Activity No. 11	
Command Line Skills	
Course Code: CPE 201A	**Program:** Computer Engineering
Course Title: COMPUTER SYSTEM ADMINISTRATION AND TROUBLESHOOTING	**Date Performed:** October 23, 2025
Section: CPE11S1	**Date Submitted:** October 23, 2025
Name: Mendoza, Nathaniel B	**Instructor:** Engr. Jimlord Quejado

1. Objective/s:

This activity aims to execute basic commands using command line interface of Linux.

2. Intended Learning Outcome/s:

The students should be able to:

2.1 Demonstrate how to use commands to explore BASH features.
2.2 Demonstrate how to use commands to display the values of Shell variables.
2.3 Demonstrate how to use quoting in Bash shells.

3. Discussion:

Command Line Interface

The Linux community promotes the CLI due to its power, speed and ability to accomplish a vast array of tasks with a single command line instruction. The CLI provides more precise control, greater speed and the ability to automate tasks more easily through scripting. By learning the CLI, a user can easily be productive almost instantly on ANY flavor or distribution of Linux.

The Shell

Once a user has entered a command , the terminal then accepts what the user has typed and passes to a shell. The shell is a program that enables text based communication between the operating system and the user. It is the command line interpreter that translates commands entered by a user into actions to be performed by the operating system. The Linux environment allows the use of many different shells.
There are several different shells on Linux, these are just a few:

- Bourne-again shell (Bash)
- C shell (csh or tcsh, the enhanced csh)
- Korn shell (ksh)
- Z shell (zsh)

The most commonly used shell for Linux distributions is called the **Bash** shell. When using an interactive shell, the user inputs commands at a so-called prompt. For each Linux distribution, the default prompt may look a little different, but it usually follows this structure:

```
username@hostname current_directory shell_type
```

On Ubuntu or Debian GNU/Linux, the prompt for a regular user will likely look like this:

```
carol@mycomputer:~$
```

The superuser's prompt will look like this:

```
root@mycomputer:~#
```

On CentOS or Red Hat Linux, the prompt for a regular user will instead look like this:

```
[dave@mycomputer ~]$
```

And the superuser's prompt will look like this:

```
[root@mycomputer ~]#
```

Let's explain each component of the structure:

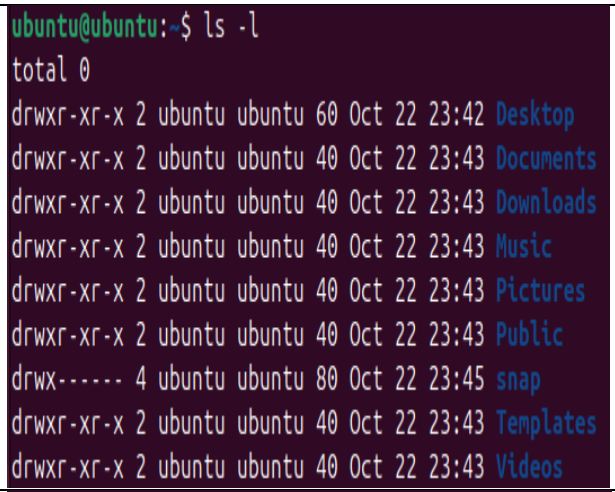
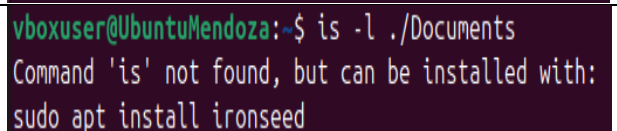
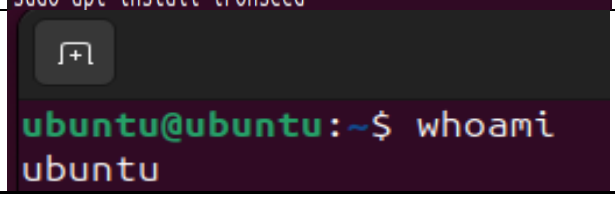
username
Name of the user that runs the shell
hostname
Name of the host on which the shell runs. There is also a command hostname , with which you can show or set the system's host name.
current_directory
The directory that the shell is currently in. A ~ means that the shell is in the current user's home directory.
shell_type
$ indicates the shell is run by a regular user.
# indicates the shell is run by the superuser root

