



Mawlana Bhashani Science and Technology University

Lab -Report

Report No:04
Course code: ICT-3110
Course title: Operating System Lab
Date of Performance:05-08-2020
Date of Submission:12-08-2020

Submitted by

Name: Ruku Shikder
ID: IT-18057
3th year 1st semester
Session: 2017-2018
Dept. of ICT
MBSTU.

Submitted To

Nazrul Islam
Assistant Professor
Dept. of ICT
MBSTU.

Experiment No: 04

Experiment Name: File operation and permission

- What is File Operation and File Permission in Linux Operating System? [1][SEP]
- Implementation of File Operation and File Permission. [1][SEP]

File Operation:

A file is an abstract data type. For defining a file properly, we need to consider the operations that can be performed on files. The operating system can provide system calls to create, write, read, reposition, delete, and truncate files. There are six basic file operations within an Operating system. These are:

- Creating a file: There are two steps necessary for creating a file. First, space in the file system must be found for the file. We discuss how to allocate space for the file. Second, an entry for the new file must be made in the directory.
- Writing a file: To write to a file, you make a system call specify about both the name of the file along with the information to be written to the file.
- Reading a file: To read from a file, you use a system call which specifies the name of the file and where within memory the next block of the file should be placed.
- Repositioning inside a file: The directory is then searched for the suitable entry, and the 'current-file-position' pointer is relocating to a given value. Relocating within a file need not require any actual I/O. This file operation is also termed as 'file seek.'
- Deleting a file: For deleting a file, you have to search the directory for the specific file. Deleting that file or directory release all file space so that other files can re-use that space.
- Truncating a file: The user may wish for erasing the contents of a file but keep the attributes same. Rather than deleting the file and then recreate it, this utility allows all attributes to remain unchanged — except the file length — and let the user add or edit the file content.

File Permission in Linux:

The Unix-like operating systems, such as Linux differ from other computing systems in that they are not only multitasking but also multi-user.

What exactly does this mean? It means that more than one user can be operating the computer at the same time. While a desktop or laptop computer only has one keyboard and monitor, it can still be used by more than one user. For example, if the computer is attached to a network, or the Internet, remote users can log in via `ssh` (secure shell) and operate the computer. In fact, remote

users can execute graphical applications and have the output displayed on a remote computer. The X Window system supports this.

The multi-user capability of Unix-like systems is a feature that is deeply ingrained into the design of the operating system. If we remember the environment in which Unix was created, this makes perfect sense. Years ago before computers were "personal," they were large, expensive, and centralized. A typical university computer system consisted of a large mainframe computer located in some building on campus and *terminals* were located throughout the campus, each connected to the large central computer. The computer would support many users at the same time.

In order to make this practical, a method had to be devised to protect the users from each other. After all, we wouldn't want the actions of one user to crash the computer, nor would we allow one user to interfere with the files belonging to another user.

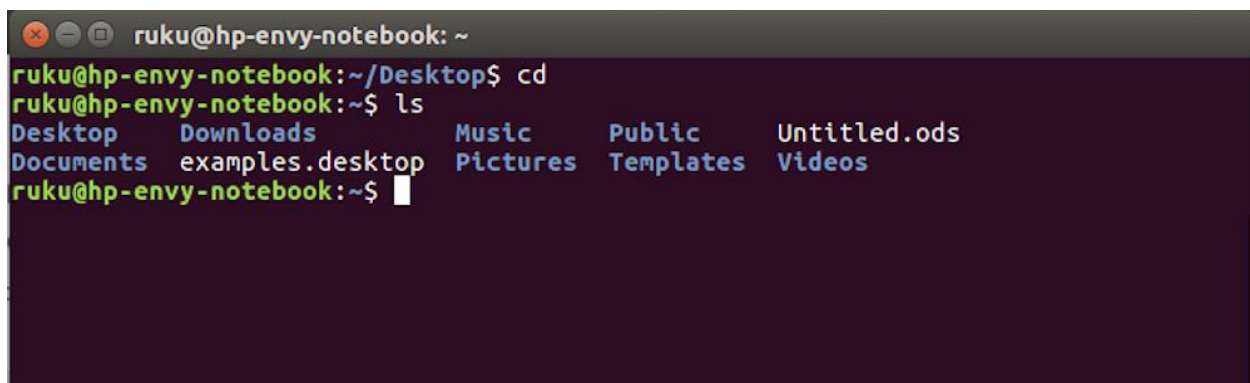
This will cover the following commands:

