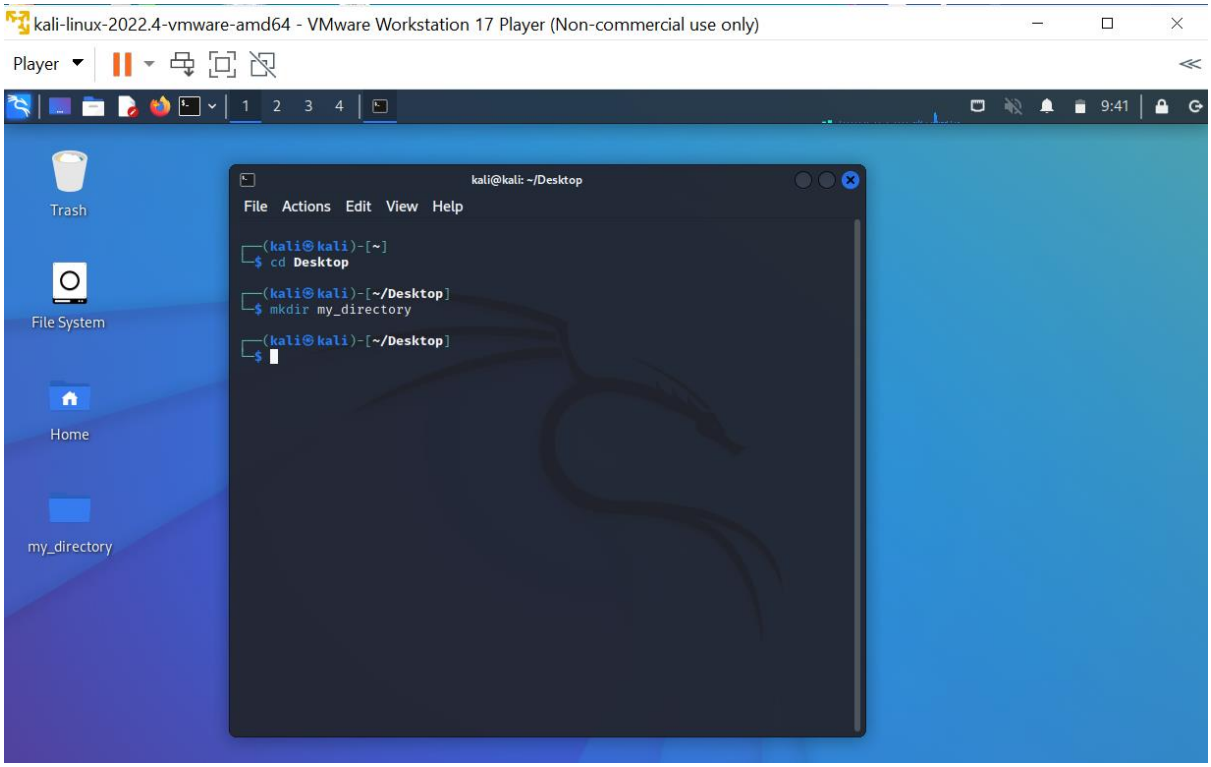
Assignment: Bash Shell Basics

Task 1: File and Directory Manipulation

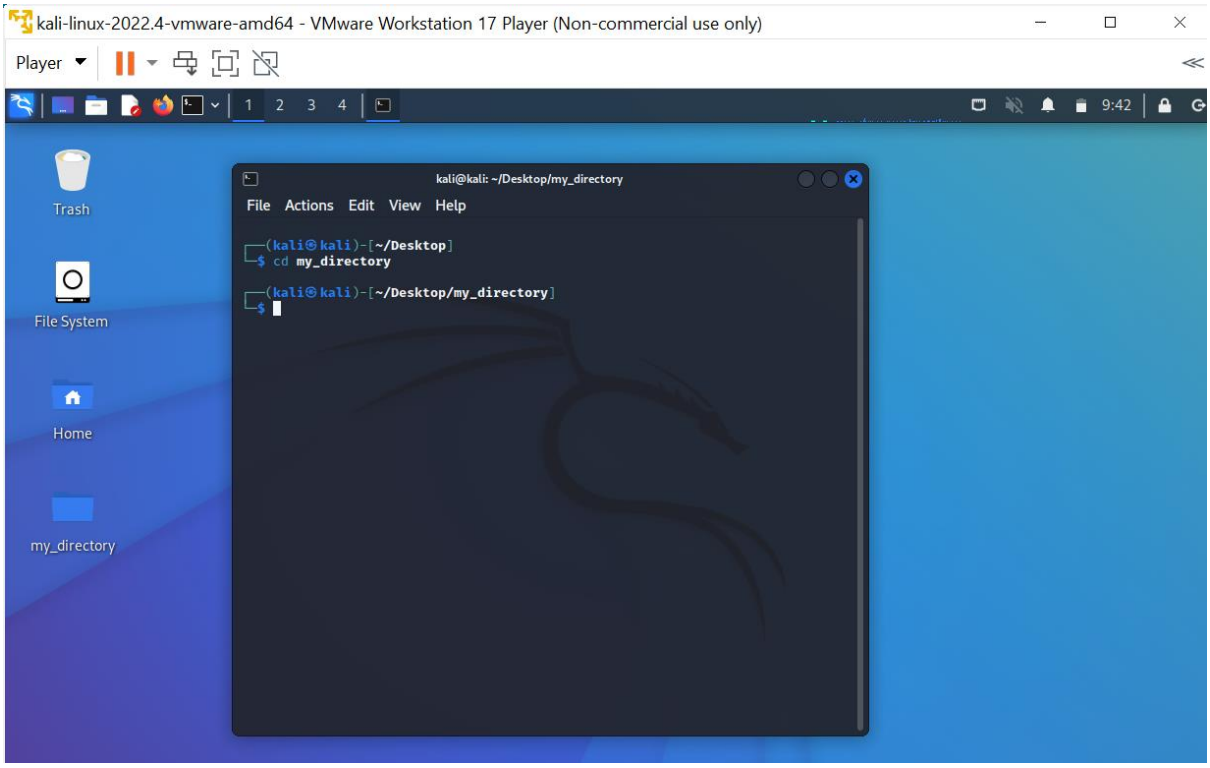
1. Create a directory called "my_directory".

