NAME

getgrnam, getgrnam_r, getgrent, getgrent_r, getgrgid, getgrgid_r, setgrent, endgrent, fgetgrent, fgetgrent_r – get group entry

SYNOPSIS

#include <grp.h>

struct group *getgrnam(const char ****name*);**

**struct group *getgrnam_r(const char ****name,* **struct group ****result,*
char ****buffer,* **int *buflen***);**

struct group *getgrent(void);

struct group *getgrent_r(struct group ****result,* **char ****buffer,* **int *buflen***);**

struct group *getgrgid(gid_t *gid***);**

struct group *getgrgid_r(gid_t *gid,* **struct group ****result,* **char ****buffer,* **int** *buflen***);**

void setgrent(void);

void endgrent(void);

struct group *fgetgrent(FILE ****f*);**

struct group *fgetgrent_r(FILE ****f,* **struct group ****result,* **char ****buffer,* **int *buflen***);**

POSIX

cc [*flag* **...]** *file* **... –D_POSIX_PTHREAD_SEMANTICS [** *library* **...]**

int getgrnam_r(const char ****name,* **struct group ****grp,* **char ****buffer,* **size_t *bufsize,*
struct group *****result*);**

int getgrgid_r(gid_t *gid,* **struct group ****grp,* **char ****buffer,* **size_t** *bufsize,*
struct group *****result*);**

MT-LEVEL

See the subsection ''Reentrant Interfaces'' in the **DESCRIPTION** section of this page.

DESCRIPTION

These functions are used to obtain entries describing user groups. Entries can come from any of the sources for **group** specified in the **/etc/nsswitch.conf** file (see **nsswitch.conf**(4)).

getgrnam( ) searches for an entry with the group name specified by the character string parameter *name*.

getgrgid( ) searches for an entry with the (numeric) group id specified by *gid*.

The functions **setgrent( )**, **getgrent( )**, and **endgrent( )** are used to enumerate group entries from the database. **setgrent( )** sets (or resets) the enumeration to the beginning of the set of group entries. This function should be called before the first call to **getgrent( )**. Calls to **getgrnam( )** and **getgrgid( )** leave the enumeration position in an indeterminate state. Successive calls to **getgrent( )** return either successive entries or NULL, indicating the end of the enumeration.

endgrent( ) may be called to indicate that the caller expects to do no further group entry retrieval operations; the system may then close the group file, deallocate resources it was using, and so forth. It is still allowed, but possibly less efficient, for the process to call more group functions after calling **endgrent( )**.

fgetgrent( ), unlike the other functions above, does not use **nsswitch.conf**; it reads and parses the next line from the stream *f*, which is assumed to have the format of the **group** file (see **group**(4)).

Reentrant Interfaces

The functions **getgrnam( )**, **getgrgid( )**, **getgrent( )**, and **fgetgrent( )** use static storage that is re-used in each call, making them unsafe for multithreaded applications.

The parallel functions **getgrnam_r( )**, **getgrgid_r( )**, **getgrent_r( )**, and **fgetgrent_r( )** provide reentrant interfaces for these operations.

Each reentrant interface performs the same operation as its non-reentrant counterpart, named by removing the ''**_r**'' suffix. The reentrant interfaces, however, use buffers supplied by the caller to store returned results, and are safe for use in both single-threaded and multithreaded applications.

Each reentrant interface takes the same parameters as its non-reentrant counterpart, as well as the following additional parameters. The parameter *result* (or *grp* for the POSIX versions) must be a pointer to a **struct group** structure allocated by the caller. On successful completion, the function returns the group entry in this structure. The parameter *buffer* is a pointer to a buffer supplied by the caller, used as storage space for the group data. All of the pointers within the returned **struct group** *result* (or *grp*) point to data stored within this buffer; see **RETURN VALUES**. The buffer must be large enough to hold all the data associated with the group entry. The parameter *buflen* (or *bufsize* for the POSIX versions) should give the size in bytes of *buffer*. The POSIX versions place a pointer to the modified *grp* structure in the *result* parameter, instead of returning a pointer to this structure.

