CPU Scheduling Algorithms: FCFS, HRRN, SJF

Khoa Vu

April 28, 2025

Abstract

This project implements and evaluates three CPU scheduling algorithms:

- First Come First Serve (FCFS)
- Highest Response Ratio Next (HRRN)
- Shortest Job First (SJF)

Each algorithm was simulated using Python, and the performance was measured across various workloads.

1 Introduction

CPU scheduling is a fundamental function of the operating system that manages how processes are assigned to the CPU. It is important because with efficient scheduling, it will improve the system's responsiveness and resource utilization. Different algorithms optimize different aspects of performance, such as waiting time, turnaround time, or fairness. This project focuses on implementing and analyzing three scheduling algorithms: First Come First Serve (FCFS), Highest Response Ratio Next (HRRN), and Shortest Job First (SJB). Each algorithm was tested on various workloads to evaluate its effectiveness and compare its performance under different conditions.

2 Algorithms Implemented

- **First Come First Serve (FCFS)**: Executes processes in the order they arrive.
- **Highest Response Ratio Next (HRRN)**: Selects processes with the highest response ratio.
- **Shortest Job First (SJF)**: Selects the process with the smallest burst time.
- **Shortest Remaining Time First (SRTF)**: Preemptive version of SJF; it always selects the process with the smallest remaining burst time.

3 Implementation Details

The project was implemented in Python. A `Process` class was created to store process attributes such as ID, arrival time, burst time, and execution statistics. Simulations for each scheduling algorithm collected metrics like average waiting time, turnaround time, CPU utilization, and throughput. Tables and graphs were generated using `tabulate` and `matplotlib` libraries.

4 Testing and Results

4.1 Small Manual Test

Algorithm	Avg Wait Time	Avg Turnaround Time	CPU Utilization (%)	Throughput (processes/unit time)
FCFS	4.3333	10	94.4444	0.1667
HRRN	4.3333	10	94.4444	0.1667
SJF	4.3333	10	94.4444	0.1667
SRTF	4.3333	10	94.4444	0.1667

Table 1: Comparison of Scheduling Algorithms With 3 Processes

4.2 Large Scale Random Test

Algorithm	Avg Wait Time	Avg Turnaround Time	CPU Utilization (%)	Throughput (processes/unit time)
FCFS	126.8	132.22	100	0.184502
HRRN	88.5	93.92	100	0.184502
SJF	87.56	92.98	100	0.184502
SRTF	87.56	92.98	100	0.184502

Table 2: Comparison of Scheduling Algorithms With 50 Random Processes

4.3 Edge Case Test 1: All Processes Arrival Time 0 With 0-10 Burst Time

Algorithm	Avg Wait Time	Avg Turnaround Time	CPU Utilization (%)	Throughput (processes/unit time)
FCFS	143.10	148.82	100	0.174825
HRRN	107.36	113.08	100	0.174825
SJF	104.06	109.78	100	0.174825
SRTF	104.06	109.78	100	0.174825

Table 3: Comparison of Scheduling Algorithms With All Processes Having Arrival Time 0 and Mixed Burst Times

4.4 Edge Case Test 2: Extremely Long and Short Burst Times

Algorithm	Avg Wait Time	Avg Turnaround Time	CPU Utilization (%)	Throughput (processes/unit time)
FCFS	546.44	568.14	100	0.046829
HRRN	320.56	342.26	100	0.046829
SJF	320.56	342.26	100	0.046829
SRTF	320.56	342.26	100	0.046829

Table 4: Comparison of Scheduling Algorithms With Extremely Long and Short Burst Times

5 Performance Comparison

We compared the four implemented scheduling algorithms (FCFS, HRRN, SJF, SRTF) under various scenarios, using metrics such as Average Waiting Time, Average Turnaround Time, CPU Utilization, and Throughput. The following graphs summarize the performance:

5.1 Small Manual Test (3 Processes)

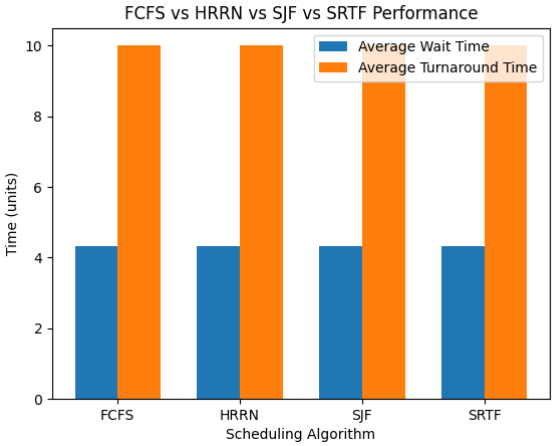

Figure 1: Performance Comparison with 3 Processes

5.2 Large Scale Random Test (50 Processes)

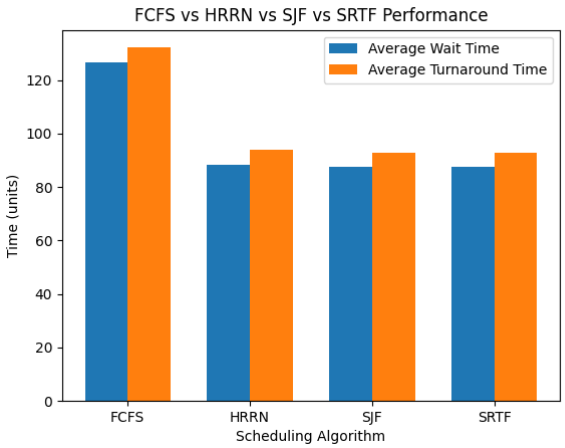

Figure 2: Performance Comparison with 50 Random Processes

5.3 Edge Case: All Processes Arrive at Time 0

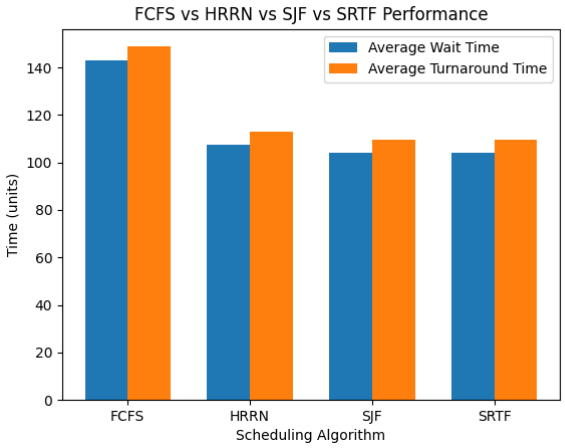

Figure 3: Performance Comparison with All Processes Having Arrival Time 0

5.4 Edge Case: Extremely Long and Short Burst Times

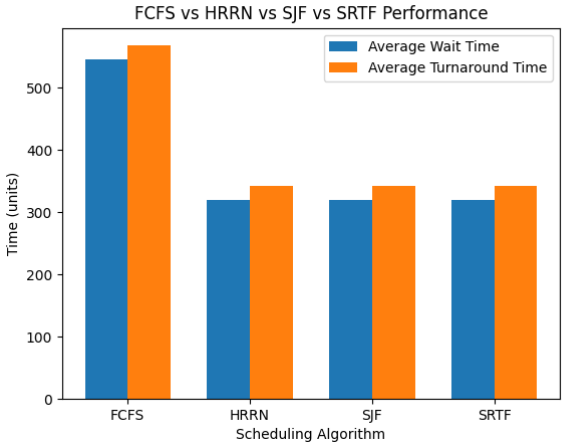

Figure 4: Performance Comparison with Mixed Long and Short Burst Times

6 Discussion

Across all tests, Highest Response Ratio Next (HRRN) and Shortest Job First (SJF) consistently outperformed the First Come First Serve (FCFS) algorithm in terms of average waiting time and average turnaround time. The main reason why this happens is that FCFS generally suffers because it strictly follows arrival order without considering burst times, leading to longer wait times when shorter processes are stuck behind longer ones.

HRRN improves the performance by prioritizing processes with high waiting times relative to their burst times, balancing the fairness and efficiency. SJF and SRTF achieve even lower waiting times by always selecting the shortest available job, minimizing overall waiting and turnaround times, though SJF requires accurate knowledge of burst times.

In the edge cases with identical arrival times or mixed burst times, SJF and SRTF showed strong performance, confirming that the dynamic job selection is crucial under the load variations.

7 Conclusion

In this project, I implemented and tested four CPU scheduling algorithms: FCFS, HRRN, SJF, and SRTF. After running each algorithm with small, random, and edge cases, the results showed that SJF and SRTF performed the best when it came to keeping the waiting time and turnaround time low. FCFS had the worst performance because it doesn't prioritize short jobs and just follows the arrival order.

HRRN was alright, it didn't perform the best out of all four, but it's not hard to implement since it didn't need to know the burst time ahead of time like SJF, but still better compared to FCFS. I would say the harder parts of this project were making sure SRTF worked correctly with constantly changing ready queues and remaining times.

Overall, this project showed how much of a difference the right scheduling algorithm can make depending on the situation. Picking the right one depends on the situation, do you value speed more, or fairness more, or simply keeping the system simple.