

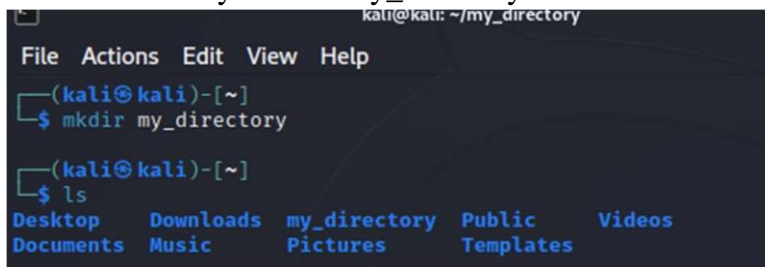
Assignment: Bash Shell Basics

NAME: RISHIK KASULA

REG.NO:20BCN7011

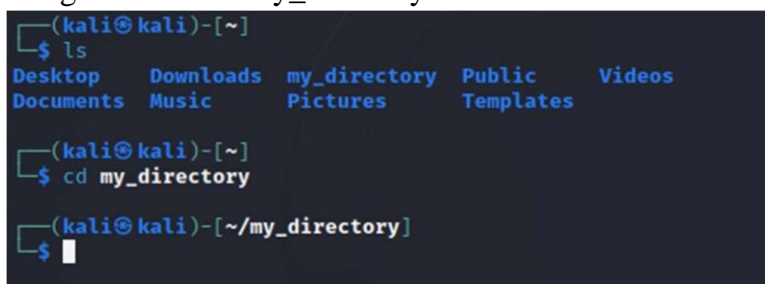
Task 1: File and Directory Manipulation:

1. Create a directory called "my_directory".

A terminal window on a Kali Linux system. The prompt is (kali@kali)-[~]. The user enters 'mkdir my_directory'. The prompt changes to (kali@kali)-[~/my_directory]. The user enters 'ls'. The output shows a list of directories: Desktop, Downloads, my_directory, Public, Videos, Documents, Music, Pictures, and Templates.

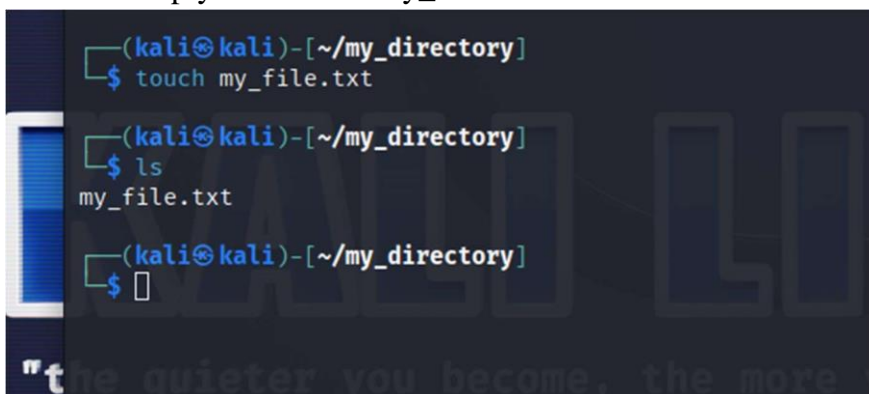
```
(kali@kali)-[~/my_directory]
File Actions Edit View Help
(kali@kali)-[~]
$ mkdir my_directory
(kali@kali)-[~]
$ ls
Desktop  Downloads  my_directory  Public  Videos
Documents Music      Pictures      Templates
```

2. Navigate into the "my_directory".

A terminal window on a Kali Linux system. The prompt is (kali@kali)-[~]. The user enters 'ls'. The prompt changes to (kali@kali)-[~/my_directory]. The user enters 'cd my_directory'. The prompt changes to (kali@kali)-[~/my_directory].

```
(kali@kali)-[~]
$ ls
Desktop  Downloads  my_directory  Public  Videos
Documents Music      Pictures      Templates
(kali@kali)-[~/my_directory]
$ cd my_directory
(kali@kali)-[~/my_directory]
$
```