- **chmod**- modify file access rights
- **su** - temporarily become the superuser
- **sudo** - temporarily become the superuser
- **chown**- change file ownership
- **chgrp** - change a file's group ownership

File Operation in Linux Operating:

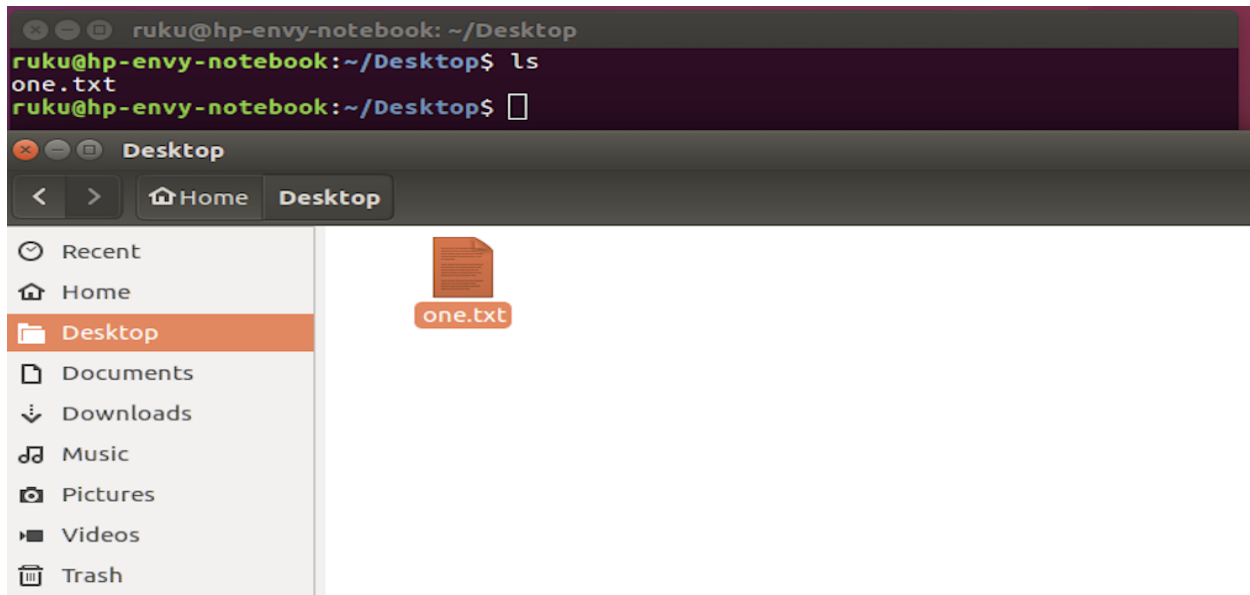
To use the Linux terminal like a pro, we'll need to know the basics of managing files and navigating directories. Different file operation is given below...

ls – The ls command lists the files in a directory. By default, ls lists files in the current directory.

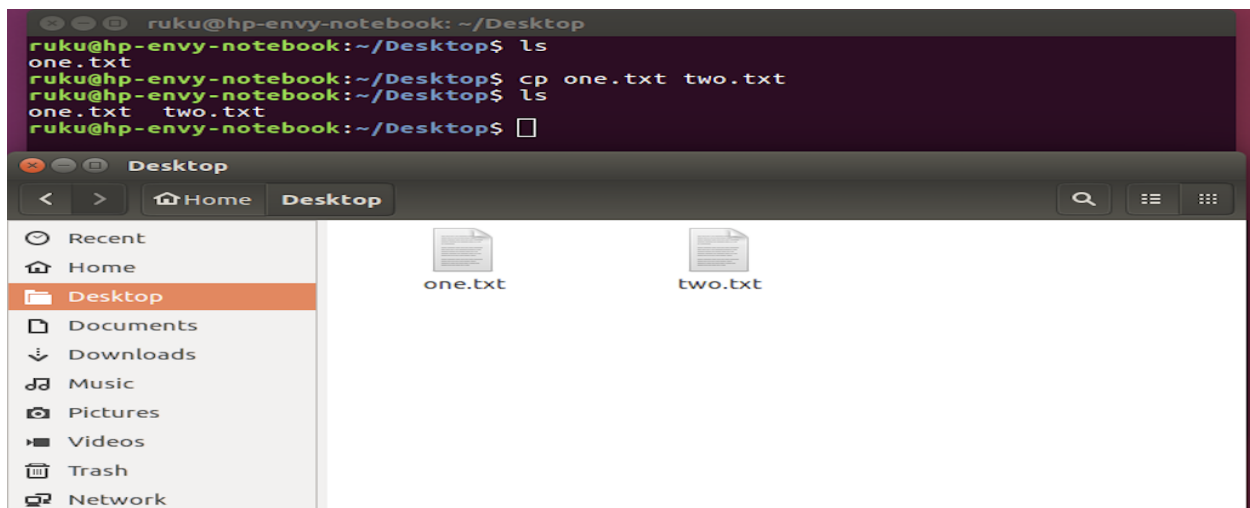


```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~/Desktop$ cd  
ruku@hp-envy-notebook:~$ ls  
Desktop    Downloads  Music      Public    Untitled.ods  
Documents  examples.desktop  Pictures   Templates Videos
```

