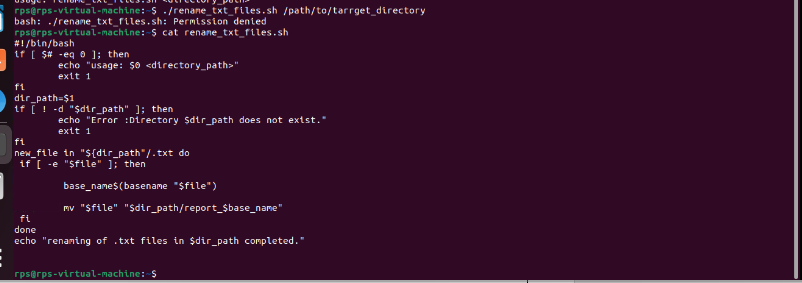
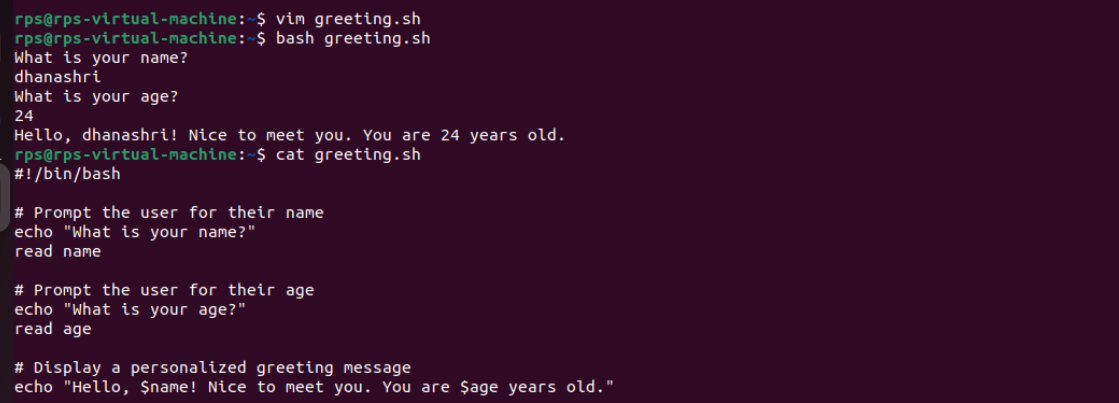
**Date:22-07-2024**

**Day\_5\_Linux\_programming**:

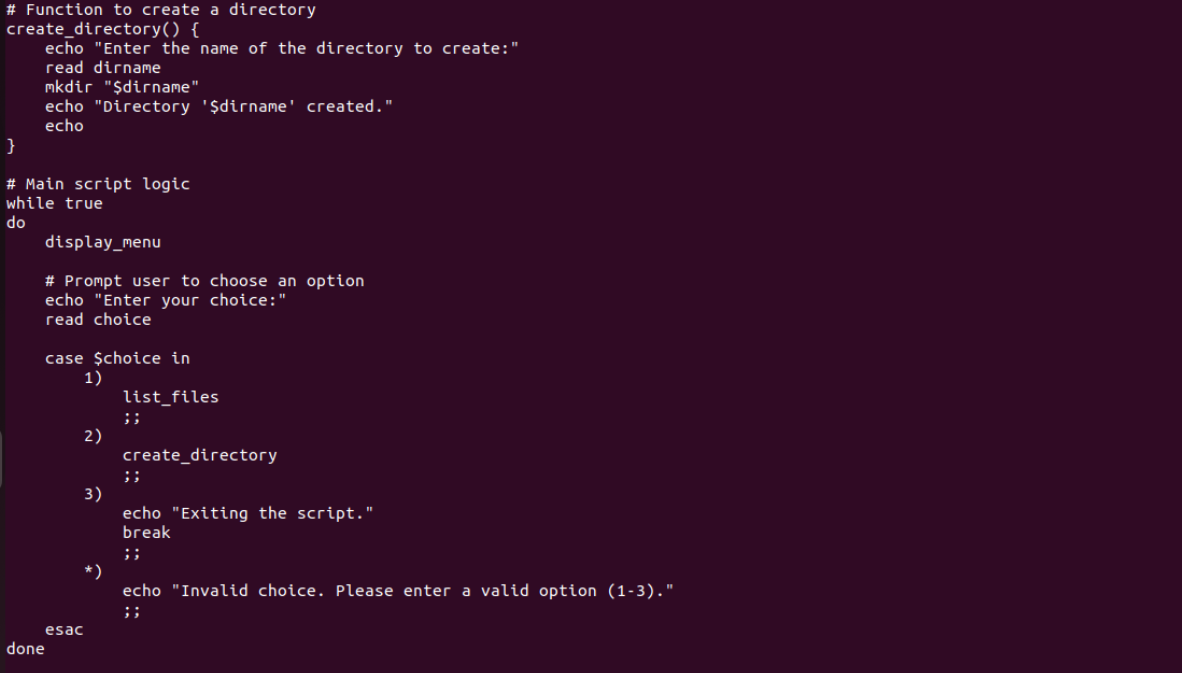
**1.Create a script that renames all files in a directory with the extension ".txt" to have a prefix of "report\_**".

