**Name: Sainyam Agrawal**

**Reg No: 20BCY10151**

**Assignment: Bash Shell Basics**

**Task 1: File and Directory Manipulation**

1. Create a directory called "my\_directory".

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated

1. Navigate into the "my\_directory".

A screenshot of a computer

Description automatically generated with medium confidence

1. Create an empty file called "my\_file.txt".

A screenshot of a computer

Description automatically generated with medium confidenceA picture containing text, screenshot, design

Description automatically generated

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

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot

Description automatically generated

1. Rename "my\_file.txt" to "new\_file.txt".

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with low confidence

1. Display the content of "new\_file.txt" using a pager tool of your choice.

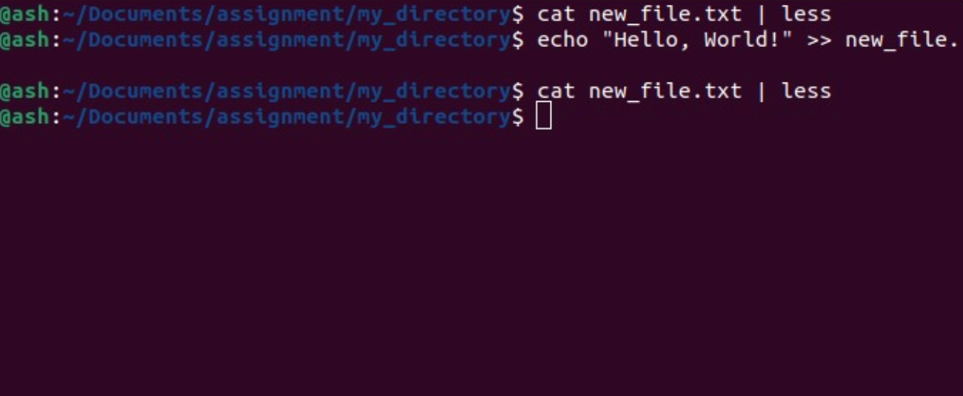
A picture containing screenshot, purple, lilac, violet

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Append the text "Hello, World!" to "new\_file.txt".



A picture containing screenshot, text, purple, lilac

Description automatically generated

1. Create a new directory called "backup" within "my\_directory".

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated with medium confidence

1. Move "new\_file.txt" to the "backup" directory. A screenshot of a computer

   Description automatically generated with medium confidenceA picture containing text, screenshot, purple, violet

   Description automatically generated
2. Verify that "new\_file.txt" is now located in the "backup" directory.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with low confidence

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

A picture containing text, screenshot, font

Description automatically generated

**Task 2: Permissions and Scripting**

* Create a new file called "my\_script.sh".

A screenshot of a computer

Description automatically generated with medium confidenceA picture containing screenshot, text, purple, lilac

Description automatically generated

* 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.**

A screenshot of a computer

Description automatically generated with medium confidence

* Make "my\_script.sh" executable.

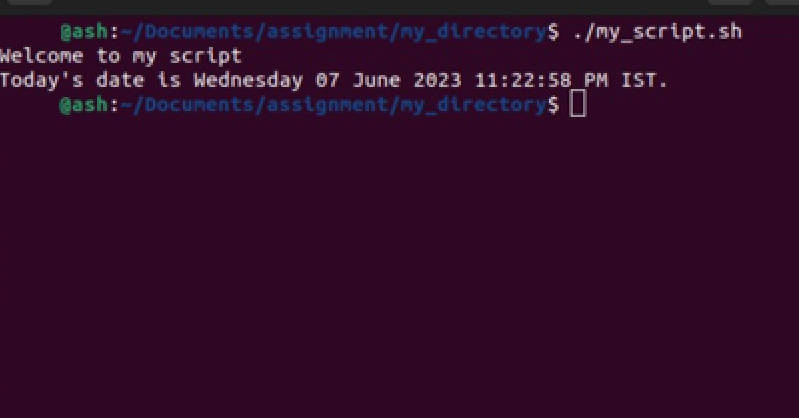
A screenshot of a computer

Description automatically generated with medium confidence

* A picture containing text, screenshot, violet

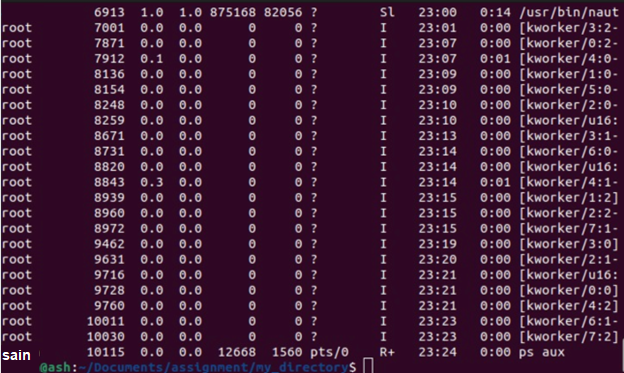
  Description automatically generated
* Run "my\_script.sh" and verify that the output matches the expected result.

A screenshot of a computer

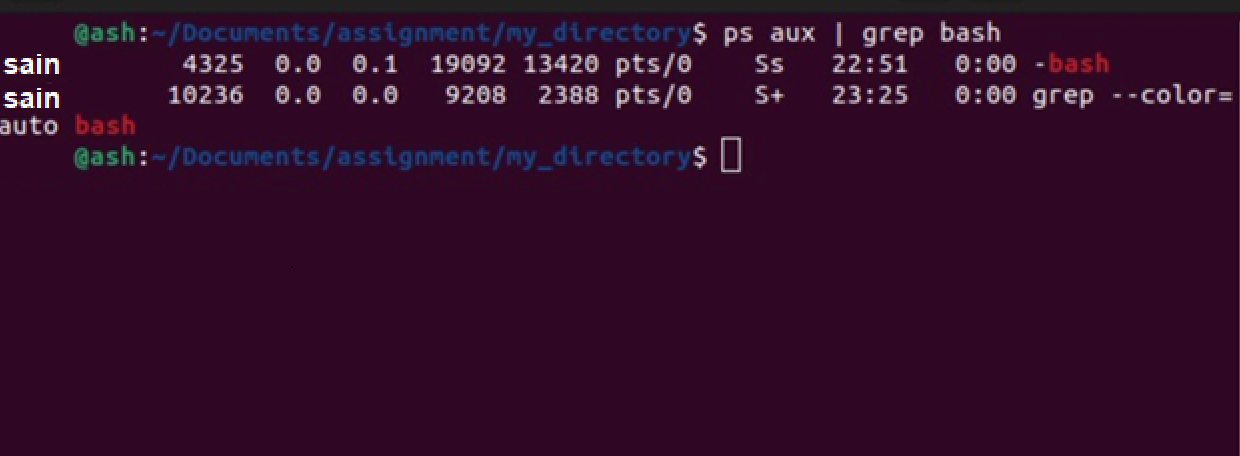
Description automatically generated with medium confidence 

**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.

A screenshot of a computer

Description automatically generated

**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.