NAME

NAME

POSIX - Perl interface to IEEE Std 1003.1

SYNOPSIS

use POSIX; use POSIX qw(setsid);use POSIX qw(:errno_h :fcntl_h);printf "EINTR is %d\n", EINTR;$sess_id = POSIX::setsid();$fd = POSIX::open($path, O_CREAT|O_EXCL|O_WRONLY, 0644);# note: that's a filedescriptor, *NOT* a filehandle

DEscriptION

The POSIX module permits you to access all (or nearly all) the standardPOSIX 1003.1 identifiers. Many of these identifiers have been given Perl-ishinterfaces. Things which are#defines in C, like EINTR or O_NDELAY, areautomatically exported into your namespace. All functions are only exportedif you ask for them explicitly. Most likely people will prefer to use the fully-qualified function names. This document gives a condensed list of the features available in the POSIXmodule. Consult your operating system's man pages for general information on most features. Consult the perlfunc man page for functions which are noted as being identical to Perl's builtin functions. The first section describes POSIX functions from the 1003.1 specification. The second section describes some classes for signal objects, TTY objects, and other miscellaneous objects. The remaining sections list variousconstants and macros in an organization which roughly follows IEEE Std1003.1b-1993.

NOTE

The POSIX module is probably the most complex Perl module supplied with the standard distribution. It incorporates auto loading, namespace games, and dynamic loading of code that's in Perl, C, or both. It's a great source of wisdom.

CAVEATS

A few functions are not implemented because they are C specific. If you at tempt to call these, they will print a message telling you that they aren't implemented, and suggest using the Perl equivalent should one exist. For example, trying to access the set jmp()call will elicit the message ``set jmp()is C-specific: use eval {} instead''. Furthermore, some evil vendors will claim 1003.1 compliance, but in fact are not so: they will not pass the PCTS

(POSIX Compliance Test Suites).For example, one vendor may not define EDEADLK, or the semantics of the errno values set by open(2) might not be quite right. Perl does not at tempt to verify POSIX compliance. That means you can currently successfully say ``use POSIX'', and then later in your program you find that your vendor has been lax and there's no usable ICANON macro after all. This could be construed to be a bug.

FUNCTIONS

_exit

This is identical to the C function
_exit().

abort

This is identical to the C function
abort().

abs

This is identical to Perl's builtin
abs()
function.

access

Determines the accessibility of a file.

```
    if( POSIX:: access( "/", &POSIX::R_OK ) ){print "have read
permission\n";}
```

Returnsundef on failure.

acos

This is identical to the C function
acos().

alarm

This is identical to Perl's builtin
alarm()
function.

asctime

This is identical to the C function
asctime().

asin

This is identical to the C function
asin().

assert

Unimplemented.
atan
This is identical to the C function
atan()
.
atan2
This is identical to Perl's builtin
atan2()
function.
At exit
at exit()is C-specific: use END {} instead.

atof

atof()is C-specific.

atoi

atoi()is C-specific.

atol

atol()is C-specific.

bsearch

bsearch()not supplied.

calloc

calloc()is C-specific.

ceil

This is identical to the C function
ceil().

chdir

This is identical to Perl's builtin
chdir()
function.

chmod

This is identical to Perl's builtin
chmod()
function.

chown

This is identical to Perl's builtin
chown()
function.

clearerr

Use method FileHandle:: clearerr()
instead.

clock

This is identical to the C function
clock().

close

Close the file. This uses file descriptors such as those obtained by callingPOSIX:: open.
$fd = POSIX::open("foo", &POSIX::O_RDONLY);POSIX::close($fd);
Returnsundef on failure.

closedir

This is identical to Perl's builtin

closedir()
function.

cos

This is identical to Perl's builtin
cos()
function.
cosh
This is identical to the C function
cosh().
creat
Create a new file. This returns a file descriptor like the ones returned byPOSIX:: open. UsePOSIX:: close to close the file.
$fd = POSIX::creat("foo", 0611);POSIX::close($fd);
ctermid
Generates the path name for the controlling terminal.
$path = POSIX::ctermid();

ctime

This is identical to the C function
ctime().

cuserid

Get the character login name of the user.
$name = POSIX::cuserid();

difftime

This is identical to the C function
difftime().

div

div()is C-specific.

