Assessment-2

Assignment: Bash Shell Basics

Name: B Venkata Mounish Reddy
Reg No: 20BDS0149
Campus: VIT Vellore

Task 1: File and Directory Manipulation

1. Create a directory called “my_directory”.

```
mkdir my_dir
```

2. Navigate into the “my_directory”.

```
cd my_dir
```

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

```
touch my_file.txt
```

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

```
ls
```

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

```
mv my_file.txt new_file.txt
```

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

```
cat new_file.txt
```

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

```
echo "Hello, World!" >> new_file.txt
```

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

```
mkdir backup
```

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

```
mv new_file.txt backup/
```

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

```
ls backup
```

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

```
rm -r backup
```

Output:


Shell
Console
~/ShamelessTechnologicalApplicationserver$ echo assessment-2
assessment-2
~/ShamelessTechnologicalApplicationserver$ mkdir my_dir
~/ShamelessTechnologicalApplicationserver$ cd my_dir
~/ShamelessTechnologicalApplicationserver/my_dir$ touch my_file.txt
~/ShamelessTechnologicalApplicationserver/my_dir$ ls
my_file.txt
~/ShamelessTechnologicalApplicationserver/my_dir$ mv my_file.txt new_file.txt
~/ShamelessTechnologicalApplicationserver/my_dir$ cat new_file.txt
~/ShamelessTechnologicalApplicationserver/my_dir$ echo "Hello, World!" >> new_file.txt
~/ShamelessTechnologicalApplicationserver/my_dir$ mkdir backup
~/ShamelessTechnologicalApplicationserver/my_dir$ mv new_file.txt backup/
~/ShamelessTechnologicalApplicationserver/my_dir$ ls backup
new_file.txt
~/ShamelessTechnologicalApplicationserver/my_dir$ rm -r backup
~/ShamelessTechnologicalApplicationserver/my_dir$ cat new_file.txt
cat: new_file.txt: No such file or directory
~/ShamelessTechnologicalApplicationserver/my_dir$

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.

Code:

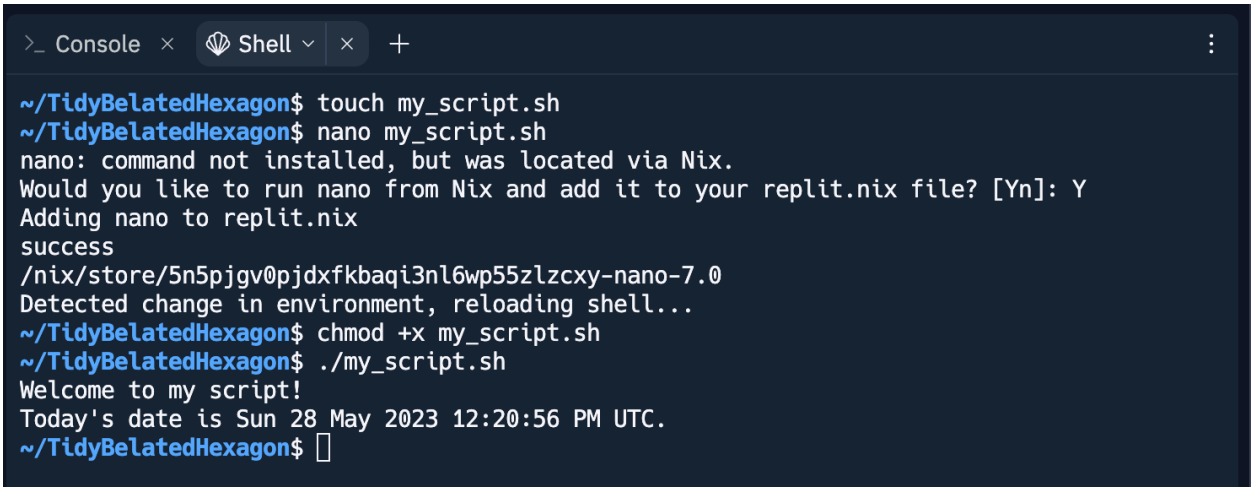
```
touch my_script.sh
nano my_script.sh
```