Commands used: mkdir my_directory



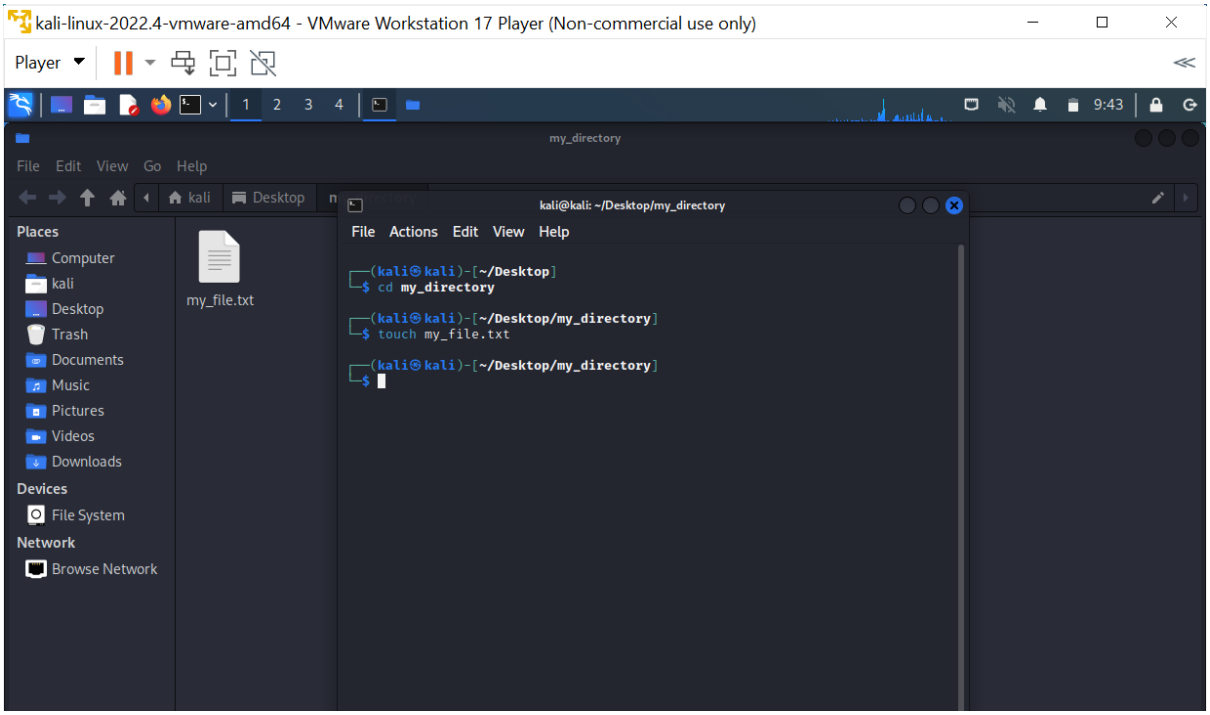
2. Navigate into the "my_directory".

Commands used: cd my_directory



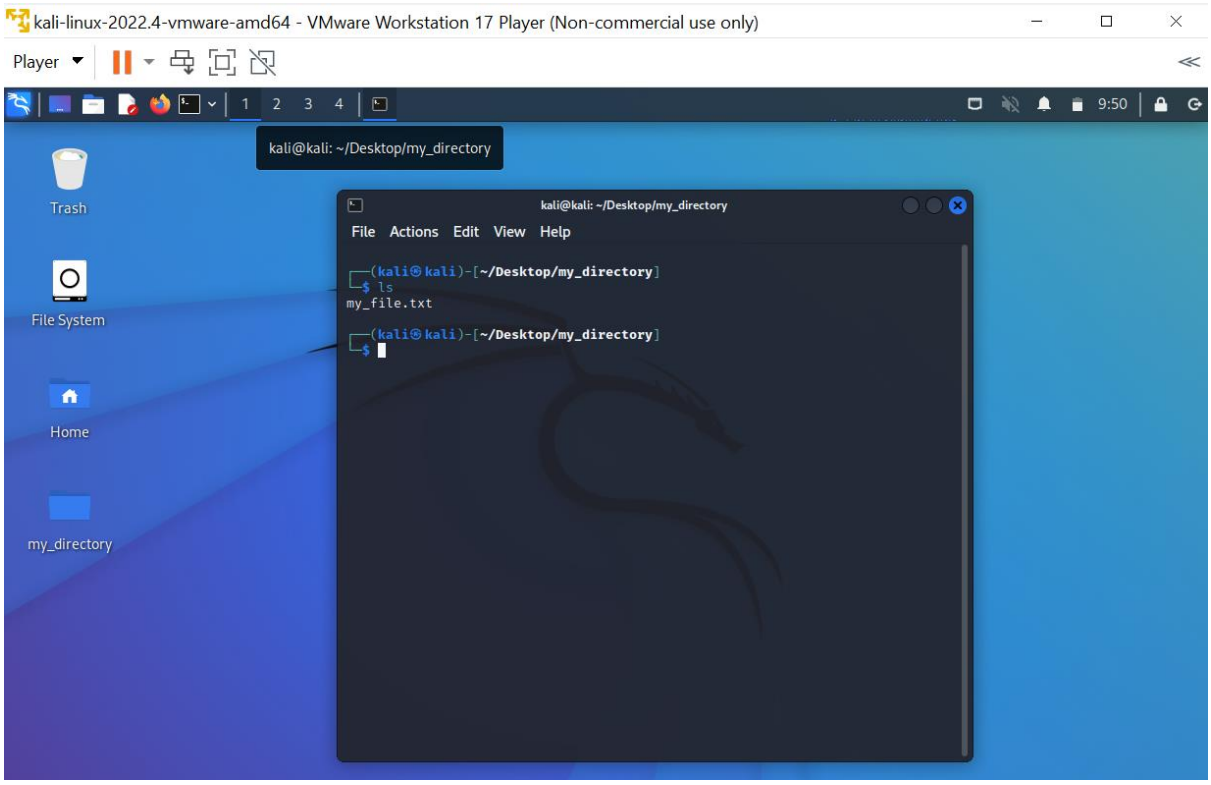
3. Create an empty file called "my_file.txt".

