

Activity No. 11	
Command Line Skills	
Course Code:CPE007	Program:
Course Title: Programming Logic and Design	Date Performed:10-23-25
Section:CPE11S1	Date Submitted:10-23-25
Name:JOEL A ALMIROL	Instructor:
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> <p>root@mycomputer:~#</p> <p>On CentOS or Red Hat Linux, the prompt for a regular user will instead look like this:</p> <p>[dave@mycomputer ~]\$</p> <p>And the superuser's prompt will look like this:</p>	

```
[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

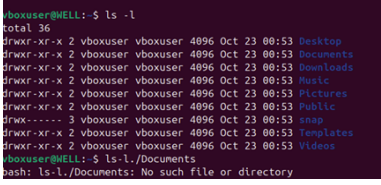
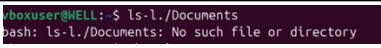
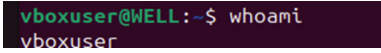
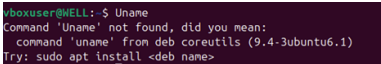
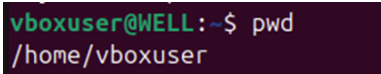
4. Resources:

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		The ls -l command provides a detailed list of file and directories. The output includes information such as file permissions the number of link the owner and group file size the last modification date and time and the name of the file or directory
2. ls -l ./Documents		The user vboxuser is trying to run the command ls-l./Documents this command is likely intended to list the contents of a directory named Documents in a long format
3. whoami		This command prints the username of the user who is currently logged in and running the shell. In your case, it outputted vboxuser
4. Uname		This command produced an error (Command 'Uname' not found) because Linux commands are case-sensitive. The correct command to print system information is uname (all lowercase).
5. pwd		This command stands for "print working directory" and displays the absolute path of the directory you

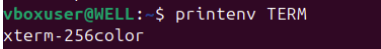
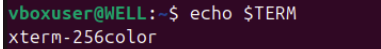
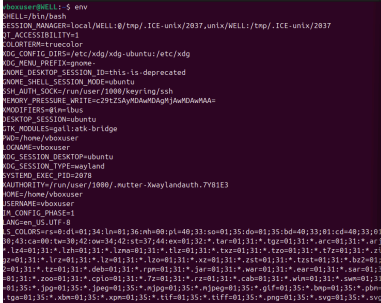
		are currently in. The output /home/vboxuser indicates the shell is in the home directory of the vboxuser.
6. echo Hi	<pre>vboxuser@WELL:~\$ echo Hi Hi</pre>	The echo command simply prints any text or arguments that follow it to the standard output (your terminal). It displayed Hi
7. history	<pre>vboxuser@WELL:~\$ history 1 ls -l 2 ls -l./Documents 3 ls -l 4 ls-l./Documents 5 whoami 6 Uname 7 pwd 8 echo Hi 9 history</pre>	This command displays the numbered list of previous commands executed in the current shell session.
8. history 5	<pre>vboxuser@WELL:~\$ history 5 6 Uname 7 pwd 8 echo Hi 9 history 10 history 5</pre>	This command lists the last 5 commands that were executed from the shell's command history.
9. !9	<pre>vboxuser@WELL:~\$!9 history 1 ls -l 2 ls -l./Documents 3 ls -l 4 ls-l./Documents 5 whoami 6 Uname 7 pwd 8 echo Hi 9 history 10 history 5 11 history</pre>	This is a Bash feature that re-executes the command from the history list that corresponds to the number 9. Since command 9 was history, executing !9 reran the history command.
10. echo Hello Student	<pre>vboxuser@WELL:~\$ echo Hello Student Hello Student</pre>	This command uses echo to simply print the text Hello Student to the terminal.
11. echo \$HISTSIZE	<pre>vboxuser@WELL:~\$ echo \$HISTSIZE 1000</pre>	The dollar sign (\$) is used to display the value of a shell variable. HISTSIZE

		is a shell variable that determines the maximum number of commands stored in the shell's history list. The output 1000 is the current limit
12. echo \$PATH	<pre>vboxuser@WELL:~\$ echo \$PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/games:/usr/local/games</pre>	This command displays the value of the PATH environment variable. This variable is a list of directories, separated by colons, that the shell searches for executable commands when you type a command name.
13. which date	<pre>vboxuser@WELL:~\$ which date /usr/bin/date</pre>	The which command searches the directories listed in the \$PATH environment variable and displays the full path to the executable file for the command you specified. The output /usr/bin/date shows where the date command's executable file is located.
14. type cd	<pre>vboxuser@WELL:~\$ type cd cd is a shell builtin</pre>	The type command tells you how the shell interprets the command name you provide. The output cd is a shell builtin means that cd is not a separate executable file but a function built directly into the shell program itself (in this case, Bash).
15. type ls	<pre>vboxuser@WELL:~\$ type ls ls is aliased to `ls --color=auto`</pre>	This shows that ls is an alias to a longer command, specifically ls --color=auto. An alias is a shortcut that maps a short name to a potentially longer command string.

16. alias	<pre>vboxuser@WELL:~\$ alias alias alert='notify-send --urgency=low -i "\${((\$? = 0))} && echo terminal echo error" "/usr/bin/notify-send --urgency=low -i "\${((\$? = 0))} && echo terminal echo error" alias sprun='sprun --color=auto' alias fgrun='fgrun --color=auto' alias grun='grun --color=auto' alias ls='ls -l' alias ll='ls -l' alias ls='ls --color=auto'</pre>	This command lists all defined aliases in the current shell session. This shows which short commands (like ls or ll) are mapped to longer commands or options.
