Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sighold.3p' command

$ man sighold.3p

SIGHOLD(3P) POSIX Programmer's Manual SIGHOLD(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

sighold, sigignore, sigpause, sigrelse, sigset ? signal management

SYNOPSIS

```
#include <signal.h>

int sighold(int sig);

int sigignore(int sig);

int sigpause(int sig);

int sigrelse(int sig);

void (*sigset(int sig, void (*disp)(int)))(int);
```

DESCRIPTION

Use of any of these functions is unspecified in a multi-threaded process.

The sighold(), sigignore(), sigpause(), sigrelse(), and sigset() func? tions provide simplified signal management.

The sigset() function shall modify signal dispositions. The sig argu? ment 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 sys? tem shall add sig to the signal mask of the calling process before exe? cuting the signal handler; when the signal handler returns, the system shall restore 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 shall be added to the signal mask of the calling process and sig's disposition shall remain unchanged. If sigset() is used, and disp is not equal to SIG_HOLD, sig shall be re? moved from the signal mask of the calling process.

The sighold() function shall add sig to the signal mask of the calling process.

The sigrelse() function shall remove sig from the signal mask of the calling process.

The sigignore() function shall set the disposition of sig to SIG_IGN.

The sigpause() function shall remove sig from the signal mask of the calling process and suspend the calling process until a signal is re? ceived. The sigpause() function shall restore the signal mask of the process to its original state before returning.

If the action for the SIGCHLD signal is set to SIG_IGN, child processes of the calling processes shall not be transformed into zombie processes when they terminate. If the calling process subsequently waits for its children, and the process has no unwaited-for children that were trans? formed into zombie processes, it shall block until all of its children terminate, and wait(), waitid(), and waitpid() shall fail and set errno to [ECHILD].

RETURN VALUE

Upon successful completion, sigset() shall return SIG_HOLD if the sig? nal had been blocked and the signal's previous disposition if it had not been blocked. Otherwise, SIG_ERR shall be returned and errno set to indicate the error.

The sigpause() function shall suspend execution of the thread until a signal is received, whereupon it shall return -1 and set errno to

[EINTR].

For all other functions, upon successful completion, 0 shall be re? turned. Otherwise, -1 shall be returned and errno set to indicate the error.

ERRORS

These functions shall fail if:

EINVAL The sig argument is an illegal signal number.

The sigset() and sigignore() functions shall fail if:

EINVAL An attempt is made to catch a signal that cannot be caught, or to ignore a signal that cannot be ignored.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The sigaction() function provides a more comprehensive and reliable mechanism for controlling signals; new applications should use the sigaction() function instead of the obsolescent sigset() function.

The sighold() function, in conjunction with sigrelse() or sigpause(), may be used to establish critical regions of code that require the de? livery of a signal to be temporarily deferred. For broader portability, the pthread_sigmask() or sigprocmask() functions should be used instead of the obsolescent sighold() and sigrelse() functions.

For broader portability, the sigsuspend() function should be used in? stead of the obsolescent sigpause() function.

RATIONALE

Each of these historic functions has a direct analog in the other func? tions which are required to be per-thread and thread-safe (aside from sigprocmask(), which is replaced by pthread_sigmask()). The sigset() function can be implemented as a simple wrapper for sigaction(). The sighold() function is equivalent to sigprocmask() or pthread_sigmask() with SIG_BLOCK set. The sigignore() function is equivalent to sigac? tion() with SIG_IGN set. The sigpause() function is equivalent to sig? suspend(). The sigrelse() function is equivalent to sigprocmask() or

pthread_sigmask() with SIG_UNBLOCK set.

FUTURE DIRECTIONS

These functions may be removed in a future version.

SEE ALSO

Section 2.4, Signal Concepts, exec, pause(), pthread_sigmask(), sigac?
tion(), signal(), sigsuspend(), wait(), waitid()

The Base Definitions volume of POSIX.1?2017, <signal.h>