COEN383
Advanced Operating System
Project 2 Report

Group 3
(Sruthi Thiyagarajan, Mitali Sahoo, Bharat Chadalawada)

Objective : This project gives us experience with process scheduling algorithms.
We have written a C program that performs runs of the following process scheduling

Algorithms.

- 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]

Constraints: 100 quanta, Each process has a random arrival time, a priority and expected run time. There exists only one process queue. There is no I/O time. The highest priority first uses 4 queues. Round Robin uses 1 quanta time slice.

The following is the final statistical output obtained from the 6 algorithms implemented. The average of the 5 runs of all algorithms is as follows:

```
First Come First Serve:
Average Response Time(RT) : 26.3
Average Wait Time(WT) : 26.7
Average Turnaround Time(TAT) :31.0
Average throughput(tr) :21.0

Shortest Job First Non-Preemptive:
Average Response Time(RT) : 4.7
Average Wait Time(WT) : 5.1
Average Turnaround Time(TAT) :8.0
Average throughput(tr) :29.0
```

Shortest Remaining Time First Preemptive:

```
Average Response Time(RT) : 2.7
Average Wait Time(WT) : 4.3
Average Turnaround Time(TAT) :7.1
Average throughput(tr) :28.0
```

Round Robin Preemptive:

```
Average Response Time(RT) : 22.7
Average Wait Time(WT) : 48.0
Average Turnaround Time(TAT) :52.6
Average throughput(tr) :25.0
```

Highest Priority First Preemptive:

```
Average Response Time(RT) : 2.2
Average Wait Time(WT) : 5.2
Average Turnaround Time(TAT) :7.8
Average throughput(tr) :53.0
```

Highest Priority First Non-Preemptive:

```
Average Response Time(RT) : 7.2
Average Wait Time(WT) : 7.6
Average Turnaround Time(TAT) :10.4
Average throughput(tr) :22.0
```

The following observations were made from the results of running the 6 algorithms:

1. First Come First Serve Algorithm:

The response time, Wait time and turnaround time are very high and as a result, the throughput is the lowest for this algorithm. Since the new processes have to wait until the earlier process completes execution, there are chances of starvation. However FCFS is easy to implement.

2. Shortest Job First Non Preemptive:

The response time, Wait time and turnaround time are lower compared to other algorithms, but for processes which have high burst time, there may be cases of starvation.

3. Shortest Remaining time to Completion Preemptive:
The response time, wait time and turnaround time are low and similar to the Shortest job first algorithm. The jobs which have least remaining time to complete execute first in this algorithm.

4. Round Robin Preemptive:
For this policy, all the processes get an equal time slice for execution. Due to this all processes in the queue get CPU for a limited time and completion of long processes does not take place after CPU allocation in a single turn. This majorly increases the turn around time of all the processes and the response time and wait time also increases in all the processes resulting in less throughput than the above algorithms. However a key challenge is deciding the duration of the time slice. If the time slice is large, the results are similar to FCFS and if we select a small time slice, there is high overhead of context switching.

5. Highest Priority First Preemptive:
Among all the observed algorithms, the HPF preemptive algorithm has the best throughput, though the processes with lesser priority can run into starvation. Due to its preemptive nature, the newer processes with high priority are allowed to run quickly. This scheduling algorithm, as per the observations, has very less response time, wait time and turn around time as preemptiveness allows newer processes with high priority to run quickly.

6. Highest Priority First Non Preemptive:
This algorithm has lowest throughput as compared to other algorithms except FCFS. The Highest priority First Preemptive has better results compared to this algorithm. This algorithm reduces the amount of starvation that occurs among long processes.

Final Conclusion:
We see that the Highest priority first preemptive algorithm has the highest throughput among the algorithms considered. The response time, wait times and turnaround time are lowest for SRTF Preemptive and HPF Preemptive.