LP ASSIGNMENT-2

ENG24CY0159

SHADIK KHAN

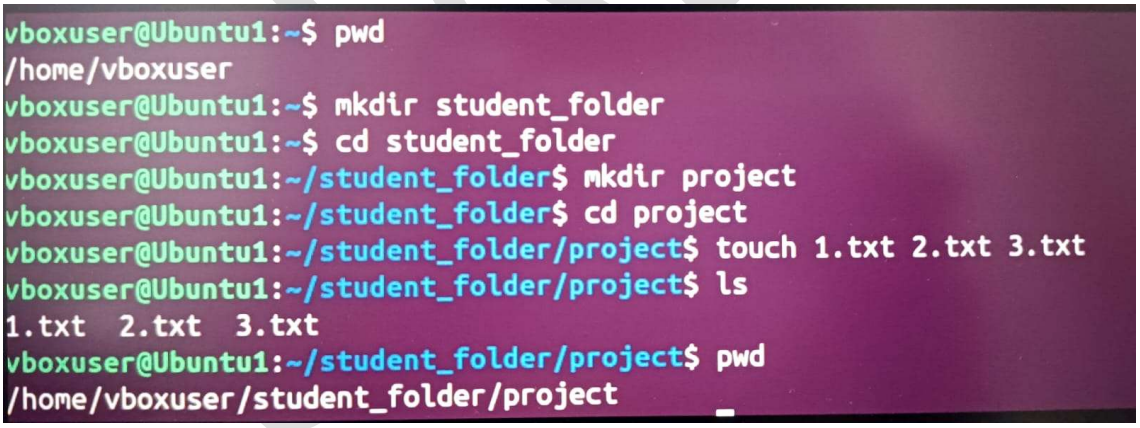
1. What do the commands pwd, whoami, and hostname display?

Ever get lost in the terminal and wonder, "Where am I? Who am I?" These three commands are your best friends for getting your bearings.

- **pwd (Print Working Directory):** This just tells you exactly which folder you're currently in. Think of it as the "You Are Here" dot on a mall map.
- **whoami:** This is like looking at your user ID badge. It prints out your current username, plain and simple.
- **hostname:** This one just shows you your computer's network name.

---

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.



---

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

Think of it like giving directions to a friend.

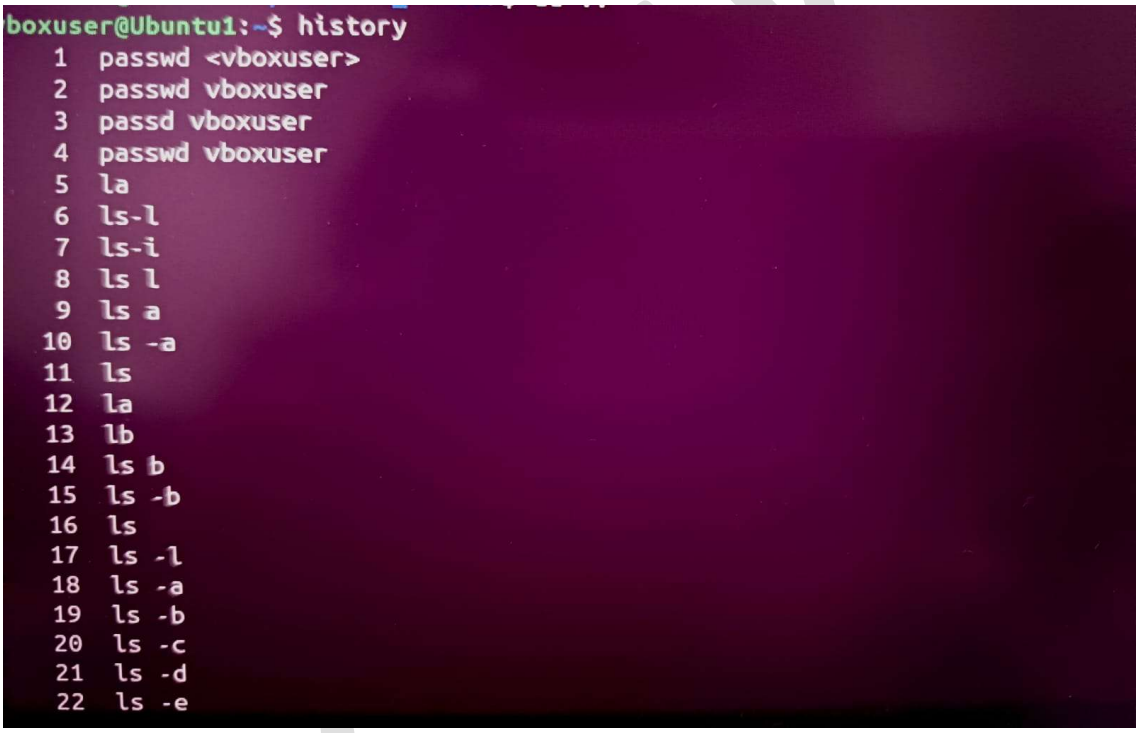
An **Absolute Path** is the full GPS address: /home/student/project/report.txt. It starts from the very root (/) of the system and gives every single turn to get to the file. You can't get it wrong, and it works from anywhere.

