lchown - Man Page

change the owner and group of a symbolic link

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 lchown(const char *path, uid_t owner, gid_t group);
```

Description

The *lchown*() function shall be equivalent to *chown*(), except in the case where the named file is a symbolic link. In this case, *lchown*() shall change the ownership of the symbolic link file itself, while *chown*() changes the ownership of the file or directory to which the symbolic link refers.

Return Value

Upon successful completion, *lchown*() shall return 0. Otherwise, it shall return -1 and set *errno* to indicate an error.

Errors

lchown - Man Page

Search permission is denied on a component of the path prefix of *path*.

EINVAL

The owner or group ID is not a value supported by the implementation.

ELOOP

A loop exists in symbolic links encountered during resolution of the *path* argument.

ENAMETOOLONG

The length of a component of a pathname is longer than {NAME_MAX}.

ENOENT

A component of *path* does not name an existing file or *path* is an empty string.

ENOTDIR

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, or the *path* argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters and the last pathname component names an existing file that is neither a directory nor a symbolic link to a directory.

EPERM

The effective user ID does not match the owner of the file and the process does not have appropriate privileges.

EROFS

The file resides on a read-only file system.

lchown - Man Page

EIO An I/O error occurred while reading or writing to the file system.

EINTR

A signal was caught during execution of the function.

ELOOP

More than {SYMLOOP_MAX} symbolic links were encountered during resolution of the *path* argument.

ENAMETOOLONG

The length of a pathname exceeds {PATH_MAX}, or pathname resolution of a symbolic link produced an intermediate result with a length that exceeds {PATH_MAX}.

The following sections are informative.

Examples

Changing the Current Owner of a File

The following example shows how to change the ownership of the symbolic link named **/modules/pass1** to the user ID associated with “jones” and the group ID associated with “cnd”.

The numeric value for the user ID is obtained by using the *getpwnam*() function. The numeric value for the group ID is obtained by using the *getgrnam*() function.

```
#include <sys/types.h>
#include <unistd.h>
```

lchown - Man Page

```
struct passwd *pwd;
struct group  *grp;
char          *path = "/modules/pass1";
...
pwd = getpwnam("jones");
grp = getgrnam("cnd");
lchown(path, pwd->pw_uid, grp->gr_gid);
```

Application Usage

On implementations which support symbolic links as directory entries rather than files, *lchown*() may fail.

Rationale

None.

Future Directions

None.

See Also

chown(), symlink()

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

Copyright

lchown - Man Page

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 .

Referenced By

chown(3p), symlink(3p), unistd.h(0p).

2017 IEEE/The Open Group POSIX Programmer's Manual

lchown - Man Page