Name : KALAMEGAM.V **Reg no : 20BIT0302**

EXTERNSHIP PROGRAM – CYBER SECURITY AND ETHICAL HACKING

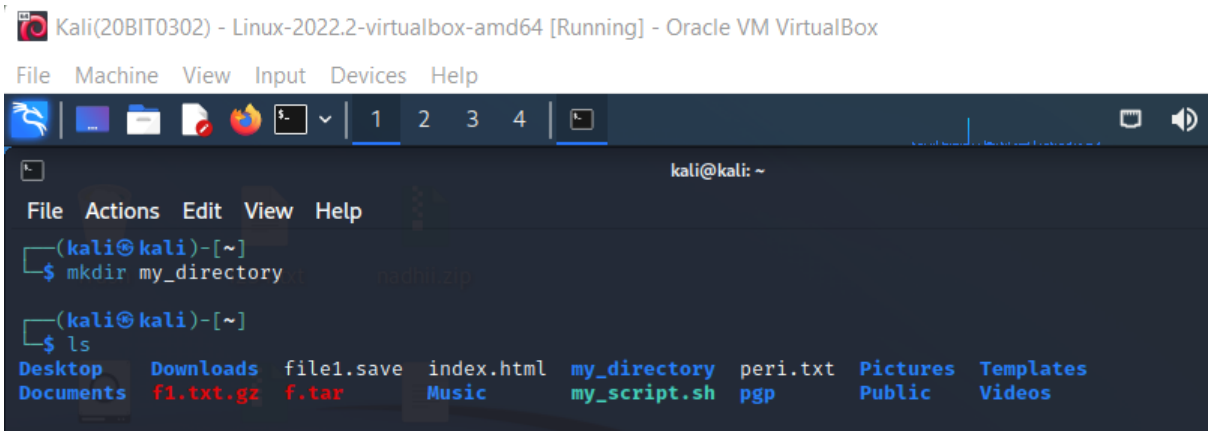
ASSIGNMENT : BASH SHELL BASICS

TASK1 . File and directory manipulation

1. Create a directory called "my_directory".

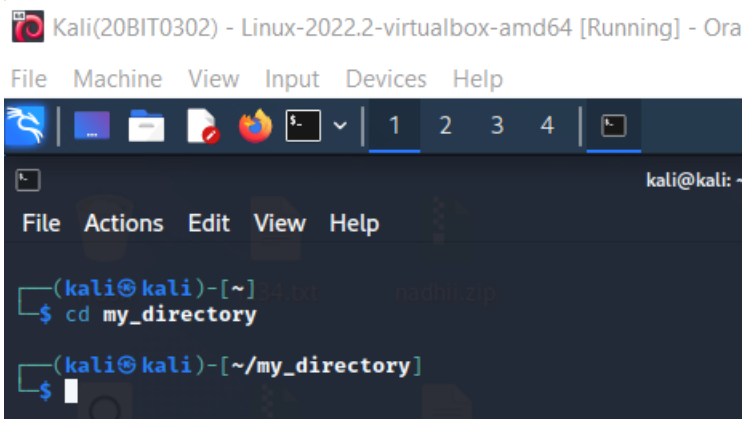
mkdir my_directory

ls



2 . Navigate into the "my_directory".

