NAME

set_tid_address − set pointer to thread ID

SYNOPSIS

#include <linux/unistd.h>

long set_tid_address(int * *tidptr* **);**

DESCRIPTION

The kernel keeps for each process two values called *set_child_tid* and *clear_child_tid* that are NULL by default.

set_child_tid

If a process is started using **clone**(2) with the **CLONE_CHILD_SETTID** flag, *set_child_tid* is set to *child_tidptr*, the fifth argument of that system call.

When *set_child_tid* is set, the very first thing the new process does is writing its PID at this address.

clear_child_tid

If a process is started using **clone**(2) with the **CLONE_CHILD_CLEARTID** flag, *clear_child_tid* is set to *child_tidptr*, the fifth argument of that system call.

The system call **set_tid_address**() sets the *clear_child_tid* value for the calling process to *tidptr*.

When *clear_child_tid* is set, and the process exits, and the process was sharing memory with other processes or threads, then 0 is written at this address, and a *futex(child_tidptr, FUTEX_WAKE, 1, NULL, NULL, 0);* call is done. (That is, wake a single process waiting on this futex.) Errors are ignored.

RETURN VALUE

set_tid_address() always returns the PID of the calling process.

ERRORS

set_tid_address() always succeeds.

VERSIONS

This call is present since Linux 2.5.48. Details as given here are valid since Linux 2.5.49.

CONFORMING TO

This system call is Linux-specific.

SEE ALSO

clone(2), **futex**(2)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.