



Linux Ubuntu 26.04 Manual Pages on command 'rt_sigsuspend.2'

\$ man rt_sigsuspend.2

sigsuspend(2) System Calls Manual sigsuspend(2)

NAME

sigsuspend, rt_sigsuspend - wait for a signal

LIBRARY

Standard C library (libc, -lc)

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
```

DESCRIPTION

sigsuspend() temporarily replaces the signal mask of the calling thread with the mask given by mask and then suspends the thread 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 thread's signal mask.

RETURN VALUE

sigsuspend() always returns -1, with errno set to indicate the error (normally, 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; signal(7).

STANDARDS

POSIX.1-2024.

HISTORY

POSIX.1-2001.

C library/kernel differences

The original Linux system call was named `sigsuspend()`. However, with the addition of real-time signals in Linux 2.2, the fixed-size, 32-bit `sigset_t` type supported by that system call was no longer fit for purpose. Consequently, a new system call, `rt_sigsuspend()`, was added to support an enlarged `sigset_t` type. The new system call takes a second argument, `size_t sigsetsize`, which specifies the size in bytes of the signal set in `mask`. This argument is currently required to have the value `sizeof(sigset_t)` (or the error `EINVAL` results). The glibc `sigsuspend()` wrapper function hides these details from us, transparently calling `rt_sigsuspend()` when the kernel provides it.

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)`