

Assignment 2 : Bash Shell Basics

:INSTRUCTIONS:

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.

COMMANDS AND SCREENSHOTS:

Task 1: File and Directory Manipulation

1. Create a directory called "my_directory".

```
mkdir my_directory
```

```
(kali㉿kali)-[~]  
└─$ mkdir my_directory
```

2. Navigate into the "my_directory".

```
cd my_directory
```

```
(kali㉿kali)-[~]  
└─$ cd my_directory
```

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

```
touch my_file.txt
```

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

2 □

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

```
ls
```

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

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

```
mv my_file.txt new_file.txt
```

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

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

```
less new_file.txt
```

```
(kali㉿kali)-[~/my_directory]
└─$ less new_file.txt
```

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

```
echo "Hello, World!" >> new_file.txt
```

```
(kali㉿kali)-[~/my_directory]
└─$ echo "Hello, World!" >> new_file.txt
```

```
dquote>
```

```
dquote> "Hello world
```

```
Hello, World >> new_file.txt
```

```
Hello world
```

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

mkdir backup

```
(kali㉿ kali)-[~/my_directory]
$ mkdir backup
```

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

mv new_file.txt backup/

```
(kali㉿ kali)-[~/my_directory]
$ mv new_file.txt backup/
```

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

ls backup

```
(kali㉿ kali)-[~/my_directory]
$ ls backup
new_file.txt
```

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

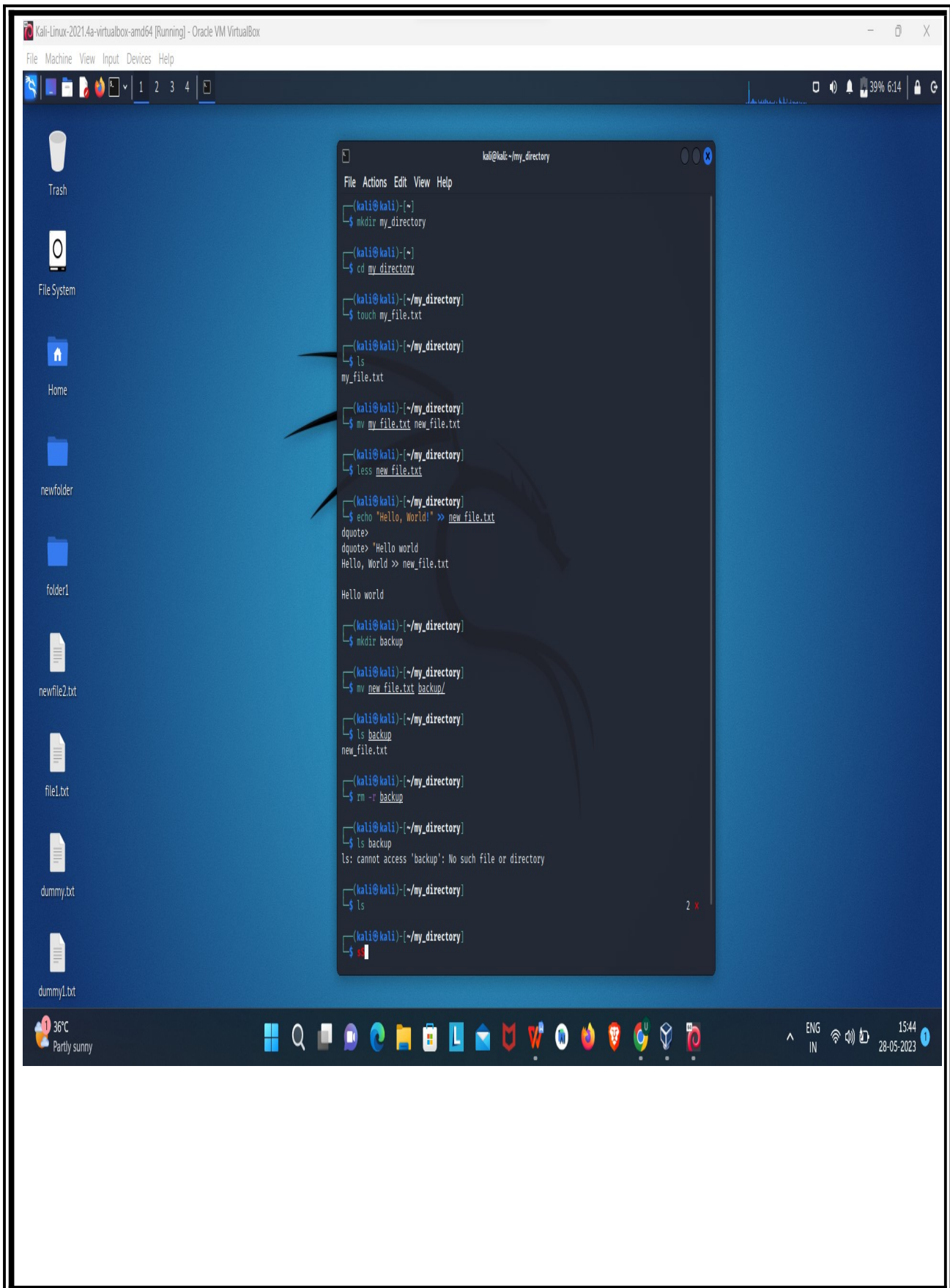
rm -r backup

```
(kali㉿ kali)-[~/my_directory]
$ rm -r backup
```

```
(kali㉿ kali)-[~/my_directory]
$ ls backup
```

