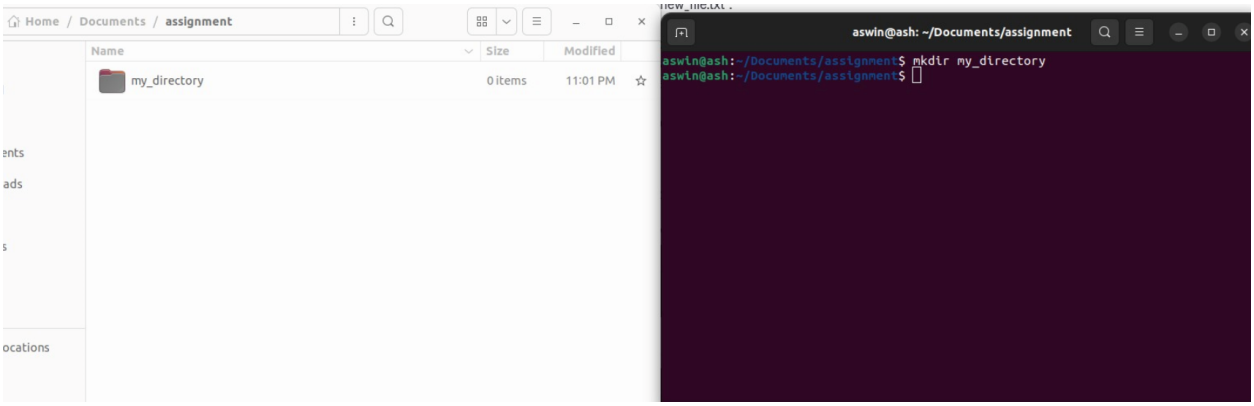
Name: Aswin Shailajan

Reg No: 20BCE10209

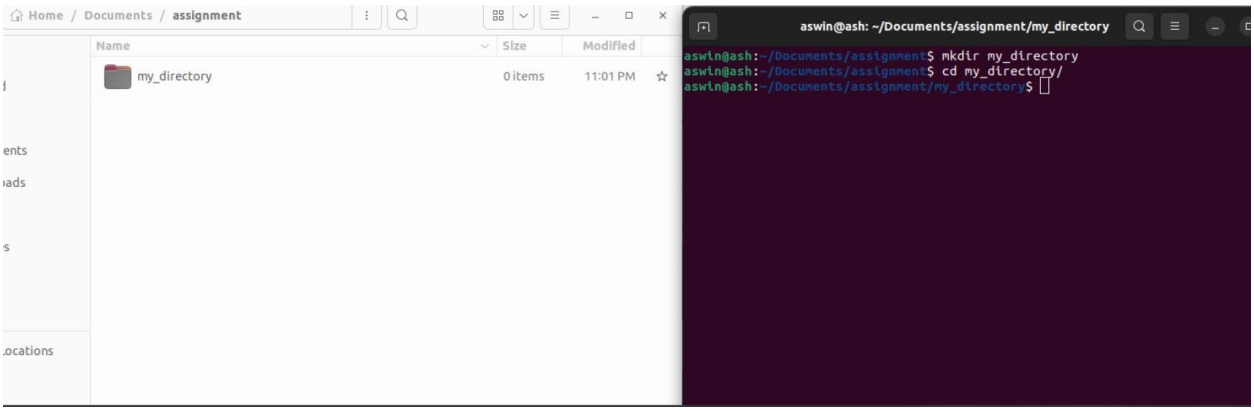
Assignment: Bash Shell Basics

Task 1: File and Directory Manipulation

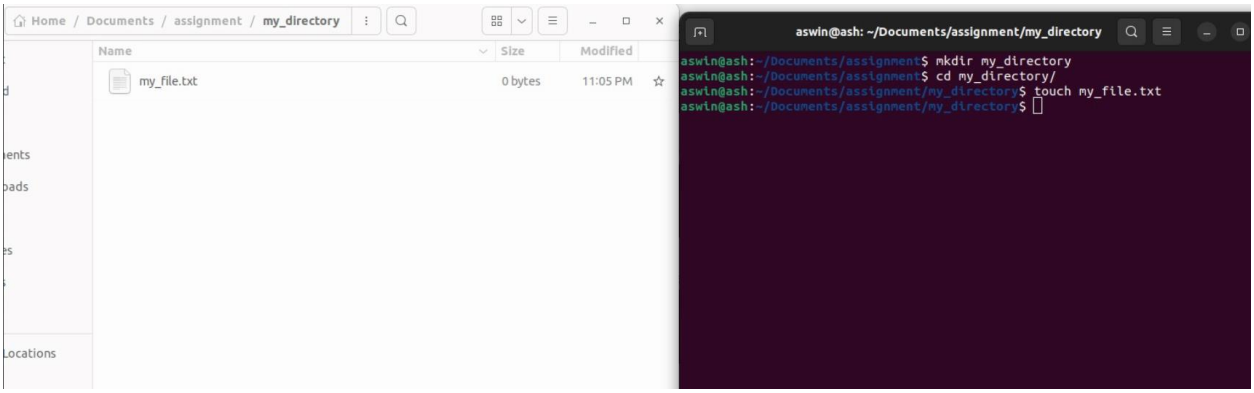
1. Create a directory called "my_directory".



