
`int fputs(const char * restrict s, FILE * restrict stream);`	MAN
`int getc(FILE *stream);`	MAN
`int getchar(void);`	MAN
`int putc(int c, FILE *stream);`	MAN
`int putchar(int c);`	MAN
`int ungetc(int c, FILE *stream);`	MAN

Function	AEP
`size_t fread(void * restrict ptr,size_t size, size_t nmemb, FILE * restrict stream);`	MAN
`size_t fwrite(const void * restrict ptr, size_t size, size_t nmemb, FILE * restrict stream);`	MAN
`int fseek(FILE *stream, long int offset, int whence);`	MAN
`long int ftell(FILE *stream);`	MAN
`void rewind(FILE *stream);`	MAN
`void clearerr(FILE *stream);`	MAN
`int feof(FILE *stream);`	MAN
`int ferror(FILE *stream);`	MAN
`void perror(const char *s);`	MAN
`FILE *tmpfile(void);`	NRQ
`char *tmpnam(char *s);`	NRQ
`int fgetpos(FILE * restrict stream, fpos_t * restrict pos);`	NRQ
`int fsetpos(FILE *stream, const fpos_t *pos);`	NRQ
`char *gets(char *s);`	NRQ
`int puts(const char *s);`	NRQ
`int scanf(const char * restrict format, ...);`	NRQ
`int vfscanf(FILE * restrict stream, const char * restrict format, va_list arg);`	NRQ
`int vscanf(const char * restrict format, va_list arg);`	NRQ
`int vsscanf(const char * restrict s, const char * restrict format, va_list arg);`	NRQ
`int sprintf(char * restrict s, const char * restrict format, ...);`	NRQ(1)
`int vfprintf(FILE * restrict stream, const char * restrict format, va_list arg);`	NRQ
`int vprintf(const char * restrict format, va_list arg);`	NRQ
`int vsnprintf(char * restrict s, size_t n, const char * restrict format, va_list arg);`	MAN
`int vsprintf(char * restrict s, const char * restrict format, va_list arg);`	NRQ

(1) removed in favor of snprintf

B.5.12 General utilities **<stdlib.h>**

Table B-31: <stdlib.h> Functions

Function	AEP
`double atof(const char *nptr);`	MAN
`int atoi(const char *nptr);`	MAN
`long int atol(const char *nptr);`	MAN
`long long int atoll(const char *nptr);`	NRQ
`double strtod(const char * restrict nptr, char ** restrict endptr);`	MAN
`long double strtold(const char * restrict nptr,char ** restrict endptr);`	NRQ
`long int strtol(const char * restrict nptr,char ** restrict endptr, int base);`	MAN
`long long int strtoll(const char * restrict nptr,char ** restrict endptr, int base);`	NRQ
`unsigned long int strtoul(const char * restrict nptr,char ** restrict endptr, int base);`	MAN
`unsigned long long int strtoull(const char * restrict nptr,char ** restrict endptr, int base);`	NRQ
`int rand(void);`	MAN
`void srand(unsigned int seed);`	MAN
`void *calloc(size_t nmemb, size_t size);`	MAN
`void free(void *ptr);`	MAN
`void *malloc(size_t size);`	MAN
`void *realloc(void *ptr, size_t size);`	MAN
`void abort(void);`	MAN
`void *bsearch(const void *key, const void *base,size_t nmemb, size_t size,int (*compar)(const void *, const void *));`	MAN
`void qsort(void *base, size_t nmemb, size_t size,int (*compar)(const void *, const void *));`	MAN
`int abs(int j);`	MAN
`long int labs(long int j);`	MAN
`long long int llabs(long long int j);`	NRQ
`div_t div(int numer, int denom);`	NRQ
`ldiv_t ldiv(long int numer, long int denom);`	NRQ
`lldiv_t lldiv(long long int numer, long long int denom);`	NRQ