dup

This is similar to the C function
dup().This uses file descriptors such as those obtained by calling POSIX:: open. Returnsundef on failure.

dup2

This is similar to the C function
dup2()
.This uses file descriptors such as those obtained by callingPOSIX:: open. Returnsun def on failure.

errno

Returns the value of errno.
$errno = POSIX:: errno();

execl

execl()is C-specific.

execle

execle()is C-specific.

execlp

execlp()is C-specific.

execv

execv()is C-specific.

execve

execve()is C-specific.

execvp

execvp()is C-specific.

exit

This is identical to Perl's builtin
exit()
function.

exp

This is identical to Perl's builtin
exp()
function.

fabs

This is identical to Perl's builtin
abs()
function.

fclose

Use method FileHandle::close()
instead.

fcntl

This is identical to Perl's builtin
fcntl()
function.

fdopen

Use method FileHandle::new_from_fd()
instead.

feof

Use method FileHandle::eof()
instead.

ferror

Use method FileHandle::error()
instead.

fflush

Use method FileHandle::flush()
instead.

fgetc

Use method FileHandle::getc()

instead.

fgetpos

Use method FileHandle::getpos()
instead.

fgets

Use method FileHandle::gets()
instead.

fileno

Use method FileHandle::fileno()
instead.

floor

This is identical to the C function
floor().

fmod

This is identical to the C function
fmod().

fopen

Use method FileHandle::open()
instead.

fork

This is identical to Perl's builtin
fork()
function.

fpathconf

Retrieves the value of a configurable limit on a file or directory. Thisuses file descriptors such as those obtained by callingPOSIX::open.The following will determine the maximum length of the longest all owablepath name on the filesystem which

holds/tmp/foo.
    $fd = POSIX::open("/tmp/foo", &POSIX::O_RDONLY);$path_max = POSIX::fpathconf($fd, &POSIX::_PC_PATH_MAX);
    Returnsundef on failure.

fprintf

fprintf()is C-specific--use printf instead.

fputc

fputc()is C-specific--use print instead.

fputs

fputs()is C-specific--use print instead.

fread

fread()is C-specific--use read instead.

free

free()is C-specific.

freopen

freopen()is C-specific--use open instead.

frexp

Return the mantissa and exponent of a floating-point number.

```
($mantissa, $exponent) = POSIX::frexp( 3.14 );
```

fscanf

fscanf()is C-specific--use <> and regular expressions instead.

fseek

Use method FileHandle::seek()
instead.

fsetpos

Use method FileHandle::setpos()
instead.

fstat

Get file status. This uses file descriptors such as those obtained bycallingPOSIX::open. The data returned is identical to the data fromPerl's builtin statfunction.
$fd = POSIX::open("foo", &POSIX::O_RDONLY);@stats = POSIX::fstat($fd);

ftell

Use method FileHandle::tell()
instead.

fwrite

fwrite()is C-specific--use print instead.

getc

This is identical to Perl's builtin
getc()
function.

getchar

Returns one character from STDIN.

getcwd

Returns the name of the current working directory.

getegid

Returns the effective group id.

getenv

Returns the value of the specified enironment variable.

geteuid

Returns the effective user id.

getgid

Returns the user's real group id.

getgrgid

This is identical to Perl's builtin
getgrgid()
function.

getgrnam

This is identical to Perl's builtin
getgrnam()
function.

getgroups

Returns the ids of the user's supplementary groups.

getlogin

This is identical to Perl's builtin
getlogin()
function.

getpgrp

This is identical to Perl's builtin
getpgrp()
function.

getpid

Returns the process's id.

getppid

This is identical to Perl's builtin
getppid()

function.

getpwnam

This is identical to Perl's builtin
getpwnam()
function.

getpwuid

This is identical to Perl's builtin
getpwuid()
function.

gets

Returns one line from STDIN.

getuid

Returns the user's id.

gmtime

This is identical to Perl's builtin
gmtime()
function.

isalnum

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isalpha

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isatty

Returns a boolean indicating whether the specified filehandle is connectedto a tty.

iscntrl

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isdigit

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isgraph

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

islower

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isprint

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

ispunct

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isspace

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isupper

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

isxdigit

This is identical to the C function, except that it can apply to a singlecharacter or to a whole string.

kill

This is identical to Perl's builtin
kill()
function.

labs

labs()is C-specific, use abs instead.

ldexp

This is identical to the C function
ldexp().

ldiv

ldiv()is C-specific, use / and int instead.

link

This is identical to Perl's builtin
link()
function.

localeconv

Get numeric formatting information. Returns a reference to a hashcontaining the current locale formatting values.The database for thede (Deutsch or German) locale.