For enumeration in multithreaded applications, the position within the enumeration is a process-wide property shared by all threads. **setgrent( )** may be used in a multithreaded application but resets the enumeration position for all threads. If multiple threads interleave calls to **getgrent_r( )**, the threads will enumerate disjoint subsets of the group database.

Like their non-reentrant counterparts, **getgrnam_r( )** and **getgrgid_r( )** leave the enumeration position in an indeterminate state.

RETURN VALUES

Group entries are represented by the **struct group** structure defined in **<grp.h>**:

```
struct group {
        char *gr_name;       /* the name of the group */
        char *gr_passwd;     /* the encrypted group password */
        gid_t gr_gid;        /* the numerical group ID */
        char **gr_mem;       /* vector of pointers to member names */
};
```

The functions **getgrnam( )**, **getgrnam_r( )**, **getgrgid( )**, and **getgrgid_r( )** each return a pointer to a **struct group** if they successfully locate the requested entry; otherwise they return NULL. The POSIX functions **getgrnam_r( )** and **getgrgid_r( )** return zero upon success, or the error number in case of failure.

The functions **getgrent( )**, **getgrent_r( )**, **fgetgrent( )**, and **fgetgrent_r( )** each return a pointer to a **struct group** if they successfully enumerate an entry; otherwise they return NULL, indicating the end of the enumeration.

The functions **getgrnam( )**, **getgrgid( )**, **getgrent( )**, and **fgetgrent( )** use static storage, so returned data must be copied before a subsequent call to any of these functions if the data is to be saved.

When the pointer returned by the reentrant functions **getgrnam_r( )**, **getgrgid_r( )**, **getgrent_r( )**, and **fgetgrent_r( )** is non-NULL, it is always equal to the *result* pointer that was supplied by the caller.

ERRORS

The reentrant functions **getgrnam_r( )**, **getgrgid_r( )**, **getgrent_r( )**, and **fgetgrent_r( )** will return NULL and set *errno* to **ERANGE** (or in the case of POSIX functions **getgrnam_r( )** and **getgrgid_r( )** return the **ERANGE** error) if the length of the buffer supplied by caller is not large enough to store the result. See **Intro**(2) for the proper usage and interpretation of *errno* in multithreaded applications.

FILES

/etc/group
/etc/nsswitch.conf

SEE ALSO

getpwnam(3C), **group**(4), **nsswitch.conf**(4), **passwd**(4)

NOTES

Programs that use the interfaces described in this manual page cannot be linked statically since the implementations of these functions employ dynamic loading and linking of shared objects at run time.

When compiling multithreaded applications, see **Intro**(3), *Notes On Multithread Applications*, for information about the use of the **_REENTRANT** flag.

Use of the enumeration interfaces **getgrent( )** and **getgrent_r( )** is discouraged; enumeration is supported for the group file, NIS, and NIS+, but in general is not efficient and may not be supported for all database sources. The semantics of enumeration are discussed further in **nsswitch.conf**(4).

Previous releases allowed the use of ''+'' and ''-'' entries in **/etc/group** to selectively include and exclude entries from NIS. The primary usage of these ''+/-'' entries is superseded by the name service switch, so *the ''+/-'' form may not be supported in future releases*.

If required, the ''+/-'' functionality can still be obtained for NIS by specifying **compat** as the source for **group**.

If the ''+/-'' functionality is required in conjunction with NIS+, specify both **compat** as the source for **group** and **nisplus** as the source for the pseudo-database **group_compat**. See **group**(4), and **nsswitch.conf**(4) for details.

The reentrant interfaces **getgrnam_r( )**, **getgrgid_r( )**, **getgrent_r( )**, and **fgetgrent_r( )** are as specified in POSIX 1003.1c Draft #10.