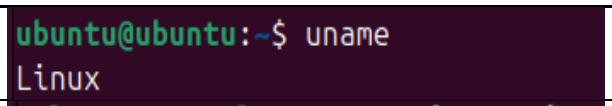
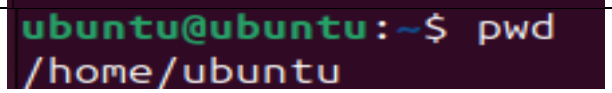
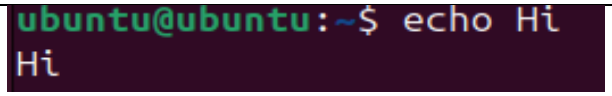
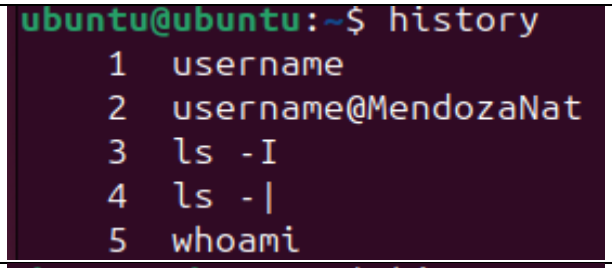
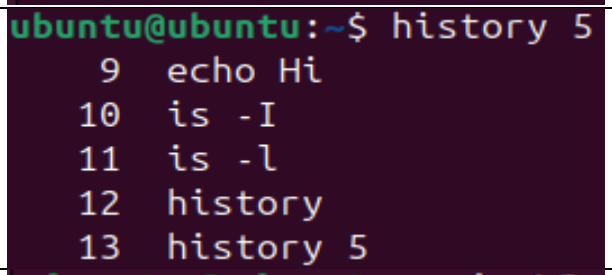
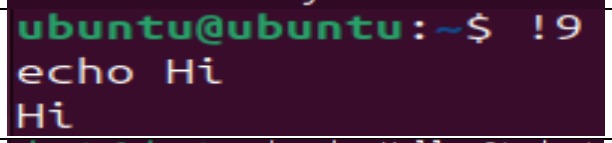
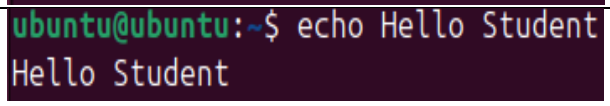
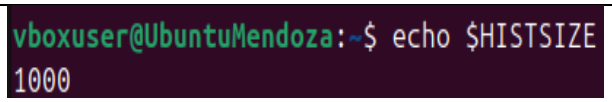
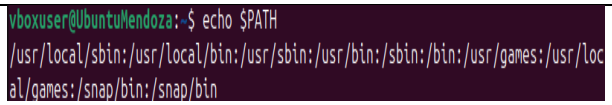
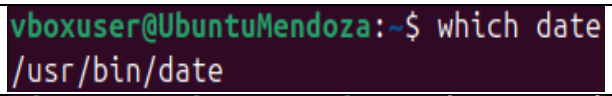
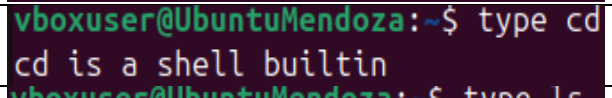
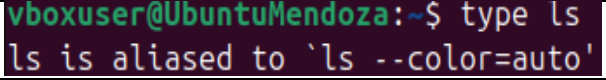
4. Resources:

