

ASSIGNMENT – 2

TEAM MEMBERS:

MEMBER 1: VENNAPUSA SARATHENDRA VENKATA SAI RAM REDDY (20FE1A03B1).

MEMBER 2: YARRAMSETTY VENKATA RAVI KIRAN (20FE1A03B3).

MEMBER 3: YERRABOTHULA NAGA VEERA VENKATESWAR REDDY (20FE1A03B6).

PROJRCT REG NO: SBAP0007919.

Bash Shell Basics

Task 1: File and Directory Manipulation

1. Create a directory called "my_directory".
2. Navigate into the "my_directory".
3. Create an empty file called "my_file.txt".
4. List all the files and directories in the current directory.
5. Rename "my_file.txt" to "new_file.txt".
6. Display the content of "new_file.txt" using a pager tool of your choice.
7. Append the text "Hello, World!" to "new_file.txt".
8. Create a new directory called "backup" within "my_directory".
9. Move "new_file.txt" to the "backup" directory.
10. Verify that "new_file.txt" is now located in the "backup" directory.
11. Delete the "backup" directory and all its contents.

COMMANDS

```
(chiranjibi@kali)-[~]  
$ mkdir my_directory  
  
(chiranjibi@kali)-[~]  
$ cd my_directory  
  
(chiranjibi@kali)-[~/my_directory]  
$ touch my_file.txt  
File System  
(chiranjibi@kali)-[~/my_directory]  
$ ls  
my_file.txt  
  
(chiranjibi@kali)-[~/my_directory]  
$ mv my_file.txt new_file.txt  
Home  
(chiranjibi@kali)-[~/my_directory]  
$ echo "hello, world!" >> new_file.txt
```

```
(chiranjibi@kali)-[~]  
$ mkdir backup  
  
(chiranjibi@kali)-[~]  
$ mv new_file.txt backup/  
mv: cannot stat 'new_file.txt': No such file or directory  
  
(chiranjibi@kali)-[~]  
$ ls backup/  
  
(chiranjibi@kali)-[~]  
$ rm -r backup  
  
(chiranjibi@kali)-[~]  
$
```

Task 2: Permissions and Scripting

- Create a new file called "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.

```
(chiranjibi@kali)-[~]
$ touch my_script.sh

(chiranjibi@kali)-[~]
$ less new_file.txt
new_file.txt: No such file or directory

(chiranjibi@kali)-[~]
$ nano my_script.sh

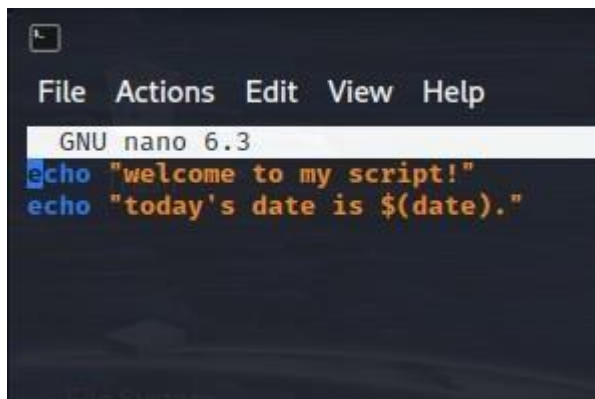
(chiranjibi@kali)-[~]
$ chmod +x my_script.sh

(chiranjibi@kali)-[~]
$ ./my_script.sh
./my_script.sh: 1: welcome: not found
./my_script.sh: 3: Syntax error: Unterminated quoted string

(chiranjibi@kali)-[~]
$ nano my_script.sh

(chiranjibi@kali)-[~]
$ ./my_script.sh
```

NANO



The screenshot shows the nano text editor interface. At the top, there is a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below the menu bar, the text 'GNU nano 6.3' is displayed. The main editing area contains two lines of code: `echo "welcome to my script!"` and `echo "today's date is $(date)."`. The first line is highlighted in blue.

Task 3: Command Execution and Pipelines

- List all the processes running on your system using the "ps" command.
- Use the "grep" command to filter the processes list and display only the processes with "bash" in their name.

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

```
File Actions Edit View Help
(chiranjibi@kali)-[~]
$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2 102104 12164 ?        Ss   04:47   0:01 /sbin/init splash
root         2  0.0  0.0      0     0 ?        S    04:47   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   04:47   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   04:47   0:00 [rcu_par_gp]
root         5  0.0  0.0      0     0 ?        I<   04:47   0:00 [netns]
root         7  0.0  0.0      0     0 ?        I<   04:47   0:00 [kworker/0:0H-events_highpri]
root         9  0.0  0.0      0     0 ?        I<   04:47   0:00 [kworker/0:1H-events_highpri]
root        10  0.0  0.0      0     0 ?        I<   04:47   0:00 [mm_percpu_wq]
root        11  0.0  0.0      0     0 ?        I    04:47   0:00 [rcu_tasks_kthread]
root        12  0.0  0.0      0     0 ?        I    04:47   0:00 [rcu_tasks_rude_kthread]
root        13  0.0  0.0      0     0 ?        I    04:47   0:00 [rcu_tasks_trace_kthread]
root        14  0.0  0.0      0     0 ?        S    04:47   0:00 [ksoftirqd/0]
root        15  0.0  0.0      0     0 ?        I    04:47   0:05 [rcu_preempt]
root        16  0.0  0.0      0     0 ?        S    04:47   0:00 [migration/0]
root        18  0.0  0.0      0     0 ?        S    04:47   0:00 [cpuhp/0]
root        20  0.0  0.0      0     0 ?        S    04:47   0:00 [kdevtmpfs]
root        21  0.0  0.0      0     0 ?        I<   04:47   0:00 [inet_frag_wq]
root        22  0.0  0.0      0     0 ?        S    04:47   0:00 [kauditd]
root        23  0.0  0.0      0     0 ?        S    04:47   0:00 [khungtaskd]
root        24  0.0  0.0      0     0 ?        S    04:47   0:00 [oom_reaper]
root        25  0.0  0.0      0     0 ?        I<   04:47   0:00 [writeback]
root        26  0.0  0.0      0     0 ?        S    04:47   0:00 [kcompactd0]
root        27  0.0  0.0      0     0 ?        SN   04:47   0:00 [ksmd]
root        28  0.0  0.0      0     0 ?        SN   04:47   0:00 [khugepaged]
root        29  0.0  0.0      0     0 ?        I<   04:47   0:00 [kintegrityd]
root        30  0.0  0.0      0     0 ?        I<   04:47   0:00 [kblockd]
root        31  0.0  0.0      0     0 ?        I<   04:47   0:00 [blkcg_punt_bio]
root        32  0.0  0.0      0     0 ?        I<   04:47   0:00 [tpm_dev_wq]
root        33  0.0  0.0      0     0 ?        I<   04:47   0:00 [edac-poller]
root        34  0.0  0.0      0     0 ?        I<   04:47   0:00 [devfreq_wq]

(chiranjibi@kali)-[~]
$ ps aux | grep bash
chiranj+ 48386 0.0 0.0 6348 2160 pts/1    S+   07:48   0:00 grep --color=auto bash

(chiranjibi@kali)-[~]
$ ps aux | grep bash | wc -l
wc: invalid option -- '1'
Try 'wc --help' for more information.

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

(chiranjibi@kali)-[~]
$
```