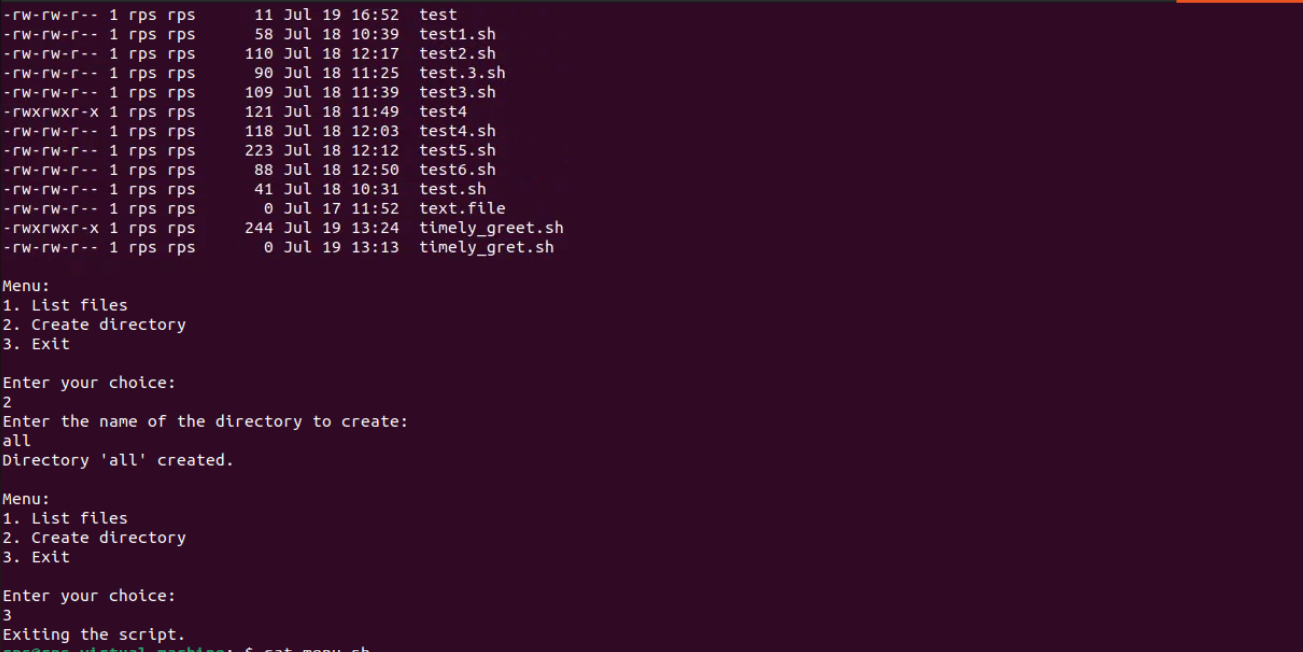


**2. Write a script that prompts the user for their name and age, then greets them with a personalized message.**

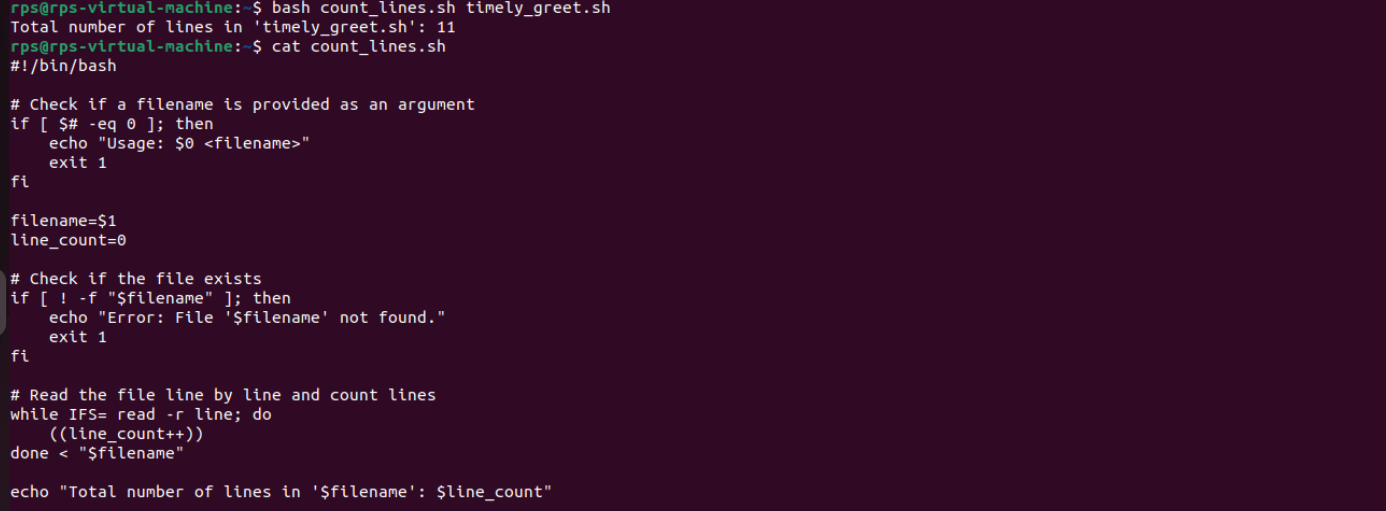


**3. Design a script that displays a menu with options like "List files," "Create directory," and "Exit." Allow the user to choose an option and perform the corresponding action.**

