

Activity No. 11	
Command Line Skills	
Course Code: CPE 201A	Program: BSCpE
Course Title: Computer System Administration and Troubleshooting	Date Performed: October 23, 2025
Section: 11S1	Date Submitted: October 23, 2025
Name: Lopez, Andrei Dion C.	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:	
<p>Command Line Interface</p> <p>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.</p> <p>The Shell</p> <p>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:</p> <ul style="list-style-type: none"> • Bourne-again shell (Bash) • C shell (csh or tcsh, the enhanced csh) • Korn shell (ksh) • Z shell (zsh) <p>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:</p> <p>username@hostname current_directory shell_type</p> <p>On Ubuntu or Debian GNU/Linux, the prompt for a regular user will likely look like this:</p> <p>carol@mycomputer:~\$</p> <p>The superuser's prompt will look like this:</p>	

`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`

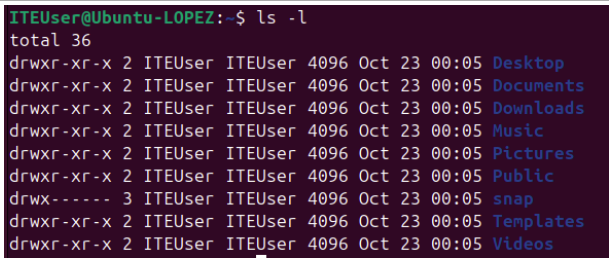
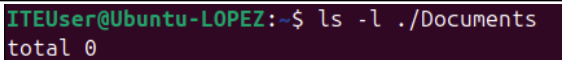
4. Resources:

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. <code>ls -l</code>		Shows a list of the files and directories in its current location, its permissions, size, and dates.
2. <code>ls -l ./Documents</code>		Shows a list of the files and directories in the Documents directory, its

		permissions, size, and dates
3. whoami	<pre>ITEUser@Ubuntu-LOPEZ:~\$ whoami ITEUser</pre>	Displays the current user logged in the virtual machine/computer.
4. Uname	<pre>ITEUser@Ubuntu-LOPEZ:~\$ uname Linux</pre>	Shows the name of the current operating system.
5. pwd	<pre>ITEUser@Ubuntu-LOPEZ:~\$ pwd /home/ITEUser</pre>	Prints the current working directory
6. echo Hi	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo Hi Hi</pre>	Prints out the text written after echo which in this case is Hi.
7. history	<pre>ITEUser@Ubuntu-LOPEZ:~\$ history 1 ~\$ 2 ls -l 3 ls -l 4 ls -l /Documents 5 ls -l ./Documents 6 ls -l 7 ls -l ./Documents 8 ls -l 9 ls -l ./Documents 10 whoami 11 Uname 12 uname 13 pwd 14 echo Hi 15 history</pre>	Shows the list of commands previously executed by the user.
8. history 5	<pre>ITEUser@Ubuntu-LOPEZ:~\$ history 5 12 uname 13 pwd 14 echo Hi 15 history 16 history 5</pre>	Shows the 5 most recent commands previously executed by the user
9. !9	<pre>ITEUser@Ubuntu-LOPEZ:~\$!9 ls -l ./Documents total 0</pre>	Repeats the command executed by the user at the history number 9.
10. echo Hello Student	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo Hello Student Hello Student</pre>	Prints out the text written after echo

