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
```

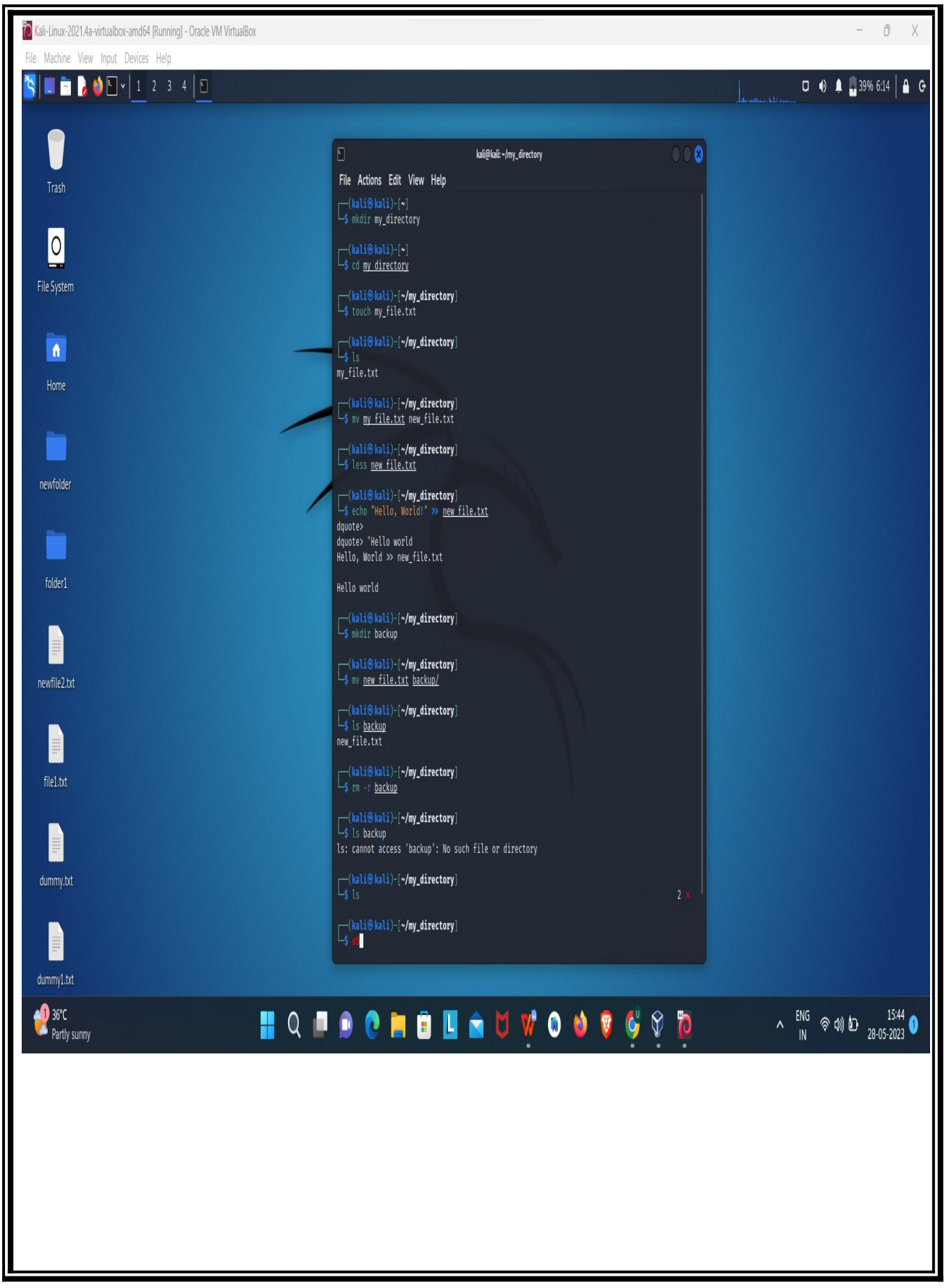
11. Delete the "backup" directory and all its contents.

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
```