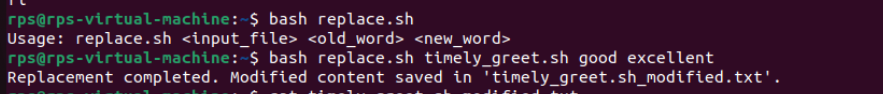


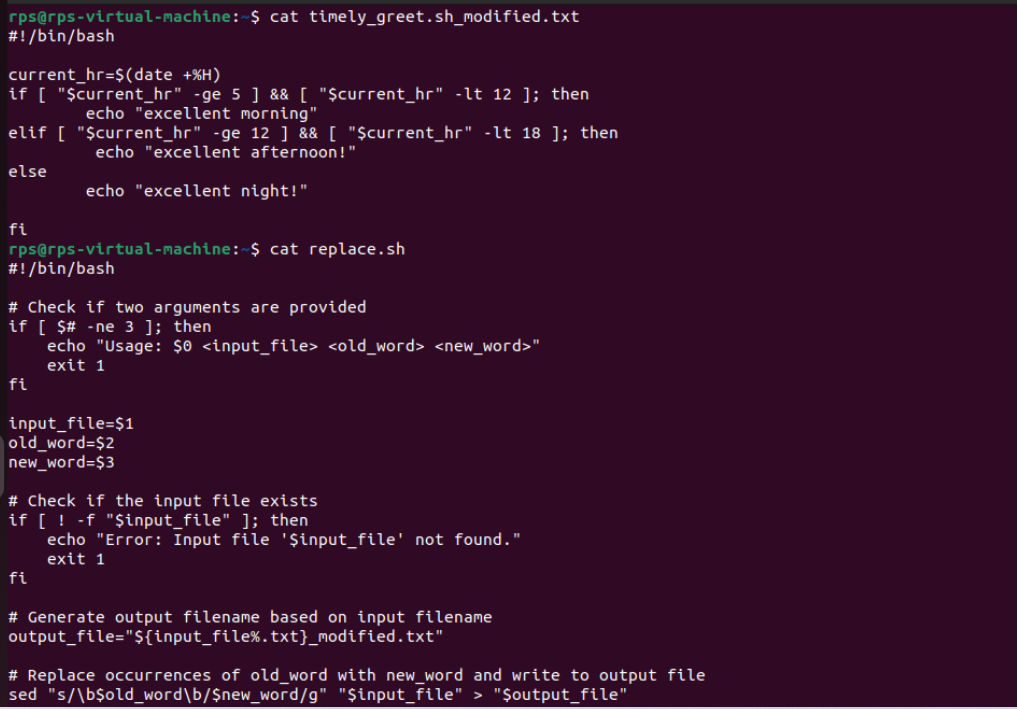


**4. Write a script that reads the contents of a file line by line, counts the number of lines, and prints the total.