NAME

ksyms - kernel symbol table interface

SYNOPSIS

device ksyms

DESCRIPTION

The */dev/ksyms* character device provides a read-only interface to a snapshot of the kernel symbol table. The in-kernel symbol manager is designed to be able to handle many types of symbols tables, however, only elf(5) symbol tables are supported by this device. The ELF format image contains two sections: a symbol table and a corresponding string table.

Symbol Table
: The SYMTAB section contains the symbol table entries present in the current running kernel, including the symbol table entries of any loaded modules. The symbols are ordered by the kernel module load time starting with kernel file symbols first, followed by the first loaded module's symbols and so on.

String Table
: The STRTAB section contains the symbol name strings from the kernel and any loaded modules that the symbol table entries reference.

Elf formatted symbol table data read from the */dev/ksyms* file represents the state of the kernel at the time when the device is opened. Since */dev/ksyms* has no text or data, most of the fields are initialized to NULL. The **ksyms** driver does not block the loading or unloading of modules into the kernel while the */dev/ksyms* file is open but may contain stale data.

FILES

/dev/ksyms

ERRORS

An open(2) of */dev/ksyms* will fail if:

[EBUSY] The device is already open. A process must close */dev/ksyms* before it can be opened again.

[ENOMEM] There is a resource shortage in the kernel.

[ENXIO] The driver was unsuccessful in creating a snapshot of the kernel symbol table. This may occur if the kernel was in the process of loading or unloading a module.

SEE ALSO

nlist(3), elf(5), kldload(8)

HISTORY

A **ksyms** device exists in many different operating systems. This implementation is similar in function to the Solaris and NetBSD **ksyms** driver.

The **ksyms** driver first appeared in FreeBSD 8.0 to support lockstat(1).

AUTHORS

The **ksyms** driver was written by Stacey Son <*sson@FreeBSD.org*>.

BUGS

Because files can be dynamically linked into the kernel at any time the symbol information can vary. When you open the */dev/ksyms* file, you have access to an ELF image which represents a snapshot of the state of the kernel symbol information at that instant in time. Keeping the device open does not block the loading or unloading of kernel modules. To get a new snapshot you must close and re-open the device.

A process is only allowed to open the */dev/ksyms* file once at a time. The process must close the */dev/ksyms* before it is allowed to open it again.