Commands used: touch my_file.txt



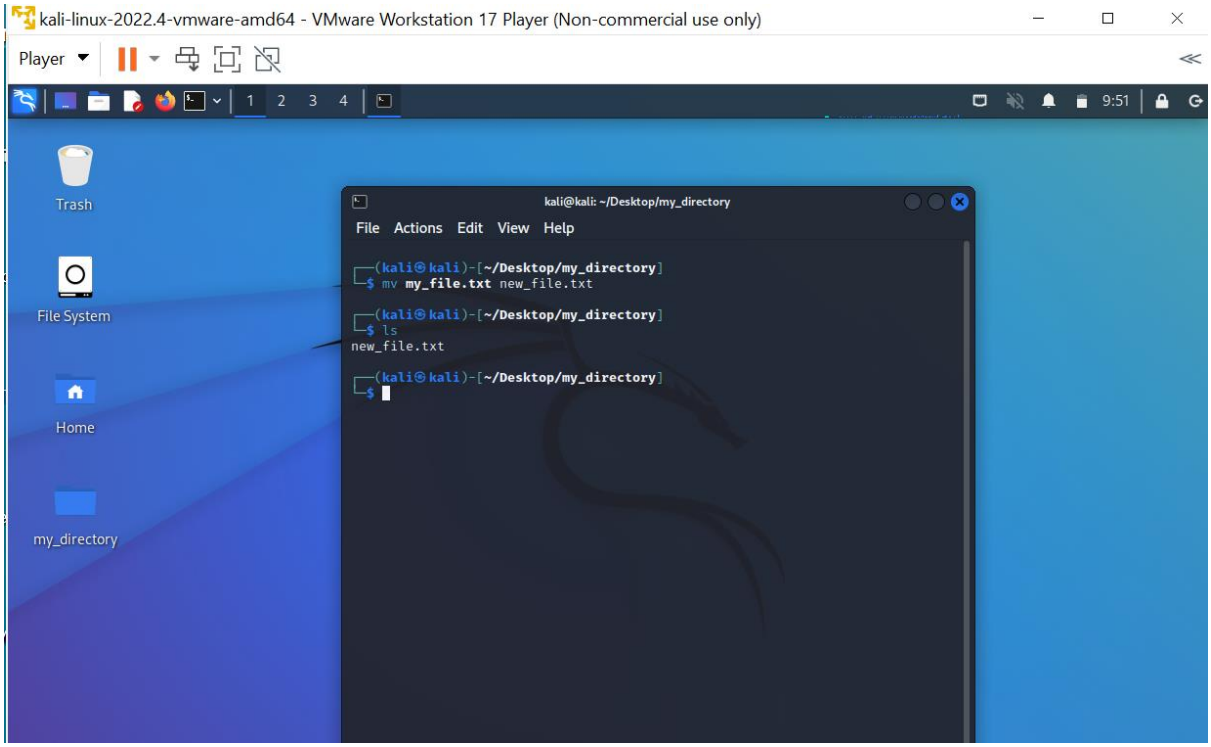
4. List all the files and directories in the current directory.

Commands used: ls



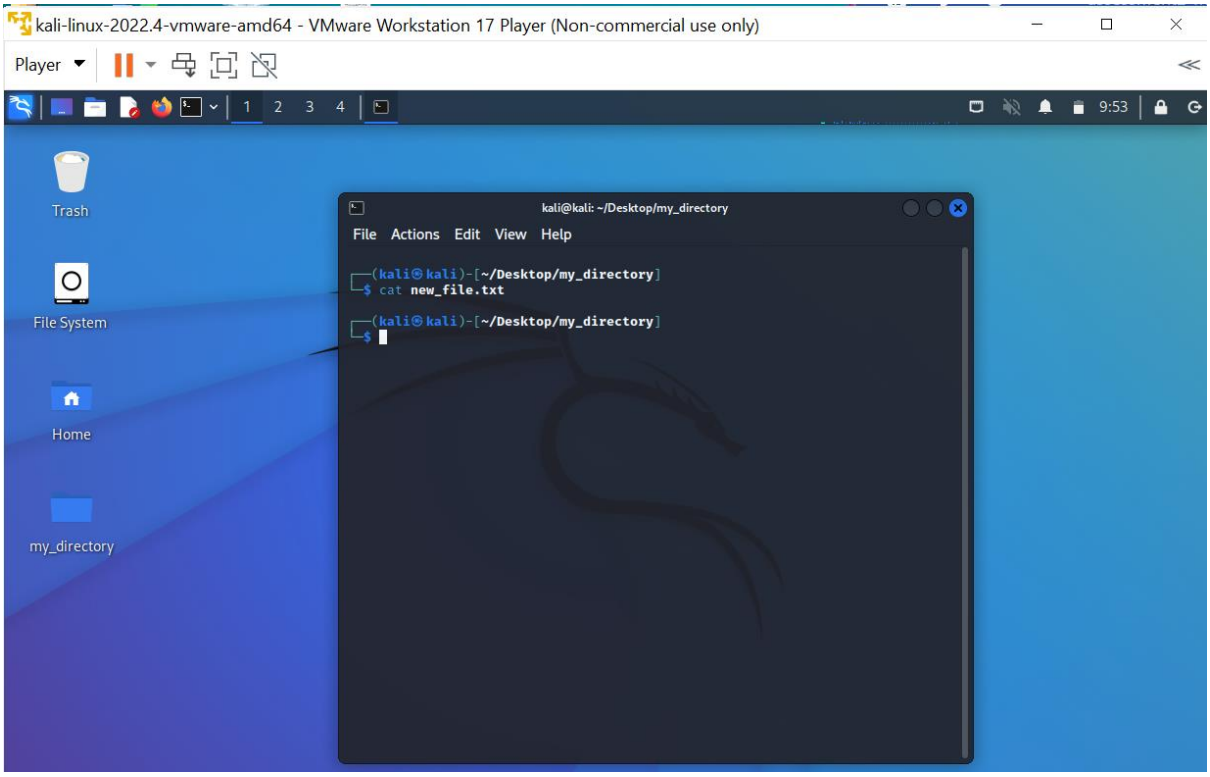
5. Rename "my_file.txt" to "new_file.txt".

