## NAME

sched_setaffinity, sched_getaffinity – set and get a process's CPU affinity mask

## SYNOPSIS

**#define _GNU_SOURCE**
**#include <sched.h>**

**int sched_setaffinity(pid_t** *pid***, size_t** *cpusetsize***,**
         **cpu_set_t \****mask***);**

**int sched_getaffinity(pid_t** *pid***, size_t** *cpusetsize***,**
         **cpu_set_t \****mask***);**

## DESCRIPTION

A process's CPU affinity mask determines the set of CPUs on which it is eligible to run. On a multiprocessor system, setting the CPU affinity mask can be used to obtain performance benefits. For example, by dedicating one CPU to a particular process (i.e., setting the affinity mask of that process to specify a single CPU, and setting the affinity mask of all other processes to exclude that CPU), it is possible to ensure maximum execution speed for that process. Restricting a process to run on a single CPU also avoids the performance cost caused by the cache invalidation that occurs when a process ceases to execute on one CPU and then recommences execution on a different CPU.

A CPU affinity mask is represented by the *cpu_set_t* structure, a "CPU set", pointed to by *mask*. A set of macros for manipulating CPU sets is described in **CPU_SET**(3).

**sched_setaffinity**() sets the CPU affinity mask of the process whose ID is *pid* to the value specified by *mask*. If *pid* is zero, then the calling process is used. The argument *cpusetsize* is the length (in bytes) of the data pointed to by *mask*. Normally this argument would be specified as *sizeof(cpu_set_t)*.

If the process specified by *pid* is not currently running on one of the CPUs specified in *mask*, then that process is migrated to one of the CPUs specified in *mask*.

**sched_getaffinity**() writes the affinity mask of the process whose ID is *pid* into the *cpu_set_t* structure pointed to by *mask*. The *cpusetsize* argument specifies the size (in bytes) of *mask*. If *pid* is zero, then the mask of the calling process is returned.

## RETURN VALUE

On success, **sched_setaffinity**() and **sched_getaffinity**() return 0. On error, −1 is returned, and *errno* is set appropriately.

## ERRORS

**EFAULT**

A supplied memory address was invalid.

**EINVAL**

The affinity bit mask *mask* contains no processors that are currently physically on the system and permitted to the process according to any restrictions that may be imposed by the "cpuset" mechanism described in **cpuset**(7).

**EINVAL**

(**sched_getaffinity**() and, in kernels before 2.6.9, **sched_setaffinity**()) *cpusetsize* is smaller than the size of the affinity mask used by the kernel.

**EPERM**

(**sched_setaffinity**()) The calling process does not have appropriate privileges. The caller needs an effective user ID equal to the user ID or effective user ID of the process identified by *pid*, or it must possess the **CAP_SYS_NICE** capability.

**ESRCH**
The process whose ID is *pid* could not be found.

**VERSIONS**
The CPU affinity system calls were introduced in Linux kernel 2.5.8. The system call wrappers were introduced in glibc 2.3. Initially, the glibc interfaces included a *cpusetsize* argument, typed as *unsigned int*. In glibc 2.3.3, the *cpusetsize* argument was removed, but was then restored in glibc 2.3.4, with type *size_t*.

**CONFORMING TO**
These system calls are Linux-specific.

**NOTES**
After a call to **sched_setaffinity**(), the set of CPUs on which the process will actually run is the intersection of the set specified in the *mask* argument and the set of CPUs actually present on the system. The system may further restrict the set of CPUs on which the process runs if the "cpuset" mechanism described in **cpuset**(7) is being used. These restrictions on the actual set of CPUs on which the process will run are silently imposed by the kernel.

**sched_setscheduler**(2) has a description of the Linux scheduling scheme.

The affinity mask is actually a per-thread attribute that can be adjusted independently for each of the threads in a thread group. The value returned from a call to **gettid**(2) can be passed in the argument *pid*. Specifying *pid* as 0 will set the attribute for the calling thread, and passing the value returned from a call to **getpid**(2) will set the attribute for the main thread of the thread group. (If you are using the POSIX threads API, then use **pthread_setaffinity_np (3)** instead of **sched_setaffinity**().)

A child created via **fork**(2) inherits its parent's CPU affinity mask. The affinity mask is preserved across an **execve**(2).

This manual page describes the glibc interface for the CPU affinity calls. The actual system call interface is slightly different, with the *mask* being typed as *unsigned long \**, reflecting the fact that the underlying implementation of CPU sets is a simple bit mask. On success, the raw **sched_getaffinity**() system call returns the size (in bytes) of the *cpumask_t* data type that is used internally by the kernel to represent the CPU set bit mask.

**SEE ALSO**
**clone**(2), **getcpu**(2), **getpriority**(2), **gettid**(2), **nice**(2), **sched_get_priority_max**(2), **sched_get_priority_min**(2), **sched_getscheduler**(2), **sched_setscheduler**(2), **setpriority**(2), **CPU_SET**(3), **sched_getcpu**(3), **capabilities**(7), **pthread_setaffinity_np**(3), **cpuset**(7)

**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/.