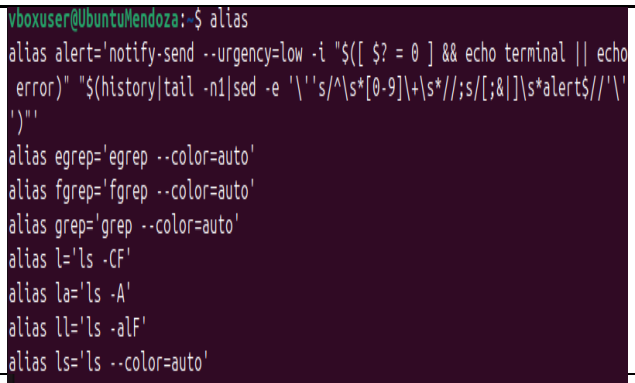
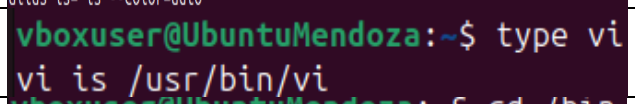
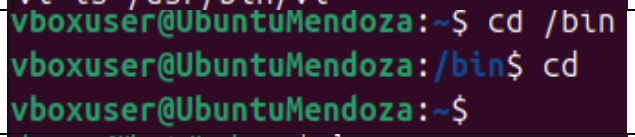
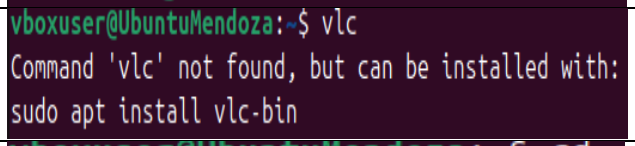
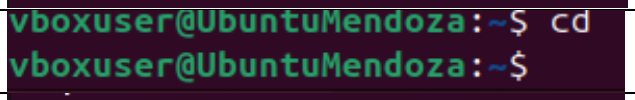
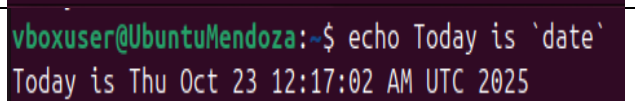
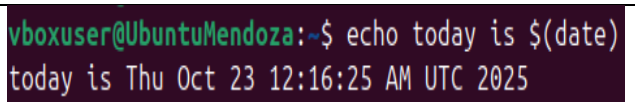
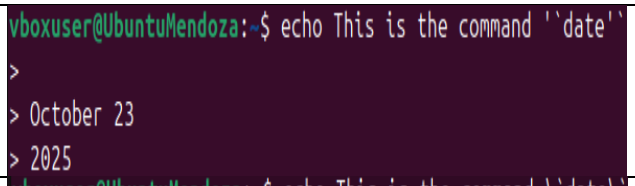
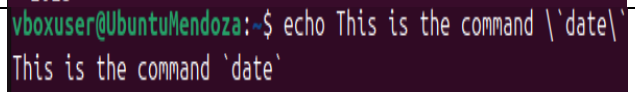
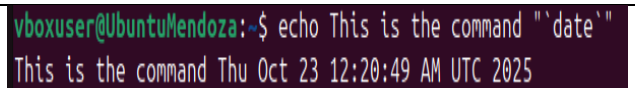
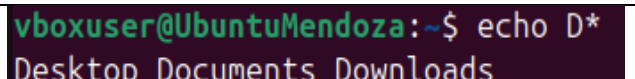
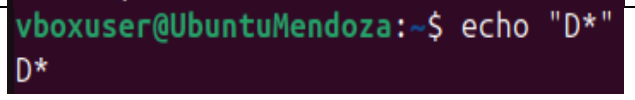
Personal Computer with installed Virtual Box
Ubuntu Server or Desktop virtual machine

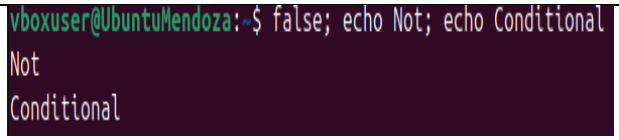
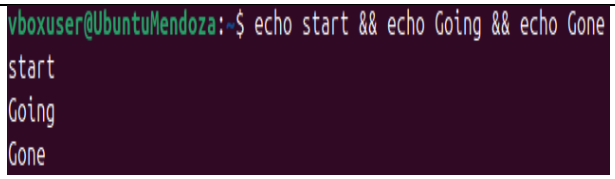
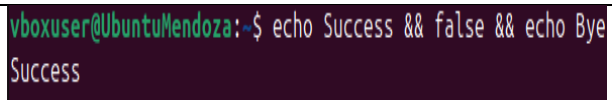
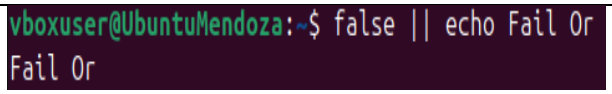
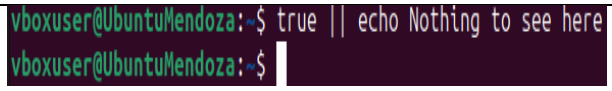
5. Procedure:

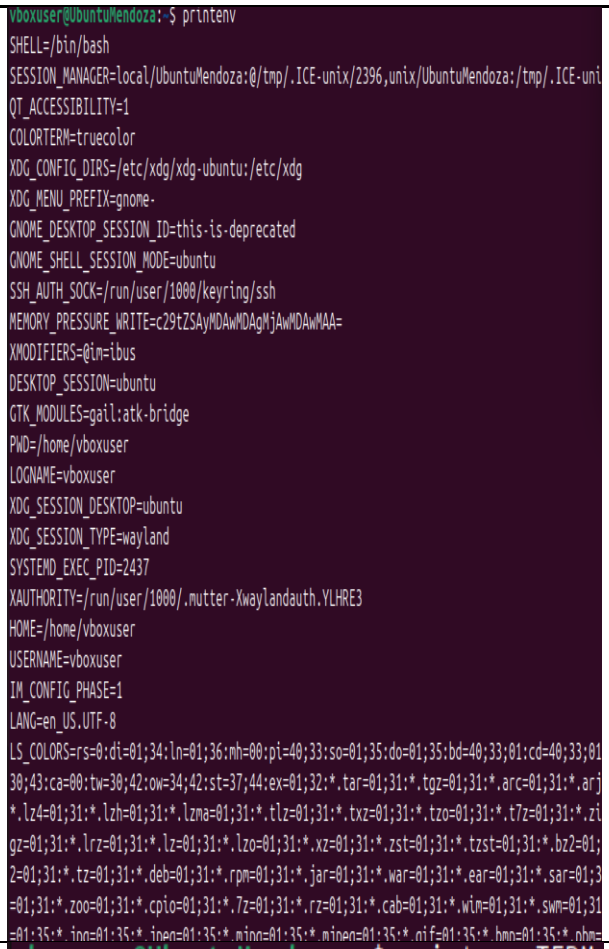
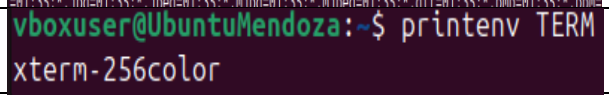
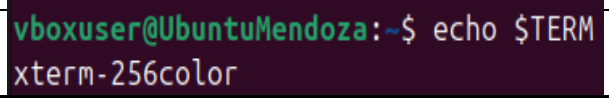
1. Login using your username and password.
2. Use terminal emulator application (if you are using desktop version)
3. Execute the following commands. Copy a screenshot as output after you execute the given command. Create a brief explanation of the command.

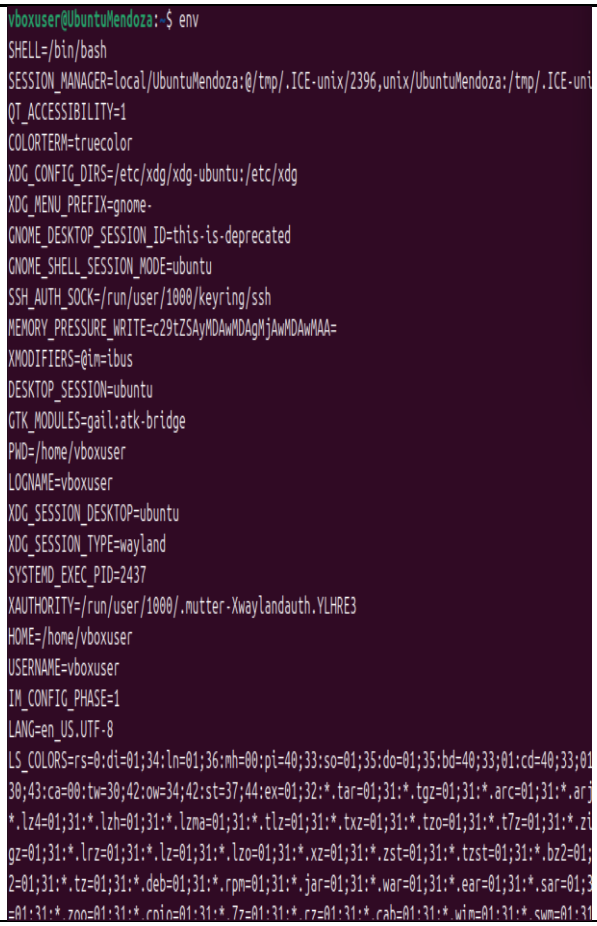
Command	Screenshot	Explanation
1 ls -l		Lists information about files and its directories within the file system.
2. ls -l ./Documents		Lists the files of a given directory. (Given not found as there are no files inside)
3. whoami		It tells which operating system is being used this displays the username of the effective user.