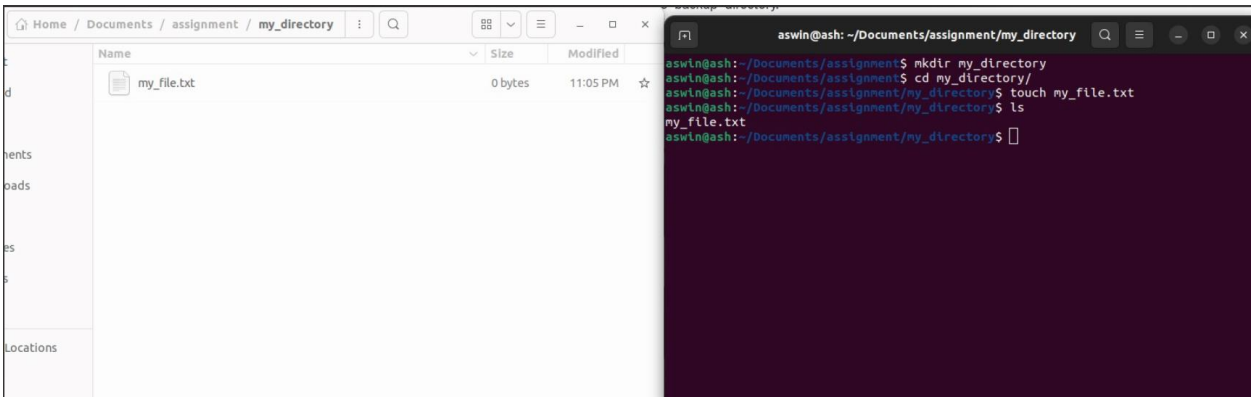
2. Navigate into the "my_directory".



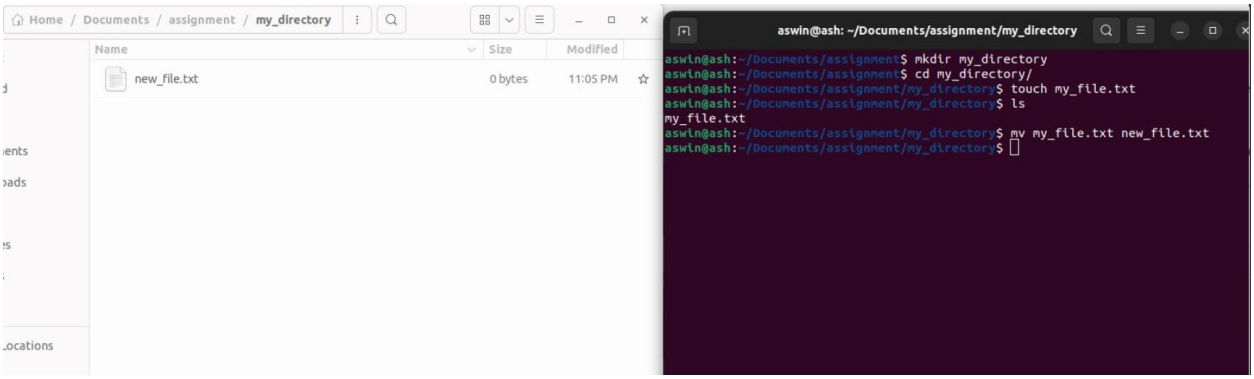
3. Create an empty file called "my_file.txt".



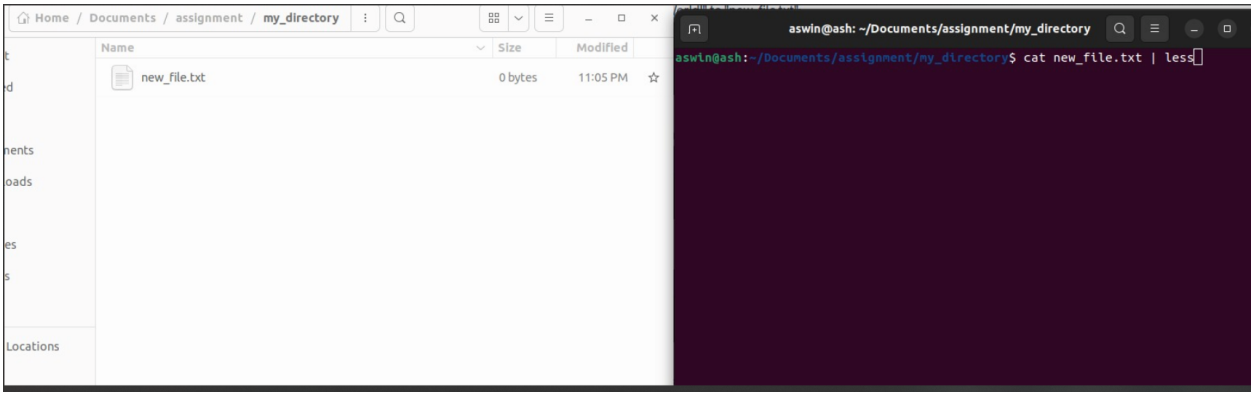
4. List all the files and directories in the current directory.