Commands used: mv my_file.txt new_file.txt



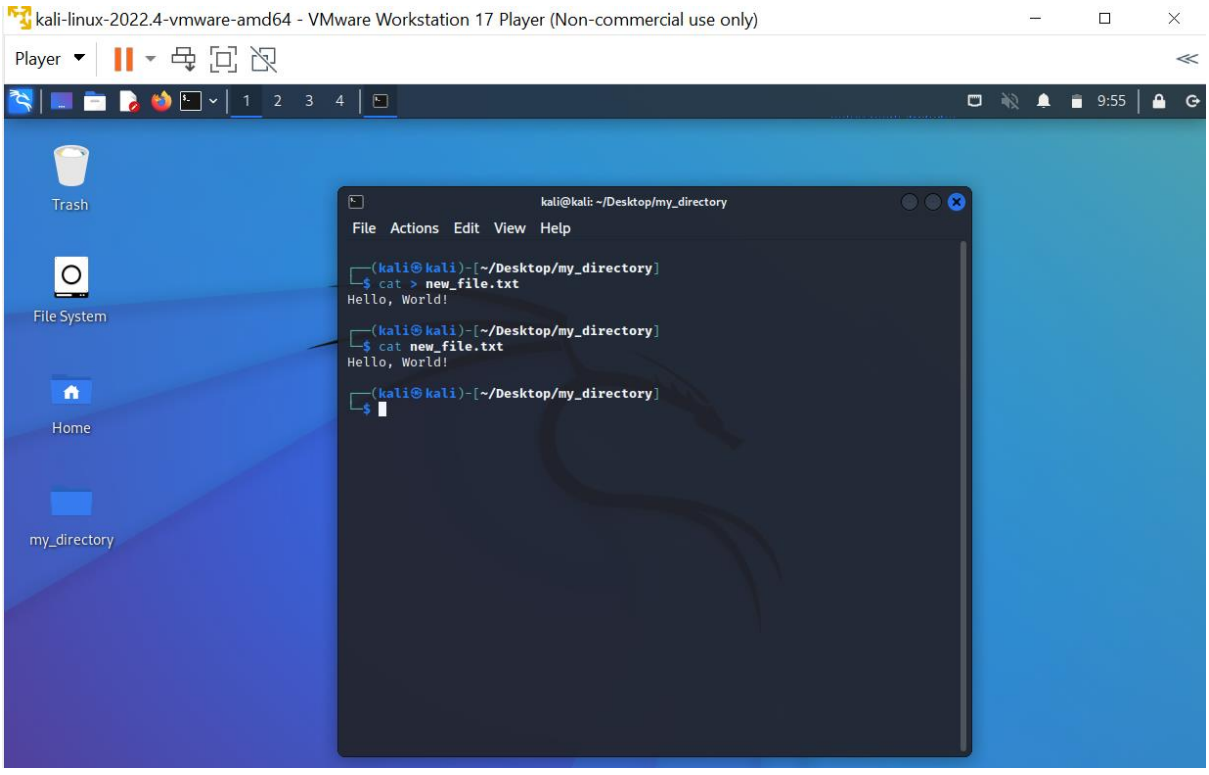
6. Display the content of "new_file.txt" using a pager tool of your choice.

Commands used: cat new_file.txt (the file is empty)



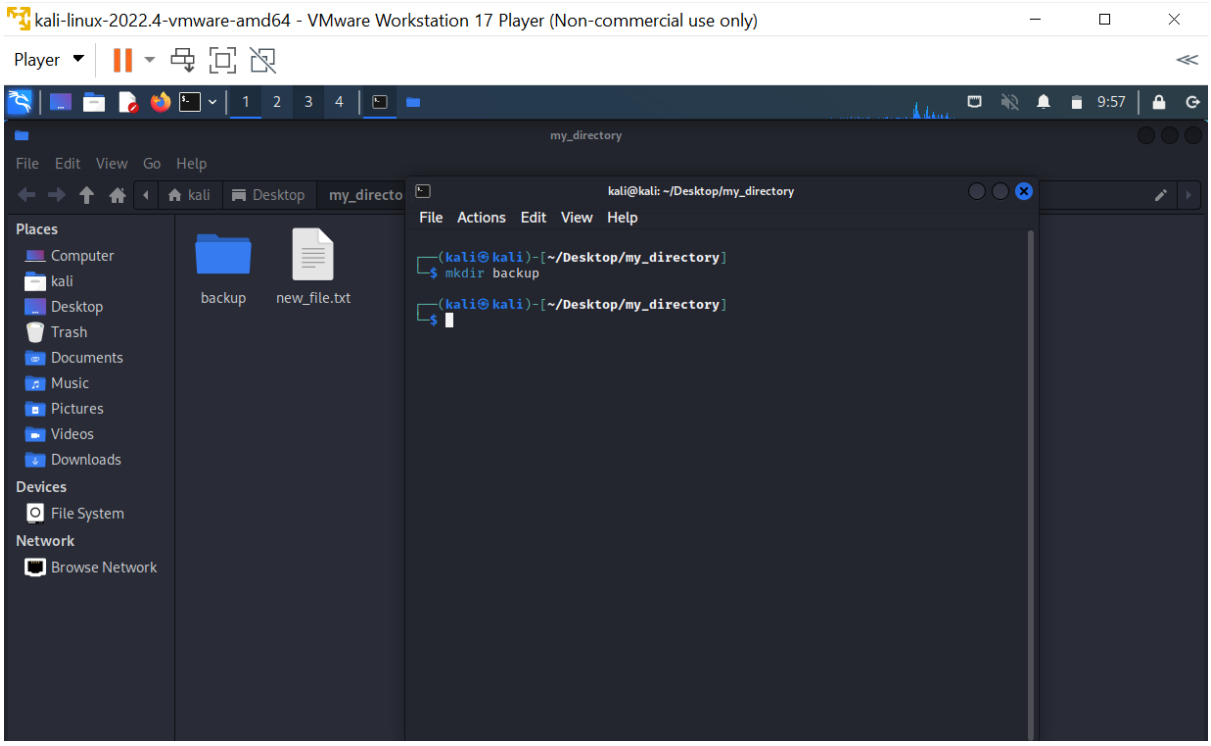
7. Append the text "Hello, World!" to "new_file.txt".

Commands used: cat > new_file.txt



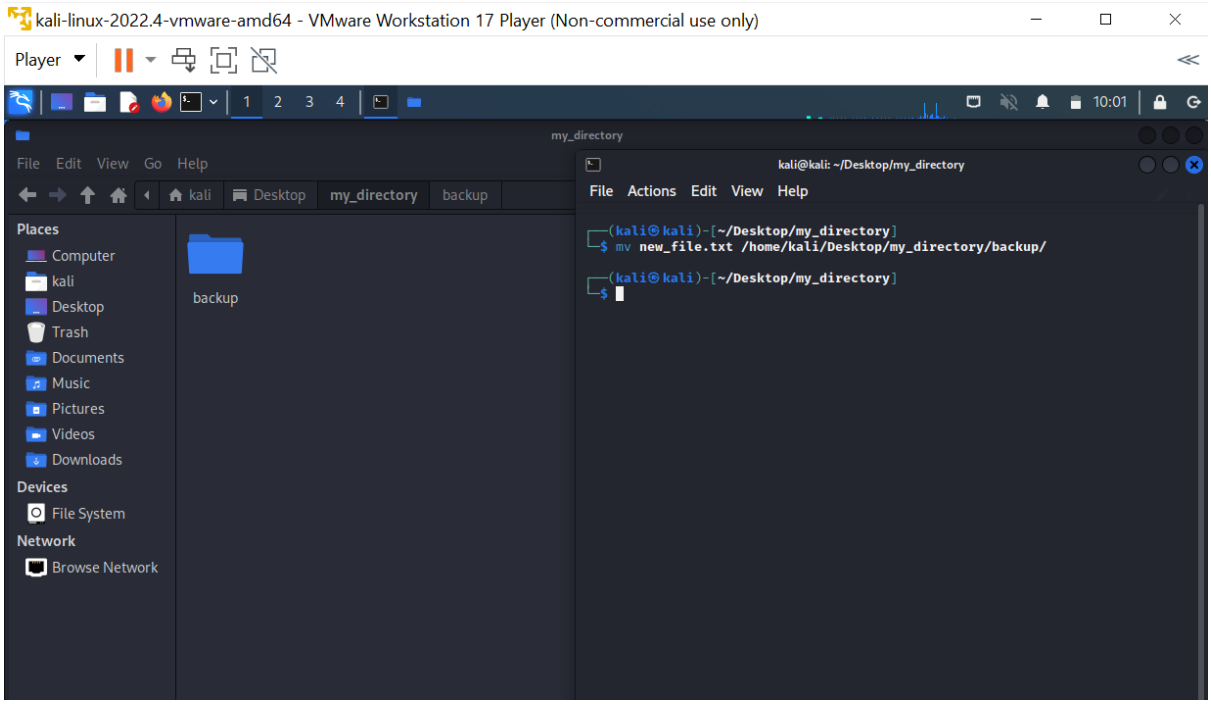
8. Create a new directory called "backup" within "my_directory".

