Introduction to shell scripts

Command line

Objectives:

- To understand what the command line is and why it is used
- To effectively navigate across different files and directories
- To perform operations on files in specific directories

Theory:

- the bash shell explained. Below is a figure that breaks down what was explained during the training. An operating system is a graphical interface between a computer user and the computer system. The following diagram illustrates the four layers of your computer:
    1. **Hardware** - Hardware consists of all physical devices attached to the System. example: Hard disk drive, Motherboard, CPU etc.
    2. **Kernel** - Kernel is the core component for any operating system which directly interacts with the hardware.
    3. **Shell** - Shell takes input from Users and sends commands to the Kernel. The shell also takes output from Kernel and sends it back to output shell.
    4. **Applications -** There are a variety of applications, eg: any apps ranging from your web browser, to media player, text editor etc.

Shell Scripting
Shell Scripting is an open-source computer program designed to be run by the Unix/Linux shell. Shell Scripting is a program to write a series of commands for the shell to execute (in the terminal). It can combine lengthy and repetitive sequences of commands into a single and simple script that can be stored and executed anytime, which reduces programming efforts.

What is Shell?
Shell is a UNIX term for an interface between a user and an operating system service. Shell provides users with an interface and accepts human-readable commands into the system and executes those commands which can run automatically and give the program's output in a shell script.
An Operating system is made of many components, but its **two prime** components:
Kernel: A kernel is the lowest level of software that can interface with computer hardware. All Linux applications and servers also interface with the Linux kernel. All Linux distributions are based on the Linux kernel and use its services to implement various software functions.

Shell: The shell is the Linux command line interpreter. It provides an interface between the user and the kernel and executes programs called commands. For example, if a user enters ls then the shell executes the ls command.
Type of shell: Bourne Shell (sh) ($), GNU Bourne-Again Shell (bash) ($), C Shell (csh) (%), Korn Shell (ksh) and The Z Shell (zsh)(%)

Terminal: The terminal is an interface in which you can type and execute text based commands. **Advantage of using terminal.** It can be much faster to complete task using than with graphical applications. Another benefit is allowing access to many more commands and scripts.

Command line interface (CLI)

In this course we will focus in the most basic and useful commands:

Command 1: ls

This command is used to list down all the directories and files inside the present working directory (or you can give the path of a specific directory).

$ ls

$ ls -al To print the detailed information of the files/directories

$ ls -l will give you a long listing (as explained above)

$ ls -a will show you ALL the files in the directory, including hidden files

$ ls -R will the subdirectories recursively, which means it will show all the directories and files within the specified directory.

$ ls -s will also show you the size of the files (in blocks, not bytes)

$ ls -h will show the size in "human readable format" (ie: 4K, 16M, 1G etc).

$ls -l will show the file permission in the start of the line

Command 2: pwd

This command refers to the present working directory in which you are operating.

$ pwd the output should looks like **/home/username**

Command 3: dir

The dir command is used to print all the available directories in the present working directory:

$ dir # expect to see all directories that in the work directories

Command 4: cd

One of the most used commands where you can change the directories in the terminal using the **"cd"** command. And cd command have more options. For instance, the following command will change directory form operating directory to desktop.

There are to two ways to move between directories whether you provide full path directory for example:

$ cd / home/user/Desktop

Or the **relative** path directory:

$ cd Desktop/

Here is other options for cd command

$ cd / ##### To change directory to root.

$ cd .. ##### To shift one level above the current directory, we input .. as the argument.

$ cd - ##### To go to the previous directory, we use - as our argument.

$ cd directory && ls ##### To go to the directory and list the files and directory in the directory you typed in the command.

For custom navigation across any directory, we can send its path as the argument.

$ cd / folder/subfolder

Command 5: ps

```
$ ps  #  By passing this command it will display which type of shell script you are using.
```

Command 6: touch

This command can be used to create a new file as well one can use it to change the time stamp of any file; the command given below will create a new text file in the same directory you operating

```
$ touch file.txt
```

by using stat it will show you if the file been accessed, modified or changed

```
$ stat file.txt
```

Command 7 : nano

Nano is text file editor used to create text file (file.txt) or bash script (script.sh)

```
$ nano # will open the text file.
```

Write your info full name and phone number save as text file call

Command 7: wc

```
$ wc test.txt  # if you write your name and surname you should get this output
```

$ wc command also including other options (-c characters and -l lines)

examples

```
$ wc -C test.txt
$ wc -l test.txt
```

Command 8: cat

This command is used to show the content of any file: For instance, the following command will display the content inside **"test.txt":**

```
$ cat test.txt
```

Or you can use this command to save the content of multiples files to one file:

```
$ cat test.txt file2.txt > output.txt
```

Command 9: mkdir

The above-mentioned command will make a new directory in your working directory for example, the following command will make the directory "**project** " as new directory in the **Desktop**

```
$ cd Desktop
$ mkdir project
```

Command 10: rm

This remove command is used to remove the specific file from a directory; For instance, below mentioned command would remove the "**test.txt**" file from the pwd:

```
$ rm test.txt
```

