Lecture _4

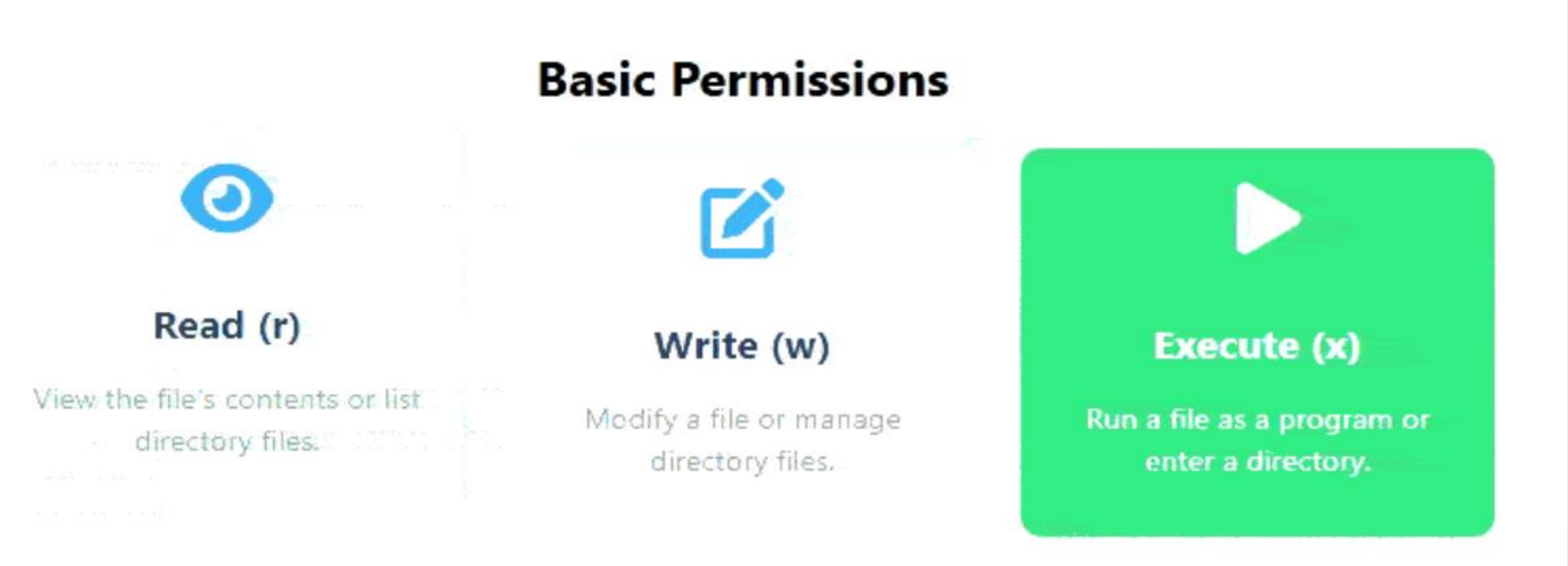
LINUX essentials

Dr .Sara Mohamed

Changing File Permissions

Linux file permissions

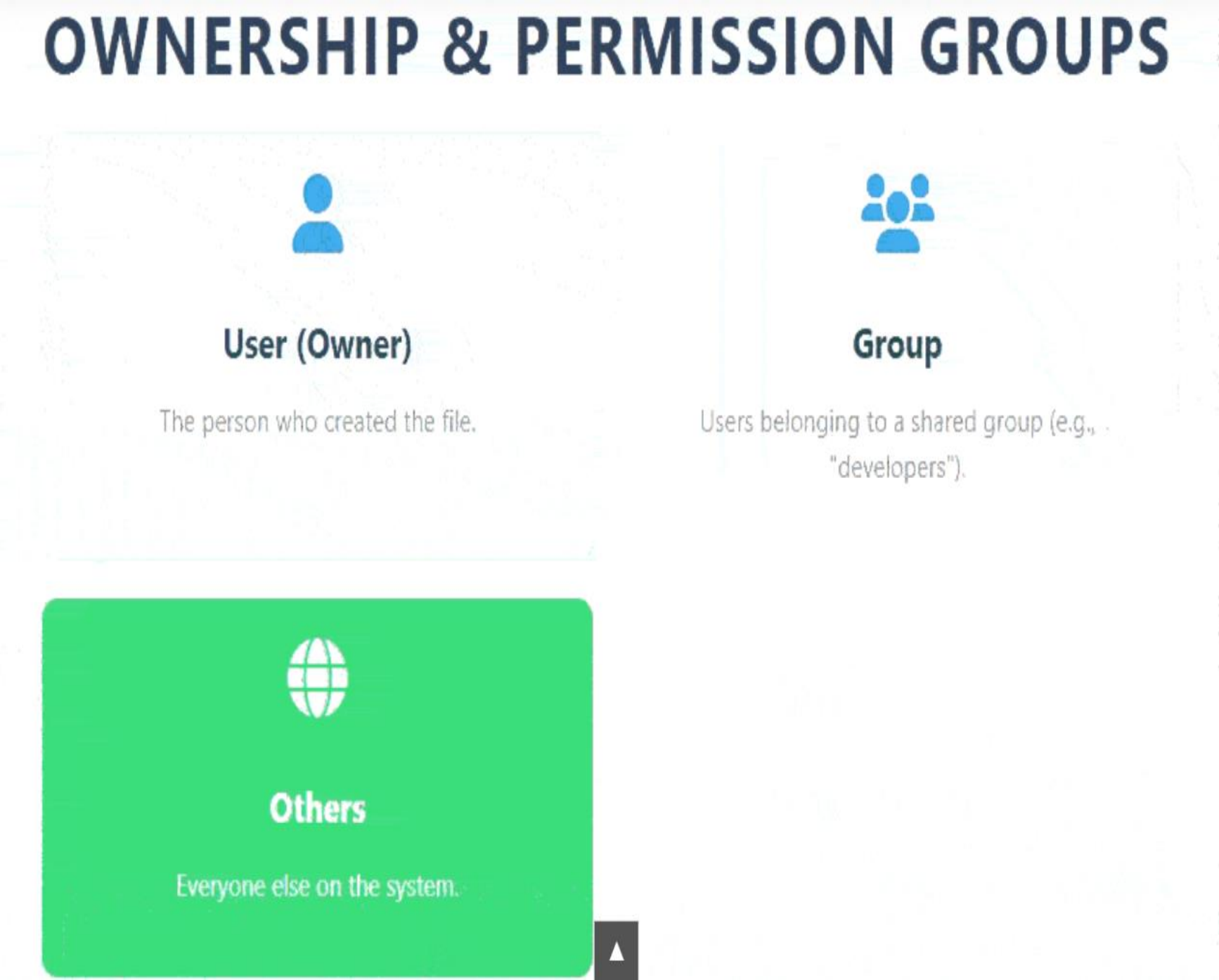
- Linux file permissions form the foundation of the system's security model. They define **who can read, write, or execute** files and directories, ensuring only authorized users or processes can access sensitive data. You can modify these permissions using the chmod command. Every file or directory has three types of permissions:



Letters	Definition
'r'	"read" the file's contents.
'w'	"write", or modify, the file's contents.
'x'	"execute" the file. This permission is given only if the file is a program.

Ownership and Permission Groups

- Permissions are assigned to three categories of users:



The chmod command

- The **chmod** command is used to change the permissions of a file or directory. Only the root user or the user who owns the file is able to change the permissions of a file.
- The command you use to change the security permissions on files is called "chmod", which stands for "**change mode**"
- You can modify permissions using **symbolic notation or octal notation**.

Symbolic Notation

- Symbolic notation allows you to **add, remove,** or **set permissions** for specific users.

```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

To use the symbolic method of chmod first indicate which set of permissions is being changed:

```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

Symbol	Meaning
u	User: The user who owns the file.
g	Group: The group who owns the file.
o	Others: Anyone other than the user owner or member of the group owner.
a	All: Refers to the user, group and others.

