Assignment-2: Bash Shell Basics

Name: Devendra Dev Roll No.: 20BCE2191 Course: CEH VIT Batch

Email: devendra.dev2020@vitstudent.ac.in

Task 1: File and Directory Manipulation

1. Create a directory called "my_directory".

mkdir my_directory

2. Navigate into the "my_directory".

cd my_directory/

3. Create an empty file called "my_file.txt".

touch my file.txt

4. List all the files and directories in the current directory.

ls

5. Rename "my file.txt" to "new file.txt".

mv my file.txt new file.txt

6. Display the content of "new_file.txt" using a pager tool of your choice.

more new file.txt

7. Append the text "Hello, World!" to "new_file.txt".

Echo "Hello, World!" >> new file.txt

This append the result of echo into the file new file.txt

8. Create a new directory called "backup" within "my_directory". mkdir backup

```
dev@DevsPredator:~/my_directory$ mkdir backup
dev@DevsPredator:~/my_directory$ mv new_file.txt backup/
dev@DevsPredator:~/my_directory$ ls
backup
dev@DevsPredator:~/my_directory$ cd backup/
dev@DevsPredator:~/my_directory/backup$ ls
new_file.txt
dev@DevsPredator:~/my_directory/backup$ cd ..
dev@DevsPredator:~/my_directory$ ls
backup
dev@DevsPredator:~/my_directory$ rm -rf backup
dev@DevsPredator:~/my_directory$ ls
```

9. Move "new_file.txt" to the "backup" directory. mv new file.txt backup

- **10. Verify that "new_file.txt" is now located in the "backup" directory.** It is verified from the above picture through "Is" command.
- 11. Delete the "backup" directory and all its contents.

rm -rf backup

This removes all the contents and subdirectories and files of backup.

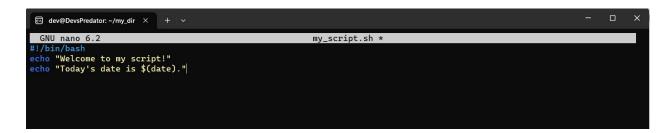
Task 2: Permissions and Scripting

Create a new file called "my_script.sh".
 touch my_script.sh

```
dev@DevsPredator:~/my_directory$ touch my_script.sh
dev@DevsPredator:~/my_directory$ nano my_script.sh
dev@DevsPredator:~/my_directory$ chmod +x my_script.sh
dev@DevsPredator:~/my_directory$ ls
my_script.sh
dev@DevsPredator:~/my_directory$ bash my_script.sh
Welcome to my script!
Today's date is Tue May 30 13:28:57 IST 2023.
```

• Edit "my_script.sh" using a text editor of your choice and add the following lines:

#!/bin/bash
echo "Welcome to my script!"
echo "Today's date is \$(date)."
<<Save and exit the file.>>



• Make "my_script.sh" executable.

chmod +x my_script.sh

This command changes the permission of the file my script.sh to executable.

Run "my_script.sh" and verify that the output matches the expected result.

"bash" or "sh" is used to run the file my_script.sh The output matches with the expected result.

Task 3: Command Execution and Pipelines

• List all the processes running on your system using the "ps" command. \$ ps -ef

```
UID
                PPID
                      C STIME TTY
                                            TIME CMD
                        13:18
                                       00:00:00 /init
root
             4
                      0 13:18 ?
                                        00:00:00 plan9 --control-socket 5 --log-level 4 --server-fd 6 --pipe-fd 8 --log-t
root
                        13:18
                                        00:00:00 /init
                      0 13:18 ?
                                        00:00:00 /init
                      0 13:18 pts/0
                                       00:00:00 -bash
            55
                        13:21 pts/0
                                        00:00:00 cat
```

This shows all the running processes in the system.

• Use the "grep" command to filter the processes list and display only the processes with "bash" in their name.

```
$ ps -ef | grep "bash" // The output of the command ps -ef is the input for grep
```

• Use the "wc" command to count the number of lines in the filtered output.

wc represents the number of lines, number of words and the number of characters/bytes.

Submission:

Provide a document or text file containing the commands used to complete the tasks above, along with any relevant output or screenshots. Include your explanations or observations where necessary.