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

```
dev@DevsPredator:~$ ls
 CSE4001_PDC_Project  'ZoomInstallerFull.exe?archType=x64'   demo.txt   directory   instructions.txt   new
dev@DevsPredator:~$ mkdir my_directory
dev@DevsPredator:~$ cd my_directory/
dev@DevsPredator:~/my_directory$ touch my_file.txt
dev@DevsPredator:~/my_directory$ mv my_file.txt new_file.txt
dev@DevsPredator:~/my_directory$ ls
new_file.txt
```

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

```
dev@DevsPredator:~/my_directory$ more new_file.txt
dev@DevsPredator:~/my_directory$ echo "Hello, World" >> new_file.txt
dev@DevsPredator:~/my_directory$ cat
^Z
[1]+  Stopped                 cat
dev@DevsPredator:~/my_directory$ cat new_file.txt
Hello, World
```

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 “ls” 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.>>

```
dev@DevsPredator: ~/my_dir
  GNU nano 6.2                                   my_script.sh *
#!/bin/bash
echo "Welcome to my script!"
echo "Today's date is $(date)."
```

- **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

```
dev@DevsPredator:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 13:18 ?        00:00:00 /init
root         4     1  0 13:18 ?        00:00:00 plan9 --control-socket 5 --log-level 4 --server-fd 6 --pipe-fd 8 --log-t
root         7     1  0 13:18 ?        00:00:00 /init
root         8     7  0 13:18 ?        00:00:00 /init
dev          9     8  0 13:18 pts/0    00:00:00 -bash
dev         55     9  0 13:21 pts/0    00:00:00 cat
dev         93     9  0 13:38 pts/0    00:00:00 ps -ef
```

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

```
dev@DevsPredator:~$ ps -ef | grep “bash”
dev         95     9  0 13:38 pts/0    00:00:00 grep --color=auto “bash”
dev@DevsPredator:~$ ps -ef | grep “bash” -n
9:dev         97     9  0 13:38 pts/0    00:00:00 grep --color=auto “bash” -n
```

- **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.

```
dev@DevsPredator:~$ ps -ef | grep “bash” | wc
      1      10      77
dev@DevsPredator:~$ ps -ef | grep “bash” | wc -l
1
```

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.