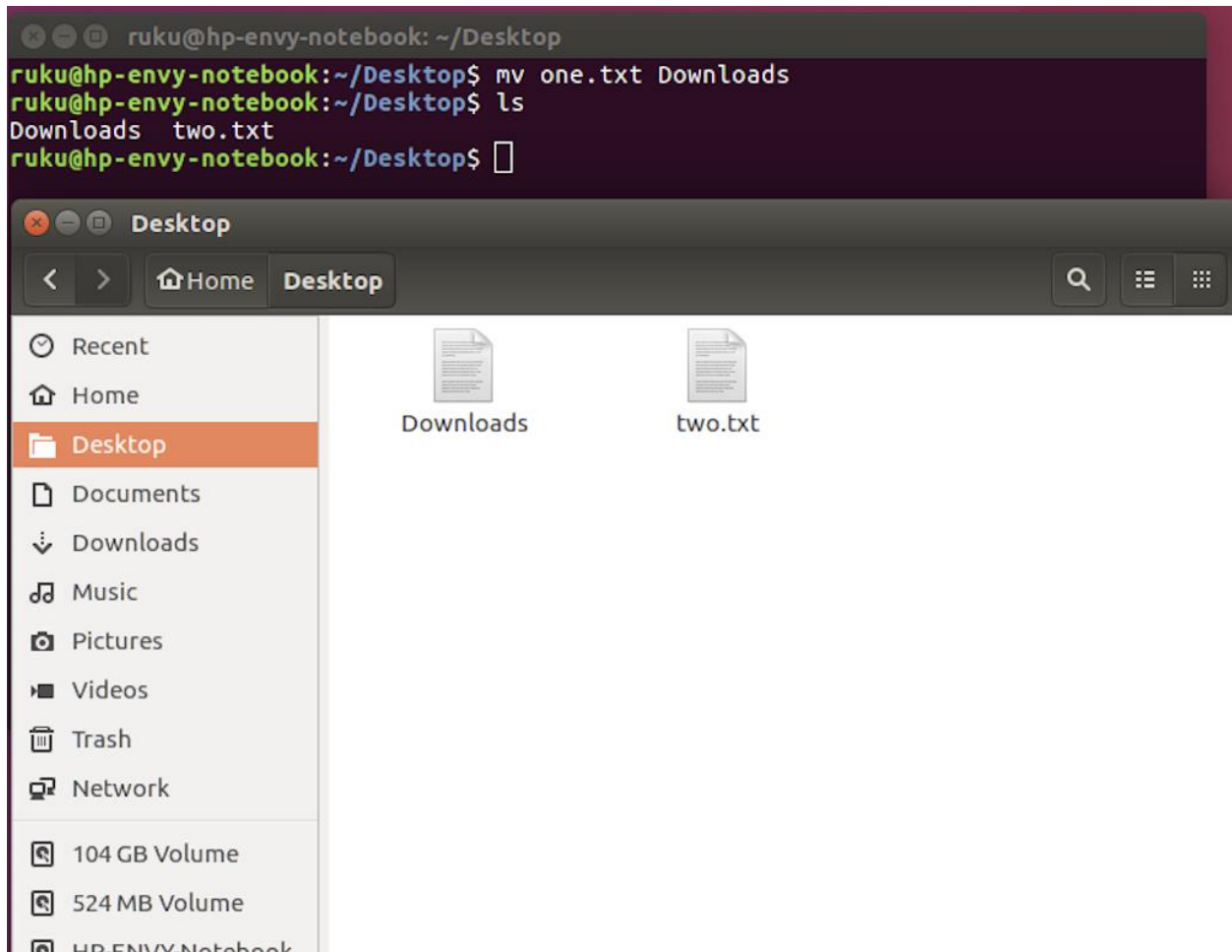
cd – Change Directory The cd command changes to another directory. For example, cd Desktop will take you to your Desktop directory if you're starting from your home directory.