		which in this case is Hello Student.
11. echo \$HISTSIZE	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo \$HISTSIZE 1000</pre>	Shows the allowed amount of commands the command line can save.
12. echo \$PATH	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo \$PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin</pre>	echos the current path of the command line
13. which date	<pre>ITEUser@Ubuntu-LOPEZ:~\$ which date /usr/bin/date</pre>	Prints out the current location of the file called "date"
14. type cd	<pre>ITEUser@Ubuntu-LOPEZ:~\$ type cd cd is a shell builtin</pre>	Displays what type the cd is, in this case it is a shell builtin.
15. type ls	<pre>ITEUser@Ubuntu-LOPEZ:~\$ type ls ls is aliased to `ls --color=auto`</pre>	It displays how the command line interprets the command ls.
16. alias	<pre>ITEUser@Ubuntu-LOPEZ:~\$ alias alias alert='notify-send --urgency=low -i "\${[\$? = 0]} && echo terminal echo error)" "\${history tail -n1 sed -e '\''s/^s*[0-9]\+\s*//;s/[:&]\s*alert\$//'\''}' alias egrep='egrep --color=auto' alias fgrep='fgrep --color=auto' alias grep='grep --color=auto' alias l='ls -CF' alias la='ls -A' alias ll='ls -aLF' alias ls='ls --color=auto'</pre>	Allows the user to set a custom command to a keyword frequently used
17. type vi	<pre>ITEUser@Ubuntu-LOPEZ:~\$ type vi vi is /usr/bin/vi</pre>	It displays how the command line would interpret the command vi.
18. cd /bin	<pre>ITEUser@Ubuntu-LOPEZ:~\$ cd /bin ITEUser@Ubuntu-LOPEZ:/bin\$</pre>	accesses the folder called bin in the computer.
19. type vlc	<pre>ITEUser@Ubuntu-LOPEZ:~\$ type vlc bash: type: vlc: not found</pre>	Would display how the command line would interpret vlc if it existed. But currently displays

		"not found", as there is no vlc file.
20. cd	<pre>ITEUser@Ubuntu-LOPEZ:~/bin\$ cd ../../ ITEUser@Ubuntu-LOPEZ:/\$ cd ITEUser@Ubuntu-LOPEZ:~\$</pre>	Returns the user back to the original directory.
21. echo Today is `date`	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo Today is `date` Today is Thu Oct 23 12:30:43 AM UTC 2025</pre>	Prints out the current date that the computer is currently set on.
22. echo Today is \$(date)	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo Today is \$(date) Today is Thu Oct 23 12:31:16 AM UTC 2025</pre>	Prints out the current date that the computer is currently set on using a different command to access the data.
23. echo This is the command `date`	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo This is the command `date` This is the command `date`</pre>	Prints out the text and the command `date` without executing its assigned command.
24. echo This is the command `date`	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo This is the command `date` This is the command `date`</pre>	Prints out the text and the command `date` using " to print out the text without executing its assigned command.
25. echo This is the command "date"	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo This is the command "date" This is the command Thu Oct 23 12:34:34 AM UTC 2025</pre>	Prints out the current date set to the pc using "date" ignoring the quotation marks.
26. echo D*	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo D* Desktop Documents Downloads</pre>	Prints out all the files with the letter D in the name of it.
27. echo "D"	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo "D" D</pre>	echoes the text D which is located in the quotation marks.

28. echo Hello; echo Linux; echo Student	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo Hello; echo Linux; echo Student Hello Linux Student</pre>	Prints out the text Hello, Linux, and student in separate lines instead of one straight line.
29. false; echo Not; echo Conditional	<pre>ITEUser@Ubuntu-LOPEZ:~\$ false; echo Not; echo Conditional Not Conditional</pre>	Would prevent the display or the usage of a command that is included with false, Otherwise will display the next lines of command written after “;”.
30. echo Start Start && echo Going && echo Gone	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo Start && echo Going && echo Gone Start Going Gone</pre>	Will only print out the text if the command on the left of the operation “&&” successfully runs.
31. echo Success && false && echo Bye	<pre>ITEUser@Ubuntu-LOPEZ:~\$ echo Success && false && echo Bye Success</pre>	Will only print out the text if the command on the left of the operation “&&” successfully runs. in this case the command false ran and the text Success was displayed.
32. false echo Fail Or	<pre>ITEUser@Ubuntu-LOPEZ:~\$ false echo Fail Or Fail Or</pre>	Displays the text if one of the conditions for “ ” is true and the other is false
33. true echo Nothing to see here	<pre>ITEUser@Ubuntu-LOPEZ:~\$ true echo Nothing to see here ITEUser@Ubuntu-LOPEZ:~\$</pre>	Prevented the display of the command as the command at the left of the operand is true and prevents

