Report on Scheduling Algorithms Simulation

Introduction:

This report presents the results and observations from the implementation of various process scheduling algorithms - First Come First Serve (FCFS), Shortest Process Next (SPN), Preemptive Priority (PP) and Priority Round Robin (PRR). The purpose of this simulation was to analyze the performance of these algorithms in terms of turnaround time, waiting time, and overall efficiency.

Testing and Implementation:

Java programming language was used to implement the scheduling algorithms. The simulation took input data containing process details, such as process ID, arrival time, service time, and priority. The algorithms were executed for the given data sets and the results were printed to the console.

The simulation involved handling dispatchers and managing the ready queue based on the characteristics of each algorithm. The goal was to mimic real-world CPU scheduling behaviors and observe how different algorithms impact process execution.

Algorithm Performance:

1. FCFS and SPN: As expected, FCFS and SPN are non-preemptive algorithms that do not consider process priorities. FCFS executes processes in the order of arrival, while SPN executes the shortest process first. These algorithms are simple to implement but can lead to longer waiting times for processes with larger service times.

2. PP: The preemptive priority algorithm considers process priorities and allows higher priority processes to interrupt lower priority ones. This can lead to better responsiveness for critical processes. However, if not properly managed, low-priority processes might suffer from starvation.

3. PRR: Priority Round Robin is a variant of the standard Round Robin algorithm. It assigns different time quanta based on priority. Higher priority processes receive more time, improving their responsiveness. Lower priority processes might experience time starvation due to the shorter time quantum.

Observations:

1. Priority-based algorithms (PP, PRR) tend to provide better response times for high-priority processes, ensuring critical tasks are completed quickly.

2. FCFS and SPN might result in longer waiting times for certain processes, especially if early-arriving processes have long service times.

3. Preemptive algorithms introduce context-switching overhead, impacting overall efficiency. Careful tuning of time quanta is necessary to balance responsiveness and efficiency.

Conclusion:

The simulation of scheduling algorithms provides valuable insights into their behaviors and performance characteristics. The choice of algorithm depends on the specific requirements of the system and the nature of the processes being managed. Preemptive algorithms improve responsiveness at the cost of overhead, while non-preemptive ones are simpler but might lead to suboptimal resource utilization.

Testing Details:

The simulation was tested with provided sample data sets and additional data sets to ensure robustness and correctness. Inputs were systematically designed to cover various scenarios, such as processes with varying service times and priorities. The outputs were cross-checked with expected values to validate algorithm accuracy.