APPLIED OPERATING SYSTEM LABORATORY

MODULE 7

LINUX FILE/DIRECTORY PERMISSION AND OWNERSHIP WITH USER/GROUP ADMINISTRATION

OBJECTIVES

Upon completion of this module, the student will be able to:

- Assign/modify permission and different ownership to files and directories using symbolic and absolute file permission commands
- Create/delete user/group account

TOPIC OUTLINE

- **Change File/Directory Permission**
  - Symbolic mode
  - Absolute mode
- **Change File/Directory Ownership**
  - Change User ownership
  - Change Group ownership
- **User and Group Administration**
  - Create/Delete User Account
  - Create/Delete Group

FILE PERMISSION COMMANDS

chmod command allows changing the file access permission of a file.

Syntax: **chmod** [reference][operator][mode] <filename>

Techniques:

- Symbolic Mode
- Absolute Mode

Note: *Permissions can only be assigned by the **root user** or the **owner** of the said file or directory*

FILE PERMISSION COMMANDS

Symbolic Mode changes a file's permission by using symbolic notation.

- The **FIRST set** determines who is granted or denied a specific set of permissions. The first 3 sets of flags are as follows:
  - u= user/owner of the file
  - g= (group) users who are members of the file's group
  - o= other users
  - a= all (owner, group and others)

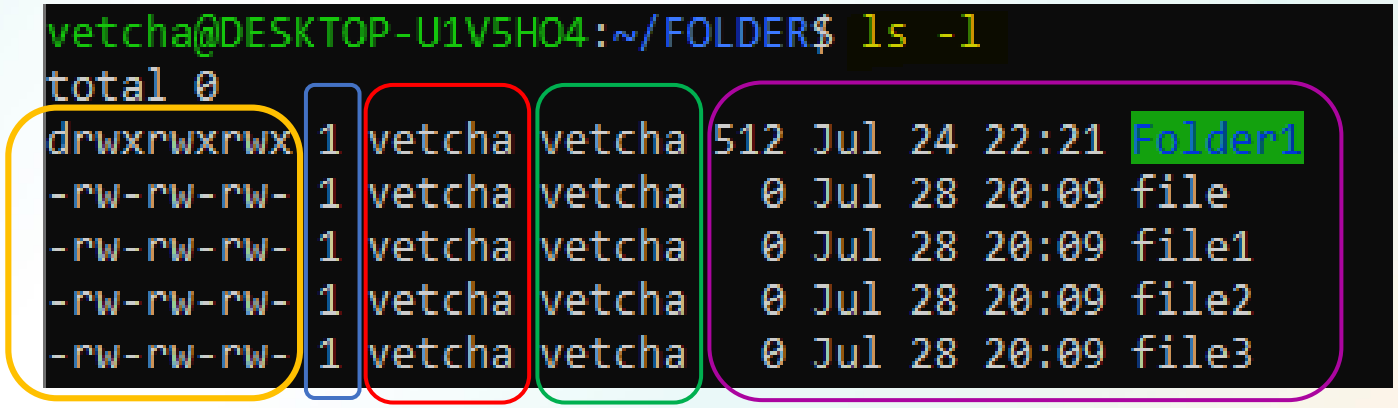
The **SECOND set** of flags determines whether permissions will be added, removed, or set:

- **+** (add permission)
- **-** (remove permission)
- **=** (set permission)

