

**NAME**

**sigtimedwait**, **sigwaitinfo** - wait for queued signals (REALTIME)

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <signal.h>
```

*int*

```
sigtimedwait(const sigset_t *restrict set, siginfo_t *restrict info, const struct timespec *restrict timeout);
```

*int*

```
sigwaitinfo(const sigset_t * restrict set, siginfo_t * restrict info);
```

**DESCRIPTION**

The **sigtimedwait**() system call is equivalent to **sigwaitinfo**() except that if none of the signals specified by *set* are pending, **sigtimedwait**() waits for the time interval specified in the *timespec* structure referenced by *timeout*. If the *timespec* structure pointed to by *timeout* is zero-valued and if none of the signals specified by *set* are pending, then **sigtimedwait**() returns immediately with an error. If *timeout* is the NULL pointer, **sigtimedwait**() blocks indefinitely. CLOCK\_MONOTONIC clock is used to measure the time interval specified by the *timeout* argument.

The **sigwaitinfo**() system call selects the pending signal from the set specified by *set*. Should any of multiple pending signals in the range SIGRTMIN to SIGRTMAX be selected, it shall be the lowest numbered one. The selection order between realtime and non-realtime signals, or between multiple pending non-realtime signals, is unspecified. If no signal in *set* is pending at the time of the call, the calling thread is suspended until one or more signals in *set* become pending or until it is interrupted by an unblocked, caught signal.

The **sigwaitinfo**() system call is equivalent to the **sigwait**() system call if the *info* argument is NULL. If the *info* argument is non-NULL, the **sigwaitinfo**() function is equivalent to **sigwait**(), except that the selected signal number shall be stored in the *si\_signo* member, and the cause of the signal shall be stored in the *si\_code* member. Besides this, the **sigwaitinfo**() and **sigtimedwait**() system calls may return EINTR if interrupted by signal, which is not allowed for the **sigwait**() function.

If any value is queued to the selected signal, the first such queued value is dequeued and, if the *info* argument is non-NULL, the value is stored in the *si\_value* member of *info*. The system resource used to queue the signal is released and returned to the system for other use. If no value is queued, the content of the *si\_value* member is zero-valued. If no further signals are queued for the selected signal, the

pending indication for that signal is reset.

## RETURN VALUES

Upon successful completion (that is, one of the signals specified by *set* is pending or is generated) **sigwaitinfo()** and **sigtimedwait()** return the selected signal number. Otherwise, the functions return a value of -1 and set the global variable *errno* to indicate the error.

## ERRORS

The **sigtimedwait()** system call will fail if:

[EAGAIN]           No signal specified by *set* was generated within the specified timeout period.

The **sigtimedwait()** and **sigwaitinfo()** system calls fail if:

[EINTR]            The wait was interrupted by an unblocked, caught signal.

The **sigtimedwait()** system call may also fail if:

[EINVAL]           The *timeout* argument specified a *tv\_nsec* value less than zero or greater than or equal to 1000 million. Kernel only checks for this error if no signal is pending in *set* and it is necessary to wait.

## SEE ALSO

sigaction(2), sigpending(2), sigqueue(2), sigsuspend(2), sigwait(2), pause(3), pthread\_sigmask(3), siginfo(3)

## STANDARDS

The **sigtimedwait()** and **sigwaitinfo()** system calls conform to ISO/IEC 9945-1:1996 ("POSIX.1"). POSIX leaves the behavior of **sigtimedwait()** with a NULL *timeout* pointer unspecified.