Next, specify an action symbol

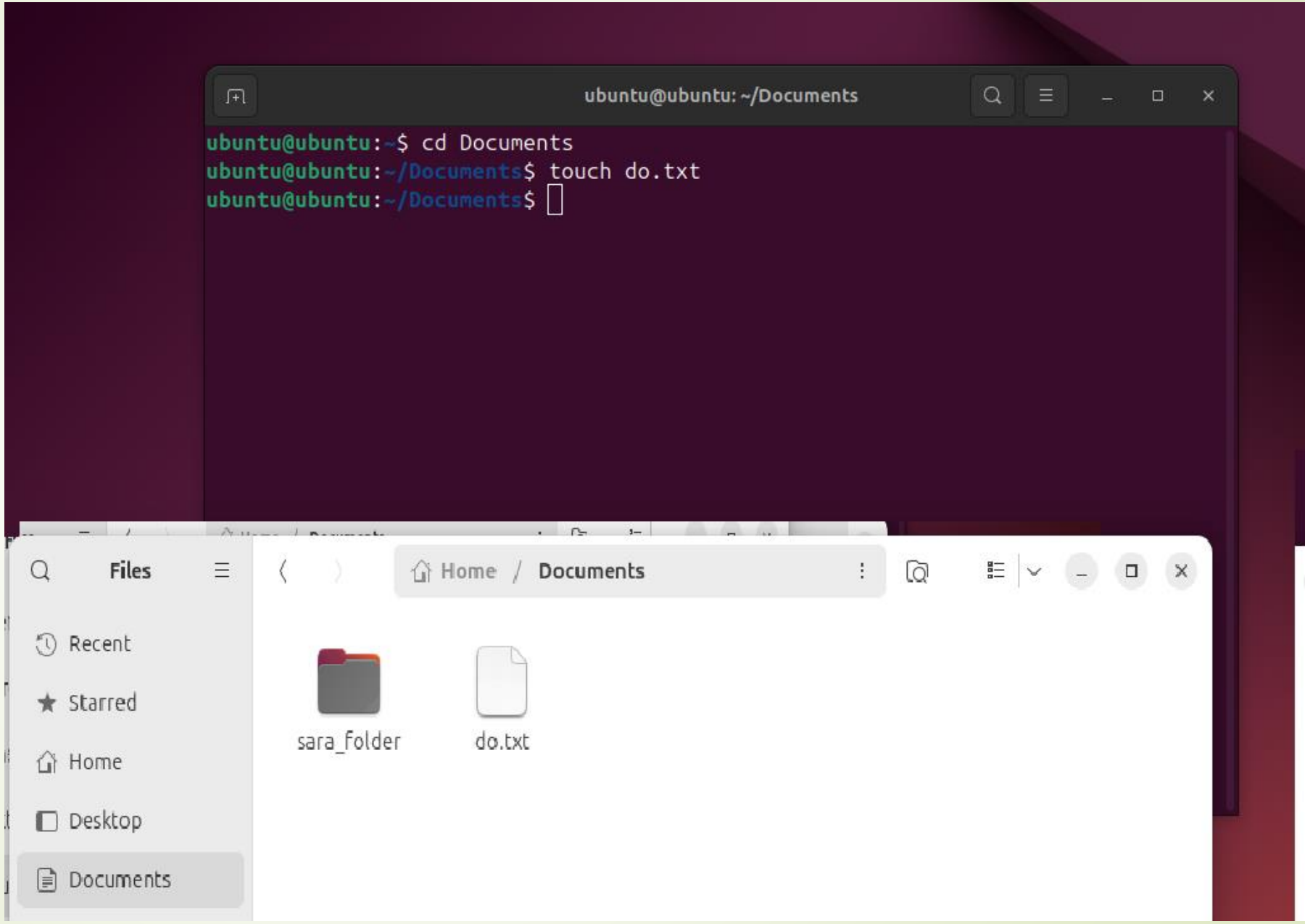
```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

Symbol	Meaning
+	Add the permission, if necessary
=	Specify the exact permission
-	Remove the permission, if necessary

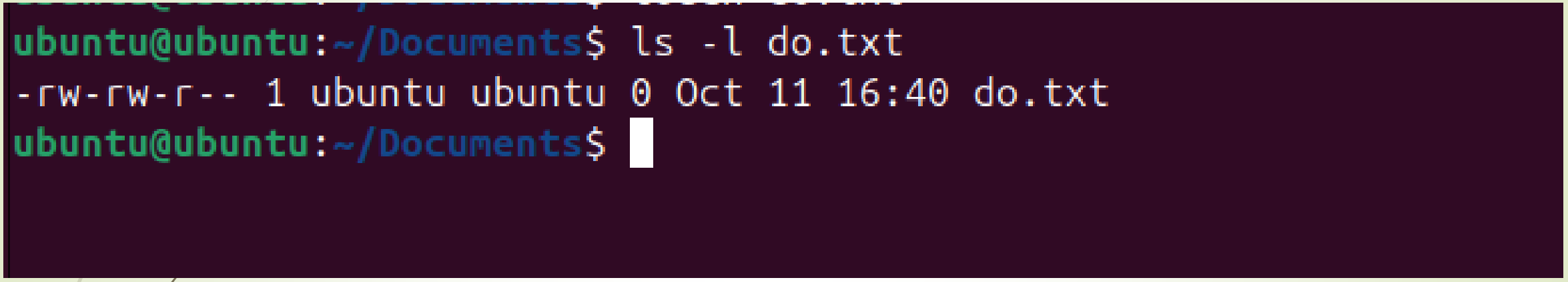
After an action symbol, specify one or more permissions to be acted upon.

Symbol	Meaning
r	read
w	write
x	execute

Create a new file called do.txt



Check the Permission of Files in Linux



How to Change Permissions in Linux

Example1: Add to the other user the permission write

