

NAME

setfsgid - set group identity used for filesystem checks

SYNOPSIS

```
#include <sys/fsuid.h>

int setfsgid(uid_t fsgid);
```

DESCRIPTION

The system call **setfsgid()** changes the value of the caller's filesystem group ID—the group ID that the Linux kernel uses to check for all accesses to the filesystem. Normally, the value of the filesystem group ID will shadow the value of the effective group ID. In fact, whenever the effective group ID is changed, the filesystem group ID will also be changed to the new value of the effective group ID.

Explicit calls to [setfsuid\(2\)](#) and **setfsgid()** are usually used only by programs such as the Linux NFS server that need to change what user and group ID is used for file access without a corresponding change in the real and effective user and group IDs. A change in the normal user IDs for a program such as the NFS server is a security hole that can expose it to unwanted signals. (But see below.)

setfsgid() will succeed only if the caller is the superuser or if *fsgid* matches either the caller's real group ID, effective group ID, saved set-group-ID, or current the filesystem user ID.

RETURN VALUE

On both success and failure, this call returns the previous filesystem group ID of the caller.

VERSIONS

This system call is present in Linux since version 1.2.

CONFORMING TO

setfsgid() is Linux-specific and should not be used in programs intended to be portable.

NOTES

Note that at the time this system call was introduced, a process could send a signal to a process with the same effective user ID. Today signal permission handling is slightly different. See [setfsuid\(2\)](#) for a discussion of why the use of both [setfsuid\(2\)](#) and **setfsgid()** is nowadays unneeded.

The original Linux **setfsgid()** system call supported only 16-bit group IDs. Subsequently, Linux 2.4 added **setfsgid32()** supporting 32-bit IDs. The glibc **setfsgid()** wrapper function transparently deals with the variation across kernel versions.

C library/kernel differences

In glibc 2.15 and earlier, when the wrapper for this system call determines that the argument can't be passed to the kernel without integer truncation (because the kernel is old and does not support 32-bit group IDs), they will return -1 and set *errno* to **EINVAL** without attempting the system call.

BUGS

No error indications of any kind are returned to the caller, and the fact that both successful and unsuccessful calls return the same value makes it impossible to directly determine whether the call succeeded or failed. Instead, the caller must resort to looking at the return value from a further call such as *setfsgid(-1)* (which will always fail), in order to determine if a preceding call to **setfsgid()** changed the filesystem group ID. At the very least, **EPERM** should be returned when the call fails (because the caller lacks the **CAP_SETGID** capability).

SEE ALSO

[kill\(2\)](#), [setfsuid\(2\)](#), [capabilities\(7\)](#), [credentials\(7\)](#)

COLOPHON

This page is part of release 4.10 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/>.