$loc = POSIX::setlocale(&POSIX::LC_ALL, "de");print "Locale = $loc\n";$lconv = POSIX::localeconv();print "decimal_point = ", $lconv->{decimal_point}, "\n";print "thousands_sep = ", $lconv->{thousands_sep}, "\n";print "grouping = ", $lconv->{grouping}, "\n";print "int_curr_symbol = ", $lconv->{int_curr_symbol}, "\n";print "currency_symbol = ", $lconv->{currency_symbol}, "\n";print "mon_decimal_point = ", $lconv->{mon_decimal_point}, "\n";print "mon_thousands_sep = ", $lconv->{mon_thousands_sep}, "\n";print "mon_grouping = ", $lconv->{mon_grouping}, "\n";print "positive_sign = ", $lconv->{positive_sign}, "\n";print "negative_sign = ", $lconv->{negative_sign}, "\n";print "int_frac_digits = ", $lconv->{int_frac_digits}, "\n";print "frac_digits = ", $lconv->{frac_digits}, "\n";print "p_cs_precedes = ",

$lconv->{p_cs_precedes}, "\n";print "p_sep_by_space = ", $lconv->{p_sep_by_space}, "\n";print "n_cs_precedes = ", $lconv->{n_cs_precedes}, "\n";print "n_sep_by_space = ", $lconv->{n_sep_by_space}, "\n";print "p_sign_posn = ", $lconv->{p_sign_posn}, "\n";print "n_sign_posn = ", $lconv->{n_sign_posn}, "\n";

localtime

This is identical to Perl's builtin
localtime()
function.

log

This is identical to Perl's builtin
log()
function.

log10

This is identical to the C function
log10().

longjmp

longjmp()is C-specific: use die instead.

lseek

Move the read/write file pointer. This uses file descriptors such asthose obtained by callingPOSIX::open.

$fd = POSIX::open("foo", &POSIX::O_RDONLY);$off_t = POSIX::lseek($fd, 0, &POSIX::SEEK_SET);

Returnsundef on failure.

malloc

malloc()is C-specific.

mblen

This is identical to the C function
mblen().

mbstowcs

This is identical to the C function
mbstowcs().

mbtowc

This is identical to the C function
mbtowc().

memchr

memchr()is C-specific, useindex() instead.

memcmp

memcmp()is C-specific, use eq instead.

memcpy

memcpy()is C-specific, use = instead.

memmove

memmove()is C-specific, use = instead.

memset

memset()is C-specific, use x instead.

mkdir

This is identical to Perl's builtin
mkdir()
function.

mkfifo

This is similar to the C function
mkfifo()
.Returnsundef on failure.

mktime

Convert date/time info to a calendar time.Synopsis:

mktime(sec, min, hour, mday, mon, year, wday = 0, yday = 0, isdst = 0)

The month (mon), weekday (wday), and yearday (yday) begin at zero.I.e. January is 0, not 1; Sunday is 0, not 1; January 1st is 0, not 1. Theyear (year) is given in years since 1900. I.e. The year 1995 is 95; theyear 2001 is 101. Consult your system's

mktime()

manpage for detailsabout these and the other arguments.Calendar time for December 12, 1995, at 10:30 am.

$time_t = POSIX::mktime(0, 30, 10, 12, 11, 95);print "Date = ", POSIX::ctime($time_t);

Returnsundef on failure.

modf

Return the integral and fractional parts of a floating-point number.

($fractional, $integral) = POSIX::modf(3.14);

nice

This is similar to the C function

nice()

.Returnsundef on failure.

Offsetof

offsetof()is C-specific.

open

Open a file for reading for writing. This returns file descriptors, notPerl filehandles. UsePOSIX::close to close the file.Open a file read-only with mode 0666.

$fd = POSIX::open("foo");

Open a file for read and write.

$fd = POSIX::open("foo", &POSIX::O_RDWR);

Open a file for write, with truncation.

