Assignment: Bash Shell Basics

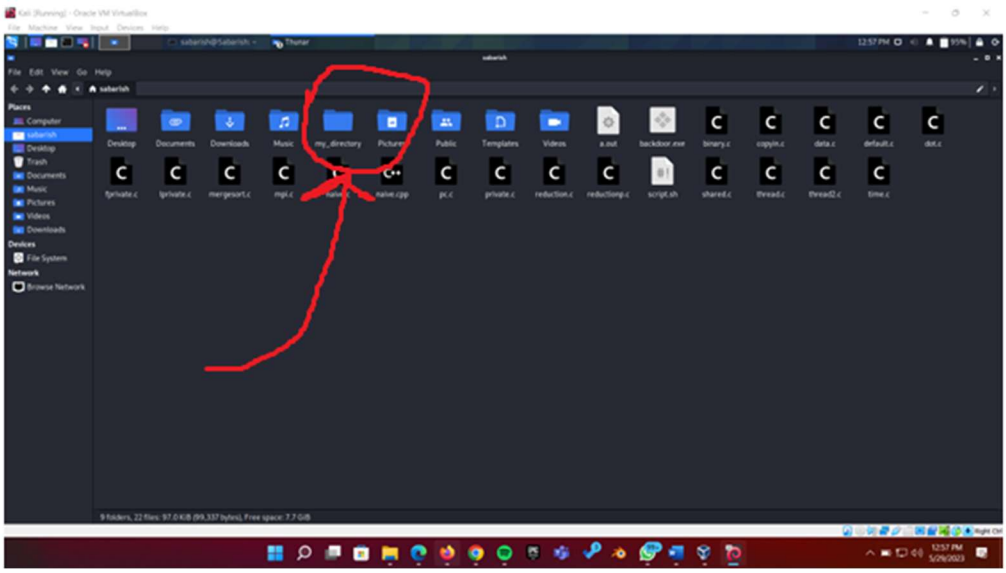
Name: Yerramsetty Sai Naga Sabarish
Reg no:20BCE2370
CAMPUS: Vit Vellore

Task 1: File and Directory Manipulation

1. Create a directory called "my_directory".

```
┌──(sabarish㉿Sabarish)-[~]
└─$ bash script.sh
 Enter the name of the directory
my_directory

┌──(sabarish㉿Sabarish)-[~]
└─$
```



2. Navigate into the "my_directory".

```
┌──(sabarish㉿Sabarish)-[~]
└─$ cd my_directory

┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$
```

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

```
┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ touch my_file.txt
```

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

```
┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ ls
my_file.txt
```

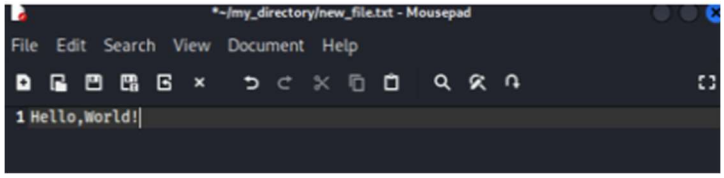
5. Rename "my_file.txt" to "new_file.txt".

```
┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ mv my_file.txt new_file.txt
```

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

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

```
┌──(sabarish㉿Sabarish)-[~]
└─$ nano new_file.txt
```



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

```
┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ mkdir backup
```

9. Move "new_file.txt" to the "backup" directory.

```
┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ mv new_file.txt backup/
```

10. Verify that "new_file.txt" is now located in the "backup" directory.

```
┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ ls backup/
new_file.txt
```

11. Delete the "backup" directory and all its contents.

```
┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ rm -r backup

┌──(sabarish㉿Sabarish)-[~/my_directory]
└─$ ls backup/
ls: cannot access 'backup/': No such file or directory
```

Task 2: Permissions and Scripting

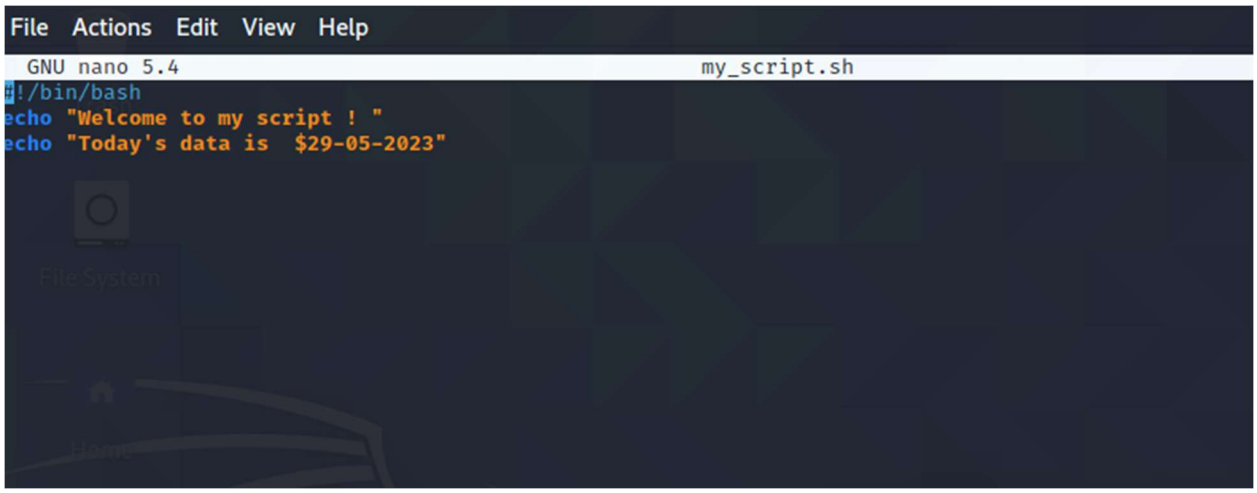
- Create a new file called "my_script.sh".

```
┌──(sabarish㉿Sabarish)-[~]
└─$ touch my_script.sh
```

