

**NAME**

sigwait – wait for a signal

**SYNOPSIS**

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

```
int sigwait(const sigset_t *set, int *sig);
```

Feature Test Macro Requirements for glibc (see [feature\\_test\\_macros\(7\)](#)):

**sigwait()**:

Since glibc 2.26: `_POSIX_C_SOURCE >= 199506L`

Glibc 2.25 and earlier: `_POSIX_C_SOURCE`

**DESCRIPTION**

The **sigwait()** function suspends execution of the calling thread until one of the signals specified in the signal set *set* becomes pending. The function accepts the signal (removes it from the pending list of signals), and returns the signal number in *sig*.

The operation of **sigwait()** is the same as [sigwaitinfo\(2\)](#), except that:

- \* **sigwait()** returns only the signal number, rather than a *siginfo\_t* structure describing the signal.
- \* The return values of the two functions are different.

**RETURN VALUE**

On success, **sigwait()** returns 0. On error, it returns a positive error number (listed in [ERRORS](#)).

**ERRORS****EINVAL**

*set* contains an invalid signal number.

**ATTRIBUTES**

For an explanation of the terms used in this section, see [attributes\(7\)](#).

Interface	Attribute	Value
<b>sigwait()</b>	Thread safety	MT-Safe

**CONFORMING TO**

POSIX.1-2001, POSIX.1-2008.

**NOTES**

**sigwait()** is implemented using [sigtimedwait\(2\)](#).

The glibc implementation of **sigwait()** silently ignores attempts to wait for the two real-time signals that are used internally by the NPTL threading implementation. See [nptl\(7\)](#) for details.

**EXAMPLE**

See [pthread\\_sigmask\(3\)](#).

**SEE ALSO**

[sigaction\(2\)](#), [signalfd\(2\)](#), [sigpending\(2\)](#), [sigsuspend\(2\)](#), [sigwaitinfo\(2\)](#), [sigsetops\(3\)](#), [signal\(7\)](#)

**COLOPHON**

This page is part of release 4.16 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.