[illegible]

6. Supplementary Activity:

Copy screen shot(s) of the following tasks:

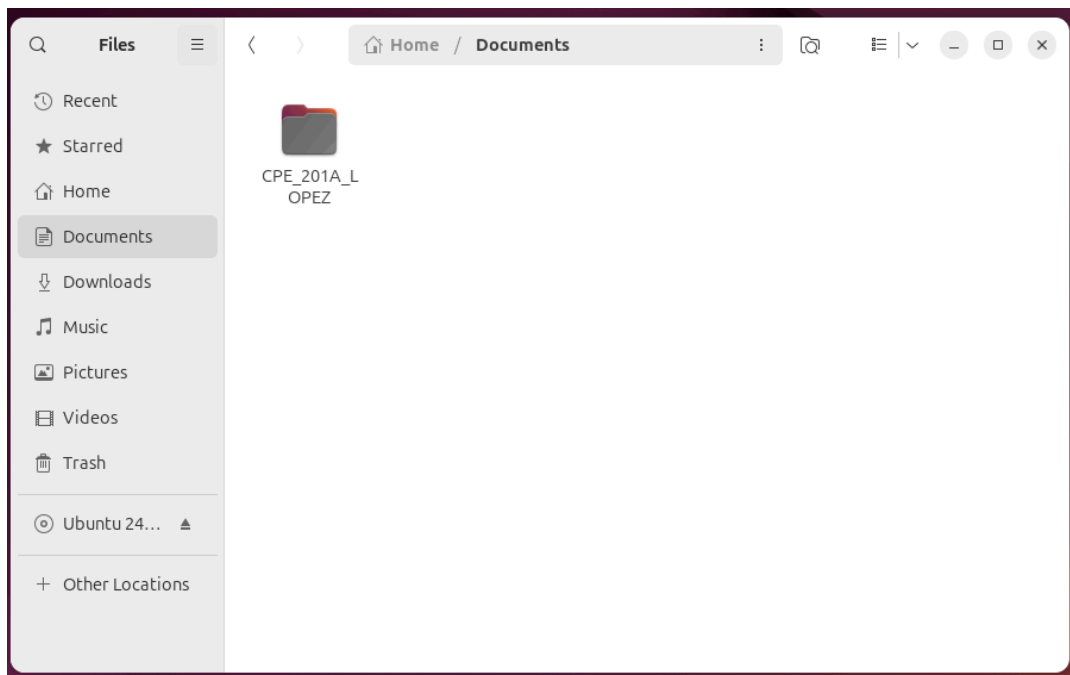
1. An alias can be used to map longer commands to shorter key sequences. Use an alias to represent a very long command.

