

Assignment 2 : Bash Shell Basics

Submitted By:

NAME: PRINCE GARG

REGISTER NUMBER: 20BKT0048

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.

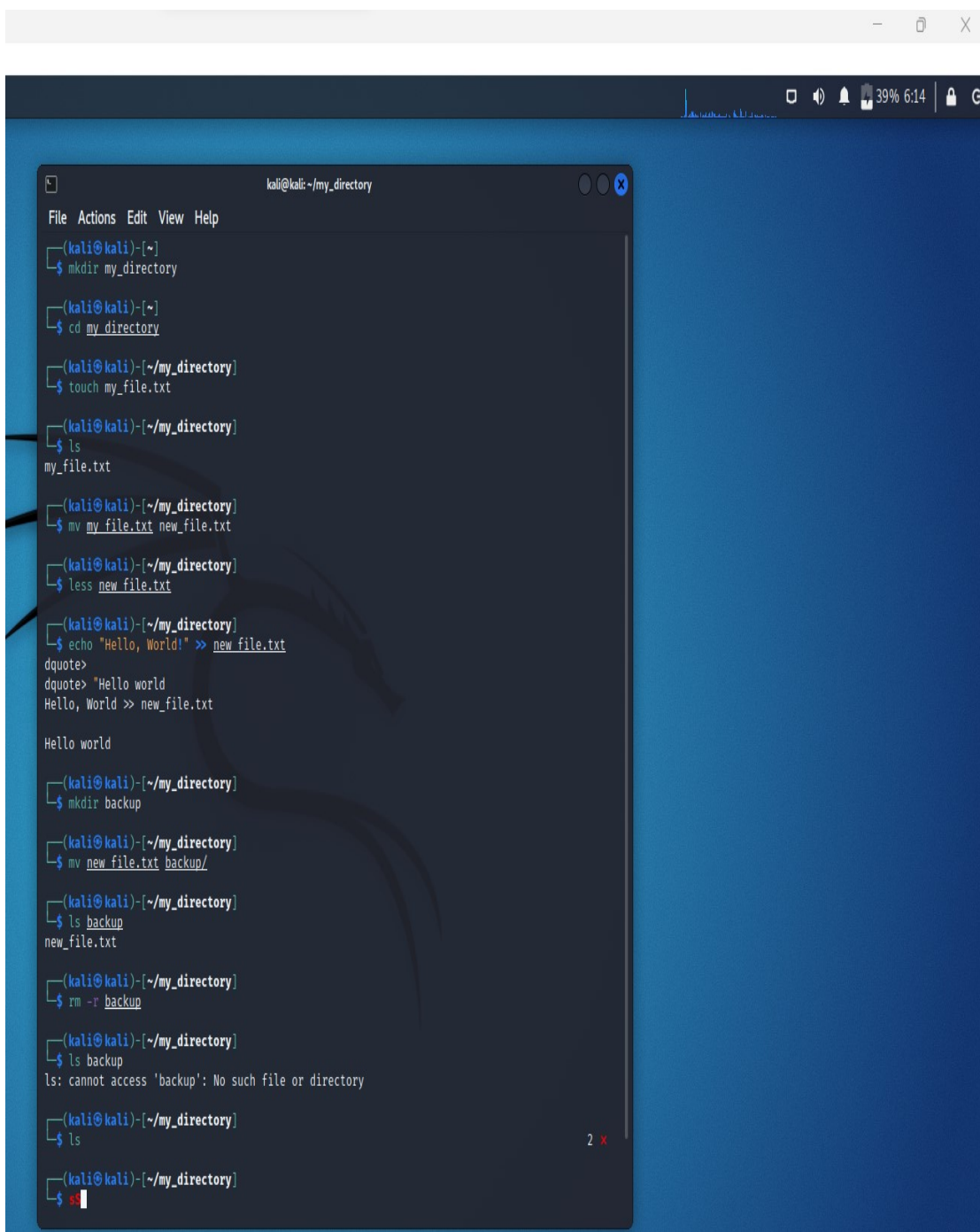
rm -r backup

```
(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@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]
$
```