Function	AEP
`int mblen(const char *s, size_t n);`	NRQ
`int mbtowc(wchar_t * restrict pwc,const char * restrict s, size_t n);`	NRQ
`int wctomb(char *s, wchar_t wchar);`	NRQ
`size_t mbstowcs(wchar_t * restrict pwcs,const char * restrict s, size_t n);`	NRQ
`size_t wcstombs(char * restrict s, const wchar_t * restrict pwcs, size_t n);`	NRQ
`float strtof(const char * restrict nptr,char ** restrict endptr);`	NRQ
`int atexit(void (*func)(void));`	NRQ
`void exit(int status);`	NRQ
`void _Exit(int status);`	NRQ
`char *getenv(const char *name);`	NRQ
`int system(const char *string);`	NRQ

B.5.13 String handling `<string.h>`

Table B-32: `<string.h>` Functions

Function	AEP
`void *memcpy(void * restrict s1,const void * restrict s2, size_t n);`	MAN
`char *strncpy(char * restrict s1,const char * restrict s2, size_t n);`	MAN
`char *strncat(char * restrict s1,const char * restrict s2, size_t n);`	MAN
`int memcmp(const void *s1, const void *s2, size_t n);`	MAN
`int strcmp(const char *s1, const char *s2);`	MAN
`int strcoll(const char *s1, const char *s2);`	MAN
`int strncmp(const char *s1, const char *s2, size_t n);`	MAN
`size_t strxfrm(char * restrict s1,const char * restrict s2, size_t n);`	MAN
`void *memchr(const void *s, int c, size_t n);`	MAN
`char *strchr(const char *s, int c);`	MAN
`size_t strcspn(const char *s1, const char *s2);`	MAN
`char *strpbrk(const char *s1, const char *s2);`	MAN
`char *strrchr(const char *s, int c);`	MAN

Function	AEP
`size_t strspn(const char *s1, const char *s2);`	MAN
`char *strstr(const char *s1, const char *s2);`	MAN
`char *strtok(char * restrict s1,const char * restrict s2);`	MAN
`void *memset(void *s, int c, size_t n);`	MAN
`size_t strlen(const char *s);`	MAN
`void *memmove(void *s1, const void *s2, size_t n);`	MAN
`char *strcat(char * restrict s1,const char * restrict s2);`	NRQ
`char *strcpy(char * restrict s1,const char * restrict s2);`	NRQ
`char *strerror(int errnum);`	MAN

B.5.14 Type-generic math **<tgmath.h>**

Table B-33: <tgmath.h> Functions

Function	AEP
`acos`	NRQ
`asin`	NRQ
`atan`	NRQ
`acosh`	NRQ
`asinh`	NRQ
`atanh`	NRQ
`cos`	NRQ
`sin`	NRQ
`tan`	NRQ
`cosh`	NRQ
`sinh`	NRQ
`tanh`	NRQ
`exp`	NRQ
`log`	NRQ
`pow`	NRQ
`sqrt`	NRQ
`fabs`	NRQ
`atan2`	NRQ
`cbrt`	NRQ
`ceil`	NRQ

Function	AEP
copysign	NRQ
erf	NRQ
erfc	NRQ
exp2	NRQ
expm1	NRQ
fdim	NRQ
floor	NRQ
fma	NRQ
fmax	NRQ
fmin	NRQ
fmod	NRQ
frexp	NRQ
hypot	NRQ
ilogb	NRQ
ldexp	NRQ
lgamma	NRQ
llrint	NRQ
llround	NRQ
log10	NRQ
log1p	NRQ
log2	NRQ
logb	NRQ
lrint	NRQ
lround	NRQ
nearbyint	NRQ
nextafter	NRQ
nexttoward	NRQ
remainder	NRQ
remquo	NRQ
rint	NRQ
round	NRQ
scalbn	NRQ
scalbln	NRQ
tgamma	NRQ

Function	AEP
`trunc`	NRQ
`carg`	NRQ
`cimag`	NRQ
`conj`	NRQ
`cproj`	NRQ
`creal`	NRQ

