How to change directory permissions in Linux

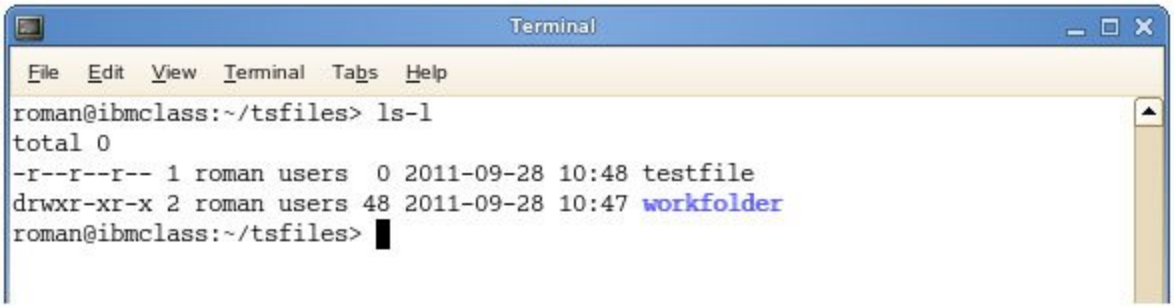
How to change directory permissions in Linux

To change directory permissions in Linux, use the following:

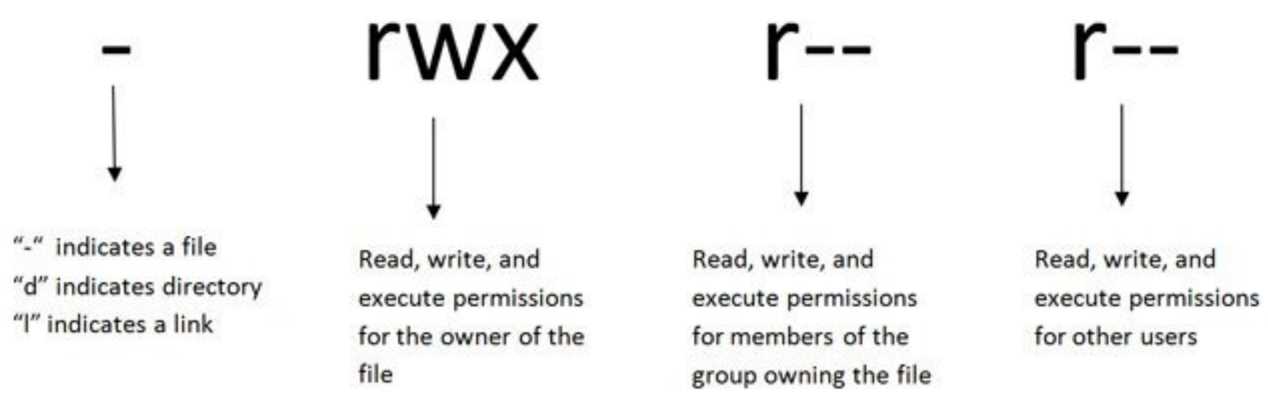
- **chmod +rwx filename** to add permissions.
- **chmod -rwx directoryname** to remove permissions.
- **chmod +x filename** to allow executable permissions.
- **chmod -wx filename** to take out write and executable permissions.

Note that “r” is for read, “w” is for write, and “x” is for execute.

This only changes the permissions for the owner of the file.



