Activity No. 11	
Command Line Skills	
Course Code: CPE 201A	**Program: Computer Engineering**
Course Title: COMPUTER SYSTEM ADMINISTRATION AND TROUBLESHOOTING	**Date Performed: 10/23/2025**
Section: CPE11S1	**Date Submitted: 10/23/2025**
Name: Canoy Hail B.	**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

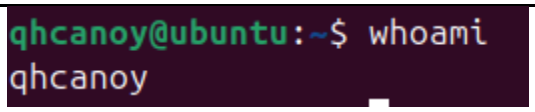
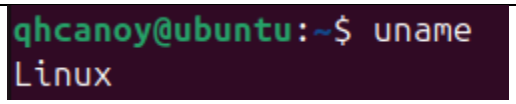
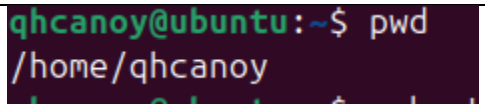
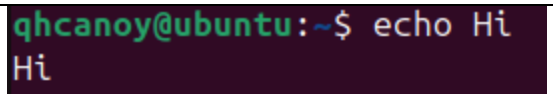
5. Procedure:

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	`qhcanoy@ubuntu:~$ ls -l`
	`total 36`
	`drwxr-xr-x 2 qhcanoy qhcanoy 4096 Oct 23 00:27`
	`drwxr-xr-x 2 qhcanoy qhcanoy 4096 Oct 23 00:27`
	`drwxr-xr-x 2 qhcanoy qhcanoy 4096 Oct 23 00:27`
	`drwxr-xr-x 2 qhcanoy qhcanoy 4096 Oct 23 00:27`
	`drwxr-xr-x 3 qhcanoy qhcanoy 4096 Oct 23 00:29`
	`drwxr-xr-x 2 qhcanoy qhcanoy 4096 Oct 23 00:27`
	`drwx------ 4 qhcanoy qhcanoy 4096 Oct 23 00:27`
	`drwxr-xr-x 2 qhcanoy qhcanoy 4096 Oct 23 00:27`
	`drwxr-xr-x 2 qhcanoy qhcanoy 4096 Oct 23 00:27`
		It shows the folders for desktop, documents, downloads, music, pictures, public, snap, templates, and videos.

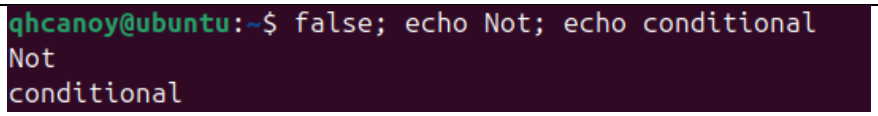
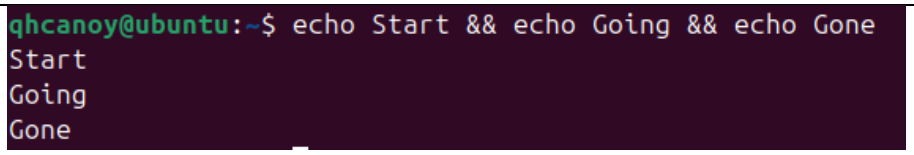
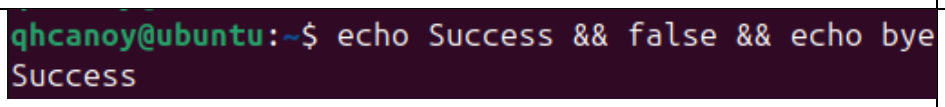
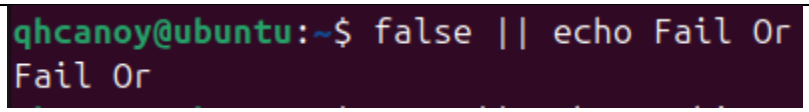
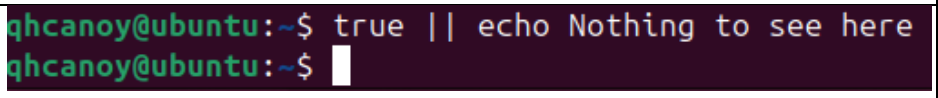
2. ls -l ./Documents	`qhcanoy@ubuntu:~$ ls -l ./Documents` `total 0`	Tells that there is no filesin that folder.

3. whoami		Displays my username.
4. Uname		Prints the kernels name which is Linux.
5. pwd		it will print the working directory.
6. echo Hi		To display information. Displaying text such us “hi”.

7. history	qhcanoy@ubuntu:~$ history 1 pwd 2 echo Hi 3 history 4 history 5 5 -l 6 -L 7 -l 8 ls -l 9 -L ./Documents 10 -l ./Documents 11 ls -l ./Documents 12 whoami 13 uname 14 -L ./Documents 15 ls -l 16 ls -l ./Documents 17 whoami 18 uname 19 pwd 20 echo hi 21 history	Displayes all the code history I did.
8. history 5	qhcanoy@ubuntu:~$ history 5 19 pwd 20 echo hi 21 history 22 -L ./Documents 23 history 5	Displays the last 5 code I entered.
9. !9	qhcanoy@ubuntu:~$!9 -L ./Documents -L: command not found	Displays the 9th code I entered.
10. echo Hello Student	qhcanoy@ubuntu:~$ echo Hello Student Hello Student	To display the text which is “Hello Student”
11. echo $HISTSIZE	qhcanoy@ubuntu:~$ echo $HISTSIZE 1000	Is the size of the

		command history.
12. echo $PATH	qhcanoy@ubuntu:~$ echo $PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin	Will display the terminal.
13. which date	qhcanoy@ubuntu:~$ which date /usr/bin/date	Tells that the date is stored in usr/bin.
14. type cd	qhcanoy@ubuntu:~$ type cd cd is a shell builtin	Tells that cd is a built in shell.
15. type ls	qhcanoy@ubuntu:~$ type ls ls is aliased to `ls --color=auto'	Shows files in color automatically.
16. alias	qhcanoy@ubuntu:~$ 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'	Shows all the command shortcut for alias.
17. type vi	qhcanoy@ubuntu:~$ type vi vi is /usr/bin/vi	it where the vi is located
18. cd /bin	qhcanoy@ubuntu:~$ cd /bin qhcanoy@ubuntu:/bin$ type vlc	Opens to the bin.
19. type vlc	qhcanoy@ubuntu:~$ cd/bin bash: cd/bin: No such file or directory	Bash not installed
20. cd	qhcanoy@ubuntu:/bin$ cd qhcanoy@ubuntu:~$	Returns to cd.
21. echo Today is `date`	qhcanoy@ubuntu:~$ echo Today is October 23 2025 Today is October 23 2025	For it to print “today is October 23 2025.
22. echo Today is $(date)	qhcanoy@ubuntu:~$ echo Today is $(date) Today is Thu Oct 23 01:07:04 AM UTC 2025	This is just another way to print the

		date today.
23. echo This is the command \`date\`	qhcanoy@ubuntu:~$ echo This is the command October 23 2025 This is the command October 23 2025	Print the date exactly as it is.
24. echo This is the command \`date\`	qhcanoy@ubuntu:~$ echo This is the command October 23 2025 This is the command October 23 2025	It will stop the shell that makes the \date\ as command and will print the text.
25. echo This is the command "`date`"	qhcanoy@ubuntu:~$ echo This is the command "'date'" This is the command 'date' qhcanoy@ubuntu:~$ echo This is the command "'October 23 2025'" This is the command 'October 23 2025'	It will print the date again.
26. echo D*	qhcanoy@ubuntu:~$ echo D* Desktop Documents Downloads	Shows the files starting with d such as desktop documents and downloads.
27. echo "D*"	qhcanoy@ubuntu:~$ echo "D*" D*	it will print d.
28. echo Hello; echo Linux; echo Student	qhcanoy@ubuntu:~$ echo Hello; echo Linux; echo Student Hello Linux Student	It seperates the command with semicolon and will print the text exactly as it is, its

		like making a newline.
29. false; echo Not; echo Conditional		False just does nothing and makes the still makes the next command run.
30. echo Start && echo Going && echo Gone		It will print the text and && will run the next text if the text is already printed.
31. echo Success && false && echo Bye		It will print success because it used false so bye is cancelled .
32. false || echo Fail Or		Flase here will be stopped by || so the Fail Or will still be printed.
33. true || echo Nothing to see here		It will have no output because true is stopped by || so

		meaning the text will not be printed.
34. printenv	qhcanoy@ubuntu:~$ printenv SHELL=/bin/bash SESSION_MANAGER=local/ubuntu:@/tmp/.ICE-unix/2366,unix/ubuntu: QT_ACCESSIBILITY=1	Shows the ubuntu varibles.
35. printenv TERM	qhcanoy@ubuntu:~$ printenv TERM xterm-256color	Shows the value of the variable like “term”
36. echo $TERM	qhcanoy@ubuntu:~$ echo $TERM xterm-256color	Shows the value of $term
37. env	qhcanoy@ubuntu:~$ env SHELL=/bin/bash SESSION_MANAGER=local/ubuntu:@/tmp/.ICE-unix/2366,unix/ubuntu:/tmp/.ICE-unix/2366	It shows all ubuntu variables that are are active.

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.

```
qhcanoy@ubuntu:~$ cd ~/Documents && mkdir CPE_201A_Canoy
mkdir: cannot create directory ‘CPE_201A_Canoy’: File exists
```

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.bash

```
qhcanoy@ubuntu:~/Documents/CPE_201A_Canoy$ mv new_file.txt sample1_Canoy.txt
qhcanoy@ubuntu:~/Documents/CPE_201A_Canoy$ ls
sample1_Canoy.txt
```

print

3. Execute a command to display the working shell.

```
qhcanoy@ubuntu:~/Documents/CPE_201A_Canoy$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/ubuntu:@/tmp/.ICE-unix/2366,unix/ubuntu:/tmp/.ICE-unix/23
6
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
MEMORY_PRESSURE_WRITE=c29tZSAyMDAwMDAgMjAwMDAwMAA=
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
GTK_MODULES=gail:atk-bridge
```

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?

```
qhcanoy@ubuntu:~/Documents$ printenv | grep ^P
PWD=/home/qhcanoy/Documents
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/u
r/local/games:/snap/bin:/snap/bin
```

7. Conclusion:

- In today's activity I have learned more about ubuntu. Codes in the terminal for displaying things such us echo and many more codes that have different functions. I keep getting wrong outputs because of the typo. It was kinda hard to read because of the text but I still adapted and finished the activity. Overall this activity really helped me in understanding ubuntu more. I can use this in the future if I have a job. I actually have a hard time reading the texts. I have a fever but I still did it, and I think I did pretty well although I still need improvements.
-

8. Assessment (Rubric for Laboratory Performance):