Assignment: Bash Shell Basics

Task 1: File and Directory Manipulation

1. Create a directory called "my_directory".

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop]
└─$ mkdir my_directory

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop]
└─$ ls
49.36.57.106     adit20bit0188.html   da5.pcapng   my_directory                          sample-mpg-file.mpg
ab.txt.gz.zip    arch1.tar            h.txt        Nessus-10.4.2-ubuntu1404_amd64.deb    vim.txt
adit             'da 1.pcapng'        index.html   sample-large-zip-file.zip             ZAP_2_12_0_unix.sh
```

2. Navigate into the "my_directory".

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop]
└─$ cd my_directory

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ touch my_file.txt
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ls
my_file.txt
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ mv my_file.txt new_file.txt

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ls
new_file.txt
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ more new_file.txt
this is adit wani's

NEW



FILE




(END)
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ echo 'Hello World!' >> new_file.txt

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ more new_file.txt
this is adit wani's

NEW



FILE
Hello World!
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ mkdir backup

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ l
backup/  new_file.txt

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ls
backup  new_file.txt
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ mv new_file.txt backup
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ls
backup

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ cd backup

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory/backup]
└─$ ls
new_file.txt
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ rm -rf backup

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ls
```

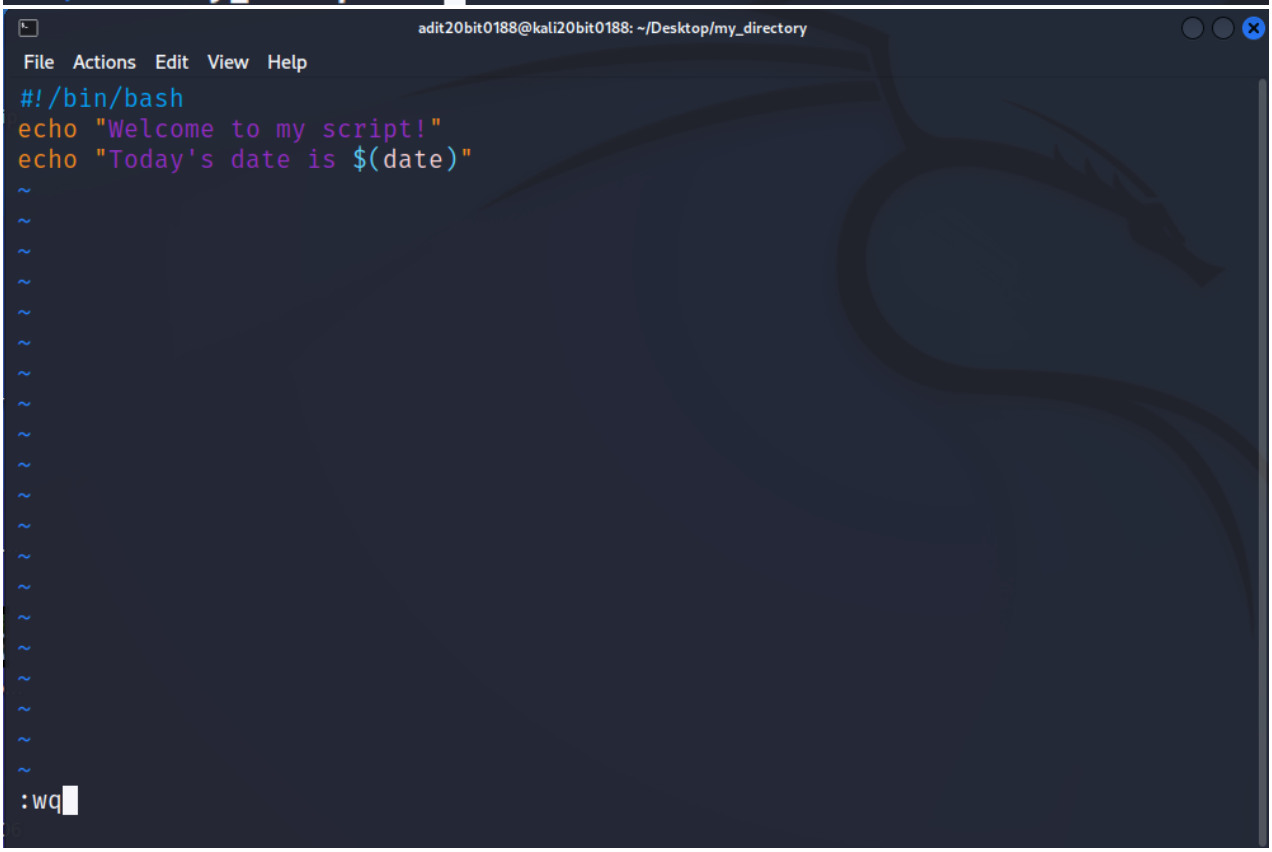
Task 2: Permissions and Scripting

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ touch my_script.sh
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ vim my_script.sh
```



:wq = write changes and exit vim

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ chmod +x my_script.sh

┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ./my_script.sh
Welcome to my script!
Today's date is Sunday 28 May 2023 02:16:12 PM IST
```

Task 3: Command Execution and Pipelines

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.3 167468 12116 ?        Ss   12:08   0:02 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    12:08   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   12:08   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   12:08   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   12:08   0:00 [netns]
root           7  0.0  0.0      0     0 ?        I<   12:08   0:00 [kworker/0:0H-events_highpri]
root           9  0.0  0.0      0     0 ?        I<   12:08   0:00 [kworker/0:1H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   12:08   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        I    12:08   0:00 [rcu_tasks_kthread]
root          12  0.0  0.0      0     0 ?        I    12:08   0:00 [rcu_tasks_rude_kthread]
root          13  0.0  0.0      0     0 ?        I    12:08   0:00 [rcu_tasks_trace_kthread]
root          14  0.0  0.0      0     0 ?        S    12:08   0:00 [ksoftirqd/0]
root          15  0.0  0.0      0     0 ?        I    12:08   0:07 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    12:08   0:00 [migration/0]
root          18  0.0  0.0      0     0 ?        S    12:08   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    12:08   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    12:08   0:00 [migration/1]
root          21  0.0  0.0      0     0 ?        S    12:08   0:00 [ksoftirqd/1]
root          23  0.0  0.0      0     0 ?        I<   12:08   0:00 [kworker/1:0H-events_highpri]
root          24  0.0  0.0      0     0 ?        S    12:08   0:00 [cpuhp/2]
root          25  0.0  0.0      0     0 ?        S    12:08   0:00 [migration/2]
root          26  0.0  0.0      0     0 ?        S    12:08   0:00 [ksoftirqd/2]
root          28  0.0  0.0      0     0 ?        I<   12:08   0:00 [kworker/2:0H-events_highpri]
root          29  0.0  0.0      0     0 ?        S    12:08   0:00 [cpuhp/3]
root          30  0.0  0.0      0     0 ?        S    12:08   0:00 [migration/3]
root          31  0.0  0.0      0     0 ?        S    12:08   0:00 [ksoftirqd/3]
root          33  0.0  0.0      0     0 ?        I<   12:08   0:00 [kworker/3:0H-events_highpri]
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ps aux | grep bash
adit20b+   35870  0.0  0.0   6228  2180 pts/0    S+   14:18   0:00 grep --color=auto bash
```

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

```
┌──(adit20bit0188㉿kali20bit0188)-[~/Desktop/my_directory]
└─$ ps aux | grep bash | wc -l
1
```

Submission:

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.