Activity No. 11	
Command Line Skills	
Course Code: CPE 201A	**Program: BSCPE**
Course Title: COMPUTER SYSTEM ADMINISTRATION AND TROUBLESHOOTING	**Date Performed: 10/23/2025**
Section: CPE11S1	**Date Submitted: 10/23/2025**
Name: Ralph Angelov F. Braganza	**Instructor: Engr. Jimlord M. 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:

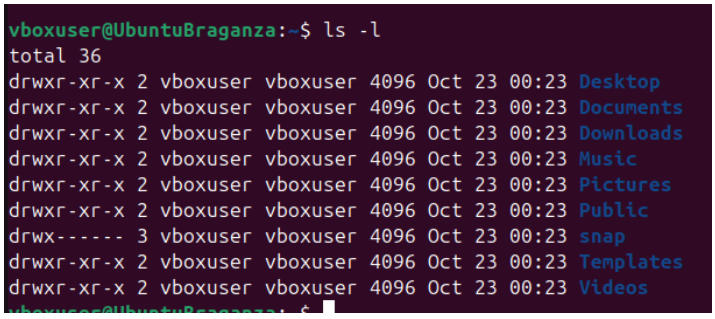
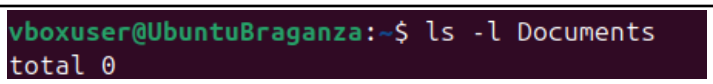
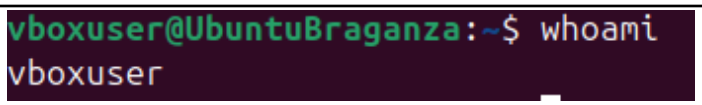
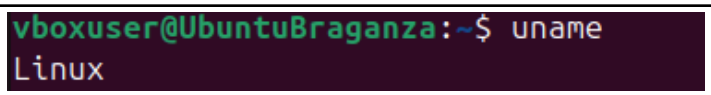
username
Name of the user that runs the shell
hostname
Name of the host on which the shell runs. There is also a command hostname , with which you can show or set the system's host name.
current_directory
The directory that the shell is currently in. A ~ means that the shell is in the current user's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

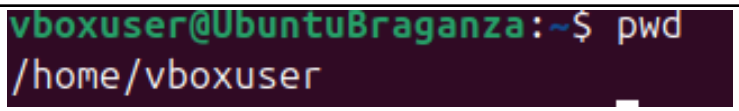
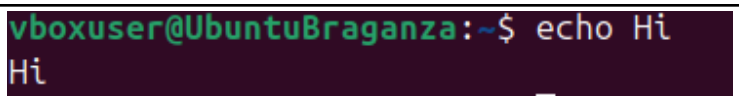
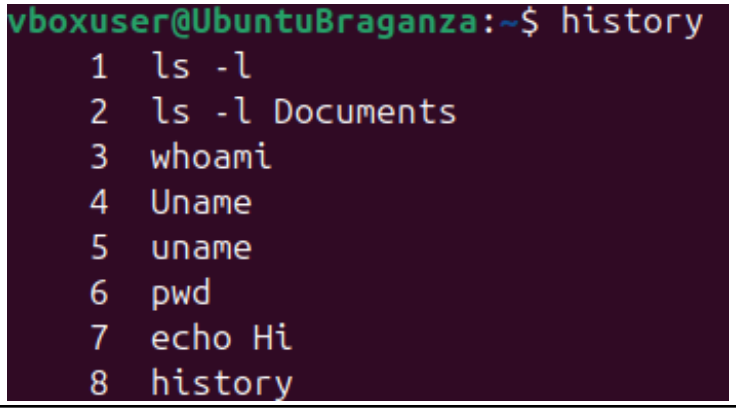
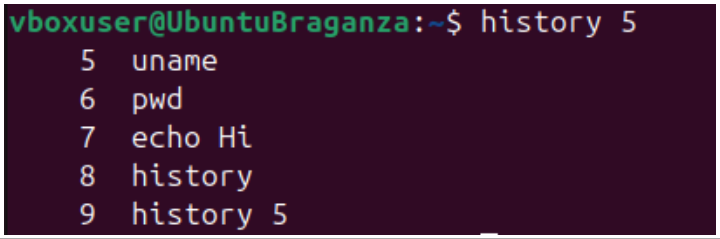
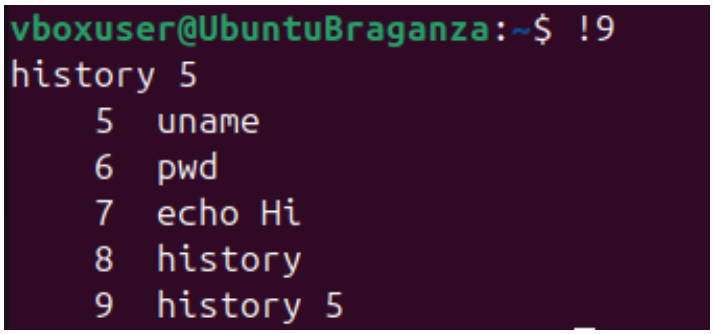
4. Resources:

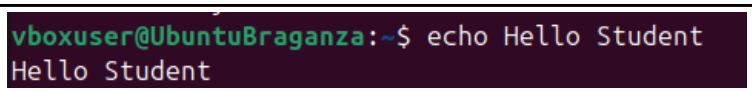
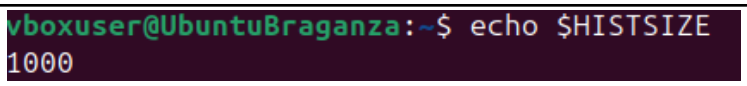
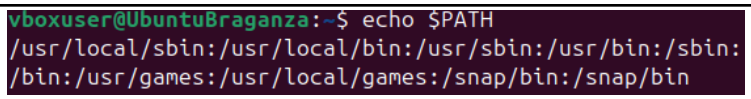
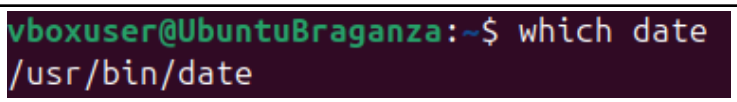
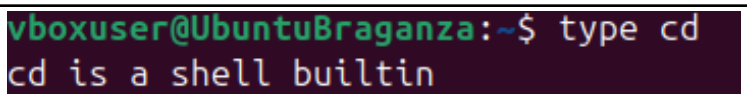
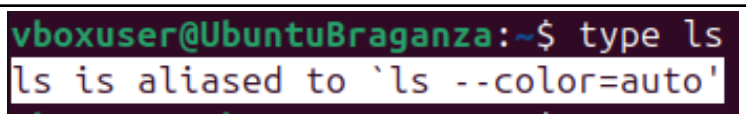
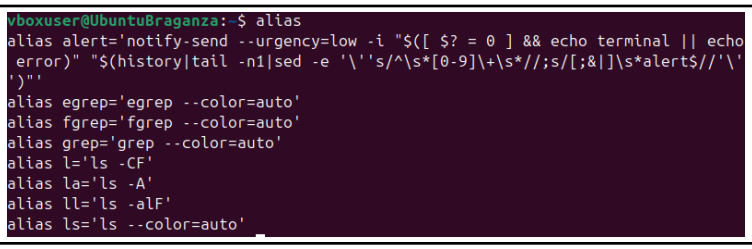
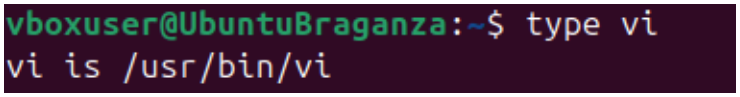
Personal Computer with installed Virtual Box
Ubuntu Server or Desktop virtual machine

