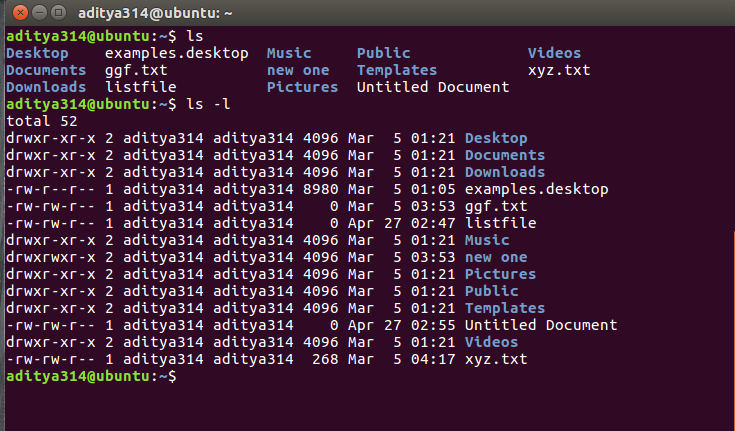
**File Permissions in Linux**

Linux is a multi-user operating system, so it has security to prevent people from accessing each other’s confidential files.

When you execute an “ls” command, you are not given any information about the security of the files, because by default “ls” only lists the names of files. You can get more information by using an “option” with the “ls” command. All options start with a ‘-‘. For example, to execute “ls” with the “long listing” option, you would type ls -l

When you do so, each file will be listed on a separate line in long format. There is an example in the window below. 



There’s a lot of information in those lines. 

1. The first character will almost always be either a ‘-‘, which means it’s a file, or a ‘d’, which means it’s a directory.
2. The next nine characters (rw-r–r–) show the security; we’ll talk about them later.
3. The next column shows the owner of the file. In this case it is me, my userID is “aditya314”.
4. The next column shows the group owner of the file. In my case I want to give the “aditya314” group of people special access to these files.
5. The next column shows the size of the file in bytes.
6. The next column shows the date and time the file was last modified.
7. And, of course, the final column gives the filename.

Deciphering the security characters will take a bit more work.

**Understanding the security permissions**

First, you must think of those nine characters as three sets of three characters (see the box at the bottom). Each of the three “rwx” characters refers to a different operation you can perform on the file. 

--- --- ---

rwx rwx rwx

user group other

**Read, write, execute and –**

The ‘r’ means you can “read” the file’s contents.   
The ‘w’ means you can “write”, or modify, the file’s contents.   
The ‘x’ means you can “execute” the file. This permission is given only if the file is a program.   
If any of the “rwx” characters is replaced by a ‘-‘, then that permission has been revoked.

**User, group and others**

user – The user permissions apply only the owner of the file or directory, they will not impact the actions of other users.   
group – The group permissions apply only to the group that has been assigned to the file or directory, they will not effect the actions of other users.   
others – The others permissions apply to all other users on the system, this is the permission group that you want to watch the most.

**Reading the security permissions**

For example, consider that the user’s permissions for some files is “rw-” as the first three characters. This means that the owner of the file (“aditya314”) can “read” it (look at its contents) and “write” it (modify its contents). I cannot execute it because it is not a program; it is a text file.

If “r-x” is the second set of 3 characters it means that the members of the group “aditya314” can only read and execute the files.

The final three characters show the permissions allowed to anyone who has a UserID on this Linux system. Let us say we have the permission (“r–“). This means anyone in our Linux world can read, but they cannot modify the contents of the files or execute it.

**Changing security permissions**

The command you use to change the security permissions on files is called “chmod”, which stands for “change mode”, because the nine security characters are collectively called the security “mode” of the file.

