ព្រះរាជាណាចក្រកម្ពុជា
ជាតិ សាសនា ព្រះមហាក្សត្រ

Operating System

Project : CPU Scheduling Simulator

Group: B-07

Name of Student	ID of Student	Score
PENG SEYHA	e20220620	……..
OUN RITHI	e20220420	……..
SOEUN SEREYVATH	e20220907	……..
SAM NANGALEX	e20220253	……..

Lecturer: Rathpisey Heng

Academic year 2025-2026

1. Introduction

CPU scheduling is a fundamental function of an operating system's process management. It determines the order in which processes access the CPU, aiming to optimize metrics such as waiting time, turnaround time, response time, and CPU utilization.

This project implements a **CPU Scheduling Algorithm Simulator** that helps visualize and compare scheduling strategies including **FCFS, SJF, SRT, Round Robin, and MLFQ**. The simulator accepts user-defined processes and generates Gantt charts along with key performance metrics.

2. Project Objectives

The main goals of this project are:

- To understand the inner workings of CPU scheduling in modern operating systems.
- To implement and simulate various scheduling policies.
- To visualize CPU allocation through Gantt charts.
- To compute and compare essential scheduling metrics.
- To allow users to define custom process sets and parameters such as time quantum.

3. System Features & Functionality

3.1 Input Specifications

Each process contains the following attributes:

- **Process ID**
- **Arrival Time**
- **Burst Time**

Input can be provided via:

- Console
- External file (CSV/JSON)

4. Scheduling Algorithms Implemented

Below is a description of each scheduling algorithm featured in the simulator.

4.1 First Come First Serve (FCFS)

A non-preemptive algorithm that schedules processes in the order they arrive.

Characteristics:

- Simple to implement
- High waiting time for long jobs (Convoy effect)

4.2 Shortest Job First (SJF) — Non-Preemptive

Selects the process with the smallest CPU burst among arrived processes.

Pros: Optimal for waiting time
Cons: Risk of starvation

4.3 Shortest Remaining Time (SRT) — Preemptive SJF

The CPU switches to a newly arrived process if it has shorter remaining time.

Pros: Better response time for short jobs
Cons: More context switching

4.4 Round Robin (RR)

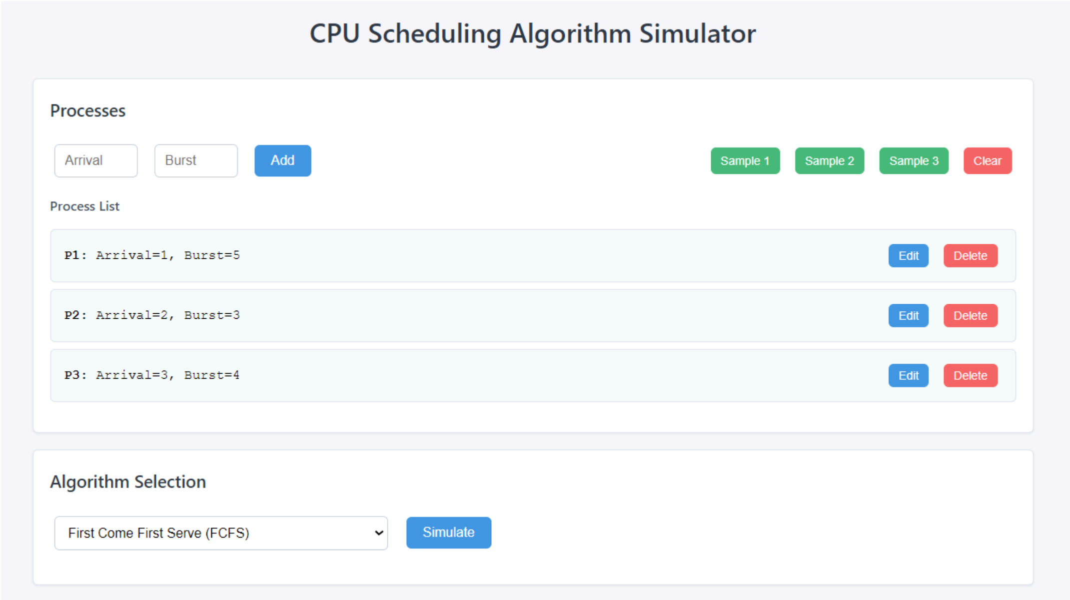
Allocates CPU time in time-slices (quantum). Preemptive and fair.