$fd = POSIX::open("foo", &POSIX::O_WRONLY | &POSIX::O_TRUNC);

Create a new file with mode 0640. Set up the file for writing.

$fd = POSIX::open("foo", &POSIX::O_CREAT | &POSIX::O_WRONLY,

0640);
Returnsundef on failure.

opendir

Open a directory for reading.

$dir = POSIX::opendir("/tmp");@files = POSIX::readdir($dir);POSIX::closedir($dir);

Returnsundef on failure.

pathconf

Retrieves the value of a configurable limit on a file or directory.The following will determine the maximum length of the longest allowablepathname on the filesystem which holds/tmp.

$path_max = POSIX::pathconf("/tmp", &POSIX::_PC_PATH_MAX);

Returnsundef on failure.

pause

This is similar to the C function
pause()
.Returnsundef on failure.

perror

This is identical to the C function
perror().

pipe

Create an interprocess channel. This returns file descriptors like thosereturned byPOSIX::open.

($fd0, $fd1) = POSIX::pipe();POSIX::write($fd0, "hello", 5);POSIX::read($fd1, $buf, 5);

pow

Computes$x raised to the power $exponent.

$ret = POSIX::pow($x, $exponent);

printf

Prints the specified arguments to STDOUT.

putc

putc()is C-specific--use print instead.

putchar

putchar()is C-specific--use print instead.

puts

puts()is C-specific--use print instead.

qsort

qsort()is C-specific, use sort instead.

raise

Sends the specified signal to the current process.

rand

rand()is non-portable, use Perl's rand instead.

read

Read from a file. This uses file descriptors such as those obtained bycallingPOSIX::open. If the buffer$buf is not large enough for theread then Perl will extend it to make room for the request.

$fd = POSIX::open("foo", &POSIX::O_RDONLY);$bytes = POSIX::read($fd, $buf, 3);

Returnsundef on failure.

readdir

This is identical to Perl's builtin
readdir()
function.

realloc

realloc()is C-specific.

remove

This is identical to Perl's builtin
unlink()
function.

rename

This is identical to Perl's builtin
rename()
function.

rewind

Seeks to the beginning of the file.

rewinddir

This is identical to Perl's builtin
rewinddir()
function.

rmdir

This is identical to Perl's builtin
rmdir()
function.

scanf

scanf()is C-specific--use <> and regular expressions instead.

setgid

Sets the real group id for this process.

setjmp

setjmp()is C-specific: use eval {} instead.

setlocale

Modifies and queries program's locale.The following will set

the traditional UNIX system locale behavior.
$loc = POSIX::setlocale(&POSIX::LC_ALL, "C");

setpgid

This is similar to the C function
setpgid()
.Returnsundef on failure.

setsid

This is identical to the C function
setsid().

setuid

Sets the real user id for this process.

sigaction

Detailed signal management. This uses POSIX::SigActionobjects for theaction andoldaction arguments. Consult your system's sigaction
man page for details.Synopsis:
sigaction(sig, action, old action = 0)
Returnsundef on failure.

siglongjmp

siglongjmp()is C-specific: use die instead.

sigpending

Examine signals that are blocked and pending. This uses POSIX::SigSet
objects for the sigset argument. Consult your system's sigpending
manpage for details.Synopsis:
sigpending(sigset)
Returnsundef on failure.
sigprocmask
Change and/or examine calling process's signal mask. This uses
POSIX::SigSetobjects for thesigset and old sigset arguments. Consult your system's sigprocmaskmanpage for details.Synopsis:

sigprocmask(how, sigset, oldsigset = 0)
Returnsundef on failure.

sigsetjmp

sigsetjmp()is C-specific: use eval {} instead.

sigsuspend

Install a signal mask and suspend process until signal arrives. This uses
POSIX::SigSetobjects for thesignal_mask argument. Consult yoursystem's sigsuspendmanpage for details.Synopsis:
sigsuspend(signal_mask)
Returnsundef on failure.

sin

This is identical to Perl's builtin
sin()
function.

sinh

This is identical to the C function
sinh().

sleep

This is identical to Perl's builtin
sleep()
function.

sprintf

This is identical to Perl's builtin
sprintf()
function.

sqrt

This is identical to Perl's builtin
sqrt()
function.

