### NAME

kvm\_dpcpu\_setcpu, kvm\_getmaxcpu, kvm\_getpcpu - access per-CPU data

## LIBRARY

Kernel Data Access Library (libkvm, -lkvm)

# SYNOPSIS

#include <sys/param.h>
#include <sys/pcpu.h>
#include <sys/sysctl.h>
#include <kvm.h>

int
kvm\_dpcpu\_setcpu(kvm\_t \*kd, u\_int cpu);

int
kvm\_getmaxcpu(kvm\_t \*kd);

int
kvm\_getncpus(kvm\_t \*kd);

void \*
kvm\_getpcpu(kvm\_t \*kd, int cpu);

ssize\_t
kvm\_read\_zpcpu(kvm\_t \*kd, u\_long base, void \*buf, size\_t size, int cpu);

uint64\_t
kvm\_counter\_u64\_fetch(kvm\_t \*kd, u\_long base);

## DESCRIPTION

The **kvm\_dpcpu\_setcpu**(), **kvm\_getmaxcpu**(), and **kvm\_getpcpu**() functions are used to access the per-CPU data of active processors in the kernel indicated by *kd*. Per-CPU storage comes in two flavours: data stored directly in a *struct pcpu* associated with each CPU, and dynamic per-CPU storage (DPCPU), in which a single kernel symbol refers to different data depending on what CPU it is accessed from.

The kvm\_getmaxcpu() function returns the maximum number of CPUs supported by the kernel.

The **kvm\_getncpus**() function returns the current number of CPUs in the kernel.

The **kvm\_getpcpu**() function returns a buffer holding the per-CPU data for a single CPU. This buffer is described by the *struct pcpu* type. The caller is responsible for releasing the buffer via a call to free(3) when it is no longer needed. If *cpu* is not active, then NULL is returned instead.

The **kvm\_read\_zpcpu**() function is used to obtain private per-CPU copy from a UMA\_ZONE\_PCPU zone(9). It takes *base* argument as base address of an allocation and copyies *size* bytes into *buf* from the part of allocation that is private to *cpu*.

The **kvm\_counter\_u64\_fetch**() function fetches value of a counter(9) pointed by *base* address.

Symbols for dynamic per-CPU data are accessed via kvm\_nlist(3) as with other symbols. **libkvm** maintains a notion of the "current CPU", set by **kvm\_dpcpu\_setcpu**(), which defaults to 0. Once another CPU is selected, kvm\_nlist(3) will return pointers to that data on the appropriate CPU.

### CACHING

**kvm\_getmaxcpu**() and **kvm\_getpcpu**() cache the nlist values for various kernel variables which are reused in successive calls. You may call either function with *kd* set to NULL to clear this cache.

### **RETURN VALUES**

On success, the **kvm\_getmaxcpu**() function returns the maximum number of CPUs supported by the kernel. If an error occurs, it returns -1 instead.

On success, the **kvm\_getpcpu**() function returns a pointer to an allocated buffer or NULL. If an error occurs, it returns -1 instead.

On success, the **kvm\_dpcpu\_setcpu**() call returns 0; if an error occurs, it returns -1 instead.

On success, the **kvm\_read\_zpcpu**() function returns number of bytes copied. If an error occurs, it returns -1 instead.

If any function encounters an error, then an error message may be retrieved via kvm\_geterr(3).

# SEE ALSO

free(3), kvm(3), counter(9), zone(9)