NAME

get_kernel_syms − retrieve exported kernel and module symbols

SYNOPSIS

#include <linux/module.h>

int get_kernel_syms(struct kernel_sym ****table*);**

DESCRIPTION

If *table* is NULL, **get_kernel_syms**() returns the number of symbols available for query. Otherwise it fills in a table of structures:

```
struct kernel_sym {
    unsigned long value;
    char          name[60];
};
```

The symbols are interspersed with magic symbols of the form #*module-name* with the kernel having an empty name. The value associated with a symbol of this form is the address at which the module is loaded.

The symbols exported from each module follow their magic module tag and the modules are returned in the reverse of the order in which they were loaded.

RETURN VALUE

On success, returns the number of symbols copied to *table*. On error, −1 is returned and *errno* is set appropriately.

ERRORS

There is only one possible error return:

ENOSYS
  get_kernel_syms() is not supported in this version of the kernel.

VERSIONS

This system call is only present on Linux up until kernel 2.4; it was removed in Linux 2.6.

CONFORMING TO

get_kernel_syms() is Linux-specific.

BUGS

There is no way to indicate the size of the buffer allocated for *table*. If symbols have been added to the kernel since the program queried for the symbol table size, memory will be corrupted.

The length of exported symbol names is limited to 59 characters.

Because of these limitations, this system call is deprecated in favor of **query_module**(2) (which is itself nowadays deprecated in favor of other interfaces described on its manual page).

SEE ALSO

create_module(2), **delete_module**(2), **init_module**(2), **query_module**(2)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.