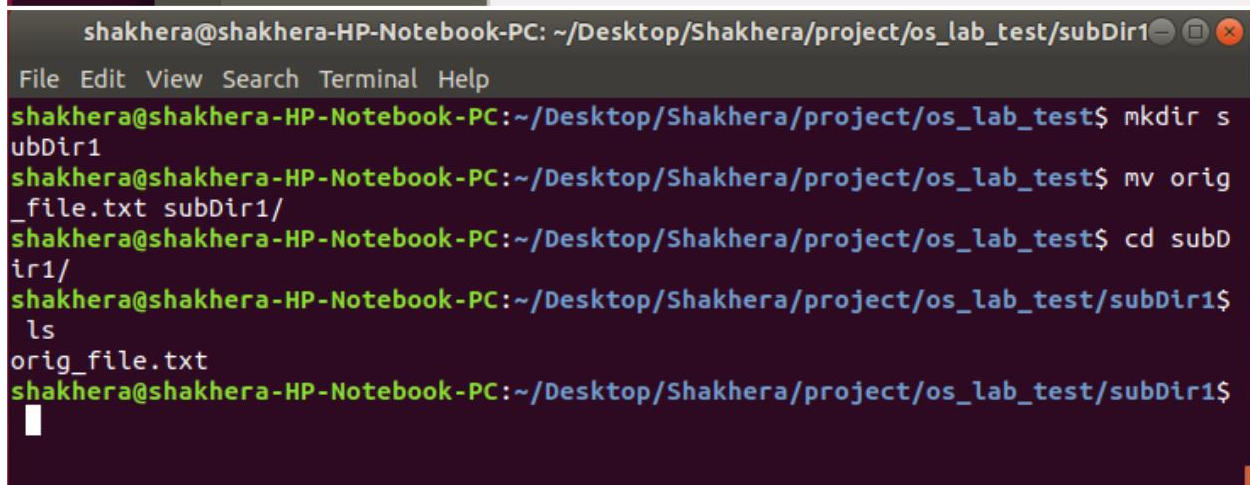
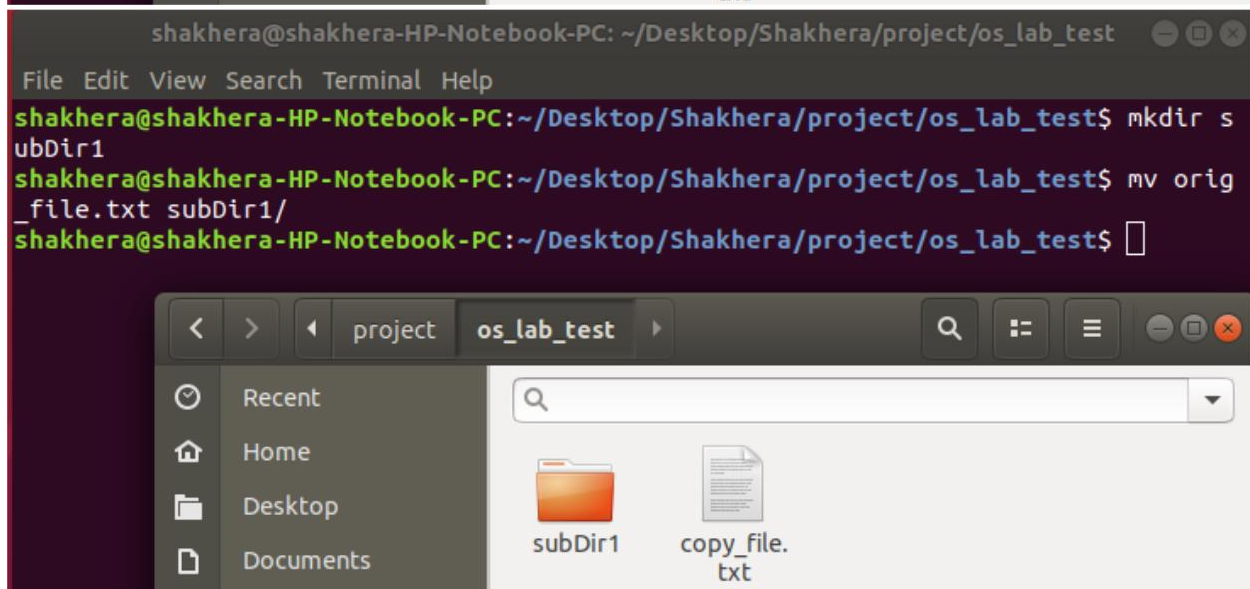