B.5.15 Date and time `<time.h>`

Table B-34: `<time.h>` Functions

Function	AEP
`clock_t clock(void);`	MAN
`time_t mktime(struct tm *timeptr);`	MAN
`time_t time(time_t *timer);`	MAN
`char *asctime(const struct tm *timeptr);`	MAN
`char *ctime(const time_t *timer);`	MAN
`struct tm *gmtime(const time_t *timer);`	MAN
`struct tm *localtime(const time_t *timer);`	MAN
`size_t strftime(char * restrict s,size_t maxsize,const char * restrict format,const struct tm * restrict timeptr);`	MAN
`double difftime(time_t time1, time_t time0);`	NRQ

B.5.16 Extended multibyte/wide character utilities `<wchar.h>`

Table B-35: `<wchar.h>` Functions

Function	AEP
`int fwprintf(FILE * restrict stream,const wchar_t * restrict format, ...);`	NRQ
`int fwscanf(FILE * restrict stream,const wchar_t * restrict format, ...);`	NRQ
`int swprintf(wchar_t * restrict s, size_t n,const wchar_t * restrict format, ...);`	NRQ
`int swscanf(const wchar_t * restrict s,const wchar_t * restrict format, ...);`	NRQ
`int vfwprintf(FILE * restrict stream,const wchar_t * restrict format, va_list arg);`	NRQ

Function	AEP
`int vfwscanf(FILE * restrict stream,const wchar_t * restrict format, va_list arg);`	NRQ
`int vswprintf(wchar_t * restrict s, size_t n,const wchar_t * restrict format,va_list arg);`	NRQ
`int vswscanf(const wchar_t * restrict s,const wchar_t * restrict format,va_list arg);`	NRQ
`int vwprintf(const wchar_t * restrict format,va_list arg);`	NRQ
`int vwscanf(const wchar_t * restrict format,va_list arg);`	NRQ
`int wprintf(const wchar_t * restrict format, ...);`	NRQ
`int wscanf(const wchar_t * restrict format, ...);`	NRQ
`wint_t fgetwc(FILE *stream);`	NRQ
`wchar_t *fgetws(wchar_t * restrict s, int n,FILE * restrict stream);`	NRQ
`wint_t fputwc(wchar_t c, FILE *stream);`	NRQ
`int fputws(const wchar_t * restrict s,FILE * restrict stream);`	NRQ
`int fwide(FILE *stream, int mode);`	NRQ
`wint_t getwc(FILE *stream);`	NRQ
`wint_t getwchar(void);`	NRQ
`wint_t putwc(wchar_t c, FILE *stream);`	NRQ
`wint_t putwchar(wchar_t c);`	NRQ
`wint_t ungetwc(wint_t c, FILE *stream);`	NRQ
`double wcstod(const wchar_t * restrict nptr,wchar_t ** restrict endptr);`	NRQ
`float wcstof(const wchar_t * restrict nptr,wchar_t ** restrict endptr);`	NRQ
`long double wcstold(const wchar_t * restrict nptr,wchar_t ** restrict endptr);`	NRQ
`long int wcstol(const wchar_t * restrict nptr,wchar_t ** restrict endptr, int base);`	NRQ
`long long int wcstoll(const wchar_t * restrict nptr,wchar_t ** restrict endptr, int base);`	NRQ
`unsigned long int wcstoul(const wchar_t * restrict nptr,wchar_t ** restrict endptr, int base);`	NRQ
`unsigned long long int wcstoull(const wchar_t * restrict nptr, wchar_t ** restrict endptr, int base);`	NRQ
`wchar_t *wcscpy(wchar_t * restrict s1,const wchar_t * restrict s2);`	NRQ