Kali-Linux-2021.4a-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~/my_directory
File Actions Edit View Help
(kali㉿kali)-[~]
$ mkdir my_directory
(kali㉿kali)-[~]
$ cd my_directory
(kali㉿kali)-[~/my_directory]
$ touch my_file.txt
(kali㉿kali)-[~/my_directory]
$ ls
my_file.txt
(kali㉿kali)-[~/my_directory]
$ mv my_file.txt new_file.txt
(kali㉿kali)-[~/my_directory]
$ less new_file.txt
(kali㉿kali)-[~/my_directory]
$ echo "Hello, World!" >> new_file.txt
dquote>
dquote> 'Hello world
Hello, World >> new_file.txt
Hello world
(kali㉿kali)-[~/my_directory]
$ mkdir backup
(kali㉿kali)-[~/my_directory]
$ mv new_file.txt backup/
(kali㉿kali)-[~/my_directory]
$ ls backup
new_file.txt
(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
(kali㉿kali)-[~/my_directory]
$ ss
Trash
File System
Home
newfolder
folder1
newfile2.txt
file1.txt
dummy.txt
dummy1.txt
36°C
Partly sunny
ENG
IN
15:44
28-05-2023

Task 2: Permissions and Scripting

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

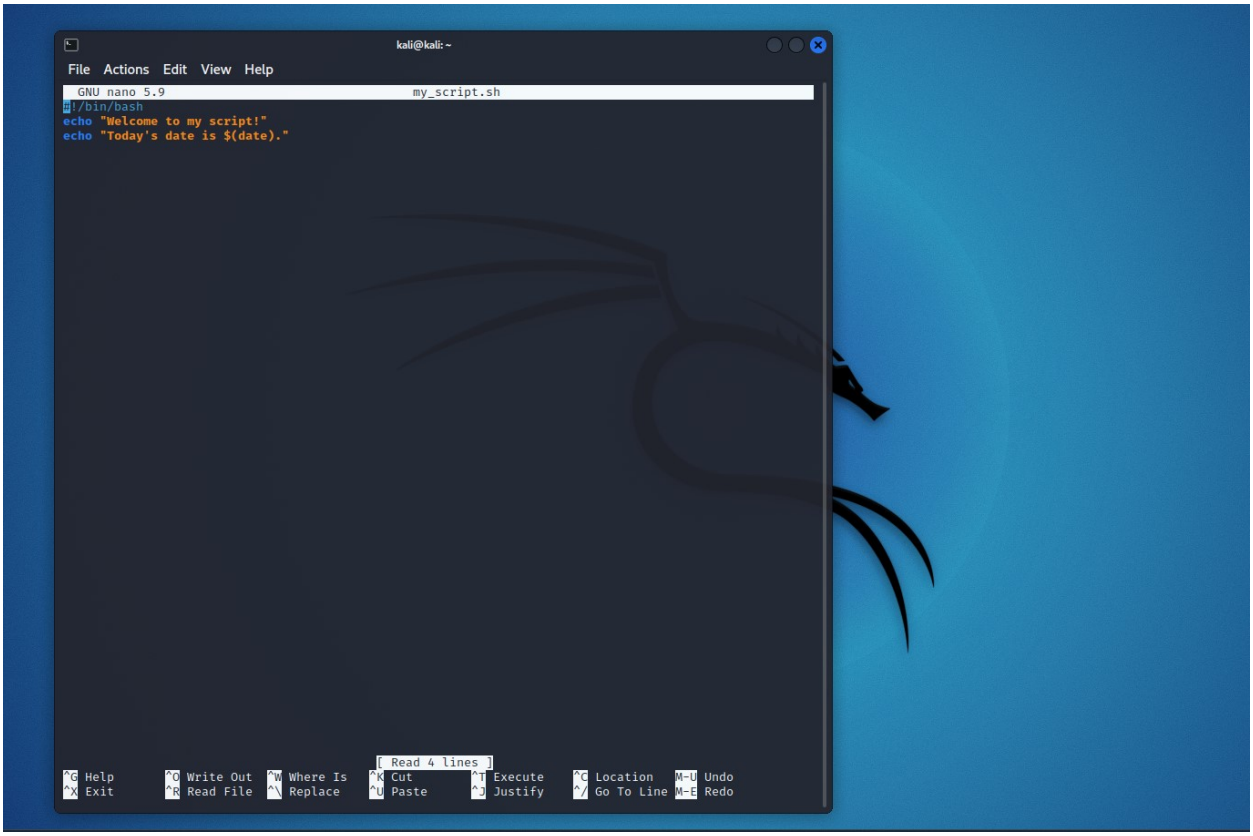
```
┌──(kali㉿kali)-[~/Desktop]
└─$ touch my_script.sh
```

