

NAME

thr_wake - wake up the suspended thread

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

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

int

```
thr_wake(long id);
```

DESCRIPTION

This function is intended for implementing threading. Normal applications should use **pthread_cond_timedwait(3)** together with **pthread_cond_broadcast(3)** for typical safe suspension with cooperation of the thread being suspended, or **pthread_suspend_np(3)** and **pthread_resume_np(3)** in some specific situations, instead.

Passing the thread identifier of the calling thread (see **thr_self(2)**) to **thr_wake()** sets a thread's flag to cause the next signal-interruptible sleep of that thread in the kernel to fail immediately with the **EINTR** error. The flag is cleared by an interruptible sleep attempt or by a call to **thr_suspend(2)**. This is used by the system threading library to implement cancellation.

If *id* is not equal to the current thread identifier, the specified thread is woken up if suspended by the **thr_suspend(2)** system call. If the thread is not suspended at the time of the **thr_wake** call, the wake is remembered and the next attempt of the thread to suspend itself with the **thr_suspend(2)** results in immediate return with success. Only one wake is remembered.

RETURN VALUES

The **thr_wake()** function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **thr_wake()** operation returns these errors:

[ESRCH]	The specified thread was not found or does not belong to the process of the calling thread.
---------	---

SEE ALSO

ps(1), **thr_self(2)**, **thr_suspend(2)**, **pthread_cancel(3)**, **pthread_resume_np(3)**, **pthread_suspend_np(3)**

STANDARDS

The **thr_suspend()** system call is non-standard and is used by 1:1 Threading Library (libthr, -lthr) to implement IEEE Std 1003.1-2001 ("POSIX.1") pthread(3) functionality.

HISTORY

The **thr_suspend()** system call first appeared in FreeBSD 5.2.