FILE PERMISSION COMMANDS

Symbolic Mode

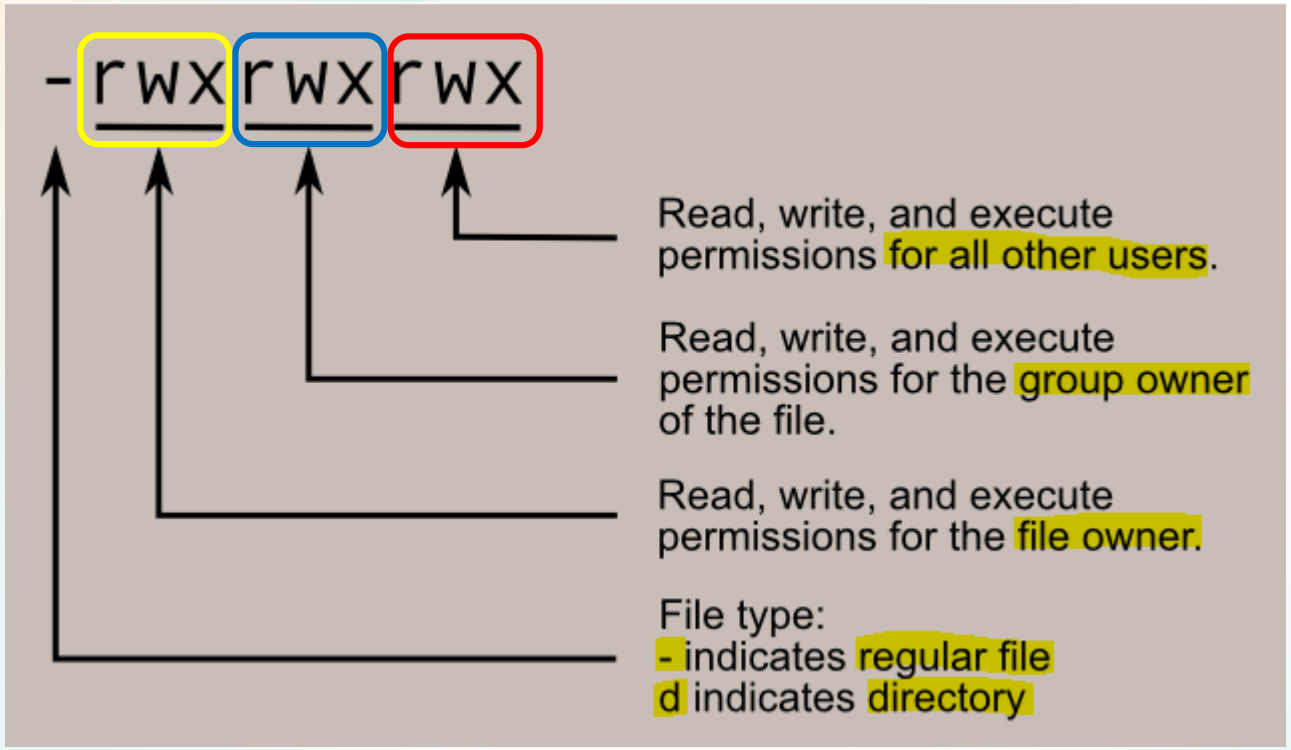
- The **THIRD set** determines what permissions will be given
  - **r** (read)
  - **w** (write)
  - **x** (execute)



The **output of the ls –l command** (as seen above) is interpreted as follows:

- The **first column**, shows the **file type** and **permissions**
- The **second column** shows the **number of links** (directory entries that refer to the file)
- The **third column** shows the **owner of the file**
- The **fourth column** shows the **group the file belongs to**
- The **other columns** show the **file's size in bytes**, **date** and **time of last modification**, and the **filename.**

FILE PERMISSION COMMANDS




```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-rw-rw-rw- 1 vetcha vetcha 0 Jul 28 20:09 file1
```

FILE PERMISSION COMMANDS

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-rw-rw-rw- 1 vetcha vetcha 0 Jul 28 20:09 file1
```

Examples:

- To change the permission of ***file1*** from –rw-rw- rw- to –rwx -rw- r-- :

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ chmod u+x file1; chmod o-w file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-rwxrw-r-- 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$
```

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ chmod u=rwx file1; chmod g=rw file1; chmod o=r file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-rwxrw-r-- 1 vetcha vetcha 0 Jul 28 20:09 file1
```

- To change the permission of *file1* from –rwx-rw-r-- to –r -- r-- r-- :

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-rwxrw-r-- 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ chmod a=r file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-r--r--r-- 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$
```

