

This is an extract from:

A Source Book from The Open Group

The Authorized Guide to the Single UNIX Specification, Version 3

The Open Group

Copyright © May 2003, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

A Source Book from The Open Group

The Authorized Guide to the Single UNIX Specification, Version 3

Published in the U.K. by The Open Group, May 2003.

Any comments relating to the material contained in this document may be submitted to:

The Open Group
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

OGSpecs@opengroup.org

System Interface Table

This chapter contains a table of all the interfaces defined in XSH, Issue 6, complete with an indication of their status for the XSI extension, the POSIX Base, and their availability in UNIX 98 (denoted by U98), UNIX 95 (denoted by U95), IEEE Std 1003.1-1996 (POSIX.1) (denoted P96), IEEE Std 1003.2-1992 (POSIX.2) (denoted P92), the ISO C standard (denoted C99), C 89, SVID, Issue 3 (denoted by SVID3) and BSD 4.3 (denoted BSD). The following conventions are used in columns 2 through 10:

- m** Indicates that the interface is defined as mandatory.
- o** Indicates that the interface is part of an Option or Feature Group.
- ob** Indicates that the interface is Obsolescent, and although mandatory for the implementation, applications are discouraged from its use.
- l** In the XSI column, indicates that the interface is part of the Legacy Option Group and need not be available on all implementations.
- opt* In the POSIX Base column, two or three letter option codes are used as described in Portability Codes, denoting the option to which the interface belongs.
- r** In the P96 column, this indicates that the interface is part of the POSIX Realtime Extension.
- t** In the P96 column, this indicates that the interface is part of the POSIX Threads Extension.
- 1** In the C89 column this indicates that the interface is part of the ISO/IEC 9899: 1990 standard.
- .** Indicates that the interface is not specified.

The table is intended as a quick reference guide for programmers migrating to or developing applications for the Single UNIX Specification, Version 3. Products that brand to a profile may not provide all of the interfaces listed, depending on which Option or Feature Groups are supported.

There are 1122 interfaces listed.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>FD_CLR()</i>	m	m	m	m
<i>FD_ISSET()</i>	m	m	m	m
<i>FD_SET()</i>	m	m	m	m
<i>FD_ZERO()</i>	m	m	m	m
<i>_Exit()</i>	m	m	.	.	.	m	.	.	.
<i>_exit()</i>	m	m	m	m	m	.	.	m	m
<i>_longjmp()</i>	m	xsi	m	m	m
<i>_setjmp()</i>	m	xsi	m	m	m
<i>_tolower()</i>	m	xsi	m	m	.	.	.	m	.
<i>_toupper()</i>	m	xsi	m	m	.	.	.	m	.
<i>a64l()</i>	m	xsi	m	m	.	.	.	m	.
<i>abort()</i>	m	m	m	m	m	m	m	m	m
<i>abs()</i>	m	m	m	m	m	m	m	m	m
<i>accept()</i>	m	m	m	m	m
<i>access()</i>	m	m	m	m	m	.	.	m	m
<i>acos()</i>	m	m	m	m	m	m	m	m	m
<i>acosf()</i>	m	m	.	.	.	m	.	.	.
<i>acosh()</i>	m	m	m	m	.	m	.	m	m
<i>acoshf()</i>	m	m	.	.	.	m	.	.	.
<i>acoshl()</i>	m	m	.	.	.	m	.	.	.
<i>acosl()</i>	m	m	.	.	.	m	.	.	.
<i>aio_cancel()</i>	o	aio	o	.	r
<i>aio_error()</i>	o	aio	o	.	r
<i>aio_fsync()</i>	o	aio	o	.	r
<i>aio_read()</i>	o	aio	o	.	r
<i>aio_return()</i>	o	aio	o	.	r
<i>aio_suspend()</i>	o	aio	o	.	r
<i>aio_write()</i>	o	aio	o	.	r
<i>alarm()</i>	m	m	m	m	m	.	.	m	m
<i>asctime()</i>	m	m	m	m	m	m	m	m	m
<i>asctime_r()</i>	m	tsf	m	.	t
<i>asin()</i>	m	m	m	m	m	m	m	m	m
<i>asinf()</i>	m	m	.	.	.	m	.	.	.
<i>asinh()</i>	m	m	m	m	.	m	.	m	m
<i>asinhf()</i>	m	m	.	.	.	m	.	.	.
<i>asinhf()</i>	m	m	.	.	.	m	.	.	.
<i>asinl()</i>	m	m	.	.	.	m	.	.	.
<i>assert()</i>	m	m	m	m	m	m	m	m	m
<i>atan()</i>	m	m	m	m	m	m	m	m	m
<i>atan2()</i>	m	m	m	m	m	m	m	m	m
<i>atan2f()</i>	m	m	.	.	.	m	.	.	.
<i>atan2l()</i>	m	m	.	.	.	m	.	.	.
<i>atanf()</i>	m	m	.	.	.	m	.	.	.
<i>atanh()</i>	m	m	m	m	.	m	.	m	m
<i>atanhf()</i>	m	m	.	.	.	m	.	.	.
<i>atanhl()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>atanl()</i>	m	m	.	.	.	m	.	.	.
<i>atexit()</i>	m	m	m	m	.	m	m	m	.
<i>atof()</i>	m	m	m	m	m	m	m	m	m
<i>atoi()</i>	m	m	m	m	m	m	m	m	m
<i>atol()</i>	m	m	m	m	m	m	m	m	m
<i>atoll()</i>	m	m	.	.	.	m	.	.	.
<i>basename()</i>	m	xsi	m	m
<i>bcmp()</i>	l	xsi	m	m	m
<i>bcopy()</i>	l	xsi	m	m	m
<i>bind()</i>	m	m	m	m	m
<i>bsd_signal()</i>	ob	xsi	m	m
<i>bsearch()</i>	m	m	m	m	m	m	m	m	.
<i>btowc()</i>	m	m	m	.	.	m	1	.	.
<i>bzero()</i>	l	xsi	m	m	m
<i>cabs()</i>	m	m	.	.	.	m	.	.	.
<i>cabsf()</i>	m	m	.	.	.	m	.	.	.
<i>cabsl()</i>	m	m	.	.	.	m	.	.	.
<i>cacos()</i>	m	m	.	.	.	m	.	.	.
<i>cacosf()</i>	m	m	.	.	.	m	.	.	.
<i>cacosh()</i>	m	m	.	.	.	m	.	.	.
<i>cacoshf()</i>	m	m	.	.	.	m	.	.	.
<i>cacoshl()</i>	m	m	.	.	.	m	.	.	.
<i>cacosl()</i>	m	m	.	.	.	m	.	.	.
<i>calloc()</i>	m	m	m	m	m	m	m	m	m
<i>carg()</i>	m	m	.	.	.	m	.	.	.
<i>cargf()</i>	m	m	.	.	.	m	.	.	.
<i>cargl()</i>	m	m	.	.	.	m	.	.	.
<i>casin()</i>	m	m	.	.	.	m	.	.	.
<i>casinf()</i>	m	m	.	.	.	m	.	.	.
<i>casinh()</i>	m	m	.	.	.	m	.	.	.
<i>casinhf()</i>	m	m	.	.	.	m	.	.	.
<i>casinhl()</i>	m	m	.	.	.	m	.	.	.
<i>casinl()</i>	m	m	.	.	.	m	.	.	.
<i>catan()</i>	m	m	.	.	.	m	.	.	.
<i>catanf()</i>	m	m	.	.	.	m	.	.	.
<i>catanh()</i>	m	m	.	.	.	m	.	.	.
<i>catanhf()</i>	m	m	.	.	.	m	.	.	.
<i>catanhl()</i>	m	m	.	.	.	m	.	.	.
<i>catanl()</i>	m	m	.	.	.	m	.	.	.
<i>catclose()</i>	m	xsi	m	m	.	.	.	m	.
<i>catgets()</i>	m	xsi	m	m	.	.	.	m	.
<i>catopen()</i>	m	xsi	m	m	.	.	.	m	.
<i>cbrt()</i>	m	m	m	m	.	m	.	m	m
<i>cbrtf()</i>	m	m	.	.	.	m	.	.	.
<i>cbrtl()</i>	m	m	.	.	.	m	.	.	.
<i>ccos()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>ccosf()</i>	m	m	.	.	.	m	.	.	.
<i>ccosh()</i>	m	m	.	.	.	m	.	.	.
<i>ccoshf()</i>	m	m	.	.	.	m	.	.	.
<i>ccoshl()</i>	m	m	.	.	.	m	.	.	.
<i>ccosl()</i>	m	m	.	.	.	m	.	.	.
<i>ceil()</i>	m	m	m	m	m	m	m	m	m
<i>ceilf()</i>	m	m	.	.	.	m	.	.	.
<i>ceilll()</i>	m	m	.	.	.	m	.	.	.
<i>cexp()</i>	m	m	.	.	.	m	.	.	.
<i>cexpf()</i>	m	m	.	.	.	m	.	.	.
<i>cexpl()</i>	m	m	.	.	.	m	.	.	.
<i>cfgetispeed()</i>	m	m	m	m	m	.	.	m	.
<i>cfgetospeed()</i>	m	m	m	m	m	.	.	m	.
<i>cfsetispeed()</i>	m	m	m	m	m	.	.	m	.
<i>cfsetospeed()</i>	m	m	m	m	m	.	.	m	.
<i>chdir()</i>	m	m	m	m	m	.	.	m	m
<i>chmod()</i>	m	m	m	m	m	.	.	m	m
<i>chown()</i>	m	m	m	m	m	.	.	m	m
<i>cimag()</i>	m	m	.	.	.	m	.	.	.
<i>cimagf()</i>	m	m	.	.	.	m	.	.	.
<i>cimagl()</i>	m	m	.	.	.	m	.	.	.
<i>clearerr()</i>	m	m	m	m	m	m	m	m	m
<i>clock()</i>	m	m	m	m	.	m	m	m	.
<i>clock_getcpuclockid()</i>	o	cpt
<i>clock_getres()</i>	o	tmr	o	.	r
<i>clock_gettime()</i>	o	tmr	o	.	r
<i>clock_nanosleep()</i>	o	cs
<i>clock_settime()</i>	o	tmr	o	.	r
<i>clog()</i>	m	m	.	.	.	m	.	.	.
<i>clogf()</i>	m	m	.	.	.	m	.	.	.
<i>clogl()</i>	m	m	.	.	.	m	.	.	.
<i>close()</i>	m	m	m	m	m	.	.	m	m
<i>closedir()</i>	m	m	m	m	m	.	.	m	m
<i>closelog()</i>	m	xsi	m	m	m
<i>confstr()</i>	m	m	m	m	m
<i>conj()</i>	m	m	.	.	.	m	.	.	.
<i>conjf()</i>	m	m	.	.	.	m	.	.	.
<i>conjl()</i>	m	m	.	.	.	m	.	.	.
<i>connect()</i>	m	m	m	m	m
<i>copysign()</i>	m	m	.	.	.	m	.	.	.
<i>copysignf()</i>	m	m	.	.	.	m	.	.	.
<i>copysignl()</i>	m	m	.	.	.	m	.	.	.
<i>cos()</i>	m	m	m	m	m	m	m	m	m
<i>cosf()</i>	m	m	.	.	.	m	.	.	.
<i>cosh()</i>	m	m	m	m	m	m	m	m	m
<i>coshf()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>coshl()</i>	m	m	.	.	.	m	.	.	.
<i>cosl()</i>	m	m	.	.	.	m	.	.	.
<i>cpow()</i>	m	m	.	.	.	m	.	.	.
<i>cpowf()</i>	m	m	.	.	.	m	.	.	.
<i>cpowl()</i>	m	m	.	.	.	m	.	.	.
<i>cproj()</i>	m	m	.	.	.	m	.	.	.
<i>cprojf()</i>	m	m	.	.	.	m	.	.	.
<i>cprojl()</i>	m	m	.	.	.	m	.	.	.
<i>creal()</i>	m	m	.	.	.	m	.	.	.
<i>crealf()</i>	m	m	.	.	.	m	.	.	.
<i>creall()</i>	m	m	.	.	.	m	.	.	.
<i>creat()</i>	m	m	m	m	m	.	.	m	m
<i>crypt()</i>	o	xsi	o	o	.	.	.	m	m
<i>csin()</i>	m	m	.	.	.	m	.	.	.
<i>csinf()</i>	m	m	.	.	.	m	.	.	.
<i>csinh()</i>	m	m	.	.	.	m	.	.	.
<i>csinhf()</i>	m	m	.	.	.	m	.	.	.
<i>csinhl()</i>	m	m	.	.	.	m	.	.	.
<i>csinl()</i>	m	m	.	.	.	m	.	.	.
<i>csqrt()</i>	m	m	.	.	.	m	.	.	.
<i>csqrtf()</i>	m	m	.	.	.	m	.	.	.
<i>csqrtl()</i>	m	m	.	.	.	m	.	.	.
<i>ctan()</i>	m	m	.	.	.	m	.	.	.
<i>ctanf()</i>	m	m	.	.	.	m	.	.	.
<i>ctanh()</i>	m	m	.	.	.	m	.	.	.
<i>ctanhf()</i>	m	m	.	.	.	m	.	.	.
<i>ctanhl()</i>	m	m	.	.	.	m	.	.	.
<i>ctanl()</i>	m	m	.	.	.	m	.	.	.
<i>ctermid()</i>	m	m	m	m	m	.	.	m	.
<i>ctime()</i>	m	m	m	m	m	m	m	m	m
<i>ctime_r()</i>	m	tsf	m	.	t
<i>daylight</i>	m	xsi	m	m	.	.	.	m	.
<i>dbm_clearerr()</i>	m	xsi	m	m
<i>dbm_close()</i>	m	xsi	m	m
<i>dbm_delete()</i>	m	xsi	m	m
<i>dbm_error()</i>	m	xsi	m	m
<i>dbm_fetch()</i>	m	xsi	m	m
<i>dbm_firstkey()</i>	m	xsi	m	m
<i>dbm_nextkey()</i>	m	xsi	m	m
<i>dbm_open()</i>	m	xsi	m	m
<i>dbm_store()</i>	m	xsi	m	m
<i>difftime()</i>	m	m	m	m	.	m	m	m	.
<i>dirname()</i>	m	xsi	m	m
<i>div()</i>	m	m	m	m	.	m	m	m	.
<i>dlclose()</i>	m	xsi	m
<i>dLError()</i>	m	xsi	m

