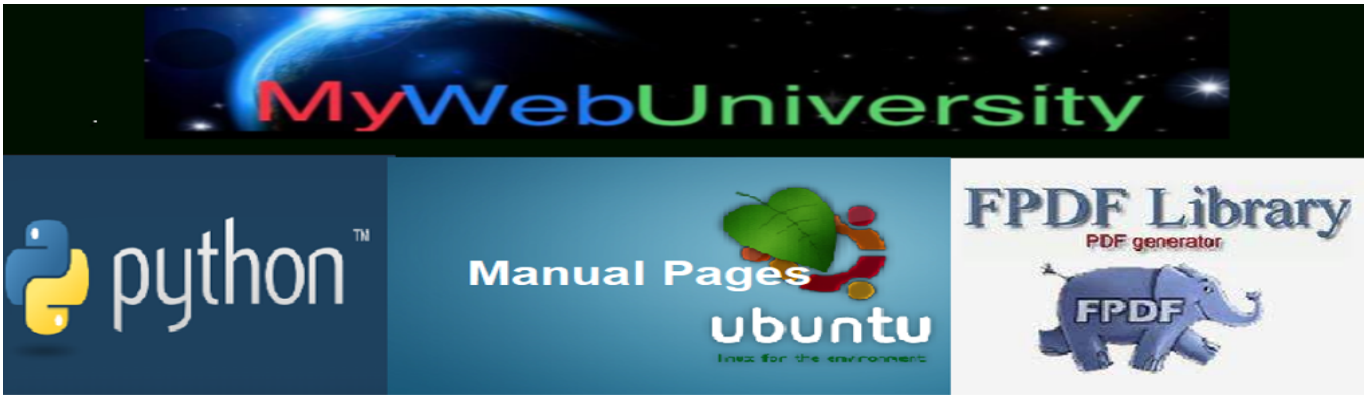

Rocky Enterprise Linux 9.2 Manual Pages on command 'getcpu.2'

$ man getcpu.2

GETCPU(2) Linux Programmer's Manual GETCPU(2)

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, and should be specified as NULL unless portability to Linux 2.6.23 or earlier is required (see NOTES).

The information placed in cpu is guaranteed to be current only 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 mini? mize movements between CPUs to keep caches hot, but it is possible.) The caller must al? low for the possibility that the information returned in cpu and node is no longer current by the time the call returns.

RETURN VALUE

On success, 0 is returned. On error, -1 is returned, and errno is set appropriately.

ERRORS

EFAULT Arguments point outside the calling process's address space.

VERSIONS

getcpu() was added in kernel 2.6.19 for x86-64 and i386. Library support was added in glibc 2.29 (Earlier glibc versions did not provide a wrapper for this system call, neces? sitating the use of syscall(2).)

CONFORMING TO

getcpu() is Linux-specific.

NOTES

Linux makes a best effort to make this call as fast as possible. (On some architectures, this is done via an implementation in the vdso(7).) The intention of getcpu() is to allow programs to make optimizations with per-CPU data or for NUMA optimization.

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 in? formation 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), vdso(7)

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/.