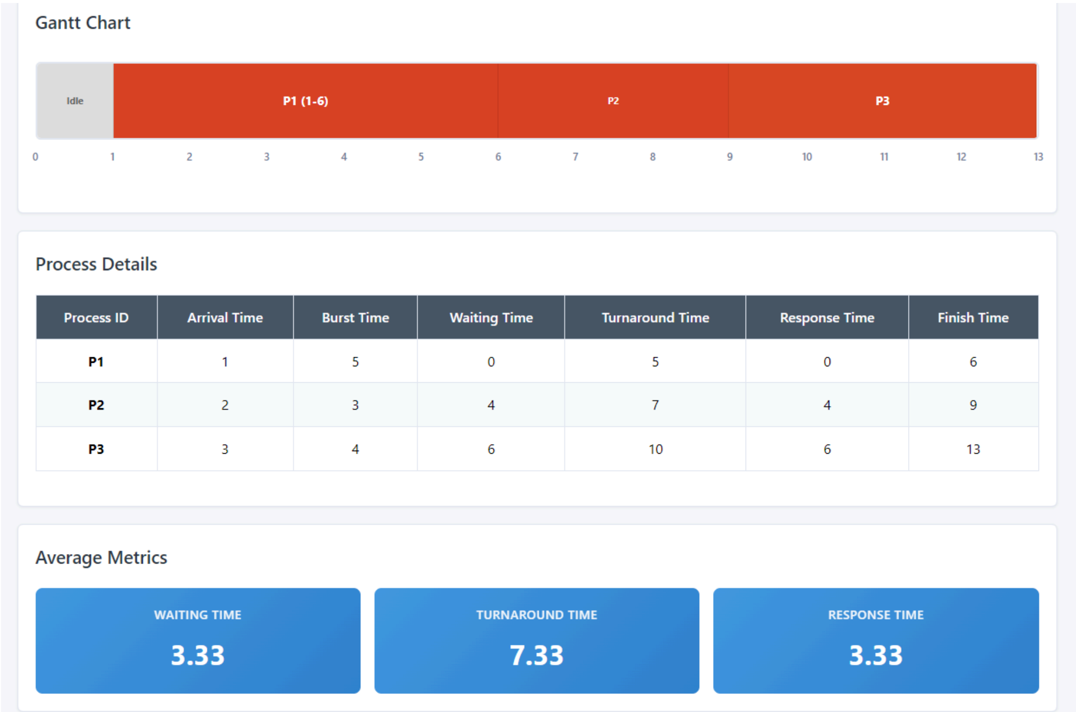
Configuration: Quantum is user-defined.

Pros: Ideal for time-sharing systems
Cons: Too small quantum → many context switches; too large quantum → becomes FCFS

5. Input and Output

FCFS:


CPU Scheduling Algorithm Simulator
Processes
Arrival
Burst
Add
Sample 1
Sample 2
Sample 3
Clear
Process List
P1: Arrival=1, Burst=5
Edit
Delete
P2: Arrival=2, Burst=3
Edit
Delete
P3: Arrival=3, Burst=4
Edit
Delete
Algorithm Selection
First Come First Serve (FCFS)
Simulate


Gantt Chart
Idle
P1 (1-6)
P2
P3
0 1 2 3 4 5 6 7 8 9 10 11 12 13
Process Details
Process ID	Arrival Time	Burst Time	Waiting Time	Turnaround Time	Response Time	Finish Time
P1	1	5	0	5	0	6
P2	2	3	4	7	4	9
P3	3	4	6	10	6	13
Average Metrics
WAITING TIME
3.33
TURNAROUND TIME
7.33
RESPONSE TIME
3.33