1. The first argument you give to the “chmod” command is ‘u’, ‘g’, ‘o’. We use:   
   u for user   
   g for group   
   o for others,   
   you can also use a combination of them (u,g,o).   
   This specifies which of the three groups you want to modify.
2. After this use   
   a ‘+’ for adding   
   a ‘-‘ for removing   
   and a “=” for assigning a permission.
3. Then specify the permission r,w or x you want to change.   
   Here also you can use a combination of r,w,x.   
   This specifies which of the three permissions “rwx” you want to modify
4. use can use commas to modify more permissions
5. Finally, the name of the file whose permission you are changing

For example, if you want to give “execute” permission to the world (“other”) for file “xyz.txt”, you would start by typing 

chmod o

Now you would type a ‘+’ to say that you are “adding” a permission. 

chmod o+

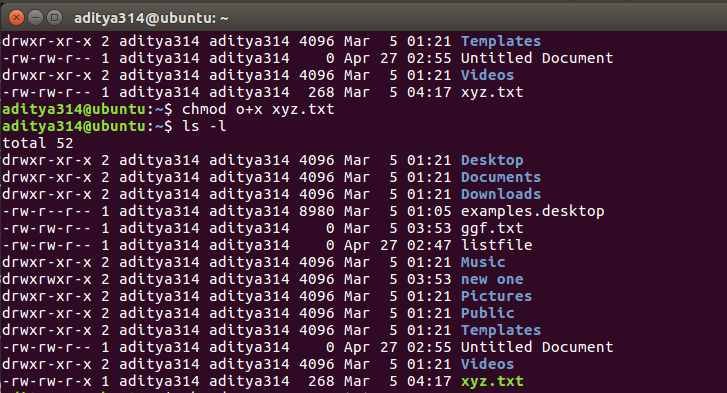
Then you would type an ‘x’ to say that you are adding “execute” permission. 

chmod o+x

Finally, specify which file you are changing. 

chmod o+x xyz.txt

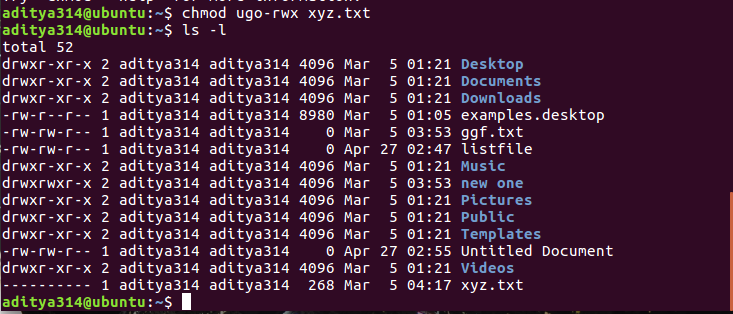
You can see the change in the picture below. 



You can also change multiple permissions at once. For example, if you want to take all permissions away from everyone, you would type 

chmod ugo-rwx xyz.txt

The code above revokes all the read(r), write(w) and execute(x) permission from all user(u), group(g) and others(o) for the file xyz.txt which results to this. 



Another example can be this: 

chmod ug+rw,o-x abc.mp4

The code above adds read(r) and write(w) permission to both user(u) and group(g) and revoke execute(x) permission from others(o) for the file abc.mp4.

Something like this: 

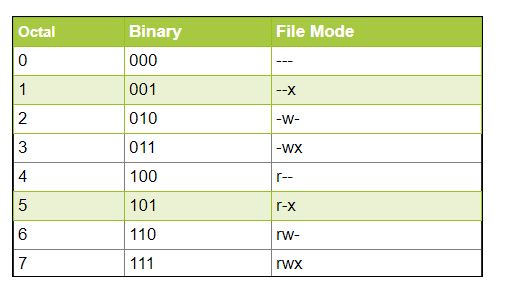
chmod ug=rx,o+r abc.c

assigns read(r) and execute(x) permission to both user(u) and group(g) and add read permission to others for the file abc.c.

There can be numerous combinations of file permissions you can invoke, revoke and assign. You can try some in your linux system.

**The octal notations**

You can also use octal notations like this. 



Using the octal notations table instead of ‘r’, ‘w’ and ‘x’. Each digit octal notation can be used of either of the group ‘u’,’g’,’o’.

So, the following work the same. 

chmod ugo+rwx [file\_name]

chmod 777 [file\_name]

Both of them provides full read write and execute permission (code=7) to all the group.

Same is the case with this.. 

chmod u=r,g=wx,o=rx [file\_name]

chmod 435 [file\_name]

Both the codes give read (code=4) permission to user, write and execute (code=3) for group and read and execute (code=5) for others.

And even this… 

chmod 775 [file\_name]

chmod ug+rwx,o=rx [file\_name]

Both the commands give all permissions (code=7) to user and group, read and execute (code=5) for others.