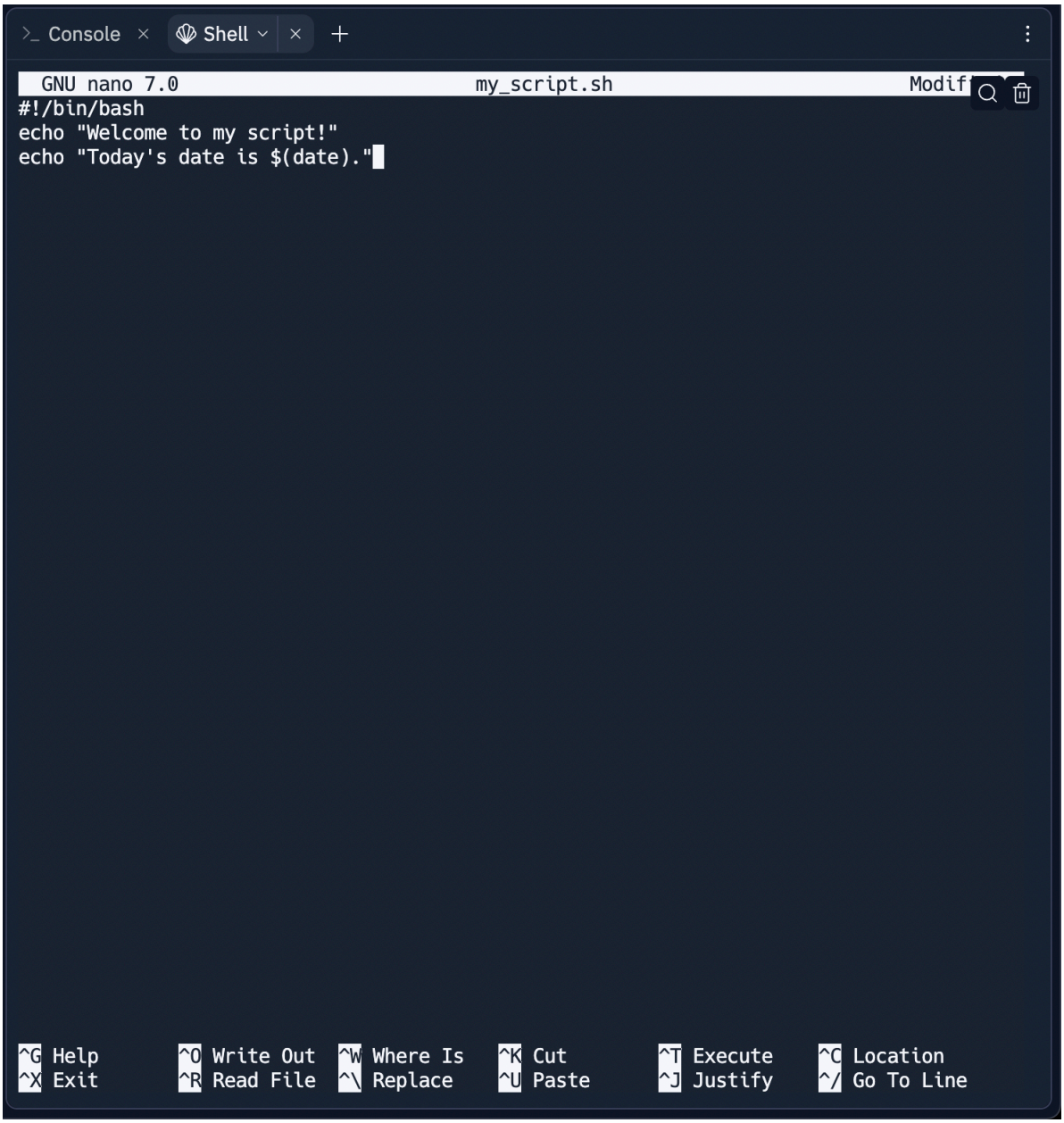
Save and exit the file in nano. In nano, you can press "Ctrl + o" to save the file, then Enter to confirm, and "Ctrl + x" to exit the editor.

```
#!/bin/bash
echo "Welcome to my script!"
echo "Today's date is $(date)."
chmod +x my_script.sh
# This command gives the file executable permissions.

./my_script.sh
```

Output:

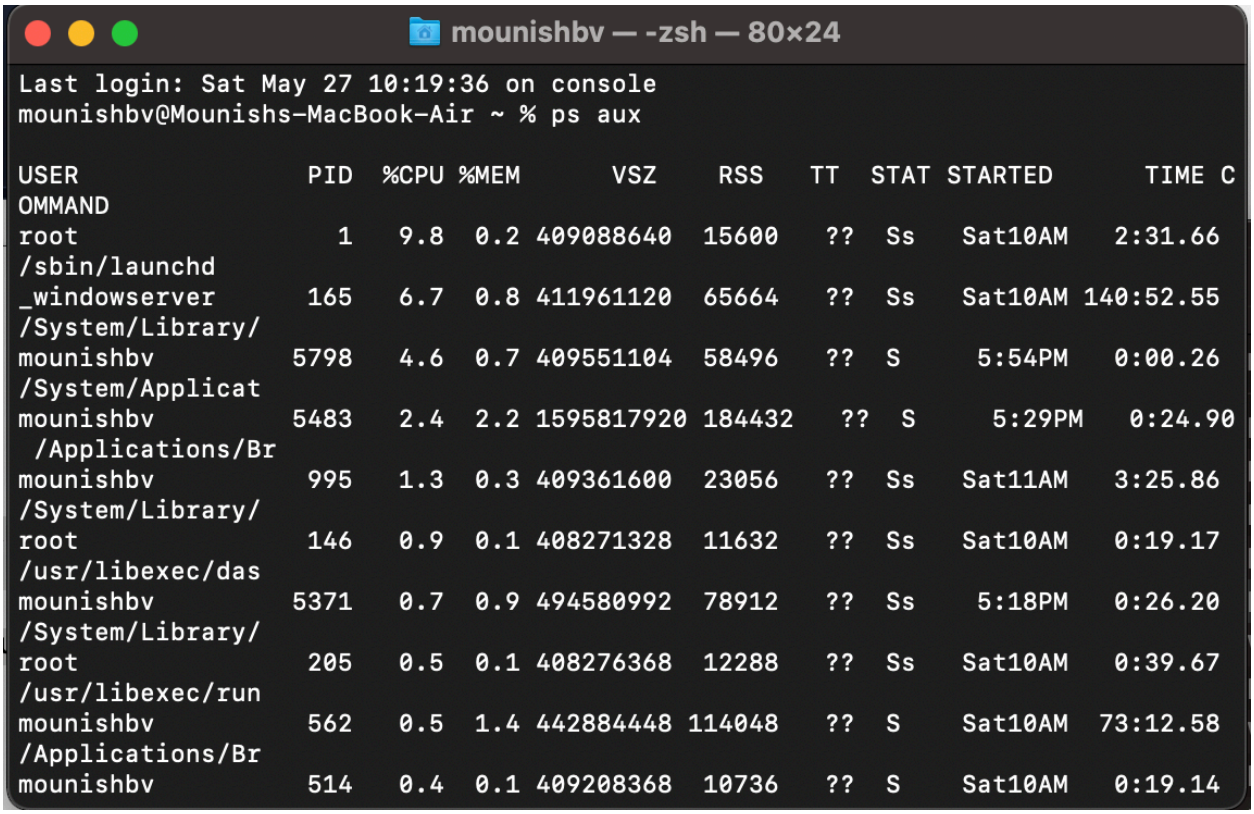

Console
Shell
~/TidyBelatedHexagon$ touch my_script.sh
~/TidyBelatedHexagon$ nano my_script.sh
nano: command not installed, but was located via Nix.
Would you like to run nano from Nix and add it to your replit.nix file? [Yn]: Y
Adding nano to replit.nix
success
/nix/store/5n5pjgv0pjdxfkbaqi3nl6wp55zlzcxy-nano-7.0
Detected change in environment, reloading shell...
~/TidyBelatedHexagon$ chmod +x my_script.sh
~/TidyBelatedHexagon$ ./my_script.sh
Welcome to my script!
Today's date is Sun 28 May 2023 12:20:56 PM UTC.
~/TidyBelatedHexagon$


Console
Shell
GNU nano 7.0 my_script.sh
#!/bin/bash
echo "Welcome to my script!"
echo "Today's date is $(date)."
^G Help
^O Write Out
^W Where Is
^K Cut
^T Execute
^C Location
^X Exit
^R Read File
^\ Replace
^U Paste
^J Justify
^/ Go To Line

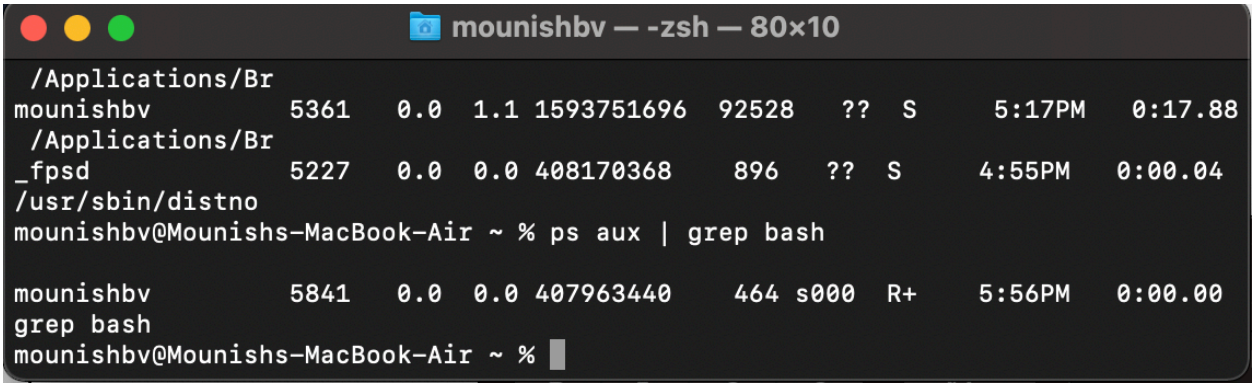
Task 3: Command Execution and Pipelines

- 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.

1. List all processes using the `ps` command:



2. Filter the processes using `grep` to display only processes with "bash" in their name:



3. Count the number of lines in the filtered output using `wc`: