Princess Sumaya University for Technology جامعـــة الأميــرة سميّــة للتكنولوجيا

CPU Scheduling Algorithms: FCFS and SRTF

Project Title: Comparison of CPU Scheduling Algorithms: FCFS and SRTF
Student Name: Ahmd Emad
Student Number: 20221059
Student Email: ahm20221059@std.psut.edu.jo

Contents

Introduction

This report examines and compares two CPU scheduling algorithms, First-Come First-Served (FCFS) and Shortest-Remaining-Time-First (SRTF), which are fundamental to the performance optimization of operating systems. The objective is to calculate and analyze the average turnaround time and average waiting time for each algorithm using a Java implementation.

Methodology (Algorithms)

First-Come First-Served (FCFS)

The FCFS algorithm schedules processes in the order they arrive in the ready queue. It is non-preemptive, meaning a process runs to completion once it starts.

Pseudocode:

1. Sort processes by arrival time.

2. Execute each process in arrival order.

3. Calculate completion, turnaround, and waiting times for each process.

Shortest-Remaining-Time-First (SRTF)

The SRTF algorithm selects the process with the shortest remaining burst time for execution. It is preemptive, allowing a shorter process to preempt the currently running process.

Pseudocode:

1. Continuously monitor the ready queue for the process with the shortest remaining time.

2. Preempt the running process if a shorter process arrives.

3. Update completion, turnaround, and waiting times as processes finish.

Results

Table Results for FCFS:

Process ID	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
1	0	7	7	7	0
2	2	4	11	9	5
3	4	1	12	8	7
4	5	4	16	11	7

Table Results for SRTF:

Process ID	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
1	0	7	16	16	9
2	2	4	7	5	1
3	4	1	5	1	0
4	5	4	11	6	2

Figures

Bar charts comparing **average turnaround time** and **average waiting time** for FCFS and SRTF:
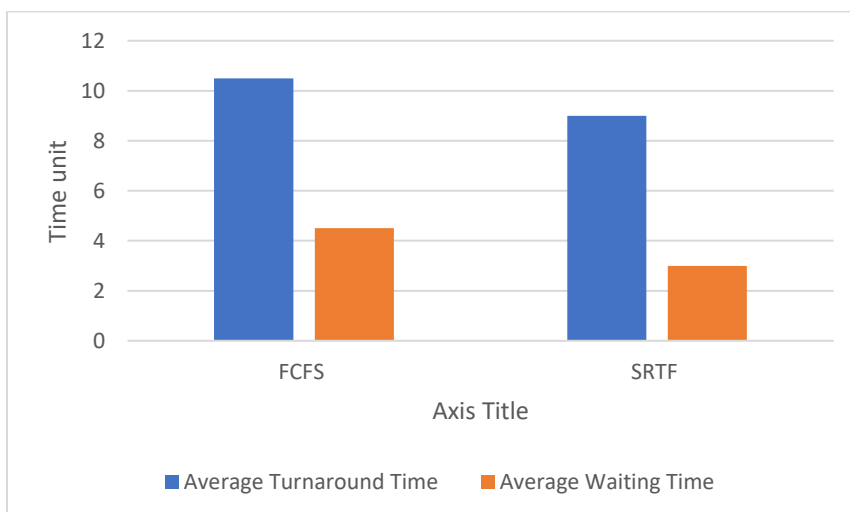
Comparative Analysis

The SRTF algorithm outperforms FCFS in terms of reducing average waiting time and turnaround time. However, it is more complex to implement and may cause starvation for longer processes.

Conclusion

In conclusion, the comparison between FCFS (First-Come, First-Served) and SRTF (Shortest Remaining Time First) highlights the inherent trade-offs between simplicity and efficiency in process scheduling. FCFS, being one of the simplest scheduling algorithms, operates on a first-come, first-served basis, ensuring fairness. However, it suffers from significant inefficiencies when handling workloads with varying burst times, leading to longer average waiting times for processes, particularly those that arrive after processes with long burst times.

On the other hand, SRTF, a preemptive version of the Shortest Job First (SJF) algorithm, is more efficient in reducing waiting times and achieving better throughput, especially in scenarios where burst times are highly variable. By always choosing the process with the shortest remaining burst time, SRTF minimizes waiting times and makes better use of system resources. However, it introduces additional computational overhead due to the need for constant monitoring of process burst times and handling preemptions. This can lead to complexities in implementation and increased system load, particularly in environments with frequent process arrivals and departures.

Additionally, SRTF can introduce issues such as starvation, where processes with longer burst times may be perpetually preempted by shorter processes, leading to their indefinite delay. This is a notable drawback, particularly in real-time systems or in scenarios where fairness and process prioritization are crucial.

Overall, the choice between FCFS and SRTF depends on the specific requirements of the system. FCFS is more suitable for systems where simplicity and fairness are prioritized, and the workload is relatively uniform. SRTF, while more efficient, is better suited for environments where minimizing waiting times is a higher priority, although it requires careful management to avoid starvation.

Future work in process scheduling could involve the exploration and analysis of other algorithms, such as Priority Scheduling, which assigns priorities to processes based on their importance or urgency, or Round Robin, which allocates fixed time slices to each process, ensuring fairness and avoiding starvation. These algorithms could offer a more balanced approach to process scheduling, combining the strengths of FCFS and SRTF while mitigating their respective drawbacks. Further research could also investigate hybrid algorithms that dynamically adjust scheduling strategies based on workload characteristics, optimizing performance in diverse system environments

---

References

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2021). *Operating System Concepts*. Wiley.