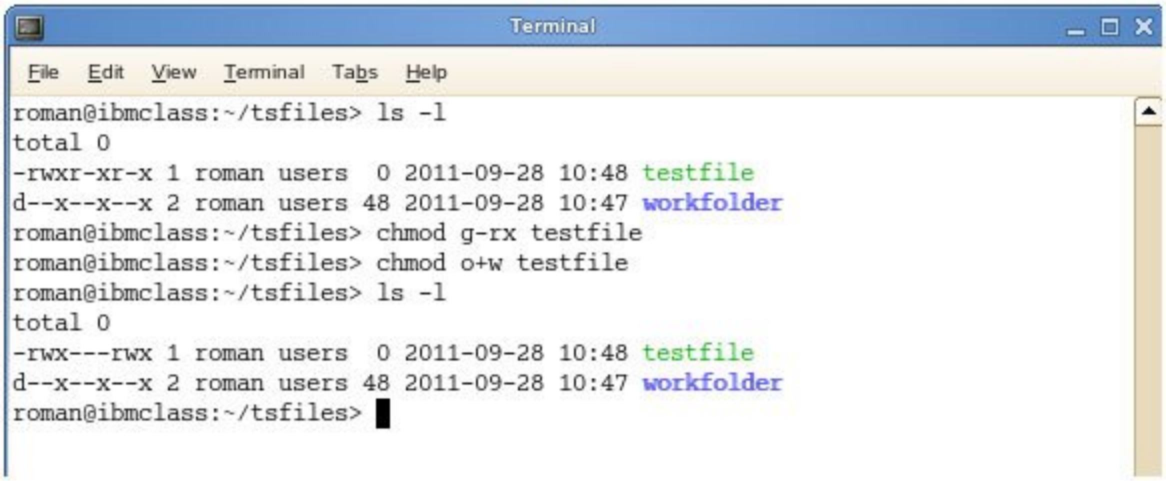
```
roman@ibmclass:~/tsfiles> ls-l
total 0
-r--r--r-- 1 roman users  0 2011-09-28 10:48 testfile
drwxr-xr-x 2 roman users 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles> 
```


-
rwx
r--
r--
“-” indicates a file
“d” indicates directory
“l” indicates a link
Read, write, and execute permissions for the owner of the file
Read, write, and execute permissions for members of the group owning the file
Read, write, and execute permissions for other users

How to Change Directory Permissions in Linux for the Group Owners and Others

The command for changing directory permissions for group owners is similar, but add a “g” for group or “o” for users:

- **chmod g+w filename**
- **chmod g-wx filename**
- **chmod o+w filename**
- **chmod o-rwx foldername**



```
roman@ibmclass:~/tsfiles> ls -l
total 0
-rwxr-xr-x 1 roman users  0 2011-09-28 10:48 testfile
d--x--x--x 2 roman users 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles> chmod g-rx testfile
roman@ibmclass:~/tsfiles> chmod o+w testfile
roman@ibmclass:~/tsfiles> ls -l
total 0
-rwx---rwx 1 roman users  0 2011-09-28 10:48 testfile
d--x--x--x 2 roman users 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles> █
```

To change directory permissions for everyone, use “u” for users, “g” for group, “o” for others, and “ugo” or “a” (for all).

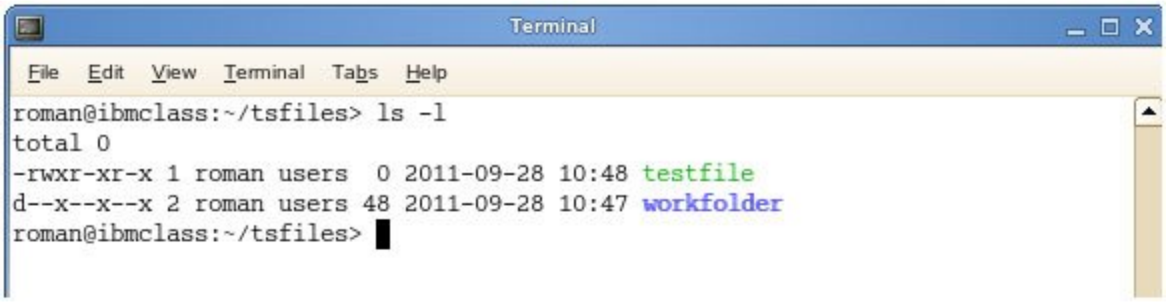
- **chmod ugo+rwx foldername** to give read, write, and execute to everyone.
- **chmod a=r foldername** to give only read permission for everyone.

How to Change Groups of Files and Directories in Linux

By issuing these commands, you can change groups of files and directories in Linux.

- **chgrp groupname filename**
- **chgrp groupname foldername**

Note that the group must exit before you can assign groups to files and directories.