4. Uname		Displays user information.
5. pwd		Displays the path of current directory you are using within the terminal.
6. echo Hi		Echo command is used to display/print a text or strings as the output.
7. history		Provides the list of previously executed commands within the terminal.
8. history 5		Provides the list of a specific "x = numbers" input executed commands within the terminal.
9. !9		Sends a signal to the process and corresponds it.
10. echo Hello Student		Echo command is used to display/print a text or strings as the output.
11. echo $HISTSIZE		Prints the number of lines/commands that are stored in memory in a history list while the terminal/bash is ongoing.
12. echo $PATH		Displays one or more directory names separated by colon.
13. which date		Displays the date directory
14. type cd		Displays the current directory which is the shell
15. type ls		Simply list and displays the contents of current directory

16. alias		Instead of it displaying a very long command it shortcuts as an alternative so that it substituted to a more complex command before its execution
17. type vi		Displays the path for Visual Editor
18. cd /bin		It changes its directory to bin .
19. type vlc		Gives the path of vlc to the media files / mp4.
20. cd		Displays / change the current drive and directory
21. echo Today is `date`		Prints the date in different form with the usage of different special characters
22. echo Today is $(date)		Prints the date in different form with the usage of different special characters
23. echo This is the command \`date\`		Command allows user input
24. echo This is the command \`date\`		Prints both text displayed instead of the “date” printing its date.
25. echo This is the command "`date`"		Prints the text and displays the date.
26. echo D*		Lists all the file types in the current working directory
27. echo "D*"		It prints D* instead of listing all its file types is because it used “”.

28. echo Hello; echo Linux; echo Student		Echo command is used to display/print a text or strings as the output. But this one, when you type a new echo it sets as a new line.
29. false; echo Not; echo Conditional		If it was set as false the command will not print, and echo is used to display/print a text or strings as its output.
30. echo Start && echo Going && echo Gone		This prints if both are true as it used a logical operators and also with the use of echo it will set as a new line.
31. echo Success && false && echo Bye		It prints success and since there is a false function related to the next print which is bye it will not be part of the output.
32. false || echo Fail Or		It used a logical or operators it printed even if it was false because by its definition it returns true if any of the statemen is true.
33. true || echo Nothing to see here		It didn't print because with the use of logic or operators since both are true, it will return as false so it didn't print.

34. printenv			Prints all environmental variables without an argument or an action.
35. printenv TERM			Prints what kind of screen you are currently using.
36. echo $TERM			Displays the screen type of you are currently using.

37.	env		Runs the program in a modified environment and displays its current environments. Basically, this displays all the list of environmental variables

6. Supplementary Activity:

Copy screen shot(s) of the following tasks:

1. An alias can be used to map longer commands to shorter key sequences. Use an alias to represent a very long command.

```
vboxuser@UbuntuMendoza:~$ alias lll='ls -lh /usr/local/bin'
vboxuser@UbuntuMendoza:~$ ls -alh --color=auto
total 76K
drwxr-x--- 15 vboxuser vboxuser 4.0K Oct 23 00:19 .
drwxr-xr-x  3 root     root     4.0K Oct 23 00:05 ..
-rw-------  1 vboxuser vboxuser  772 Oct 23 01:15 .bash_history
-rw-r--r--  1 vboxuser vboxuser  220 Mar 31  2024 .bash_logout
-rw-r--r--  1 vboxuser vboxuser 3.7K Mar 31  2024 .bashrc
drwx------ 10 vboxuser vboxuser 4.0K Oct 23 00:14 .cache
drwx------ 11 vboxuser vboxuser 4.0K Oct 23 00:42 .config
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Desktop
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Documents
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Downloads
drwx------  4 vboxuser vboxuser 4.0K Oct 23 00:05 .local
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Music
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Pictures
-rw-r--r--  1 vboxuser vboxuser  807 Mar 31  2024 .profile
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Public
drwx------  3 vboxuser vboxuser 4.0K Oct 23 00:05 snap
drwx------  2 vboxuser vboxuser 4.0K Oct 23 00:05 .ssh
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Templates
drwxr-xr-x  2 vboxuser vboxuser 4.0K Oct 23 00:05 Videos
vboxuser@UbuntuMendoza:~$ echo "alias ;;='alh --color=auto'">> ~/.bashrc
vboxuser@UbuntuMendoza:~$ source ~/.bashrc
bash: /home/vboxuser/.bashrc: line 118: syntax error near unexpected token `;
bash: /home/vboxuser/.bashrc: line 118: `alias ;;='alh --color=auto''
```

