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 'pthread_kill.3p' command

$ man pthread_kill.3p

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

pthread_kill ? send a signal to a thread

SYNOPSIS

```
#include <signal.h>
int pthread_kill(pthread_t thread, int sig);
```

DESCRIPTION

The pthread_kill() function shall request that a signal be delivered to the specified thread.

As in kill(), if sig is zero, error checking shall be performed but no signal shall actually be sent.

RETURN VALUE

Upon successful completion, the function shall return a value of zero. Otherwise, the function shall return an error number. If the pthread_kill() function fails, no signal shall be sent.

ERRORS

The pthread_kill() function shall fail if:

EINVAL The value of the sig argument is an invalid or unsupported sig?

nal number.

The pthread_kill() function shall not return an error code of [EINTR].

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The pthread_kill() function provides a mechanism for asynchronously di? recting a signal at a thread in the calling process. This could be used, for example, by one thread to affect broadcast delivery of a sig? nal to a set of threads.

Note that pthread_kill() only causes the signal to be handled in the context of the given thread; the signal action (termination or stop? ping) affects the process as a whole.

RATIONALE

If an implementation detects use of a thread ID after the end of its lifetime, it is recommended that the function should fail and report an [ESRCH] error.

Existing implementations vary on the result of a pthread_kill() with a thread ID indicating an inactive thread (a terminated thread that has not been detached or joined). Some indicate success on such a call, while others give an error of [ESRCH]. Since the definition of thread lifetime in this volume of POSIX.1?2017 covers inactive threads, the [ESRCH] error as described is inappropriate in this case. In particu? lar, this means that an application cannot have one thread check for termination of another with pthread_kill().

FUTURE DIRECTIONS

A future version of this standard may require that pthread_kill() not fail with [ESRCH] in the case of sending signals to an inactive thread (a terminated thread not yet detached or joined), even though no signal will be delivered because the thread is no longer running.

SEE ALSO

kill(), pthread_self(), raise()

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