

**NAME**

**cpuset\_getaffinity**, **cpuset\_setaffinity** - manage CPU affinity

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

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

*int*

```
cpuset_getaffinity(cpulevel_t level, cpuwhich_t which, id_t id, size_t setsize, cpuset_t *mask);
```

*int*

```
cpuset_setaffinity(cpulevel_t level, cpuwhich_t which, id_t id, size_t setsize, const cpuset_t *mask);
```

**DESCRIPTION**

**cpuset\_getaffinity()** and **cpuset\_setaffinity()** allow the manipulation of sets of CPUs available to processes, threads, interrupts, jails and other resources. These functions may manipulate sets of CPUs that contain many processes or per-object anonymous masks that effect only a single object.

The valid values for the *level* and *which* arguments are documented in `cpuset(2)`. These arguments specify which object and which set of the object we are referring to. Not all possible combinations are valid. For example, only processes may belong to a numbered set accessed by a *level* argument of `CPU_LEVEL_CPUSET`. All resources, however, have a mask which may be manipulated with `CPU_LEVEL_WHICH`.

Masks of type *cpuset\_t* are composed using the `CPU_SET` macros. If the user-supplied mask is not large enough to fit all of the matching CPUs, **cpuset\_getaffinity()** fails with `ERANGE`. Calls to **cpuset\_setaffinity()** tolerate masks of any size with no restrictions. The kernel uses the meaningful part of the mask, where the upper bound is the maximum CPU id present in the system. If bits for non-existing CPUs are set, calls to **cpuset\_setaffinity()** fail with `EINVAL`.

The supplied mask should have a size of *setsize* bytes. This size is usually provided by calling `sizeof(mask)` which is ultimately determined by the value of `CPU_SETSIZE` as defined in `<sys/cpuset.h>`.

**cpuset\_getaffinity()** retrieves the mask from the object specified by *level*, *which* and *id* and stores it in the space provided by *mask*.

**cpuset\_setaffinity()** attempts to set the mask for the object specified by *level*, *which* and *id* to the value in *mask*.

## RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

The following error codes may be set in *errno*:

[EINVAL]	The <i>level</i> or <i>which</i> argument was not a valid value.
[EINVAL]	The <i>mask</i> argument specified when calling <b>cpuset_setaffinity()</b> was not a valid value.
[EDEADLK]	The <b>cpuset_setaffinity()</b> call would leave a thread without a valid CPU to run on because the set does not overlap with the thread's anonymous mask.
[EFAULT]	The mask pointer passed was invalid.
[ESRCH]	The object specified by the <i>id</i> and <i>which</i> arguments could not be found.
[ERANGE]	The <i>cpuset_size</i> was smaller than needed to fit all of the matching CPUs.
[EPERM]	The calling process did not have the credentials required to complete the operation.
[ECAPMODE]	The calling process attempted to act on a process other than itself, while in capability mode. See <b>capsicum(4)</b> .

## SEE ALSO

**cpuset(1)**, **cpuset(2)**, **cpuset\_getdomain(2)**, **cpuset\_getid(2)**, **cpuset\_setdomain(2)**, **cpuset\_setid(2)**, **pthread\_affinity\_np(3)**, **pthread\_attr\_affinity\_np(3)**, **capsicum(4)**, **cpuset(9)**

## HISTORY

The **cpuset\_getaffinity** family of system calls first appeared in FreeBSD 7.1.

## AUTHORS

Jeffrey Roberson <[jeff@FreeBSD.org](mailto:jeff@FreeBSD.org)>