getgroups - Man Page

get supplementary group IDs

Prolog

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

Synopsis

```
#include <unistd.h>

int getgroups(int gidsetsize, gid_t grouplist[]);
```

Description

The *getgroups*() function shall fill in the array *grouplist* with the current supplementary group IDs of the calling process. It is implementation-defined whether *getgroups*() also returns the effective group ID in the *grouplist* array.

The *gidsetsize* argument specifies the number of elements in the array *grouplist*. The actual number of group IDs stored in the array shall be returned. The values of array entries with indices greater than or equal to the value returned are undefined.

If *gidsetsize* is 0, *getgroups*() shall return the number of group IDs that it would otherwise return without modifying the array pointed to by *grouplist*.

getgroups - Man Page

than or equal to one and less than or equal to the value of {NGROUPS_MAX}+1.

Return Value

Upon successful completion, the number of supplementary group IDs shall be returned. A return value of -1 indicates failure and *errno* shall be set to indicate the error.

Errors

The *getgroups*() function shall fail if:

EINVAL

The *gidsetsize* argument is non-zero and less than the number of group IDs that would have been returned.

The following sections are informative.

Examples

Getting the Supplementary Group IDs of the Calling Process

The following example places the current supplementary group IDs of the calling process into the *group* array.

```
#include <sys/types.h>
#include <unistd.h>
...
gid_t *group;
int nogroups;
long ngroups_max;
```

getgroups - Man Page

```
   ngroups = getgroups(ngroups_max, group);
```

Application Usage

None.

Rationale

The related function *setgroups*() is a privileged operation and therefore is not covered by this volume of POSIX.1-2017.

As implied by the definition of supplementary groups, the effective group ID may appear in the array returned by *getgroups*() or it may be returned only by *getegid*(). Duplication may exist, but the application needs to call *getegid*() to be sure of getting all of the information. Various implementation variations and administrative sequences cause the set of groups appearing in the result of *getgroups*() to vary in order and as to whether the effective group ID is included, even when the set of groups is the same (in the mathematical sense of “set”). (The history of a process and its parents could affect the details of the result.)

Application developers should note that {NGROUPS_MAX} is not necessarily a constant on all implementations.

Future Directions

None.

See Also

getgroups - Man Page

The Base Definitions volume of POSIX.1-2017, <sys_types.h>, <unistd.h>

Copyright

Referenced By

id(1p), unistd.h(0p).

2017 IEEE/The Open Group POSIX Programmer's Manual

getgroups - Man Page

Home Blog About