Task 2: Permissions and Scripting

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

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

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

bash

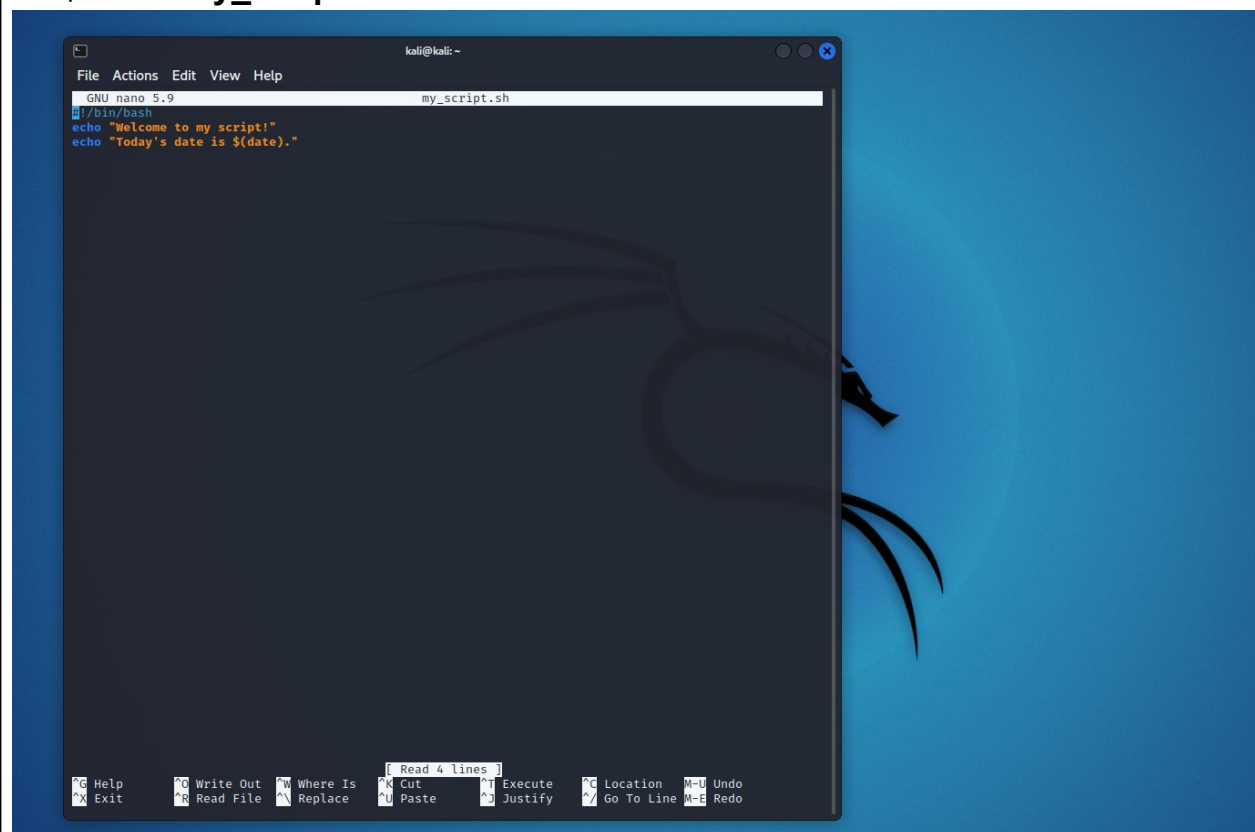
#!/bin/bash

echo "Welcome to my script!"

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

Save and exit the file.

```
(kali㉿kali)-[~]  
└─$ nano my_script.sh
```



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 5.9 my_script.sh  
#!/bin/bash  
echo "Welcome to my script!"  
echo "Today's date is $(date)."  
  
Read 4 lines  
Help Write Out Where Is Cut Execute Location M-U Undo  
Exit Read File Replace Paste Justify Go To Line M-E Redo
```

- Make "my_script.sh" executable.