ls: cannot access 'backup': No such file or directory

```
(kali㉿ kali)-[~/my_directory]
$ ls
```



Task 2: Permissions and Scripting

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

```
(kali㉿ kali)-[~/Desktop]  
$ touch my_script.sh
```

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

bash

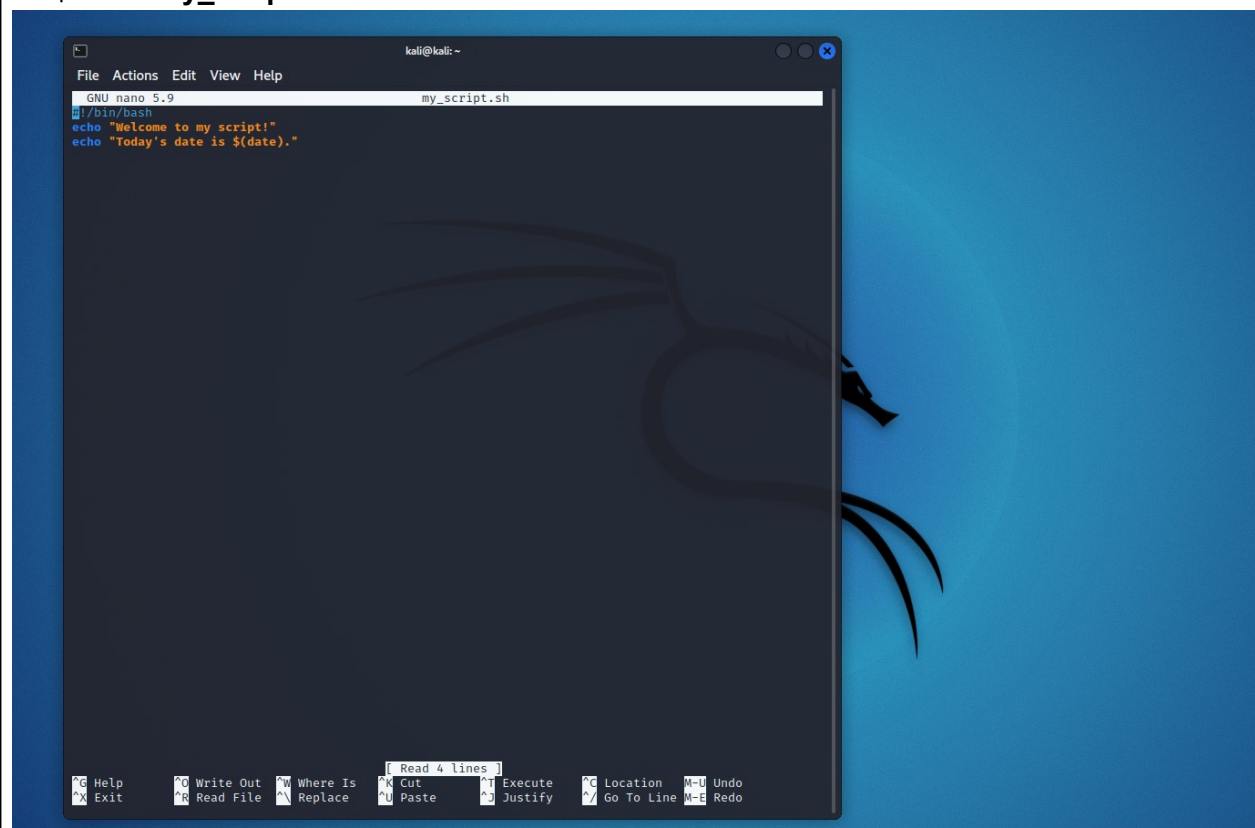
#!/bin/bash

echo "Welcome to my script!"

echo "Today's date is \$(date)."