FILE PERMISSION COMMANDS

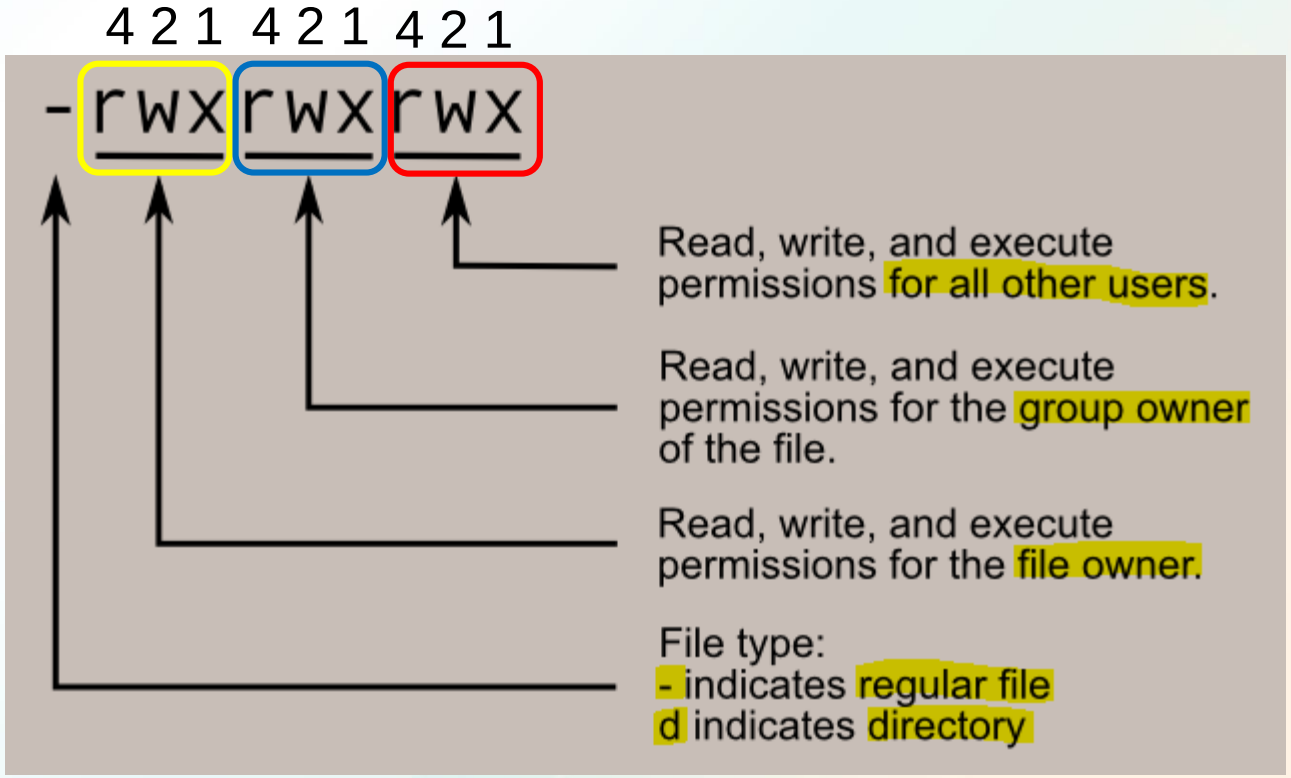
Absolute Mode changes a file's permission by using numbers/octal notation. The numeric mode is the sum of one or more of the following values:

r (read) = 4

w (write) = 2

x (execute) =1

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-r--r--r-- 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ chmod 666 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-rw-rw-rw- 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ chmod 777 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-rwxrwxrwx 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$
```



FILE PERMISSION COMMANDS

Examples:

To change the permission of ***file1*** from –r--r—r-- to –r-x--- --- :

Syntax: **chmod 500 file1**

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-r--r--r-- 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ chmod 500 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l file1
-r-x------ 1 vetcha vetcha 0 Jul 28 20:09 file1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$
```

