homework4

April 6, 2020

Homework #4

[7]:
```
man chmod
```

```
CHMOD(1)                          User Commands                          CHMOD(1)

NAME
       chmod - change file mode bits

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
       file mode bits 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 mode bits.

       The format of a symbolic mode is  [ugoa…][[-+=][perms…]…],  where
       perms  is  either zero or more letters from the set rwxXst, or a single
       letter from the set ugo.  Multiple symbolic modes can be  given,  sepa-
       rated 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 selected file mode bits to be added to the
       existing file mode bits of each file; - causes them to be removed;  and
       =  causes  them  to  be added and causes unmentioned bits to be removed
       except that a directory's unmentioned set user and group  ID  bits  are
       not affected.

       The  letters  rwxXst select file mode bits for the affected users: read
       (r), write (w), execute (or search for directories) (x), execute/search
```

only if the file is a directory or already has execute permission for some user (X), set user or group ID on execution (s), restricted deletion flag or sticky bit (t). Instead of one or more of these letters, you can specify exactly one of the letters ugo: 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. Omitted digits are assumed to be leading zeros. The first digit selects the set user ID (4) and set group ID (2) and restricted deletion or 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.

SETUID AND SETGID BITS

chmod clears the set-group-ID bit of a regular file if the file's group ID does not match the user's effective group ID or one of the user's supplementary group IDs, unless the user has appropriate privileges. Additional restrictions may cause the set-user-ID and set-group-ID bits of MODE or RFILE to be ignored. This behavior depends on the policy and functionality of the underlying chmod system call. When in doubt, check the underlying system behavior.

chmod preserves a directory's set-user-ID and set-group-ID bits unless you explicitly specify otherwise. You can set or clear the bits with symbolic modes like u+s and g-s, and you can set (but not clear) the bits with a numeric mode.

RESTRICTED DELETION FLAG OR STICKY BIT

The restricted deletion flag or sticky bit is a single bit, whose interpretation depends on the file type. For directories, it prevents unprivileged users from removing or renaming a file in the directory unless they own the file or the directory; this is called the restricted deletion flag for the directory, and is commonly found on world-writable directories like /tmp. For regular files on some older systems, the bit saves the program's text image on the swap device so it will load more quickly when run; this is called the sticky bit.

OPTIONS

Change the mode of each FILE to MODE. With --reference, change the mode of each FILE to that of RFILE.

-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

--no-preserve-root
do not treat '/' specially (the default)

--preserve-root
fail to operate recursively on '/'

--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 of the form '[ugoa]*([-+=]([rwxXst]*|[ugo]))+|[-+=][0-7]+'.

AUTHOR

Written by David MacKenzie and Jim Meyering.

REPORTING BUGS

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report chmod translation bugs to <http://translationproject.org/team/>

COPYRIGHT

Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

```
chmod(2)

Full documentation at: <http://www.gnu.org/software/coreutils/chmod>
or available locally via: info '(coreutils) chmod invocation'

GNU coreutils 8.28              January 2018                        CHMOD(1)
```

PRACTICE CHMOD

[8]:
```
pwd
```

```
/home/jupyter-zhannaspace
```

[14]:
```
touch testfile1
```

[15]:
```
cd homework4
```

[16]:
```
pwd
```

```
/home/jupyter-zhannaspace/homework4
```

[17]:
```
touch tst1
```

[19]:
```
ls -l tst1
```

```
-rw-r--r-- 1 jupyter-zhannaspace jupyter-zhannaspace 0 Apr  6 15:18 tst1
```

[23]:
```
chmod u+x tst1 #add execute right to user
```

[24]:
```
ls -l tst1
```

```
-rwxr--r-- 1 jupyter-zhannaspace jupyter-zhannaspace 0 Apr  6 15:18 tst1
```

[28]:
```
chmod g+wx tst1 #add write and execute right to group
```

[29]:
```
ls -l tst1
```

```
-rwxrwxr-- 1 jupyter-zhannaspace jupyter-zhannaspace 0 Apr  6 15:18 tst1
```

[30]:
```
chmod o-r tst1 #remove read right from others
```

[31]:
```
ls -l tst1
```

```
-rwxrwx--- 1 jupyter-zhannaspace jupyter-zhannaspace 0 Apr  6 15:18 tst1
```

[3]:
```
cd homework4
```

[4]:
```
chmod u-wx tst1  #remove write and execute rights from user
```

[5]:
```
ls -l tst1
```

```
-r--rwx--- 1 jupyter-zhannaspace jupyter-zhannaspace 0 Apr  6 15:18 tst1
```

[7]:
```
chmod o+x tst1   # add execute right to others
```

[8]:
```
ls -l tst1
```

```
-r--rwx--x 1 jupyter-zhannaspace jupyter-zhannaspace 0 Apr  6 15:18 tst1
```

0.1 Practicing WildCards

[10]:
```
mkdir test
```

[11]:
```
mkdir fest
```

[12]:
```
mkdir nest
```

[13]:
```
ls
```

```
fest  nest  test  tst1  Untitled.ipynb
```

[14]:
```
ls *est # represents zero or more
```

```
fest:

nest:

test:
```

[22]:
```
touch cow
```

[23]:
```
touch low
```

[24]:
```
touch below
```

[26]:
```
ls *ow # find all files ended "ow"
```

```
below  cow  low
```

[27]:
```
ls ?ow #find files with ow
```

```
cow  low
```

[29]:
```
touch ace.txt
```

[30]:
```
touch bee.txt
```

[31]:
```
touch zeta.txt
```

[33]:
```
ls [a-n]*.txt #find all txt files with a range [a-n]
```

```
ace.txt  bee.txt
```

[34]:
```
ls ??ta.txt
```

```
zeta.txt
```

[35]:
```
ls [m-z]*.txt
```

```
zeta.txt
```

[36]:
```
ls [a-b]*
```

```
ace.txt  bee.txt  below
```

[37]:
```
mkdir Zhanna
```

[38]:
```
cd Zhanna
```

[39]:
```
touch loft.png
```

[40]:
```
touch soft.jpg
```

[41]:
```
touch hoft.png
```

[42]:
```
ls
```

```
hoft.png  loft.png  soft.jpg
```

[43]:
```
ls *.[jp][pn]g  #find all image files
```

```
hoft.png  loft.png  soft.jpg
```

[44]:
```
ls *ft.jpg
```

```
soft.jpg
```

[47]:
```
ls *ft.png
```

```
hoft.png  loft.png
```

[ ]:
```

```