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>dlopen()</i>	m	xsi	m
<i>dlsym()</i>	m	xsi	m
<i>drand48()</i>	m	xsi	m	m	.	.	.	m	.
<i>dup()</i>	m	m	m	m	m	.	.	m	m
<i>dup2()</i>	m	m	m	m	m	.	.	m	m
<i>ecvt()</i>	l	xsi	m	m	m
<i>encrypt()</i>	o	xsi	o	o	.	.	.	m	m
<i>endgrent()</i>	m	xsi	m	m	.	.	.	m	m
<i>endhostent()</i>	m	m	m	m	m
<i>endnetent()</i>	m	m	m	m	m
<i>endprotoent()</i>	m	m	m	m	m
<i>endpwent()</i>	m	xsi	m	m	.	.	.	m	m
<i>endservent()</i>	m	m	m	m	m
<i>endutxent()</i>	m	xsi	m	m
<i>environ</i>	m	m	m	m	m	.	.	m	m
<i>erand48()</i>	m	xsi	m	m	.	.	.	m	.
<i>erf()</i>	m	m	m	m	.	m	.	m	m
<i>erfc()</i>	m	m	m	m	.	m	.	m	m
<i>erfcf()</i>	m	m	.	.	.	m	.	.	.
<i>erfcl()</i>	m	m	.	.	.	m	.	.	.
<i>erff()</i>	m	m	.	.	.	m	.	.	.
<i>erfl()</i>	m	m	.	.	.	m	.	.	.
<i>errno</i>	m	m	m	m	m	m	m	m	m
<i>execl()</i>	m	m	m	m	m	.	.	m	m
<i>execle()</i>	m	m	m	m	m	.	.	m	m
<i>execlp()</i>	m	m	m	m	m	.	.	m	m
<i>execv()</i>	m	m	m	m	m	.	.	m	m
<i>execve()</i>	m	m	m	m	m	.	.	m	m
<i>execvp()</i>	m	m	m	m	m	.	.	m	m
<i>exit()</i>	m	m	m	m	m	m	m	m	m
<i>exp()</i>	m	m	m	m	m	m	m	m	m
<i>exp2()</i>	m	m	.	.	.	m	.	.	.
<i>exp2f()</i>	m	m	.	.	.	m	.	.	.
<i>exp2l()</i>	m	m	.	.	.	m	.	.	.
<i>expf()</i>	m	m	.	.	.	m	.	.	.
<i>expl()</i>	m	m	.	.	.	m	.	.	.
<i>expm1()</i>	m	m	m	m	.	m	.	.	m
<i>expm1f()</i>	m	m	.	.	.	m	.	.	.
<i>expm1l()</i>	m	m	.	.	.	m	.	.	.
<i>fabs()</i>	m	m	m	m	m	m	m	m	m
<i>fabsf()</i>	m	m	.	.	.	m	.	.	.
<i>fabsl()</i>	m	m	.	.	.	m	.	.	.
<i>fattach()</i>	o	xsr	m	m	.	.	.	m	.
<i>fchdir()</i>	m	xsi	m	m	.	.	.	m	.
<i>fchmod()</i>	m	m	m	m	.	.	.	m	m
<i>fchown()</i>	m	m	m	m	.	.	.	m	m

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>fclose()</i>	m	m	m	m	m	m	m	m	m
<i>fcntl()</i>	m	m	m	m	m	.	.	m	m
<i>fcvt()</i>	l	xsi	m	m	m
<i>fdatasync()</i>	o	sio	o	.	r
<i>fdetach()</i>	o	xsr	m	m	.	.	.	m	.
<i>fdim()</i>	m	m	.	.	.	m	.	.	.
<i>fdimf()</i>	m	m	.	.	.	m	.	.	.
<i>fdiml()</i>	m	m	.	.	.	m	.	.	.
<i>fdopen()</i>	m	m	m	m	m	.	.	m	m
<i>feclearexcept()</i>	m	m	.	.	.	m	.	.	.
<i>fegetenv()</i>	m	m	.	.	.	m	.	.	.
<i>fegetexceptflag()</i>	m	m	.	.	.	m	.	.	.
<i>fegetround()</i>	m	m	.	.	.	m	.	.	.
<i>feholdexcept()</i>	m	m	.	.	.	m	.	.	.
<i>feof()</i>	m	m	m	m	m	m	m	m	m
<i>feraiseexcept()</i>	m	m	.	.	.	m	.	.	.
<i>ferror()</i>	m	m	m	m	m	m	m	m	m
<i>fesetenv()</i>	m	m	.	.	.	m	.	.	.
<i>fesetexceptflag()</i>	m	m	.	.	.	m	.	.	.
<i>fesetround()</i>	m	m	.	.	.	m	.	.	.
<i>fetestexcept()</i>	m	m	.	.	.	m	.	.	.
<i>feupdateenv()</i>	m	m	.	.	.	m	.	.	.
<i>fflush()</i>	m	m	m	m	m	m	m	m	m
<i>ffs()</i>	m	xsi	m	m	m
<i>fgetc()</i>	m	m	m	m	m	m	m	m	m
<i>fgetpos()</i>	m	m	m	m	.	m	m	m	.
<i>fgets()</i>	m	m	m	m	m	m	m	m	m
<i>fgetwc()</i>	m	m	m	m	.	m	1	.	.
<i>fgetws()</i>	m	m	m	m	.	m	1	.	.
<i>fileno()</i>	m	m	m	m	m	.	.	m	m
<i>flockfile()</i>	m	tsf	m	.	t
<i>floor()</i>	m	m	m	m	m	m	m	m	m
<i>floorf()</i>	m	m	.	.	.	m	.	.	.
<i>floorl()</i>	m	m	.	.	.	m	.	.	.
<i>fma()</i>	m	m	.	.	.	m	.	.	.
<i>fmaf()</i>	m	m	.	.	.	m	.	.	.
<i>fmal()</i>	m	m	.	.	.	m	.	.	.
<i>fmax()</i>	m	m	.	.	.	m	.	.	.
<i>fmaxf()</i>	m	m	.	.	.	m	.	.	.
<i>fmaxl()</i>	m	m	.	.	.	m	.	.	.
<i>fmin()</i>	m	m	.	.	.	m	.	.	.
<i>fminf()</i>	m	m	.	.	.	m	.	.	.
<i>fminl()</i>	m	m	.	.	.	m	.	.	.
<i>fmod()</i>	m	m	m	m	m	m	m	m	.
<i>fmodf()</i>	m	m	.	.	.	m	.	.	.
<i>fmodl()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>fmtmsg()</i>	m	xsi	m	m	.	.	.	m	.
<i>fnmatch()</i>	m	m	m	m	m
<i>fopen()</i>	m	m	m	m	m	m	m	m	m
<i>fork()</i>	m	m	m	m	m	.	.	m	m
<i>fpathconf()</i>	m	m	m	m	m	.	.	m	.
<i>fpclassify()</i>	m	m	.	.	.	m	.	.	.
<i>fprintf()</i>	m	m	m	m	m	m	m	m	m
<i>fputc()</i>	m	m	m	m	m	m	m	m	m
<i>fputs()</i>	m	m	m	m	m	m	m	m	m
<i>fputwc()</i>	m	m	m	m	.	m	1	.	.
<i>fputws()</i>	m	m	m	m	.	m	1	.	.
<i>fread()</i>	m	m	m	m	m	m	m	m	m
<i>free()</i>	m	m	m	m	m	m	m	m	m
<i>freeaddrinfo()</i>	m	m
<i>freopen()</i>	m	m	m	m	m	m	m	m	m
<i>frexp()</i>	m	m	m	m	m	m	m	m	m
<i>frexpf()</i>	m	m	.	.	.	m	.	.	.
<i>frexpl()</i>	m	m	.	.	.	m	.	.	.
<i>fscanf()</i>	m	m	m	m	m	m	m	m	m
<i>fseek()</i>	m	m	m	m	m	m	m	m	m
<i>fseeko()</i>	m	m	m
<i>fsetpos()</i>	m	m	m	m	.	m	m	m	.
<i>fstat()</i>	m	m	m	m	m	.	.	m	m
<i>fstatvfs()</i>	m	xsi	m	m	.	.	.	m	.
<i>fsync()</i>	m	fsc	m	m	.	.	.	m	m
<i>ftell()</i>	m	m	m	m	m	m	m	m	m
<i>ftello()</i>	m	m	m
<i>ftime()</i>	l	xsi	m	m	m
<i>ftok()</i>	m	xsi	m	m
<i>ftruncate()</i>	m	m	m	m
<i>ftwlockfile()</i>	m	tsf	m	.	t
<i>ftw()</i>	m	xsi	m	m	.	.	.	m	.
<i>funlockfile()</i>	m	tsf	m	.	t
<i>fwide()</i>	m	m	m	.	.	m	1	.	.
<i>fwprintf()</i>	m	m	m	.	.	m	1	.	.
<i>fwrite()</i>	m	m	m	m	m	m	m	m	m
<i>fwscanf()</i>	m	m	m	.	.	m	1	.	.
<i>gai_strerror()</i>	m	m
<i>gcvt()</i>	l	xsi	m	m	m
<i>getaddrinfo()</i>	m	m
<i>getc()</i>	m	m	m	m	m	m	m	m	m
<i>getc_unlocked()</i>	m	tsf	m	.	t
<i>getchar()</i>	m	m	m	m	m	m	m	m	m
<i>getchar_unlocked()</i>	m	tsf	m	.	t
<i>getcontext()</i>	m	xsi	m	m	.	.	.	m	.
<i>getcwd()</i>	m	m	m	m	m	.	.	m	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>getdate()</i>	m	xsi	m	m	.	.	.	m	.
<i>getegid()</i>	m	m	m	m	m	.	.	m	m
<i>getenv()</i>	m	m	m	m	m	m	m	m	m
<i>geteuid()</i>	m	m	m	m	m	.	.	m	m
<i>getgid()</i>	m	m	m	m	m	.	.	m	m
<i>getgrent()</i>	m	xsi	m	m	.	.	.	m	m
<i>getgrgid()</i>	m	m	m	m	m	.	.	m	m
<i>getgrgid_r()</i>	m	tsf	m	.	t
<i>getgrnam()</i>	m	m	m	m	m	.	.	m	.
<i>getgrnam_r()</i>	m	tsf	m	.	t
<i>getgroups()</i>	m	m	m	m	m	.	.	m	m
<i>gethostbyaddr()</i>	ob	ob	m	m	m
<i>gethostbyname()</i>	ob	ob	m	m	m
<i>gethostent()</i>	m	m	m	m	m
<i>gethostid()</i>	m	xsi	m	m	m
<i>gethostname()</i>	m	m	m	m	m
<i>getitimer()</i>	m	xsi	m	m	.	.	.	m	m
<i>getlogin()</i>	m	m	m	m	m	.	.	m	m
<i>getlogin_r()</i>	m	tsf	m	.	t
<i>getmsg()</i>	o	xsr	m	m	.	.	.	m	.
<i>getnameinfo()</i>	m	m
<i>getnetbyaddr()</i>	m	m	m	m	m
<i>getnetbyname()</i>	m	m	m	m	m
<i>getnetent()</i>	m	m	m	m	m
<i>getopt()</i>	m	m	m	m	m	.	.	m	m
<i>getpeername()</i>	m	m	m	m	m
<i>getpgid()</i>	m	xsi	m	m	.	.	.	m	.
<i>getpgrp()</i>	m	m	m	m	m	.	.	m	m
<i>getpid()</i>	m	m	m	m	m	.	.	m	m
<i>getpmsg()</i>	o	xsr	m	m	.	.	.	m	.
<i>getppid()</i>	m	m	m	m	m	.	.	m	m
<i>getpriority()</i>	m	xsi	m	m	m
<i>getprotobyname()</i>	m	m	m	m	m
<i>getprotobynumber()</i>	m	m	m	m	m
<i>getprotoent()</i>	m	m	m	m	m
<i>getpwent()</i>	m	xsi	m	m	.	.	.	m	m
<i>getpwnam()</i>	m	m	m	m	m	.	.	m	m
<i>getpwnam_r()</i>	m	tsf	m	.	t
<i>getpwuid()</i>	m	m	m	m	m	.	.	m	m
<i>getpwuid_r()</i>	m	tsf	m	.	t
<i>getrlimit()</i>	m	xsi	m	m	.	.	.	m	m
<i>getrusage()</i>	m	xsi	m	m	m
<i>gets()</i>	m	m	m	m	m	m	m	m	m
<i>getservbyname()</i>	m	m	m	m	m
<i>getservbyport()</i>	m	m	m	m	m
<i>getservent()</i>	m	m	m	m	m

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>getsid()</i>	m	xsi	m	m	.	.	.	m	.
<i>getsockname()</i>	m	m	m	m	m
<i>getsockopt()</i>	m	m	m	m	m
<i>getsubopt()</i>	m	xsi	m	m	.	.	.	m	.
<i>gettimeofday()</i>	m	xsi	m	m	.	.	.	m	m
<i>getuid()</i>	m	m	m	m	m	.	.	m	m
<i>getutxent()</i>	m	xsi	m	m
<i>getutxid()</i>	m	xsi	m	m
<i>getutxline()</i>	m	xsi	m	m
<i>getwc()</i>	m	m	m	m	.	m	1	.	.
<i>getwchar()</i>	m	m	m	m	.	m	1	.	.
<i>getwd()</i>	l	xsi	m	m	m
<i>glob()</i>	m	m	m	o	m
<i>globfree()</i>	m	m	m	o	m
<i>gmtime()</i>	m	m	m	m	m	m	m	m	m
<i>gmtime_r()</i>	m	tsf	m	.	t
<i>grantpt()</i>	m	xsi	m	m	.	.	.	m	.
<i>h_errno</i>	ob	ob
<i>hcreate()</i>	m	xsi	m	m	.	.	.	m	.
<i>hdestroy()</i>	m	xsi	m	m	.	.	.	m	.
<i>hsearch()</i>	m	xsi	m	m	.	.	.	m	.
<i>htonl()</i>	m	m	m	m	m
<i>htons()</i>	m	m	m	m	m
<i>hypot()</i>	m	m	m	m	.	m	.	m	m
<i>hypotf()</i>	m	m	.	.	.	m	.	.	.
<i>hypotl()</i>	m	m	.	.	.	m	.	.	.
<i>iconv()</i>	m	xsi	m	m
<i>iconv_close()</i>	m	xsi	m	m
<i>iconv_open()</i>	m	xsi	m	m
<i>if_freenameindex()</i>	m	m
<i>if_indextoname()</i>	m	m
<i>if_nameindex()</i>	m	m
<i>if_nametoindex()</i>	m	m
<i>ilogb()</i>	m	m	m	m	.	m	.	.	.
<i>ilogbf()</i>	m	m	.	.	.	m	.	.	.
<i>ilogbl()</i>	m	m	.	.	.	m	.	.	.
<i>imaxabs()</i>	m	m	.	.	.	m	.	.	.
<i>imaxdiv()</i>	m	m	.	.	.	m	.	.	.
<i>index()</i>	l	xsi	m	m	m
<i>inet_addr()</i>	m	m	m	m	m
<i>inet_ntoa()</i>	m	m	m	m	m
<i>inet_ntop()</i>	m	m
<i>inet_pton()</i>	m	m
<i>initstate()</i>	m	xsi	m	m	m
<i>insque()</i>	m	xsi	m	m	m
<i>ioctl()</i>	o	xsr	m	m	.	.	.	m	m

