

# getlogin - Man Page

*get login name*

## 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>
```

```
char *getlogin(void);  
int getlogin_r(char *name, size_t namesize);
```

## Description

The *getlogin()* function shall return a pointer to a string containing the user name associated by the login activity with the controlling terminal of the current process. If *getlogin()* returns a non-null pointer, then that pointer points to the name that the user logged in under, even if there are several login names with the same user ID.

The *getlogin()* function need not be thread-safe.

The *getlogin\_r()* function shall put the name associated by the login activity with the controlling terminal of the current process in the character array pointed to by *name*. The array is *namesize* characters long and should have space for the name and the terminating null character. The maximum size of the login name is {LOGIN\_NAME\_MAX}.

# getlogin - Man Page

The *getlogin()* and *getlogin\_r()* functions may make use of file descriptors 0, 1, and 2 to find the controlling terminal of the current process, examining each in turn until the terminal is found. If in this case none of these three file descriptors is open to the controlling terminal, these functions may fail. The method used to find the terminal associated with a file descriptor may depend on the file descriptor being open to the actual terminal device, not `/dev/tty`.

## Return Value

Upon successful completion, *getlogin()* shall return a pointer to the login name or a null pointer if the user's login name cannot be found. Otherwise, it shall return a null pointer and set *errno* to indicate the error.

The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to *getlogin()*. The returned pointer and the string content might also be invalidated if the calling thread is terminated.

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

## Errors

These functions may fail if:

### EMFILE

All file descriptors available to the process are currently open.

### ENFILE

# getlogin - Man Page

## ENOTTY

None of the file descriptors 0, 1, or 2 is open to the controlling terminal of the current process.

## ENXIO

The calling process has no controlling terminal.

The `getlogin_r()` function may fail if:

## ERANGE

The value of `namesize` is smaller than the length of the string to be returned including the terminating null character.

*The following sections are informative.*

## Examples

### Getting the User Login Name S

The following example calls the `getlogin()` function to obtain the name of the user associated with the calling process, and passes this information to the `getpwnam()` function to get the associated user database information.

```
#include <unistd.h>
#include <sys/types.h>
#include <pwd.h>
#include <stdio.h>
...
char *lgn;
struct passwd *pw;
```

# getlogin - Man Page

```
}
```

## Application Usage

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

The *getlogin\_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.

## Rationale

The *getlogin()* function returns a pointer to the user's login name. The same user ID may be shared by several login names. If it is desired to get the user database entry that is used during login, the result of *getlogin()* should be used to provide the argument to the *getpwnam()* function. (This might be used to determine the user's login shell, particularly where a single user has multiple login shells with distinct login names, but the same user ID.)

The information provided by the *cuserid()* function, which was originally defined in the POSIX.1-1988 standard and subsequently removed, can be obtained by the following:

```
getpwuid(geteuid())
```

while the information provided by historical implementations of *cuserid()* can be obtained by:

# getlogin - Man Page

The thread-safe version of this function places the user name in a user-supplied buffer and returns a non-zero value if it fails. The non-thread-safe version may return the name in a static data area that may be overwritten by each call.

## Future Directions

None.

## See Also

`getpwnam()`, `getpwuid()`, `geteuid()`, `getuid()`

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

## Copyright

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications 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.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

# getlogin - Man Page

`endgrent(3p)`, `endpwent(3p)`, `getpwuid(3p)`, `logname(1p)`, `unistd.h(0p)`.

2017 IEEE/The Open Group POSIX Programmer's Manual