Commands used: mkdir backup



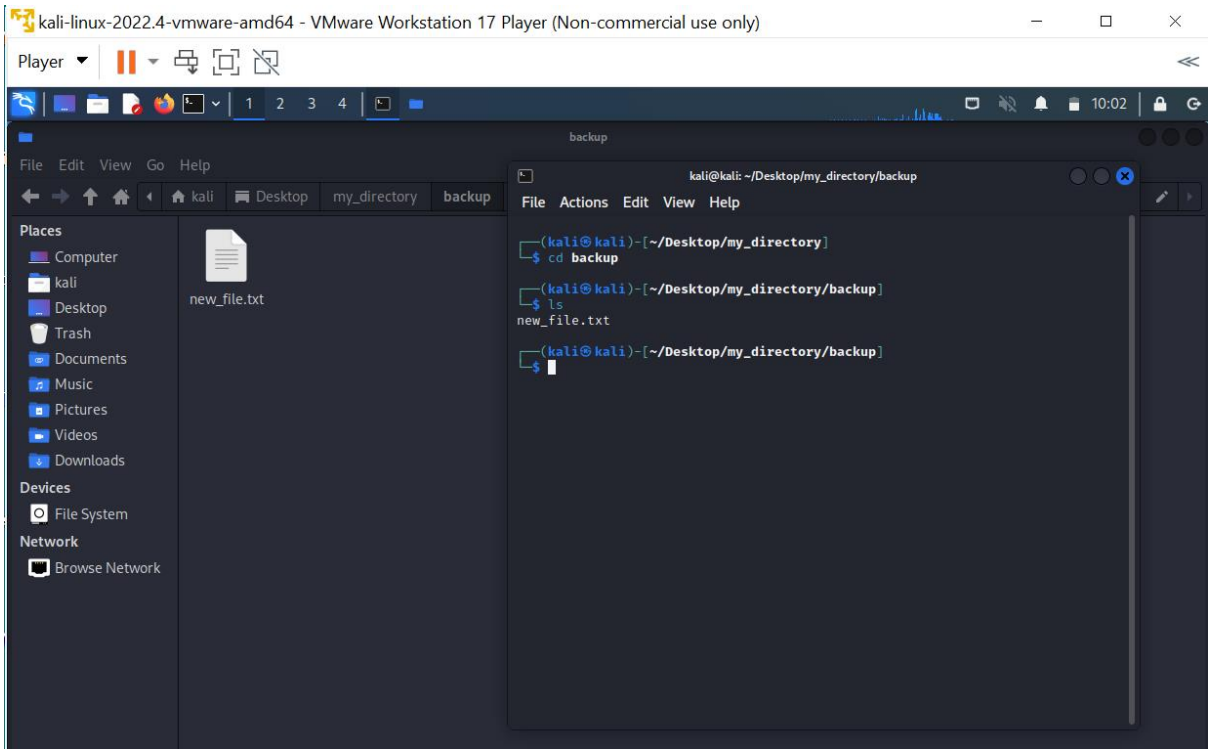
9. Move "new_file.txt" to the "backup" directory.

Commands used: mv new_file.txt /home/kali/Desktop/my_directory/backup



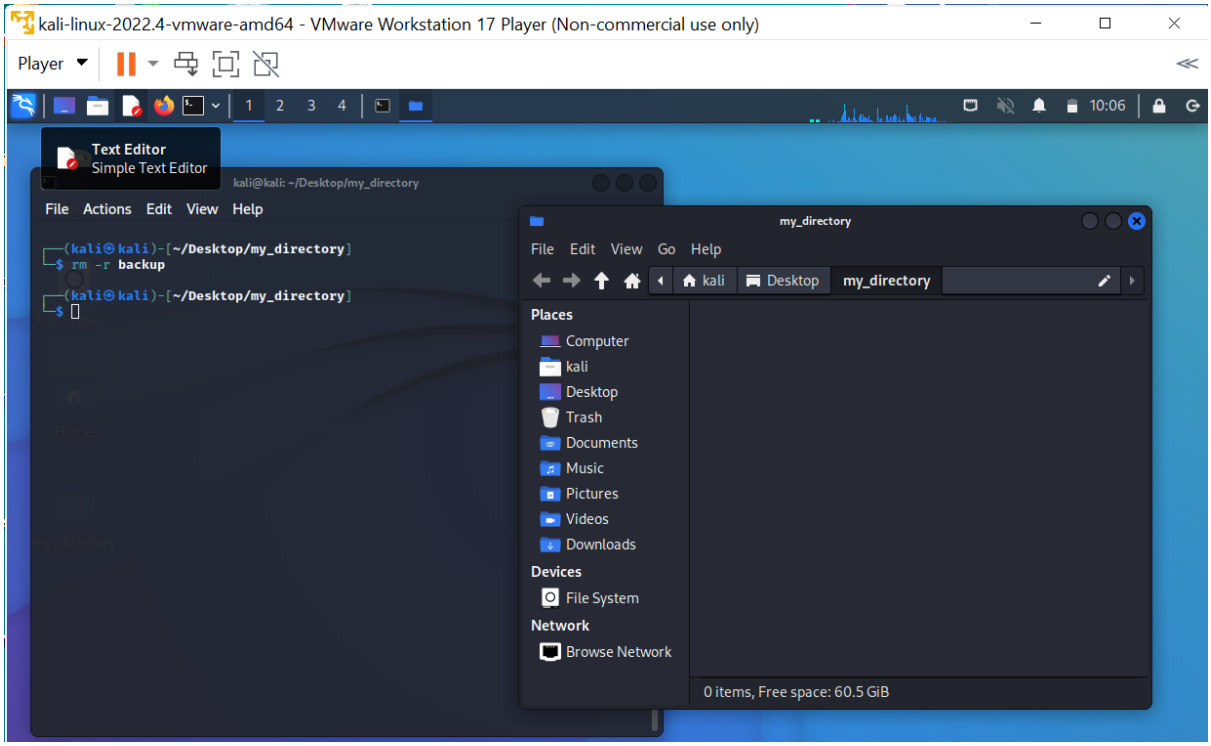
10. Verify that "new_file.txt" is now located in the "backup" directory.