To change the permission of all the files in the directory ***Folder1***:

Syntax: **chmod –R 755 *Folder1***

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l
total 0
drwxrwxrwx 1 vetcha vetcha 512 Jul 29 01:29 Folder1
-rwxrw-r-- 1 vetcha vetcha   0 Jul 28 20:09 file
-rwxrwxrwx 1 vetcha vetcha   0 Jul 28 20:09 file1
-rw-rw-rw- 1 vetcha vetcha   0 Jul 28 20:09 file2
-rw-rw-rw- 1 vetcha vetcha   0 Jul 28 20:09 file3
```

```
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ chmod -R 755 Folder1
vetcha@DESKTOP-U1V5HO4:~/FOLDER$ ls -l
total 0
drwxr-xr-x 1 vetcha vetcha 512 Jul 29 01:29 Folder1
-rwxrw-r-- 1 vetcha vetcha   0 Jul 28 20:09 file
-rwxrwxrwx 1 vetcha vetcha   0 Jul 28 20:09 file1
-rw-rw-rw- 1 vetcha vetcha   0 Jul 28 20:09 file2
-rw-rw-rw- 1 vetcha vetcha   0 Jul 28 20:09 file3
```

FILE OWNERSHIP COMMANDS

chown command

- To change the ownership of a file or directory.

  Syntax: **chown** owner file/directory

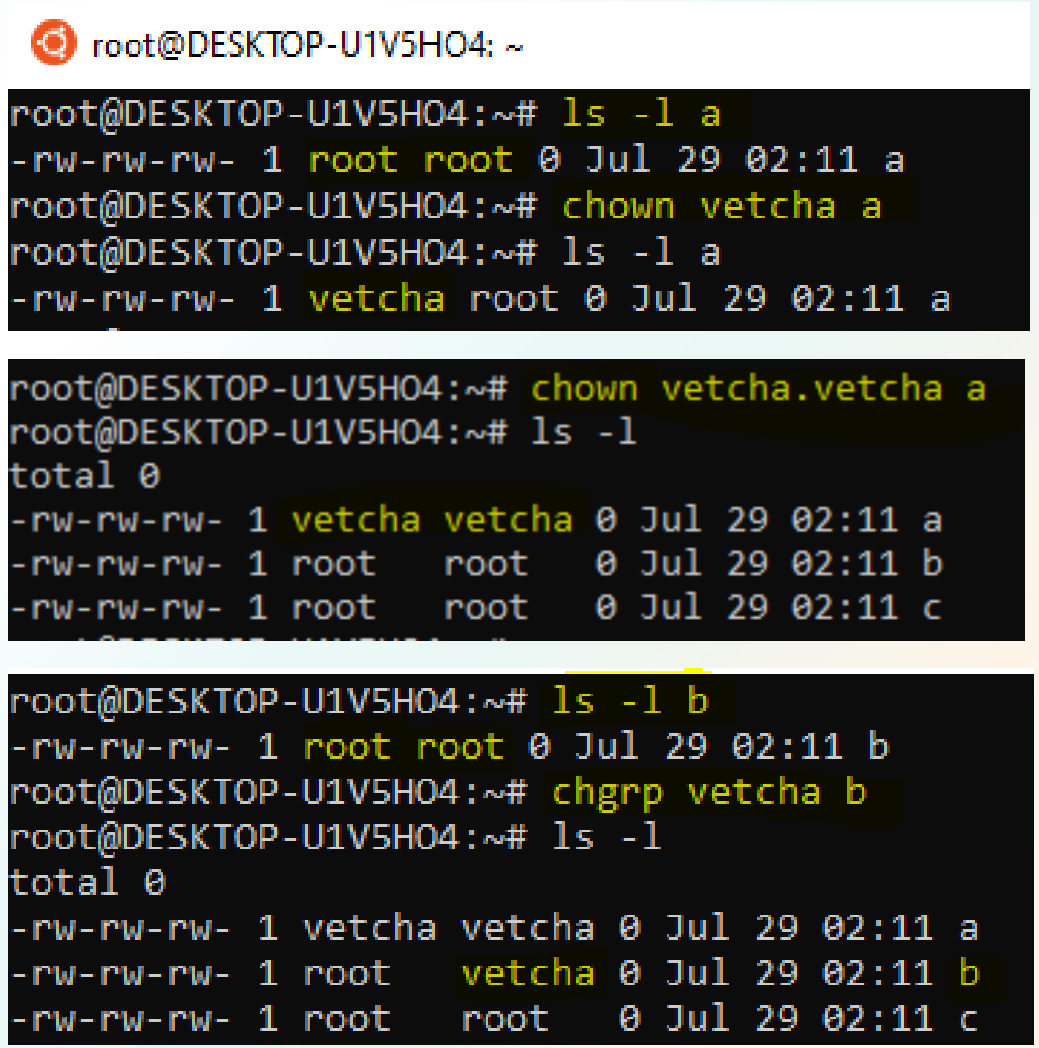
- Changing group ownership

  Syntax: **chown** owner.group file/dir

chgrp command

- To change the group ownership of a file or directory:

  Syntax: **chgrp** groupname file/directory



USER AND GROUP ADMINISTRATION

useradd command

- To create a user account

  Syntax: **useradd** username

userdel command

- To remove/delete a user account