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>isalnum()</i>	m	m	m	m	m	m	m	m	m
<i>isalpha()</i>	m	m	m	m	m	m	m	m	m
<i>isascii()</i>	m	xsi	m	m	.	.	.	m	m
<i>isastream()</i>	o	xsr	m	m	.	.	.	m	.
<i>isatty()</i>	m	m	m	m	m	.	.	m	m
<i>isblank()</i>	m	m	.	.	.	m	.	.	.
<i>iscntrl()</i>	m	m	m	m	m	m	m	m	m
<i>isdigit()</i>	m	m	m	m	m	m	m	m	m
<i>isfinite()</i>	m	m	.	.	.	m	.	.	.
<i>isgraph()</i>	m	m	m	m	m	m	m	m	m
<i>isgreater()</i>	m	m	.	.	.	m	.	.	.
<i>isgreaterequal()</i>	m	m	.	.	.	m	.	.	.
<i>isinf()</i>	m	m	.	.	.	m	.	.	.
<i>isless()</i>	m	m	.	.	.	m	.	.	.
<i>islessequal()</i>	m	m	.	.	.	m	.	.	.
<i>islessgreater()</i>	m	m	.	.	.	m	.	.	.
<i>islower()</i>	m	m	m	m	m	m	m	m	m
<i>isnan()</i>	m	m	m	m	.	.	.	m	.
<i>isnormal()</i>	m	m	.	.	.	m	.	.	.
<i>isprint()</i>	m	m	m	m	m	m	m	m	m
<i>ispunct()</i>	m	m	m	m	m	m	m	m	m
<i>isspace()</i>	m	m	m	m	m	m	m	m	m
<i>isunordered()</i>	m	m	.	.	.	m	.	.	.
<i>isupper()</i>	m	m	m	m	m	m	m	m	m
<i>iswalnum()</i>	m	m	m	m	.	m	1	.	.
<i>iswalpha()</i>	m	m	m	m	.	m	1	.	.
<i>iswblank()</i>	m	m	.	.	.	m	.	.	.
<i>iswcntrl()</i>	m	m	m	m	.	m	1	.	.
<i>iswctype()</i>	m	m	m	m	.	m	1	.	.
<i>iswdigit()</i>	m	m	m	m	.	m	1	.	.
<i>iswgraph()</i>	m	m	m	m	.	m	1	.	.
<i>iswlower()</i>	m	m	m	m	.	m	1	.	.
<i>iswprint()</i>	m	m	m	m	.	m	1	.	.
<i>iswpunct()</i>	m	m	m	m	.	m	1	.	.
<i>iswspace()</i>	m	m	m	m	.	m	1	.	.
<i>iswupper()</i>	m	m	m	m	.	m	1	.	.
<i>iswxdigit()</i>	m	m	m	m	.	m	1	.	.
<i>isxdigit()</i>	m	m	m	m	m	m	m	m	m
<i>j0()</i>	m	xsi	m	m	.	.	.	m	m
<i>j1()</i>	m	xsi	m	m	.	.	.	m	m
<i>jn()</i>	m	xsi	m	m	.	.	.	m	m
<i>jrands48()</i>	m	xsi	m	m	.	.	.	m	.
<i>kill()</i>	m	m	m	m	m	.	.	m	m
<i>killpg()</i>	m	xsi	m	m	m
<i>l64a()</i>	m	xsi	m	m	.	.	.	m	.
<i>labs()</i>	m	m	m	m	.	m	m	m	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>lchown()</i>	m	xsi	m	m	.	.	.	m	.
<i>lcong48()</i>	m	xsi	m	m	.	.	.	m	.
<i>ldexp()</i>	m	m	m	m	m	m	m	m	m
<i>ldexpf()</i>	m	m	.	.	.	m	.	.	.
<i>ldexpl()</i>	m	m	.	.	.	m	.	.	.
<i>ldiv()</i>	m	m	m	m	.	m	m	m	.
<i>lfind()</i>	m	xsi	m	m	.	.	.	m	.
<i>lgamma()</i>	m	m	m	m	.	m	.	m	m
<i>lgammaf()</i>	m	m	.	.	.	m	.	.	.
<i>lgammal()</i>	m	m	.	.	.	m	.	.	.
<i>link()</i>	m	m	m	m	m	.	.	m	m
<i>lio_listio()</i>	o	aio	o	.	r
<i>listen()</i>	m	m	m	m	m
<i>llabs()</i>	m	m	.	.	.	m	.	.	.
<i>lldiv()</i>	m	m	.	.	.	m	.	.	.
<i>llrint()</i>	m	m	.	.	.	m	.	.	.
<i>llrintf()</i>	m	m	.	.	.	m	.	.	.
<i>llrintl()</i>	m	m	.	.	.	m	.	.	.
<i>llround()</i>	m	m	.	.	.	m	.	.	.
<i>llroundf()</i>	m	m	.	.	.	m	.	.	.
<i>llroundl()</i>	m	m	.	.	.	m	.	.	.
<i>localeconv()</i>	m	m	m	m	.	m	m	m	.
<i>localtime()</i>	m	m	m	m	m	m	m	m	m
<i>localtime_r()</i>	m	tsf	m	.	t
<i>lockf()</i>	m	xsi	m	m	.	.	.	m	.
<i>log()</i>	m	m	m	m	m	m	m	m	m
<i>log10()</i>	m	m	m	m	m	m	m	m	m
<i>log10f()</i>	m	m	.	.	.	m	.	.	.
<i>log10l()</i>	m	m	.	.	.	m	.	.	.
<i>log1p()</i>	m	m	m	m	.	m	.	.	m
<i>log1pf()</i>	m	m	.	.	.	m	.	.	.
<i>log1pl()</i>	m	m	.	.	.	m	.	.	.
<i>log2()</i>	m	m	.	.	.	m	.	.	.
<i>log2f()</i>	m	m	.	.	.	m	.	.	.
<i>log2l()</i>	m	m	.	.	.	m	.	.	.
<i>logb()</i>	m	m	m	m	.	m	.	m	m
<i>logbf()</i>	m	m	.	.	.	m	.	.	.
<i>logbl()</i>	m	m	.	.	.	m	.	.	.
<i>logf()</i>	m	m	.	.	.	m	.	.	.
<i>logl()</i>	m	m	.	.	.	m	.	.	.
<i>longjmp()</i>	m	m	m	m	m	m	m	m	m
<i>lrand48()</i>	m	xsi	m	m	.	.	.	m	.
<i>lrint()</i>	m	m	.	.	.	m	.	.	.
<i>lrintf()</i>	m	m	.	.	.	m	.	.	.
<i>lrintl()</i>	m	m	.	.	.	m	.	.	.
<i>lround()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>lroundf()</i>	m	m	.	.	.	m	.	.	.
<i>lroundl()</i>	m	m	.	.	.	m	.	.	.
<i>lsearch()</i>	m	xsi	m	m	.	.	.	m	.
<i>lseek()</i>	m	m	m	m	m	.	.	m	m
<i>lstat()</i>	m	m	m	m	.	.	.	m	m
<i>makecontext()</i>	m	xsi	m	m
<i>malloc()</i>	m	m	m	m	m	m	m	m	m
<i>mblen()</i>	m	m	m	m	.	m	m	m	.
<i>mbrlen()</i>	m	m	m	.	.	m	1	.	.
<i>mbrtowc()</i>	m	m	m	.	.	m	1	.	.
<i>mbsinit()</i>	m	m	m	.	.	m	1	.	.
<i>mbsrtowcs()</i>	m	m	m	.	.	m	1	.	.
<i>mbstowcs()</i>	m	m	m	m	.	m	m	m	.
<i>mbtowc()</i>	m	m	m	m	.	m	m	m	.
<i>memccpy()</i>	m	xsi	m	m	.	.	.	m	.
<i>memchr()</i>	m	m	m	m	.	m	m	m	.
<i>memcmp()</i>	m	m	m	m	.	m	m	m	.
<i>memcpy()</i>	m	m	m	m	.	m	m	m	.
<i>memmove()</i>	m	m	m	m	.	m	m	m	.
<i>memset()</i>	m	m	m	m	.	m	m	m	.
<i>mkdir()</i>	m	m	m	m	m	.	.	m	m
<i>mkfifo()</i>	m	m	m	m	m	.	.	m	.
<i>mknod()</i>	m	xsi	m	m	.	.	.	m	m
<i>mkstemp()</i>	m	xsi	m	m	m
<i>mktemp()</i>	l	xsi	m	m	.	.	.	m	m
<i>mtime()</i>	m	m	m	m	m	m	m	m	.
<i>mlock()</i>	o	mr	o	.	r
<i>mlockall()</i>	o	ml	o	.	r
<i>mmap()</i>	m	mf shm tym	m	m	.	.	.	m	.
<i>modf()</i>	m	m	m	m	m	m	m	m	m
<i>modff()</i>	m	m	.	.	.	m	.	.	.
<i>modfl()</i>	m	m	.	.	.	m	.	.	.
<i>mprotect()</i>	m	mpr	m	m	.	.	.	m	.
<i>mq_close()</i>	o	msg	o	.	r
<i>mq_getattr()</i>	o	msg	o	.	r
<i>mq_notify()</i>	o	msg	o	.	r
<i>mq_open()</i>	o	msg	o	.	r
<i>mq_receive()</i>	o	msg	o	.	r
<i>mq_send()</i>	o	msg	o	.	r
<i>mq_setattr()</i>	o	msg	o	.	r
<i>mq_timedreceive()</i>	o	msg
<i>mq_timedsend()</i>	o	msg
<i>mq_unlink()</i>	o	msg	o	.	r
<i>rand48()</i>	m	xsi	m	m	.	.	.	m	.
<i>msgctl()</i>	m	xsi	m	m	.	.	.	m	.
<i>msgget()</i>	m	xsi	m	m	.	.	.	m	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>msgrcv()</i>	m	xsi	m	m	.	.	.	m	.
<i>msgsnd()</i>	m	xsi	m	m	.	.	.	m	.
<i>msync()</i>	m	mf sio	m	m	.	.	.	m	.
<i>munlock()</i>	o	mr	o	.	r
<i>munlockall()</i>	o	ml	o	.	r
<i>munmap()</i>	m	mf shm tym	m	m	.	.	.	m	.
<i>nan()</i>	m	m	.	.	.	m	.	.	.
<i>nanf()</i>	m	m	.	.	.	m	.	.	.
<i>nanl()</i>	m	m	.	.	.	m	.	.	.
<i>nanosleep()</i>	o	tmr	o	.	r
<i>nearbyint()</i>	m	m	.	.	.	m	.	.	.
<i>nearbyintf()</i>	m	m	.	.	.	m	.	.	.
<i>nearbyintl()</i>	m	m	.	.	.	m	.	.	.
<i>nextafter()</i>	m	m	m	m	.	m	.	m	.
<i>nextafterf()</i>	m	m	.	.	.	m	.	.	.
<i>nextafterl()</i>	m	m	.	.	.	m	.	.	.
<i>nexttoward()</i>	m	m	.	.	.	m	.	.	.
<i>nexttowardf()</i>	m	m	.	.	.	m	.	.	.
<i>nexttowardl()</i>	m	m	.	.	.	m	.	.	.
<i>nftw()</i>	m	xsi	m	m	.	.	.	m	.
<i>nice()</i>	m	xsi	m	m	.	.	.	m	m
<i>nl_langinfo()</i>	m	xsi	m	m	.	.	.	m	.
<i>nrnd48()</i>	m	xsi	m	m	.	.	.	m	.
<i>ntohl()</i>	m	m	m	m	m
<i>ntohs()</i>	m	m	m	m	m
<i>open()</i>	m	m	m	m	m	.	.	m	m
<i>opendir()</i>	m	m	m	m	m	.	.	m	m
<i>openlog()</i>	m	xsi	m	m	m
<i>optarg</i>	m	m	m	m	m	.	.	m	m
<i>opterr</i>	m	m	m	m	m	.	.	m	m
<i>optind</i>	m	m	m	m	m	.	.	m	m
<i>optopt</i>	m	m	m	m	m	.	.	m	m
<i>pathconf()</i>	m	m	m	m	m	.	.	m	.
<i>pause()</i>	m	m	m	m	m	.	.	m	m
<i>pclose()</i>	m	m	m	m	m	.	.	m	m
<i>perror()</i>	m	m	m	m	m	m	m	m	m
<i>pipe()</i>	m	m	m	m	m	.	.	m	m
<i>poll()</i>	m	xsi	m	m	.	.	.	m	.
<i>popen()</i>	m	m	m	m	m	.	.	m	m
<i>posix_fadvise()</i>	o	adv
<i>posix_fallocate()</i>	o	adv
<i>posix_madvise()</i>	o	adv
<i>posix_mem_offset()</i>	o	tym
<i>posix_memalign()</i>	o	adv
<i>posix_openpt()</i>	m	xsi
<i>posix_spawn()</i>	o	spn

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>posix_spawn_file_actions_addclose()</i>	0	spn
<i>posix_spawn_file_actions_adddup2()</i>	0	spn
<i>posix_spawn_file_actions_addopen()</i>	0	spn
<i>posix_spawn_file_actions_destroy()</i>	0	spn
<i>posix_spawn_file_actions_init()</i>	0	spn
<i>posix_spawnattr_destroy()</i>	0	spn
<i>posix_spawnattr_getflags()</i>	0	spn
<i>posix_spawnattr_getpgroup()</i>	0	spn
<i>posix_spawnattr_getschedparam()</i>	0	spn ps
<i>posix_spawnattr_getschedpolicy()</i>	0	spn ps
<i>posix_spawnattr_getsigdefault()</i>	0	spn
<i>posix_spawnattr_getsigmask()</i>	0	spn
<i>posix_spawnattr_init()</i>	0	spn
<i>posix_spawnattr_setflags()</i>	0	spn
<i>posix_spawnattr_setpgroup()</i>	0	spn
<i>posix_spawnattr_setschedparam()</i>	0	spn ps
<i>posix_spawnattr_setschedpolicy()</i>	0	spn ps
<i>posix_spawnattr_setsigdefault()</i>	0	spn
<i>posix_spawnattr_setsigmask()</i>	0	spn
<i>posix_spawnnp()</i>	0	spn
<i>posix_trace_attr_destroy()</i>	0	trc
<i>posix_trace_attr_getclockres()</i>	0	trc
<i>posix_trace_attr_getcreatetime()</i>	0	trc
<i>posix_trace_attr_getgenversion()</i>	0	trc
<i>posix_trace_attr_getinherited()</i>	0	trc tri
<i>posix_trace_attr_getlogfullpolicy()</i>	0	trc trl
<i>posix_trace_attr_getlogsize()</i>	0	trc trl
<i>posix_trace_attr_getmaxdatasize()</i>	0	trc
<i>posix_trace_attr_getmaxsystemeventsz()</i>	0	trc
<i>posix_trace_attr_getmaxusereventsz()</i>	0	trc
<i>posix_trace_attr_getname()</i>	0	trc
<i>posix_trace_attr_getstreamfullpolicy()</i>	0	trc
<i>posix_trace_attr_getstreamsize()</i>	0	trc
<i>posix_trace_attr_init()</i>	0	trc
<i>posix_trace_attr_setinherited()</i>	0	trc tri
<i>posix_trace_attr_setlogfullpolicy()</i>	0	trc trl
<i>posix_trace_attr_setlogsize()</i>	0	trc trl
<i>posix_trace_attr_setmaxdatasize()</i>	0	trc
<i>posix_trace_attr_setname()</i>	0	trc
<i>posix_trace_attr_setstreamfullpolicy()</i>	0	trc
<i>posix_trace_attr_setstreamsize()</i>	0	trc
<i>posix_trace_clear()</i>	0	trc
<i>posix_trace_close()</i>	0	trc trl
<i>posix_trace_create()</i>	0	trc
<i>posix_trace_create_withlog()</i>	0	trc trl
<i>posix_trace_event()</i>	0	trc

