

**NAME**

query\_module – query the kernel for various bits pertaining to modules

**SYNOPSIS**

```
#include <linux/module.h>
```

```
int query_module(const char *name, int which, void *buf,
                 size_t bufsize, size_t *ret);
```

*Note:* No declaration of this system call is provided in glibc headers; see NOTES.

**DESCRIPTION**

*Note:* This system call is present only in kernels before Linux 2.6.

**query\_module()** requests information from the kernel about loadable modules. The returned information is placed in the buffer pointed to by *buf*. The caller must specify the size of *buf* in *bufsize*. The precise nature and format of the returned information depend on the operation specified by *which*. Some operations require *name* to identify a currently loaded module, some allow *name* to be NULL, indicating the kernel proper.

The following values can be specified for *which*:

- 0** Returns success, if the kernel supports **query\_module()**. Used to probe for availability of the system call.

**QM\_MODULES**

Returns the names of all loaded modules. The returned buffer consists of a sequence of null-terminated strings; *ret* is set to the number of modules.

**QM\_DEPS**

Returns the names of all modules used by the indicated module. The returned buffer consists of a sequence of null-terminated strings; *ret* is set to the number of modules.

**QM\_REFS**

Returns the names of all modules using the indicated module. This is the inverse of **QM\_DEPS**. The returned buffer consists of a sequence of null-terminated strings; *ret* is set to the number of modules.

**QM\_SYMBOLS**

Returns the symbols and values exported by the kernel or the indicated module. The returned buffer is an array of structures of the following form

```
struct module_symbol {
    unsigned long value;
    unsigned long name;
};
```

followed by null-terminated strings. The value of *name* is the character offset of the string relative to the start of *buf*; *ret* is set to the number of symbols.

**QM\_INFO**

Returns miscellaneous information about the indicated module. The output buffer format is:

```
struct module_info {
    unsigned long address;
    unsigned long size;
    unsigned long flags;
};
```

where *address* is the kernel address at which the module resides, *size* is the size of the module in bytes, and *flags* is a mask of **MOD\_RUNNING**, **MOD\_AUTOCLEAN**, and so on, that indicates the current status of the module (see the Linux kernel source file *include/linux/module.h*). *ret* is set to the size of the *module\_info* structure.

## RETURN VALUE

On success, zero is returned. On error, `-1` is returned and *errno* is set appropriately.

## ERRORS

### EFAULT

At least one of *name*, *buf*, or *ret* was outside the program's accessible address space.

### EINVAL

Invalid *which*; or *name* is NULL (indicating "the kernel"), but this is not permitted with the specified value of *which*.

### ENOENT

No module by that *name* exists.

### ENOSPC

The buffer size provided was too small. *ret* is set to the minimum size needed.

### ENOSYS

**query\_module()** is not supported in this version of the kernel (e.g., the kernel is version 2.6 or later).

## VERSIONS

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

## CONFORMING TO

**query\_module()** is Linux-specific.

## NOTES

Some of the information that was formerly available via **query\_module()** can be obtained from */proc/modules*, */proc/kallsyms*, and the files under the directory */sys/module*.

The **query\_module()** system call is not supported by glibc. No declaration is provided in glibc headers, but, through a quirk of history, glibc does export an ABI for this system call. Therefore, in order to employ this system call, it is sufficient to manually declare the interface in your code; alternatively, you can invoke the system call using **syscall(2)**.

## SEE ALSO

**create\_module(2)**, **delete\_module(2)**, **get\_kernel\_syms(2)**, **init\_module(2)**, **lsmod(8)**, **modinfo(8)**

## COLOPHON

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.