Commands used: ls



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

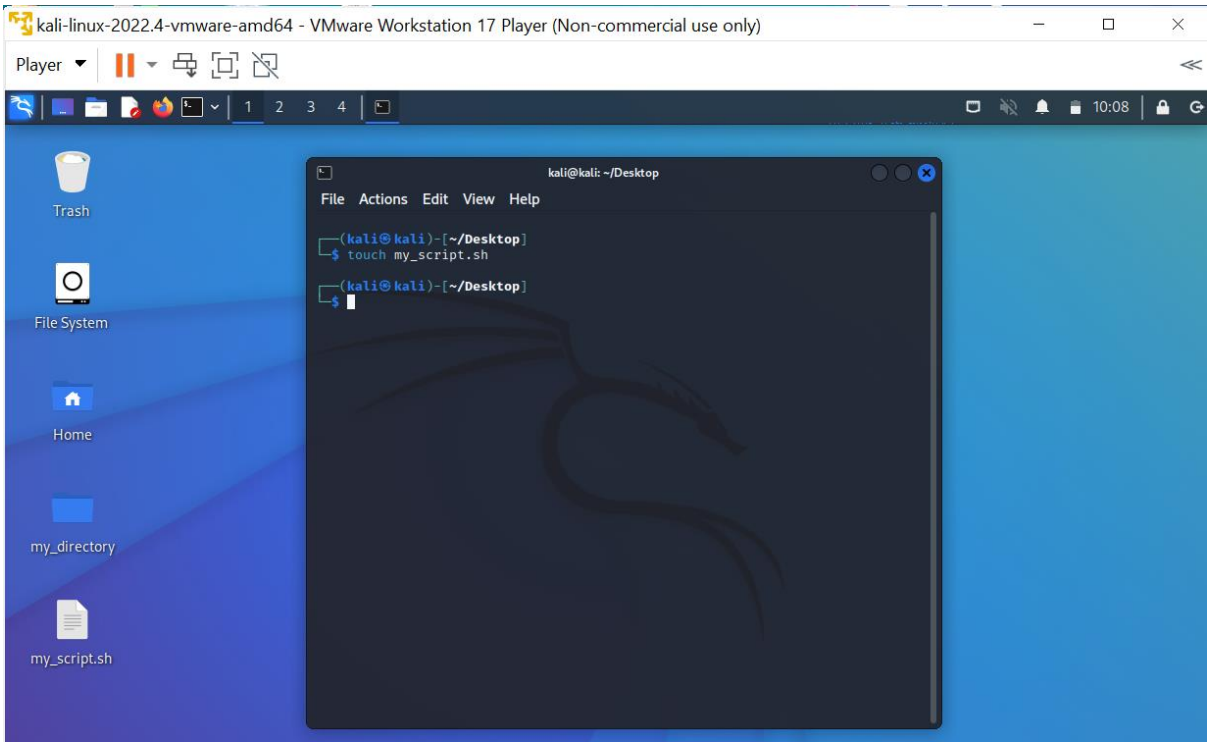
Commands used: rm -r backup



Task 2: Permissions and Scripting

● Create a new file called "my_script.sh".