```
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rw-rw-r-- 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$ chmod o+w do.txt
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rw-rw-rw- 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$
```

Example2: remove the write permission from the group owner

```
-rw-rw-rw- 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$ chmod g-w do.txt
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rw-r--rw- 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$
```

Example3: add the write permission for the group and other user

```
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rw-r--r-- 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$ chmod go+w do.txt
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rw-rw-rw- 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$
```

Octal notation

- The octal notation is used to represent file permission in Linux by using three user group by denoting 3 digits i.e.
  - user
  - group
  - other users
- The octal or numeric method requires knowledge of the octal value of each of the permissions and requires all three sets of permissions (user, group, other) to be specified every time.

You can also use octal notations like this.

Octal	Binary	File Mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

octal notations

```
chmod 775 [file_name]
```

The commands give all permissions (code=7) to the user and group, read and execute

Example to give all users types all permissions using the octal method

```
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rw-rw-rw- 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$ chmod 777 do.txt
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rwxrwxrwx 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$
```

Changing File Ownership

Changing File Ownership

- Initially, the owner of a file is the user who creates it. The chown command is used to change the ownership of files and directories. Changing the user owner requires administrative access. A regular user cannot use this command to change the user owner of a file, even to give the ownership of one of their own files to another user. However, the chown command also permits changing group ownership, which can be accomplished by either root or the owner of the file.

Syntax of chown Command in Linux

The chown command is used to modify ownerships and permissions. The basic chown syntax is:

```
chown [options] new_owner[:new_group] file(s)
```

Here's a breakdown of the components:

- `chown`: The base command.
- `options`: Optional flags that modify the behavior of the `chown` command.
- `new_owner[:new_group]`: The new owner and optionally the new group. If `new_group` is omitted, only the owner is changed.
- `file(s)`: The file or files for which ownership is to be changed.

Understanding User Ownership and Permissions in Linux

- Different users in the operating system have ownership and permission to ensure that the files are secure and put restrictions on who can modify the contents of the files. In Linux, different users use the system:
- **Root User:** It is a superuser who has access to all the directories and files in our system, and it can perform any operation. An important thing to note is that only the root user can perform changing of permissions or ownerships of the files that are not owned by them.
- **Regular User:** These users have limited access to files and directories and can only modify the files that they own.

To change the user owner of a file, the following syntax can be used. The first argument, [OWNER], specifies which user is to be the new owner. The second argument, FILE, specifies which file's ownership is changing:

```
chown [OPTIONS] [OWNER] FILE
```

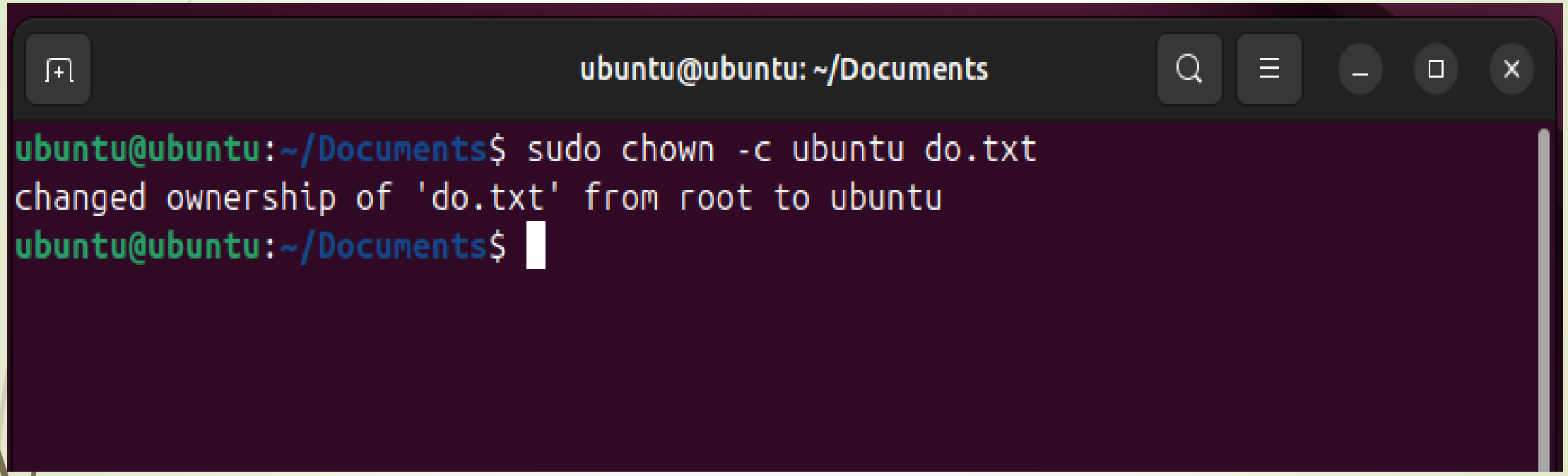
Example_1: change the ownership of do.txt file to root

```
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rwxrwxrwx 1 ubuntu ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$ chown root do.txt
chown: changing ownership of 'do.txt': Operation not permitted
ubuntu@ubuntu:~/Documents$ sudo chown root do.txt
ubuntu@ubuntu:~/Documents$ ls -l  do.txt
-rwxrwxrwx 1 root ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$
```

Using `-c` Option in `chown` to Change File Ownership

- The `**-c**` option in the `**chown**` command is utilized to report when a file change is made. This option is beneficial when you want to receive notifications about ownership alterations.
- This command notifies you when the ownership of file is changed, providing valuable feedback for tracking modifications.

Example2: **Using `-c` Option in `chown`**



Change Group Ownership

Syntax