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

bash

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



- Make "my_script.sh" executable.

```
┌──(sabarish㉿Sabarish)-[~]
└─$ chmod +x my_script.sh
```

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

```
┌──(sabarish㉿Sabarish)-[~]
└─$ ./my_script.sh
Welcome to my script !
Today's data is  9-05-2023
```

Task 3: Command Execution and Pipelines

- List all the processes running on your system using the "ps" command.

```
┌──(sabarish㉿Sabarish)-[~]
└─$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.5  0.5 164172 10484 ?        Ss   12:53   0:08 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    12:53   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   12:53   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   12:53   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   12:53   0:00 [kworker/0:0H-events_highpri]
root           7  0.0  0.0      0     0 ?        I    12:53   0:00 [kworker/u8:0-events_unbound]
root           8  0.0  0.0      0     0 ?        I<   12:53   0:00 [mm_percpu_wq]
root           9  0.0  0.0      0     0 ?        S    12:53   0:00 [rcu_tasks_rude_]
root          10  0.0  0.0      0     0 ?        S    12:53   0:00 [rcu_tasks_trace]
root          11  0.0  0.0      0     0 ?        S    12:53   0:00 [ksoftirqd/0]
root          12  0.2  0.0      0     0 ?        I    12:53   0:04 [rcu_sched]
root          13  0.0  0.0      0     0 ?        S    12:53   0:00 [migration/0]
root          14  0.0  0.0      0     0 ?        I    12:53   0:00 [kworker/0:1-events]
root          15  0.0  0.0      0     0 ?        S    12:53   0:00 [cpuhp/0]
root          16  0.0  0.0      0     0 ?        S    12:53   0:00 [cpuhp/1]
root          17  0.0  0.0      0     0 ?        S    12:53   0:00 [migration/1]
root          18  0.0  0.0      0     0 ?        S    12:53   0:00 [ksoftirqd/1]
root          20  0.0  0.0      0     0 ?        I<   12:53   0:00 [kworker/1:0H-events_highpri]
root          21  0.0  0.0      0     0 ?        S    12:53   0:00 [cpuhp/2]
root          22  0.0  0.0      0     0 ?        S    12:53   0:00 [migration/2]
root          23  0.0  0.0      0     0 ?        S    12:53   0:00 [ksoftirqd/2]
root          25  0.0  0.0      0     0 ?        I<   12:53   0:00 [kworker/2:0H-events_highpri]
root          26  0.0  0.0      0     0 ?        S    12:53   0:00 [cpuhp/3]
root          27  0.0  0.0      0     0 ?        S    12:53   0:00 [migration/3]
root          28  0.0  0.0      0     0 ?        S    12:53   0:00 [ksoftirqd/3]
root          30  0.0  0.0      0     0 ?        I<   12:53   0:00 [kworker/3:0H-events_highpri]
root          35  0.0  0.0      0     0 ?        S    12:53   0:00 [kdevtmpfs]
root          36  0.0  0.0      0     0 ?        I<   12:53   0:00 [netns]
root          37  0.0  0.0      0     0 ?        S    12:53   0:00 [kauditd]
root          40  0.0  0.0      0     0 ?        S    12:53   0:00 [khungtaskd]
root          41  0.0  0.0      0     0 ?        S    12:53   0:00 [oom_reaper]
root          42  0.0  0.0      0     0 ?        I<   12:53   0:00 [writeback]
root          43  0.0  0.0      0     0 ?        S    12:53   0:00 [kcompactd0]
root          44  0.0  0.0      0     0 ?        SN   12:53   0:00 [ksmd]
root          45  0.1  0.0      0     0 ?        SN   12:53   0:02 [khugepaged]
root          47  0.4  0.0      0     0 ?        I    12:53   0:07 [kworker/3:1-events]
root          64  0.0  0.0      0     0 ?        I<   12:53   0:00 [kintegrityd]
root          65  0.0  0.0      0     0 ?        I<   12:53   0:00 [kblockd]
root          66  0.0  0.0      0     0 ?        I<   12:53   0:00 [blkcg_punt_bio]
root          67  0.0  0.0      0     0 ?        I<   12:53   0:00 [edac-poller]
root          68  0.0  0.0      0     0 ?        I<   12:53   0:00 [devfreq_wq]
root          70  0.0  0.0      0     0 ?        I<   12:53   0:00 [kworker/3:1H-kblockd]
root          71  0.0  0.0      0     0 ?        S    12:53   0:00 [kswapd0]
root          72  0.0  0.0      0     0 ?        I<   12:53   0:00 [kthrotld]
root          73  0.0  0.0      0     0 ?        I<   12:53   0:00 [acpi_thermal_pm]
root          74  0.0  0.0      0     0 ?        I<   12:53   0:00 [ipv6_addrconf]
root          75  0.0  0.0      0     0 ?        I<   12:53   0:00 [kworker/2:1H-kblockd]
root          84  0.0  0.0      0     0 ?        I<   12:53   0:00 [kstrp]
```

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

```
┌──(sabarish㉿Sabarish)-[~]
└─$ ps aux | grep bash
sabarish    1498  0.0  0.0   6180   648 pts/2    S+   13:22   0:00 grep --color=auto bash
```

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

```
┌──(sabarish㉿Sabarish)-[~]
└─$ ps aux | grep bash| wc -l
1
```