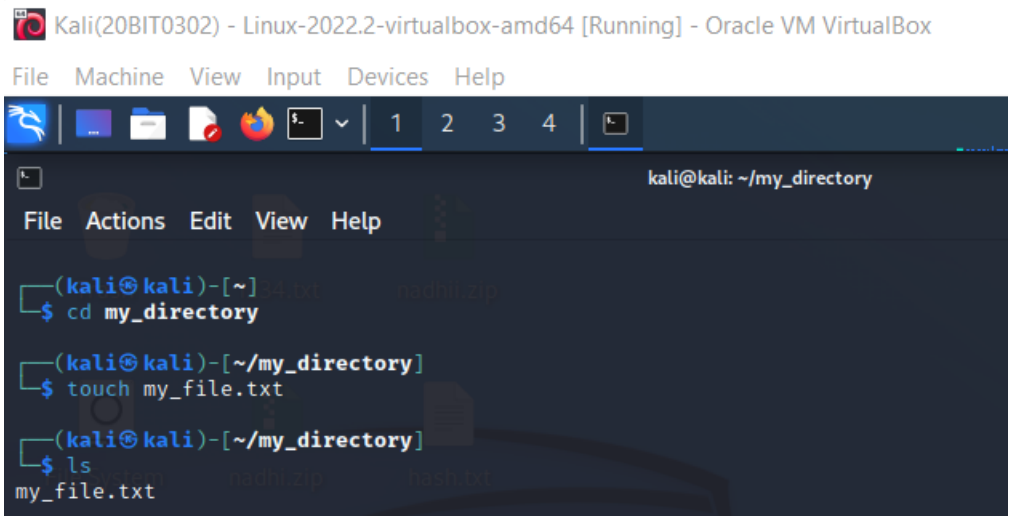
cd my_directory



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

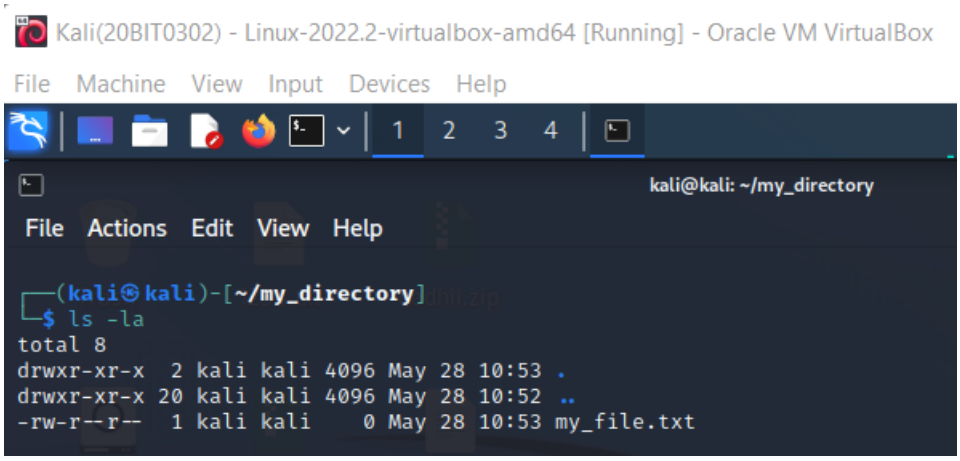
touch my_file.txt

ls



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

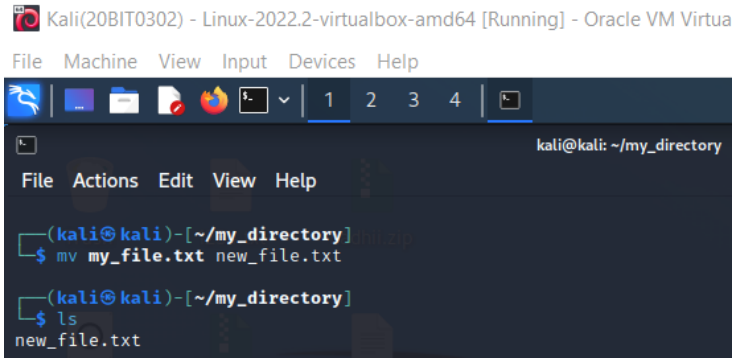
ls -la



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

mv my_file.txt new_file.txt

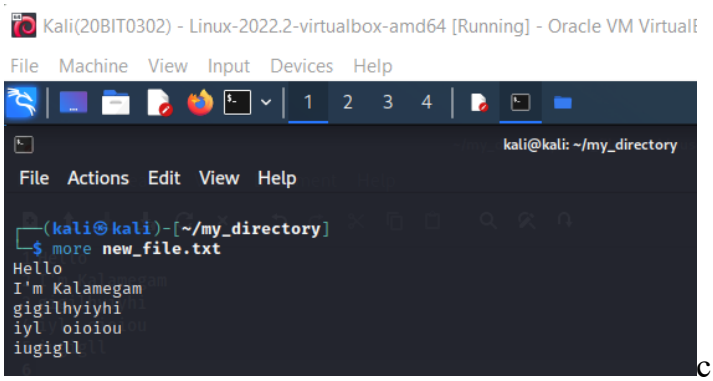
ls



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

more new_file.txt

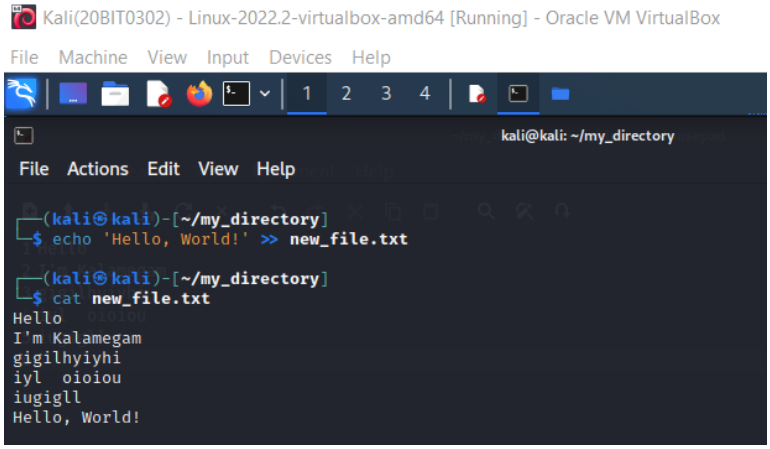
