

# Linux Programming: Assignment-2

**NAME-Abinaya.P USN:ENG24CY0075 ROLL:44**

**Q1) What does the command pwd, whoami, and hostname display?**

**1. pwd (Print Working Directory):**

This command shows the exact path of the directory where we are currently working. For example, if we are inside a folder called *Documents* under your user account, it will display `/home/username/Documents`. It helps the user know their present location in the Linux file system.

**2. whoami:**

This command prints the username of the person who is logged in. It is useful when multiple users are working on the same machine or when switching between accounts.

**3. hostname:**

This command displays the name given to the computer (the host) in the network. For example. It helps in identifying the machine especially when it is connected in a network with many other systems.

**Q2) Write the command to create a directory named “project” inside the /home/student folder and keep three .txt file into it. Give output snapshot.**

Step 1: Create the directory

```
bash ---          mkdir /home/student/project
```

This makes a new folder called project inside /home/student.

Step 2: Move into that directory

```
Bash----          cd /home/student/project
```

Now you are inside the project folder.

Step 3: Create three text files

```
Bash---          touch file1.txt file2.txt file3.txt
```

This will create three empty text files in one command

**Q3) Explain the difference between absolute path and relative path with proper examples.**

**1. Absolute Path**

1. Definition: An absolute path is the complete path of a file or directory starting from the root `/`.

2. It always begins with / and shows the full location in the file system, no matter where you currently are.

Example:

- a) /home/vishwas/project/file1.txt
- b) Here, it starts from the root / → home → vishwas → project → file1.txt.

Use case: Useful when you want to be sure of the exact location of a file, independent of the current working directory.

## 2. Relative Path

1. Definition: A relative path is the location of a file or directory relative to the directory you are currently working in.
2. It does not start with /. Instead, it uses . (current directory) or .. (parent directory).

Example:

- a) Suppose your current location is /home/vishwas.
- b) The relative path to file1.txt inside the project folder will be: project/file1.txt
- c) To go one level up: ../Downloads/file2.txt

Absolute Path	Relative Path
1. Always starts with /	1. Does not start with /
2. Complete location from root	2. Location depends on where you are (current directory)
3. Works anywhere in the system	3. Only works correctly if you are in the expected directory
4. Example: /home/vishwas/project/file1.txt	4. Example: project/file1.txt

**Q4) What command will give you the already executed command traces in the terminal. Give output snapshot**

In Linux, the command used is:

Bash programming command ----- history

1. It shows a numbered list of all the commands that have been executed in the current shell session (and also from previous sessions, depending on settings).
2. You can re-run a command directly using `!  
list.`

**Q5) Compare the working functionality of find and locate command. Which one is faster and why?**

**ANS--- 1.find command**

1. Functionality: Searches for files and directories by walking through the filesystem in real time.
2. It can search by name, type, size, permission, owner, date, etc.
3. Works even if the file was just created a few seconds ago.
4. Example:
5. `find /home/vishwas -name "file1.txt"`

→ This will search the entire /home/vishwas directory for file1.txt.

**2.locate command**

1. Functionality: Searches files by checking a pre-built database (mlocate.db) instead of scanning the disk live.
2. Faster because it does not look at the actual filesystem each time.
3. But if the database is not updated (sudo updatedb), newly created files may not appear.
4. Example:
5. `locate file1.txt`

→ This will instantly display the path of file1.txt (if it is already recorded in the database).

**Q6)Which command is used to modify file permissions in Linux? Give an example.**

- The command used is `chmod` (change mode).
- It allows you to set or change the read (r), write (w), and execute (x) permissions for the owner, group, and others on a file or directory.

Example 1: Using symbolic method

```
chmod u+x file1.txt
```

This gives the user (owner) execute permission on file1.txt.

Example 2: Using numeric (octal) method

```
chmod 755 file1.txt
```

This sets permissions as:

- 7 → owner = read + write + execute (rwx)
- 5 → group = read + execute (r-x)
- 5 → others = read + execute (r-x)

**Q7) A file has permissions -rw -r- -r- -. What does this mean ?**

In Linux, file permissions are shown in 10 characters. It means like this:

Bash command is -rw-r--r--

1. First character (-)
  1. Shows the type of file.
  2. -= regular file
2. Next 3 characters (rw-) → Owner (User) permissions
  1. r = read allowed
  2. w = write allowed
  3. -= no execute permission
3. Next 3 characters (r--) → Group permissions
  1. r = read allowed
  2. --= no write, no execute

**Q8) Explain the difference between chown and chgrp with an example ?**

1. chown (Change Owner)
  1. Used to change the owner of a file or directory.

Syntax:

```
chown new_owner filename
```

Example:

- sudo chown vishwas file1.txt

2. chgrp (Change Group)

1. Used to change the group ownership of a file or directory.

Syntax:

chgrp new\_group filename

Example:

- sudo chgrp student file1.txt

**Q9) A file needs to be accessible by multiple users but only writable by the owner. How will you set permissions?**

Requirement:

1. Owner → should have *read and write* access.
2. Group & Others (all users) → should have *read-only* access.
3. No one except the owner should be able to modify the file.

Using chmod (symbolic method)

**Bash command---- chmod 644 filename**

1. 6 → Owner = read (4) + write (2) = rw-
2. 4 → Group = read only = r--
3. 4 → Others = read only = r--

final permission will be:

**-rw-r--r--**

**10. How do you check the manual page for any Linux commands?.**

1. In Linux, almost every command has a manual (man) page that explains what the command does, its syntax, options, and examples.
2. To open the manual page, we use the man command followed by the command name.

**Syntax:**

**man <command>**

**Example:**

**man ls**