SRT

CPU Scheduling Algorithm Simulator

Processes

Arrival | Burst | Add

P1: Arrival=0, Burst=4 Edit Delete

P2: Arrival=1, Burst=3 Edit Delete

P3: Arrival=2, Burst=1 Edit Delete

Algorithm Selection

Shortest Remaining Time (SRT) | Simulate

Gantt Chart

Process Details

Process ID	Arrival Time	Burst Time	Waiting Time	Turnaround Time	Response Time	Finish Time
P1	0	4	1	5	0	5
P2	1	3	4	7	4	8
P3	2	1	0	1	0	3

Average Metrics

WAITING TIME	TURNAROUND TIME	RESPONSE TIME
1.67	4.33	1.33

SJF

Processes

Arrival | Burst | Add

P1: Arrival=0, Burst=4 Edit Delete

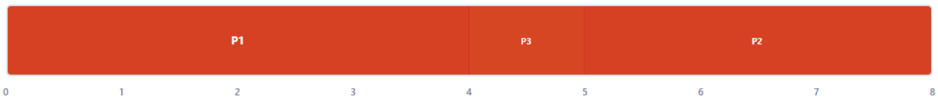
P2: Arrival=1, Burst=3 Edit Delete

P3: Arrival=2, Burst=1 Edit Delete

Algorithm Selection

Shortest Job First (SJF) | Simulate

Gantt Chart



Process Details

Process ID	Arrival Time	Burst Time	Waiting Time	Turnaround Time	Response Time	Finish Time
P1	0	4	0	4	0	4
P2	1	3	4	7	4	8
P3	2	1	2	3	2	5