srand

srand().
sscanf
sscanf()is C-specific--use regular expressions instead.

stat

This is identical to Perl's builtin
stat()
function.

strcat

strcat()is C-specific, use .= instead.

strchr

strchr()is C-specific, useindex() instead.

strcmp

strcmp()is C-specific, use eq instead.

strcoll

This is identical to the C function
strcoll().

strcpy

strcpy()is C-specific, use = instead.

strcspn

strcspn()is C-specific, use regular expressions instead.

strerror

Returns the error string for the specified errno.

strftime

Convert date and time information to string. Returns the string.Synopsis:

strftime(fmt, sec, min, hour, mday, mon, year, wday = 0, yday = 0, isdst = 0)

The month (mon), weekday (wday), and yearday (yday) begin at zero.I.e. January is 0, not 1; Sunday is 0, not 1; January 1st is 0, not 1. Theyear (year) is given in years since 1900. I.e. The year 1995 is 95; theyear 2001 is 101. Consult your system's

strftime()

manpage for detailsabout these and the other arguments.The string for Tuesday, December 12, 1995.

$str = POSIX::strftime("%A, %B %d, %Y", 0, 0, 0, 12, 11, 95, 2);print "$str\n";

strlen

strlen()is C-specific, use length instead.

strncat

strncat()is C-specific, use .= instead.

strncmp

strncmp()is C-specific, use eq instead.

strncpy

strncpy()is C-specific, use = instead.

stroul

stroul()is C-specific.

strpbrk

strpbrk()is C-specific.

strrchr

strrchr()is C-specific, userindex() instead.

strspn

strspn()is C-specific.

strstr

This is identical to Perl's builtinindex() function.

strtod

strtod()is C-specific.

strtok

strtok()is C-specific.

strtol

strtol()is C-specific.

strxfrm

String transformation. Returns the transformed string.
$dst = POSIX::strxfrm($src);

sysconf

Retrieves values of system configurable variables.The following will get the machine's clock speed.
$clock_ticks = POSIX::sysconf(&POSIX::_SC_CLK_TCK);
Returnsundef on failure.

system

This is identical to Perl's builtin
system()
function.

tan

This is identical to the C function
tan().

tanh

This is identical to the C function

tanh().

tcdrain

This is similar to the C function
tcdrain()
.Returnsundef on failure.

tcflow

This is similar to the C function
tcflow()
.Returnsundef on failure.

tcflush

This is similar to the C function
tcflush()
.Returnsundef on failure.

tcgetpgrp

This is identical to the C function
tcgetpgrp().

tcsendbreak

This is similar to the C function
tcsendbreak()
.Returnsundef on failure.

tcsetpgrp

This is similar to the C function
tcsetpgrp()
.Returnsundef on failure.

time

This is identical to Perl's builtin
time()
function.

times

The times()function returns elapsed realtime since some point in the past(such as system startup), user and system times for this process, and userand system times used by child processes. All times are returned in clockticks.

($realtime, $user, $system, $cuser, $csystem) = POSIX::times();

Note: Perl's builtin
times()
function returns four values, measured inseconds.

tmpfile

Use method FileHandle::new_tmpfile()
instead.

tmpnam

Returns a name for a temporary file.
$tmpfile = POSIX::tmpnam();

tolower

This is identical to Perl's builtinlc() function.

toupper

This is identical to Perl's builtinuc() function.

ttyname

This is identical to the C function
ttyname().

tzname

Retrieves the time conversion information from the tznamevariable.

POSIX::tzset();($std, $dst) = POSIX::tzname();

tzset

This is identical to the C function
tzset().

umask

This is identical to Perl's builtin
umask()
function.

uname

Get name of current operating system.
($sysname, $nodename, $release, $version, $machine) =
POSIX::uname();

ungetc

Use method FileHandle::ungetc()
instead.

unlink

This is identical to Perl's builtin
unlink()

function.

utime
This is identical to Perl's builtin
utime()
function.

vfprintf

vfprintf()is C-specific.

vprintf

vprintf()is C-specific.

vsprintf

vsprintf()is C-specific.

wait

This is identical to Perl's builtin
wait()
function.

waitpid

Wait for a child process to change state. This is identical to Perl'sbuiltin
waitpid()
function.
$pid = POSIX::waitpid(-1, &POSIX::WNOHANG);print "status = ", ($? / 256), "\n";

wcstombs

This is identical to the C function
wcstombs().

wctomb

This is identical to the C function
wctomb().

write

Write to a file. This uses file descriptors such as those obtained bycallingPOSIX::open.
$fd = POSIX::open("foo", &POSIX::O_WRONLY);$buf = "hello";$bytes = POSIX::write($b, $buf, 5);
Returnsundef on failure.