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>posix_trace_eventid_equal()</i>	o	trc
<i>posix_trace_eventid_get_name()</i>	o	trc
<i>posix_trace_eventid_open()</i>	o	trc
<i>posix_trace_eventset_add()</i>	o	trc tef
<i>posix_trace_eventset_del()</i>	o	trc tef
<i>posix_trace_eventset_empty()</i>	o	trc tef
<i>posix_trace_eventset_fill()</i>	o	trc tef
<i>posix_trace_eventset_ismember()</i>	o	trc tef
<i>posix_trace_eventtypelist_getnext_id()</i>	o	trc
<i>posix_trace_eventtypelist_rewind()</i>	o	trc
<i>posix_trace_flush()</i>	o	trc trl
<i>posix_trace_get_attr()</i>	o	trc
<i>posix_trace_get_filter()</i>	o	trc tef
<i>posix_trace_get_status()</i>	o	trc
<i>posix_trace_getnext_event()</i>	o	trc
<i>posix_trace_open()</i>	o	trc trl
<i>posix_trace_rewind()</i>	o	trc trl
<i>posix_trace_set_filter()</i>	o	trc tef
<i>posix_trace_shutdown()</i>	o	trc
<i>posix_trace_start()</i>	o	trc
<i>posix_trace_stop()</i>	o	trc
<i>posix_trace_timedgetnext_event()</i>	o	trc tmo
<i>posix_trace_trid_eventid_open()</i>	o	trc tef
<i>posix_trace_trygetnext_event()</i>	o	trc
<i>posix_typed_mem_get_info()</i>	o	tym
<i>posix_typed_mem_open()</i>	o	tym
<i>pow()</i>	m	m	m	m	m	m	m	m	m
<i>powf()</i>	m	m	.	.	.	m	.	.	.
<i>powl()</i>	m	m	.	.	.	m	.	.	.
<i>pread()</i>	m	xsi	m
<i>printf()</i>	m	m	m	m	m	m	m	m	m
<i>pselect()</i>	m	m
<i>pthread_atfork()</i>	m	thr	m	.	t
<i>pthread_attr_destroy()</i>	m	thr	m	.	t
<i>pthread_attr_getdetachstate()</i>	m	thr	m	.	t
<i>pthread_attr_getguardsize()</i>	m	xsi	m
<i>pthread_attr_getinheritsched()</i>	o	thr tps	o	.	t
<i>pthread_attr_getschedparam()</i>	m	thr	m	.	t
<i>pthread_attr_getschedpolicy()</i>	o	thr tps	o	.	t
<i>pthread_attr_getscope()</i>	o	thr tps	o	.	t
<i>pthread_attr_getstack()</i>	m	thr tsa tss
<i>pthread_attr_getstackaddr()</i>	m	thr tsa	m	.	t
<i>pthread_attr_getstacksize()</i>	m	thr tss	m	.	t
<i>pthread_attr_init()</i>	m	thr	m	.	t
<i>pthread_attr_setdetachstate()</i>	m	thr	m	.	t
<i>pthread_attr_setguardsize()</i>	m	xsi	m

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>pthread_attr_setinheritsched()</i>	o	thr tps	o	.	t
<i>pthread_attr_setschedparam()</i>	m	thr	m	.	t
<i>pthread_attr_setschedpolicy()</i>	o	thr tps	o	.	t
<i>pthread_attr_setscope()</i>	o	thr tps	o	.	t
<i>pthread_attr_setstack()</i>	m	thr tsa tss
<i>pthread_attr_setstackaddr()</i>	m	thr tsa	m	.	t
<i>pthread_attr_setstacksize()</i>	m	thr tss	m	.	t
<i>pthread_barrier_destroy()</i>	o	thr bar
<i>pthread_barrier_init()</i>	o	thr bar
<i>pthread_barrier_wait()</i>	o	thr bar
<i>pthread_barrierattr_destroy()</i>	o	thr bar
<i>pthread_barrierattr_getpshared()</i>	o	thr bar tsh
<i>pthread_barrierattr_init()</i>	o	thr bar
<i>pthread_barrierattr_setpshared()</i>	o	thr bar tsh
<i>pthread_cancel()</i>	m	thr	m	.	t
<i>pthread_cleanup_pop()</i>	m	thr	m	.	t
<i>pthread_cleanup_push()</i>	m	thr	m	.	t
<i>pthread_cond_broadcast()</i>	m	thr	m	.	t
<i>pthread_cond_destroy()</i>	m	thr	m	.	t
<i>pthread_cond_init()</i>	m	thr	m	.	t
<i>pthread_cond_signal()</i>	m	thr	m	.	t
<i>pthread_cond_timedwait()</i>	m	thr	m	.	t
<i>pthread_cond_wait()</i>	m	thr	m	.	t
<i>pthread_condattr_destroy()</i>	m	thr	m	.	t
<i>pthread_condattr_getclock()</i>	o	thr cs
<i>pthread_condattr_getpshared()</i>	m	thr tsh	m	.	t
<i>pthread_condattr_init()</i>	m	thr	m	.	t
<i>pthread_condattr_setclock()</i>	o	thr cs
<i>pthread_condattr_setpshared()</i>	m	thr tsh	m	.	t
<i>pthread_create()</i>	m	thr	m	.	t
<i>pthread_detach()</i>	m	thr	m	.	t
<i>pthread_equal()</i>	m	thr	m	.	t
<i>pthread_exit()</i>	m	thr	m	.	t
<i>pthread_getconcurrency()</i>	m	xsi	m
<i>pthread_getcpuclockid()</i>	o	thr tct
<i>pthread_getschedparam()</i>	o	thr tps	o	.	t
<i>pthread_getspecific()</i>	m	thr	m	.	t
<i>pthread_join()</i>	m	thr	m	.	t
<i>pthread_key_create()</i>	m	thr	m	.	t
<i>pthread_key_delete()</i>	m	thr	m	.	t
<i>pthread_kill()</i>	m	thr	m	.	t
<i>pthread_mutex_destroy()</i>	m	thr	m	.	t
<i>pthread_mutex_getprioceiling()</i>	o	thr tpp	o	.	t
<i>pthread_mutex_init()</i>	m	thr	m	.	t
<i>pthread_mutex_lock()</i>	m	thr	m	.	t
<i>pthread_mutex_setprioceiling()</i>	o	thr tpp	o	.	t

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>pthread_mutex_timedlock()</i>	o	thr tmo
<i>pthread_mutex_trylock()</i>	m	thr	m	.	t
<i>pthread_mutex_unlock()</i>	m	thr	m	.	t
<i>pthread_mutexattr_destroy()</i>	m	thr	m	.	t
<i>pthread_mutexattr_getprioceiling()</i>	o	thr tpp	o	.	t
<i>pthread_mutexattr_getprotocol()</i>	o	thr tpp tpi	o	.	t
<i>pthread_mutexattr_getpshared()</i>	m	thr tsh	m	.	t
<i>pthread_mutexattr_gettype()</i>	m	xsi	m
<i>pthread_mutexattr_init()</i>	m	thr	m	.	t
<i>pthread_mutexattr_setprioceiling()</i>	o	thr tpp	o	.	t
<i>pthread_mutexattr_setprotocol()</i>	o	thr tpp tpi	o	.	t
<i>pthread_mutexattr_setpshared()</i>	m	thr tsh	m	.	t
<i>pthread_mutexattr_settype()</i>	m	xsi	m
<i>pthread_once()</i>	m	thr	m	.	t
<i>pthread_rwlock_destroy()</i>	m	thr	m
<i>pthread_rwlock_init()</i>	m	thr	m
<i>pthread_rwlock_rdlock()</i>	m	thr	m
<i>pthread_rwlock_timedrdlock()</i>	o	thr tmo
<i>pthread_rwlock_timedwrlock()</i>	o	thr tmo
<i>pthread_rwlock_tryrdlock()</i>	m	thr	m
<i>pthread_rwlock_trywrlock()</i>	m	thr	m
<i>pthread_rwlock_unlock()</i>	m	thr	m
<i>pthread_rwlock_wrlock()</i>	m	thr	m
<i>pthread_rwlockattr_destroy()</i>	m	thr	m
<i>pthread_rwlockattr_getpshared()</i>	m	thr tsh	m
<i>pthread_rwlockattr_init()</i>	m	thr	m
<i>pthread_rwlockattr_setpshared()</i>	m	thr tsh	m
<i>pthread_self()</i>	m	thr	m	.	t
<i>pthread_setcancelstate()</i>	m	thr	m	.	t
<i>pthread_setcanceltype()</i>	m	thr	m	.	t
<i>pthread_setconcurrency()</i>	m	xsi	m
<i>pthread_setschedparam()</i>	o	thr tps	o	.	t
<i>pthread_setschedprio()</i>	o	thr tps
<i>pthread_setspecific()</i>	m	thr	m	.	t
<i>pthread_sigmask()</i>	m	thr	m	.	t
<i>pthread_spin_destroy()</i>	o	thr spi
<i>pthread_spin_init()</i>	o	thr spi
<i>pthread_spin_lock()</i>	o	thr spi
<i>pthread_spin_trylock()</i>	o	thr spi
<i>pthread_spin_unlock()</i>	o	thr spi
<i>pthread_testcancel()</i>	m	thr	m	.	t
<i>ptsname()</i>	m	xsi	m	m	.	.	.	m	.
<i>putc()</i>	m	m	m	m	m	m	m	m	m
<i>putc_unlocked()</i>	m	tsf	m	.	t
<i>putchar()</i>	m	m	m	m	m	m	m	m	m
<i>putchar_unlocked()</i>	m	tsf	m	.	t

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>putenv()</i>	m	xsi	m	m	.	.	.	m	.
<i>putmsg()</i>	o	xsr	m	m	.	.	.	m	.
<i>putpmsg()</i>	o	xsr	m	m	.	.	.	m	.
<i>puts()</i>	m	m	m	m	m	m	m	m	m
<i>pututxline()</i>	m	xsi	m	m
<i>putwc()</i>	m	m	m	m	.	m	1	.	.
<i>putwchar()</i>	m	m	m	m	.	m	1	.	.
<i>pwrite()</i>	m	xsi	m
<i>qsort()</i>	m	m	m	m	m	m	m	m	m
<i>raise()</i>	m	m	m	m	.	m	m	m	.
<i>rand()</i>	m	m	m	m	m	m	m	m	m
<i>rand_r()</i>	m	tsf	m	.	t
<i>random()</i>	m	xsi	m	m	m
<i>read()</i>	m	xsi	m	m	m	.	.	m	m
<i>readdir()</i>	m	m	m	m	m	.	.	m	m
<i>readdir_r()</i>	m	tsf	m	.	t
<i>readlink()</i>	m	m	m	m	.	.	.	m	m
<i>readv()</i>	m	xsi	m	m	.	.	.	m	m
<i>realloc()</i>	m	m	m	m	m	m	m	m	m
<i>realpath()</i>	m	xsi	m	m
<i>recv()</i>	m	m	m	m	m
<i>recvfrom()</i>	m	m	m	m	m
<i>recvmsg()</i>	m	m	m	m	m
<i>regcomp()</i>	m	m	m	o	m
<i>regerror()</i>	m	m	m	o	m
<i>regexec()</i>	m	m	m	o	m
<i>regfree()</i>	m	m	m	o	m
<i>remainder()</i>	m	m	m	m	.	m	.	m	.
<i>remainderf()</i>	m	m	.	.	.	m	.	.	.
<i>remainderl()</i>	m	m	.	.	.	m	.	.	.
<i>remove()</i>	m	m	m	m	m	m	m	m	.
<i>remque()</i>	m	xsi	m	m	m
<i>remquo()</i>	m	m	.	.	.	m	.	.	.
<i>remquof()</i>	m	m	.	.	.	m	.	.	.
<i>remquol()</i>	m	m	.	.	.	m	.	.	.
<i>rename()</i>	m	m	m	m	m	m	m	m	m
<i>rewind()</i>	m	m	m	m	m	m	m	m	m
<i>rewinddir()</i>	m	m	m	m	m	.	.	m	m
<i>rindex()</i>	l	xsi	m	m	m
<i>rint()</i>	m	m	m	m	.	m	.	.	m
<i>rintf()</i>	m	m	.	.	.	m	.	.	.
<i>rintl()</i>	m	m	.	.	.	m	.	.	.
<i>rmdir()</i>	m	m	m	m	m	.	.	m	m
<i>round()</i>	m	m	.	.	.	m	.	.	.
<i>roundf()</i>	m	m	.	.	.	m	.	.	.
<i>roundl()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>scalb()</i>	ob	xsi	m	m	.	.	.	m	m
<i>scalbln()</i>	m	m	.	.	.	m	.	.	.
<i>scalblnf()</i>	m	m	.	.	.	m	.	.	.
<i>scalbln()</i>	m	m	.	.	.	m	.	.	.
<i>scalbn()</i>	m	m	.	.	.	m	.	.	.
<i>scalbnf()</i>	m	m	.	.	.	m	.	.	.
<i>scalbnl()</i>	m	m	.	.	.	m	.	.	.
<i>scanf()</i>	m	m	m	m	m	m	m	m	m
<i>sched_get_priority_max()</i>	o	ps	o	.	r
<i>sched_get_priority_min()</i>	o	ps	o	.	r
<i>sched_getparam()</i>	o	ps	o	.	r
<i>sched_getscheduler()</i>	o	ps	o	.	r
<i>sched_rr_get_interval()</i>	o	ps	o	.	r
<i>sched_setparam()</i>	o	ps	o	.	r
<i>sched_setscheduler()</i>	o	ps	o	.	r
<i>sched_yield()</i>	m	ps thr	m	.	r
<i>seed48()</i>	m	xsi	m	m	.	.	.	m	.
<i>seekdir()</i>	m	xsi	m	m	.	.	.	m	m
<i>select()</i>	m	m	m	m	m
<i>sem_close()</i>	o	sem	o	.	r
<i>sem_destroy()</i>	o	sem	o	.	r
<i>sem_getvalue()</i>	o	sem	o	.	r
<i>sem_init()</i>	o	sem	o	.	r
<i>sem_open()</i>	o	sem	o	.	r
<i>sem_post()</i>	o	sem	o	.	r
<i>sem_timedwait()</i>	o	sem tmo
<i>sem_trywait()</i>	o	sem	o	.	r
<i>sem_unlink()</i>	o	sem	o	.	r
<i>sem_wait()</i>	o	sem	o	.	r
<i>semctl()</i>	m	xsi	m	m	.	.	.	m	.
<i>semget()</i>	m	xsi	m	m	.	.	.	m	.
<i>semop()</i>	m	xsi	m	m	.	.	.	m	.
<i>send()</i>	m	m	m	m	m
<i>sendmsg()</i>	m	m	m	m	m
<i>sendto()</i>	m	m	m	m	m
<i>setbuf()</i>	m	m	m	m	m	m	m	m	m
<i>setcontext()</i>	m	xsi	m	m	.	.	.	m	.
<i>setegid()</i>	m	m
<i>setenv()</i>	m	m
<i>seteuid()</i>	m	m
<i>setgid()</i>	m	m	m	m	m	.	.	m	m
<i>setgrent()</i>	m	xsi	m	m	.	.	.	m	m
<i>sethostent()</i>	m	m	m	m	m
<i>setitimer()</i>	m	xsi	m	m	.	.	.	m	m
<i>setjmp()</i>	m	m	m	m	m	m	m	m	m
<i>setkey()</i>	o	xsi	o	o	.	.	.	m	m

