

20BCE1210-VISHNURAM G
VIT CHENNAI

Assignment: 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.

```
gvr@EXCALIBUR: ~/my_direct × + v
(gvr@EXCALIBUR)~[~]
$ mkdir my_directory

(gvr@EXCALIBUR)~[~]
$ cd my_directory

(gvr@EXCALIBUR)~/my_directory]
$ touch my_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ ls
my_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ mv my_file.txt new_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ ls
new_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ less new_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ echo "Hello, World!" >> new_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ cat new_file.txt
Hello, World!
```

```

(gvr@EXCALIBUR)~/my_directory]
$ mkdir backup

(gvr@EXCALIBUR)~/my_directory]
$ ls
backup  new_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ mv new_file.txt backup/

(gvr@EXCALIBUR)~/my_directory]
$ ls backup
new_file.txt

(gvr@EXCALIBUR)~/my_directory]
$ ls
backup

(gvr@EXCALIBUR)~/my_directory]
$ rm -r backup

(gvr@EXCALIBUR)~/my_directory]
$ ls

(gvr@EXCALIBUR)~/my_directory]
$ |

```

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.

```
gvr@EXCALIBUR: ~  
  
(gvr@EXCALIBUR)-[~]  
$ ls  
Desktop      Music      Public      Videos  
Documents    my_script.sh  Templates  
Downloads    Pictures     'Tor Browser'  
  
(gvr@EXCALIBUR)-[~]  
$ cat my_script.sh  
#!/bin/bash  
echo "Welcome to my script!"  
echo "Today's date is $(date)."  
  
(gvr@EXCALIBUR)-[~]  
$ chmod +x my_script.sh  
  
(gvr@EXCALIBUR)-[~]  
$ ./my_script.sh  
Welcome to my script!  
Today's date is Sun May 28 01:09:57 PM IST 2023.  
  
(gvr@EXCALIBUR)-[~]  
$ |
```

[illegible]

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.

```
gvr@EXCALIBUR: ~  
$ ps aux  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root         1  0.0  0.0   2324   1500 ?        SL    12:50    0:00 /init  
root         4  0.0  0.0   2324     4 ?        SL    12:50    0:00 plan9 --control-socket 5 --log-level 4 --server-fd 6 --  
root         7  0.0  0.0   2328   104 ?        Ss    12:50    0:00 /init  
root         8  0.0  0.0   2344   108 ?        S     12:50    0:00 /init  
gvr         9  0.0  0.0   7244   3816 pts/0    Ss    12:50    0:00 -bash  
gvr        22 33.3  0.1  11052  4348 pts/0    R+    12:59    0:00 ps aux  
  
$ ps aux|grep bash  
gvr         9  0.0  0.1   7244   3928 pts/0    Ss    12:50    0:00 -bash  
gvr        24  0.0  0.0   6404   1920 pts/0    S+    12:59    0:00 grep --color=auto bash  
  
$ ps aux|grep bash|wc -l  
3
```