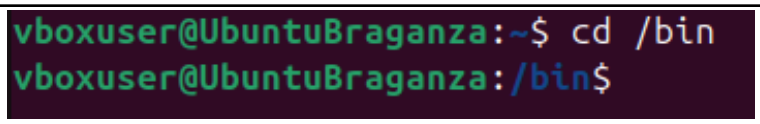
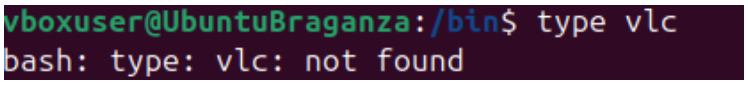
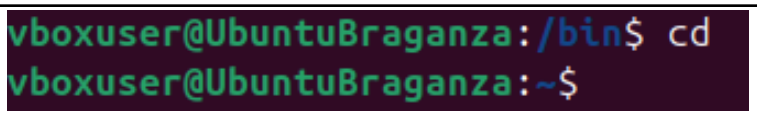
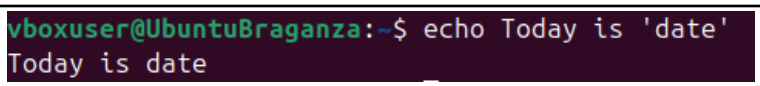
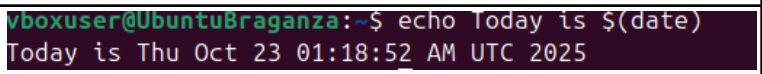
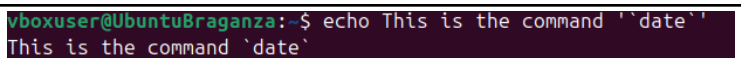
5. Procedure:

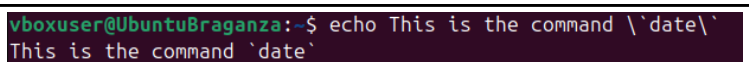
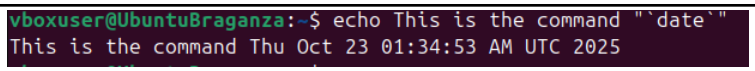
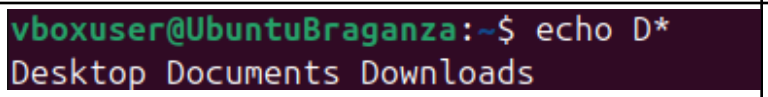
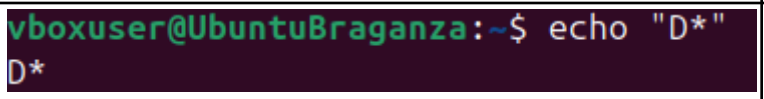
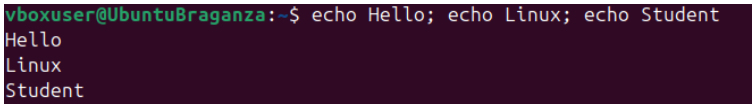
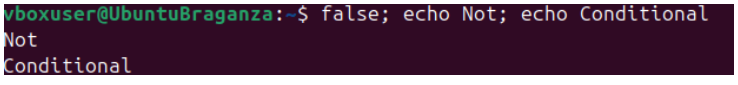
1. Login using your username and password.
2. Use terminal emulator application (if you are using desktop version)
3. Execute the following commands. Copy a screenshot as output after you execute the given command. Create a brief explanation of the command.

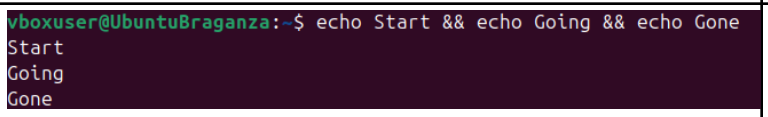
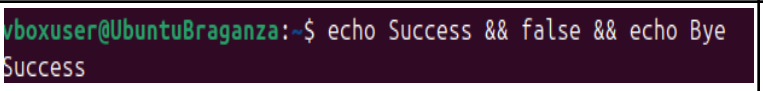
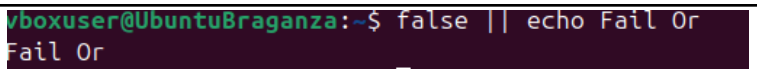
Command	Screenshot	Explanation
1. ls -l		It is used to list information about files and directories within the file system
2. ls -l ./Documents		Lists the contents of your current working directory
3. whoami		Displays the effective username of the current user.
4. uname		Displays system information about the operating system and the

