

National University of Computer and Emerging
Sciences Chiniot-Faisalabad Campus



Lab 09

CL1006 – Operating System - Lab

Course Instructor	Juhinah Batool Asif
Lab Instructor	Juhinah Batool Asif
Semester	Fall 2024

FAST School of Computing
Department of AI & Data Science

Instructions

1. Make a word document with the convention “ROLLNO_ LAB#_ SECTION” and put all your source code and snapshots of its output in it. You have to submit a pdf file.
2. Plagiarism is strictly prohibited.
3. Do not discuss solutions with one another. Copying the solution from any source can lead to ZERO marks.

Lab Task

Objective

In this lab, you will implement three CPU scheduling algorithms: First-Come-First-Serve (FCFS) and Round Robin (RR). You'll compare their performance based on average waiting time and turnaround time.

Instructions

1. Set up the environment:

- Create a console-based program in C, C++, Python, or Java.
- Design your program to accept a set of processes with the following properties:
 - **Process ID** (integer): Identifier for each process.
 - **Burst Time** (integer): The CPU time required by the process.
 - **Arrival Time** (integer): The time at which the process arrives in the system.

2. Implement FCFS Scheduling:

- In this algorithm, processes are scheduled in the order of their arrival time.
- Calculate the **waiting time** and **turnaround time** for each process.
- Display the process order and average waiting and turnaround times.

3. Implement Round Robin Scheduling:

- This algorithm assigns a fixed time quantum to each process. If a process's burst time is greater than the quantum, it goes back to the end of the queue after using up its quantum.
- Allow the user to set the **time quantum**.
- Calculate and display the waiting time and turnaround time for each process.
- Display the process order and average waiting and turnaround times.

4. Input and Output Requirements:

- Prompt the user to enter the **number of processes**.
- For each process, input the **Process ID**, **Burst Time**, and **Arrival Time**.
- Display results for each scheduling algorithm, including:
 - The **order of process execution**.
 - The **waiting time** and **turnaround time** for each process.
 - The **average waiting time** and **average turnaround time** for each scheduling algorithm.

5. Compare Results:

- At the end of your program, display a summary that compares the average waiting and turnaround times for each algorithm.
- Reflect on the strengths and weaknesses of each scheduling algorithm based on your results.