

NAME

chmod – change file access permissions

SYNOPSIS

chmod [*OPTION*]... *MODE*[,*MODE*]... *FILE*...

chmod [*OPTION*]... *OCTAL-MODE* *FILE*...

chmod [*OPTION*]... --reference=*RFILE* *FILE*...

DESCRIPTION

This manual page documents the GNU version of **chmod**. **chmod** changes the permissions of each given file according to *mode*, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new permissions.

The format of a symbolic mode is '[ugoa...][[+|=][rwxXstugo...][...][,...]'. Multiple symbolic operations can be given, separated by commas.

A combination of the letters 'ugoa' controls which users' access to the file will be changed: the user who owns it (u), other users in the file's group (g), other users not in the file's group (o), or all users (a). If none of these are given, the effect is as if 'a' were given, but bits that are set in the umask are not affected.

The operator '+' causes the permissions selected to be added to the existing permissions of each file; '-' causes them to be removed; and '=' causes them to be the only permissions that the file has.

The letters 'rwxXstugo' select the new permissions for the affected users: read (r), write (w), execute (or access for directories) (x), execute only if the file is a directory or already has execute permission for some user (X), set user or group ID on execution (s), sticky (t), the permissions granted to the user who owns the file (u), the permissions granted to other users who are members of the file's group (g), and the permissions granted to users that are in neither of the two preceding categories (o).

A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Any omitted digits are assumed to be leading zeros. The first digit selects the set user ID (4) and set group ID (2) and sticky (1) attributes. The second digit selects permissions for the user who owns the file: read (4), write (2), and execute (1); the third selects permissions for other users in the file's group, with the same values; and the fourth for other users not in the file's group, with the same values.

chmod never changes the permissions of symbolic links; the **chmod** system call cannot change their permissions. This is not a problem since the permissions of symbolic links are never used. However, for each symbolic link listed on the command line, **chmod** changes the permissions of the pointed-to file. In contrast, **chmod** ignores symbolic links encountered during recursive directory traversals.

STICKY FILES

On older Unix systems, the sticky bit caused executable files to be hoarded in swap space. This feature is not useful on modern VM systems, and the Linux kernel ignores the sticky bit on files. Other kernels may use the sticky bit on files for system-defined purposes. On some systems, only the superuser can set the sticky bit on files.

STICKY DIRECTORIES

When the sticky bit is set on a directory, files in that directory may be unlinked or renamed only by root or their owner. Without the sticky bit, anyone able to write to the directory can delete or rename files. The sticky bit is commonly found on directories, such as /tmp, that are world-writable.

OPTIONS

Change the mode of each FILE to MODE.

-c, --changes

like verbose but report only when a change is made

-f, --silent, --quiet

suppress most error messages

-v, --verbose

output a diagnostic for every file processed

--reference=*RFILE*
use *RFILE*'s mode instead of *MODE* values

-R, --recursive
change files and directories recursively

--help display this help and exit

--version
output version information and exit

Each *MODE* is one or more of the letters ugoa, one of the symbols += and one or more of the letters rwxXstugo.

AUTHOR

Written by David MacKenzie.

REPORTING BUGS

Report bugs to <bug-fileutils@gnu.org>.

COPYRIGHT

Copyright © 2001 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

SEE ALSO

The full documentation for **chmod** is maintained as a Texinfo manual. If the **info** and **chmod** programs are properly installed at your site, the command

info chmod

should give you access to the complete manual.