

Assignment: Bash Shell Basics
Registration Number: 20BRS1237
Name: Kodhai U

Task 1: File and Directory Manipulation

1. Create a directory called "my_directory".

```
(kodhai@kali)-[~]
$ bash
(kodhai@kali)-[~]
$ ps
  PID TTY          TIME CMD
 1743 pts/0    00:00:00 zsh
 2059 pts/0    00:00:00 bash
 2086 pts/0    00:00:00 ps
(kodhai@kali)-[~]
$ mkdir my_directory
```

2. Navigate into the "my_directory".

```
(kodhai@kali)-[~]
$ cd my_directory
(kodhai@kali)-[~/my_directory]
$ ls
(kodhai@kali)-[~/my_directory]
$
```

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

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

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

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

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

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

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

```
(kodhai@kali)-[~/my_directory]
$ cat new_file.txt
Hello! This is Kodhai!
```

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

```

(kodhai@kali)-[~/my_directory]
$ sudo sh -c 'echo my_text >> new_file.txt'
[sudo] password for kodhai:

(kodhai@kali)-[~/my_directory]
$ sudo -- bash -c 'echo "Hello, World!" >> new_file.txt'

(kodhai@kali)-[~/my_directory]
$ cat new_file.txt
Hello! This is Kodhai!
my_text
Hello, World!

(kodhai@kali)-[~/my_directory]
$

```

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

```

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

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

```

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

```

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

```

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

```

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

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

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

```

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

```

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

(kodhai@kali)-[~/my_directory]
$ ls

```

Task 2: Permissions and Scripting

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

```

(kodhai@kali)-[~]
$ 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.
- Run "my_script.sh" and verify that the output matches the expected result.

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

(kodhai@kali) - [~/my_directory]
$ ./my_script.sh

(kodhai@kali) - [~/my_directory]
```

Task 3: Command Execution and Pipelines

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

```
(kodhai@kali) - [~/my_directory]
$ ps
  PID TTY          TIME CMD
 1743 pts/0        00:00:00 zsh
 2059 pts/0        00:00:00 bash
14304 pts/0        00:00:00 bash
14305 pts/0        00:00:00 bash
15347 pts/0        00:00:00 bash
15348 pts/0        00:00:00 bash
15439 pts/0        00:00:00 ps
```

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

```
(kodhai@kali) - [~/my_directory]
$ ps -e | grep -E 'bash'
 2059 pts/0        00:00:00 bash
14304 pts/0        00:00:00 bash
14305 pts/0        00:00:00 bash
15347 pts/0        00:00:00 bash
15348 pts/0        00:00:00 bash
```

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

```
(kodhai@kali) - [~/my_directory]
$ wc -l my_script.sh
5 my_script.sh
```