

Linux Programming: Assignment-2

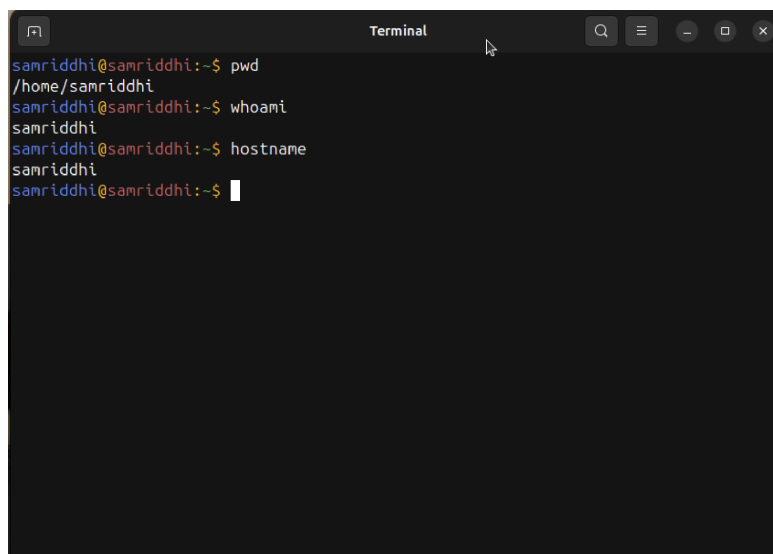
Samriddhi Guha | ENG24CY0157 | 3-C (25)

1. What does the command `pwd`, `whoami`, and `hostname` display? (CO1)

The `pwd` command expands to ‘*present working directory*’. This command displays the directory which the user is currently interacting with.

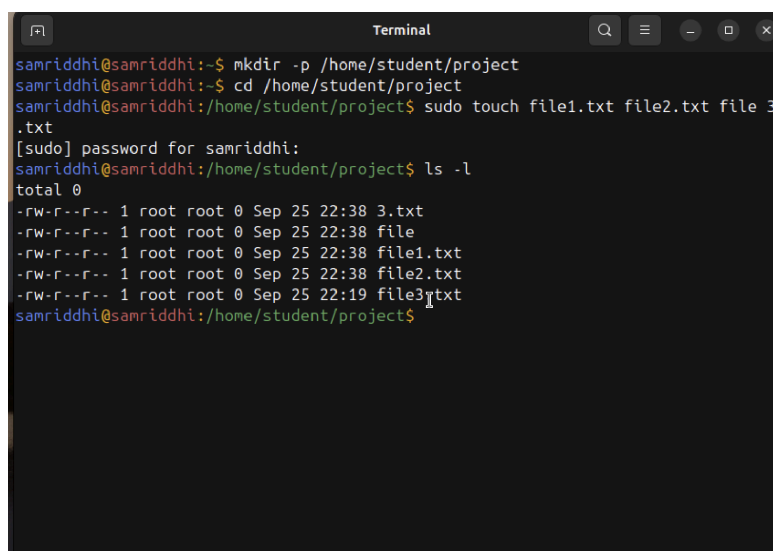
The `whoami` command displays the username of the user who is currently logged in and interacting with the command line.

The `hostname` command displays the network name of the computer that the user is using.



```
Terminal
samriddhi@samriddhi:~$ pwd
/home/samriddhi
samriddhi@samriddhi:~$ whoami
samriddhi
samriddhi@samriddhi:~$ hostname
samriddhi
samriddhi@samriddhi:~$
```

