COEN 383: Advance Operating Systems

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

Group 5

Bharti Prakash W1652174

Iniyan Chandran Ramachandran W1651510

Ruthu Rajendra W1653722

Siddhi Sanjay Powar W1650216

Vaibhav Sachdeva W1650084

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:

ALGORITHM: FIRST COME FIRST SERVE:

Average Response Time(RT) : 18.1

Average Wait Time(WT) : 18.5

Average Turn Around Time(TAT) :23.1

Average Throughput(Tr) :26.0

ALGORITHM: ROUND ROBIN PREEMPTIVE:

Average Response Time(RT) : 1.5

Average Wait Time(WT) : 24.0

Average Turn Around Time(TAT) :28.6

Average Throughput(Tr) :24.6

ALGORITHM: SHORTEST JOB FIRST NON PREEMPTIVE:

Average Response Time(RT) : 11.1

Average Wait Time(WT) : 11.5

Average Turn Around Time(TAT) :16.1

Average Throughput(Tr) :26.0

ALGORITHM: SHORTEST REMAINING TIME FIRST PREEMPTIVE:

Average Response Time(RT) : 7.1

Average Wait Time(WT) : 13.0

Average Turn Around Time(TAT) :17.6

Average Throughput(Tr) :26.0

ALGORITHM: HIGHEST PRIORITY FIRST PREEMPTIVE:

Average Response Time(RT) : 57.5

Average Wait Time(WT) : 86.3

Average Turn Around Time(TAT) :91.8

Average Throughput(Tr) :130.2

ALGORITHM: HIGHEST PRIORITY FIRST NON PREEMPTIVE:

Average Response Time(RT) : 16.6

Average Wait Time(WT) : 17.1

Average Turn Around Time(TAT) :21.2

Average Throughput(Tr) :26.0

OBSERVATION:

First Come First Serve (FCFS):
Observations:

- Processes in FCFS are executed in the order they arrive, leading to high average wait, turnaround, and response times.
- The lack of consideration for execution time or priority may result in inefficiencies, especially in scenarios where shorter tasks arrive later.
- Precedence is given to the processes that arrive first, potentially causing delays for later-arriving, shorter tasks.
- The average throughput remains relatively low, indicating suboptimal resource usage as tasks are processed in a rigid arrival order.
  FCFS may not be the best choice for scenarios where maximizing resource usage and minimizing wait times are crucial.

Round Robin Pre-emptive:
Observations:

- Round Robin provides fairness in CPU time distribution but may lead to higher average wait and turnaround times.

- The constant time slices may not be optimal for all types of tasks, impacting overall efficiency. Quick initial responses are achieved with low response times, but the approach may not prioritize tasks effectively.

- Throughput is moderate, suggesting a balance between fairness and efficiency.

Shortest Job First Non Pre-emptive (SJF):
Observations:

- SJF non-pre-emptive excels in prioritizing shorter jobs, resulting in lower average response, wait, and turnaround times However, it may suffer from "starvation" if long jobs consistently arrive, impacting overall fairness.

- The throughput is decent, indicating effective prioritization of shorter tasks.

Shortest Remaining Time First Pre-emptive (SRTF):
Observations:

- SRTF pre-emptive minimizes response times by pre-emptively choosing the shortest remaining job, providing a good balance.

- It may involve more context switches due to its pre-emptive nature, affecting overall system overhead.

- Throughput is comparable to other algorithms, with a focus on minimizing response and turnaround times.

Highest Priority First Pre-emptive (HPF):
Observations:

- HPF pre-emptive has high response and wait times, indicating delays for lower-priority tasks. The high throughput suggests a focus on completing high-priority tasks but may neglect lower-priority tasks for extended periods.

- Priority inversion and starvation issues may arise, impacting the overall efficiency.

Highest Priority First Non Pre-emptive (HPF):
Observations:

- HPF non-pre-emptive performs better in terms of response, wait, and turnaround times compared to its pre-emptive counterpart.

- It ensures that high-priority tasks are completed before lower-priority ones, but it may lead to potential starvation for lower-priority tasks.
  The throughput is moderate, reflecting a balance between priority-based scheduling and overall efficiency

CONCLUSION

In conclusion, each scheduling algorithm has its trade-offs. FCFS is simple but inefficient, Round Robin is fair but may lead to higher wait times, SJF prioritizes shorter tasks but risks starvation, and HPF, while effective for high-priority tasks, may cause delays for lower-priority ones. The choice depends on system priorities and characteristics