NAME

gettid − get thread identification

SYNOPSIS

#include <sys/types.h>

pid_t gettid(void);

DESCRIPTION

gettid() returns the caller's thread ID (TID). In a single-threaded process, the thread ID is equal to the process ID (PID, as returned by **getpid**(2)). In a multithreaded process, all threads have the same PID, but each one has a unique TID. For further details, see the discussion of **CLONE_THREAD** in **clone**(2).

RETURN VALUE

On success, returns the thread ID of the calling thread.

ERRORS

This call is always successful.

VERSIONS

The **gettid**() system call first appeared on Linux in kernel 2.4.11. Library support was added in glibc 2.30. (Earlier glibc versions did not provide a wrapper for this system call, necessitating the use of **syscall**(2).)

CONFORMING TO

gettid() is Linux-specific and should not be used in programs that are intended to be portable.

NOTES

The thread ID returned by this call is not the same thing as a POSIX thread ID (i.e., the opaque value returned by **pthread_self**(3)).

In a new thread group created by a **clone**(2) call that does not specify the **CLONE_THREAD** flag (or, equivalently, a new process created by **fork**(2)), the new process is a thread group leader, and its thread group ID (the value returned by **getpid**(2)) is the same as its thread ID (the value returned by **gettid**()).

SEE ALSO

capget(2), **clone**(2), **fcntl**(2), **fork**(2), **getpid**(2), **get_robust_list**(2), **ioprio_set**(2), **perf_event_open**(2), **sched_setaffinity**(2), **sched_setparam**(2), **sched_setscheduler**(2), **tgkill**(2), **timer_create**(2)

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/.