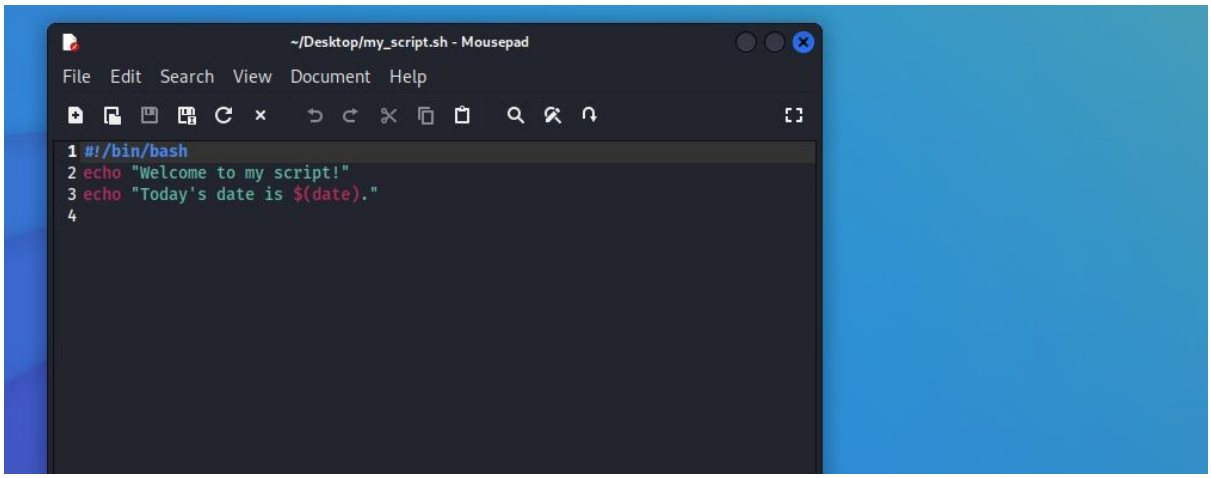
Commands used: touch my_script.sh

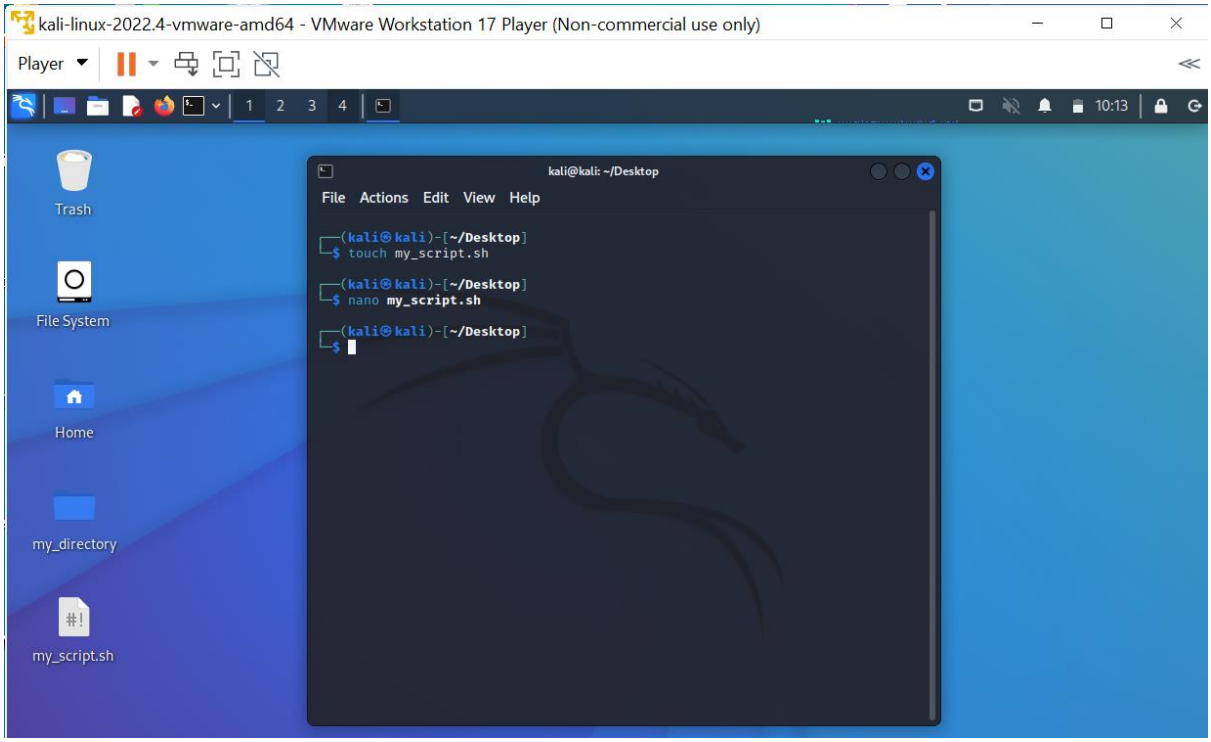


● Edit "my_script.sh" using a text editor of your choice and add the following lines: bash

#!/bin/bash
echo "Welcome to my script!"
echo "Today's date is $(date)."
Save and exit the file.

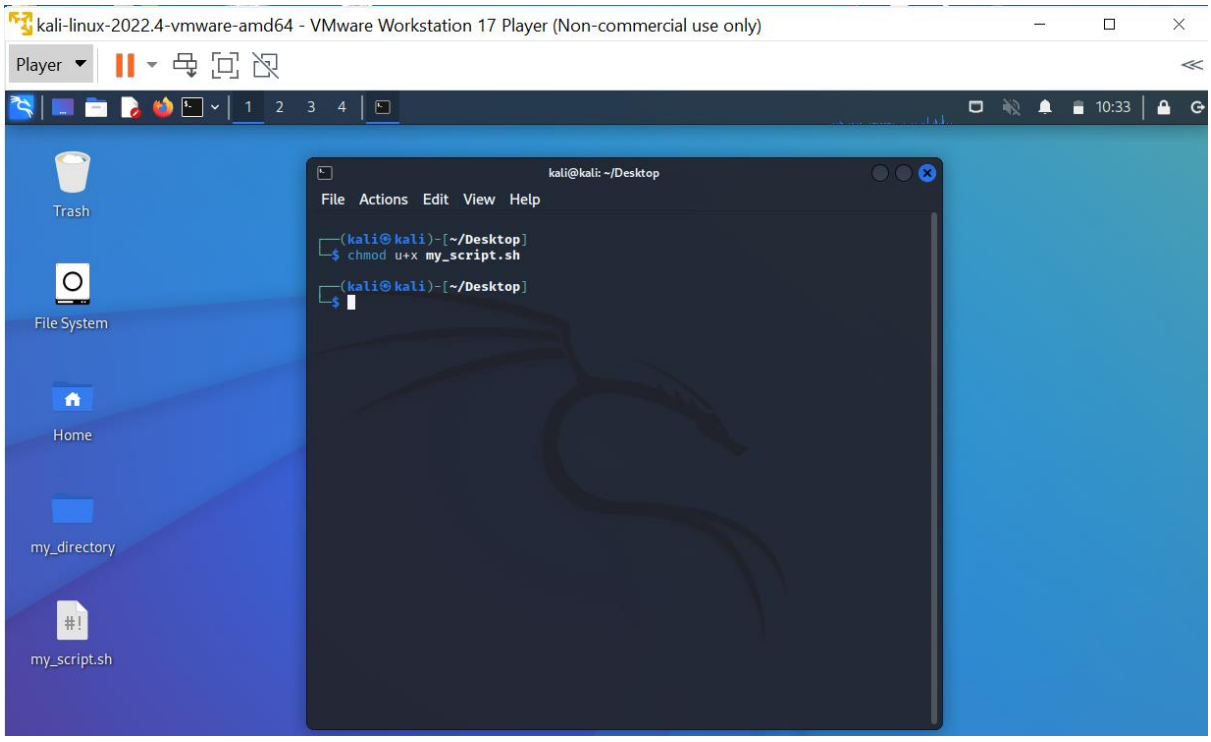
Commands used: nano my_script.sh





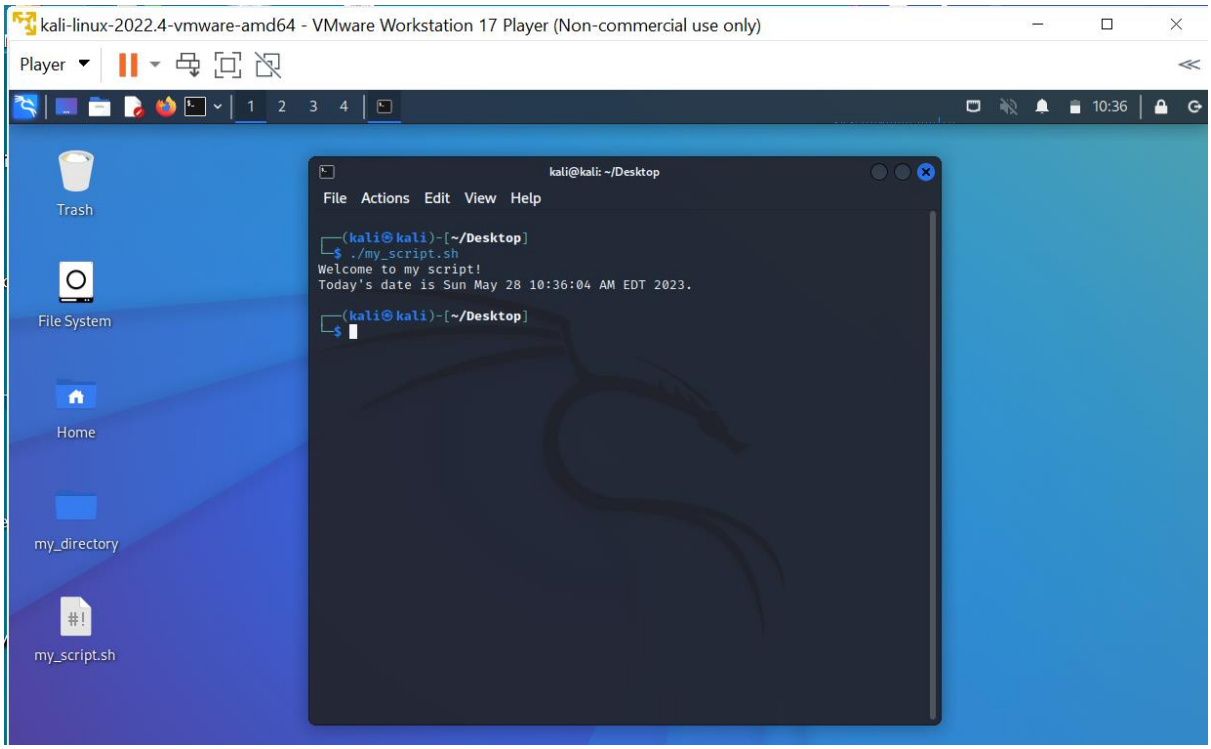
● Make "my_script.sh" executable.

