**LAB REPORT**

**Lab Assignment 6: CPU Scheduling Algorithms**

**1. Objective**

* To design a **menu-driven program** that allows the user to select a CPU scheduling algorithm.
* To implement **FCFS (First-Come, First-Served)**, **SJF (Shortest Job First)**, **SRTN (Shortest Remaining Time Next)**, and **RR (Round Robin)** algorithms.
* To calculate and display the **Average Turnaround Time (TAT)** and **Average Waiting Time (WT)** for each scheduling method.
* To understand how different scheduling algorithms affect process performance.

**2. Theory**

**CPU Scheduling** is a core concept in operating systems that determines the order in which processes are executed by the CPU. Each scheduling algorithm has its own logic and performance characteristics:

* **First-Come, First-Served (FCFS)**: Processes are executed in the order they arrive. Simple but can lead to the **convoy effect**.
* **Shortest Job First (SJF)**: Executes the process with the smallest burst time first. Reduces average waiting time but requires knowledge of burst times.
* **Shortest Remaining Time Next (SRTN)**: Preemptive version of SJF where the process with the shortest remaining burst time is executed next.
* **Round Robin (RR)**: Each process is given a fixed time quantum in rotation, ensuring fairness but possibly increasing context switches.

**Key Metrics:**

* **Turnaround Time (TAT)** = Completion Time − Arrival Time
* **Waiting Time (WT)** = Turnaround Time − Burst Time

**3. Tools and Commands Used**

* **gcc** – GNU Compiler Collection for compiling C programs.
* **Linux Terminal** – To execute and test the program.
* **C Programming Language** – For implementing scheduling algorithms.

**4. Procedure**

1. **Menu Creation**
   * Display scheduling options (FCFS, SJF, SRTN, RR).
   * Accept the user’s choice.
2. **Input Process Data**
   * Take the number of processes, burst times, and arrival times from the user.
   * For RR, also take the time quantum.
3. **Algorithm Implementation**
   * **FCFS**: Sort by arrival time and calculate completion times.
   * **SJF**: Select the shortest burst time available at each step.
   * **SRTN**: Continuously check for the process with the shortest remaining time and execute it.
   * **RR**: Allocate CPU to each process in a cyclic order with the given time quantum
4. **Calculate and Display Results**
   * For each method, calculate **TAT** and **WT** for all processes.
   * Compute and display **average TAT** and **average WT**.

**5. Program**

a. Initiate the Directory.

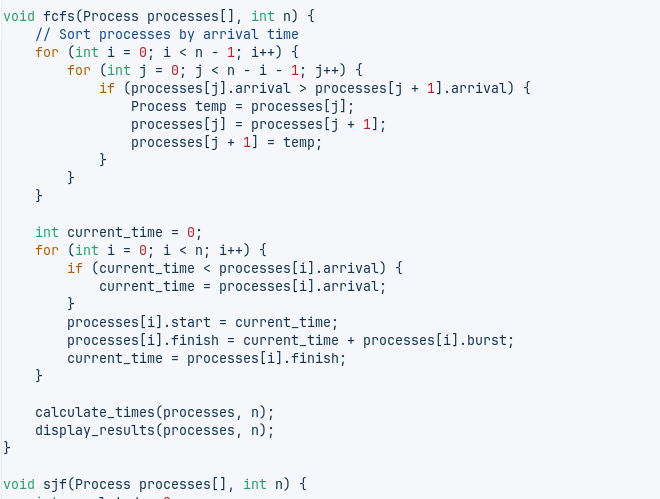


b. Create the C-Language file.  


c. Write the single menu-driven Program code.













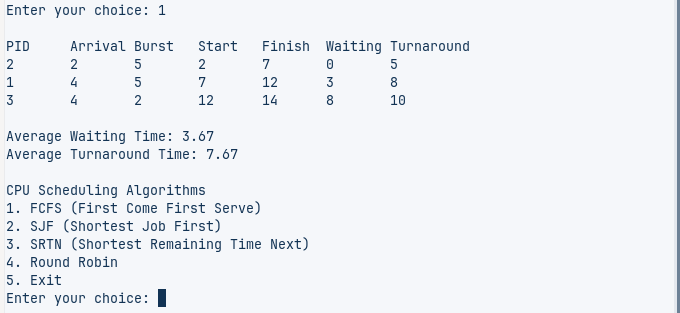
d. Compile the C-Language file.

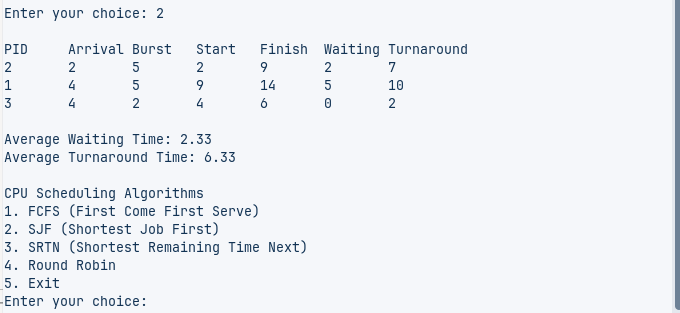
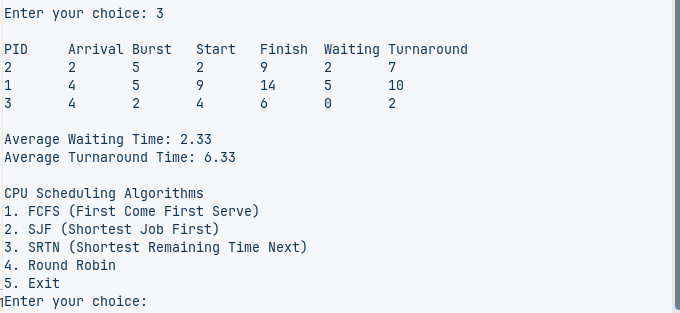


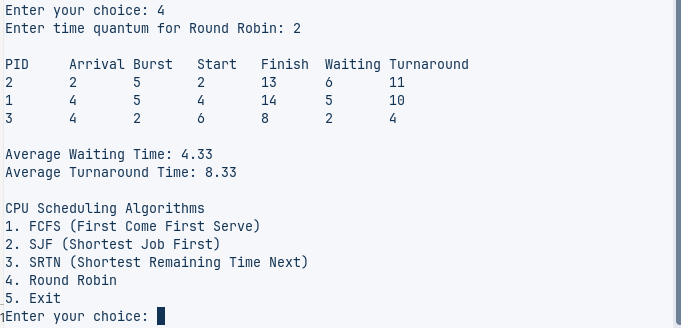
e. Execute the File.



f. Respectively run the menu driven .







**6. Conclusion**

This lab demonstrated how different CPU scheduling algorithms impact process execution efficiency. We implemented FCFS, SJF, SRTN, and RR in a single menu-driven program, allowing direct comparison of their performance. By calculating average turnaround time and waiting time, we observed that SJF and SRTN often minimize delays, while RR provides fairness among processes. Understanding these algorithms is essential for designing effective process scheduling in operating systems.