CLASSES

FileHandle

new

Open a file and return a Perl filehandle. The first parameter is thefilename and the second parameter is the mode. The mode should be specifiedasa for append,w for write, and < or``'' for read.Open a file for reading.
$fh = FileHandle->new("foo", "");die "Unable to open foo for reading" unless $fh;
Open a file for writing.

$fh = FileHandle->new("foo", "w");die "Unable to open foo for writing" unless $fh;
Use FileHandle::close()
to close the file or let the FileHandle object'sdestructor perform the close.

clearerr

Resets the error indicator and EOF indicator to zero.
$fh->clearerr;

close

Close the file.
$fh->close;

eof

Tests for end of file.
if($fh->eof){print "end of file\n";}

error

Returns non-zero if there has been an error while reading or writing a file.
if($fh->error){print "error\n";}

fileno

Returns the integer file descriptor associated with the file.
$fileno = $fh->fileno;

flush

Flush the stream.
$fh->flush;
Returnsundef on failure.

getc

Get a character from the stream.
$ch = $fh->getc;

getpos

Retrieve the file pointer position. The returned value can be used as anargument to
setpos().
$pos = $fh->getpos;

gets

Retrieve a line from the open file.
$line = $fh->gets;

new_from_fd

Open a file using a file descriptor. Return a Perl filehandle. The firstparameter should be a file descriptor, which can come fromPOSIX::open().The second parameter, the mode, should bea for append,w for write,and < or``'' for read. The mode should match the mode which was usedwhen the file descriptor was created.
$fd = POSIX::open("typemap");$fh = FileHandle->new_from_fd($fd, "

new_tmpfile

Creates a temporary file, opens it for writing, and returns a Perlfilehandle. Consult your system's
tmpfile()
manpage for details.
$fh = FileHandle->new_tmpfile;die "FileHandle failed" unless $fh;
seek
Reposition file pointer.
$fh->seek(2, &POSIX::SEEK_SET);

setbuf

setpos

Set the file pointer position.
$pos = $fh->getpos;$fh->setpos($pos);
Returnsundef on failure.

setvbuf

Returnsundef on failure.

tell

Returns the current file position, in bytes.
$pos = $fh->tell;

ungetc

POSIX::SigAction

new

Creates a new POSIX::SigActionobject which corresponds to the Cstruct sigaction. This object will be destroyed automatically when it isno longer needed. The first parameter is the fully-qualified name of a subwhich is a signal-handler. The second parameter is a POSIX::SigSet
object. The third parameter contains thesa_flags.
$sigset = POSIX::SigSet->new;$sigaction = POSIX::SigAction->new('main::handler', $sigset, &POSIX::SA_NOCLDSTOP);
This POSIX::SigActionobject should be used with thePOSIX::sigaction()
function.

POSIX::SigSet

new

Create a new SigSet object. This object will be destroyed automaticallywhen it is no longer needed. Arguments may be supplied to initialize theset.Create an empty set.
$sigset = POSIX::SigSet->new;
Create a set with SIGUSR1.
$sigset = POSIX::SigSet->new(&POSIX::SIGUSR1);

addset

Add a signal to a SigSet object.
$sigset->addset(&POSIX::SIGUSR2);
Returnsundef on failure.

delset

Remove a signal from the SigSet object.
$sigset->delset(&POSIX::SIGUSR2);
Returnsundef on failure.

emptyset

Initialize the SigSet object to be empty.
$sigset->emptyset();
Returnsundef on failure.

fillset

Initialize the SigSet object to include all signals.
$sigset->fillset();
Returnsundef on failure.

ismember

Tests the SigSet object to see if it contains a specific signal.
if($sigset->ismember(&POSIX::SIGUSR1)){print "contains SIGUSR1\n";}

