

# **Laboratory Assignments**

## **Subject: Design of Operating Systems**

### **Subject code: CSE 4049**

## **Assignment 5: End term project on CPU Scheduling**

### **Objective of this Assignment:**

- To design a CPU scheduler for simulating a few CPU scheduling policies.

### **Overview of the Project:**

One of the main tasks of an operating system is scheduling processes to run on the CPU. The goal of this programming project is to build a program using C to implement a simulator with different scheduling algorithms discussed in theory. The simulator should select a process to run from the ready queue based on the scheduling algorithm chosen at runtime. Since the assignment intends to simulate a CPU scheduler, it does not require any actual process creation or execution.

**Project Description:** The C program provides an interface to the user to implement the following scheduling policies as per the choice provided:

1. First Come First Served (FCFS)
2. Shortest Job First (SJF)
3. Shortest Remaining Time First (SRTF)
4. Round Robin (RR)

Appropriate option needs to be chosen from a switch case based menu driven program with an option of “Exit from program” in case 5 and accordingly a scheduling policy will print the Gantt chart and the average waiting time, average turnaround time and average response time. The program will take Process ids, its arrival time, and its CPU burst time as input. For implementing RR scheduling, user also needs to specify the time quantum. Assume that the process ids should be unique for all processes. Each process consists of a single CPU burst (no I/O bursts), and processes are listed in order of their arrival time. Further assume that an interrupted process gets placed at the back of the Ready queue, and a newly arrived process gets placed at the back of the Ready queue as well. The output should be displayed in a formatted way for clarity of understanding and visual.

### Test Cases:

The program should be able to produce correct answer or appropriate error message corresponding to the following test cases:

1. Consider the set of processes with arrival time (in milliseconds), CPU burst time (in milliseconds), and time quantum = 4ms as shown below.

Process	Arrival time	Burst Time
P1	0	10
P2	0	1
P3	0	2
P4	0	1
P5	0	5

- Input choice 1, and print the Gantt charts that illustrate the execution of these processes using the FCFS scheduling algorithm and then print the average turnaround time, average waiting time and average response time.
  - Input choice 2, and print the Gantt charts that illustrate the execution of these processes using the SJF scheduling algorithm and then print the average turnaround time, average waiting time and average response time.
  - Input choice 3, and print the Gantt charts that illustrate the execution of these processes using the SRTF scheduling algorithm and then print the average turnaround time, average waiting time and average response time.
  - Input choice 4, and print the Gantt charts that illustrate the execution of these processes using the RR scheduling algorithm and then print the average turnaround time, average waiting time and average response time.
  - Analyze the results and determine which of the algorithms results in the minimum average waiting time over all processes?
2. Consider the set of processes with arrival time (in milliseconds), CPU burst time (in milliseconds), and time quantum = 2ms as shown below.

Process	Arrival time	Burst Time
P1	0	4
P2	0	2
P3	1	3
P4	2	2

- Input choice 1, and print the Gantt charts that illustrate the execution of these processes using the FCFS scheduling algorithm and then print the average turnaround time, average waiting time and average response time.

- Input choice 2, and print the Gantt charts that illustrate the execution of these processes using the SJF scheduling algorithm and then print the average turnaround time, average waiting time and average response time.
- Input choice 3, and print the Gantt charts that illustrate the execution of these processes using the SRTF scheduling algorithm and then print the average turnaround time, average waiting time and average response time.
- Input choice 4, and print the Gantt charts that illustrate the execution of these processes using the RR scheduling algorithm and then print the average turnaround time, average waiting time and average response time.
- Analyze the results and determine which of the algorithms results in the minimum average waiting time over all processes?