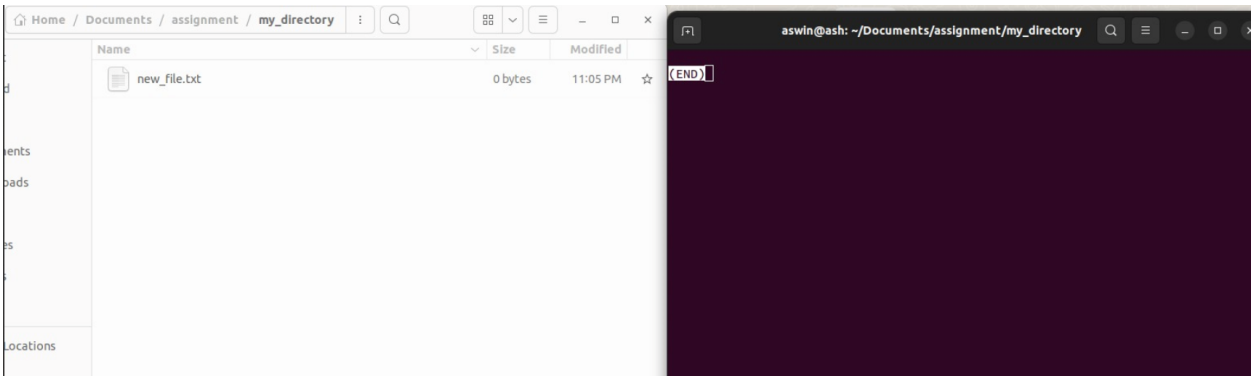


5. Rename "my_file.txt" to "new_file.txt".

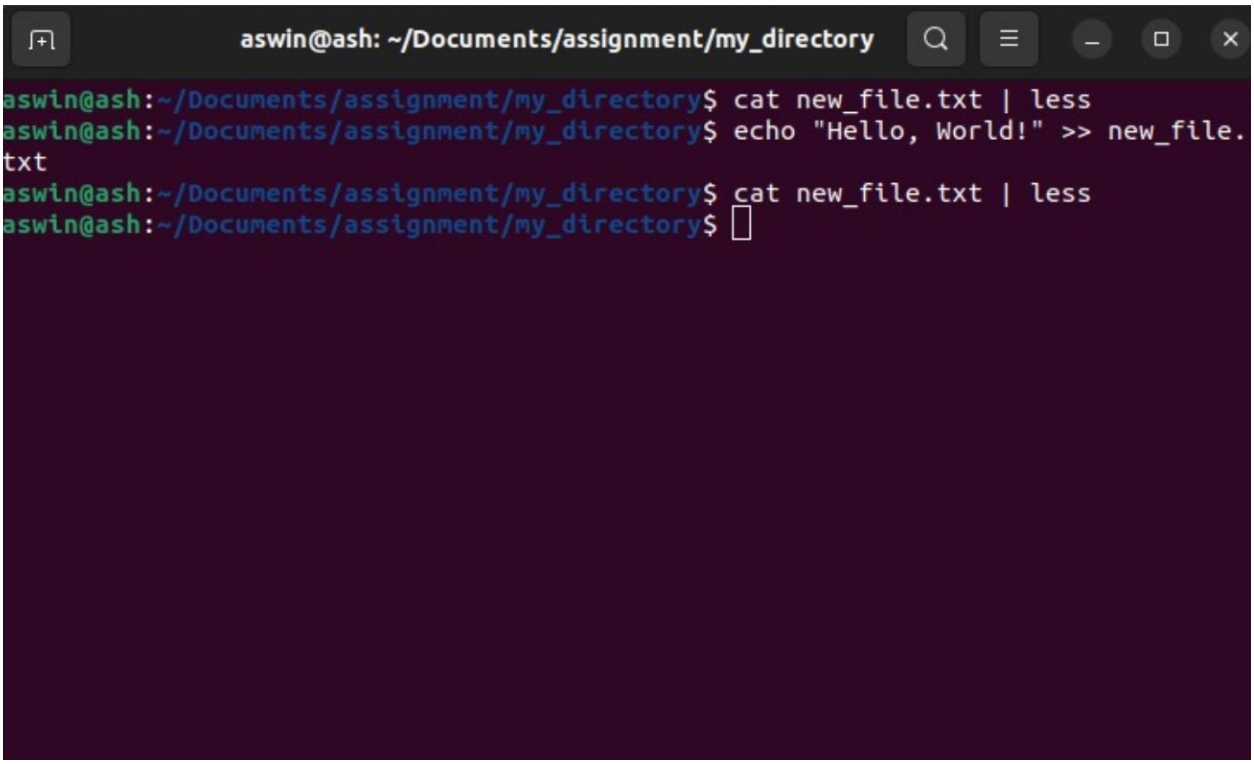


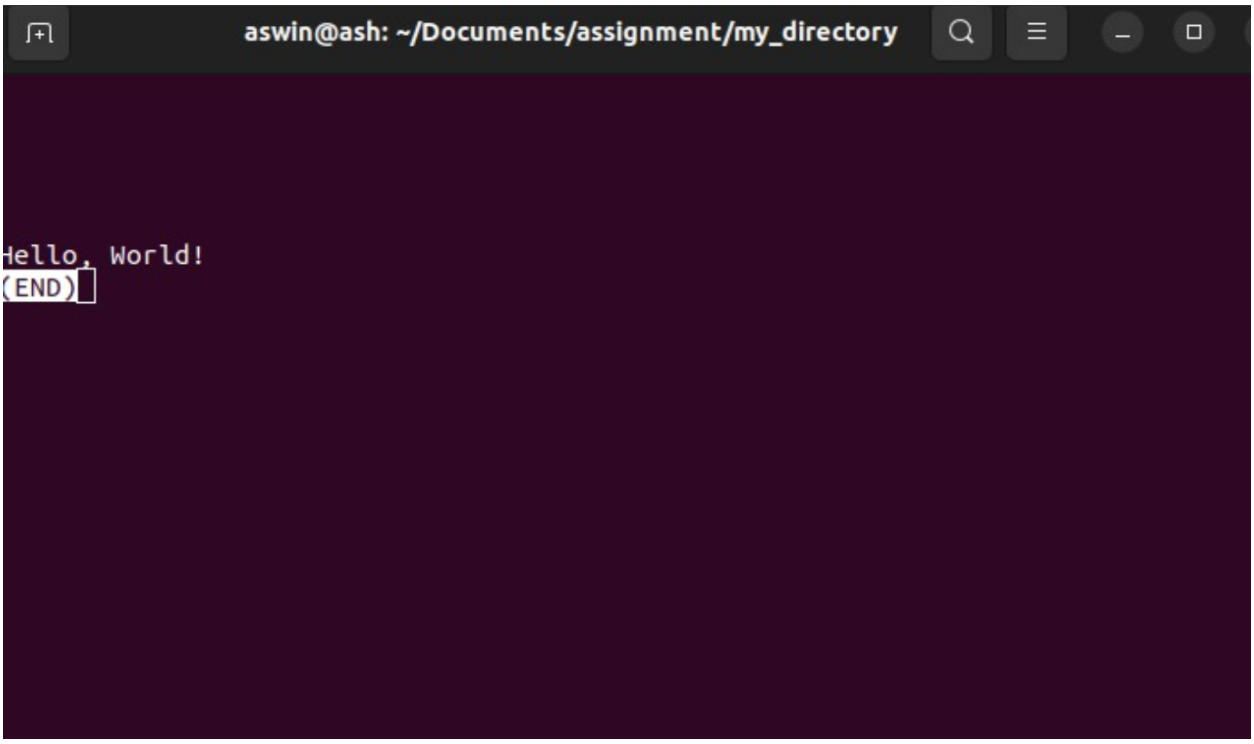
6. Display the content of "new_file.txt" using a pager tool of your choice.



