OPERATING SYSTEM

Users and Groups Management and Permission Commands

show users and groups

First you should be admin to allow

```
~ $ sudo  -i
```

- Show current user

```
~ # users
```

- List users

```
~ # getent passwd    OR  cat   /etc/passwd
```

- Show groups for current user

```
~ # groups
```

- List groups

```
- ~ # getent passwd    OR   cat   /etc/group
```

Creating users (adduser vs useradd)

Before add user you should to be admin by using command

```
~ $ sudo -i
~ # useradd username
~ # useradd user1
```

After adding a user with the command “***useradd***” we need to set a password for the user using the command “***passwd***”.

```
 ~ # passwd  user1
```

- changing a user password :

```
~ # passwd user1
```

remove users (userdel)

- remove user:

```
 ~ # userdel  user1
```

(-r)remove user home and mail tail when we use (adduser)

```
 ~ # userdel  -r  user1
```

-Modifying user name

```
~ # usermod  -d  new_username   old_username
```

creating group : groupadd

- To create groups we’ll use first the command “groupadd” followed by the group’s name. The syntax is: “groupadd <group-name>”
- □ Example:

```
~# groupadd group1
```

- Create user with specific group

```
~# useradd –G group1 user2
```

Modify group name

```
~# groupmod –n new-group old-group
~# groupmod –n group2 group1
```

groups and users

- Remove group

```
~# groupdel group2
```

- Show groups for a user

```
  ~ # groups username
```

- Add existing user to existing group

```
  ~ # usermod   -a  -G  group1  user2
```

- Add user to groups

```
- ~ # useradd   -G  group1,group2,group3,..     username
```

modify group (groupmod)

Syntax:

```
 ~ # groupmod  -n   newname  old-groupname
```

Example :

```
~ # groupmod  -n   group23  group2
```

removing user from group

Example :

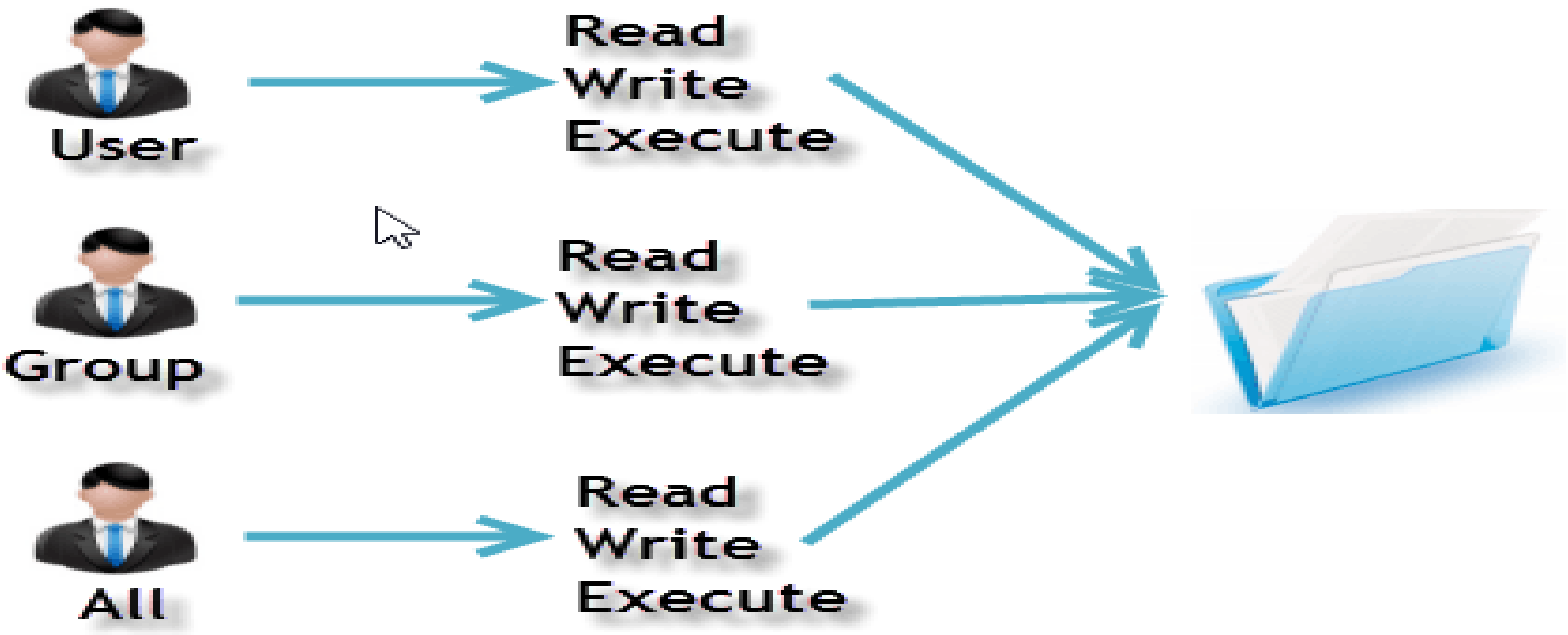
```
~ # gpasswd  -d   username  groupname
~ # gpasswd  -d   user2  group3
```

- Change user primary group

```
~ # usermod  -g   groupname  username
~ # usermod  -g   group2  user2
```

Permissions in Linux

Owners assigned Permission On Every File and Directory



File Permissions in Linux/Unix

chmod

The “**chmod**” command in **Linux** enables you to control the access of scripts, directories, and your system files. This command is utilized to change the Linux file permissions

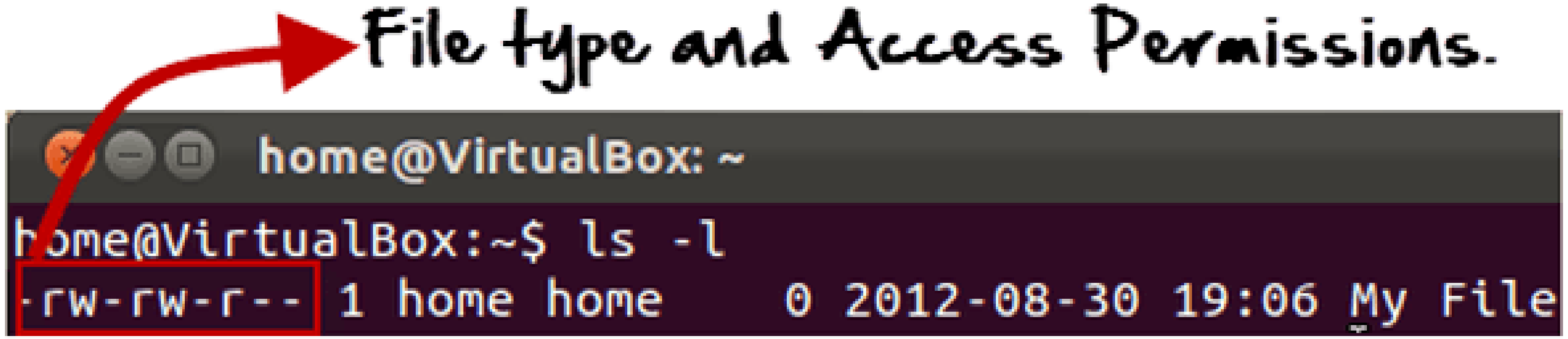
What are file permissions in Linux

File permission is the type of access associated with a file. Each file in Linux has its owner, a group, and permission access for **three main types of users**: the **file owner**, the **group** members, **and others**. Each of these user classes has **three types of file permissions**: **read**, **write**, and **execute** permissions. Knowing about the file permission helps you specify which users can execute, read, or write the file.

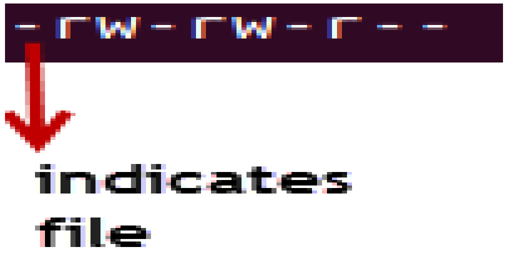
chmod

The “**ls -l**” is used to check the permission of the files present on your system. To view permission of a single file, add its name to the “**ls -l**” command.

