LINUX PROGRAMMING

Assignment-2

Rohith reddy

1.What does the command pwd, whoami, and hostname display?

1. pwd – Print Working Directory

- **Purpose**: Shows the **full path** of the current directory you are in.

EX :/home/username/Documents

2. whoami – Who Am I

- **Purpose**: Displays the **username** of the currently logged-in user

EX : username

3. hostname – Host Name

- **Purpose**: Displays the **name of the system** (the computer's network name).

EX : my-computer

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

1. mkdir /home/student/project **(creates a new directory named project inside /home/student)**
2. touch /home/student/project/file1.txt /home/student/project/file2.txt /home/student/project/file3.txt **(creates three empty text files inside the `project` directory.)**

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

1. Absolute Path in Linux

An absolute path in Linux is the full path to a file or directory starting from the root directory /.

🔷 Characteristics:

- Always begins with a /
- Independent of your current working directory
- Points to the same file no matter where you run it from

📌 Example:

/home/student/documents/report.txt

2. Relative Path in Linux

A **relative path** is the path to a file or directory **relative to your current working directory**.

🔷 **Characteristics:**

- Does **not** start with /
- Depends on your **current location** in the file system

- Shorter and more flexible in scripts or terminal use

📌 **Example:**

Assume your current directory is:

/home/student/

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

In Linux, the command used to **view the history of already executed commands** in the terminal is:

history

Explanation:

- The history command shows a list of previously executed commands in the current shell session.
- It reads from a hidden file called ~/.bash_history (for Bash shell).
- Each command is displayed with a number (line ID), which you can use to re-run with !<number>.

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

Both find and locate are Linux commands used to **search for files and directories**, but they work in **different ways** and have **different performance characteristics**.

1. find Command

🔹 **Functionality:**

- Searches files and directories **in real-time** by walking through the actual directory tree.
- Can search based on name, type, size, modification date, permissions, etc.
- Syntax:
- find [path] [options] [expression]

  Example:

  find /home/user -name "report.txt"

How it works:

- It **physically scans** the file system.
- Always returns the most **up-to-date** results.

2. locate Command

🔹 **Functionality:**

- Searches files **using a pre-built index database** (/var/lib/mlocate/mlocate.db).
- Much faster because it doesn't access the actual file system at runtime.

- Syntax:
- locate [filename]

Example:

locate report.txt

How it works:

- Uses a database built by the updatedb command.
- The database may be outdated if updatedb hasn't been run recently.

- `locate` **is faster** because it searches a **cached index**, not the actual file system.
- `find` **is slower**, but it is more **accurate and flexible** for real-time and advanced searches.

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

Command to modify file permissions in Linux:

chmod

Example:

chmod 755 myfile.txt

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

The permission string -rw-r--r-- means:

```
-  rw-  r--  r--
|  |   |   |
|  |   |   └── Others: read only
|  |   └────── Group: read only
|  └────────── Owner: read and write
└──────────── File type ( - means regular file)
```

8.Explain the difference between chown and chgrp with an example.

In Linux, both chown and chgrp are used to **change ownership** of files and directories, but they affect **different types of ownership**.

1. chown – Change File Owner (and Group)

- Used to change the **owner** of a file.
- Can also change the group **at the same time**.

Syntax:

chown [new_owner] filename

Example:

chown alice report.txt

- Changes the **owner** of report.txt to user alice.

2. chgrp – Change Group Ownership

- Used to change the **group** associated with a file.
- Cannot change the user (owner).

Syntax:

chgrp [new_group] filename

Example:

chgrp developers report.txt

- Changes the **group** of report.txt to developers.

9.A file needs to be accessible by multiple users but only writable by the owner.

How will you set **permissions**?

To make a file **readable by multiple users** but **writable only by the owner**, you should set the permissions like this:

-rw-r--r—

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

To check the manual page (detailed help and documentation) for any Linux command, use the man command followed by the command name.

Syntax:

man [command_name]