POSIX::Termios

new

Create a new Termios object. This object will be destroyed automaticallywhen it is no longer needed.
$termios = POSIX::Termios->new;

getattr

Get terminal control attributes.Obtain the attributes for stdin.
$termios->getattr()
Obtain the attributes for stdout.
$termios->getattr(1)
Returnsundef on failure.

getcc

Retrieve a value from the c_cc field of a termios object. The c_cc field isan array so an index must be specified.
$c_cc[1] = $termios->getcc(1);

getcflag

Retrieve the c_cflag field of a termios object.

```
$c_cflag = $termios->getcflag;
```

getiflag

Retrieve the c_iflag field of a termios object.

```
$c_iflag = $termios->getiflag;
```

getispeed

Retrieve the input baud rate.

```
$ispeed = $termios->getispeed;
```

getlflag

Retrieve the c_lflag field of a termios object.

```
$c_lflag = $termios->getlflag;
```

getoflag

Retrieve the c_oflag field of a termios object.

```
$c_oflag = $termios->getoflag;
```

getospeed

Retrieve the output baud rate.

```
$ospeed = $termios->getospeed;
```

setattr

Set terminal control attributes.Set attributes immediately for stdout.

```
$termios->setattr( 1, &POSIX::TCSANOW );
```

Returnsundef on failure.

setcc

Set a value in the c_cc field of a termios object. The c_cc field is anarray so an index must be specified.

```
$termios->setcc( 1, &POSIX::VEOF );
```

setcflag

Set the c_cflag field of a termios object.
$termios->setcflag(&POSIX::CLOCAL);

setiflag

Set the c_iflag field of a termios object.
$termios->setiflag(&POSIX::BRKINT);

setispeed

Set the input baud rate.
$termios->setispeed(&POSIX::B9600);
Returnsundef on failure.

setlflag

Set the c_lflag field of a termios object.
$termios->setlflag(&POSIX::ECHO);

setoflag

Set the c_oflag field of a termios object.
$termios->setoflag(&POSIX::OPOST);

setospeed

Set the output baud rate.
$termios->setospeed(&POSIX::B9600);
Returnsundef on failure.

Baud rate values

B38400 B75 B200 B134 B300 B1800 B150 B0 B19200 B1200 B9600 B600 B4800 B50 B2400 B110

Terminal interface values

TCSADRAIN TCSANOW TCOON TCIOFLUSH TCOFLUSH TCION TCIFLUSH TCSAFLUSH TCIOFF TCOOFF

c_cc field values

VEOF VEOL VERASE VINTR VKILL VQUIT VSUSP VSTART VSTOP VMIN VTIME NCCS

c_cflag field values

CLOCAL CREAD CSIZE CS5 CS6 CS7 CS8 CSTOPB HUPCL PARENB PARODD

c_iflag field values

BRKINT ICRNL IGNBRK IGNCR IGNPAR INLCR INPCK ISTRIP IXOFF IXON PARMRK

c_lflag field values

ECHO ECHOE ECHOK ECHONL ICANON IEXTEN ISIG NOFLSH TOSTOP

c_oflag field values

OPOST

PATHNAME CONSTANTS

Constants

_PC_CHOWN_RESTRICTED _PC_LINK_MAX _PC_MAX_CANON _PC_MAX_INPUT _PC_NAME_MAX _PC_NO_TRUNC _PC_PATH_MAX _PC_PIPE_BUF _PC_VDISABLE

POSIX CONSTANTS

Constants

_POSIX_ARG_MAX _POSIX_CHILD_MAX _POSIX_CHOWN_RESTRICTED _POSIX_JOB_CONTROL _POSIX_LINK_MAX _POSIX_MAX_CANON _POSIX_MAX_INPUT _POSIX_NAME_MAX _POSIX_NGROUPS_MAX _POSIX_NO_TRUNC _POSIX_OPEN_MAX _POSIX_PATH_MAX _POSIX_PIPE_BUF _POSIX_SAVED_IDS _POSIX_SSIZE_MAX _POSIX_STREAM_MAX _POSIX_TZNAME_MAX _POSIX_VDISABLE _POSIX_VERSION