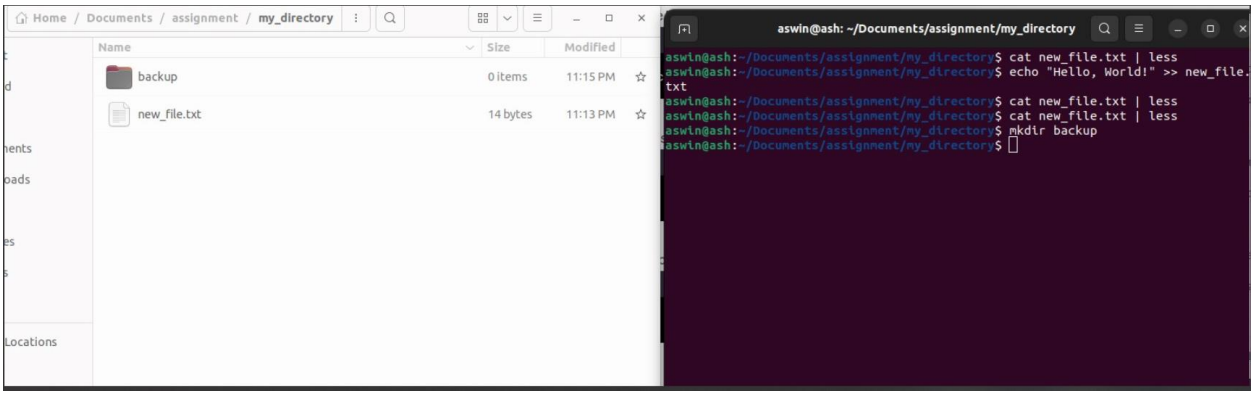


7. Append the text "Hello, World!" to "new_file.txt".

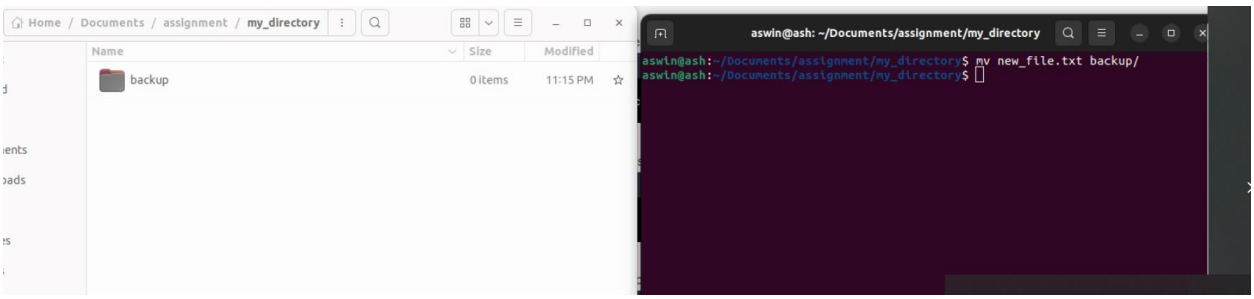




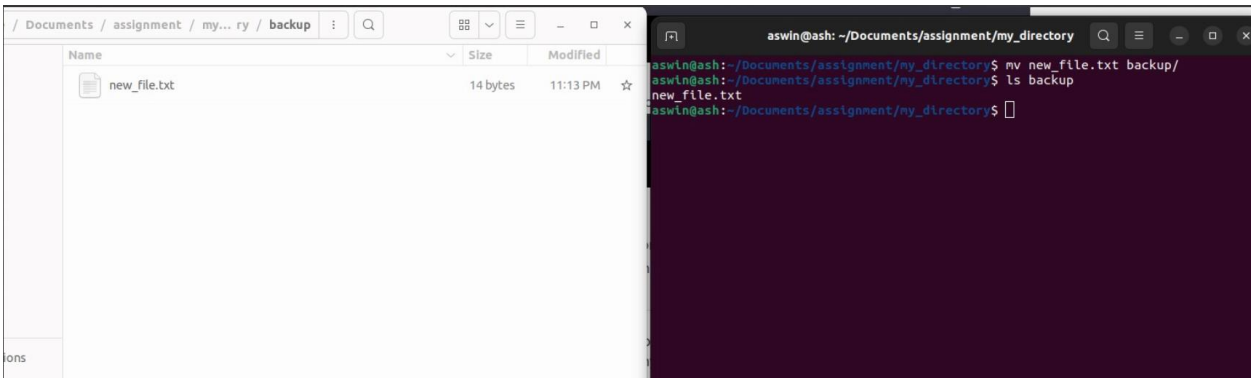
8. Create a new directory called "backup" within "my_directory".



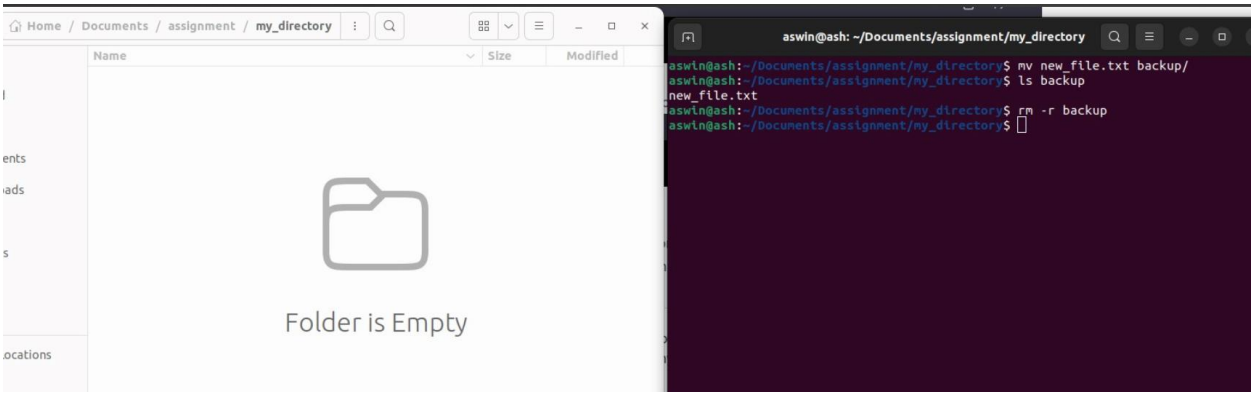