2. Write the command to create a directory named “project” inside the `/home/student` folder and keep three `.txt` file into it. Give output snapshot. (CO1)



```
Terminal
samriddhi@samriddhi:~$ mkdir -p /home/student/project
samriddhi@samriddhi:~$ cd /home/student/project
samriddhi@samriddhi:/home/student/project$ sudo touch file1.txt file2.txt file 3
.txt
[sudo] password for samriddhi:
samriddhi@samriddhi:/home/student/project$ ls -l
total 0
-rw-r--r-- 1 root root 0 Sep 25 22:38 3.txt
-rw-r--r-- 1 root root 0 Sep 25 22:38 file
-rw-r--r-- 1 root root 0 Sep 25 22:38 file1.txt
-rw-r--r-- 1 root root 0 Sep 25 22:38 file2.txt
-rw-r--r-- 1 root root 0 Sep 25 22:19 file3.txt
samriddhi@samriddhi:/home/student/project$
```

The procedure uses the *mkdir* command that is used to create a directory named “student”. This is like creating a folder on our computer storage. As we are in the *home directory by default*, we need to get inside the newly created “student” directory so we use *cd* to change the directory to that directory. The *touch* command is used to create a new file inside a directory, which in this case are 3 text files. Finally, in order to check if these files are created, the *ls -l* command is used which lists all the file names along with the date and time of their creation.

3. Explain the difference between absolute path and relative path with proper examples. (CO2)

Absolute Path:

- An *absolute path* gives the *full location* of a file or directory starting from the *root directory (/)*.
- It's like giving someone the *full address* of your house (country → city → street → house number).
- No matter where you are in the system, the absolute path will always point to the same place.

Ex: */home/student/project/file.txt*

Starts from root (/) and is followed by home -> student -> project -> file.txt

Relative Path:

- A relative path specifies the position of a file or directory in relation to your present working directory (pwd).
- The result is relative to where you are in the directory tree.

Ex:

/home/student/

cd project

project/file.txt

Here, project is the relative path as it is relative to our current position.

4. What command will give you the already executed command traces in the terminal. Give output snapshot. (CO1)

The following commands can be used to check the executed command traces on a Linux CLI:

- The *history* command prints the list of commands executed in the current shell session (and also reads from the shell history file).
- The *file ~/.bash_history* command stores past commands between sessions.
- To include timestamps, we use the set HISTTIMEFORMAT before running commands (or add to ~/.bashrc).

```
Terminal
samriddhi@samriddhi:/home/student/project$ history
1 sudo apt update
2 clear
3 sudo apt install ubuntu-desktop
4 sudo reboot now
5 sudo apt install spice vdagent spice-webdavd
6 sudo apt install spice-vdagent spice-webdavd
7 whoami
8 [200~curl -fsS https://dl.brave.com/install.sh | sh-
9 curl -fsS https://dl.brave.com/install.sh | sh
10 clear
11 whoami
12 ipconfig
13 ifconfig
14 sudo apt install net-tools
15 clear
16 ipconfig
17 ifconfig
18 man ls
19 tldr man ls
20 hi
21 ifconfig
22 wifi password
23 whwho mwho am u
```

5. Compare the working functionality of find and locate command. Which one is faster and why? (CO1)

The *find* command:

- Searches for files and directories in real time by scanning the directory tree.
- Syntax: *find /path-name filename*

The *locate* command:

- Searches for files using a pre-built database.
- Syntax: *locate filename*

Therefore, locate is faster because it uses a pre-indexed database instead of scanning directories in real time. find is more accurate and real-time, but comparatively slower

6. Which command is used to modify file permissions in Linux? Give an example. (CO1)

The *chmod* is the command to change file/directory permissions.

There are two ways to use it: symbolic (u/g/o/a with + - =) OR numeric (octal) (0–7 for each class).

Common permissions that can be modified using this command include:

- r = read (4)
- w = write (2)
- x = execute (1)

7. A file has permissions -rw-r--r-. What does this mean? (CO1)

The file permission string -rw-r--r- clearly indicates how one can access a file in Linux. The given permissions can be broken down as:

(-) : Indicates that the file is a regular file, not a special file or directory.

(rw) [owner] : The owner of the file has read and write permission but cannot run the file.

(r-) [group] : The members of the group can read the file only, they can't write or execute.

(r-) [others] : Other users (everybody else) can also read only.