```
(kali㉿kali)-[~]
```

└─\$ chmod +x my_script.sh

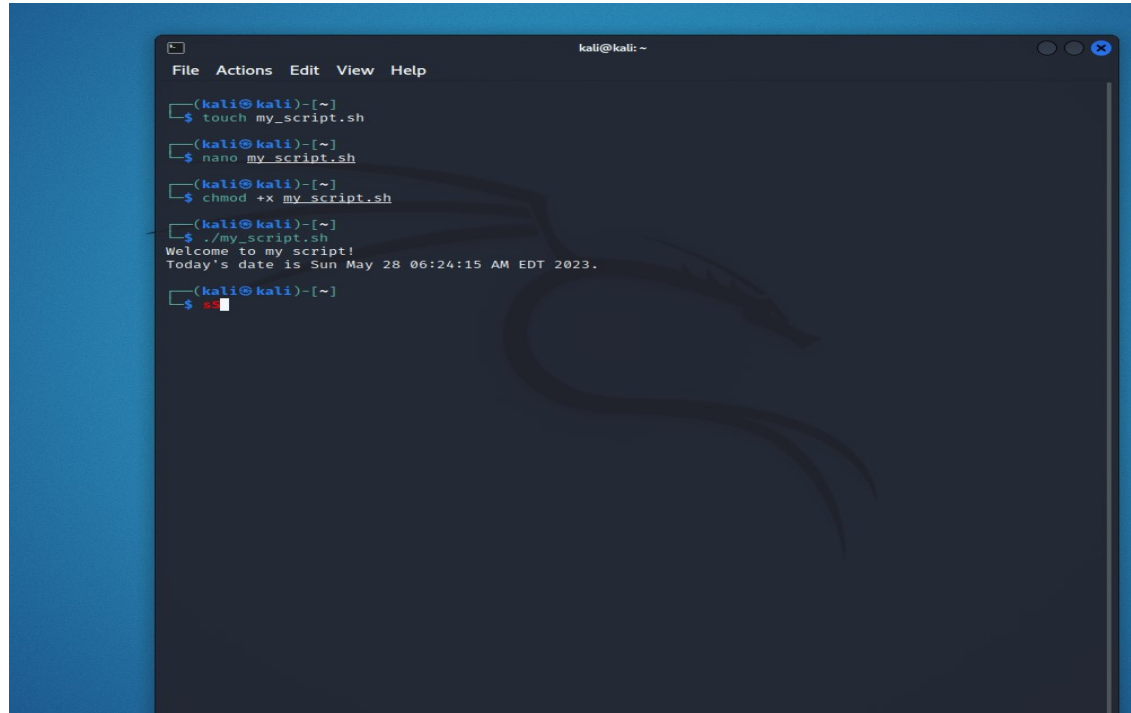
- Run "my_script.sh" and verify that the output matches the expected result.

└─(kali㉿kali)-[~]

└─\$./my_script.sh

Welcome to my script!

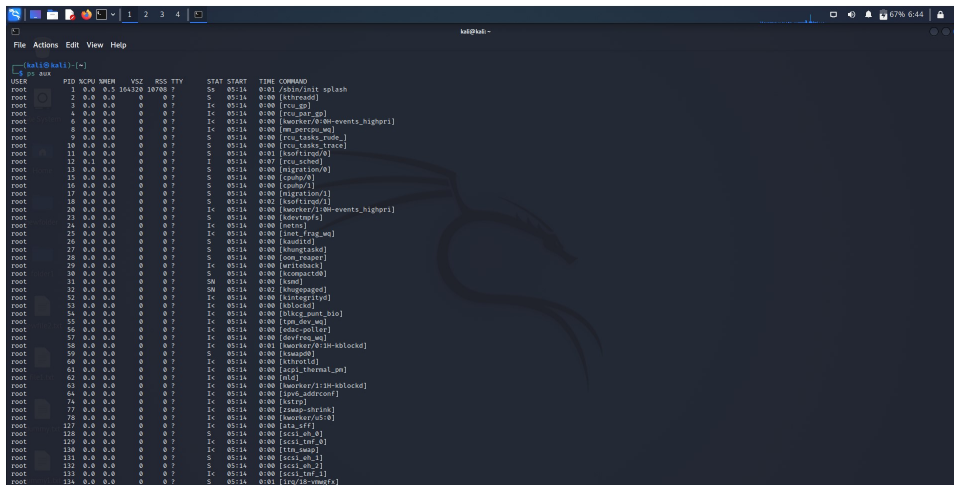
Today's date is Sun May 28 06:24:15 AM EDT 2023.

A screenshot of a terminal window titled 'kali@kali: ~'. The window shows a series of commands and their outputs. The commands are: 'touch my_script.sh', 'nano my_script.sh', 'chmod +x my_script.sh', and './my_script.sh'. The output of the last command is 'Welcome to my script!' followed by 'Today's date is Sun May 28 06:24:15 AM EDT 2023.'. The prompt is now 's\$'.

Task 3: Command Execution and Pipelines

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