(to display content of new_file.txt , I have added random words to it)



c

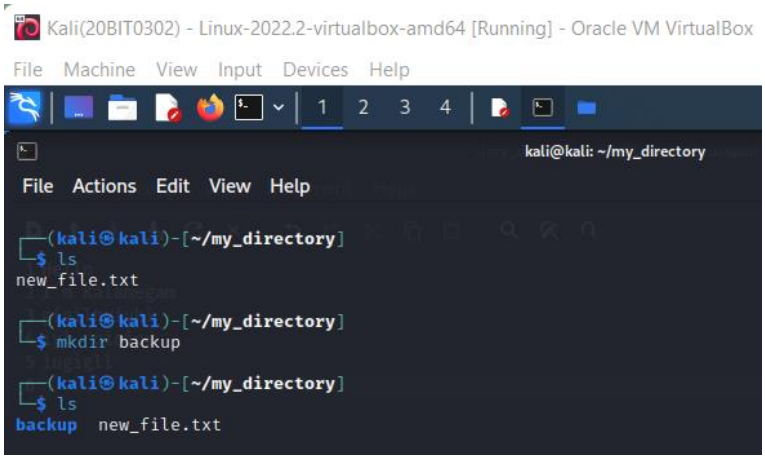
7 . Append the text "Hello, World!" to "new_file.txt".

echo ‘Hello, World!’ >> new_file.txt



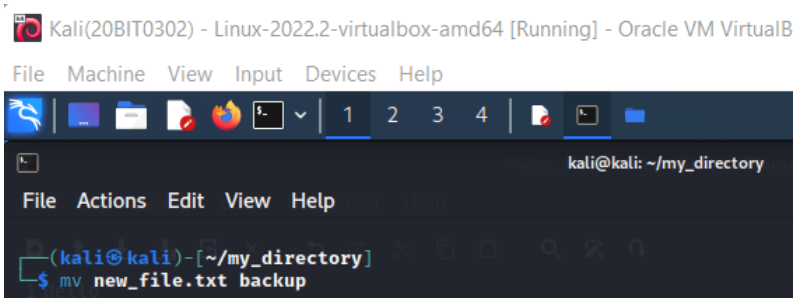
8 . Create a new directory called "backup" within "my_directory". ls

mkdir backup ls



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

mv new_file.txt backup

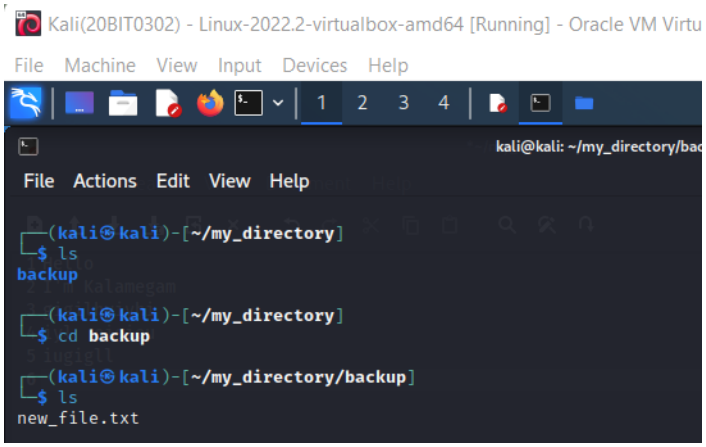


10 . Verify that "new_file.txt" is now located in the "backup" directory.

