NAME

sigsuspend − wait for a signal

SYNOPSIS

#include <signal.h>

int sigsuspend(const sigset_t **mask*);

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

sigsuspend(): _POSIX_C_SOURCE >= 1 || _XOPEN_SOURCE || _POSIX_SOURCE

DESCRIPTION

sigsuspend() temporarily replaces the signal mask of the calling process with the mask given by *mask* and then suspends the process until delivery of a signal whose action is to invoke a signal handler or to terminate a process.

If the signal terminates the process, then **sigsuspend**() does not return. If the signal is caught, then **sigsuspend**() returns after the signal handler returns, and the signal mask is restored to the state before the call to **sigsuspend**().

It is not possible to block **SIGKILL** or **SIGSTOP**; specifying these signals in *mask*, has no effect on the process's signal mask.

RETURN VALUE

sigsuspend() always returns −1, normally with the error **EINTR**.

ERRORS

EFAULT

mask points to memory which is not a valid part of the process address space.

EINTR

The call was interrupted by a signal.

CONFORMING TO

POSIX.1-2001.

NOTES

Normally, **sigsuspend**() is used in conjunction with **sigprocmask**(2) in order to prevent delivery of a signal during the execution of a critical code section. The caller first blocks the signals with **sigprocmask**(2). When the critical code has completed, the caller then waits for the signals by calling **sigsuspend**() with the signal mask that was returned by **sigprocmask**(2) (in the *oldset* argument).

See **sigsetops**(3) for details on manipulating signal sets.

SEE ALSO

kill(2), **pause**(2), **sigaction**(2), **signal**(2), **sigprocmask**(2), **sigwaitinfo**(2), **sigsetops**(3), **sigwait**(3), **signal**(7)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.