Function	AEP
`wchar_t *wcsncpy(wchar_t * restrict s1,const wchar_t * restrict s2, size_t n);`	NRQ
`wchar_t *wcscat(wchar_t * restrict s1,const wchar_t * restrict s2);`	NRQ
`wchar_t *wcsncat(wchar_t * restrict s1,const wchar_t * restrict s2, size_t n);`	NRQ
`int wcscmp(const wchar_t *s1, const wchar_t *s2);`	NRQ
`int wcscoll(const wchar_t *s1, const wchar_t *s2);`	NRQ
`int wcsncmp(const wchar_t *s1, const wchar_t *s2,size_t n);`	NRQ
`size_t wcsxfrm(wchar_t * restrict s1,const wchar_t * restrict s2, size_t n);`	NRQ
`wchar_t *wcschr(const wchar_t *s, wchar_t c);`	NRQ
`size_t wcscspn(const wchar_t *s1, const wchar_t *s2);`	NRQ
`size_t wcslen(const wchar_t *s);`	NRQ
`wchar_t *wcspbrk(const wchar_t *s1, const wchar_t *s2);`	NRQ
`wchar_t *wcsrchr(const wchar_t *s, wchar_t c);`	NRQ
`size_t wcsspn(const wchar_t *s1, const wchar_t *s2);`	NRQ
`wchar_t *wcsstr(const wchar_t *s1, const wchar_t *s2);`	NRQ
`wchar_t *wcstok(wchar_t * restrict s1, const wchar_t * restrict s2,wchar_t ** restrict ptr);`	NRQ
`wchar_t *wmemchr(const wchar_t *s, wchar_t c, size_t n);`	NRQ
`int wmemcmp(wchar_t * restrict s1,const wchar_t * restrict s2, size_t n);`	NRQ
`wchar_t *wmemcpy(wchar_t * restrict s1,const wchar_t * restrict s2, size_t n);`	NRQ
`wchar_t *wmemmove(wchar_t *s1, const wchar_t *s2,size_t n);`	NRQ
`wchar_t *wmemset(wchar_t *s, wchar_t c, size_t n);`	NRQ
`size_t wcsftime(wchar_t * restrict s, size_t maxsize,const wchar_t * restrict format,const struct tm * restrict timeptr);`	NRQ
`wint_t btowc(int c);`	NRQ
`int wctob(wint_t c);`	NRQ
`int mbsinit(const mbstate_t *ps);`	NRQ
`size_t mbrlen(const char * restrict s, size_t n,mbstate_t * restrict ps);`	NRQ

Function	AEP
`size_t mbrtowc(wchar_t * restrict pwc, const char * restrict s, size_t n,mbstate_t * restrict ps);`	NRQ
`size_t wcrtomb(char * restrict s, wchar_t wc,mbstate_t * restrict ps);`	NRQ
`size_t mbsrtowcs(wchar_t * restrict dst,const char ** restrict src, size_t len,mbstate_t * restrict ps);`	NRQ
`size_t wcsrtombs(char * restrict dst,const wchar_t ** restrict src, size_t len,mbstate_t * restrict ps);`	NRQ

B.5.17 Wide character classification and mapping utilities **<wctype.h>**

Table B-36: <wctype.h> Functions

Function	AEP
`wint_t wctrans_t wctype_t WEOF`	NRQ
`int iswalnum(wint_t wc);`	NRQ
`int iswalpha(wint_t wc);`	NRQ
`int iswblank(wint_t wc);`	NRQ
`int iswcntrl(wint_t wc);`	NRQ
`int iswdigit(wint_t wc);`	NRQ
`int iswgraph(wint_t wc);`	NRQ
`int iswlower(wint_t wc);`	NRQ
`int iswprint(wint_t wc);`	NRQ
`int iswpunct(wint_t wc);`	NRQ
`int iswspace(wint_t wc);`	NRQ
`int iswupper(wint_t wc);`	NRQ
`int iswxdigit(wint_t wc);`	NRQ
`int iswctype(wint_t wc, wctype_t desc);`	NRQ
`wctype_t wctype(const char *property);`	NRQ
`wint_t towlower(wint_t wc);`	NRQ
`wint_t towupper(wint_t wc);`	NRQ
`wint_t towctrans(wint_t wc, wctrans_t desc);`	NRQ
`wctrans_t wctrans(const char *property);`	NRQ