Commands used: chmod u+x my_script.sh



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

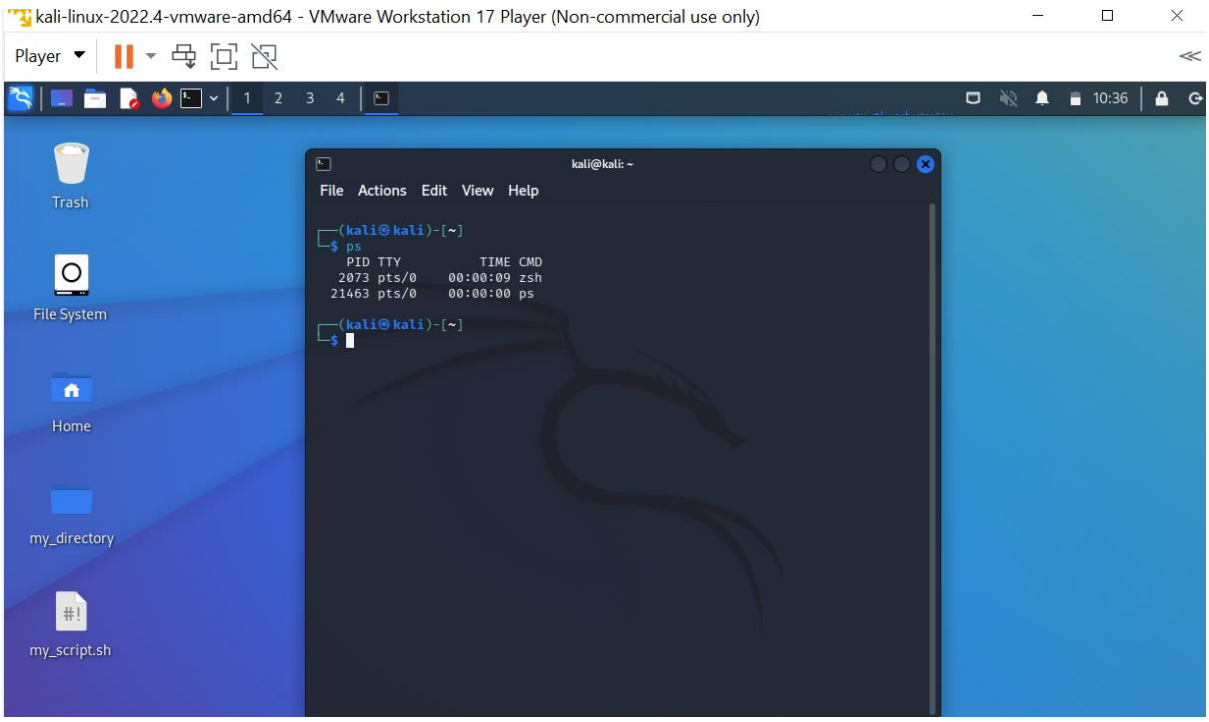
Commands used: ./my_script.sh



Task 3: Command Execution and Pipelines

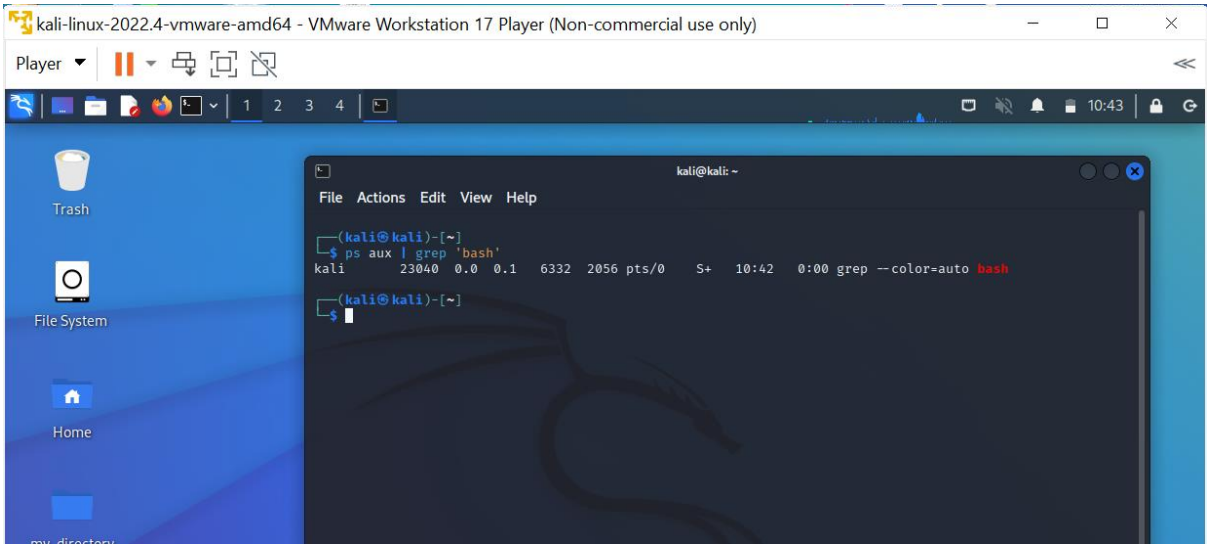
● List all the processes running on your system using the "ps" command.

Commands used: ps – this command lists the active processes and their PIDs



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

Commands used: ps aux | grep ‘bash’



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

Commands used: ps aux | grep ‘bash’ | wc