```
chown :group1 file1.txt
```

- Change Owner as well as Group

Syntax

```
chown master:group1 greek1
```

```
ubuntu@ubuntu:~/Documents$ sudo chown -c root do.txt
changed ownership of 'do.txt' from ubuntu to root
ubuntu@ubuntu:~/Documents$ sudo chown -c root:root do.txt
changed ownership of 'do.txt' from root:ubuntu to root:root
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rwxrwxrwx 1 root root 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$ sudo chown -c :ubuntu do.txt
changed ownership of 'do.txt' from root:root to :ubuntu
ubuntu@ubuntu:~/Documents$ ls -l do.txt
-rwxrwxrwx 1 root ubuntu 0 Oct 11 16:40 do.txt
ubuntu@ubuntu:~/Documents$
```

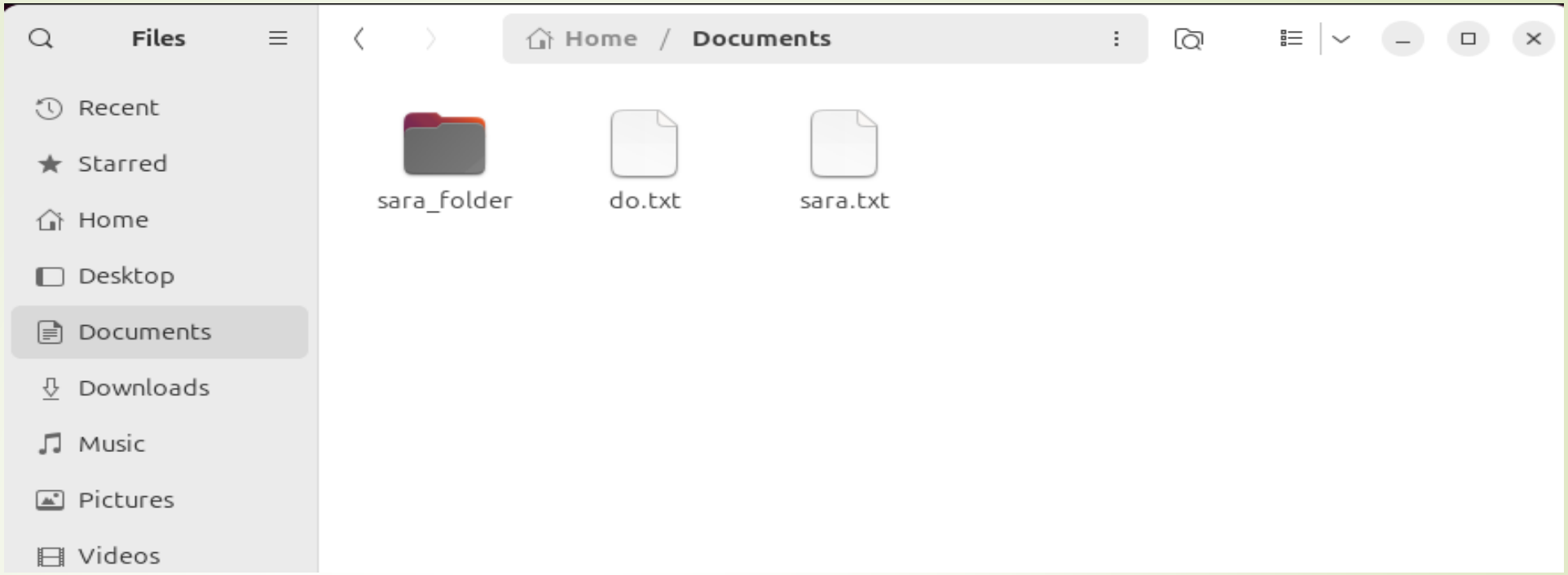
How to Change Owner of Multiple Files

- For simultaneous changes in the owner and group of multiple files.
- In this example, both "greek2" and "greek3" will have their owner set to "master" and their group set to "group.

Syntax

```
chown master:group greek2 greek3
```

Create a new file named sara.txt in the Documents directory




```
ubuntu@ubuntu:~/Documents$ touch sara.txt
ubuntu@ubuntu:~/Documents$ ls -l
total 0
-rwxrwxrwx 1 root   ubuntu  0 Oct 11 16:40 do.txt
-rw-rw-r-- 1 ubuntu ubuntu  0 Oct 11 22:38 sara.txt
drwxrwxr-x 2 ubuntu ubuntu 40 Sep 28 09:30 sara_folder
ubuntu@ubuntu:~/Documents$
```

Change Owner of sara.txt File and sara_folder directory

```
ubuntu@ubuntu:~/Documents$ ls -l
total 0
-rwxrwxrwx 1 root   ubuntu  0 Oct 11 16:40 do.txt
-rw-rw-r-- 1 ubuntu ubuntu  0 Oct 11 22:38 sara.txt
drwxrwxr-x 2 ubuntu ubuntu 40 Sep 28 09:30 sara_folder
ubuntu@ubuntu:~/Documents$ sudo  chown -c root sara.txt sara_folder
changed ownership of 'sara.txt' from ubuntu to root
changed ownership of 'sara_folder' from ubuntu to root
ubuntu@ubuntu:~/Documents$
```

References

- https://www.geeksforgeeks.org/linux-unix/set-file-permissions-linux/
- https://www.geeksforgeeks.org/linux-unix/chown-command-in-linux-with-examples/
- Ramses van Zon," Securing File Access Permissions on Linux ", SciNet HPC, University of Toronto ,27 October 2022.