**

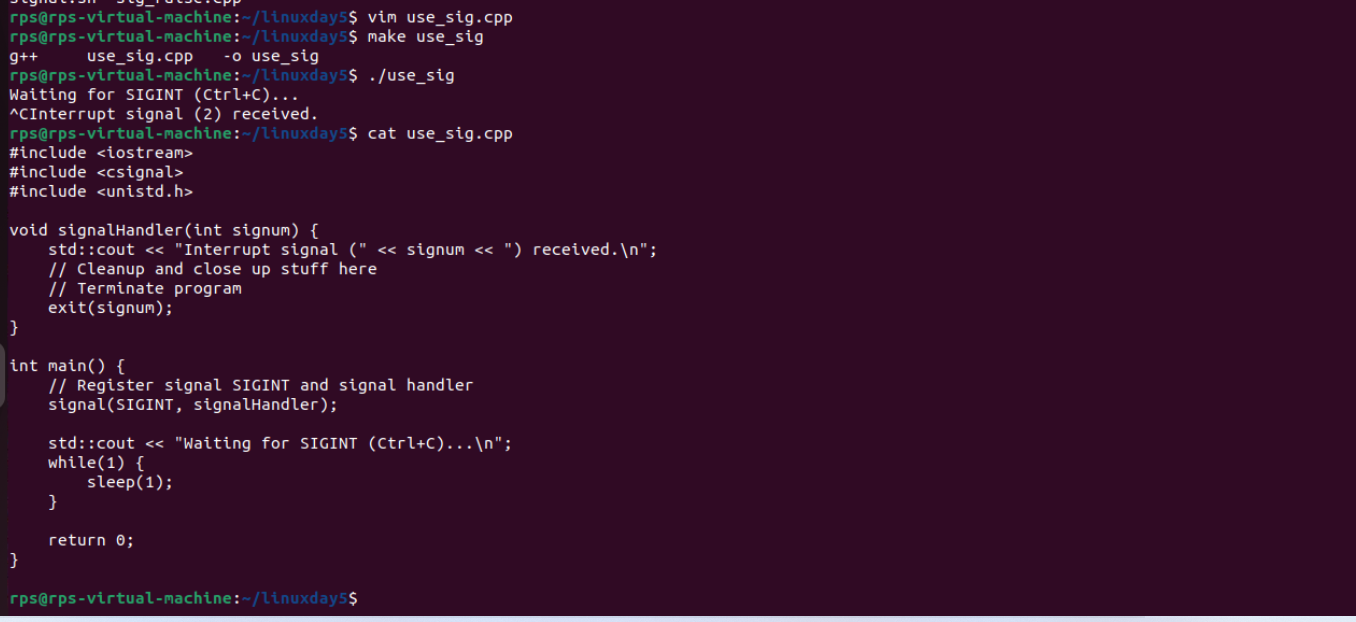


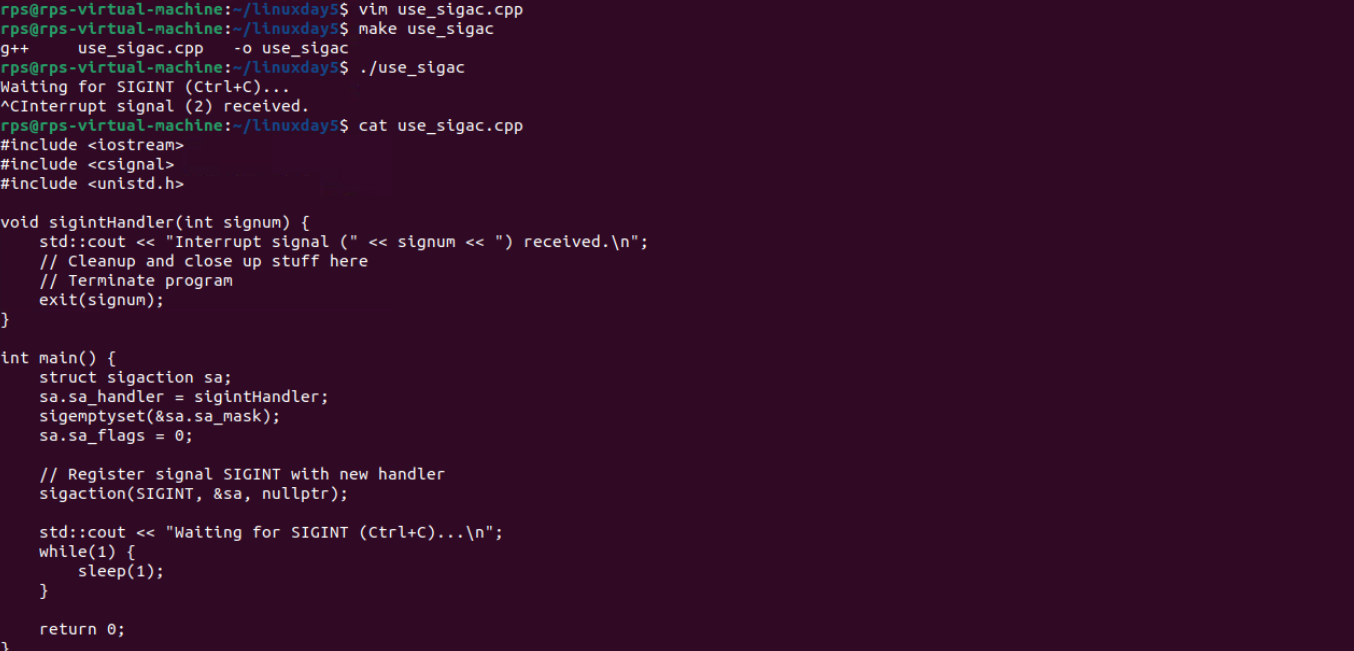
**5. Create a script that takes a text file as input and replaces all occurrences of a specific word with another word.**



