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 'getpwuid.3p' command

$ man getpwuid.3p

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

getpwuid, getpwuid_r ? search user database for a user ID

SYNOPSIS

```
#include <pwd.h>
struct passwd *getpwuid(uid_t uid);
int getpwuid_r(uid_t uid, struct passwd *pwd, char *buffer,
    size_t bufsize, struct passwd **result);
```

DESCRIPTION

The getpwuid() function shall search the user database for an entry with a matching uid.

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

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

The getpwuid_r() function shall update the passwd structure pointed to by pwd and store a pointer to that structure at the location pointed to by result. The structure shall contain an entry from the user database

with a matching uid. Storage referenced by the structure is allocated from the memory provided with the buffer parameter, which is bufsize bytes in size. A call to sysconf(_SC_GETPW_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

The getpwuid() function shall return a pointer to a struct passwd with the structure as defined in <pwd.h> with a matching entry if found. A null pointer shall be returned if the requested entry is not found, or an error occurs. 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 getpwent(), getpwnam(), or getp? wuid(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

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

ERRORS

These functions may fail if:

EIO An I/O error has occurred.

EINTR A signal was caught during getpwuid().

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 getpwuid_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 passwd struc? ture.

The following sections are informative.

EXAMPLES

Note that sysconf(_SC_GETPW_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 getpwuid_r().

```
long int initlen = sysconf(_SC_GETPW_R_SIZE_MAX);
size_t len;
if (initlen == -1)
    /* Default initial length. */
    len = 1024;
else
    len = (size_t) initlen;
struct passwd result;
struct passwd *resultp;
char *buffer = malloc(len);
if (buffer == NULL)
    ...handle error...
int e;
while ((e = getpwuid_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);
```

Getting an Entry for the Root User

The following example gets the user database entry for the user with user ID 0 (root).

```
#include <sys/types.h>
#include <pwd.h>
...
uid_t id = 0;
struct passwd *pwd;
pwd = getpwuid(id);
```

Finding the Name for the Effective User ID

The following example defines pws as a pointer to a structure of type passwd, which is used to store the structure pointer returned by the call to the getpwuid() function. The geteuid() function shall return the effective user ID of the calling process; this is used as the search criteria for the getpwuid() function. The call to getpwuid() shall return a pointer to the structure containing that user ID value.

```
#include <unistd.h>
#include <sys/types.h>
#include <pwd.h>
...
struct passwd *pws;
pws = getpwuid(geteuid());
```

Finding an Entry in the User Database

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

```
#include <sys/types.h>
#include <pwd.h>
#include <stdio.h>
...
struct stat statbuf;
struct passwd *pwd;
```

```
...
if ((pwd = getpwuid(statbuf.st_uid)) != NULL)
    printf(" %-8.8s", pwd->pw_name);
else
    printf(" %-8d", statbuf.st_uid);
```

APPLICATION USAGE

Three names associated with the current process can be determined: get?pwuid(geteuid()) returns the name associated with the effective user ID of the process; getlogin() returns the name associated with the current login activity; and getpwuid(getuid()) returns the name associated with the real user ID of the process.

The getpwuid_r() function is thread-safe and returns 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_GETPW_R_SIZE_MAX.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

getpwnam(), geteuid(), getuid(), getlogin(), sysconf()

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

COPYRIGHT

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 .