NAME

chmod, fchmod – change access permission mode of file

SYNOPSIS

#include <sys/types.h>
#include <sys/stat.h>

int chmod(const char **path*, **mode_t** *mode***);**

int fchmod(int *fildes*, **mode_t** *mode***);**

MT-LEVEL

chmod( ) is Async-Signal-Safe

DESCRIPTION

chmod( ) and **fchmod( )** set the access permission portion of the mode of the file whose name is given by *path* or referenced by the open file descriptor *fildes* to the bit pattern contained in *mode*. Access permission bits are interpreted as follows:

S_ISUID	04000	Set user ID on execution.
S_ISGID	020#0	Set group ID on execution if # is **7**, **5**, **3**, or **1**. Enable mandatory file/record locking if # is **6**, **4**, **2**, or **0**.
S_ISVTX	01000	Save text image after execution.
S_IRWXU	00700	Read, write, execute by owner.
S_IRUSR	00400	Read by owner.
S_IWUSR	00200	Write by owner.
S_IXUSR	00100	Execute (search if a directory) by owner.
S_IRWXG	00070	Read, write, execute by group.
S_IRGRP	00040	Read by group.
S_IWGRP	00020	Write by group.
S_IXGRP	00010	Execute by group.
S_IRWXO	00007	Read, write, execute (search) by others.
S_IROTH	00004	Read by others.
S_IWOTH	00002	Write by others.
S_IXOTH	00001	Execute by others.

Modes are constructed by **OR**'ing the access permission bits.

The effective user ID of the process must match the owner of the file or the process must have the appropriate privilege to change the mode of a file.

If the process is not a privileged process and the file is not a directory, mode bit 01000 (save text image on execution) is cleared.

If neither the process is privileged, nor the file's group is a member of the process's supplementary group list, and the effective group ID of the process does not match the group ID of the file, mode bit 02000 (set group ID on execution) is cleared.

If a directory is writable and has S_ISVTX (the sticky bit) set, files within that directory can be removed or renamed only if one or more of the following is true (see **unlink**(2) and **rename**(2)):

- the user owns the file
- the user owns the directory
- the file is writable by the user
- the user is a privileged user

If a directory has the set group ID bit set, a given file created within that directory will have the same group ID as the directory, if that group ID is part of the group ID set of the process that created the file. Otherwise, the newly created file's group ID will be set to the effective group ID of the creating process.

If the mode bit 02000 (set group ID on execution) is set and the mode bit 00010 (execute or search by group) is not set, mandatory file/record locking will exist on a regular file. This may affect future calls to **open**(2), **creat**(2), **read**(2), and **write**(2) on this file.

Upon successful completion, **chmod( )** and **fchmod( )** 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

chmod( ) will fail and the file mode will be 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 execution of the function.

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.

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.

ENOENT Either a component of the path prefix, or the file referred to by *path* does not exist or is a null pathname.

ENOLINK *fildes* points to a remote machine and the link to that machine is no longer active.

ENOTDIR A component of the prefix of *path* is not a directory.

EPERM	The effective user ID does not match the owner of the file and is not super-user.
EROFS	The file referred to by *path* resides on a read-only file system.

fchmod( ) will fail and the file mode will be unchanged if:

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 **fchmod( )** function.
ENOLINK	*path* points to a remote machine and the link to that machine is no longer active.
EPERM	The effective user ID does not match the owner of the file and the process does not have appropriate privilege.
EROFS	The file referred to by *fildes* resides on a read-only file system.

SEE ALSO

chmod(1), **chown**(2), **creat**(2), **fcntl**(2), **mknod**(2), **open**(2), **read**(2), **rename**(2), **stat**(2), **write**(2), **mkfifo**(3C), **stat**(5)

System Interfaces Guide