NAME

chown, lchown, fchown – change owner and group of a file

SYNOPSIS

```
#include <unistd.h>
#include <sys/types.h>
int chown(const char *path, uid_t owner, gid_t group);
int lchown(const char *path, uid_t owner, gid_t group);
int fchown(int fildes, uid_t owner, gid_t group);
```

MT-LEVEL

chown( ) is Async-Signal-Safe

DESCRIPTION

chown( ) sets the owner ID and group ID of the file specified by *path* or referenced by the open file descriptor *fildes* to *owner* and *group* respectively. If *owner* or *group* is specified as –1, **chown( )** does not change the corresponding ID of the file.

The function **lchown( )** sets the owner ID and group ID of the named file just as **chown( )** does, except in the case where the named file is a symbolic link. In this case, **lchown( )** changes the ownership of the symbolic link file itself, while **chown( )** changes the ownership of the file or directory to which the symbolic link refers.

If **chown( )**, **lchown( )**, or **fchown( )** is invoked by a process other than super-user, the set-user-ID and set-group-ID bits of the file mode, **S_ISUID** and **S_ISGID** respectively, are cleared (see **chmod**(2)).

The operating system has a configuration option, {**_POSIX_CHOWN_RESTRICTED**}, to restrict ownership changes for the **chown( )**, **lchown( )**, and **fchown( )** functions. When {**_POSIX_CHOWN_RESTRICTED**} is not in effect, the effective user ID of the process must match the owner of the file or the process must be the super-user to change the ownership of a file. When {**_POSIX_CHOWN_RESTRICTED**} is in effect, the **chown( )**, **lchown( )**, and **fchown( )** functions, for users other than super-user, prevent the owner of the file from changing the owner ID of the file and restrict the change of the group of the file to the list of supplementary group IDs.

Upon successful completion, **chown( )**, **fchown( )** and **lchown( )** mark for update the **st_ctime** field of the file.

RETURN VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of –1 is returned and **errno** is set to indicate the error.

ERRORS

chown( ) and **lchown( )** fail and the owner and group of the named file remain unchanged if one or more of the following are true:

EACCES	Search permission is denied on a component of the path prefix of *path*.
EFAULT	*path* points to an illegal address.
EINTR	A signal was caught during the **chown( )** or **lchown( )** functions.
EINVAL	*group* or *owner* is out of range.

EIO	An I/O error occurred while reading from or writing to the file system.
ELOOP	Too many symbolic links were encountered in translating *path*.
EMULTIHOP	Components of *path* require hopping to multiple remote machines and file system type does not allow it. Too many symbolic links were encountered in translating *path*.
ENAMETOOLONG	The length of the *path* argument exceeds {**PATH_MAX**}, or the length of a *path* component exceeds {**NAME_MAX**} while {**_POSIX_NO_TRUNC**} is in effect.
ENOLINK	*path* points to a remote machine and the link to that machine is no longer active.
ENOENT	Either a component of the path prefix or the file referred to by *path* does not exist or is a null pathname.
ENOTDIR	A component of the path prefix of *path* is not a directory.
EPERM	The effective user ID does not match the owner of the file or the process is not the super-user and {**_POSIX_CHOWN_RESTRICTED**} indicates that such privilege is required.
EROFS	The named file resides on a read-only file system.

fchown() fails and the owner and group of the named file remain unchanged if one or more of the following are true:

EBADF	*fildes* is not an open file descriptor.
EIO	An I/O error occurred while reading from or writing to the file system.
EINTR	A signal was caught during execution of the function.
ENOLINK	*fildes* points to a remote machine and the link to that machine is no longer active.
EINVAL	*group* or *owner* is out of range.
EPERM	The effective user ID does not match the owner of the file or the process is not the super-user and {**_POSIX_CHOWN_RESTRICTED**} indicates that such privilege is required.
EROFS	The named file referred to by *fildes* resides on a read-only file system.

SEE ALSO **chgrp**(1), **chown**(1), **chmod**(2)