

NAME

setgroups - set group access list

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

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

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

```
int
```

```
setgroups(int ngroups, const gid_t *gidset);
```

DESCRIPTION

The **setgroups()** system call sets the group access list of the current user process according to the array *gidset*. The *ngroups* argument indicates the number of entries in the array and must be no more than $\{\text{NGROUPS_MAX}\}+1$.

Only the super-user may set a new group list.

The first entry of the group array (*gidset[0]*) is used as the effective group-ID for the process. This entry is over-written when a **setgid** program is run. To avoid losing access to the privileges of the *gidset[0]* entry, it should be duplicated later in the group array. By convention, this happens because the group value indicated in the password file also appears in */etc/group*. The group value in the password file is placed in *gidset[0]* and that value then gets added a second time when the */etc/group* file is scanned to create the group set.

RETURN VALUES

The **setgroups()** function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **setgroups()** system call will fail if:

- | | |
|----------|---|
| [EPERM] | The caller is not the super-user. |
| [EINVAL] | The number specified in the <i>ngroups</i> argument is larger than the $\{\text{NGROUPS_MAX}\}+1$ limit. |
| [EFAULT] | The address specified for <i>gidset</i> is outside the process address space. |

SEE ALSO

getgroups(2), initgroups(3)

HISTORY

The **setgroups()** system call appeared in 4.2BSD.