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>setlocale()</i>	m	m	m	m	m	m	m	m	.
<i>setlogmask()</i>	m	xsi	m	m	m
<i>setnetent()</i>	m	m	m	m	m
<i>setpgid()</i>	m	m	m	m	m	.	.	m	.
<i>setpgrp()</i>	m	xsi	m	m	.	.	.	m	m
<i>setpriority()</i>	m	xsi	m	m	m
<i>setprotoent()</i>	m	m	m	m	m
<i>setpwent()</i>	m	xsi	m	m	.	.	.	m	m
<i>setregid()</i>	m	xsi	m	m	m
<i>setreuid()</i>	m	xsi	m	m	m
<i>setrlimit()</i>	m	xsi	m	m	.	.	.	m	m
<i>setservent()</i>	m	m	m	m	m
<i>setsid()</i>	m	m	m	m	m	.	.	m	.
<i>setsockopt()</i>	m	m	m	m	m
<i>setstate()</i>	m	xsi	m	m	m
<i>setuid()</i>	m	m	m	m	m	.	.	m	m
<i>setutxent()</i>	m	xsi	m	m
<i>setvbuf()</i>	m	m	m	m	.	m	m	m	.
<i>shm_open()</i>	o	shm	o	.	r
<i>shm_unlink()</i>	o	shm	o	.	r
<i>shmat()</i>	m	xsi	m	o	.	.	.	m	.
<i>shmctl()</i>	m	xsi	m	o	.	.	.	m	.
<i>shmdt()</i>	m	xsi	m	o	.	.	.	m	.
<i>shmget()</i>	m	xsi	m	o	.	.	.	m	.
<i>shutdown()</i>	m	m	m	m	m
<i>sigaction()</i>	m	m	m	m	m	.	.	m	.
<i>sigaddset()</i>	m	m	m	m	m	.	.	m	.
<i>sigaltstack()</i>	m	xsi	m	m	.	.	.	m	.
<i>sigdelset()</i>	m	m	m	m	m	.	.	m	.
<i>sigemptyset()</i>	m	m	m	m	m	.	.	m	.
<i>sigfillset()</i>	m	m	m	m	m	.	.	m	.
<i>sighold()</i>	m	xsi	m	m	.	.	.	m	.
<i>sigignore()</i>	m	xsi	m	m	.	.	.	m	.
<i>siginterrupt()</i>	m	xsi	m	m	m
<i>sigismember()</i>	m	m	m	m	m	.	.	m	.
<i>siglongjmp()</i>	m	m	m	m	m	.	.	m	.
<i>signal()</i>	m	m	m	m	.	m	m	m	m
<i>sigbit()</i>	m	m	.	.	.	m	.	.	.
<i>siggam</i>	m	xsi	m	.
<i>sigpause()</i>	m	xsi	m	m	.	.	.	m	m
<i>sigpending()</i>	m	m	m	m	m	.	.	m	.
<i>sigprocmask()</i>	m	thr	m	m	m	.	.	m	.
<i>sigqueue()</i>	o	rts	o	.	r
<i>sigrelse()</i>	m	xsi	m	m	.	.	.	m	.
<i>sigset()</i>	m	xsi	m	m	.	.	.	m	.
<i>sigsetjmp()</i>	m	m	m	m	m	.	.	m	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>sigsuspend()</i>	m	m	m	m	m	.	.	m	.
<i>sigtimedwait()</i>	o	rts	o	.	r
<i>sigwait()</i>	m	m	m	.	t
<i>sigwaitinfo()</i>	o	rts	o	.	r
<i>sin()</i>	m	m	m	m	m	m	m	m	m
<i>sinf()</i>	m	m	.	.	.	m	.	.	.
<i>sinh()</i>	m	m	m	m	m	m	m	m	m
<i>sinhf()</i>	m	m	.	.	.	m	.	.	.
<i>sinhl()</i>	m	m	.	.	.	m	.	.	.
<i>sinl()</i>	m	m	.	.	.	m	.	.	.
<i>sleep()</i>	m	m	m	m	m	.	.	m	m
<i>snprintf()</i>	m	m	m	.	.	m	.	.	.
<i>socketmark()</i>	m	m
<i>socket()</i>	m	m	m	m	m
<i>socketpair()</i>	m	m	m	m	m
<i>sprintf()</i>	m	m	m	m	m	m	m	m	m
<i>sqrt()</i>	m	m	m	m	m	m	m	m	m
<i>sqrtf()</i>	m	m	.	.	.	m	.	.	.
<i>sqrtl()</i>	m	m	.	.	.	m	.	.	.
<i>srand()</i>	m	m	m	m	m	m	m	m	m
<i>srand48()</i>	m	xsi	m	m	.	.	.	m	.
<i>srandom()</i>	m	xsi	m	m	m
<i>sscanf()</i>	m	m	m	m	m	m	m	m	m
<i>stat()</i>	m	m	m	m	m	.	.	m	m
<i>statvfs()</i>	m	xsi	m	m	.	.	.	m	.
<i>stderr</i>	m	m	m	m	m	m	m	m	m
<i>stdin</i>	m	m	m	m	m	m	m	m	m
<i>stdout</i>	m	m	m	m	m	m	m	m	m
<i>strcasecmp()</i>	m	xsi	m	m
<i>strcat()</i>	m	m	m	m	m	m	m	m	m
<i>strchr()</i>	m	m	m	m	m	m	m	m	.
<i>strcmp()</i>	m	m	m	m	m	m	m	m	m
<i>strcoll()</i>	m	m	m	m	.	m	m	m	.
<i>strcpy()</i>	m	m	m	m	m	m	m	m	m
<i>strcspn()</i>	m	m	m	m	m	m	m	m	.
<i>strdup()</i>	m	xsi	m	m	.	.	.	m	.
<i>strerror()</i>	m	m	m	m	.	m	m	m	.
<i>strerror_r()</i>	m	tsf
<i>strfmon()</i>	m	xsi	m	o
<i>strftime()</i>	m	m	m	m	m	m	m	m	.
<i>strlen()</i>	m	m	m	m	m	m	m	m	m
<i>strncasecmp()</i>	m	xsi	m	m
<i>strncat()</i>	m	m	m	m	m	m	m	m	m
<i>strncmp()</i>	m	m	m	m	m	m	m	m	m
<i>strncpy()</i>	m	m	m	m	m	m	m	m	m
<i>strpbrk()</i>	m	m	m	m	m	m	m	m	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>strptime()</i>	m	xsi	m	o
<i>strchr()</i>	m	m	m	m	m	m	m	m	.
<i>strspn()</i>	m	m	m	m	m	m	m	m	.
<i>strstr()</i>	m	m	m	m	m	m	m	m	.
<i>strtod()</i>	m	m	m	m	.	m	m	m	.
<i>strtof()</i>	m	m	.	.	.	m	.	.	.
<i>strtoimax()</i>	m	m	.	.	.	m	.	.	.
<i>strtok()</i>	m	m	m	m	m	m	m	m	.
<i>strtok_r()</i>	m	tsf	m	.	t
<i>strtol()</i>	m	m	m	m	.	m	m	m	.
<i>strtold()</i>	m	m	.	.	.	m	.	.	.
<i>strtoll()</i>	m	m	.	.	.	m	.	.	.
<i>strtoul()</i>	m	m	m	m	.	m	m	m	.
<i>strtoull()</i>	m	m	.	.	.	m	.	.	.
<i>strtoumax()</i>	m	m	.	.	.	m	.	.	.
<i>strxfrm()</i>	m	m	m	m	.	m	m	m	.
<i>swab()</i>	m	xsi	m	m	.	.	.	m	m
<i>swapcontext()</i>	m	xsi	m	m
<i>swprintf()</i>	m	m	m	.	.	m	1	.	.
<i>swscanf()</i>	m	m	m	.	.	m	1	.	.
<i>symlink()</i>	m	m	m	m	.	.	.	m	m
<i>sync()</i>	m	xsi	m	m	.	.	.	m	m
<i>sysconf()</i>	m	m	m	m	m	.	.	m	.
<i>syslog()</i>	m	xsi	m	m	m
<i>system()</i>	m	m	m	m	.	m	m	m	m
<i>tan()</i>	m	m	m	m	m	m	m	m	m
<i>tanf()</i>	m	m	.	.	.	m	.	.	.
<i>tanh()</i>	m	m	m	m	m	m	m	m	m
<i>tanhf()</i>	m	m	.	.	.	m	.	.	.
<i>tanhl()</i>	m	m	.	.	.	m	.	.	.
<i>tanl()</i>	m	m	.	.	.	m	.	.	.
<i>tcdrain()</i>	m	m	m	m	m	.	.	m	.
<i>tcflow()</i>	m	m	m	m	m	.	.	m	.
<i>tcflush()</i>	m	m	m	m	m	.	.	m	.
<i>tcgetattr()</i>	m	m	m	m	m	.	.	m	.
<i>tcgetpgrp()</i>	m	m	m	m	m	.	.	m	.
<i>tcgetsid()</i>	m	xsi	m	m	.	.	.	m	.
<i>tcsendbreak()</i>	m	m	m	m	m	.	.	m	.
<i>tcsetattr()</i>	m	m	m	m	m	.	.	m	.
<i>tcsetpgrp()</i>	m	m	m	m	m	.	.	m	.
<i>tdelete()</i>	m	xsi	m	m	.	.	.	m	.
<i>telldir()</i>	m	xsi	m	m	.	.	.	m	m
<i>tempnam()</i>	m	xsi	m	m	.	.	.	m	.
<i>tfind()</i>	m	xsi	m	m	.	.	.	m	.
<i>tgamma()</i>	m	m	.	.	.	m	.	.	.
<i>tgammaf()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96	C99	C89	SVID3	BSD
<i>tgammal()</i>	m	m	.	.	.	m	.	.	.
<i>time()</i>	m	m	m	m	m	m	m	m	m
<i>timer_create()</i>	o	tmr	o	.	r
<i>timer_delete()</i>	o	tmr	o	.	r
<i>timer_getoverrun()</i>	o	tmr	o	.	r
<i>timer_gettime()</i>	o	tmr	o	.	r
<i>timer_settime()</i>	o	tmr	o	.	r
<i>times()</i>	m	m	m	m	m	.	.	m	m
<i>timezone</i>	m	xsi	m	m	.	.	.	m	.
<i>tmpfile()</i>	m	m	m	m	m	m	m	m	.
<i>tmpnam()</i>	m	m	m	m	m	m	m	m	.
<i>toascii()</i>	m	xsi	m	m	.	.	.	m	m
<i>tolower()</i>	m	m	m	m	m	m	m	m	m
<i>toupper()</i>	m	m	m	m	m	m	m	m	m
<i>towctrans()</i>	m	m	m	.	.	m	1	.	.
<i>towlower()</i>	m	m	m	m	.	m	1	.	.
<i>towupper()</i>	m	m	m	m	.	m	1	.	.
<i>trunc()</i>	m	m	.	.	.	m	.	.	.
<i>truncate()</i>	m	xsi	m	m	m
<i>truncf()</i>	m	m	.	.	.	m	.	.	.
<i>truncl()</i>	m	m	.	.	.	m	.	.	.
<i>tsearch()</i>	m	xsi	m	m	.	.	.	m	.
<i>ttynam()</i>	m	tsf	m	m	m	.	.	m	m
<i>ttynam_r()</i>	m	tsf	m	.	t
<i>twalk()</i>	m	xsi	m	m	.	.	.	m	.
<i>tzname</i>	m	xsi	m	m	m	.	.	m	.
<i>tzset()</i>	m	xsi	m	m	m	.	.	m	.
<i>ualarm()</i>	ob	xsi	m	m	m
<i>ulimit()</i>	m	xsi	m	m	.	.	.	m	.
<i>umask()</i>	m	m	m	m	m	.	.	m	m
<i>uname()</i>	m	m	m	m	m	.	.	m	.
<i>ungetc()</i>	m	m	m	m	m	m	m	m	m
<i>ungetwc()</i>	m	m	m	m	.	m	1	.	.
<i>unlink()</i>	m	m	m	m	m	.	.	m	m
<i>unlockpt()</i>	m	xsi	m	m	.	.	.	m	.
<i>unsetenv()</i>	m	m
<i>usleep()</i>	ob	xsi	m	m	m
<i>utime()</i>	m	m	m	m	m	.	.	m	m
<i>utimes()</i>	l	xsi	m	m	m
<i>va_arg()</i>	m	m
<i>va_copy()</i>	m	m
<i>va_end()</i>	m	m
<i>va_start()</i>	m	m
<i>vfork()</i>	ob	xsi	m	m	m
<i>vfprintf()</i>	m	m	m	m	.	.	.	m	.
<i>vfscanf()</i>	m	m	.	.	.	m	.	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>vfwprintf()</i>	m	m	m	.	.	m	1	.	.
<i>vfwscanf()</i>	m	m	.	.	.	m	.	.	.
<i>vprintf()</i>	m	m	m	m	.	m	m	m	.
<i>vscanf()</i>	m	m	.	.	.	m	.	.	.
<i>vsnprintf()</i>	m	m	m	.	.	m	.	.	.
<i>vsprintf()</i>	m	m	m	m	.	m	m	m	.
<i>vsscanf()</i>	m	m	.	.	.	m	.	.	.
<i>vswprintf()</i>	m	m	m	.	.	m	1	.	.
<i>vswscanf()</i>	m	m	.	.	.	m	.	.	.
<i>vwprintf()</i>	m	m	m	.	.	m	1	.	.
<i>vwscanf()</i>	m	m	.	.	.	m	.	.	.
<i>wait()</i>	m	m	m	m	m	.	.	m	m
<i>waitid()</i>	m	xsi	m	m	.	.	.	m	.
<i>waitpid()</i>	m	m	m	m	m	.	.	m	m
<i>wcrtomb()</i>	m	m	m	.	.	m	1	.	.
<i>wcscat()</i>	m	m	m	m	.	m	1	.	.
<i>wcschr()</i>	m	m	m	m	.	m	1	.	.
<i>wcscmp()</i>	m	m	m	m	.	m	1	.	.
<i>wcscoll()</i>	m	m	m	o	.	m	1	.	.
<i>wcscopy()</i>	m	m	m	m	.	m	1	.	.
<i>wcscspn()</i>	m	m	m	m	.	m	1	.	.
<i>wcsftime()</i>	m	m	m	o	.	m	1	.	.
<i>wcslen()</i>	m	m	m	m	.	m	1	.	.
<i>wcsncat()</i>	m	m	m	m	.	m	1	.	.
<i>wcsncmp()</i>	m	m	m	m	.	m	1	.	.
<i>wcsncpy()</i>	m	m	m	m	.	m	1	.	.
<i>wcspbrk()</i>	m	m	m	m	.	m	1	.	.
<i>wcsrchr()</i>	m	m	m	m	.	m	1	.	.
<i>wcrtombs()</i>	m	m	m	.	.	m	1	.	.
<i>wcsspn()</i>	m	m	m	m	.	m	1	.	.
<i>wcsstr()</i>	m	m	m	.	.	m	1	.	.
<i>wcstod()</i>	m	m	m	m	.	m	1	.	.
<i>wcstof()</i>	m	m	.	.	.	m	.	.	.
<i>wcstoimax()</i>	m	m	.	.	.	m	.	.	.
<i>wcstok()</i>	m	m	m	m	.	m	1	.	.
<i>wcstol()</i>	m	m	m	m	.	m	1	.	.
<i>wcstold()</i>	m	m	.	.	.	m	.	.	.
<i>wcstoll()</i>	m	m	.	.	.	m	.	.	.
<i>wcstombs()</i>	m	m	m	m	.	m	m	m	.
<i>wcstoul()</i>	m	m	m	m	.	m	1	.	.
<i>wcstoull()</i>	m	m	.	.	.	m	.	.	.
<i>wcstoumax()</i>	m	m	.	.	.	m	.	.	.
<i>wcswcs()</i>	l	xsi	m	m
<i>wcswidth()</i>	m	xsi	m	m
<i>wcsxfrm()</i>	m	m	m	o	.	m	1	.	.
<i>wctob()</i>	m	m	m	.	.	m	1	.	.