ls

cd backup

ls

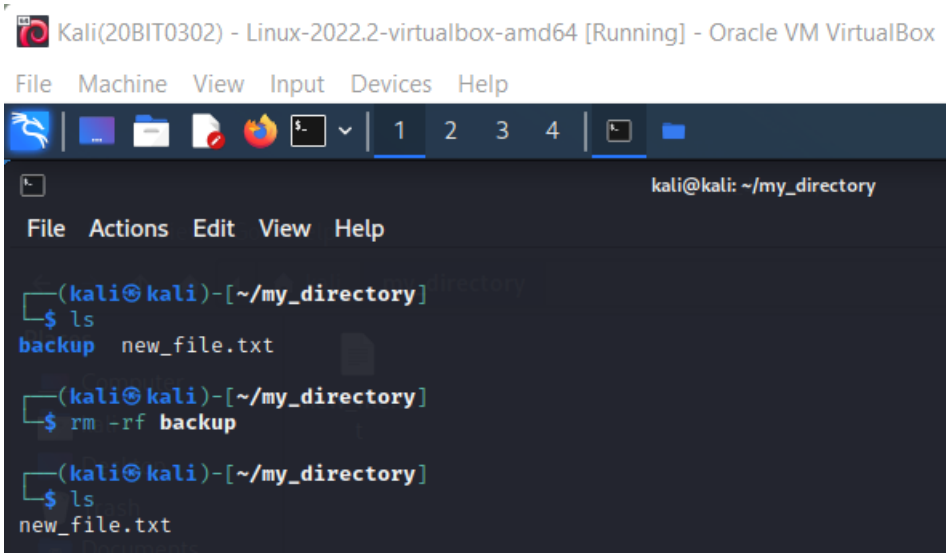


11 . Delete the "backup" directory and all its contents.

ls

rm -rf backup

ls



r : recursive (remove directories and their contents recursively)

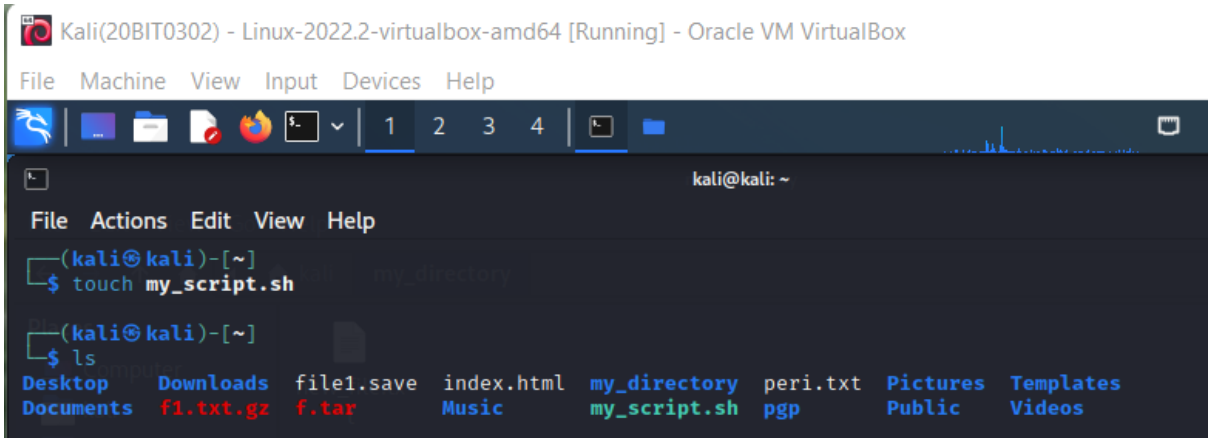
f: force (ignore non-existent file , never prompt)

TASK 2 : PERMISSIONS AND SCRIPTING

1.Create a new file called “my_script.sh”

touch my_script.sh

ls



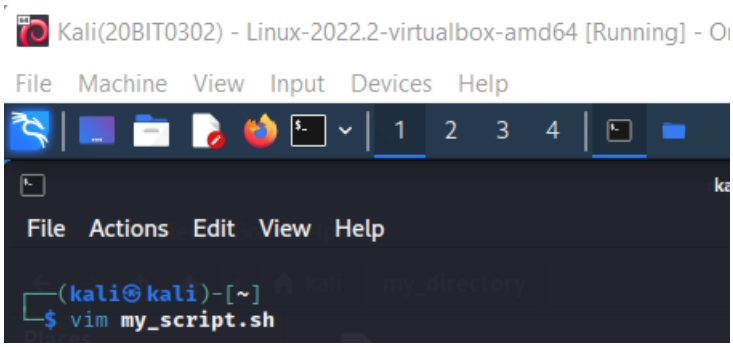
2. Edit my_script.sh using any text editor , add the given lines, make it executable , and run.