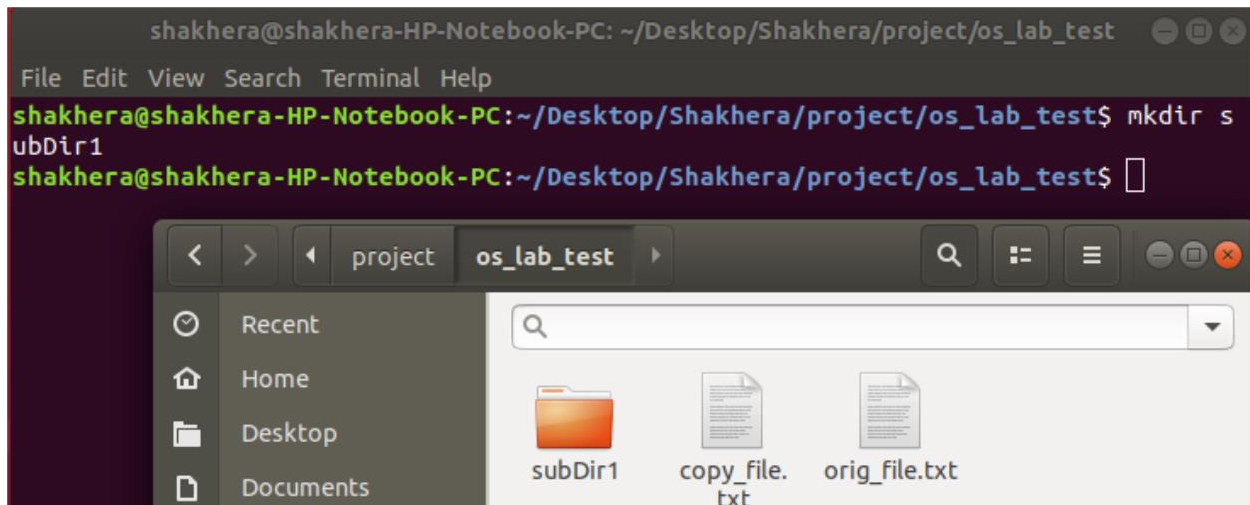