System Interface Table

Interface	XSI	POSIX Base	U98	U95	P96 P92	C99	C89	SVID3	BSD
<i>wctomb()</i>	m	m	m	m	.	m	m	m	.
<i>wctrans()</i>	m	m	m	.	.	m	1	.	.
<i>wctype()</i>	m	m	m	m	.	m	1	.	.
<i>wcwidth()</i>	m	xsi	m	m
<i>wmemchr()</i>	m	m	m	.	.	m	1	.	.
<i>wmemcmp()</i>	m	m	m	.	.	m	1	.	.
<i>wmemcpy()</i>	m	m	m	.	.	m	1	.	.
<i>wmemmove()</i>	m	m	m	.	.	m	1	.	.
<i>wmemset()</i>	m	m	m	.	.	m	1	.	.
<i>wordexp()</i>	m	m	m	o	m
<i>wordfree()</i>	m	m	m	o	m
<i>wprintf()</i>	m	m	m	.	.	m	1	.	.
<i>write()</i>	m	xsi	m	m	m	.	.	m	m
<i>writev()</i>	m	xsi	m	m	.	.	.	m	m
<i>wscanf()</i>	m	m	m	.	.	m	1	.	.
<i>y0()</i>	m	xsi	m	m	.	.	.	m	m
<i>y1()</i>	m	xsi	m	m	.	.	.	m	m
<i>yn()</i>	m	xsi	m	m	.	.	.	m	m

Utilities Interface Table

This chapter lists all the utilities described in XCU, Issue 6, and indicates what other specifications support the interfaces.

The following conventions are used in columns 2 through 8:

- m** Indicates that the interface is defined as mandatory.
- opt* In the POSIX Base column, two or three letter option codes are used as described in Portability Codes, denoting the option to which the interface belongs.
- d** Indicates that the interface is part of the DEVELOPMENT Option or Feature Group.
- f** Indicates that the interface is part of the FORTRAN Option or Feature Group.
- 2d** Indicates that the interface is part of IEEE Std 1003.2d-1994 (Batch Environment).
- o** Indicates that the interface is optional.
- .** Indicates that the interface is not specified.

It should be noted that while another specification may support the interface, some of the interface semantics may have changed with evolution and standardization. A developer should not assume that because the interface appears in other specifications, it will behave exactly as described in the Single UNIX Specification, Version 3.

There are 160 utilities listed.

Interface	XSI	POSIX Base	UNIX 98	UNIX 95	POSIX.2-92	SVID 3	4.3BSD
<i>admin</i>	d	xsi	d	d	.	m	.
<i>alias</i>	m	up	m	m	m	.	m
<i>ar</i>	m	sd	m	m	o	m	m
<i>asa</i>	m	FR	m	m	o	.	.
<i>at</i>	m	up	m	m	m	m	m
<i>awk</i>	m	m	m	m	m	m	m
<i>basename</i>	m	m	m	m	m	m	m
<i>batch</i>	m	up	m	m	m	m	.
<i>bc</i>	m	m	m	m	m	.	m
<i>bg</i>	m	up	m	m	m	.	m
<i>c99</i>	m	cd
<i>cal</i>	m	xsi	m	m	.	m	m
<i>cat</i>	m	m	m	m	m	m	m
<i>cd</i>	m	m	m	m	m	m	m
<i>cflow</i>	d	xsi	d	d	.	m	.
<i>chgrp</i>	m	m	m	m	m	m	m
<i>chmod</i>	m	m	m	m	m	m	m
<i>chown</i>	m	m	m	m	m	m	m
<i>cksum</i>	m	m	m	m	m	.	.
<i>cmp</i>	m	m	m	m	m	m	m
<i>comm</i>	m	m	m	m	m	m	m
<i>command</i>	m	up	m	m	m	.	.
<i>compress</i>	m	xsi	m	m	.	m	m
<i>cp</i>	m	m	m	m	m	m	m
<i>crontab</i>	m	up	m	m	m	m	.
<i>csplit</i>	m	up	m	m	m	m	.
<i>ctags</i>	m	up	d	d	m	m	m
<i>cut</i>	m	m	m	m	m	m	.
<i>cxref</i>	d	xsi	d	d	.	m	.
<i>date</i>	m	m	m	m	m	m	m
<i>dd</i>	m	m	m	m	m	m	m
<i>delta</i>	d	xsi	d	d	.	m	.
<i>df</i>	m	up	m	m	m	m	m
<i>diff</i>	m	m	m	m	m	m	m
<i>dirname</i>	m	m	m	m	m	m	.
<i>du</i>	m	up	l	m	m	m	m
<i>echo</i>	m	m	m	m	m	m	m
<i>ed</i>	m	m	m	m	m	m	m
<i>env</i>	m	m	m	m	m	m	.
<i>ex</i>	m	up	m	m	m	m	m
<i>expand</i>	m	up	m	m	m	.	m
<i>expr</i>	m	m	m	m	m	m	m
<i>false</i>	m	m	m	m	m	m	m
<i>fc</i>	m	up	m	m	m	.	m
<i>fg</i>	m	up	m	m	m	.	m
<i>file</i>	m	up	m	m	m	m	m

Utilities Interface Table

Interface	XSI	POSIX Base	UNIX 98	UNIX 95	POSIX.2-92	SVID 3	4.3BSD
<i>find</i>	m	m	m	m	m	m	m
<i>fold</i>	m	m	m	m	m	.	m
<i>fort77</i>	o	fd	o	f	o	.	.
<i>fuser</i>	m	xsi	m
<i>genscat</i>	m	xsi	m	m	.	.	.
<i>get</i>	d	xsi	d	d	.	m	.
<i>getconf</i>	m	m	m	m	.	.	.
<i>getopts</i>	m	m	m	m	.	.	.
<i>grep</i>	m	m	m	m	m	m	m
<i>hash</i>	m	xsi	m	m	.	.	m
<i>head</i>	m	m	m	m	m	m	m
<i>iconv</i>	m	m	m	m	.	.	.
<i>id</i>	m	m	m	m	m	m	.
<i>ipcrm</i>	m	xsi	m
<i>ipcs</i>	m	xsi	m
<i>jobs</i>	m	up	m	m	m	.	m
<i>join</i>	m	m	m	m	m	m	m
<i>kill</i>	m	m	m	m	m	m	m
<i>lex</i>	d	cd	d	d	o	m	m
<i>link</i>	m	xsi	m	.	.	.	m
<i>ln</i>	m	m	m	m	m	m	m
<i>locale</i>	m	m	m	m	m	.	.
<i>localedef</i>	m	m	m	m	m	.	.
<i>logger</i>	m	m	m	m	m	.	m
<i>logname</i>	m	m	m	m	m	m	.
<i>lp</i>	m	m	m	m	m	m	.
<i>ls</i>	m	xsi	m	m	m	m	m
<i>m4</i>	d	xsi	d	d	.	m	m
<i>mailx</i>	m	m	m	m	m	m	.
<i>make</i>	m	sd	d	d	m	m	m
<i>man</i>	m	m	m	m	m	.	m
<i>mesg</i>	m	up	m	m	m	m	m
<i>mkdir</i>	m	m	m	m	m	m	m
<i>mkfifo</i>	m	m	m	m	m	.	.
<i>more</i>	m	up	m	m	m	m	m
<i>mv</i>	m	m	m	m	m	m	m
<i>newgrp</i>	m	up	m	m	m	m	.
<i>nice</i>	m	up	m	m	m	m	m
<i>nl</i>	m	m	m	m	.	m	.
<i>nm</i>	m	up	d	d	m	m	m
<i>nohup</i>	m	m	m	m	m	m	.
<i>od</i>	m	m	m	m	m	m	m
<i>paste</i>	m	m	m	m	m	m	.
<i>patch</i>	m	up	m	m	m	.	m
<i>pathchk</i>	m	m	m	m	m	.	.
<i>pax</i>	m	m	m	m	m	.	.

Interface	XSI	POSIX Base	UNIX 98	UNIX 95	POSIX.2-92	SVID 3	4.3BSD
<i>pr</i>	m	m	m	m	m	m	m
<i>printf</i>	m	m	m	m	m	m	.
<i>prs</i>	d	xsi	d	d	.	m	.
<i>ps</i>	m	up	m	m	m	m	m
<i>pwd</i>	m	m	m	m	m	m	m
<i>qalter</i>	o	be	.	.	2d	.	
<i>qdel</i>	o	be	.	.	2d	.	
<i>qhold</i>	o	be	.	.	2d	.	
<i>qmove</i>	o	be	.	.	2d	.	
<i>qmsg</i>	o	be	.	.	2d	.	
<i>qrerun</i>	o	be	.	.	2d	.	
<i>qrls</i>	o	be	.	.	2d	.	
<i>qselect</i>	o	be	.	.	2d	.	
<i>qsig</i>	o	be	.	.	2d	.	
<i>qstat</i>	o	be	.	.	2d	.	
<i>qsub</i>	o	be	.	.	2d	.	
<i>read</i>	m	m	m	m	m	.	m
<i>renice</i>	m	up	m	m	m	.	m
<i>rm</i>	m	m	m	m	m	m	m
<i>rmdel</i>	d	xsi	d	d	.	m	.
<i>rmdir</i>	m	m	m	m	m	m	m
<i>sact</i>	d	xsi	d	d	.	m	.
<i>sccs</i>	d	xsi	d	d	.		m
<i>sed</i>	m	m	m	m	m	m	m
<i>sh</i>	m	m	m	m	m	m	m
<i>sleep</i>	m	m	m	m	m	m	m
<i>sort</i>	m	m	m	m	m	m	m
<i>split</i>	m	up	m	m	m	m	m
<i>strings</i>	m	up	m	m	m	.	m
<i>strip</i>	m	sd	d	d	o	m	m
<i>stty</i>	m	m	m	m	m	m	m
<i>tabs</i>	m	up	m	m	m	m	m
<i>tail</i>	m	m	m	m	m	m	m
<i>talk</i>	m	up	m	m	m	.	m
<i>tee</i>	m	m	m	m	.	m	m
<i>test</i>	m	m	m	m	m	m	m
<i>time</i>	m	up	m	m	m	m	m
<i>touch</i>	m	m	m	m	m	m	m
<i>tput</i>	m	up	m	m	m	.	m
<i>tr</i>	m	m	m	m	m	m	m
<i>true</i>	m	m	m	m	m	m	m
<i>tsort</i>	m	xsi	m	m	.	m	m
<i>tty</i>	m	m	m	m	m	m	m
<i>type</i>	m	xsi	m	m	.	.	m
<i>ulimit</i>	m	xsi	m	m	.	.	m
<i>umask</i>	m	m	m	m	m	m	.

Utilities Interface Table

Interface	XSI	POSIX Base	UNIX 98	UNIX 95	POSIX.2-92	SVID 3	4.3BSD
<i>unalias</i>	m	up	m	m	m	.	m
<i>uname</i>	m	m	m	m	m	m	.
<i>uncompress</i>	m	xsi	m	m	.	m	.
<i>unexpand</i>	m	up	m	m	m	.	.
<i>unget</i>	d	xsi	d	d	.	m	.
<i>uniq</i>	m	m	m	m	m	m	m
<i>unlink</i>	m	xsi	m	.	.	.	m
<i>uucp</i>	m	xsi	m	m	.	m	m
<i>uudecode</i>	m	up	m	m	m	m	m
<i>uuencode</i>	m	up	m	m	m	m	m
<i>uustat</i>	m	xsi	m	m	.	m	.
<i>uux</i>	m	xsi	m	m	.	m	m
<i>val</i>	d	xsi	d	d	.	m	.
<i>vi</i>	m	up	m	m	m	m	m
<i>wait</i>	m	m	m	m	m	m	m
<i>wc</i>	m	m	m	m	m	m	m
<i>what</i>	d	xsi	d	d	.	m	m
<i>who</i>	m	up	m	m	m	m	m
<i>write</i>	m	up	m	m	m	m	m
<i>xargs</i>	m	m	m	m	m	m	.
<i>yacc</i>	d	cd	d	d	o	m	m
<i>zcat</i>	m	xsi	m	m	.	m	.