A **Relative Path** is when you say, "it's the 'project' folder, and the 'report.txt' file is inside." That only works if your friend is already standing in the /home/student directory with you. It's shorter and more convenient for things that are nearby.

---

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

Forgot the command you ran five minutes ago? Happens to everyone. The history command is your savior. It's literally a log of all the recent commands you've typed.



---

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

Imagine you're looking for a book in a massive library.

- **find** is like you personally walking up and down every single aisle, checking every shelf. It's 100% accurate and will find a book someone just put back, but it takes forever.

- **locate** is like using the library's computer catalog. You type in the name, and it instantly tells you where the book is.

So, locate is way, way faster. 🚀 Why? Because looking something up in an index is always faster than searching the entire place from scratch.

The only catch is that the library's catalog is only updated at night. So if you're looking for a book someone put on the shelf 10 minutes ago, locate won't know about it yet, but the slow-and-steady find command will.

---

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

In Linux, you need to give files permission to do things. The command to do this is chmod (short for **ch**ange **mod**e). It’s like being the bouncer for your files.

For example, say you write a little script called backup.sh. Right now, it's just a plain text file. To give it the 'green light' to actually run, you’d do this:

Bash

chmod u+x backup.sh

That tells the system to give the **u**ser (you) permission to e**x**ecute (+x) that file.

---

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

Let's decode that secret message. It's actually three sets of instructions for three groups of people: the **Owner**, the **Group**, and **Everyone Else**.

- First, the - at the beginning just means it's a regular file.
- rw- (Owner): You, the owner, can **r**ead it and **w**rite (change) it.
- r-- (Group): Anyone in the file's group can only **r**ead it. They can look, but not touch.
- r-- (Others): Same for everyone else on the system. They can **r**ead it, but that's all.

Basically: **The owner can do anything, but everyone else is in read-only mode.**

---

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

Think of a file like a work document. There's always one person who is the main **owner** (chown) and a **team** or group (chgrp) that it belongs to.

- **chown (change owner)** is what you use when you need to hand over ownership of the document to a new person entirely.
- **chgrp (change group)** is what you use when the document needs to be moved to a different team, like from 'Marketing' to 'Sales'.

So if a file draft.txt belongs to user bob, you'd use chown alice draft.txt to make alice the new owner. If it belonged to the writers group, you'd use chgrp editors draft.txt to move it to the editors group.

---

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

This is a classic situation: You have a file you need to edit, but you want your teammates to be able to see it without messing it up.

The magic number for this is **644**. Here's the command:

Bash

chmod 644 filename.txt

It's a simple code:

- The owner gets 6 (Read + Write).
- The group gets 4 (Read-only).
- Everyone else gets 4 (Read-only).

---

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

Stuck and don't want to Google? Linux has a built-in instruction book for every command, and it's called man (short for **man**ual). It's the original "right-click and ask for help."

Just type man and then the command you're curious about. For example:

>Bash

>man ls

This will pull up the official guide, telling you everything that command can do. It's an absolute lifesaver.