Cp- file1 file2 is the command which makes a copy of file1 in the current working directory and calls it file2.

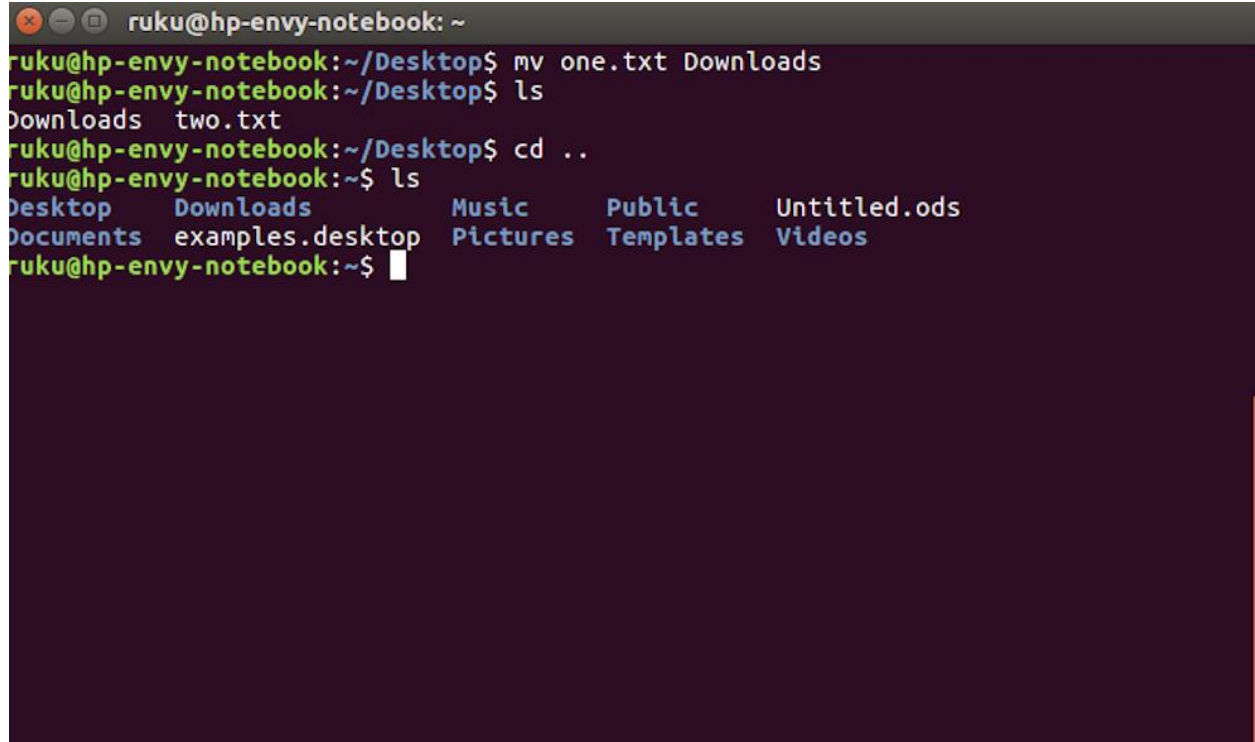


mv- the mv command moves files and directories from one directory to another or renames a file or directory. If you move a file or directory to new directory, it retains the base file name.





`cd ..` will take you up a directory.

A terminal window titled 'ruku@hp-envy-notebook: ~' showing a series of commands and their outputs. The user moves a file from Desktop to Downloads, lists the contents of Desktop, changes to the parent directory, and lists the root directory contents.

```
ruku@hp-envy-notebook:~/Desktop$ mv one.txt Downloads
ruku@hp-envy-notebook:~/Desktop$ ls
Downloads  two.txt
ruku@hp-envy-notebook:~/Desktop$ cd ..
ruku@hp-envy-notebook:~$ ls
Desktop    Downloads      Music    Public    Untitled.ods
Documents  examples.desktop  Pictures  Templates  Videos
ruku@hp-envy-notebook:~$
```

File Permission in Linux Operating:

Linux is a clone of UNIX, the multi-user operating system which can be accessed by many users simultaneously. Linux can also be used in mainframes and servers without any modifications. But this raises security concerns as an unsolicited or malign user can corrupt, change or remove crucial data. For effective security, Linux divides authorization into 2 levels.

- Ownership
- Permission

Every file and directory in your Linux system has following 3 permissions defined for all the 3 owners discussed above.

