

# System call

## 13. setgid()

setgid() system call is a fundamental function in Unix-like operating systems (including Linux, macOS, and others) used to change the effective group ID of a running process. In essence, it allows a program to alter the set of permissions it has when accessing files, directories, and other resources.

This system call allows a process to change its effective group ID (GID). If the calling process is the superuser, it can modify the real GID and the saved GID as well. If the calling process is not the superuser, it can only modify the effective GID

```
#include <unistd.h>
```

```
#include <stdio.h>
```

```
#include <errno.h>
```

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

```
#include <grp.h>
```

```
int main() {
```

```
    gid_t new_gid = 1000; // gid_t gid: The new group ID to set for the process.
```

```
    if (setgid(new_gid) == 0) {
```

```
        printf("Successfully set GID to: %d\n", new_gid);
```

```
        gid_t current_gid = getgid();
```

```
        printf("Current GID: %d\n", current_gid);
```

```
    } else {
```

```
        perror("setgid() failed");
```

```
        printf("errno: %d\n", errno); // print errno for more specific error info
```

```
        return 1; // Indicate an error
```

```
    }
```