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 'sigwait.3p' command

$ man sigwait.3p

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

sigwait ? wait for queued signals

SYNOPSIS

#include <signal.h>

int sigwait(const sigset_t *restrict set, int *restrict sig);

DESCRIPTION

The sigwait() function shall select a pending signal from set, atomi? cally clear it from the system's set of pending signals, and return that signal number in the location referenced by sig. If prior to the call to sigwait() there are multiple pending instances of a single sig? nal number, it is implementation-defined whether upon successful return there are any remaining pending signals for that signal number. If the implementation supports queued signals and there are multiple signals queued for the signal number selected, the first such queued signal shall cause a return from sigwait() and the remainder shall remain queued. If no signal in set is pending at the time of the call, the thread shall be suspended until one or more becomes pending. The sig?

nals defined by set shall have been blocked at the time of the call to sigwait(); otherwise, the behavior is undefined. The effect of sig? wait() on the signal actions for the signals in set is unspecified.

If more than one thread is using sigwait() to wait for the same signal, no more than one of these threads shall return from sigwait() with the signal number. If more than a single thread is blocked in sigwait() for a signal when that signal is generated for the process, it is unspeci? fied which of the waiting threads returns from sigwait(). If the sig? nal is generated for a specific thread, as by pthread_kill(), only that thread shall return.

Should any of the multiple pending signals in the range SIGRTMIN to SIGRTMAX be selected, it shall be the lowest numbered one. The selec? tion order between realtime and non-realtime signals, or between multi? ple pending non-realtime signals, is unspecified.

RETURN VALUE

Upon successful completion, sigwait() shall store the signal number of the received signal at the location referenced by sig and return zero.

Otherwise, an error number shall be returned to indicate the error.

ERRORS

The sigwait() function may fail if:

EINVAL The set argument contains an invalid or unsupported signal num? ber.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

To provide a convenient way for a thread to wait for a signal, this volume of POSIX.1?2017 provides the sigwait() function. For most cases where a thread has to wait for a signal, the sigwait() function should be quite convenient, efficient, and adequate.

However, requests were made for a lower-level primitive than sigwait()

and for semaphores that could be used by threads. After some considera?
tion, threads were allowed to use semaphores and sem_post() was defined
to be async-signal-safe.

In summary, when it is necessary for code run in response to an asyn?
chronous signal to notify a thread, sigwait() should be used to handle
the signal. Alternatively, if the implementation provides semaphores,
they also can be used, either following sigwait() or from within a sig?
nal handling routine previously registered with sigaction().

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.4, Signal Concepts, Section 2.8.1, Realtime Signals, pause(),
pthread_sigmask(), sigaction(), sigpending(), sigsuspend(), sigtimed?
wait()

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