```
(kali㉿kali)-[~]
$ ps aux
```



```

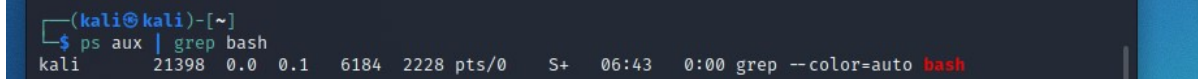
(kali㉿kali)-[~]
$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0  16120 18760 ?        Ss   05:14   0:01 /sbin/init splash
root           2  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthreadd]
root           3  0.0  0.0      0   0 ?        Ss   05:14   0:00 [rcu_gp]
root           4  0.0  0.0      0   0 ?        Ss   05:14   0:00 [rcu_bh_gp]
root           5  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kworker/0:0+events_highpri]
root           6  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kworker/0:0+events_lowpri]
root           7  0.0  0.0      0   0 ?        Ss   05:14   0:00 [rcu_tasks_rude_]
root           8  0.0  0.0      0   0 ?        Ss   05:14   0:00 [rcu_tasks_trace]
root          10  0.0  0.0      0   0 ?        Ss   05:14   0:00 [ksoftirqd/0]
root          11  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kworker/0:0]
root          12  0.0  0.0      0   0 ?        Ss   05:14   0:00 [migration/0]
root          13  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kswapd0]
root          14  0.0  0.0      0   0 ?        Ss   05:14   0:00 [migration/1]
root          15  0.0  0.0      0   0 ?        Ss   05:14   0:00 [ksoftirqd/1]
root          16  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kworker/1:0+events_highpri]
root          17  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kworker/1:0+events_lowpri]
root          18  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kdevtmpfs]
root          19  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kswapd1]
root          20  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kworker/1:1]
root          21  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          22  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          23  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          24  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          25  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          26  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          27  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          28  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          29  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          30  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          31  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          32  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          33  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          34  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          35  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          36  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          37  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          38  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          39  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          40  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          41  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          42  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          43  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          44  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          45  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          46  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          47  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          48  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          49  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          50  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          51  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          52  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          53  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          54  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          55  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          56  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          57  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          58  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          59  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          60  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          61  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          62  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          63  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          64  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          65  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          66  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          67  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          68  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          69  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          70  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          71  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          72  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          73  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          74  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          75  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          76  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          77  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          78  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          79  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          80  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          81  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          82  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          83  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          84  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          85  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          86  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          87  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          88  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          89  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          90  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          91  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          92  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          93  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          94  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          95  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          96  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          97  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          98  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root          99  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]
root         100  0.0  0.0      0   0 ?        Ss   05:14   0:00 [kthrottld]

```

- 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



```

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

```

ASSESSMENT1 FILE LINK

<https://drive.google.com/file/d/1RI6mKVLmWyWyuw7YtcMTkPdoKqDNEH-j/view?usp=sharing>

ASSESSMENT2 FILE LINK

<https://drive.google.com/file/d/1BDK14I9HdQwdZxHHmMugpM0IZqCUQcWV/view?usp=sharing>

FOLDER LINK

https://drive.google.com/drive/folders/1M65vhtpONkvHRYGQ0yljcfU_wgKe1hDu?usp=sharing