2. Create a new directory in the Documents directory. Rename the directory as CPE_201A _(lastname). Create a new file inside the CPE_201A_(lastname) directory. Rename the file as sample1_lastname.txt. Display the content of the CPE_201A _(lastname) directory by executing one line of command only.

```
vboxuser@UbuntuMendoza:~$ cd ~/Documents
vboxuser@UbuntuMendoza:~/Documents$ mkdir CPE11S1_Mendoza
vboxuser@UbuntuMendoza:~/Documents$ cd CPE11S1_Mendoza
vboxuser@UbuntuMendoza:~/Documents/CPE11S1_Mendoza$ touch sample1_Mendoza
vboxuser@UbuntuMendoza:~/Documents/CPE11S1_Mendoza$ ls ~/Documents/CPE11S1_Mend
za
sample1_Mendoza
```

3. Execute a command to display the working shell.

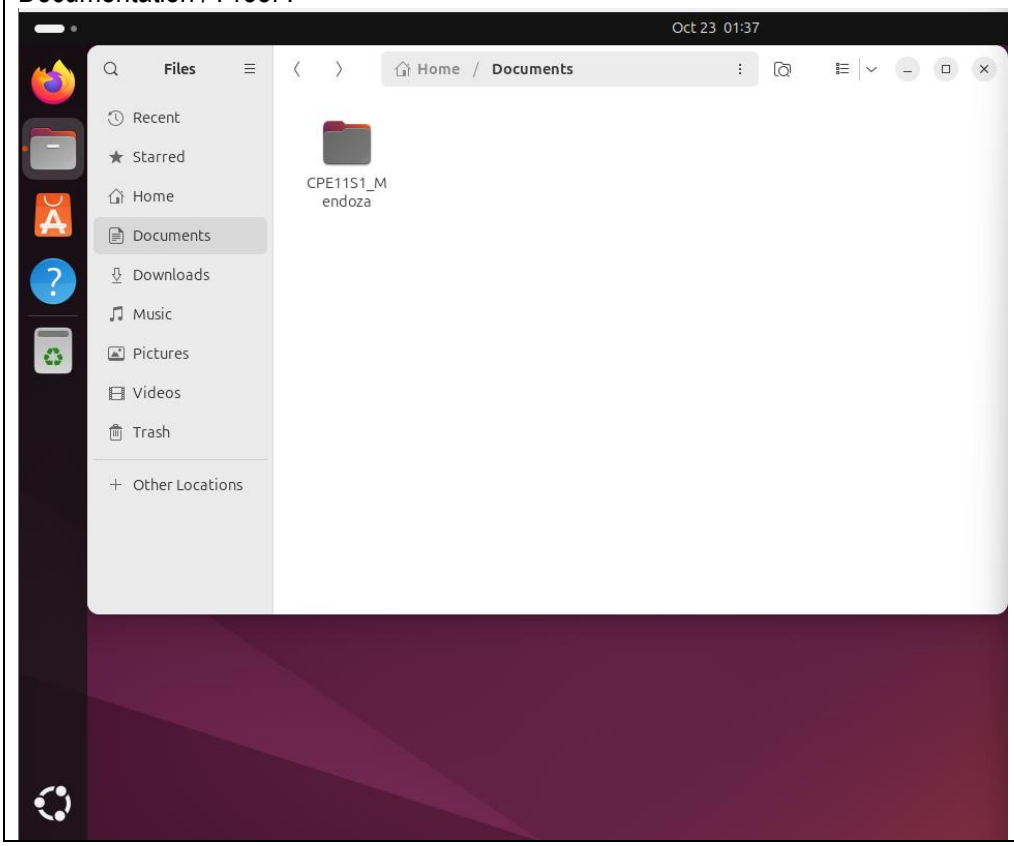
```
vboxuser@UbuntuMendoza:~/Documents/CPE11S1_Mendoza$ echo $SHELL
/bin/bash
```