$ ls -l testfile

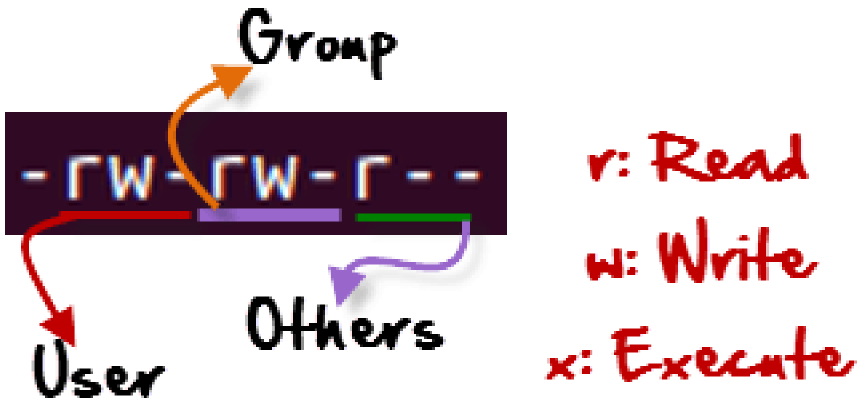


```
home@VirtualBox:~$ ls -l
-rw-rw-r-- 1 home home    0 2012-08-30 19:06 My File
```



r = read permission
w = write permission
x = execute permission
– = no permission

chmod



chmod

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

chmod

Examples :

if you want to give “execute” permission to the world (“other”) for file “xyz.txt”

```
chmod o+x xyz.txt
```

- adds read(r) , write(w),execute(x) permission to user(u) and group(g) and other(o)

```
chmod ugo+rwx [file_name]
```

-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.

```
chmod ug+rw,o-x abc.mp4
```

-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.

```
chmod ug=rx,o+r abc.c
```

chmod

Examples :

- Remove permission write from group

```
chmod  g-w  abc.jpg
```

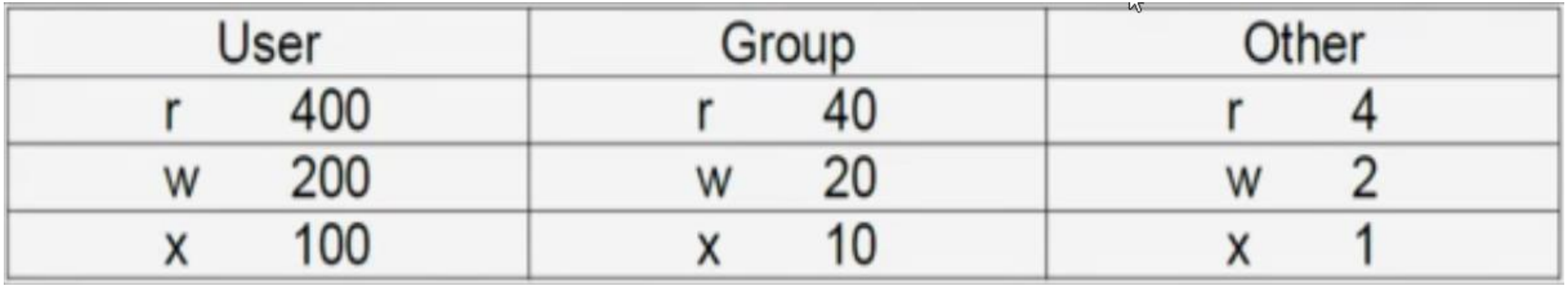
 - assign permissions

```
chmod  u=rw,g=wx,o=rx abc.jpg
```

- Remove execute from all

```
chmod  -x  abc.jpg
```

chmod (another method)



User	Group	Other
r 400	r 40	r 4
w 200	w 20	w 2
x 100	x 10	x 1

Examples : user : read write execute , group : read write ,

Others :read

chmod 764 abc