

NAME

sysfs – get filesystem type information

SYNOPSIS

```
[[deprecated]] int sysfs(int option, const char *fsname);  
[[deprecated]] int sysfs(int option, unsigned int fs_index, char *buf);  
[[deprecated]] int sysfs(int option);
```

DESCRIPTION

Note: if you are looking for information about the **sysfs** filesystem that is normally mounted at */sys*, see **sysfs(5)**.

The (obsolete) **sysfs()** system call returns information about the filesystem types currently present in the kernel. The specific form of the **sysfs()** call and the information returned depends on the *option* in effect:

- 1 Translate the filesystem identifier string *fsname* into a filesystem type index.
- 2 Translate the filesystem type index *fs_index* into a null-terminated filesystem identifier string. This string will be written to the buffer pointed to by *buf*. Make sure that *buf* has enough space to accept the string.
- 3 Return the total number of filesystem types currently present in the kernel.

The numbering of the filesystem type indexes begins with zero.

RETURN VALUE

On success, **sysfs()** returns the filesystem index for option **1**, zero for option **2**, and the number of currently configured filesystems for option **3**. On error, *-1* is returned, and *errno* is set to indicate the error.

ERRORS**EFAULT**

Either *fsname* or *buf* is outside your accessible address space.

EINVAL

fsname is not a valid filesystem type identifier; *fs_index* is out-of-bounds; *option* is invalid.

STANDARDS

SVr4.

NOTES

This System-V derived system call is obsolete; don't use it. On systems with */proc*, the same information can be obtained via */proc*; use that interface instead.

BUGS

There is no libc or glibc support. There is no way to guess how large *buf* should be.

SEE ALSO

proc(5), **sysfs(5)**