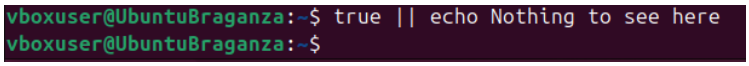
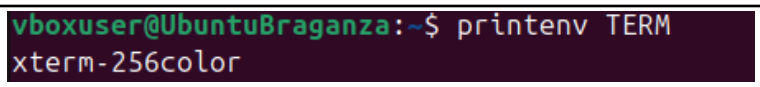
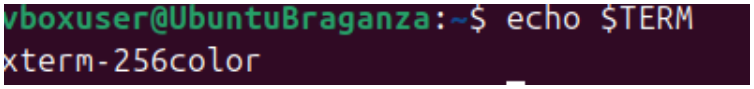
		hardware it is running on.
5. pwd		Displays the full, absolute path of the current working directory (shows you exactly where you are in the file system hierarchy).
6. echo Hi		displays the string "Hi" to the standard output. Essentially, it prints the text you provide as an argument directly onto the screen.
7. history		Displays a list of previously executed commands in the terminal, allowing you to recall, reuse, and edit them.
8. history 5		Displays the last 5 commands that you have executed in your Ubuntu terminal.
9. !9		An attempt to re-run a previous command by its history number

10. echo Hello Student		Displays the text "Hello Student" on the terminal screen and then moves the cursor to the next line.
11. echo $HISTSIZE		This command determines the number of commands that the shell should remember in its history list for the current session.
12. echo $PATH		List of pre-designated directories is stored in a special variable called “PATH”
13. which date		Displays and sets the system date and time.
14. type cd		It is used to display information about a command and in this case it is telling me what the command “cd” does.
15. type ls		It tells you how the ls command is interpreted by your shell. In this case ls is aliased to ‘ls –color=auto’
16. alias		Allows users to create custom shortcuts for longer or more complex commands.
17. type vi		Displays the location and type

		of the vi command.
18. cd /bin		Changes the current working directory in the terminal to the /bin directory.
19. type vlc		Would display information about the vlc (a video player) but it is not installed in this case.
20. cd		It allows users to move from their current working directory to another specified directory.
21. echo Today is `date`		Will print the literal string to the terminal. The ‘ ‘ around the word “date” prevents it from executing the command.
22. echo Today is $(date)		The date command is executed within the echo statement, and its output is displayed.
23. echo This is the command \`date\`		Will print the literal string This is the command ‘date’ because the single quotes prevent the shell from performing the command substitution normally triggered by the backticks.

24. echo This is the command \`date\`		Prints the literal string to the terminal because the backslashes escape the backticks.
25. echo This is the command "`date`"		Prints the literal string "This is the command " followed by the current date and time, as returned by the date command, to the standard output.
26. echo D*		List of all files and directories in the current directory that begin with the letter "D", and then prints that list to standard output.
27. echo "D*"		Prints the literal string "D*" to the standard output, as the double quotes prevent the shell from performing filename expansion
28. echo Hello; echo Linux; echo Student		Executes three separate echo commands sequentially, each printing its respective string ("Hello", "Linux", and "Student") to the terminal on a new line.
29. false; echo Not; echo Conditional		Executes the false command, which always returns a

		non-zero exit status (indicating failure), and then unconditionally executes echo Not and echo Conditional in sequence, printing "Not" and "Conditional" on separate lines.
30. echo Start && echo Going && echo Gone		Executes the three echo commands, printing "Start", then "Going", and "Gone", with each word appearing on its own line.
31. echo Success && false && echo Bye		First prints "Success", and then the entire sequence stops because the second command, false, immediately returns a non-zero (failure) exit status, preventing the final echo Bye from being executed.
32. false || echo Fail Or		First executes the false command, which intentionally returns a failure status, and because the logical OR operator (||) only executes the command that follows it if the