9. Move "new_file.txt" to the "backup" directory.



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

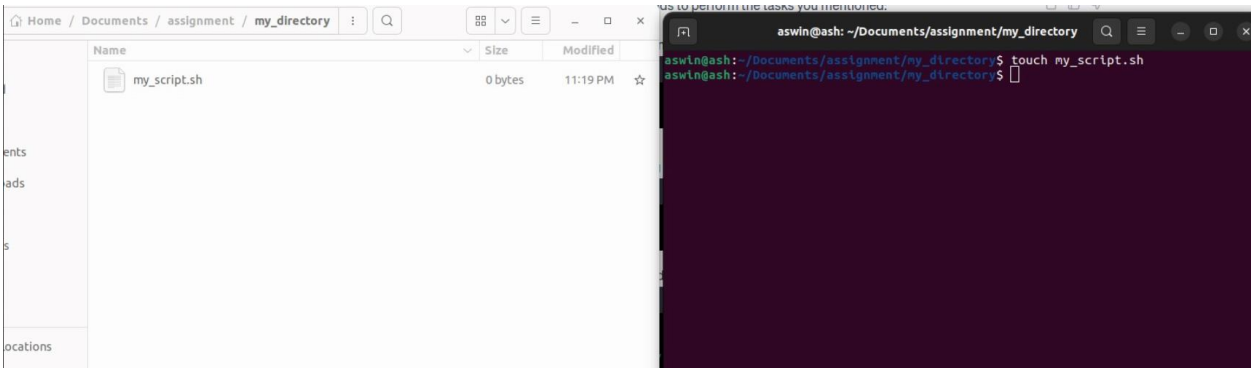


11. Delete the "backup" directory and all its contents.



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:

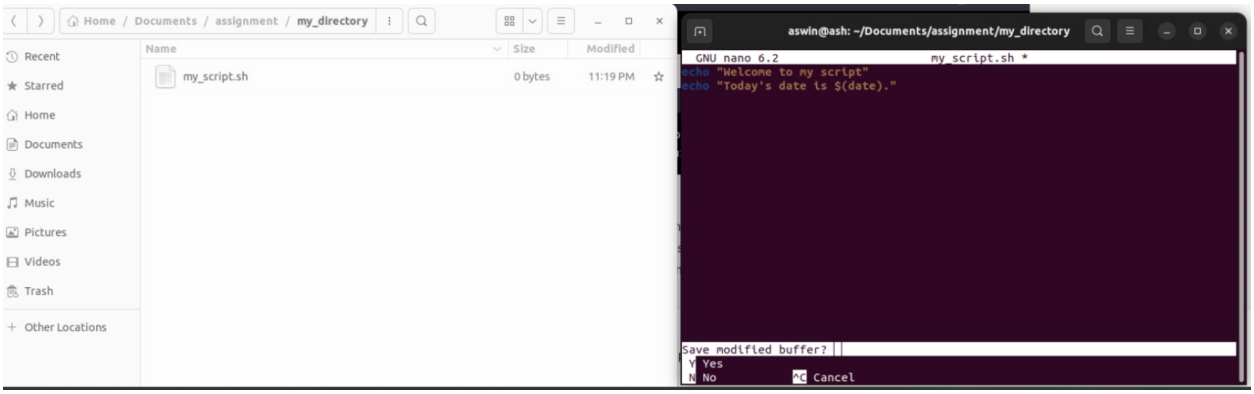
bash

#!/bin/bash

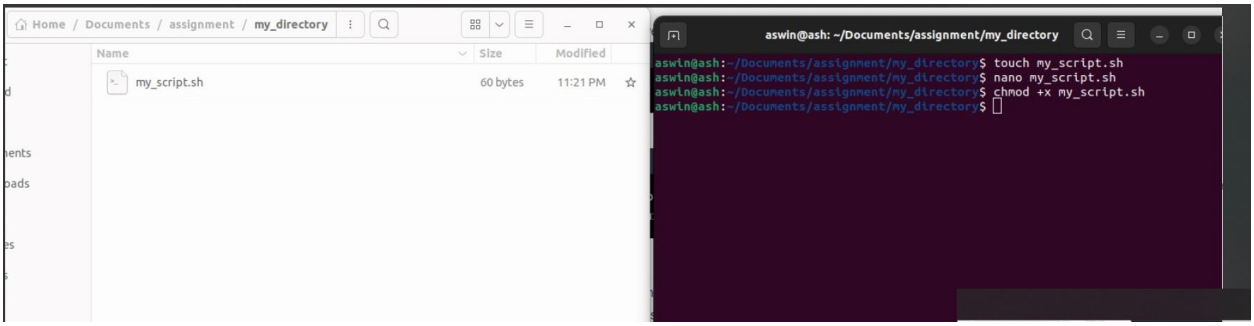
echo "Welcome to my script!"

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

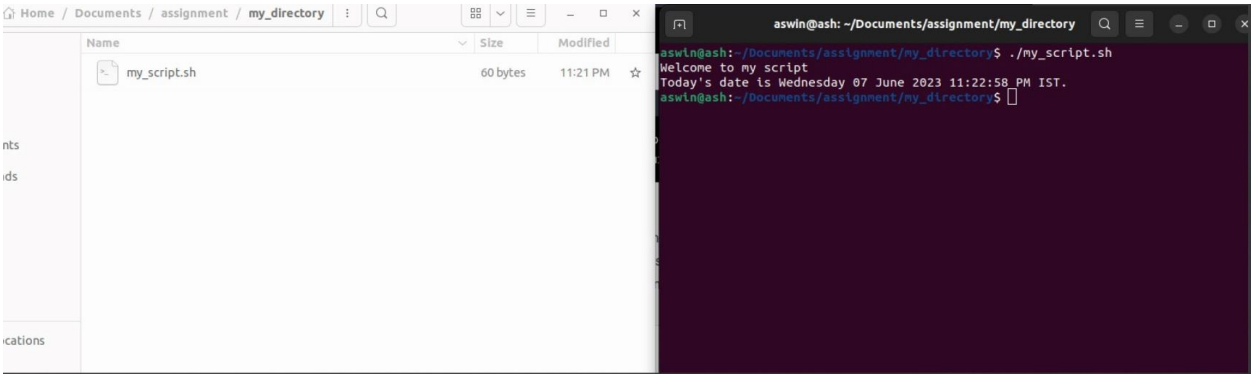
Save and exit the file.



- Make "my_script.sh" executable.

