COEN 383: Advance Operating Systems

Project 2 Scheduling Algorithms
Fall 2023 Mon-Wed 7:10 am Group 8

Akaash Meghraj Trivedi (W1650955)
Anagha Viswanath (W1651360)
Michael Zarrabi (W1479183)
Nakul Pravin Thombare (W1650130)
Sai Sanjana Reddy Vatte (W1650167)

Objective:

We have implemented the following process scheduling algorithms using C .

- First come first-served (FCFS) [non-preemptive]
- Shortest job first (SJF) [non-preemptive]
- Shortest remaining time (SRT) [preemptive]
- Round robin (RR) [preemptive]
- Highest priority first (HPF) [both non-preemptive and preemptive]

The average of the 5 runs of all algorithms is as follows:

FIRST COME FIRST SERVE:

Average Response Time(RT) : 36.3
Average Wait Time(WT) : 36.7
Average Turn Around Time(TAT) :41.6
Average throughput(tr) :19.0

ROUND ROBIN PREEMPTIVE:

Average Response Time(RT) : 31.1

Average Wait Time(WT) : 63.9
Average Turn Around Time(TAT) :68.6
Average throughput(tr) :27.0

SHORTEST JOB FIRST NON PREEMPTIVE:

Average Response Time(RT) : 6.1
Average Wait Time(WT) : 6.6
Average Turn Around Time(TAT) :9.5
Average throughput(tr) :30.0

SHORTEST REMAINING TIME FIRST PREEMPTIVE:

Average Response Time(RT) : 4.7
Average Wait Time(WT) : 6.1
Average Turn Around Time(TAT) :9.0
Average throughput(tr) :30.0

HIGHEST PRIORITY FIRST PREEMPTIVE:

Average Response Time(RT) : 3.8
Average Wait Time(WT) : 5.9
Average Turn Around Time(TAT) :9.2
Average throughput(tr) :50.0

HIGHEST PRIORITY FIRST NON PREEMPTIVE:

Average Response Time(RT) : 10.7
Average Wait Time(WT) : 11.0
Average Turn Around Time(TAT) :15.1
Average throughput(tr) :18.0

Observations

First Come First Serve (FCFS):

In the First Come First Serve (FCFS) scheduling algorithm, processes are executed in the order they arrive. This approach continues to demonstrate relatively high average wait times, turnaround times, and response times. The processes that arrive first are given precedence, potentially causing delays for later-arriving tasks. The average throughput remains relatively low, indicating that FCFS may not be the best choice

when it comes to maximizing resource usage. It's worth noting that FCFS doesn't consider the execution time or priority of processes, which can lead to inefficiencies in certain scenarios.

Round Robin (RR) Preemptive:

Round Robin (RR) Preemptive scheduling still provides better response times compared to FCFS. However, it tends to have higher average wait times and turnaround times. This is because processes may not complete within a single time quantum, causing longer waiting periods for processes with longer execution times. The average throughput is improved compared to FCFS, thanks to the preemptive nature of the algorithm, which ensures that all processes receive some CPU time.

Shortest Job First (SJF) Non-Preemptive:

The SJF Non-Preemptive scheduling algorithm continues to exhibit low average wait times and turnaround times. It excels in providing quick responses to shorter tasks, making it an efficient choice when minimizing waiting periods is crucial. The average throughput remains high, indicating efficient resource utilization. However, it still faces potential issues if long-running processes arrive first, which could lead to some processes experiencing delays.

Shortest Remaining Time First (SRT) Preemptive:

SRT Preemptive scheduling still prioritizes shorter jobs efficiently. It offers low average response times, wait times, and turnaround times, making it a strong choice for scenarios where minimizing these metrics is essential. The high average throughput reflects the effectiveness of this algorithm in balancing the execution of short and long tasks.

Highest Priority First (HPF) Preemptive:

HPF Preemptive scheduling continues to provide the best response times and throughput among all algorithms. High-priority tasks are executed promptly, resulting in low response times. However, it maintains higher average wait times and turnaround times for lower-priority tasks. The risk of lower-priority processes facing starvation remains, but the high throughput suggests that this algorithm efficiently utilizes available resources.

Highest Priority First (HPF) Non-Preemptive:

In its non-preemptive form, HPF still offers relatively low response times. However, wait times and turnaround times are higher compared to its preemptive counterpart. Lower-priority tasks still experience some delays, which can lead to increased wait times. The average throughput is lower than HPF Preemptive, but it still reflects efficient resource utilization.

Conclusion

In conclusion, The scheduling algorithm with the best throughput is typically the Highest Priority First (HPF) Preemptive method. This algorithm ensures that high-priority tasks are executed promptly, which can lead to efficient resource utilization and a higher throughput rate.