Save and exit the file.

```
(kali㉿ kali)-[~/Desktop]  
$ nano my_script.sh
```



- Make "my_script.sh" executable.

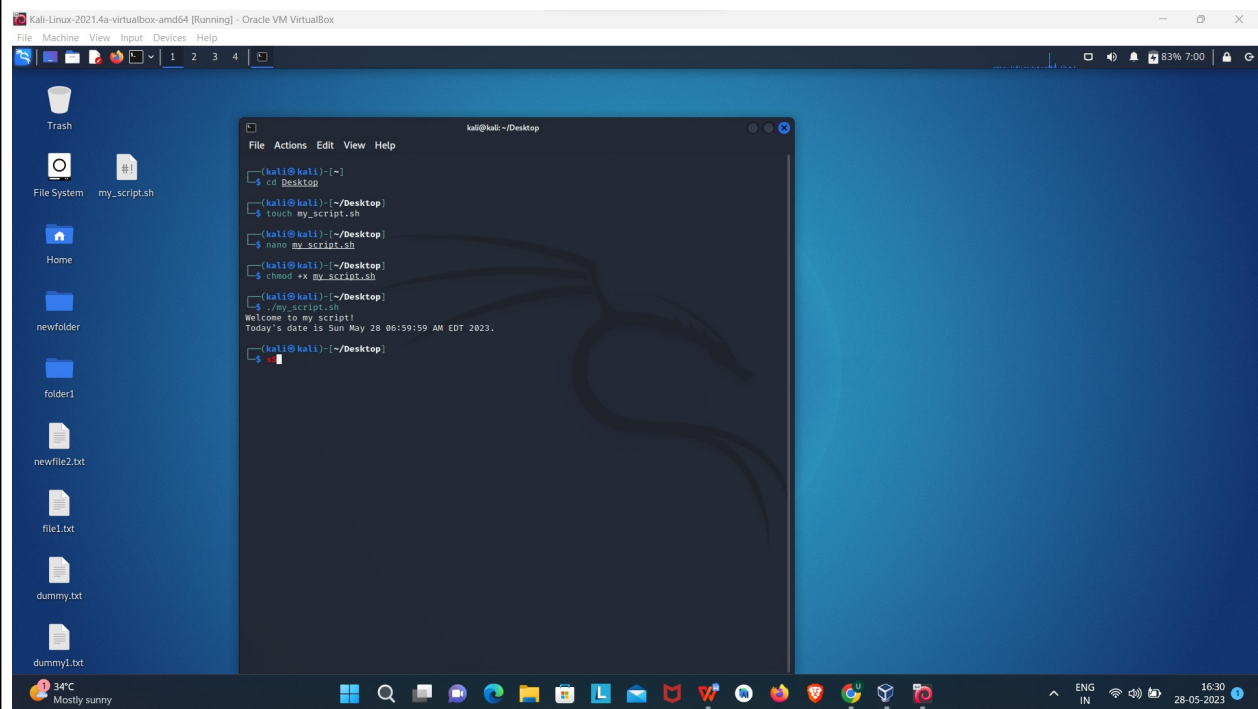
```
(kali㉿kali)-[~/Desktop]
$ chmod +x my_script.sh
```

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

```
(kali㉿kali)-[~/Desktop]
$ ./my_script.sh
```

Welcome to my script!

Today's date is Sun May 28 06:59:59 AM EDT 2023.



