Process Scheduling Algorithms
Project 2

COEN 383 –Advanced Operating Systems

Group-1
Sowmya Baluvu (07700011659)
Tishyaketu Atul Deshpande (7700000172)
Simran Dhawan (07700011429)
Ziang Chen (7700006146)
Atharva Abhay Bal (7700005835)

Table of Contents

Objective

The objective of this project is Process Scheduling Algorithms.

The following scheduling algorithms are:

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

Constraints

1. Processes are constrained to have an Arrival Time value from 0 through 99 (measured in quanta).
2. An expected total run time is from 1 through 10 quanta.
3. A priority integer 1, 2, 3, or 4 (1 is highest) (4 is lowest)
4. Arrival Time, Expected Run Time, and Priority are randomly generated for each process.
5. The time Slice for Round Robin is set to 1 quantum.
6. For Highest Priority First (HPF), four distinct queues are employed.
7. There is only one process queue
8. No, I/O time

Algorithms

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

This algorithm has the lowest throughput because of its extremely high response time, wait time and turnaround times. There is a risk of starvation since the new processes must wait for the earlier process to finish executing. FCFS is simple to use**.**

2. Shortest Job First (Non-Preemptive):

SJF shows reduced response, wait, and turnaround times when compared to other algorithms. It is important to remember that processes with high burst time could occasionally encounter starvation.
A non-preemptive scheduling technique called the Shortest Job First (SJF) algorithm prioritizes processes according to their burst time. The process that has the least burst time is always chosen from the ready queue to decrease average waiting time and optimize system performance. Since SJF is non-preemptive, a process can continue uninterrupted until it is finished. Shorter tasks are completed more rapidly by this algorithm, which also saves time on lengthier tasks.

3. Shortest Remaining Time First (Preemptive):

The Shortest Remaining Time (SRT) algorithm is essentially Shortest Job First with preemption mode. It exhibits minimal response time, wait time, and turnaround time, like SJF.
At each quantum, the jobs with the least amount of burst me or remaining me to complete will execute first. This results in decreased wait times as well as quicker completion of shorter jobs.

4. Round Robin (Preemptive):

An equal portion of time is allotted to each procedure for execution. As a result, each process in the queue receives CPU time for a set amount of time, and lengthy processes are not completed in a single CPU allocation cycle. This results in a poorer throughput than other algorithms because it greatly increases the turnaround time, as well as the response and wait times for all processes.
Determining the optimal duration of the me slice is a significant challenge. A tiny me slice adds significant context-switching overhead, whereas a big me slice typically produces outcomes comparable to First Come First Serve (FCFS).

5. Highest Priority First Preemptive:

Among all the algorithms, the HPF preemptive algorithm has the best throughput, though the processes with lesser priority can run into starvation. As it is preemptive, the newer processes with high priority are allowed to run quickly. This scheduling algorithm, as per

the observations, has very little response time, wait time, and turnaround time allowing newer processes with high priority to run quickly.

6. Highest Priority First Non-Preemptive:

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

7. AGING

Aging is a scheduling technique used to prevent starvation in priority-based scheduling algorithms, particularly when there are processes with very low priorities that may never get CPU time because higher-priority processes dominate.

7.1. Highest Priority First Preemptive Aging:

This scheduling algorithm combines preemptive priority scheduling with aging to ensure fairness and prevent starvation.
HPF Preemptive with Aging ensures that high-priority processes get immediate attention while also promoting fairness by using aging to prevent low-priority processes from starving.
It is well-suited for systems where responsiveness and fairness are both important, such as in real-time systems, operating systems, and multitasking environments.

7.2. Highest Priority First Non-Preemptive Aging:

This scheduling algorithm combines non-preemptive priority-based scheduling with aging to prevent starvation and ensure fairness.
HPF Non-Preemptive with Aging is a scheduling algorithm that ensures fairness by using aging to gradually increase the priority of waiting processes, preventing starvation.
Even though processes run to completion without preemption, aging ensures that lower-priority processes eventually get a chance to run.
It balances fairness and efficiency by combining the simplicity of non-preemptive scheduling with the fairness of aging.

Code Execution

1. A Make file has been created to compile all the files and showcase the output. To execute the code, use the command:
`make execute`

2. To remove the object files and clean up the project directory, employ the command:
`make clean`

3. The default time slice for Round Robin is set to 1. To customize the time slice for Round Robin, use the following command:
`make run RR_TIME_SLICE =“1”`

Replace "1" with your desired time slice value.