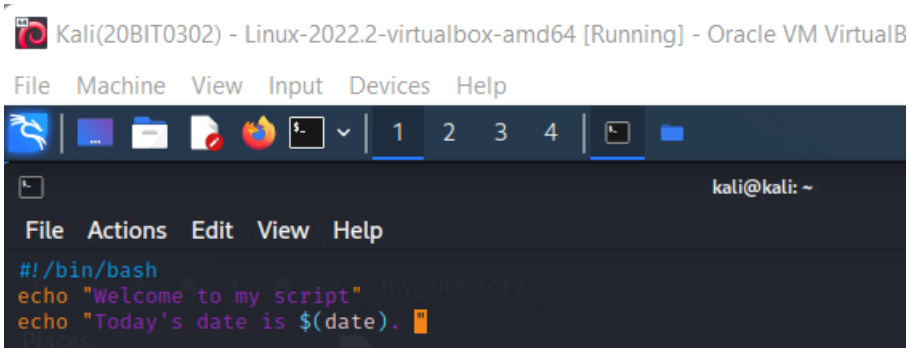
vim my_script.sh

#!/bin/bash

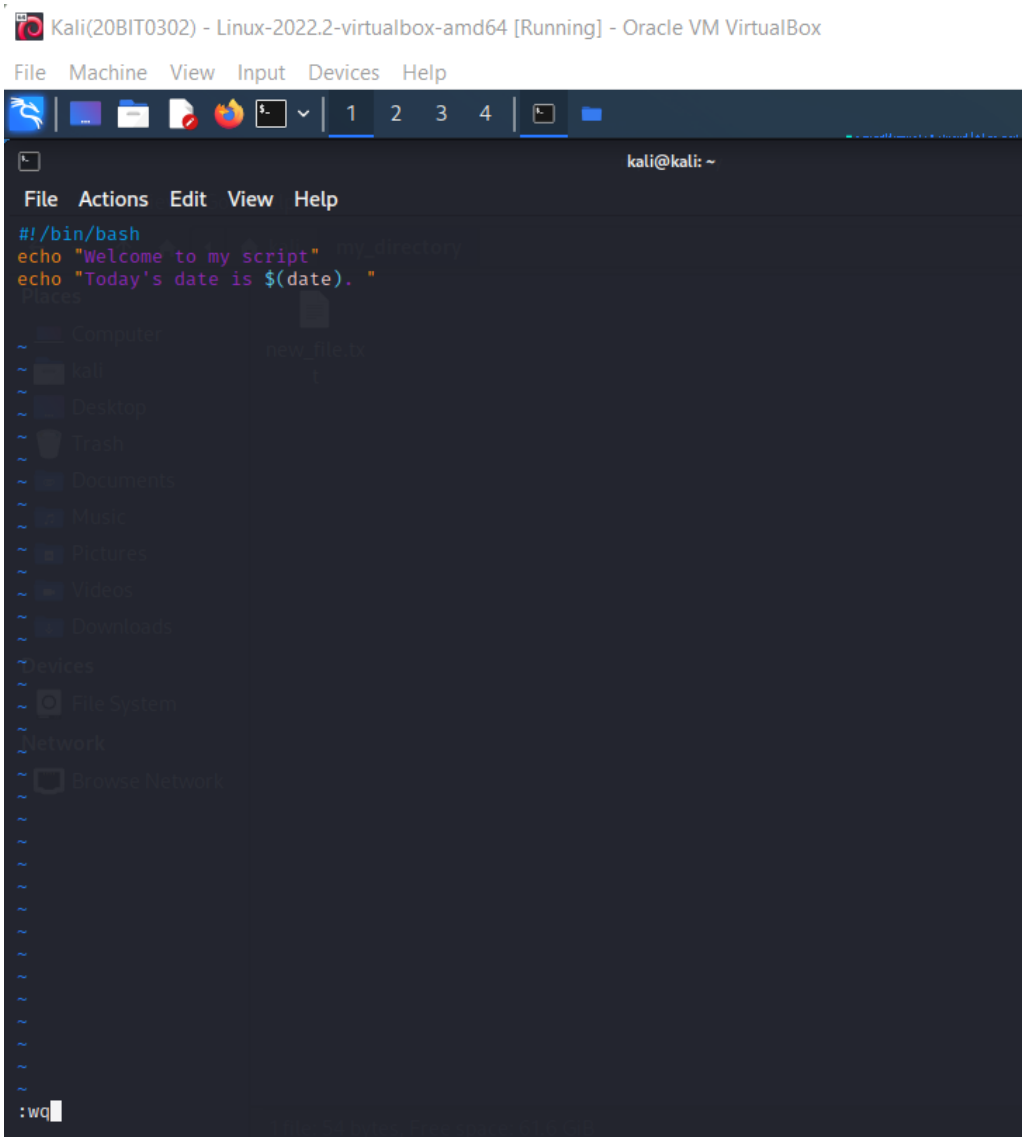
echo “Welcome to my script!”

