



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'set\_tid\_address.2' command**

### **\$ man set\_tid\_address.2**

SET\_TID\_ADDRESS(2)      Linux Programmer's Manual      SET\_TID\_ADDRESS(2)

#### NAME

set\_tid\_address - set pointer to thread ID

#### SYNOPSIS

```
#include <linux/unistd.h>
```

```
pid_t set_tid_address(int *tidptr);
```

Note: There is no glibc wrapper for this system call; see NOTES.

#### DESCRIPTION

For each thread, the kernel maintains two attributes (addresses) called `set_child_tid` and `clear_child_tid`. These two attributes contain the value `NULL` by default.

##### set\_child\_tid

If a thread is started using `clone(2)` with the `CLONE_CHILD_SET?`

`TID` flag, `set_child_tid` is set to the value passed in the `ctid` argument of that system call.

When `set_child_tid` is set, the very first thing the new thread does is to write its thread ID at this address.

##### clear\_child\_tid

If a thread is started using `clone(2)` with the `CLONE_CHILD_CLEAR` flag, `clear_child_tid` is set to the value passed in the `ctid` argument of that system call.

The system call `set_tid_address()` sets the `clear_child_tid` value for the calling thread to `tidptr`.

When a thread whose `clear_child_tid` is not `NULL` terminates, then, if the thread is sharing memory with other threads, then 0 is written at the address specified in `clear_child_tid` and the kernel performs the following operation:

```
futex(clear_child_tid, FUTEX_WAKE, 1, NULL, NULL, 0);
```

The effect of this operation is to wake a single thread that is performing a `futex` wait on the memory location. Errors from the `futex` wake operation are ignored.

## RETURN VALUE

`set_tid_address()` always returns the caller's thread ID.

## 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.

## NOTES

Glibc does not provide a wrapper for this system call; call it using `syscall(2)`.

## SEE ALSO

`clone(2)`, `futex(2)`, `gettid(2)`

## COLOPHON

This page is part of release 5.10 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/>.