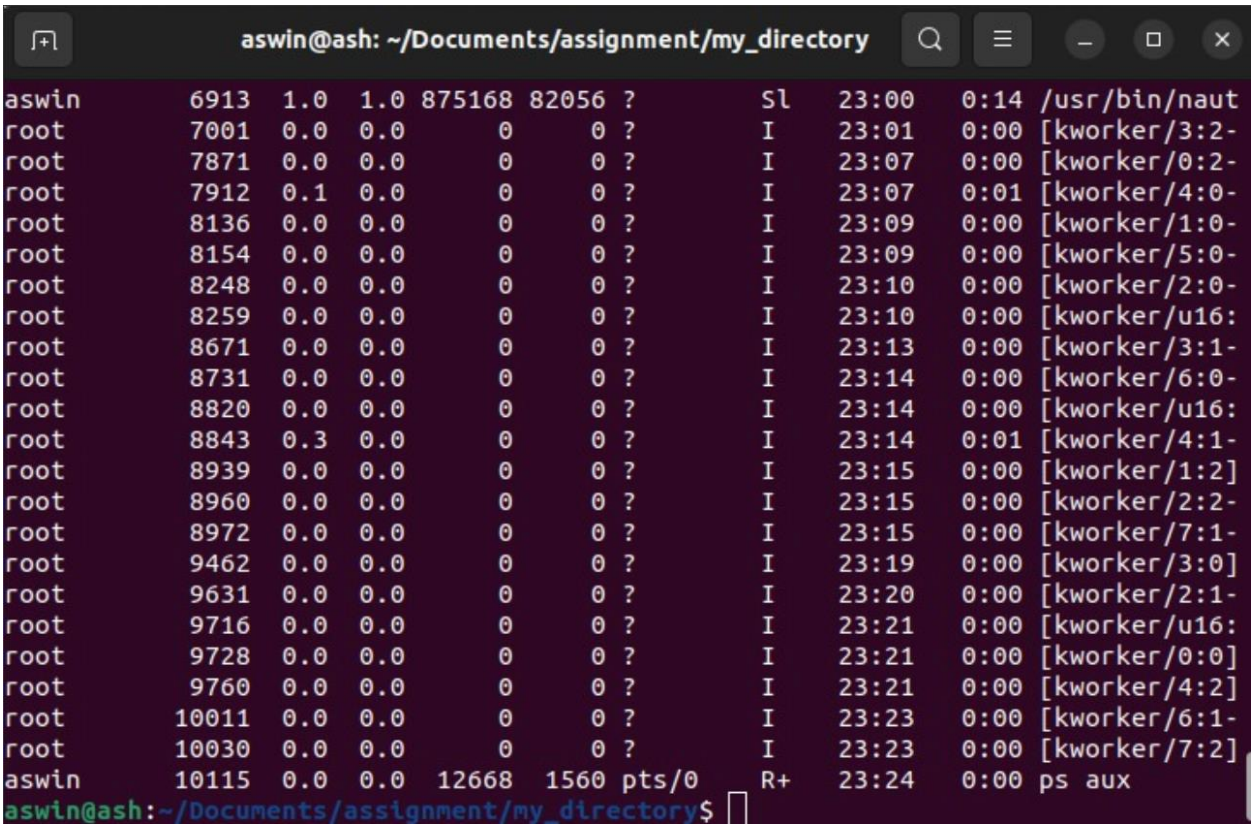
- 

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



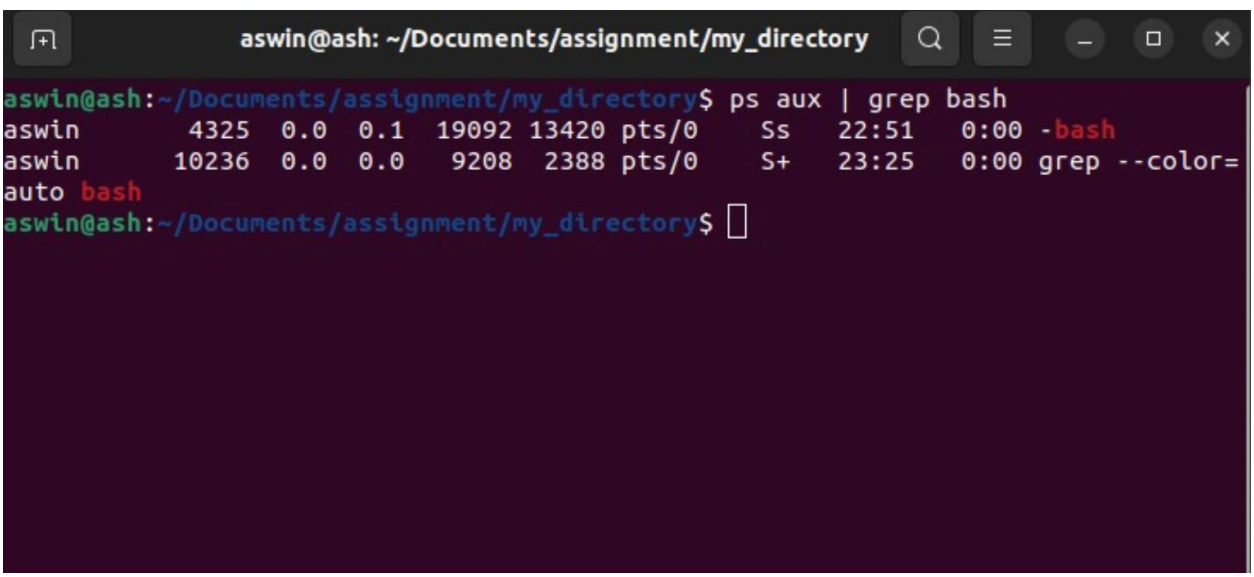
Task 3: Command Execution and Pipelines

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



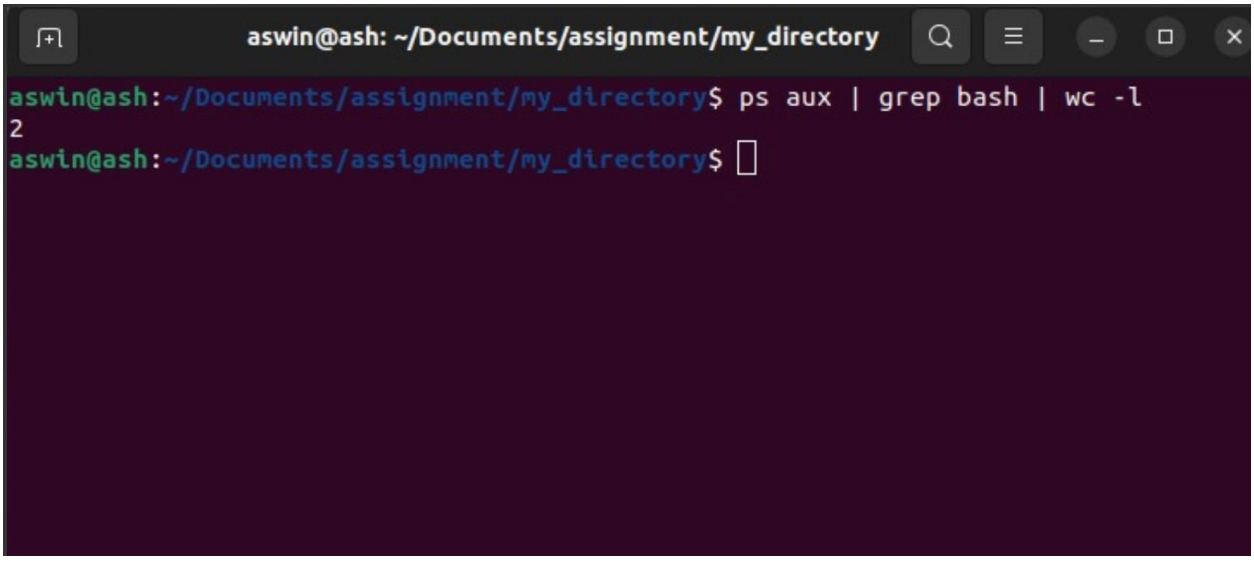
```
aswin@ash: ~/Documents/assignment/my_directory
aswin       6913  1.0  1.0 875168 82056 ?        Sl   23:00   0:14 /usr/bin/naut
root        7001  0.0  0.0      0     0 ?        I    23:01   0:00 [kworker/3:2-
root        7871  0.0  0.0      0     0 ?        I    23:07   0:00 [kworker/0:2-
root        7912  0.1  0.0      0     0 ?        I    23:07   0:01 [kworker/4:0-
root        8136  0.0  0.0      0     0 ?        I    23:09   0:00 [kworker/1:0-
root        8154  0.0  0.0      0     0 ?        I    23:09   0:00 [kworker/5:0-
root        8248  0.0  0.0      0     0 ?        I    23:10   0:00 [kworker/2:0-
root        8259  0.0  0.0      0     0 ?        I    23:10   0:00 [kworker/u16:
root        8671  0.0  0.0      0     0 ?        I    23:13   0:00 [kworker/3:1-
root        8731  0.0  0.0      0     0 ?        I    23:14   0:00 [kworker/6:0-
root        8820  0.0  0.0      0     0 ?        I    23:14   0:00 [kworker/u16:
root        8843  0.3  0.0      0     0 ?        I    23:14   0:01 [kworker/4:1-
root        8939  0.0  0.0      0     0 ?        I    23:15   0:00 [kworker/1:2]
root        8960  0.0  0.0      0     0 ?        I    23:15   0:00 [kworker/2:2-
root        8972  0.0  0.0      0     0 ?        I    23:15   0:00 [kworker/7:1-
root        9462  0.0  0.0      0     0 ?        I    23:19   0:00 [kworker/3:0]