Or you can remove the empty directory, as the command given below will remove the "test" directory:

```
$ rmdir test
```

In case the directory has some files in it will not allow us to deleted unless we **(rm -r)**

```
$ rm -r test3   ### It will delete the directory
```

Command 11: cp

The cp command will help you to copy any file or folder to any directory; To copy a file to directory1:

```
$ cp test.txt project
```

Command 12: mv

You can use this command to move files around the computer, and you can also rename files or directories inside a specific directory: the command given below will move the **"test.txt"** to "**project**":

```
$ mv test.txt project
```

Command 13: head

This command helps you to get the first ten lines of a text file; for instance, the following command will help to get the first ten lines of the "**test.txt**" file:

```
$ head file.txt
```

Command 14: tail

The tail command is used to get the last ten lines of the text file; the command below will print the ten lines from the bottom of **"test.txt**":

```
$ tail test.txt
```

Command 15: uname

You can use the command to get the release number, version of Linux, and much more. The "**-a**" flag is used to get detailed information.

```
$ uname -a
```

Command 16: wget

You can use the wget command to download the content from the internet; for instance, the following command will download Vlc.

```
$ wget https://www.videolan.org/vlc/download-debian.html
```

Command 17: apt-get or -apt

This is one of the most important and most used commands of Ubuntu that works with Ubuntu Advanced Packaging Tool (APT); you can use this "**-apt-get**" or "**-apt**" to install or remove packages, or you can perform other maintenance tasks. The "**apt**" requires **Super User DO (sudo)** privileges to successfully execute the command.
sudo command in Linux is generally used as a prefix for some commands that only superusers are allowed to run.
The syntax stated below will help you to install the required package:

```
$ sudo apt install [packagename] ##
```

```
$  sudo apt install ncbi-blast+
```

Or you can remove the package by executing the command given below:

```
$ sudo apt install vlc # example
```

```
$ sudo apt remove [packagename]  # example try to delete vlc
```

Command 18: history
The history command shows the list of commands (with numeric numbers) executed:

```
$ history
```

Command 19: grep
With the help of grep, you can search for a pattern in which a specific word lies; for instance, the command given below will print all the lines that contain "**and**" from "**test.txt**":

```
$ cat test.txt | grep 20
```

Command 20: man
The man command will help you to get the complete user manual of any specific command; for instance, the following command will list down the detailed usage of the "**cat**" command:

```
$ man cat
```

Command 21: zip or unzip
To convert your files to zip archive; you can get help by using the "**gzip**" command; moreover, a zipped file can be unzipped using the "**gunzip**" command:

```
$ gzip test.txt
$ gunzip test.txt
```

Command 22: hostname

This command will print your hostname on the terminal:

```
$ hostname
```

Command 23: passwd

With the help of the passwd command, you can change the password of your Ubuntu user:

```
$ passwd [user name]          ### type new password
```

Command 24: Create bash script or **.sh** script file

Shell script creation: Launch Nano, Vim, or any other txt editor. Name a new file with an essential ".sh" extension. Like "script.sh". Start the file with the shebang line to designate the interpreter. Bash shebang lines are: #!/bin/bash

Example:

```
#!/bin/bash

# This is a simple Bash script to greet the user and display the current date and time.

# Greet the user

echo "Hello! Welcome to the example script."

# Display the current date and time

current_date=$(date +"%Y-%m-%d")

current_time=$(date +"%H:%M:%S")

echo "Today is $current_date, and the current time is $current_time."
```

Command 25: Chmod

The `chmod` command allows you to modify file permissions. You can specify permissions in both numeric and symbolic notation. By using **ls -l** it will show all the file permission.

Numeric Notation:

- Each permission is represented by a digit: 4 for read (`**r**`), 2 for write (`**w**`), and 1 for execute (`**x**`).

- To set permissions, add the digits of the desired permissions for owner, group, and others. For example, `chmod 644 file.txt` sets read and write permissions for the owner and read-only permissions for group and others.

Symbolic Notation:

- The symbolic notation uses letters to represent the permissions: `**u**` for user/owner, `**g**` for group, and `**o**` for others.
- The `+`, **`-`,** and `=` signs are used to add, remove, or set permissions, respectively.
- `**r**`, `**w**`, and `**x**` are used for read, write, and execute permissions, respectively.
- For example, `**chmod u+x file.txt**` adds execute permission for the owner.

```
$chmod u+x file.txt
```

You can read more aboout file permisson in the following link **https://www.pluralsight.com/blog/it-ops/linux-file-permissions .**

Practical

Open the terminal and generate the following directories and files.

1- Create the directory in the desktop that named “project1” while you are in home directory.

2- Change the directory from home to the project1 and list the files in project1 at same time.

3- By using nano text editor create txt file named (my_info.txt) contain your name and surname with phone number.

A) Check how many zero in the file and A latter in your name.

B) How many latter, characters and line in the my_info.txt

4 - Change directory from project1 to home then create another directory called project2 in the desktop

A) Move the my_info.txt that you create from project1 to project2.

B) Rename project1 as data_analysis

6 - Remove or delete project1 directory from desktop.