echo “Today’s date is $(date).”





Name : KALAMEGAM.V **Reg no : 20BIT0302**

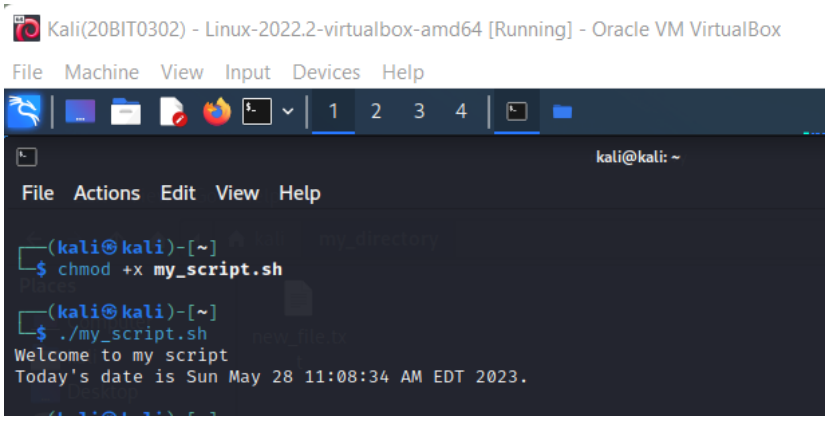


w : save changes made to the file

q : exit Vim

chmod +x my_script.sh

./my_script.sh



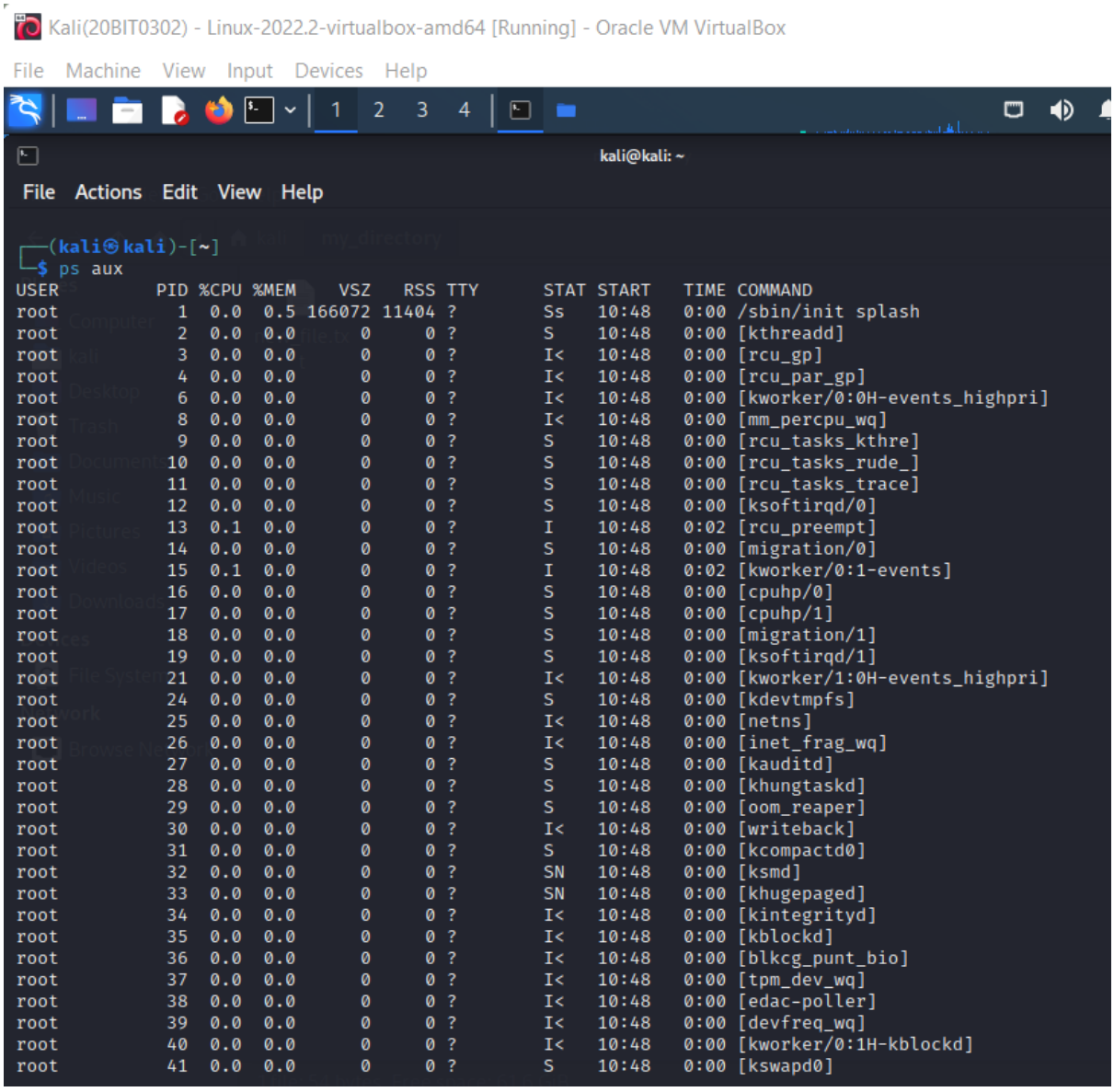
Name : KALAMEGAM.V **Reg no : 20BIT0302**

TASK 3 : COMMAND EXECUTION AND PIPELINES

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

ps aux