Task 3: Command Execution and Pipelines

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

└─(kali⊕ kali)-[~]
└─\$ ps aux

```

File Actions Edit View Help

[~] hali@hali:~$ cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 64 | xargs -n 1 sh

user PID KCPU NPMI SWIN VZS RSS TTV STAT START TIME COMMAND
root 1 0.0 0.0 164320 107608 0 0 0:01:14 0.00 /sbin/init splash
root 2 0.0 0.0 0 0 0 0:00:00 0.00 [kthreadd]
root 3 0.0 0.0 0 0 0 0:00:00 0.00 [rcu_gp]
root 4 0.0 0.0 0 0 0 0:00:00 0.00 [rcu_gpd]
root 5 0.0 0.0 0 0 0 0:00:00 0.00 [kworker/0:0-events_highpri]
root 6 0.0 0.0 0 0 0 0:00:00 0.00 [kmmq]
root 9 0.0 0.0 0 0 0 0:00:00 0.00 [rcu_tasks_rude_]
root 10 0.0 0.0 0 0 0 0:00:00 0.00 [rcu_tasks_trace]
root 11 0.0 0.0 0 0 0 0:00:00 0.00 [knsirqd]
root 12 0.0 0.0 0 0 0 0:00:00 0.00 [rcu_sched]
root 13 0.0 0.0 0 0 0 0:00:00 0.00 [migration/0]
root 14 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd0]
root 15 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd1]
root 17 0.0 0.0 0 0 0 0:00:00 0.00 [migration/1]
root 18 0.0 0.0 0 0 0 0:00:00 0.00 [knsirqd/1]
root 19 0.0 0.0 0 0 0 0:00:00 0.00 [kworker/1:0-events_highpri]
root 21 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd/1]
root 24 0.0 0.0 0 0 0 0:00:00 0.00 [kcsd]
root 25 0.0 0.0 0 0 0 0:00:00 0.00 [init_frag_wq]
root 26 0.0 0.0 0 0 0 0:00:00 0.00 [kcsd/1]
root 27 0.0 0.0 0 0 0 0:00:00 0.00 [shmgstask]
root 28 0.0 0.0 0 0 0 0:00:00 0.00 [com_parser]
root 29 0.0 0.0 0 0 0 0:00:00 0.00 [writeback]
root 30 0.0 0.0 0 0 0 0:00:00 0.00 [kccommon]
root 31 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd]
root 32 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd/1]
root 52 0.0 0.0 0 0 0 0:00:00 0.00 [ksterge[1234]
root 53 0.0 0.0 0 0 0 0:00:00 0.00 [kfsck]
root 54 0.0 0.0 0 0 0 0:00:00 0.00 [klogd_gst_bin]
root 55 0.0 0.0 0 0 0 0:00:00 0.00 [com_wq]
root 56 0.0 0.0 0 0 0 0:00:00 0.00 [udev-poller]
root 57 0.0 0.0 0 0 0 0:00:00 0.00 [com_frag_wq]
root 58 0.0 0.0 0 0 0 0:00:00 0.00 [kworker/0:16-hblockd]
root 59 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd]
root 60 0.0 0.0 0 0 0 0:00:00 0.00 [kthreadd]
root 61 0.0 0.0 0 0 0 0:00:00 0.00 [rcu1:thermal_gov]
root 62 0.0 0.0 0 0 0 0:00:00 0.00 [kfsck]
root 63 0.0 0.0 0 0 0 0:00:00 0.00 [kworker/7:16-hblockd]
root 64 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd_addrconf]
root 71 0.0 0.0 0 0 0 0:00:00 0.00 [kfsck]
root 77 0.0 0.0 0 0 0 0:00:00 0.00 [kswapd-shrink]
root 78 0.0 0.0 0 0 0 0:00:00 0.00 [kworker/1:0-]
root 127 0.0 0.0 0 0 0 0:00:00 0.00 [ata_fff]
root 138 0.0 0.0 0 0 0 0:00:00 0.00 [ata_ah_0]
root 139 0.0 0.0 0 0 0 0:00:00 0.00 [ata_tf_0]
root 140 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 141 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 142 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 143 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 144 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 145 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 146 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 147 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 148 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 149 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 150 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 151 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 152 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 153 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 154 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 155 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 156 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 157 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 158 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 159 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 160 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 161 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 162 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 163 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 164 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 165 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 166 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 167 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 168 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 169 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 170 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 171 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 172 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 173 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 174 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 175 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 176 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 177 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 178 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 179 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 180 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 181 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 182 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 183 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 184 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 185 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 186 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 187 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 188 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 189 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 190 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 191 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 192 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 193 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 194 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 195 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 196 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 197 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 198 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 199 0.0 0.0 0 0 0 0:00:00 0.00 [ata_wq]
root 
```

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

```
(kali㉿ kali)-[~]  
$ ps aux | grep bash
```

```
kali 21398 0.0 0.1 6184 2228 pts/0 S+ 06:43 0:00 grep --color=auto bash
```

```
(kali㉿kali)-[~]
└─$ ps aux | grep bash
kali      21398  0.0  0.1  6184  2228 pts/0    S+   06:43   0:00 grep --color=auto bash
```

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

```
(kali㉿kali)-[~]  
$ ps aux | grep bash | wc -l
```

1 ☐

1

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

Submitted By:

NAME: UTKARSH NARAYAN

REGISTER NUMBER: 20BKT0022