root        9631  0.0  0.0      0     0 ?        I    23:20   0:00 [kworker/2:1-
root        9716  0.0  0.0      0     0 ?        I    23:21   0:00 [kworker/u16:
root        9728  0.0  0.0      0     0 ?        I    23:21   0:00 [kworker/0:0]
root        9760  0.0  0.0      0     0 ?        I    23:21   0:00 [kworker/4:2]
root       10011  0.0  0.0      0     0 ?        I    23:23   0:00 [kworker/6:1-
root       10030  0.0  0.0      0     0 ?        I    23:23   0:00 [kworker/7:2]
aswin      10115  0.0  0.0  12668  1560 pts/0    R+   23:24   0:00 ps aux
aswin@ash:~/Documents/assignment/my_directory$
```

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



```
aswin@ash: ~/Documents/assignment/my_directory
aswin@ash:~/Documents/assignment/my_directory$ ps aux | grep bash
aswin       4325  0.0  0.1  19092 13420 pts/0    Ss   22:51   0:00 -bash
aswin      10236  0.0  0.0   9208  2388 pts/0    S+   23:25   0:00 grep --color=
auto bash
aswin@ash:~/Documents/assignment/my_directory$
```

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



Explanation of the commands:

The ps aux command lists all the running processes on the system.
The | (pipe) symbol takes the output of the previous command and passes it as input to the next command.
The grep command is used to search for a specific pattern or text in the input.
In this case, grep bash filters the processes list and displays only the lines that contain the word "bash".
Finally, the wc -l command counts the number of lines in the filtered output and displays the result.
By combining these commands with the pipe operator, you can filter and process the output of one command using another command.