COEN 383
Advanced Operating System
Project-2 Report

Group 8:

Avani Sanjay Vaidya

Anju Varghese

Jefferson Warie

Nehru Yalla

Objective

The purpose of this project is to gain practical knowledge of process scheduling algorithms. We have executed all of the following algorithms using C program.

- 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) (preemptive)
- Highest priority first (HPF) (non-preemptive)

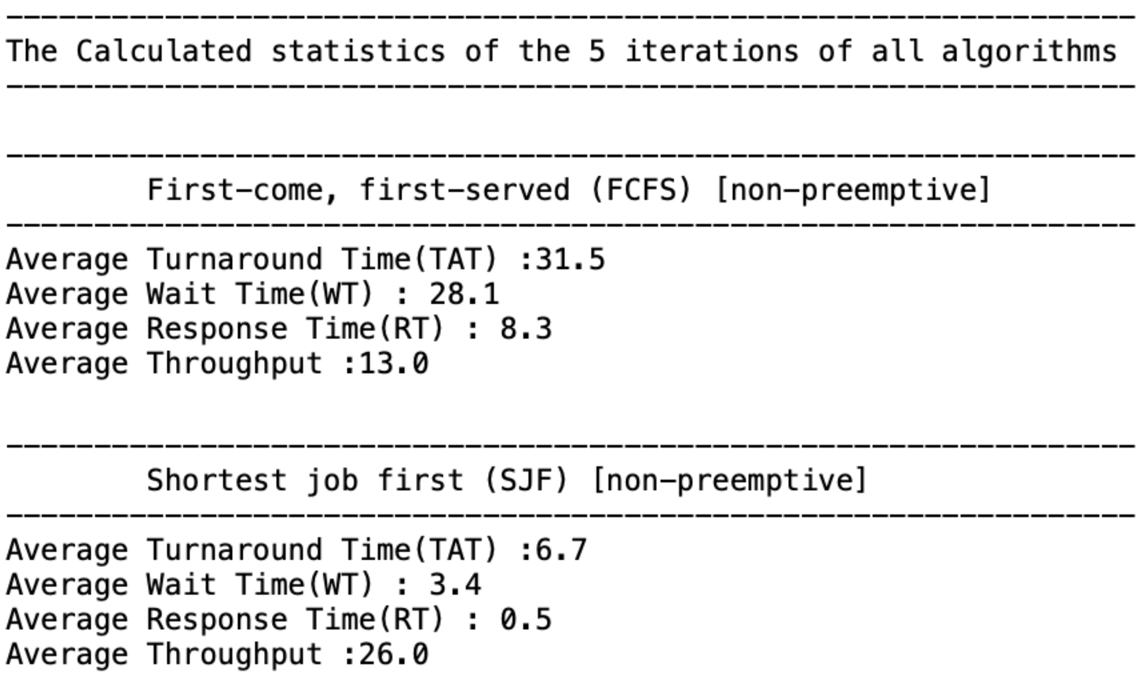
Constraints

- Only 1 priority queue is used to implement the process.
- Arrival Time of process should be less than 100 quanta.
- I/O time is not considered.
- Arrival Time, Expected Run Time & Priority are generated randomly.
- Round Robin time slice is 1 quanta
- The Highest priority first uses 4 queues.

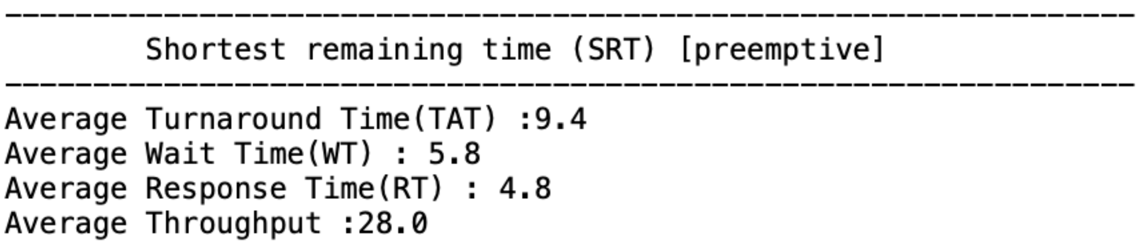
Output (Screenshots)

The final statistical output obtained from the 6 algorithms implemented on 52 processes, taken into account all constraints, is as follows.

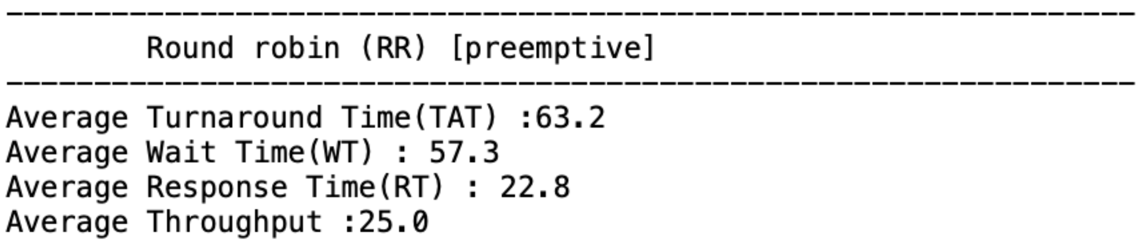
The 5 runs of all algorithms have an average of as follows:



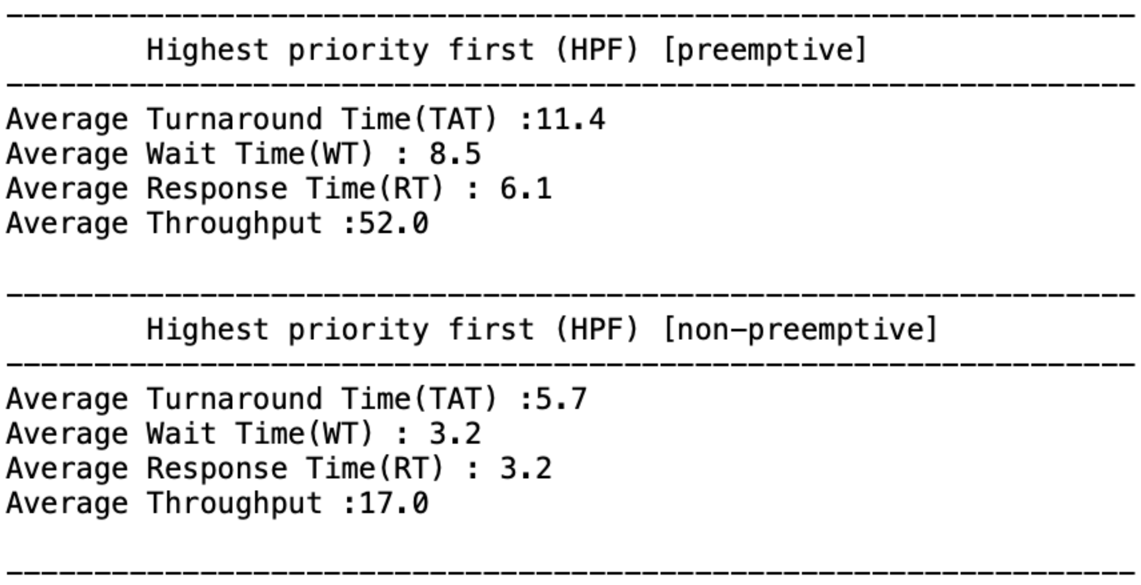
```
----------------------------------------------------------------
The Calculated statistics of the 5 iterations of all algorithms
----------------------------------------------------------------

----------------------------------------------------------------
        First-come, first-served (FCFS) [non-preemptive]
----------------------------------------------------------------
Average Turnaround Time(TAT) :31.5
Average Wait Time(WT) : 28.1
Average Response Time(RT) : 8.3
Average Throughput :13.0

----------------------------------------------------------------
        Shortest job first (SJF) [non-preemptive]
----------------------------------------------------------------
Average Turnaround Time(TAT) :6.7
Average Wait Time(WT) : 3.4
Average Response Time(RT) : 0.5
Average Throughput :26.0
```



```
----------------------------------------------------------------
        Shortest remaining time (SRT) [preemptive]
----------------------------------------------------------------
Average Turnaround Time(TAT) :9.4
Average Wait Time(WT) : 5.8
Average Response Time(RT) : 4.8
Average Throughput :28.0
```



```
----------------------------------------------------------------
        Round robin (RR) [preemptive]
----------------------------------------------------------------
Average Turnaround Time(TAT) :63.2
Average Wait Time(WT) : 57.3
Average Response Time(RT) : 22.8
Average Throughput :25.0
```



```
----------------------------------------------------------------
        Highest priority first (HPF) [preemptive]
----------------------------------------------------------------
Average Turnaround Time(TAT) :11.4
Average Wait Time(WT) : 8.5
Average Response Time(RT) : 6.1
Average Throughput :52.0

----------------------------------------------------------------
        Highest priority first (HPF) [non-preemptive]
----------------------------------------------------------------
Average Turnaround Time(TAT) :5.7
Average Wait Time(WT) : 3.2
Average Response Time(RT) : 3.2
Average Throughput :17.0

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

Observations

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

1. First Come First Serve Algorithm (Non-Preemptive):

For FCFS, response time, wait time, and turnaround time are high, which results in the lowest throughput for this algorithm. Despite FCFS being easy to implement, we can still experience starvation because the new process will only begin execution until the earlier processes are finished.

2. Shortest Job First (Non-Preemptive):

Although the response time, wait time, and turnaround time are lower than those of other algorithms, there may be cases of starvation in processes with high burst time.

3. Shortest Remaining Time to Completion (Preemptive):

The response time, wait time, and turnaround time are significantly lower than those of the Shortest job first (SJC) algorithm. In this algorithm, the process that has the least remaining time is executed first. Sorting and tracking the remaining time for each process is a time-consuming process, which is a drawback of this.

4. Round Robin (Preemptive):

Each process in this algorithm is given equal time to execute. All processes in the queue receive CPU for only a limited amount of time because of this, and long processes do not complete in a single turn after CPU allocation. This majorly increases the turnaround 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. The challenge is to determine the length of the time segment. The results are similar to FCFS when the time slice is large, but if we choose a small time slice, context switching is costly.

5. Highest Priority First (Preemptive):

The HPF preemptive algorithm has the best throughput out of all the observed algorithms, even though processes with lower priority may experience starvation. The preemptive nature allows newer processes with high priority to run quickly. According to observations, this scheduling algorithm has very short response times, wait times, and turnaround times due to its preemptiveness, which allows newer processes with high priority to run quickly.

6. Highest Priority First (Non-Preemptive):

From all the algorithms discussed the HPF preemptive algorithm has the best throughput. The throughput of this algorithm is lower than that of other algorithms. The Highest priority First Preemptive has better results compared to this algorithm. The amount of starvation that occurs during prolonged processes is decreased by this algorithm.

Conclusion

According to our observations, the Highest priority first (preemptive) algorithm is the one with the highest throughput among the algorithms considered. Both SRTF Preemptive and HPF Preemptive have the quickest response time, wait times, and turnaround time. The Highest priority first (Non-Preemptive) and FCFS algorithms have the lowest throughput, while the highest wait time for running the processes is Round Robin.