Numeric (octal) equivalent: 644 → 6 for the owner (read + write), 4 for the group (read only), and 4 for others (read only).

This means that only the owner can change the file while the group members and different users can read it. This is a typical permission setting used for configuration or text files, which still allows and can be read by others, but prevents unintended changes for security reasons.

8. Explain the difference between `chown` and `chgrp` with an example. (CO1)

In Linux, the commands `chown` and `chgrp` allow you to set ownership of files, but they perform different tasks.

`chown` (change owner): Used to change the owner of a file or directory.

`chgrp` (change group): Used to change the group ownership of a file or directory.

A file has two types of ownership in Linux : user (owner) and group.

Syntax:

`chown new_owner filename`

`chgrp new_group filename`

Example: Let's say file `report.txt` is owned by user `samriddhi` and group `student`. To change the owner to `admin`:

`chown admin report.txt`

To change the group to `teachers`:

`chgrp teachers report.txt`

In summary `chown` changes who owns the file, while `chgrp` changes which group has access rights. Both are critical to controlling permissions in a multi-user system.

9. A file needs to be accessible by multiple users but only writable by the owner. How will you set permissions? (CO1)

A file with read permission (are able to read the file) for multiple users but with write permission for the owner should have the right permission string:

- Owner (user): Read + Write → rw-
- Group: Read permission only → r--
- Others: Read permission only → r--

This is rw-r--r--

It will be 644 octal.

Command to set this:

```
chmod 644 filename
```

Thus, the owner will be able to read and write the file. The group and all others will be able to view (read) the file, but they will not be able to write or delete the file. This results in restricted access in a communal setting while protecting the file.

10. How do you check the manual page for any Linux commands? (CO1)

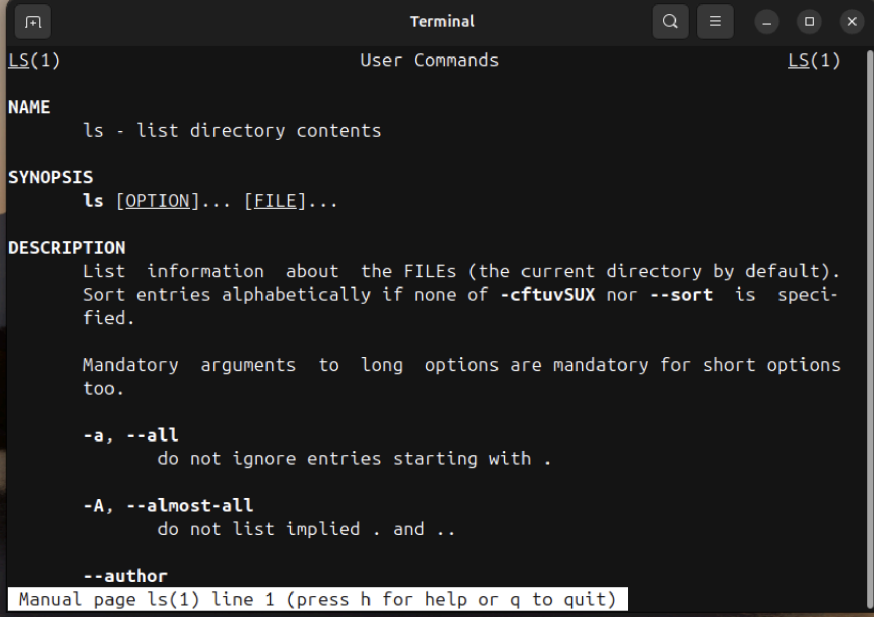
In Linux, there is nearly a manual page (or man page) associated with each command that explains the usage, options, or examples to assist the user. Accessing it is done via the `man` command.

Syntax:

```
man (command_name)
```

Example:

```
samriddhi@samriddhi:~$ man ls
```



```
Terminal
LS(1) User Commands LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
    Manual page ls(1) line 1 (press h for help or q to quit)
```