17. type vi	<pre>vboxuser@WELL:~\$ type vi vi is /usr/bin/vi</pre>	The output vi is /usr/bin/vi indicates that vi is a separate executable program located in the /usr/bin/ directory.
18. cd /bin	<pre>vboxuser@WELL:~\$ cd /bin</pre>	The cd (change directory) command changes the shell's current working directory to the specified path. This command changed the directory to /bin, as indicated by the prompt changing from ~ to /bin
19. type vlc	<pre>vboxuser@WELL:/bin\$ type vlc bash: type: vlc: not found</pre>	This command attempts to determine how the shell interprets vlc. The output bash: type: vlc: not found means the shell could not find an executable or builtin/alias named vlc. This typically indicates the VLC media player application is not installed or not in a directory listed in your \$PATH.
20. cd	<pre>vboxuser@WELL:/bin\$ cd</pre>	When the cd command is executed without any arguments, it returns the shell's current working directory to the current user's home directory. The prompt, which was /bin, changes back to ~ (which

		represents the home directory)
21. echo Today is `date`	<pre>vboxuser@WELL:~\$ echo Today is `date` Today isdate</pre>	When single quotes (') are used, they tell the shell to treat the enclosed characters literally and prevent the shell from performing command substitution. Therefore, the shell prints date as a literal word instead of executing the date command
22. echo Today is \$(date)	<pre>vboxuser@WELL:~\$ echo Today is \$(date) Today is Thu Oct 23 01:16:34 AM UTC 2025</pre>	Command substitution occurs when a command enclosed in \$(...) is executed, and its output replaces the \$(...) expression in the main command. The date command is executed, and its output (the current date and time) is printed along with the text "Today is".
23. echo This is the command `date`	<pre>vboxuser@WELL:~\$ echo This is the command `date` This is the command`date`</pre>	Backticks (`) are an older form of command substitution. In this case, because the backticks are enclosed in single quotes, they are treated as literal characters and prevent the shell from executing the date command.
24. echo This is the command `date`\`	<pre>vboxuser@WELL:~\$ echo This is the command `date`\` This is the command`date`\`</pre>	The backslash (\) is an escape character that removes the special meaning of the character immediately following it. The backslashes before the backticks prevent the command substitution, so the literal string date is printed.

<p>25. echo This is the command "date"</p>	<pre>vboxuser@WELL: \$ echo This is the command`date` This is the commandThu Oct 23 01:24:20 AM UTC 2025</pre>	<p>Double quotes (") allow command substitution. The backticks are executed, running the date command, and its output is printed. However, the output of date is printed directly adjacent to the preceding text because the space is between the command's output and the closing quote.</p>
<p>26. echo D*</p>	<pre>vboxuser@WELL:~\$ echo D* Desktop Documents Downloads</pre>	<p>The asterisk (*) acts as a wildcard (or globbing) character, matching any sequence of zero or more characters. The shell replaces D* with all files and directories in the current directory whose names start with the capital letter 'D', which in this case were Desktop, Documents, and Downloads.</p>
<p>27. echo "D"</p>	<pre>vboxuser@WELL:~\$ echo "D*" D*</pre>	<p>Double quotes prevent wildcard expansion. The shell treats D* as a literal string and prints it as-is.</p>
<p>28. echo Hello; echo Linux; echo Student</p>	<pre>vboxuser@WELL:~\$ echo Hello; echo Linux; echo Student Hello Linux Student</pre>	<p>The semicolon (;) acts as a command separator. It allows multiple commands to be executed sequentially on a single line, regardless of the success or failure of the preceding command. All three echo commands ran and printed their respective strings on new lines.</p>

35. printenv TERM		This command lists only the value of the specified environment variable, TERM. The TERM variable defines the type of terminal emulation the user is working with, which affects how programs display text and colors. The output xterm-256color is the current setting
36. echo \$TERM		The echo command is used with the dollar sign (\$\$) to display the value of the shell variable TERM. It produces the same output as printenv TERM, confirming the terminal type is xterm-256color
37. env		The env command, similar to printenv without arguments, is used to list all environment variables in the current shell context

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

7. Conclusion:

The completion of this activity has effectively showcased the performance of basic Linux commands through the Bash shell and the Command Line Interface (CLI), thus achieving the goals set for the exercise. The tasks given offered the participants some practical hands-on experience in the use of BASH features like command history management (history, !n), defining command behavior (type, alias), and gaining knowledge of core directory and system utilities (pwd, whoami). Among other things, the activities showed the ability to operate in the shell environment, particularly by showing Shell variable values such as \$HISTSIZE and \$PATH, and listing all active variables using printenv and env.

Moreover, the tasks not only highlighted the command-line syntax but also showed the significance of quoting in Bash—where single quotes (') treat the enclosed expression literally while double quotes (") allow substitution—and illustrated command substitution by means of both modern \$(...) and traditional backticks. The successful processing of commands governed by the semicolon (;) and the conditional operators && (AND) and || (OR) showed the technique of controlling the execution flow of commands. These are none other than the basic skills required in the COMPUTER SYSTEM ADMINISTRATION AND TROUBLESHOOTING course (\$\text{CPE 201A}\$) thus confirming the user's competence to be both efficient and accurate in any Linux distribution.

8. Assessment (Rubric for Laboratory Performance):