Headers Interface Table

This chapter lists all the headers described in XBD, Issue 6, complete with an indication of their status for the XSI extension, the POSIX Base, and their availability in UNIX 98, UNIX 95, IEEE Std 1003.1-1996 (POSIX.1) (denoted P96), IEEE Std 1003.2-1992 (POSIX.2) (denoted P92), the ISO C standard (denoted C99), and C 89. The following conventions are used in columns 2 through 8:

- m** Indicates that the interface is defined as mandatory.
- o** Indicates that the interface is part of an Option or Feature Group.
- opt* In the POSIX Base column, two or three letter option codes are used as described in Portability Codes, denoting the option to which the interface belongs.
- r** In the P96 column, this indicates that the interface is part of the POSIX Realtime Extension.
- t** In the P96 column, this indicates that the interface is part of the POSIX Threads Extension.
- 1** In the C89 column, this indicates that the interface is part of the ISO/IEC 9899: 1990 standard.
- .** Indicates that the interface is not specified.

Interface	XSI	POSIX Base	UNIX 98	UNIX 95	P96 P92	C99	C89
<aio.h>	o	aio	o	.	r	.	.
<arpa/inet.h>	m	m	m	m	.	.	.
<assert.h>	m	m	m	m	.	m	m
<complex.h>	m	m	.	.	.	m	.
<cpio.h>	m	xsi	m	m	.	.	.
<ctype.h>	m	m	m	m	.	m	m
<dirent.h>	m	m	m	m	m	.	.
<dlfcn.h>	m	xsi	m
<errno.h>	m	m	m	m	m	m	m
<fcntl.h>	m	m	m	m	m	.	.
<fenv.h>	m	m	.	.	.	m	.
<float.h>	m	m	m	m	m	m	m
<fmtmsg.h>	m	xsi	m	m	m	.	.
<fnmatch.h>	m	m	m	m	m	.	.
<ftw.h>	m	xsi	m	m	.	.	.
<glob.h>	m	m	m	m	m	.	.
<grp.h>	m	m	m	m	m	.	.
<iconv.h>	m	xsi	m	m	.	.	.
<inttypes.h>	m	m	m	.	.	m	.
<iso646.h>	m	m	m	.	.	m	1
<langinfo.h>	m	xsi	m	m	.	.	.
<libgen.h>	m	xsi	m	m	.	.	.
<limits.h>	m	m	m	m	m	m	m
<locale.h>	m	m	m	m	m	m	m
<math.h>	m	m	m	m	m	m	m
<monetary.h>	m	xsi	m	m	.	.	.
<mqueue.h>	o	msg	o	.	r	.	.
<ndbm.h>	m	xsi	m	m	.	.	.
<net/if.h>	m	m	m	m	.	.	.
<netdb.h>	m	m	m	m	m	.	.
<netinet/in.h>	m	m	m	m	m	.	.
<netinet/tcp.h>	m	m	m	m	m	.	.
<nl_types.h>	m	xsi	m	m	.	.	.
<poll.h>	m	m	m	m	.	.	.
<pthread.h>	m	thr	m	.	t	.	.
<pwd.h>	m	m	m	m	m	.	.
<regex.h>	m	m	m	m	m	.	.
<sched.h>	o	ps	o	.	r	.	.
<search.h>	m	xsi	m	m	.	.	.
<semaphore.h>	o	sem	o	.	r	.	.
<setjmp.h>	m	m	m	m	m	m	m
<signal.h>	m	m	m	m	m	m	m
<spawn.h>	o	spn
<stdarg.h>	m	m	m	m	m	m	m
<stdbool.h>	m	m	m	m	m	m	.
<stddef.h>	m	m	m	m	m	m	m

Headers Interface Table

Interface	XSI	POSIX Base	UNIX 98	UNIX 95	P96 P92	C99	C89
<stdint.h>	m	m	.	.	.	m	.
<stdio.h>	m	m	m	m	m	m	m
<stdlib.h>	m	m	m	m	m	m	m
<string.h>	m	m	m	m	m	m	m
<strings.h>	m	xsi	m	m	.	.	.
<stropts.h>	m	xsr	m	m	.	.	.
<sys/ipc.h>	m	xsi	m	m	.	.	.
<sys/mman.h>	m	m	m	m	m	.	.
<sys/msg.h>	m	xsi	m	m	.	.	.
<sys/resource.h>	m	xsi	m	m	.	.	.
<sys/select.h>	m	m	m	m	.	.	.
<sys/sem.h>	m	xsi	m	m	m	.	.
<sys/shm.h>	m	xsi	m	m	.	.	.
<sys/socket.h>	m	m	m	m	.	.	.
<sys/stat.h>	m	m	m	m	m	.	.
<sys/statvfs.h>	m	xsi	m	m	m	.	.
<sys/time.h>	m	m	xsi	m	.	.	.
<sys/timeb.h>	m	xsi	m	m	.	.	.
<sys/times.h>	m	m	m	m	m	.	.
<sys/types.h>	m	m	m	m	m	.	.
<sys/uio.h>	m	xsi	m	m	.	.	.
<sys/un.h>	m	xsi	m	m	.	.	.
<sys/utsname.h>	m	m	m	m	m	.	.
<sys/wait.h>	m	m	m	m	m	.	.
<syslog.h>	m	xsi	m	m	.	.	.
<tar.h>	m	m	m	m	m	.	.
<termios.h>	m	m	m	m	m	.	.
<tgmath.h>	m	m	.	.	.	m	.
<time.h>	m	m	m	m	.	m	m
<trace.h>	o	trc
<ucontext.h>	m	xsi	m	m	.	.	.
<ulimit.h>	m	xsi	m	m	.	.	.
<unistd.h>	m	m	m	m	m	.	.
<utime.h>	m	m	m	m	m	.	.
<utmpx.h>	m	xsi	m	m	.	.	.
<wchar.h>	m	m	m	m	.	m	1
<wctype.h>	m	m	m	m	.	m	1
<wordexp.h>	m	m	m	m	m	.	.

XCURSES Interface Table

This chapter contains a table of all the interfaces defined in X/Open Curses, Issue 4, Version 2, complete with an indication of their availability for the Single UNIX Specification (denoted by SUSv3), and their availability in UNIX 98, UNIX 95, POSIX.1, POSIX.2, the ISO C standard, SVID 3, and 4.3BSD. The XCURSES interfaces are not specified for the POSIX base in IEEE Std 1003.1-2001.

The following conventions are used in columns 2 through 8:

m Indicates that the interface is defined as mandatory.

. Indicates that the interface is not specified.

The table is intended as a quick reference guide for programmers migrating to or developing applications for the Single UNIX Specification, Version 3.

There are 372 interfaces listed, and in addition XCURSES defines three headers: **<curses.h>**, **<term.h>**, and **<unctrl.h>**.

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>add_wch()</i>	m	m	m
<i>add_wchnstr()</i>	m	m	m
<i>add_wchstr()</i>	m	m	m
<i>addch()</i>	m	m	m	.	.	m	.
<i>addchnstr()</i>	m	m	m	.	.	m	.
<i>addchstr()</i>	m	m	m	.	.	m	.
<i>addnstr()</i>	m	m	m	.	.	m	.
<i>addnwstr()</i>	m	m	m
<i>addstr()</i>	m	m	m	.	.	m	.
<i>addwstr()</i>	m	m	m
<i>attr_get()</i>	m	m	m
<i>attr_off()</i>	m	m	m
<i>attr_on()</i>	m	m	m
<i>attr_set()</i>	m	m	m
<i>attroff()</i>	m	m	m	.	.	m	.
<i>attron()</i>	m	m	m	.	.	m	.
<i>attrset()</i>	m	m	m	.	.	m	.
<i>baudrate()</i>	m	m	m	.	.	m	.
<i>beep()</i>	m	m	m	.	.	m	.
<i>bkgd()</i>	m	m	m	.	.	m	.
<i>bkgdset()</i>	m	m	m	.	.	m	.

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>bkgrnd()</i>	m	m	m
<i>bkgrndset()</i>	m	m	m
<i>border()</i>	m	m	m	.	.	m	.
<i>border_set()</i>	m	m	m
<i>box()</i>	m	m	m	.	.	m	.
<i>box_set()</i>	m	m	m
<i>can_change_color()</i>	m	m	m	.	.	m	.
<i>chgat()</i>	m	m	m
<i>clear()</i>	m	m	m	.	.	m	.
<i>clearerr()</i>	m	m	m	m	m	m	m
<i>clearok()</i>	m	m	m	.	.	m	.
<i>clrtoebot()</i>	m	m	m	.	.	m	.
<i>clrtoeol()</i>	m	m	m	.	.	m	.
<i>color_content()</i>	m	m	m
<i>color_set()</i>	m	m	m
<i>copywin()</i>	m	m	m	.	.	m	.
<i>curs_set()</i>	m	m	m	.	.	m	.
<i>def_prog_mode()</i>	m	m	m	.	.	m	.
<i>def_shell_mode()</i>	m	m	m	.	.	m	.
<i>del_curterm()</i>	m	m	m	.	.	m	.
<i>delay_output()</i>	m	m	m	.	.	m	.
<i>delch()</i>	m	m	m	.	.	m	.
<i>deleteln()</i>	m	m	m	.	.	m	.
<i>delscreen()</i>	m	m	m	.	.	m	.
<i>delwin()</i>	m	m	m	.	.	m	.
<i>derwin()</i>	m	m	m	.	.	m	.
<i>dupwin()</i>	m	m	m	.	.	m	.
<i>echo()</i>	m	m	m	.	.	m	.
<i>echo_wchar()</i>	m	m	m
<i>echochar()</i>	m	m	m	.	.	m	.
<i>endwin()</i>	m	m	m	.	.	m	.
<i>erase()</i>	m	m	m	.	.	m	.
<i>erasechar()</i>	m	m	m	.	.	m	.
<i>erasewchar()</i>	m	m	m
<i>filter()</i>	m	m	m	.	.	m	.
<i>flash()</i>	m	m	m	.	.	m	.
<i>flushinp()</i>	m	m	m	.	.	m	.
<i>get_wch()</i>	m	m	m
<i>get_wstr()</i>	m	m	m
<i>getbegyx()</i>	m	m	m	.	.	m	.
<i>getbkgd()</i>	m	m	m
<i>getbkgrnd()</i>	m	m	m
<i>getmaxyx()</i>	m	m	m	.	.	m	.
<i>getn_wstr()</i>	m	m	m
<i>getparyx()</i>	m	m	m	.	.	m	.
<i>getstr()</i>	m	m	m	.	.	m	.

XCURSES Interface Table

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>getwin()</i>	m	m	m
<i>getyx()</i>	m	m	m	.	.	m	.
<i>halfdelay()</i>	m	m	m	.	.	m	.
<i>has_colors()</i>	m	m	m	.	.	m	.
<i>has_ic()</i>	m	m	m	.	.	m	.
<i>has_il()</i>	m	m	m	.	.	m	.
<i>hline()</i>	m	m	m
<i>hline_set()</i>	m	m	m
<i>idcok()</i>	m	m	m	.	.	m	.
<i>idlok()</i>	m	m	m	.	.	m	.
<i>ilogb()</i>	m	m	m
<i>immedok()</i>	m	m	m	.	.	m	.
<i>in_wch()</i>	m	m	m
<i>in_wchnstr()</i>	m	m	m
<i>in_wchstr()</i>	m	m	m
<i>inch()</i>	m	m	m	.	.	m	.
<i>inchnstr()</i>	m	m	m	.	.	m	.
<i>inchstr()</i>	m	m	m	.	.	m	.
<i>init_color()</i>	m	m	m	.	.	m	.
<i>init_pair()</i>	m	m	m	.	.	m	.
<i>initscr()</i>	m	m	m	.	.	m	.
<i>innstr()</i>	m	m	m	.	.	m	.
<i>innwstr()</i>	m	m	m
<i>ins_nwstr()</i>	m	m	m
<i>ins_wch()</i>	m	m	m
<i>ins_wstr()</i>	m	m	m
<i>insch()</i>	m	m	m	.	.	m	.
<i>insdelln()</i>	m	m	m	.	.	m	.
<i>insertln()</i>	m	m	m	.	.	m	.
<i>insnstr()</i>	m	m	m	.	.	m	.
<i>insstr()</i>	m	m	m	.	.	m	.
<i>instr()</i>	m	m	m	.	.	m	.
<i>intrflush()</i>	m	m	m	.	.	m	.
<i>inwstr()</i>	m	m	m
<i>is_linetouched()</i>	m	m	m	.	.	m	.
<i>is_wintouched()</i>	m	m	m	.	.	m	.
<i>isendwin()</i>	m	m	m	.	.	m	.
<i>key_name()</i>	m	m	m
<i>keyname()</i>	m	m	m	.	.	m	.
<i>keypad()</i>	m	m	m	.	.	m	.
<i>killchar()</i>	m	m	m
<i>killwchar()</i>	m	m	m
<i>leaveok()</i>	m	m	m	.	.	m	.
<i>longname()</i>	m	m	m	.	.	m	.
<i>meta()</i>	m	m	m	.	.	m	.
<i>move()</i>	m	m	m	.	.	m	.