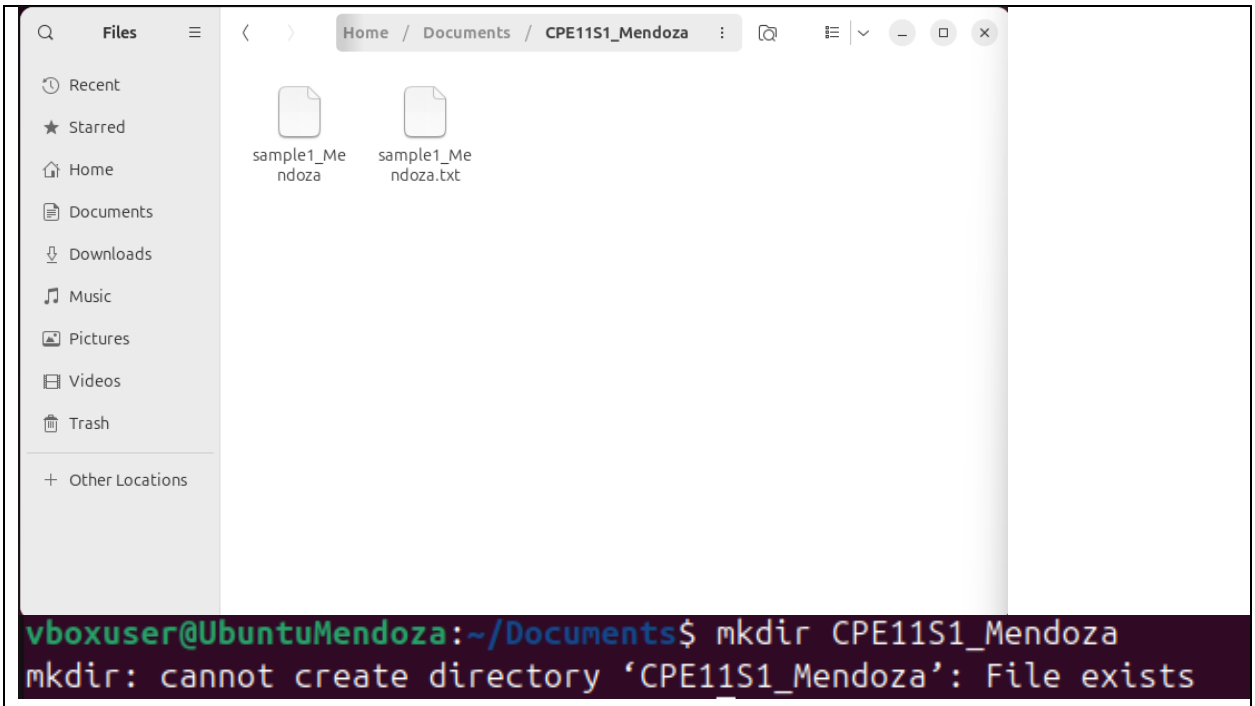
print

4. Shell variables, called environment variables, have the string data type and typically are named with capital letters and the _ (underline) character. Names are case sensitive. The env command will list all the environment variables. The printenv command will list all or will list only the names on its command line. List all environment variables. Which start with P?

```
vboxuser@UbuntuMendoza:~$ env | grep ^P
PWD=/home/vboxuser
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/u
r/local/games:/snap/bin:/snap/bin
```

Documentation / Proof :





7. Conclusion:

After performing the activity, I found out that there are some similarities between the commands on Linux and Windows as there are related commands that works the same as the other one. My main conclusion is that Linux commands are easier based off my performance but there are commands/functions that is sometimes hard to understand or wouldn't understand its function even after typing its action. But I found it flexible as it can do simpler management and navigation so that it can be efficient for experiences users and its command line is very helpful as it consists of different text colors that makes it efficient and easier for the users on using its command line if it were to be comparing on the windows command prompt. Therefore, I can say that it is more user friendly than windows command line as I see it because it sort of looks like for experiences and mastered users only. Although it consists of "similar" commands but some are harder to navigate/understand through. To conclude, the reason why windows is hard is that its for general use while Linux is mainly for system administration and automation. Finally, I was able to apply appropriate techniques and requires skills with the use of communication and research purposes to perform a specific task which is necessary as a Computer Engineer.

8. Assessment (Rubric for Laboratory Performance):