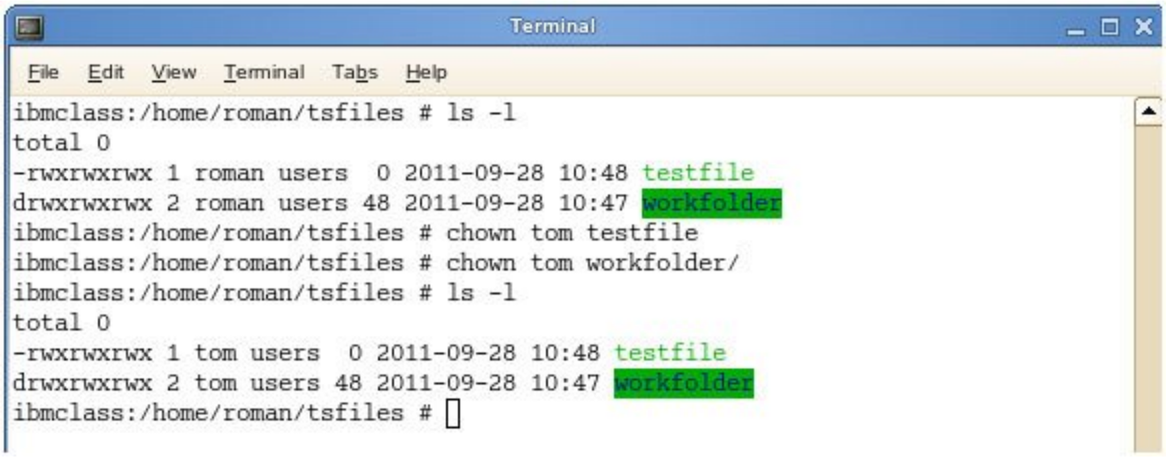


```
roman@ibmclass:~/tsfiles> ls -l
total 0
-rwxr-xr-x 1 roman users  0 2011-09-28 10:48 testfile
d--x--x--x 2 roman users 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles> █
```

How to Change Ownership in Linux

Another helpful command is changing ownerships of files and directories in Linux:

- **chown name filename**
- **chown name foldername**

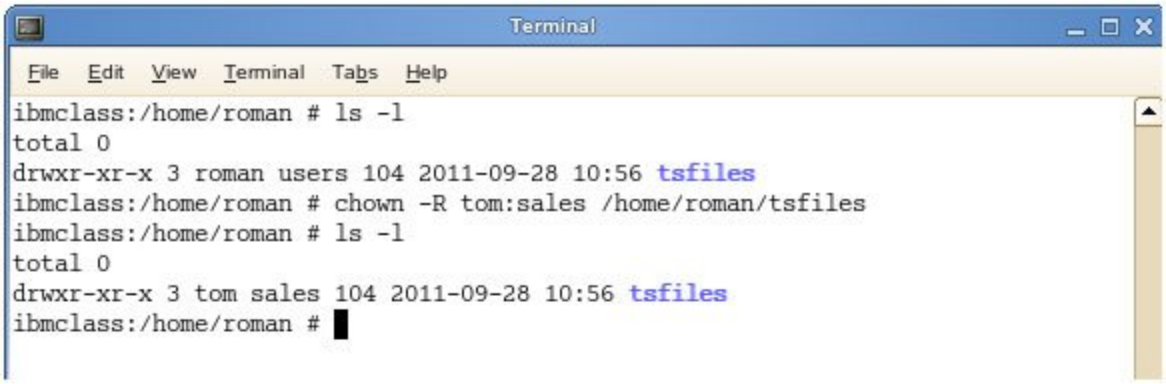


```
ibmclass:/home/roman/tsfiles # ls -l
total 0
-rwxrwxrwx 1 roman users  0 2011-09-28 10:48 testfile
drwxrwxrwx 2 roman users 48 2011-09-28 10:47 workfolder
ibmclass:/home/roman/tsfiles # chown tom testfile
ibmclass:/home/roman/tsfiles # chown tom workfolder/
ibmclass:/home/roman/tsfiles # ls -l
total 0
-rwxrwxrwx 1 tom users  0 2011-09-28 10:48 testfile
drwxrwxrwx 2 tom users 48 2011-09-28 10:47 workfolder
ibmclass:/home/roman/tsfiles # 
```

These commands will give ownership to someone, but all sub files and directories still belong to the original owner.

