



## Rocky Enterprise Linux 9.2 Manual Pages on command 'pthread\_kill.3'

C:\>man pthread\_kill.3

PTHREAD\_KILL(3)                   Linux Programmer's Manual                   PTHREAD\_KILL(3)

### NAME

pthread\_kill - send a signal to a thread

### SYNOPSIS

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

```
int pthread_kill(pthread_t thread, int sig);
```

Compile and link with -pthread.

Feature Test Macro Requirements for glibc (see feature\_test\_macros(7)):

```
pthread_kill():
```

```
  _POSIX_C_SOURCE >= 199506L || _XOPEN_SOURCE >= 500
```

### DESCRIPTION

The pthread\_kill() function sends the signal sig to thread, a thread in the same process as the caller. The signal is asynchronously directed to thread.

If sig is 0, then no signal is sent, but error checking is still performed.

### RETURN VALUE

On success, pthread\_kill() returns 0; on error, it returns an error number, and no signal is sent.

### ERRORS

EINVAL An invalid signal was specified.

### ATTRIBUTES

For an explanation of the terms used in this section, see attributes(7).

??

?Interface ? Attribute ? Value ?  
??  
?pthread\_kill() ? Thread safety ? MT-Safe ?  
??

CONFORMING TO

POSIX.1-2001, POSIX.1-2008.

NOTES

Signal dispositions are process-wide: if a signal handler is installed, the handler will be invoked in the thread thread, but if the disposition of the signal is "stop", "continue", or "terminate", this action will affect the whole process.

The glibc implementation of pthread\_kill() gives an error (EINVAL) on attempts to send either of the real-time signals used internally by the NPTL threading implementation. See nptl(7) for details.

POSIX.1-2008 recommends that if an implementation detects the use of a thread ID after the end of its lifetime, pthread\_kill() should return the error ESRCH. The glibc implementation returns this error in the cases where an invalid thread ID can be detected. But note also that POSIX says that an attempt to use a thread ID whose lifetime has ended produces undefined behavior, and an attempt to use an invalid thread ID in a call to pthread\_kill() can, for example, cause a segmentation fault.

SEE ALSO

kill(2), sigaction(2), sigpending(2), pthread\_self(3), pthread\_sigmask(3), raise(3), pthreads(7), signal(7)

COLOPHON

This page is part of release 5.05 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.