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>mvadd_wch()</i>	m	m	m
<i>mvadd_wchnstr()</i>	m	m	m
<i>mvadd_wchstr()</i>	m	m	m
<i>mvaddch()</i>	m	m	m	.	.	m	.
<i>mvaddchnstr()</i>	m	m	m	.	.	m	.
<i>mvaddchstr()</i>	m	m	m	.	.	m	.
<i>mvaddnstr()</i>	m	m	m	.	.	m	.
<i>mvaddnwstr()</i>	m	m	m
<i>mvaddstr()</i>	m	m	m	.	.	m	.
<i>mvaddwstr()</i>	m	m	m
<i>mvchgat()</i>	m	m	m
<i>mvcur()</i>	m	m	m	.	.	m	.
<i>mvdelch()</i>	m	m	m	.	.	m	.
<i>mvderwin()</i>	m	m	m	.	.	m	.
<i>mvget_wch()</i>	m	m	m
<i>mvget_wstr()</i>	m	m	m
<i>mvgetch()</i>	m	m	m	.	.	m	.
<i>mvgetn_wstr()</i>	m	m	m
<i>mvgetnstr()</i>	m	m	m	.	.	m	.
<i>mvgetstr()</i>	m	m	m
<i>mvhline()</i>	m	m	m
<i>mvhline_set()</i>	m	m	m
<i>mvin_wch()</i>	m	m	m
<i>mvin_wchnstr()</i>	m	m	m
<i>mvin_wchstr()</i>	m	m	m
<i>mvinch()</i>	m	m	m	.	.	m	.
<i>mvinchnstr()</i>	m	m	m	.	.	m	.
<i>mvinchstr()</i>	m	m	m	.	.	m	.
<i>mvinnstr()</i>	m	m	m	.	.	m	.
<i>mvinnwstr()</i>	m	m	m
<i>mvins_nwstr()</i>	m	m	m
<i>mvins_wch()</i>	m	m	m
<i>mvins_wstr()</i>	m	m	m
<i>mvinsch()</i>	m	m	m	.	.	m	.
<i>mvinsnstr()</i>	m	m	m	.	.	m	.
<i>mvinsstr()</i>	m	m	m	.	.	m	.
<i>mvinstr()</i>	m	m	m	.	.	m	.
<i>mvinwstr()</i>	m	m	m
<i>mvprintw()</i>	m	m	m	.	.	m	.
<i>mvscanw()</i>	m	m	m	.	.	m	.
<i>mvvline()</i>	m	m	m
<i>mvvline_set()</i>	m	m	m
<i>mvwadd_wch()</i>	m	m	m
<i>mvwadd_wchnstr()</i>	m	m	m
<i>mvwadd_wchstr()</i>	m	m	m
<i>mvwaddch()</i>	m	m	m	.	.	m	.

XCURSES Interface Table

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>mvwaddchnstr()</i>	m	m	m	.	.	m	.
<i>mvwaddchstr()</i>	m	m	m	.	.	m	.
<i>mvwaddnstr()</i>	m	m	m	.	.	m	.
<i>mvwaddnwstr()</i>	m	m	m
<i>mvwaddstr()</i>	m	m	m	.	.	m	.
<i>mvwaddwstr()</i>	m	m	m
<i>mvwchgat()</i>	m	m	m
<i>mvwdelch()</i>	m	m	m	.	.	m	.
<i>mvwget_wch()</i>	m	m	m
<i>mvwget_wstr()</i>	m	m	m
<i>mvwgetch()</i>	m	m	m	.	.	m	.
<i>mvwgetn_wstr()</i>	m	m	m
<i>mvwgetnstr()</i>	m	m	m
<i>mvwgetstr()</i>	m	m	m	.	.	m	.
<i>mvwhline()</i>	m	m	m
<i>mvwhline_set()</i>	m	m	m
<i>mvwin()</i>	m	m	m	.	.	m	.
<i>mvwin_wch()</i>	m	m	m
<i>mvwin_wchnstr()</i>	m	m	m
<i>mvwin_wchstr()</i>	m	m	m
<i>mvwinch()</i>	m	m	m	.	.	m	.
<i>mvwinchnstr()</i>	m	m	m	.	.	m	.
<i>mvwinchstr()</i>	m	m	m	.	.	m	.
<i>mvwinnstr()</i>	m	m	m	.	.	m	.
<i>mvwinnwstr()</i>	m	m	m
<i>mvwins_nwstr()</i>	m	m	m
<i>mvwins_wch()</i>	m	m	m
<i>mvwins_wstr()</i>	m	m	m
<i>mvwinsch()</i>	m	m	m	.	.	m	.
<i>mvwinsnstr()</i>	m	m	m	.	.	m	.
<i>mvwinsstr()</i>	m	m	m	.	.	m	.
<i>mvwinstr()</i>	m	m	m	.	.	m	.
<i>mvwinwstr()</i>	m	m	m
<i>mvwprintw()</i>	m	m	m	.	.	m	.
<i>mvwscanw()</i>	m	m	m	.	.	m	.
<i>mvwvline()</i>	m	m	m
<i>mvwvline_set()</i>	m	m	m
<i>napms()</i>	m	m	m	.	.	m	.
<i>newpad()</i>	m	m	m	.	.	m	.
<i>newterm()</i>	m	m	m	.	.	m	.
<i>newwin()</i>	m	m	m	.	.	m	.
<i>nl()</i>	m	m	m	.	.	m	.
<i>nocbreak()</i>	m	m	m	.	.	m	.
<i>nodelay()</i>	m	m	m	.	.	m	.
<i>noecho()</i>	m	m	m	.	.	m	.
<i>nonl()</i>	m	m	m	.	.	m	.

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>noqiflush()</i>	m	m	m	.	.	m	.
<i>noraw()</i>	m	m	m	.	.	m	.
<i>notimeout()</i>	m	m	m	.	.	m	.
<i>overlay()</i>	m	m	m	.	.	m	.
<i>overwrite()</i>	m	m	m	.	.	m	.
<i>pair_content()</i>	m	m	m	.	.	m	.
<i>pecho_wchar()</i>	m	m	m
<i>pechochar()</i>	m	m	m	.	.	m	.
<i>pnoutrefresh()</i>	m	m	m	.	.	m	.
<i>prefresh()</i>	m	m	m	.	.	m	.
<i>printw()</i>	m	m	m	.	.	m	.
<i>putp()</i>	m	m	m	.	.	m	.
<i>putwin()</i>	m	m	m	.	.	m	.
<i>qiflush()</i>	m	m	m	.	.	m	.
<i>raw()</i>	m	m	m	.	.	m	.
<i>redrawwin()</i>	m	m	m	.	.	m	.
<i>refresh()</i>	m	m	m	.	.	m	.
<i>reset_prog_mode()</i>	m	m	m	.	.	m	.
<i>reset_shell_mode()</i>	m	m	m	.	.	m	.
<i>resetty()</i>	m	m	m	.	.	m	.
<i>restartterm()</i>	m	m	m	.	.	m	.
<i>ripoffline()</i>	m	m	m	.	.	m	.
<i>savetty()</i>	m	m	m	.	.	m	.
<i>scr_dump()</i>	m	m	m	.	.	m	.
<i>scr_init()</i>	m	m	m	.	.	m	.
<i>scr_restore()</i>	m	m	m	.	.	m	.
<i>scr_set()</i>	m	m	m	.	.	m	.
<i>scr1()</i>	m	m	m	.	.	m	.
<i>scroll()</i>	m	m	m	.	.	m	.
<i>scrollok()</i>	m	m	m	.	.	m	.
<i>set_curterm()</i>	m	m	m	.	.	m	.
<i>set_term()</i>	m	m	m	.	.	m	.
<i>setcchar()</i>	m	m	m
<i>setscrreg()</i>	m	m	m	.	.	m	.
<i>setupterm()</i>	m	m	m	.	.	m	.
<i>slk_attr_off()</i>	m	m	m
<i>slk_attr_on()</i>	m	m	m
<i>slk_attr_set()</i>	m	m	m
<i>slk_attroff()</i>	m	m	m	.	.	m	.
<i>slk_attron()</i>	m	m	m	.	.	m	.
<i>slk_attrset()</i>	m	m	m	.	.	m	.
<i>slk_clear()</i>	m	m	m	.	.	m	.
<i>slk_color()</i>	m	m	m
<i>slk_init()</i>	m	m	m	.	.	m	.
<i>slk_label()</i>	m	m	m	.	.	m	.
<i>slk_noutrefresh()</i>	m	m	m	.	.	m	.

XCURSES Interface Table

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>slk_refresh()</i>	m	m	m	.	.	m	.
<i>slk_restore()</i>	m	m	m	.	.	m	.
<i>slk_set()</i>	m	m	m	.	.	m	.
<i>slk_touch()</i>	m	m	m	.	.	m	.
<i>slk_wset()</i>	m	m	m
<i>standend()</i>	m	m	m	.	.	m	.
<i>standout()</i>	m	m	m	.	.	m	.
<i>start_color()</i>	m	m	m	.	.	m	.
<i>stdscr()</i>	m	m	m	.	.	m	.
<i>subpad()</i>	m	m	m	.	.	m	.
<i>subwin()</i>	m	m	m	.	.	m	.
<i>syncok()</i>	m	m	m	.	.	m	.
<i>term_attrs()</i>	m	m	m
<i>termattrs()</i>	m	m	m	.	.	m	.
<i>termname()</i>	m	m	m	.	.	m	.
<i>tgetent()</i>	m	m	m	.	.	m	.
<i>tgetflag()</i>	m	m	m	.	.	m	.
<i>tgetnum()</i>	m	m	m	.	.	m	.
<i>tgetstr()</i>	m	m	m	.	.	m	.
<i>tgoto()</i>	m	m	m	.	.	m	.
<i>tigetflag()</i>	m	m	m	.	.	m	.
<i>tigetnum()</i>	m	m	m	.	.	m	.
<i>tigetstr()</i>	m	m	m	.	.	m	.
<i>timeout()</i>	m	m	m	.	.	m	.
<i>touchline()</i>	m	m	m	.	.	m	.
<i>touchwin()</i>	m	m	m	.	.	m	.
<i>tparm()</i>	m	m	m	.	.	m	.
<i>tputs()</i>	m	m	m	.	.	m	.
<i>typeahead()</i>	m	m	m	.	.	m	.
<i>unctrl()</i>	m	m	m	.	.	m	.
<i>unget_wch()</i>	m	m	m
<i>ungetch()</i>	m	m	m	.	.	m	.
<i>untouchwin()</i>	m	m	m	.	.	m	.
<i>use_env()</i>	m	m	m	.	.	m	.
<i>vid_attr()</i>	m	m	m
<i>vid_puts()</i>	m	m	m
<i>vidattr()</i>	m	m	m	.	.	m	.
<i>vidputs()</i>	m	m	m	.	.	m	.
<i>vline()</i>	m	m	m
<i>vline_set()</i>	m	m	m
<i>vw_printw()</i>	m	m	m
<i>vw_scanw()</i>	m	m	m
<i>vwprintw()</i>	m	m	m	.	.	m	.
<i>wscanw()</i>	m	m	m	.	.	m	.
<i>wadd_wch()</i>	m	m	m
<i>wadd_wchnstr()</i>	m	m	m

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>wadd_wchstr()</i>	m	m	m
<i>waddch()</i>	m	m	m	.	.	m	.
<i>waddchnstr()</i>	m	m	m	.	.	m	.
<i>waddchstr()</i>	m	m	m	.	.	m	.
<i>waddnstr()</i>	m	m	m	.	.	m	.
<i>waddnwstr()</i>	m	m	m
<i>waddstr()</i>	m	m	m	.	.	m	.
<i>waddwstr()</i>	m	m	m
<i>wattr_get()</i>	m	m	m
<i>wattr_off()</i>	m	m	m
<i>wattr_on()</i>	m	m	m
<i>wattr_set()</i>	m	m	m
<i>wattroff()</i>	m	m	m	.	.	m	.
<i>wattron()</i>	m	m	m	.	.	m	.
<i>wattrset()</i>	m	m	m	.	.	m	.
<i>wbkgd()</i>	m	m	m	.	.	m	.
<i>wbkgdset()</i>	m	m	m	.	.	m	.
<i>wbkgrnd()</i>	m	m	m
<i>wbkgrndset()</i>	m	m	m
<i>wborder()</i>	m	m	m	.	.	m	.
<i>wborder_set()</i>	m	m	m
<i>wchgat()</i>	m	m	m
<i>wclear()</i>	m	m	m	.	.	m	.
<i>wclrtoolt()</i>	m	m	m	.	.	m	.
<i>wclrtoeol()</i>	m	m	m	.	.	m	.
<i>wcolor_set()</i>	m	m	m
<i>wcursyncup()</i>	m	m	m	.	.	m	.
<i>wdelch()</i>	m	m	m	.	.	m	.
<i>wdeleteln()</i>	m	m	m	.	.	m	.
<i>wecho_wchar()</i>	m	m	m
<i>wechochar()</i>	m	m	m	.	.	m	.
<i>werase()</i>	m	m	m	.	.	m	.
<i>wget_wch()</i>	m	m	m
<i>wget_wstr()</i>	m	m	m
<i>wgetbkgrnd()</i>	m	m	m
<i>wgetch()</i>	m	m	m	.	.	m	.
<i>wgetn_wstr()</i>	m	m	m
<i>wgetnstr()</i>	m	m	m	.	.	m	.
<i>wgetstr()</i>	m	m	m	.	.	m	.
<i>whline()</i>	m	m	m	.	.	m	.
<i>whline_set()</i>	m	m	m
<i>win_wch()</i>	m	m	m
<i>win_wchnstr()</i>	m	m	m
<i>win_wchstr()</i>	m	m	m
<i>winch()</i>	m	m	m	.	.	m	.
<i>winchnstr()</i>	m	m	m	.	.	m	.

XCURSES Interface Table

Interface	SUSv3	UNIX 98	UNIX 95	POSIX.1/ POSIX.2	ISO C Standard	SVID 3	4.3BSD
<i>winchstr()</i>	m	m	m	.	.	m	.
<i>winnstr()</i>	m	m	m	.	.	m	.
<i>winnwstr()</i>	m	m	m
<i>wins_nwstr()</i>	m	m	m
<i>wins_wch()</i>	m	m	m
<i>wins_wstr()</i>	m	m	m
<i>winsch()</i>	m	m	m	.	.	m	.
<i>winsdelln()</i>	m	m	m	.	.	m	.
<i>winsertln()</i>	m	m	m	.	.	m	.
<i>winsnstr()</i>	m	m	m	.	.	m	.
<i>winsstr()</i>	m	m	m	.	.	m	.
<i>winstr()</i>	m	m	m	.	.	m	.
<i>winwstr()</i>	m	m	m
<i>wmove()</i>	m	m	m	.	.	m	.
<i>wnoutrefresh()</i>	m	m	m	.	.	m	.
<i>wprintw()</i>	m	m	m	.	.	m	.
<i>wredrawln()</i>	m	m	m	.	.	m	.
<i>wrefresh()</i>	m	m	m	.	.	m	.
<i>wscr1()</i>	m	m	m	.	.	m	.
<i>wsetscrreg()</i>	m	m	m	.	.	m	.
<i>wstandend()</i>	m	m	m	.	.	m	.
<i>wstandout()</i>	m	m	m	.	.	m	.
<i>wsyncdown()</i>	m	m	m	.	.	m	.
<i>wsyncup()</i>	m	m	m	.	.	m	.
<i>wtimeout()</i>	m	m	m	.	.	m	.
<i>wtouchln()</i>	m	m	m	.	.	m	.
<i>wunctrl()</i>	m	m	m
<i>wvline()</i>	m	m	m	.	.	m	.
<i>wvline_set()</i>	m	m	m

