NAME

ksyms – kernel symbols

SYNOPSIS

/dev/ksyms

DESCRIPTION

The file **/dev/ksyms** is a character special file that allows read-only access to an ELF format image containing two sections: a symbol table and a corresponding string table. The contents of the symbol table reflect the symbol state of the currently running kernel. You can determine the size of the image with the **fstat**( ) system call. The recommended method for accessing the **/dev/ksyms** file is by using the ELF access library. See **elf**(3E) for details. If you are unfamiliar with the ELF format, consult the **a.out**(4) manual page.

/dev/ksyms is an executable for the processor on which you are accessing it. It contains ELF program headers which describe the text and data segment(s) in kernel memory. Since **/dev/ksyms** has no text or data, the fields specific to file attributes are initialized to NULL. The remaining fields describe the text or data segment(s) in kernel memory.

Symbol table The SYMTAB section contains the symbol table entries present in the currently running kernel. This section is ordered as defined by the ELF definition with locally-defined symbols first, followed by globally-defined symbols. Within symbol type, the symbols are ordered by kernel module load time. For example, the kernel file symbols are first, followed by the first module’s symbols, and so on, ending with the symbols from the last module loaded.

The section header index (st_shndx) field of each symbol entry in the symbol table is set to SHN_ABS, because any necessary symbol relocations are performed by the kernel link editor at module load time.

String table The STRTAB section contains the symbol name strings that the symbol table entries reference.

SEE ALSO

kernel(1M), **mmap**(2), **stat**(2), **elf**(3E), **kvm_open**(3K), **a.out**(4), **mem**(7D)

WARNINGS

The kernel is dynamically configured. It loads kernel modules when necessary. Because of this aspect of the system, the symbol information present in the running system can vary from time to time, as kernel modules are loaded and unloaded.

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. While the **/dev/ksyms** file remains open, kernel module autounloading is disabled, so that you are protected from the possibility of acquiring stale symbol data. Note that new modules can still be loaded, however. If kernel modules are loaded while you have the **/dev/ksyms** file open, the snapshot held by you will not be updated. In order to have access to the symbol information of the newly loaded modules, you must first close and then reopen the **/dev/ksyms** file. Be aware that the size of the **/dev/ksyms** file will have changed. You will need to use the **fstat( )** function (see **stat**(2)) to determine the new size of the file.

Avoid keeping the **/dev/ksyms** file open for extended periods of time, either by using **kvm_open**(3K) of the default namelist file or with a direct open. There are two reasons why you should not hold **/dev/ksyms** open. First, the system's ability to dynamic configure itself is partially disabled by the locking down of loaded modules. Second, the snapshot of symbol information held by you will not reflect the symbol information of modules loaded after your initial open of **/dev/ksyms**.

Note that the **ksyms** driver is a loadable module, and that the kernel driver modules are only loaded during an open system call. Thus it is possible to run **stat**(2) on the **/dev/ksyms** file without causing the **ksyms** driver to be loaded. In this case, the file size will appear to be zero. A workaround for this behavior is to first open the **/dev/ksyms** file, causing the **ksyms** driver to be loaded (if necessary). You can then use the file descriptor from this open in a **fstat**( ) system call to get the file's size.

NOTES

The kernel virtual memory access library (**libkvm**) routines use **/dev/ksyms** as the default namelist file. See **kvm_open**(3K) for more details.

The **/dev/ksyms** ELF image can be mapped into a process's address space. See **mmap**(2) for details.