NAME

`thr_kill` — send signal to thread

LIBRARY

Standard C Library (libc, −lc)

SYNOPSIS

```
#include <sys/thr.h>

int
thr_kill(long id, int sig);

int
thr_kill2(pid_t pid, long id, int sig);
```

DESCRIPTION

The `thr_kill`() and `thr_kill2`() system calls allow sending a signal, specified by the *sig* argument, to some threads in a process. For the `thr_kill`() function, signalled threads are always limited to the current process. For the `thr_kill2`() function, the *pid* argument specifies the process with threads to be signalled.

The *id* argument specifies which threads get the signal. If *id* is equal to −1, all threads in the specified process are signalled. Otherwise, only the thread with the thread identifier equal to the argument is signalled.

The *sig* argument defines the delivered signal. It must be a valid signal number or zero. In the latter case no signal is actually sent, and the call is used to verify the liveness of the thread.

The signal is delivered with `siginfo si_code` set to `SI_LWP`.

RETURN VALUES

If successful, `thr_kill`() and `thr_kill2`() will return zero, otherwise −1 is returned, and *errno* is set to indicate the error.

ERRORS

The `thr_kill`() and `thr_kill2`() operations return the following errors:

[`EINVAL`] The *sig* argument is not zero and does not specify valid signal.

[`ESRCH`] The specified process or thread was not found.

Additionally, the `thr_kill2`() may return the following errors:

[`EPERM`] The current process does not have sufficient privilege to check existence or send a signal to the specified process.

SEE ALSO

`kill`(2), `thr_exit`(2), `thr_new`(2), `thr_self`(2), `thr_set_name`(2), `_umtx_op`(2), `pthread_kill`(3), `signal`(3)

STANDARDS

The `thr_kill`() and `thr_kill2`() system calls are non-standard and are used by the 1:1 Threading Library (libthr, −lthr) to implement IEEE Std 1003.1-2001 (“POSIX.1”) `pthread`(3) functionality.