**Assignment: Bash Shell Basics**

***NAME: E VARUN SAI***

***REG.NO: 20BCD7179***

***CAMPUS: VIT-AP***

**Task 1: File and Directory Manipulation**

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

A screenshot of a computer

Description automatically generated with medium confidence

This command creates a new directory named "my\_directory" in the current working directory.

1. Navigate into the "my\_directory".

A picture containing text, screenshot, font, black

Description automatically generated

This command changes the current working directory to "my\_directory".

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

A picture containing text, font, screenshot

Description automatically generated  
The **touch** command is used to create an empty file. In this case, it creates a file named "my\_file.txt" in the current directory.

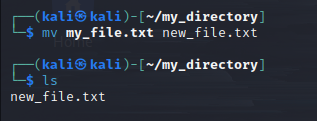
1. List all the files and directories in the current directory.

A picture containing text, screenshot, font

Description automatically generated

The **ls** command lists the files and directories in the current directory.

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



The **mv** command is used to move or rename files. In this case, it renames the file "my\_file.txt" to "new\_file.txt"

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

A screen shot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

The **less** command is a pager tool that allows you to view the content of a file page by page. In this case, it displays the content of the file "new\_file.txt". You can scroll through the content using the arrow keys and press "q" to exit.

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

A screen shot of a computer

Description automatically generated with medium confidence

The **echo** command is used to print text. The **>>** operator is used to append the output to a file. In this case, it appends the text "Hello, World!" to the file "new\_file.txt"

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

A black background with white text

Description automatically generated with low confidence

This command creates a new directory named "backup" within the "my\_directory" directory.

1. A black background with white text

   Description automatically generated with low confidenceMove "new\_file.txt" to the "backup" directory.

This command moves the file "new\_file.txt" to the "backup" directory.

1. Verify that "new\_file.txt" is now located in the "backup" directory.

A screen shot of a computer

Description automatically generated with medium confidence

This command lists the contents of the "backup" directory to verify that "new\_file.txt" is present there.

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

A picture containing text, screenshot, font

Description automatically generated

The **rm** command is used to remove files and directories. The **-r** option is used to recursively remove directories and their contents. In this case, it deletes the "backup" directory and all its contents

**Task 2: Permissions and Scripting**

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

A picture containing text, font, screenshot

Description automatically generated

This command creates a new file named "my\_script.sh" in the current directory.

* Edit "my\_script.sh" using a text editor of your choice and add the following lines

A white text on a black background

Description automatically generated with low confidence

**Bash**

**#!/bin/bash**

**echo "Welcome to my script!"**

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

**Save and exit the file.**

A screen shot of a computer

Description automatically generated with medium confidence

These lines are added to the "my\_script.sh" file. The first line specifies the interpreter

(**#!/bin/bash**), and the subsequent lines use the **echo** command to print text.

* Make "my\_script.sh" executable.

A picture containing text, screenshot, font

Description automatically generated

The **chmod** command is used to change the permissions of a file. The **+x** option makes the file executable, allowing it to be run as a script.

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

A picture containing text, screenshot, font

Description automatically generated

This command executes the "my\_script.sh" file, and the output should display the text

specified in the script, including the current date and time

**Task 3: Command Execution and Pipelines**

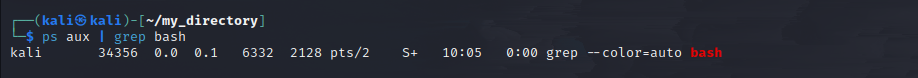
* List all the processes running on your system using the "ps" command.

A screen shot of a computer

Description automatically generated with medium confidence

The **ps** command is used to display information about active processes. The **aux** options provide a detailed list of all processes running on the system.

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



The **grep** command is used to search for specific patterns in the input. In this case, it filters the output of the **ps aux** command to display only the processes that contain the word "bash"

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

A picture containing screenshot, text, font, black

Description automatically generated

The **wc** command is used to count the number of lines, words, and characters in the input. The **-l** option tells **wc** to count only the lines. In this case, it counts the number of lines in the filtered output of the previous command, giving the total number of processes with "bash" in their name.