- 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
```

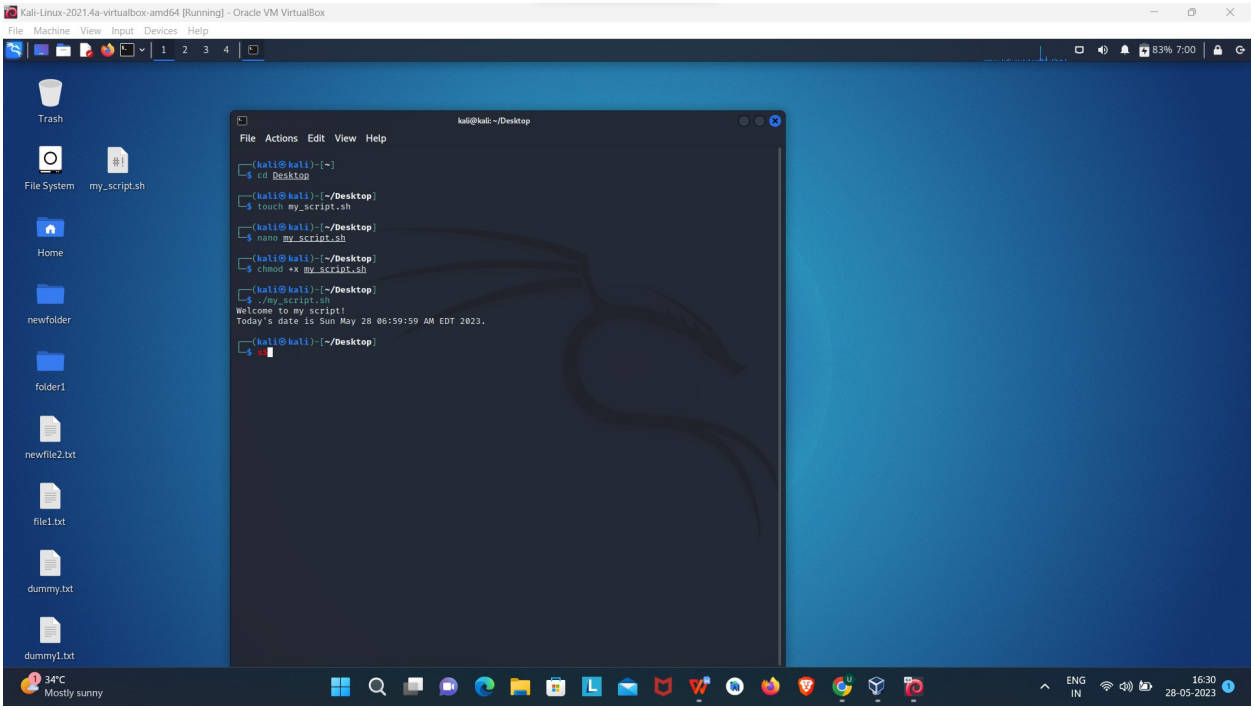


- Make "my_script.sh" executable.

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

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

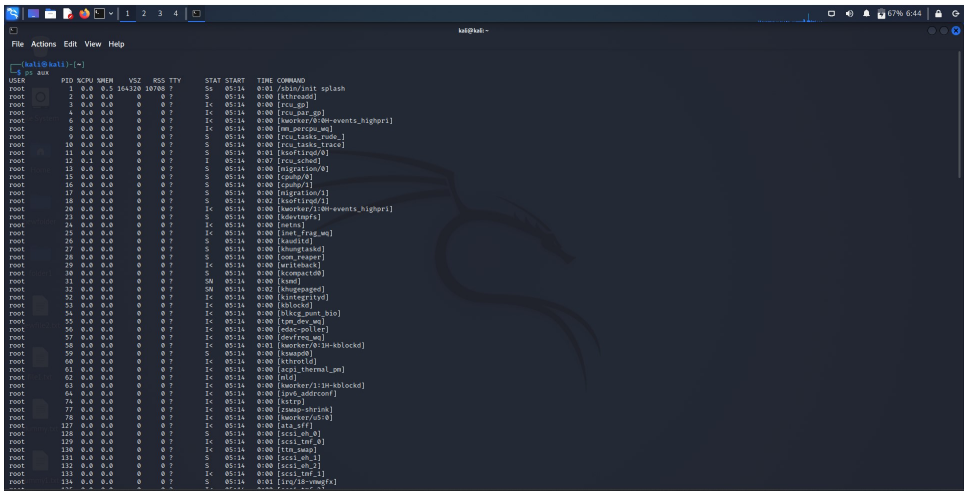
```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./my_script.sh
Welcome to my script!
Today's date is Sun May 28 06:59:59 AM EDT 2023.
```



Task 3: Command Execution and Pipelines

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

```
┌──(kali㉿kali)-[~]
└─$ ps aux
```



- 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
```

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

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

Submitted By:
NAME: UTKARSH NARAYAN
REGISTER NUMBER: 20BKT0022