Advanced Operating Systems
COEN 383 Project-2

Group 2

Viral Damaniya
Dhruv Desai
Sourabh Deshmukh
Tampara Venkata Santosh Anish Dora
Pooja Gona

Objective :

The objective of this project is to acquire hands-on experience with various process scheduling algorithms. To accomplish this, we have created a C program that puts into practice the following algorithms:

1. First come, first served (FCFS) [non-preemptive]
2. Round robin (RR) [preemptive]
3. Shortest job first (SJF) [non-preemptive]
4. Highest priority first (HPF) [non-preemptive]
5. Highest priority first (HPF) [preemptive]
6. Shortest remaining time (SRT) [preemptive]

Constraints:

1. Arrival Time, Expected Run Time, and priority are generated randomly.
2. There is no I/O time.
3. The arrival time of the process should be less than 100 quanta.
4. The time Slice for Round Robin is 1 quantum.
5. Only One process Queue exists.
6. Use 4 queues for HPF.

Below, you will find the statistical outcomes obtained from the execution of six distinct algorithms on a dataset comprising 52 processes, with meticulous consideration of all relevant constraints. The figures presented herein represent the arithmetic mean derived from five separate iterations for each algorithm.

The average of the 5 runs of every algorithm is as follows:

ALGORITHM: First-come First-served (FCFS) [non-preemptive]:

Average Wait Time: 34.2
Average Response Time: 34.2
Average _throughput: 18.0
Average Turn Around Time: 39.9

ALGORITHM: Round robin (RR) [preemptive]:

Average Wait Time: 116.7
Average Response Time: 6.2
Average _throughput: 44.0
Average Turn Around Time: 122.4

ALGORITHM: Shortest job first (SJF) [non-preemptive]:

Average Wait Time: 7.2
Average Response Time: 7.2
Average _throughput: 28.0
Average Turn Around Time: 10.7

ALGORITHM: Shortest remaining time (SRT) [preemptive]:

Average Wait Time: 5.2
Average Response Time: 4.1
Average _throughput: 29.0
Average Turn Around Time: 8.6

```
ALGORITHM: Highest _priority first (HPF) [preemptive]:

Average Wait Time: 34.2
Average Response Time: 34.2
Average _throughput: 18.0
Average Turn Around Time: 39.9

-----------------------------------------------------------------

ALGORITHM: Highest _priority first (HPF) [non-preemptive]:

Average Wait Time: 5.7
Average Response Time: 5.7
Average _throughput: 16.0
Average Turn Around Time: 7.7


-----------------------------------------------------------------
```

The subsequent observations have been derived from the outcomes of executing the six algorithms:

1. **First come, first served (FCFS) [non-preemptive] :**
   This algorithm exhibits relatively longer response time, waiting time, and turnaround time in comparison to other algorithms. Consequently, it yields a lower throughput. Moreover, there exists a potential risk of starvation as new processes must await the completion of preceding processes. However, it is noteworthy for its simplicity of implementation.

2. **Round robin (RR) [preemptive]**
   The round-robin algorithm provides equal time for all the processes in the queue for their execution, resulting in increased turnaround time, response time, and wait time for all the processes. As a consequence, the throughput of this algorithm is lower than that of the previously mentioned algorithms. The time slice's length is a critical factor in this algorithm as if the slice is too long, the results will be similar to FCFS, and if it is too short, there will be a high overhead of context switching.

3. **Shortest job first (SJF) [non-preemptive]**
   Compared to other algorithms, the response time, waiting time, and turnaround time are comparatively shorter, but there is a possibility of starvation for processes with high burst time

4. **Highest priority first (HPF) [non-preemptive]**
   This algorithm has the lowest throughput compared to other algorithms. However, it reduces the amount of starvation during long processes. The Highest Priority First Preemptive algorithm has better results than this algorithm.

5. **Highest priority first (HPF) [preemptive]**
   Out of all the algorithms observed, the HPF preemptive algorithm had the highest throughput. However, lower-priority processes may suffer from starvation. This is because the algorithm is designed to allow newer processes with high priority to run quickly, which results in very low response time, wait time, and turnaround time. The preemptive nature of this algorithm makes it efficient in terms of scheduling.

6. **Shortest remaining time (SRT) [preemptive]**
   The response time, waiting time, and turnaround time are relatively shorter and are similar to the Shortest Job First algorithm. This algorithm prioritizes jobs with the least remaining time to complete execution.

Conclusion:

The results reveal that the HPF preemptive algorithm achieves the highest throughput. In contrast, the SRTF preemptive and HPF preemptive algorithms exhibit the slowest response, wait, and turnaround times. The round-robin algorithm experiences prolonged wait times during process execution, while the non-preemptive HPF approach demonstrates the lowest throughput.