

NAME

sighold, sigignore, sigpause, sigrelse, sigset - legacy interface for signal management

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <signal.h>

int

sighold(*int sig*);

int

sigignore(*int sig*);

int

xsi_sigpause(*int sigmask*);

int

sigrelse(*int sig*);

void () (int)*

sigset(*int, void (*) (int)*);

int

sigpause(*int sigmask*);

DESCRIPTION

This interface is made obsolete by sigsuspend(2) **and** sigaction(2).

The **sigset()** function modifies signal dispositions. The *sig* argument specifies the signal, which may be any signal except SIGKILL and SIGSTOP. The *disp* argument specifies the signal's disposition, which may be SIG_DFL, SIG_IGN, or the address of a signal handler. If **sigset()** is used, and *disp* is the address of a signal handler, the system adds *sig* to the signal mask of the calling process before executing the signal handler; when the signal handler returns, the system restores the signal mask of the calling process to its state prior to the delivery of the signal. In addition, if **sigset()** is used, and *disp* is equal to SIG_HOLD, *sig* is added to the signal mask of the calling process and *sig*'s disposition remains unchanged. If **sigset()** is used, and *disp* is not equal to SIG_HOLD, *sig* is removed from the signal mask of the calling process.

The **sighold()** function adds *sig* to the signal mask of the calling process.

The **sigrelse()** function removes *sig* from the signal mask of the calling process.

The **sigignore()** function sets the disposition of *sig* to SIG_IGN.

The **xsi_sigpause()** function removes *sig* from the signal mask of the calling process and suspend the calling process until a signal is received. The **xsi_sigpause()** function restores the signal mask of the process to its original state before returning.

The **sigpause()** function assigns *sigmask* to the set of masked signals and then waits for a signal to arrive; on return the set of masked signals is restored. The *sigmask* argument is usually 0 to indicate that no signals are to be blocked.

RETURN VALUES

The **sigpause()** and **xsi_sigpause()** functions always terminate by being interrupted, returning -1 with *errno* set to EINTR.

Upon successful completion, **sigset()** returns SIG_HOLD if the signal had been blocked and the signal's previous disposition if it had not been blocked. Otherwise, SIG_ERR is returned and *errno* set to indicate the error.

For all other functions, upon successful completion, 0 is returned. Otherwise, -1 is returned and *errno* is set to indicate the error:

[EINVAL] The *sig* argument is not a valid signal number.

[EINVAL] For **sigset()** and **sigignore()** functions, an attempt was made to catch or ignore SIGKILL or SIGSTOP.

SEE ALSO

kill(2), sigaction(2), sigblock(2), sigprocmask(2), sigsuspend(2), sigvec(2)

STANDARDS

The **sigpause()** function is implemented for compatibility with historic 4.3BSD applications. An incompatible interface by the same name, which used a single signal number rather than a mask, was present in AT&T System V UNIX, and was copied from there into the **X/Open System Interfaces** (XSI) option of IEEE Std 1003.1-2001 ("POSIX.1"). FreeBSD implements it under the name **xsi_sigpause()**. The **sighold()**, **sigignore()**, **sigrelse()** and **sigset()** functions are implemented for compatibility with **System V** and **XSI** interfaces.

HISTORY

The **sigpause()** function appeared in 4.2BSD and has been deprecated. All other functions appeared in FreeBSD 8.1 and were deprecated before being implemented.