

**NAME**

**sigsuspend** - atomically release blocked signals and wait for interrupt

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

*int*

```
sigsuspend(const sigset_t *sigmask);
```

**DESCRIPTION**

The **sigsuspend()** system call temporarily changes the blocked signal mask to the set to which *sigmask* points, and then waits for a signal to arrive; on return the previous set of masked signals is restored. The signal mask set is usually empty to indicate that all signals are to be unblocked for the duration of the call.

In normal usage, a signal is blocked using **sigprocmask(2)** to begin a critical section, variables modified on the occurrence of the signal are examined to determine that there is no work to be done, and the process pauses awaiting work by using **sigsuspend()** with the previous mask returned by **sigprocmask(2)**.

**RETURN VALUES**

The **sigsuspend()** system call always terminates by being interrupted, returning -1 with *errno* set to EINTR.

**SEE ALSO**

**pselect(2)**, **sigaction(2)**, **sigpending(2)**, **sigprocmask(2)**, **sigtimedwait(2)**, **sigwait(2)**, **sigwaitinfo(2)**, **sigsetops(3)**

**STANDARDS**

The **sigsuspend()** system call is expected to conform to IEEE Std 1003.1-1990 ("POSIX.1").