		preceding command fails, it then executes echo Fail Or, printing the text "Fail Or" to the terminal.
33. true || echo Nothing to see here		First executes the true command, which immediately succeeds, causing the shell to ignore and not execute the subsequent echo command due to the nature of the logical OR operator (||)
34. printenv		Displays a list of all environment variables and their values for the current shell session.
35. printenv TERM		Displays the value of the TERM environment variable, which indicates the type of terminal emulation being used
36. echo $TERM		Prints the value of the shell's TERM environment variable, which specifies the type

		of terminal emulation currently in use.
37. env		Executes a program in a modified environment or, when run without arguments, prints a list of all current environment variables and their values.

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@UbuntuBraganza:~$ alias Bob="echo Today is $(date)"
vboxuser@UbuntuBraganza:~$ Bob
Today is Thu Oct 23 02:16:01 AM UTC 2025
vboxuser@UbuntuBraganza:~$
```

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@UbuntuBraganza:~$ cd ~/Documents
vboxuser@UbuntuBraganza:~/Documents$ mkdir CPE_201A_Braganza
vboxuser@UbuntuBraganza:~/Documents$ cd CPE_201A_Braganza
vboxuser@UbuntuBraganza:~/Documents/CPE_201A_Braganza$ touch sample1_Braganza.txt
vboxuser@UbuntuBraganza:~/Documents/CPE_201A_Braganza$ ls
sample1_Braganza.txt
vboxuser@UbuntuBraganza:~/Documents/CPE_201A_Braganza$
```

3. Execute a command to display the working shell.

```
vboxuser@UbuntuBraganza:~/Documents/CPE_201A_Braganza$ echo $0
bash
```

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@UbuntuBraganza:~/Documents/CPE_201A_Braganza$ printenv | grep '^P'
PWD=/home/vboxuser/Documents/CPE_201A_Braganza
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
vboxuser@UbuntuBraganza:~/Documents/CPE_201A_Braganza$
```

7. Conclusion:

Today's activity has shown me a lot of commands that I can execute in CLI. I probably won't be using most of these commands right now, but I'm sure somewhere in the future I'll be utilizing these commands in my own workflow. It was tiring to understand 37 commands and type them manually because I can't copy and paste anything from the host system because it's a virtual machine, but I managed to pull through and understand most of them and how they operate or what they generally do when you execute them. One thing that surprised me is that I never knew I could download things like VLC straight from the command prompt itself. It's pretty amazing how much potential the terminal has once you start to scratch the surface. At first, it might seem overwhelming, but once I get the hang of a few basic commands, I'll eventually get really good at it and work much faster and more efficiently as I go on. I look forward to using Linux even more in the future and exploring more advanced commands in CLI because I am planning to experiment a bit more to see if I am able to shift to Linux as my main OS. The only problem I am having right now is the compatibility of apps that I usually use, but other than that, I have no other doubts about Linux as of now.

8. Assessment (Rubric for Laboratory Performance):

TIP Rubric E (1) (1)

Criteria	Ratings		Pts
Performance Indicators 1. Apply appropriate techniques, skills, and modern tools to perform a discipline-specific engineering task.	**4 pts** **Very Satisfactory** Applies the most appropriate modern technique in performing discipline-specific engineering task exceeding the requirements.	**0 pts** **No Marks**	4 pts
Performance Indicators 2. Demonstrate skills in applying different techniques and modern tools to solve engineering problems.1. Apply appropriate techniques, skills, and modern tools to perform a discipline-specific engineering task.	**4 pts** **Very Satisfactory** Applies the most appropriate modern technique in performing discipline-specific engineering task exceeding the requirements.	**0 pts** **No Marks**	4 pts
Performance Indicators 3. Recognize the benefits and constraints of modern engineering tools.Demonstrate skills in applying different techniques and modern tools to solve engineering problems.1. Apply appropriate techniques, skills, and modern tools to perform a discipline-specific engineering task.	**4 pts** **Very Satisfactory** Applies the most appropriate modern technique in performing discipline-specific engineering task exceeding the requirements.	**0 pts** **No Marks**	4 pts
			Total Points: 12