Process Scheduling Algorithms Report

Group Members:

1. Aifaz Maknojia
2. Samarth Kulkarni
3. Samuel Karumanchi
4. Sashidhar Kancharla
5. Vedant Raj Kavi

Introduction:

This project aims to implement and analyse six different process scheduling algorithms: First-Come-First-Served (FCFS), Shortest Job First (SJF), Shortest Remaining Time (SRT), Round Robin (RR), Highest Priority First (HPF) Non-Pre-emptive, and Highest Priority First (HPF) Pre-emptive. Each algorithm was run for 5 times with a fixed workload, simulating processes with randomly generated arrival times, run times, and priorities.

The goal of this report is to compare these algorithms based on key metrics, including Average Response Time (ART),Average Wait Time (AWT), Average Turnaround Time (TAT), and Throughput. These metrics are averaged over 5 runs for each algorithm.

Results:

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

ALGORITHM: FIRST COME FIRST SERVE:

Average Response Time (ART) : 33.3
Average Wait Time (AWT) : 33.8
Average Turn Around Time (TAT) :38.3
Average Throughput :20.0

ALGORITHM: ROUND ROBIN PREEMPTIVE:

Average Response Time (ART) : 30.1
Average Wait Time (AWT) : 53.0
Average Turn Around Time (TAT) :57.8
Average Throughput :24.0

ALGORITHM: SHORTEST JOB FIRST NON PREEMPTIVE:

Average Response Time (ART) : 7.5
Average Wait Time (AWT) : 8.0
Average Turn Around Time (TAT) :11.2
Average Throughput :28.0

ALGORITHM: SHORTEST REMAINING TIME FIRST PREEMPTIVE:

Average Response Time (ART) : 6.6
Average Wait Time (AWT) : 7.7
Average Turn Around Time (TAT) :10.9
Average Throughput :28.0

ALGORITHM: HIGHEST PRIORITY FIRST PREEMPTIVE:

Average Response Time (ART) : 3.7
Average Wait Time (AWT) : 5.6
Average Turn Around Time (TAT) :8.1
Average Throughput :53.0

ALGORITHM: HIGHEST PRIORITY FIRST NON PREEMPTIVE:

Average Response Time (ART) : 9.6
Average Wait Time (AWT) : 9.9
Average Turn Around Time (TAT) :12.1
Average Throughput :20.0
```

Analysis:

1. **First-Come-First-Served (FCFS):**

- **ART: 33.3 | AWT: 33.8 | TAT: 38.3 | Throughput: 20.0**

- **Observations**: The FCFS algorithm, while simple, results in high wait and response times due to its non-pre-emptive nature. Processes arriving earlier are favoured, leading to significant delays for longer processes that arrive later.

- **Use Case:** Suitable for batch processing where fairness is important and there is no requirement for prioritizing smaller or more urgent tasks.

2. Round Robin (RR):

- **ART: 30.1 | AWT: 53.0 | TAT: 57.8 | Throughput: 24.0**

- **Observations:** Round Robin shows a higher wait time and turnaround time compared to FCFS, despite lower response times. The frequent context switching and time-slicing allow better CPU utilization, but result in higher overhead.

- **Use Case:** Ideal for time-sharing systems where fairness and responsiveness are crucial. It is a balanced approach but not the most efficient for throughput.

3. Shortest Job First (SJF) Non-Pre-emptive:

- ART: 7.5 | AWT: 8.0 | TAT: 11.2 | Throughput: 28.0

- Observations: SJF non-pre-emptive performs much better than FCFS in terms of ART, AWT, and TAT. This is because shorter processes are executed first, minimizing the time a process spends waiting.

- Use Case: Suitable for systems where shorter processes are more critical, such as interactive systems. It is prone to starvation, as longer processes may never get a chance to execute.

4. Shortest Remaining Time (SRT) Pre-emptive:

- ART: 6.6 | AWT: 7.7 | TAT: 10.9 | Throughput: 28.0

- Observations: SRT improves upon SJF by pre-empting longer processes in Favor of shorter remaining tasks. This results in slightly better ART, AWT, and TAT than non-pre-emptive SJF.

- Use Case: Suitable for real-time systems where responsiveness to short tasks is critical. However, it may introduce frequent context switches, increasing CPU overhead.

5. Highest Priority First (HPF) Pre-emptive:

- ART: 3.7 | AWT: 5.6 | TAT: 8.1 | Throughput: 53.0

- Observations: HPF pre-emptive is the most efficient in terms of all metrics. By prioritizing the highest-priority processes, it ensures that important tasks are completed first, resulting in a high throughput and low ART and AWT.

- Use Case: Ideal for systems with strict prioritization requirements, such as critical real-time systems. It may, however, cause starvation of low-priority processes.

6. Highest Priority First (HPF) Non-Pre-emptive:

- ART: 9.6 | AWT: 9.9 | TAT: 12.1 | Throughput: 20.0

- Observations: HPF non-pre-emptive still prioritizes processes, but without pre-emption, lower-priority processes are allowed to finish once started. This increases response time and reduces throughput compared to the pre-emptive version.

- Use Case: Suitable for priority-based systems where pre-emption is not desired or necessary, but prioritization is still important.

Conclusion:

- Best Algorithm for Response Time (ART):HPF Pre-emptive outperforms all other algorithms, followed by SRT and SJF.

- Best Algorithm for Wait Time (AWT): HPF Pre-emptive achieves the lowest average wait time, making it the most efficient.

- Best Algorithm for Turnaround Time (TAT): HPF Pre-emptive and SRT show the lowest turnaround times.

- Best Algorithm for Throughput: HPF Pre-emptive has the highest throughput, processing the most jobs in a given time.

Summary:

From this analysis, Highest Priority First Pre-emptive (HPF) is the best-performing algorithm overall, achieving the best results across all metrics. However, if system fairness is a priority, Round Robin or FCFS might be considered for time-sharing environments. Shortest Job First and Shortest Remaining Time are effective when minimizing response time for short processes, but may cause starvation for longer tasks.