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)