Manage file ownership and permission

chown:

chown is for changing the permission of a file and the folder
we can change the permission to any user with the chown command
to change the group permission we can use the chgrp but it can be done with the chown command also.
So we can change the chown command for also the user and the group

you need to be root to do that thing
=>**chown <username> <file name>**
example:
=>**chown jamal file.txt**

then the owner of the file will be the jamal instead of root or other user

now we can do it recursively in a folder
to change the ownership of all the content of a folder

the command is

=>**chown -R <username> <folder>**

to change the group permission

the command is:

=> **chown : <group_name> <file>**

to change the user and the group permission all

the command is

=> **chown <user_name>: <group_name> <file>**

CHGRP COMMAND:

there is a separate command for changing the group ownership of the file

command is

=>**chgrp <usergroupname> <filename>**
example:

=>**chgrp ornob file.txt**

explaining the long listing format in permission:

example:

drwxr-xr-x 15 pirate pirate   4096 Oct 22 07:20 oop
drwxr-xr-x. 7 pirate pirate   4096 Oct 21 14:44 'power stations'
-rw-rw-r--   1 pirate pirate   1078257 Oct 27 09:14
Screenshot_2018-10-27_09-14-46.png
-rw-rw-r--   1 pirate pirate   1080882 Oct 27 09:15
Screenshot_2018-10-27_09-15-06.png

---

explaining this thing:

drwxr-xr--

d means it is a directory if there is - that (-rw-rw-r--)means it is a file

after this symbol there is nine more spaces in group of three

every three option are read,write and execute

first three (rwx):

first three oprion is for the owner of the file .
So what permission the owner has??
read write and execute (all the permission)

now the next three what permission the group has

in this example it has (r-x):

that means the read and execute but not write

the group can read it and execute but cant modify it

and the last three sows the permission everybody else has

in this example it is (r--) that means other user or group can only read this file can not execute can not write.
Now this is basically represent with the octal number
read(r)=4
write(w)=2
execute(x)=1
so if you want only the read and execute
it will be 4+1=5

type	**user**	**group**	**other**
D(folder)	**rxw=7**	**r-x=5**	**r-x=5**
-(file)	**6=rw-**	**6=rw-**	**4=r--**

Changing the permission:

This (+) and (-) symbol is used for adding and removing permission from the files and folder

and user=>u
group=>g
other=>o

so if we want to remove permission from a file from user group and other (that means only administrator(root) can access it not any user any group or other)

the command is

=>chmod ugo-x f<file_name>

to add all the read write and execute permission to the file

=>chmod ugo+x+w+x <file_name>

doing the same thing with the octal value:

---

so we know that 7 stands for read write and execute(4+2+1)

and read and execute stands for 5 (4+0+1)

and read stands for only 4(4+0+0)

now if we want to give user all the permission , the group read and execute permission and others only read permission the command is

=>chmod 754 <file_name>

explanation:

	user	group	other	filename
	-------	---------	--------	---------
=>chmod	**7(rwx)**	**5(r-x)**	**4(r--)**	**<file_name>**

umask

when we create file and folder and there is a default permission when we type
‘umask’ in the terminal we get a n octal value like 022 so when we create a file the permission will be the 777-umask value
example
777-022=755 means
rwx for user
r-x for group
r-x for other

we can change the umask value and set it to as our need . and to do it permanently we would add the umask value to the .profile

SUID

suid = set user id

suppose you have to do some command but you dont have the permission of running that file other or may be most of the time is root so if you want that if you want something like that when you execute the program it will automatically run it as a root then comes the topic set user id or suid

suppose you eant to run the ‘ping’ command to know the network status but ping command has to access the low level hardwere and only the root user has that kind of access so if you want to do ping as other user but it will run as a root user then you have sti set user id

command is:

=>chmod u+s <file>

remember it will be with ‘u’ because it is suid
example:
[vagrant@localhost ~]$ chmod u+s base.sh
[vagrant@localhost ~]$ ll
total 24
-rw**s**r-xr-x 1 vagrant vagrant 450 Jul 16 2015 base.sh

so if anybody wants to run the base.sh file it will be run as a user vagrant.suppose jamal user run this script but it will be run as a vagrant

guid

guid=group user id

its just like the suid but instead of user it will run as the permission of the group that is given to

guid is very effective in folder .for example if you set a folder a group user id or guid any file that is created inside the folder become the property of that group.
For example there is two group marketing and production and two folder are given guid for marketing and production so any folder inside the production folder will be the property of the production and any file inside the marketing will be the property of the marketing.

Command is:

=>chmod g+s folder

after this command on a folder if you add any file inside the folder it will also consider as the permission of the user

Sticky bit

if a file has been set with the sticky bit .only the owner of the file can delete file others can’t .it is very important in the share folder(or may be the /tmp directory) cause everybody writes there and if we want to prevent delete one user file by another user we have to set the sticky bit .so the other user cant delete but the owner can do that . remember sticky bit actually use on folder not file

command:

=>chmnod o+t <foldername>

remember suid will modify the user portion giud will modify the group portion and the sticky bit will affect the other portion

using the octal value

we know how to do permission in the files and folder so we don't have to change this thing just before the three number permission(example 777 or 755) we have to add the sticky bit before the permission

	Octal value
suid	**4**
guid	**2**
Sticky bit	**1**

Example if we want to set the guid and suid both in a file which has a permission of 755 we will do
(6755) that means last 755 is the permission and first is the (4+2) suid+guid