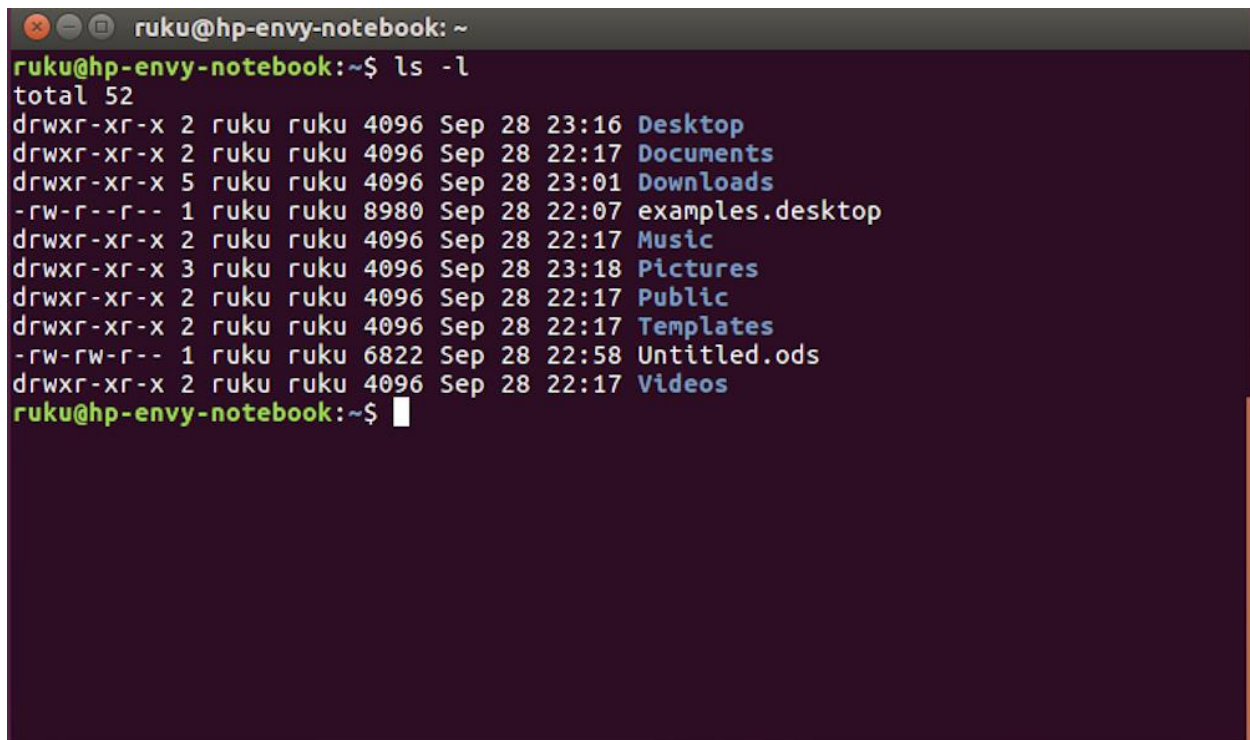
- Read
- Write
- Execute permission

Read (r): This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.

Write (w): The write permission gives you the authority to modify the contents of a file. That's mean to add, remove and rename files stored in the directory.

Execute (x): In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code (provided read & write permissions are set), but not run it.

ls -l



```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ ls -l  
total 52  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 23:16 Desktop  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 22:17 Documents  
drwxr-xr-x 5 ruku ruku 4096 Sep 28 23:01 Downloads  
-rw-r--r-- 1 ruku ruku 8980 Sep 28 22:07 examples.desktop  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 22:17 Music  
drwxr-xr-x 3 ruku ruku 4096 Sep 28 23:18 Pictures  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 22:17 Public  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 22:17 Templates  
-rw-rw-r-- 1 ruku ruku 6822 Sep 28 22:58 Untitled.ods  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 22:17 Videos  
ruku@hp-envy-notebook:~$
```



```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ ls -l  
total 52  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 23:16 Desktop  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 22:17 Documents  
drwxr-xr-x 5 ruku ruku 4096 Sep 28 23:01 Downloads  
-rw-r--r-- 1 ruku ruku 8980 Sep 28 22:07 examples.desktop
```

```
-rw-rw-r-- 1 ruku ruku 6822 Sep 28 22:58 Untitled.ods
```



File type and Access Permissions.

Here , we have highlighted ‘-rw-r—r—’ and this weird looking code is the one that tells us about the permissions gives to the owner, user group and the world. Here the first ‘-’ implies that we have selected a file.p>

```
-rw-r--r--
```

→ indicates file

Else, if it were a directory, d would have been shown.

```
drwxr-xr-x 2 ruku ruku 4096 Sep 28 23:16 Desktop  
drwxr-xr-x 2 ruku ruku 4096 Sep 28 22:17 Documents  
drwxr-xr-x 5 ruku ruku 4096 Sep 28 23:01 Downloads
```

d means represents directory.

The characters are pretty easy to remember.

r = read permission
w = write permission

x = execute permission
= no permission

Let us look at it this way.

The first part of the code is '**rw-**'. This suggests that the owner 'Home' can:

-rw-r--r--



no execute permission

- Read the file
- Write or edit the file
- He cannot execute the file since the execute bit is set to '-'.

By design, many Linux distributions like Fedora, CentOS, Ubuntu, etc. will add users to a group of the same group name as the user name. Thus, a user 'tom' is added to a group named 'tom'.

The second part is '**rw-**'. It for the user group 'Home' and group-members can:

- Read the file
- Write or edit the file

The third part is for the world which means any user. It says '**r--**'. This means the user can only:

- Read the file

Discussion:

From this lab I learn from file handling and file operation,also I learned implementation through linux operating system.

Operating systems control the file access by setting permissions for files and directories. Permissions can be set to grant or deny access to specific files and directories. When permission is granted, you can access and perform any function on the file or directory. When permission is denied, you cannot access that file or directory. The most common permissions are read, write, delete, and execute. File access control functions much like a bank.