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

A terminal window on a Kali Linux system. The prompt is (kali@kali)-[~/my_directory]. The user enters 'touch my_file.txt'. The prompt changes to (kali@kali)-[~/my_directory]. The user enters 'ls'. The output shows 'my_file.txt'. The prompt changes to (kali@kali)-[~/my_directory].

```
(kali@kali)-[~/my_directory]
$ touch my_file.txt
(kali@kali)-[~/my_directory]
$ ls
my_file.txt
(kali@kali)-[~/my_directory]
$
```

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

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

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

(kali㉿kali)-[~/my_directory]
$
```

5. Rename "my_file.txt" to "new_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]
$ ls
new_file.txt

(kali㉿kali)-[~/my_directory]
$
```

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

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

(kali㉿kali)-[~/my_directory]
$ cat new_file.txt
hi
this is karthik
```

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

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

(kali㉿kali)-[~/my_directory]
$ cat new_file.txt
Hello, World!
```

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

```

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

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

(kali㉿kali)-[~/my_directory]
$

```

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

```

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

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

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

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

(kali㉿kali)-[~/my_directory/backup]
$

```

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

```

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

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

(kali㉿kali)-[~/my_directory/backup]
$

```

```

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

(kali㉿kali)-[~/my_directory]
$ ls -la
total 12
drwxr-xr-x  3 kali kali   80 May 28 15:49 .
drwx----- 17 kali kali  600 May 28 15:48 ..
drwxr-xr-x  2 kali kali   60 May 28 15:49 backup
-rw-----  1 kali kali 12288 May 28 15:44 .new_file.swp

```

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

```

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

(kali㉿kali)-[~/my_directory]
$ ls

(kali㉿kali)-[~/my_directory]
$

```

Task 2: Permissions and Scripting

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

```

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

(kali㉿kali)-[~/my_directory]
$ ls
my_script.sh

(kali㉿kali)-[~/my_directory]
$

```

- 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.

```

File Actions Edit View Help
#!/bin/bash
echo "Welcome to my script!"
echo "Today's date is $(date)."
```

```

(kali㉿kali)-[~/my_directory]
$ vim my_script.sh

(kali㉿kali)-[~/my_directory]
$ cat my_script.sh
#!/bin/bash
echo "Welcome to my script!"
echo "Today's date is $(date)."
```

- Make "my_script.sh" executable.

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

(kali㉿kali)-[~/my_directory]
$ ls -la
total 16
drwxr-xr-x  2 kali kali   80 May 28 15:53 .
drwx----- 17 kali kali  600 May 28 15:53 ..
-rwxr-xr-x  1 kali kali   74 May 28 15:53 my_script.sh
-rw-----  1 kali kali 12288 May 28 15:44 .new_file.swp

(kali㉿kali)-[~/my_directory]
$
```

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

```
(kali㉿kali)-[~/my_directory]
$ ./my_script.sh
Welcome to my script!
Today's date is Sun May 28 03:54:41 PM UTC 2023.

(kali㉿kali)-[~/my_directory]
$
```

Task 3: Command Execution and Pipelines

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

```
(kali㉿kali)-[~/my_directory]
$ ps
  PID TTY          TIME CMD
 3453 pts/0        00:00:07 zsh
 7419 pts/0        00:00:00 vi
13738 pts/0        00:00:00 ps

(kali㉿kali)-[~/my_directory]
$
```

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

```
(kali㉿kali)-[~/my_directory]
$ ps -ef | grep "bash"
kali    14251    3453  0 15:57 pts/0    00:00:00 grep --color=auto bash

(kali㉿kali)-[~/my_directory]
$
```

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

```
(kali㉿kali)-[~/my_directory]
$ ps -ef | grep "bash" | wc -l
1
(kali㉿kali)-[~/my_directory]
$
```