```
root@DESKTOP-U1V5HO4:~# adduser euni
Adding user `euni' ...
Adding new group `euni' (1001) ...
Adding new user `euni' (1001) with group `euni' ...
Creating home directory `/home/euni' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for euni
Enter the new value, or press ENTER for the default
        Full Name []:
```

root@DESKTOP-U1V5HO4: ~

```
root@DESKTOP-U1V5HO4:~# useradd SAM
root@DESKTOP-U1V5HO4:~#
```

```
root@DESKTOP-U1V5HO4:~# useradd SAM
root@DESKTOP-U1V5HO4:~# useradd SAM
useradd: user 'SAM' already exists
root@DESKTOP-U1V5HO4:~# userdel SAM
root@DESKTOP-U1V5HO4:~# userdel SAM
userdel: user 'SAM' does not exist
root@DESKTOP-U1V5HO4:~#
```

USER AND GROUP ADMINISTRATION

groupadd command

- To create new group; a group can only be added one at a time

  Syntax: **groupadd** groupname

```
root@DESKTOP-U1V5HO4:~# useradd USER
root@DESKTOP-U1V5HO4:~# useradd USER
useradd: user 'USER' already exists
root@DESKTOP-U1V5HO4:~# groupadd USER
```

groupdel command

- To delete group;

  Syntax: **groupdel** groupname

```
root@DESKTOP-U1V5HO4:~# groupdel SAM
]groupdel: cannot remove the primary group of user 'SAM'
root@DESKTOP-U1V5HO4:~# groupdel -f SAM
root@DESKTOP-U1V5HO4:~# groupdel SAM
groupdel: group 'SAM' does not exist
root@DESKTOP-U1V5HO4:~#
```

groups command

- To view the group where a certain user belongs to

```
root@DESKTOP-U1V5HO4:~# groups USER
USER : USER
root@DESKTOP-U1V5HO4:~# groups SAM
SAM : SAM
root@DESKTOP-U1V5HO4:~#
```

REFERENCES

- Sobell, M., et al. (2017). A Practical Guide to Linux Commands, Editors, and Shell Programming, 4th Ed. Addison-Wesley Professional
- Cobbaut, P. (2016). Mastering Linux- Networking
- Blum, R., (2015). Linux Command Line and Shell Scripting Bible