Average Metrics

WAITING TIME	TURNAROUND TIME	RESPONSE TIME
2.00	4.67	2.00

RR

Processes

Arrival | Burst | Add

P1: Arrival=0, Burst=4 Edit Delete

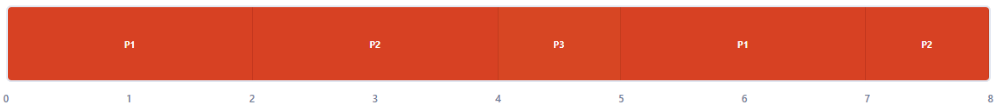
P2: Arrival=1, Burst=3 Edit Delete

P3: Arrival=2, Burst=1 Edit Delete

Algorithm Selection

Round Robin (RR) Quantum: 2 Simulate

Gantt Chart



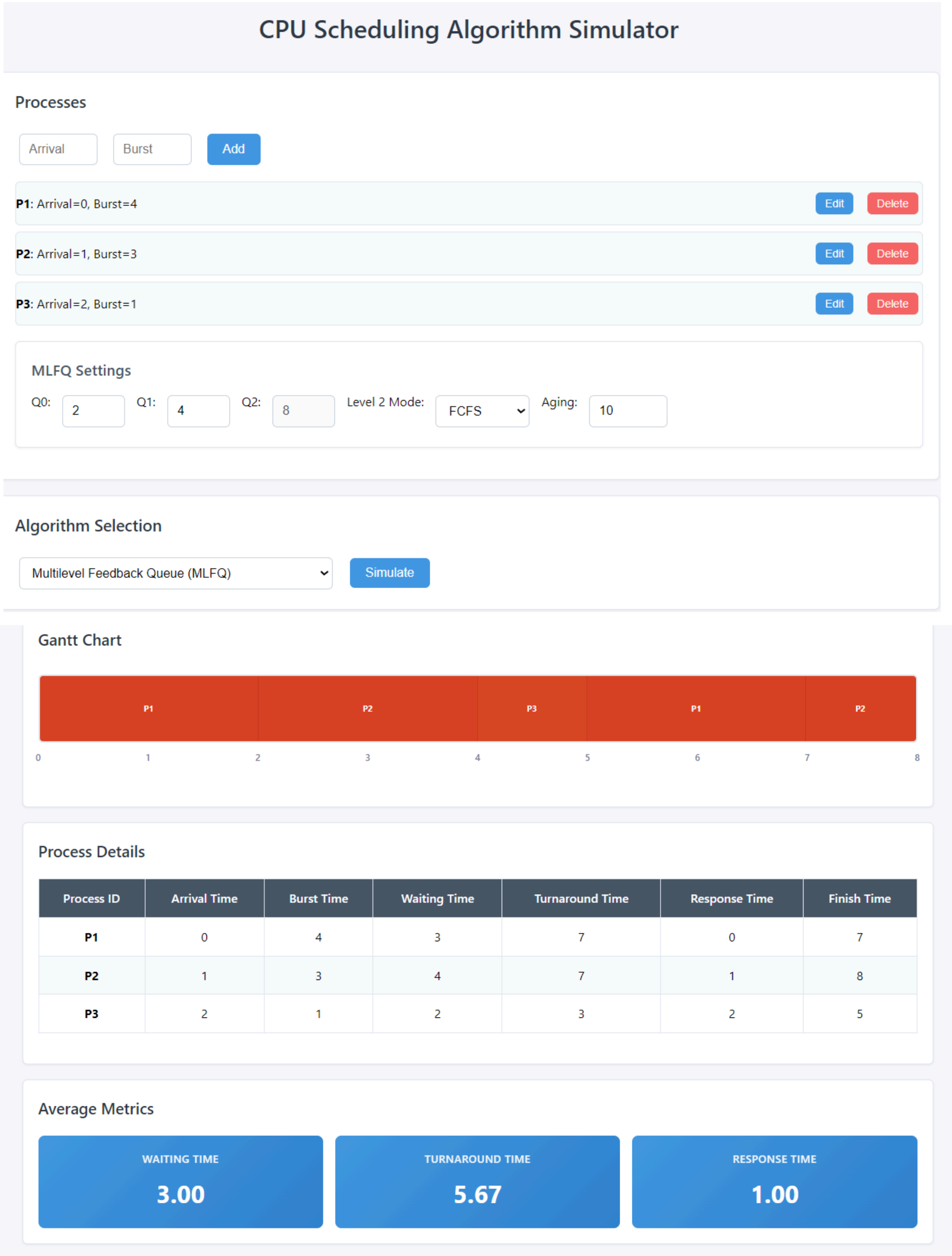
Process Details

Process ID	Arrival Time	Burst Time	Waiting Time	Turnaround Time	Response Time	Finish Time
P1	0	4	3	7	0	7
P2	1	3	4	7	1	8
P3	2	1	2	3	2	5

Average Metrics

WAITING TIME	TURNAROUND TIME	RESPONSE TIME
3.00	**5.67**	**1.00**

MLDQ



6. Challenges & Solutions

6.1 Handling Preemption

Preemptive algorithms (SRT, RR, MLFQ) require:

- Checking arrivals dynamically
- Updating remaining burst times

- Tracking when context switches occur

Maintained by sorting ready queues and recalculating remaining times efficiently.

6.2 Gantt Chart Construction

Challenge: Representing executions with interruptions.
Solution: Store segments as tuples `(process, start, end)` and render them cleanly.

6.3 MLFQ Queue Management

MLFQ requires:

- Demotion
- Promotion (aging)
- Multiple dynamic queues
  Solution: Implemented separate queues with timers and periodic checks.

7. Conclusion

This CPU Scheduling Simulator successfully demonstrates how different scheduling algorithms behave under various workloads. It provides insights into:

- Fairness vs. efficiency
- Preemptive vs. non-preemptive behavior
- Importance of quantum size
- Handling starvation via aging

The project serves as an excellent educational tool for understanding real-world OS scheduling policies.