You can also combine the group and ownership command by using:

- **chown -R name:filename /home/name/directoryname**



```
ibmclass:/home/roman # ls -l
total 0
drwxr-xr-x 3 roman users 104 2011-09-28 10:56 tsfiles
ibmclass:/home/roman # chown -R tom:sales /home/roman/tsfiles
ibmclass:/home/roman # ls -l
total 0
drwxr-xr-x 3 tom sales 104 2011-09-28 10:56 tsfiles
ibmclass:/home/roman # 
```

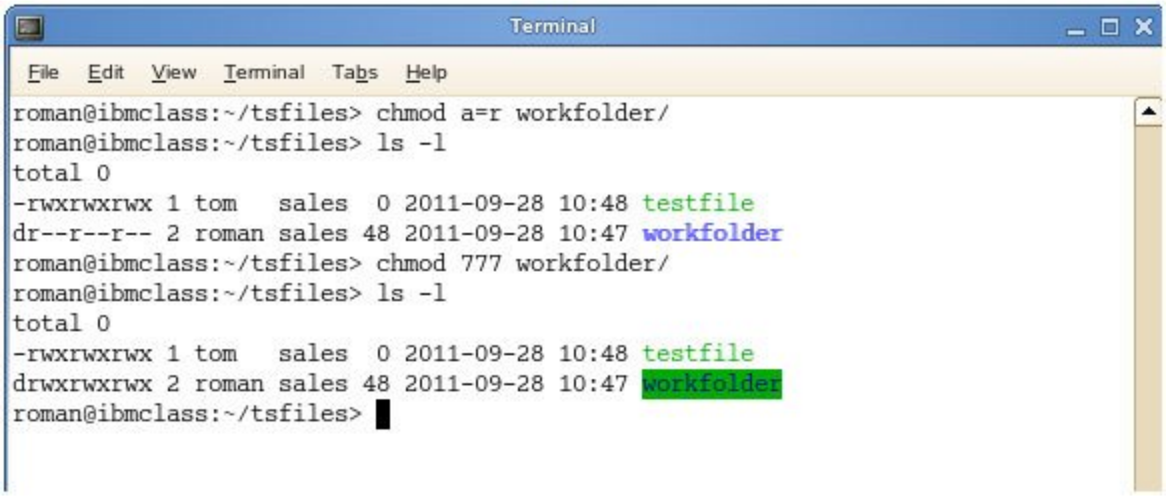
This command gives someone the ownership of the directory tsfiles, and all files and subfolders. The -R stands for recursive, which transfers ownership of all sub directories to the new owner.

How to Change Permissions in Numeric Code in Linux

You may need to know how to change permissions in numeric code in Linux, so to do this you use numbers instead of “r”, “w”, or “x”.

- **0 = No Permission**
- **1 = Execute**
- **2 = Write**
- **4 = Read**

Basically, you add up the numbers depending on the level of permission you want to give.



```
roman@ibmclass:~/tsfiles> chmod a=r workfolder/
roman@ibmclass:~/tsfiles> ls -l
total 0
-rwxrwxrwx 1 tom   sales  0 2011-09-28 10:48 testfile
dr--r--r-- 2 roman sales 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles> chmod 777 workfolder/
roman@ibmclass:~/tsfiles> ls -l
total 0
-rwxrwxrwx 1 tom   sales  0 2011-09-28 10:48 testfile
drwxrwxrwx 2 roman sales 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles> 
```

Permission numbers are:

- **0 = ---**
- **1 = --x**
- **2 = -w-**
- **3 = -wx**
- **4 = r-**
- **5 = r-x**
- **6 = rw-**
- **7 = rwx**

For example:

- **chmod 777 foldername** will give read, write, and execute permissions for everyone.
- **chmod 700 foldername** will give read, write, and execute permissions for the user only.
- **chmod 327 foldername** will give write and execute (3) permission for the user, w (2) for the group, and read, write, and execute for the users.

As you can see, there are several options when it comes to permissions. You have the capability to dictate usability among users. While it may be easier to just give all permission to everyone, it may end up biting you in the end. So choose wisely.