The ps aux command is used to display a detailed list of all running processes on a Linux or Unix system.

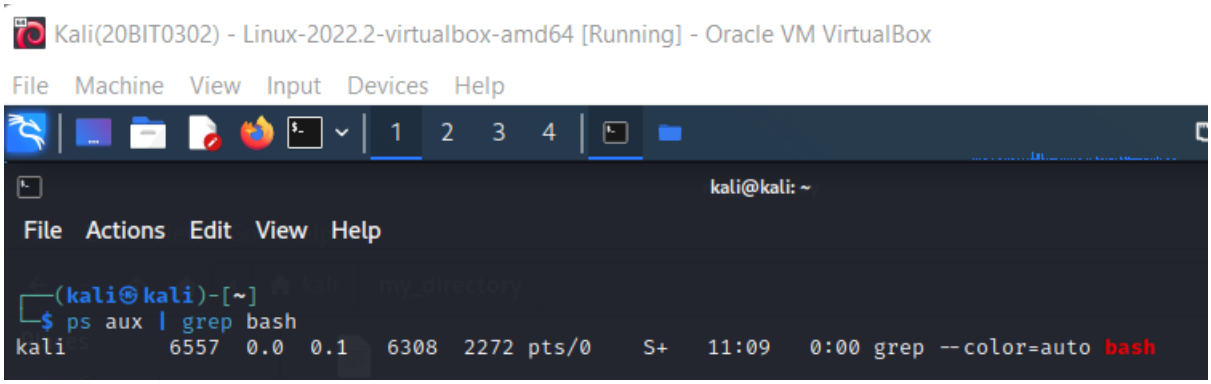


2.Use the "grep" command to filter the processes list and display only the processes with

"bash" in their name.

ps aux | grep bash

(grep is used for matching a pattern or string)



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

ps aux | grep bash | wc -l