CPU Scheduling Algorithm Analysis

Your Name

April 24, 2025

1 Best Algorithm for CPU-bound Processes

Based on the simulation results, **SRT (Shortest Remaining Time)** is the best algorithm for CPU-bound processes. The data shows that SRT provides the lowest average turnaround time for CPU-bound processes at 1,181ms, significantly outperforming FCFS (51,829ms), SJF (51,831ms), and RR (52,691ms).

SRT also demonstrates the best CPU utilization at 73.959%, which is higher than both FCFS and SJF (68.121%) and slightly better than RR (73.088%). This higher utilization indicates that the CPU spends more time executing processes rather than being idle.

While SRT does introduce more context switches for CPU-bound processes (35 compared to 25 in FCFS/SJF), this overhead is justified by the dramatic improvement in turnaround time. The preemptive nature of SRT allows it to respond quickly to the arrival of shorter processes, preventing long CPU bursts from monopolizing the processor.

2 Optimal α Value for SJF and SRT

The simulation used $\alpha = 0.75$ for both SJF and SRT algorithms. This value appears to produce good results, particularly for SRT. We can observe the effectiveness of this α value by examining how tau values are recalculated throughout the simulation.

For example, with process A1 (an I/O-bound process with shorter bursts), we see:

- Initial tau: 1000ms
- After first burst (971ms): tau recalculated to 979ms
- After second burst (129ms): tau recalculated to 342ms
- After third burst (188ms): tau recalculated to 227ms

This demonstrates how the exponential averaging with $\alpha = 0.75$ effectively adapts to the actual burst patterns, giving more weight to recent observations while still considering historical data.

Compared to a hypothetical baseline without exponential averaging ($\alpha = -1$), the chosen α value allows the scheduler to make more informed decisions about which process to run next, particularly benefiting the SRT algorithm where accurate prediction of remaining time is crucial for preemption decisions.

3 RR Time Slice Heuristic

The simulation results do not confirm the heuristic that RR performs best when 80% of processes complete their CPU bursts within one time slice. According to the output:

- CPU-bound processes: 12% of bursts completed within one time slice
- I/O-bound processes: 70% of bursts completed within one time slice
- Overall: 37.778% of bursts completed within one time slice

This is significantly below the 80% threshold suggested by the heuristic. The time slice used (256ms) appears too small for the CPU-bound process A0, which has many long bursts (average over 1,500ms), resulting in frequent preemptions. This is evidenced by the high number of preemptions for CPU-bound processes (13).

The RR algorithm shows the worst wait time for CPU-bound processes (791,557ms) despite having reasonable turnaround times. This suggests that while processes are making progress, they spend excessive time waiting due to the round-robin scheduling.

A larger time slice would likely improve performance for CPU-bound processes by reducing the number of preemptions and context switches, though it might increase response time for I/O-bound processes.