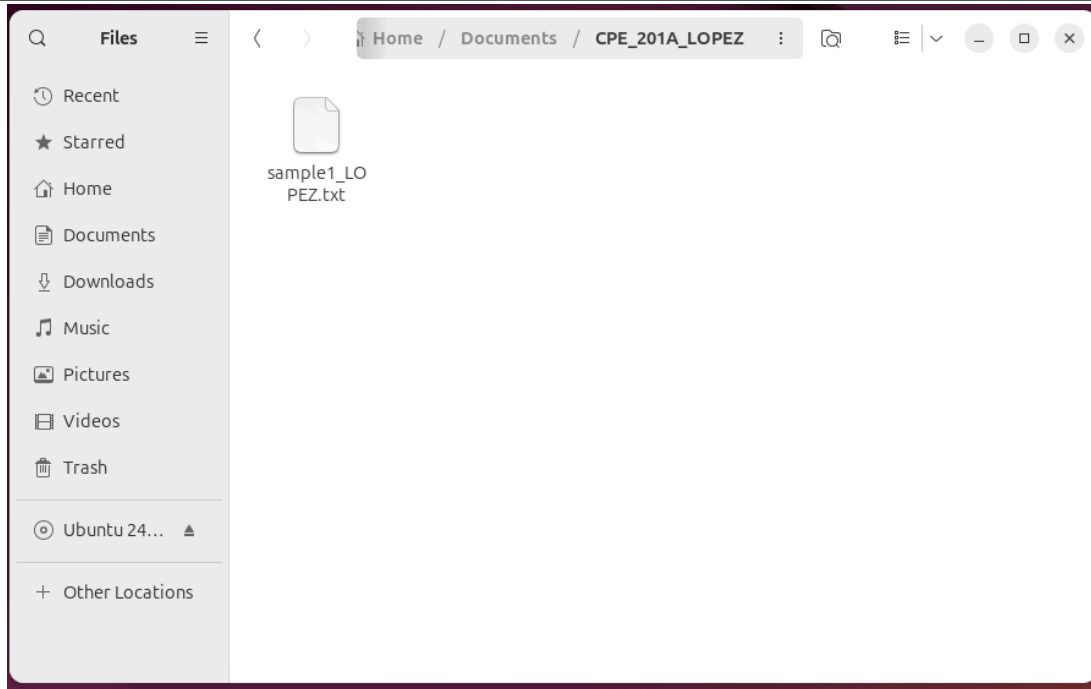
```
ITEUser@Ubuntu-LOPEZ:~/Documents/CPE_201A_LOPEZ$ alias folder_deletion='cd Documents; cd CPE_201A_LOPEZ; rm -f sample1_LOPEZ; cd; cd Documents; rmdir CPE_201A_LOPEZ'
ITEUser@Ubuntu-LOPEZ:~/Documents/CPE_201A_LOPEZ$ folder_deletion
bash: cd: Documents: No such file or directory
bash: cd: CPE_201A_LOPEZ: No such file or directory
ITEUser@Ubuntu-LOPEZ:~/Documents$
```

-
- What this code does is it removes the files and directories made in step 2. by naming the alias folder_deletion, running the alias will then execute the command string that is located in the alias folder deletion.

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.

```
ITEUser@Ubuntu-LOPEZ:~/Documents$ mkdir CPE_201A_LOPEZ
ITEUser@Ubuntu-LOPEZ:~/Documents$ cd CPE_201A_LOPEZ
ITEUser@Ubuntu-LOPEZ:~/Documents/CPE_201A_LOPEZ$ touch sample1_LOPEZ
ITEUser@Ubuntu-LOPEZ:~/Documents/CPE_201A_LOPEZ$ ls -m
sample1_LOPEZ
```





-
- 3. Execute a command to display the working shell.

```
ITEUser@Ubuntu-LOPEZ:~$ echo $SHELL
/bin/bash
ITEUser@Ubuntu-LOPEZ:~$ echo $0
bash
```

-
- In this activity I used these two commands to display the current working shell. The \$SHELL command displays the default shell configured for the current users account. The \$0 command displays the name of the current shell interpreter that the command line is currently using.
- 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?

```
ITEUser@Ubuntu-LOPEZ:~$ printenv |grep ^P
PWD=/home/ITEUser
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
```

7. Conclusion:

This activity tested my ability on using the command line and my understanding in the different commands that are used to navigate and access different files in the command line interface. It has also taught me how to use aliases to store long lines of code into one simple keyword that would execute a string of commands in one command, How to create a folder and files in the ubuntu command line interface using mkdir to create directories and touch to create files. It took me a while to make the

commands in the supplementary activity 1 as I was inputting the wrong command to access the file. Other than that I feel that I did pretty well in the activity.

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			