Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'getgrgid.3p' command

$ man getgrgid.3p

GETGRGID(3P) POSIX Programmer's Manual GETGRGID(3P)

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.

NAME

getgrgid, getgrgid_r ? get group database entry for a group ID

SYNOPSIS

```
#include <grp.h>
struct group *getgrgid(gid_t gid);
int getgrgid_r(gid_t gid, struct group *grp, char *buffer,
    size_t bufsize, struct group **result);
```

DESCRIPTION

The getgrgid() function shall search the group database for an entry with a matching gid.

The getgrgid() function need not be thread-safe.

Applications wishing to check for error situations should set errno to 0 before calling getgrgid(). If getgrgid() returns a null pointer and errno is set to non-zero, an error occurred.

The getgrgid_r() function shall update the group structure pointed to by grp and store a pointer to that structure at the location pointed to by result. The structure shall contain an entry from the group data?

base with a matching gid. Storage referenced by the group structure is allocated from the memory provided with the buffer parameter, which is bufsize bytes in size. A call to sysconf(_SC_GETGR_R_SIZE_MAX) returns either -1 without changing errno or an initial value suggested for the size of this buffer. A null pointer shall be returned at the location pointed to by result on error or if the requested entry is not found.

RETURN VALUE

Upon successful completion, getgrgid() shall return a pointer to a struct group with the structure defined in <grp.h> with a matching en? try if one is found. The getgrgid() function shall return a null pointer if either the requested entry was not found, or an error oc? curred. If the requested entry was not found, errno shall not be changed. On error, errno shall be set to indicate the error.

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getgrent(), getgrgid(), or getgr? nam(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

If successful, the getgrgid_r() function shall return zero; otherwise, an error number shall be returned to indicate the error.

ERRORS

The getgrgid() and getgrgid_r() functions may fail if:

EIO An I/O error has occurred.

EINTR A signal was caught during getgrgid().

EMFILE All file descriptors available to the process are currently open.

ENFILE The maximum allowable number of files is currently open in the system.

The getgrgid_r() function may fail if:

ERANGE Insufficient storage was supplied via buffer and bufsize to con? tain the data to be referenced by the resulting group structure.

The following sections are informative.

EXAMPLES

Note that sysconf(_SC_GETGR_R_SIZE_MAX) may return -1 if there is no hard limit on the size of the buffer needed to store all the groups re? turned. This example shows how an application can allocate a buffer of sufficient size to work with getgrid_r().

```
long int initlen = sysconf(_SC_GETGR_R_SIZE_MAX);
size_t len;
if (initlen == -1)
    /* Default initial length. */
    len = 1024;
else
    len = (size_t) initlen;
struct group result;
struct group *resultp;
char *buffer = malloc(len);
if (buffer == NULL)
    ...handle error...
int e;
while ((e = getgrgid_r(42, &result, buffer, len, &resultp)) == ERANGE)
    {
    size_t newlen = 2 * len;
    if (newlen < len)
        ...handle error...
    len = newlen;
    char *newbuffer = realloc(buffer, len);
    if (newbuffer == NULL)
        ...handle error...
    buffer = newbuffer;
    }
if (e != 0)
    ...handle error...
free (buffer);
```

Finding an Entry in the Group Database

The following example uses getgrgid() to search the group database for a group ID that was previously stored in a stat structure, then prints out the group name if it is found. If the group is not found, the pro? gram prints the numeric value of the group for the entry.

```
#include <sys/types.h>
#include <grp.h>
#include <stdio.h>
...
struct stat statbuf;
struct group *grp;
...
if ((grp = getgrgid(statbuf.st_gid)) != NULL)
    printf(" %-8.8s", grp->gr_name);
else
    printf(" %-8d", statbuf.st_gid);
...
```

APPLICATION USAGE

The getgrgid_r() function is thread-safe and shall return values in a user-supplied buffer instead of possibly using a static data area that may be overwritten by each call.

Portable applications should take into account that it is usual for an implementation to return -1 from sysconf() indicating that there is no maximum for _SC_GETGR_R_SIZE_MAX.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

endgrent(), getgrnam(), sysconf()

The Base Definitions volume of POSIX.1?2017, <grp.h>, <sys_types.h>

COPYRIGHT

from IEEE Std 1003.1-2017, Standard for Information Technology -- Por? table Operating System Interface (POSIX), The Open Group Base Specifi? cations Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at http://www.opengroup.org/unix/online.html .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker? nel.org/doc/man-pages/reporting_bugs.html .