NAME

kvm_open, **kvm_open2**, **kvm_openfiles**, **kvm_close** - initialize kernel virtual memory access

LIBRARY

Kernel Data Access Library (libkvm, -lkvm)

SYNOPSIS

#include <fcntl.h>
#include <kvm.h>

*kvm_t **
kvm_open(*const char *execfile, const char *corefile, const char *swapfile, int flags, const char *errstr*);

*kvm_t **
kvm_open2(*const char *execfile, const char *corefile, int flags, char *errbuf,*
*int (*resolver)(const char *name, kvaddr_t *addr)*);

*kvm_t **
kvm_openfiles(*const char *execfile, const char *corefile, const char *swapfile, int flags, char *errbuf*);

int
kvm_close(*kvm_t *kd*);

DESCRIPTION

The functions **kvm_open**(), **kvm_open2**(), and **kvm_openfiles**() return a descriptor used to access kernel virtual memory via the kvm(3) library routines. Both active kernels and crash dumps are accessible through this interface.

The *execfile* argument is the executable image of the kernel being examined. This file must contain a symbol table. If this argument is NULL, the currently running system is assumed, as determined from getbootfile(3).

The *corefile* argument is the kernel memory device file. It can be either */dev/mem* or a crash dump core generated by savecore(8). If *corefile* is NULL, the default indicated by _PATH_MEM from <*paths.h*> is used. It can also be set to a special value */dev/null* by utilities like ps(1) that do not directly access kernel memory.

The *swapfile* argument is currently unused.

The *flags* argument indicates read/write access as in open(2) and applies only to the core file. Only

O_RDONLY, O_WRONLY, and O_RDWR are permitted.

The **kvm** library provides two different error reporting mechanisms. One provides backward compatibility with the SunOS kvm library, while the other provides an improved error reporting framework. The mechanism used by a descriptor is determined by the function used to open the descriptor.

The **kvm_open**() function is the Sun kvm compatible open call. Here, the *errstr* argument indicates how errors should be handled. If it is NULL, no errors are reported and the application cannot know the specific nature of the failed kvm call. If it is not NULL, errors are printed to stderr with *errstr* prepended to the message, as in perror(3). Normally, the name of the program is used here. The string is assumed to persist at least until the corresponding **kvm_close**() call.

The **kvm_open2**() and **kvm_openfiles**() functions provide BSD style error reporting. Here, error messages are not printed out by the library. Instead, the application obtains the error message corresponding to the most recent kvm library call using **kvm_geterr**() (see kvm_geterr(3)). The results are undefined if the most recent kvm call did not produce an error. Since **kvm_geterr**() requires a kvm descriptor, but the open routines return NULL on failure, **kvm_geterr**() cannot be used to get the error message if open fails. Thus, **kvm_open2**() and **kvm_openfiles**() will place any error message in the *errbuf* argument. This buffer should be _POSIX2_LINE_MAX characters large (from <limits.h>).

The *resolver* argument points to a function used by the **kvm** library to map symbol names to kernel virtual addresses. When the *resolver* function is called, *name* specifies the requested symbol name. If the function is able to resolve the name to an address, the address should be set in *addr* and the function should return zero. If the function is not able to resolve the name to an address, it should return a non-zero value. When opening a native kernel image, *resolver* may be set to NULL to use an internal function to resolve symbol names. Non-native kernel images (such as when cross-debugging a crash dump) require a valid *resolver*.

RETURN VALUES

The **kvm_open**(), **kvm_open2**(), and **kvm_openfiles**() functions return a descriptor to be used in all subsequent kvm library calls. The library is fully re-entrant. On failure, NULL is returned, in which case **kvm_open2**() and **kvm_openfiles**() write the error message into *errbuf*.

The **kvm_close**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **kvm_close**() function may fail and set the global variable *errno* for any of the errors specified for close(2).

The **kvm_close**() function may also fail and set *errno* if:

[EINVAL] The value passed via *kd* was NULL.

SEE ALSO

close(2), open(2), kvm(3), kvm_getargv(3), kvm_getenvv(3), kvm_geterr(3), kvm_getprocs(3), kvm_native(3), kvm_nlist(3), kvm_read(3), kvm_write(3), kmem(4), mem(4)

BUGS

There should not be three open calls. The ill-defined error semantics of the Sun library and the desire to have a backward-compatible library for BSD left little choice.

HISTORY

The **kvm_open2**() function first appeared in FreeBSD 11.0.