

CSE513 Assignment 1

CPU Scheduling Simulator

Objective

Develop a browser-based simulator that demonstrates how CPU scheduling algorithms select processes and how their decisions affect performance. Your work should show both a correct implementation and your understanding of the scheduling decisions.

1 Scheduling algorithms

Implement the following algorithms:

- First-Come, First-Served (FCFS)
- Shortest Job First (SJF) — non-preemptive
- Shortest Remaining Time First (SRTF) — preemptive
- Round Robin (RR) — with a configurable time quantum

Use the following model:

- One CPU.
- Each process has one CPU burst and performs no I/O.
- Context switching takes zero time.
- Arrival times are nonnegative integers.
- CPU bursts and the round-robin quantum are positive integers.
- Process IDs must be unique.
- The CPU must run an eligible process whenever the ready queue is nonempty.

Reject invalid inputs with a clear explanation.

2 Scheduling rules

Use these rules so that results are consistent across submissions:

- FCFS: Select the earliest-arriving ready process.
- SJF: Whenever the CPU becomes free, select the ready process with the shortest CPU burst.
- SRTF: Select the ready process with the shortest remaining CPU time. Preempt the running process only when another process has a strictly shorter remaining time. On an equal remaining time, keep the current process running.
- Tie-breaking: When selecting between otherwise tied processes, prefer the earlier arrival time, then the earlier position in the input list.
- Round robin: Maintain a FIFO ready queue. Arrivals join the tail. At a quantum boundary, enqueue processes arriving at that exact time before re-enqueuing the unfinished running process. A completed process never rejoins the queue.
- If the ready queue is empty, show an idle interval until the next arrival.

Use intervals consistently: a process running from time 2 to time 5 receives three units of CPU time.

3 Website requirements

The website must let a user:

- Add, edit and remove processes.
- Enter each process's ID, arrival time and CPU burst.
- Select a scheduling algorithm.
- Set the round-robin time quantum.
- Run the simulation and inspect its results.
- Load the supplied example workload.

Display a labeled Gantt chart, including CPU idle intervals. Provide a results table containing each process's arrival time, CPU burst, first start time, completion time, turnaround time, waiting time and response time. Also display average turnaround, waiting and response times.

Calculate the metrics as follows:

Turnaround time = Completion time – Arrival time

Waiting time = Turnaround time – CPU burst

Response time = First start time – Arrival time

For round robin, make quantum boundaries identifiable even when the same process receives consecutive time slices.

A simple, readable interface is sufficient. A backend, database, login system and paid hosting are unnecessary.

4 Required workloads

Simulate both workloads using all four algorithms. Use a quantum of 2 for round robin.

Workload A

Process	Arrival time	CPU burst
P1	0	8
P2	1	4
P3	2	2
P4	4	1

Workload B

Process	Arrival time	CPU burst
P1	2	3
P2	2	1
P3	8	2
P4	9	1

Before using your simulator to obtain the answers, calculate the timelines and metrics by hand. Include your calculations in the report and compare them with the simulator's output. Explain and resolve any discrepancies.

5 Your own test cases

Create at least three additional workloads:

- One containing CPU idle time.
- One containing simultaneous arrivals and a scheduling tie.
- One where SRTF preempts a running process.

For each test, state what behavior it checks and provide the expected result. Screenshots alone are not sufficient evidence of correctness.

Also verify these properties:

- Every process receives exactly its specified CPU burst.
- No process runs before its arrival.
- Processes do not overlap on the single CPU.
- Waiting and response times are nonnegative.
- The CPU is never idle while a process is ready.

6 Analysis

In a short report, answer the following:

1. For Workload A, which algorithm gives the lowest average waiting time? Which gives the lowest average response time? Explain the scheduling decisions behind the results.
2. How do the round-robin results change when the quantum is 1, 2 and 4? Explain why a smaller quantum does not necessarily improve every metric.
3. Under what conditions do SJF and SRTF produce the same schedule? Give an example.
4. Why must a real operating system estimate future CPU bursts, while this simulator can use exact values?
5. Can starvation occur under SJF or SRTF if new short jobs keep arriving? Explain how this differs from the finite workloads used here.
6. Describe one implementation mistake you encountered and how a test exposed it.

This simulator models scheduling policies. It does not reproduce Ubuntu's actual scheduler.

7 AI assistance

AI assistance is permitted for implementation, debugging and explanation. You remain responsible for the correctness of everything submitted.

Include a brief account of:

- Which tools you used and for what purposes.
- How you checked generated code or explanations.
- One suggestion you corrected or rejected, if applicable.

You must be able to explain your implementation and trace its decisions for an unfamiliar workload.

8 What to submit to Teams

Upload the following to the assignment in Microsoft Teams:

- A PDF report with the working website address clearly shown at the top of the first page, immediately below the assignment heading. Make the address clickable and check that it opens without requiring access permission.
- The complete source code as a ZIP archive, or an accessible repository link included in the PDF report.
- A README included with the source code explaining how to run the simulator locally, its dependencies and any limitations.
- In the PDF report, include at least two worked examples for each algorithm: FCFS, SJF, SRTF and round robin (at least eight algorithm–workload examples in total). Use Workloads A and B for these examples. For each example, show the input data, hand-calculated timeline and metrics, simulator output, and a comparison explaining any discrepancy.

The PDF report must also include the three additional student-designed test cases, answers to the analysis questions, and the AI-use statement. Submit the report as a PDF, not as a Word document or a collection of screenshots. Include a locally runnable version of the simulator with the source submission. Do not publish student numbers or other private information on the website.

Be prepared to demonstrate the simulator and explain how changing an arrival time, burst or quantum changes the result.

Submission channel: Microsoft Teams. Deadline: To be announced.

9 Assessment

Criterion	Weight
Algorithm correctness and scheduling-rule compliance	50%
Tests, hand calculations and verification	20%
Explanation and comparative analysis	20%
Interface usability and clear results	10%

Correct reasoning and reproducible results matter more than visual polish.