NAME

getcpu − determine CPU and NUMA node on which the calling thread is running

SYNOPSIS

#include <linux/getcpu.h>

int getcpu(unsigned ***_cpu_, unsigned ***_node_**, struct getcpu_cache ***_tcache_**);**

DESCRIPTION

The **getcpu**() system call identifies the processor and node on which the calling thread or process is currently running and writes them into the integers pointed to by the *cpu* and *node* arguments. The processor is a unique small integer identifying a CPU. The node is a unique small identifier identifying a NUMA node. When either *cpu* or *node* is NULL nothing is written to the respective pointer.

The third argument to this system call is nowadays unused.

The information placed in *cpu* is only guaranteed to be current at the time of the call: unless the CPU affinity has been fixed using **sched_setaffinity**(2), the kernel might change the CPU at any time. (Normally this does not happen because the scheduler tries to minimize movements between CPUs to keep caches hot, but it is possible.) The caller must be prepared to handle the situation when *cpu* and *node* are no longer the current CPU and node.

VERSIONS

getcpu() was added in kernel 2.6.19 for x86_64 and i386.

CONFORMING TO

getcpu() is Linux specific.

NOTES

Linux makes a best effort to make this call as fast possible. The intention of **getcpu**() is to allow programs to make optimizations with per-CPU data or for NUMA optimization.

Glibc does not provide a wrapper for this system call; call it using **syscall**(2); or use **sched_getcpu**(3) instead.

The *tcache* argument is unused since Linux 2.6.24. In earlier kernels, if this argument was non-NULL, then it specified a pointer to a caller-allocated buffer in thread-local storage that was used to provide a caching mechanism for **getcpu**(). Use of the cache could speed **getcpu**() calls, at the cost that there was a very small chance that the returned information would be out of date. The caching mechanism was considered to cause problems when migrating threads between CPUs, and so the argument is now ignored.

SEE ALSO

mbind(2), **sched_setaffinity**(2), **set_mempolicy**(2), **sched_getcpu**(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/.