

NAME

`pthread_kill_other_threads_np` – terminate all other threads in process

SYNOPSIS

```
#include <pthread.h>
```

```
void pthread_kill_other_threads_np(void);
```

DESCRIPTION

`pthread_kill_other_threads_np()` has an effect only in the LinuxThreads threading implementation. On that implementation, calling this function causes the immediate termination of all threads in the application, except the calling thread. The cancellation state and cancellation type of the to-be-terminated threads are ignored, and the cleanup handlers are not called in those threads.

ATTRIBUTES

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

Interface	Attribute	Value
<code>pthread_kill_other_threads_np()</code>	Thread safety	MT-Safe

CONFORMING TO

This function is a nonstandard GNU extension; hence the suffix "_np" (nonportable) in the name.

NOTES

`pthread_kill_other_threads_np()` is intended to be called just before a thread calls `execve(2)` or a similar function. This function is designed to address a limitation in the obsolete LinuxThreads implementation whereby the other threads of an application are not automatically terminated (as POSIX.1-2001 requires) during `execve(2)`.

In the NPTL threading implementation, `pthread_kill_other_threads_np()` exists, but does nothing. (Nothing needs to be done, because the implementation does the right thing during an `execve(2)`.)

SEE ALSO

`execve(2)`, `pthread_cancel(3)`, `pthread_setcancelstate(3)`, `pthread_setcanceltype(3)`, `pthreads(7)`

COLOPHON

This page is part of release 5.02 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/>.