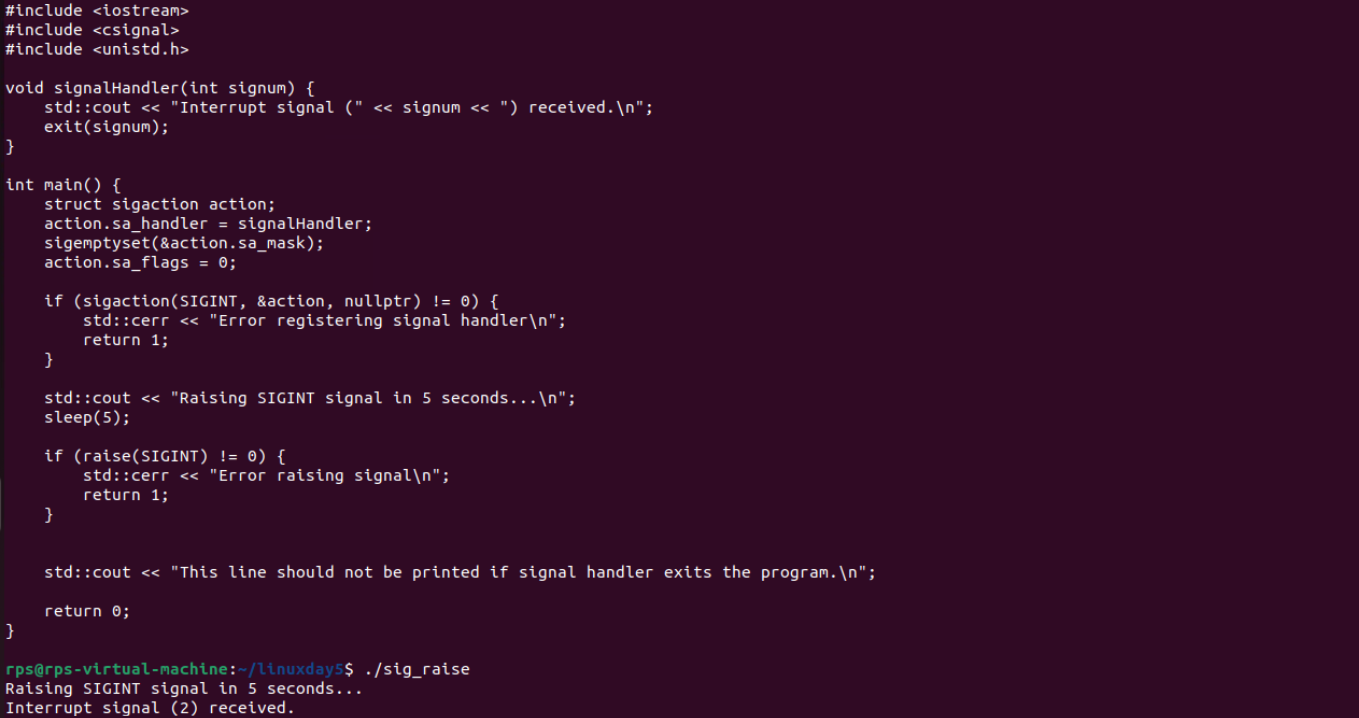


**6. Basic Handling vs. Advanced Control: Implement signal handling using both signal and sigaction (in separate program runs). Observe the behavior. Which API allows for more control over the signal handler? Explain the key difference in a comment within your code.**



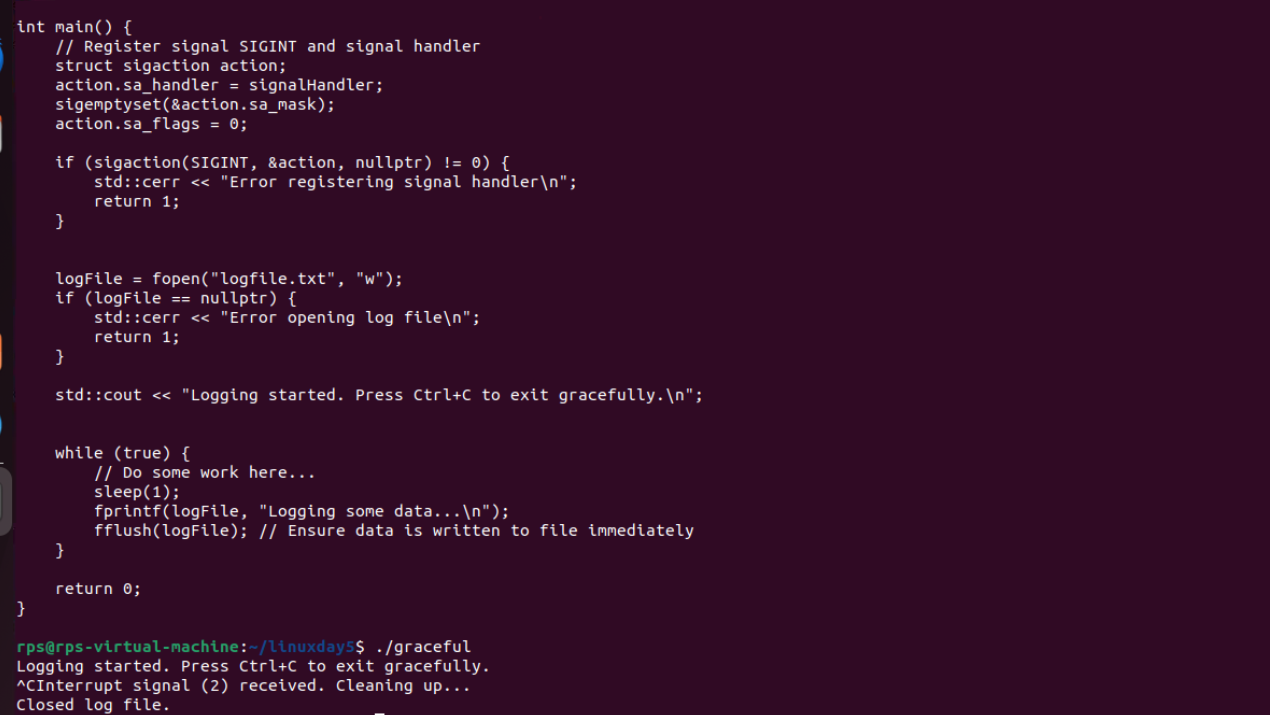


7.



**8. Graceful Termination with Signal HandlingObjective: Modify your program to demonstrate graceful termination upon receiving a specific signal (e.g., SIGINT). Within the signal handler, perform any necessary cleanup tasks (e.g., closing files, releasing resources) before exiting the program gracefully.Implementation:In your signal handler function, include code to perform cleanup actions. This might involve closing open files, releasing memory, or writing data to disk.Use exit(0) or similar methods to terminate the program after cleanup is complete.**





9.