SYSTEM CONFIGURATION

Constants

_SC_ARG_MAX _SC_CHILD_MAX _SC_CLK_TCK _SC_JOB_CONTROL

_SC_NGROUPS_MAX _SC_OPEN_MAX _SC_SAVED_IDS _SC_STREAM_MAX _SC_TZNAME_MAX _SC_VERSION

ERRNO

Constants

E2BIG EACCES EAGAIN EBADF EBUSY ECHILD EDEADLK EDOM EEXIST EFAULT EFBIG EINTR EINVAL EIO EISDIR EMFILE EMLINK ENAMETOOLONG ENFILE ENODEV ENOENT ENOEXEC ENOLCK ENOMEM ENOSPC ENOSYS ENOTDIR ENOTEMPTY ENOTTY ENXIO EPERM EPIPE ERANGE EROFS ESPIPE EsrcH EXDEV

FCNTL

Constants

FD_CLOEXEC F_DUPFD F_GETFD F_GETFL F_GETLK F_OK F_RDLCK F_SETFD F_SETFL F_SETLK F_SETLKW F_UNLCK F_WRLCK O_ACCMODE O_APPEND O_CREAT O_EXCL O_NOCTTY O_NONBLOCK O_RDONLY O_RDWR O_TRUNC O_WRONLY

FLOAT

Constants

DBL_DIG DBL_EPSILON DBL_MANT_DIG DBL_MAX DBL_MAX_10_EXP DBL_MAX_EXP DBL_MIN DBL_MIN_10_EXP DBL_MIN_EXP FLT_DIG FLT_EPSILON FLT_MANT_DIG FLT_MAX FLT_MAX_10_EXP FLT_MAX_EXP FLT_MIN FLT_MIN_10_EXP FLT_MIN_EXP FLT_RADIX FLT_ROUNDS LDBL_DIG LDBL_EPSILON LDBL _MANT_DIG LDBL_MAX LDBL_MAX_10_EXP LDBL_MAX_EXP LDBL_MIN LDBL_MIN_10_EXP LDBL_MIN_EXP

LIMITS

Constants

ARG_MAX CHAR_BIT CHAR_MAX CHAR_MIN CHILD_MAX INT_MAX INT_MIN LINK_MAX LONG_MAX LONG_MIN MAX_CANON MAX_INPUT MB_LEN_MAX NAME_MAX NGROUPS_MAX OPEN_MAX PATH_MAX PIPE_BUF SCHAR_MAX SCHAR_MIN SHRT_MAX SHRT_MIN SSIZE_MAX STREAM_MAX TZNAME_MAX UCHAR_MAX UINT_MAX ULONG_MAX USHRT_MAX

LOCALE

Constants

LC_ALL LC_COLLATE LC_CTYPE LC_MONETARY LC_NUMERIC LC_TIME

MATH

Constants
HUGE_VAL

SIGNAL

Constants

SA_NOCLDSTOP SIGABRT SIGALRM SIGCHLD SIGCONT SIGFPE SIGHUP SIGILL SIGINT SIGKILL SIGPIPE SIGQUIT SIGSEGV SIGSTOP SIGTERM SIGTSTP SIGTTIN SIGTTOU SIGUSR1 SIGUSR2 SIG_BLOCK SIG_DFL SIG_ERR SIG_IGN SIG_SETMASK SIG_UNBLOCK

STAT

Constants

S_IRGRP S_IROTH S_IRUSR S_IRWXG S_IRWXO S_IRWXU S_ISGID S_ISUID S_IWGRP S_IWOTH S_IWUSR S_IXGRP S_IXOTH S_IXUSR

Macros

S_ISBLK S_ISCHR S_ISDIR S_ISFIFO S_ISREG

STDLIB

Constants

EXIT_FAILURE EXIT_SUCCESS MB_CUR_MAX RAND_MAX

STDIO

Constants

BUFSIZ EOF FILENAME_MAX L_ctermid L_cuserid L_tmpname TMP_MAX _IOFBF _IOLBF _IONBF

TIME

Constants

CLK_TCK CLOCKS_PER_SEC

UNISTD

Constants

R_OK SEEK_CUR SEEK_END SEEK_SET STDIN_FILENO STDOUT_FILENO STRERR_FILENO W_OK X_OK

WAIT

Constants

WNOHANG WUNTRACED

Macros

WIFEXITED WEXITSTATUS WIFSIGNALED WTERMSIG WIFSTOPPED WSTOPSIG

CREATION

This document generated by ./mkposixman.PL version 19951212.