4. To save the output in the file, and to execute the binary file as
`make execute > output.txt`

Output

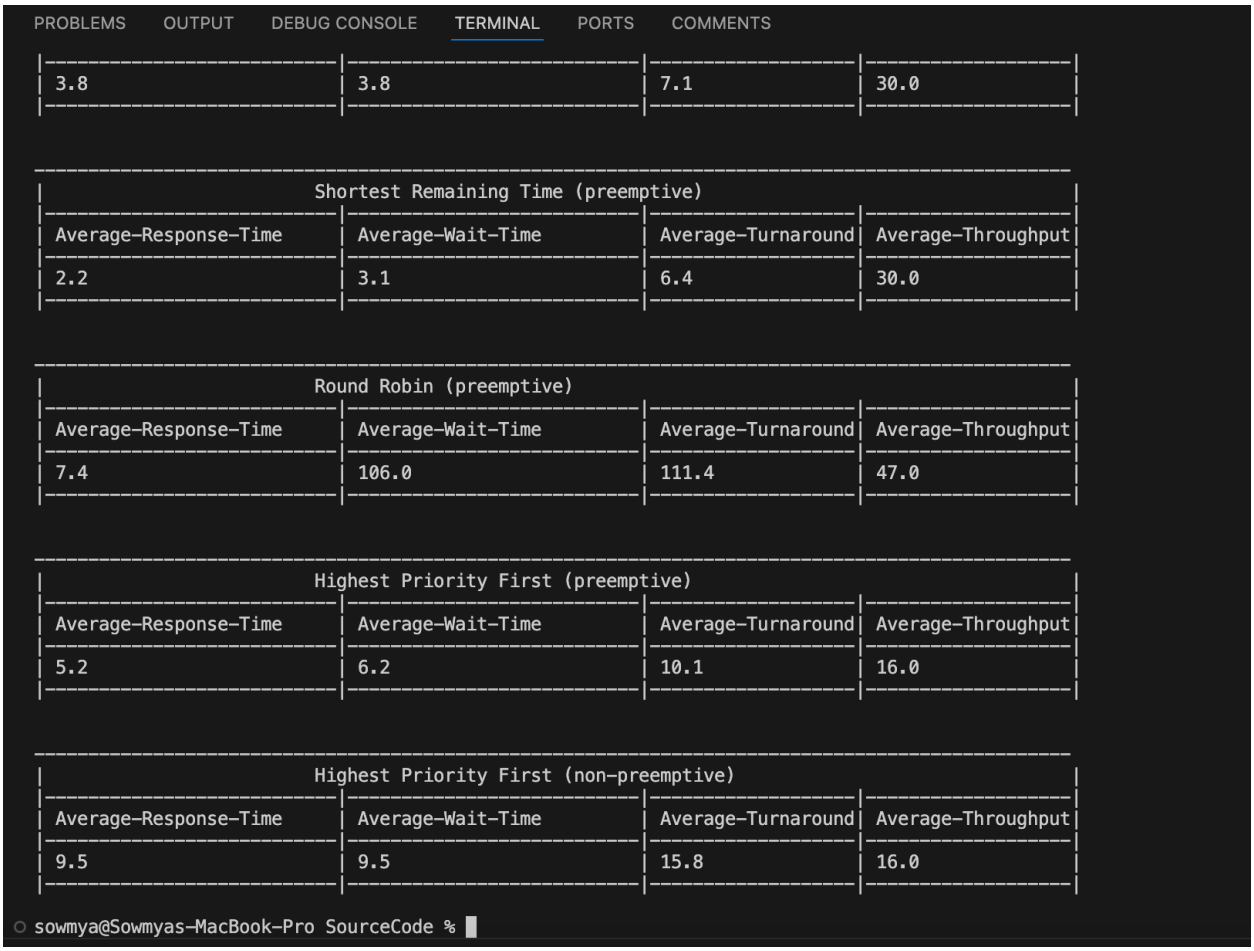
The observations from running the six algorithms have been obtained from implementing distinct algorithms on a set of 52 processes.

The average metrics obtained from 5 runs for each algorithm:

First Come First Served (non-preemptive)			
Average-Response-Time	Average-Wait-Time	Average-Turnaround	Average-Throughput
28.8	28.8	34.2	19.0

Shortest Job First (non-preemptive)			
Average-Response-Time	Average-Wait-Time	Average-Turnaround	Average-Throughput
3.8	3.8	7.1	30.0

Shortest Remaining Time (preemptive)			
Average-Response-Time	Average-Wait-Time	Average-Turnaround	Average-Throughput
2.2	3.1	6.4	30.0



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

3.8	3.8	7.1	30.0

Shortest Remaining Time (preemptive)

Average-Response-Time	Average-Wait-Time	Average-Turnaround	Average-Throughput
2.2	3.1	6.4	30.0

Round Robin (preemptive)

Average-Response-Time	Average-Wait-Time	Average-Turnaround	Average-Throughput
7.4	106.0	111.4	47.0

Highest Priority First (preemptive)

Average-Response-Time	Average-Wait-Time	Average-Turnaround	Average-Throughput
5.2	6.2	10.1	16.0

Highest Priority First (non-preemptive)

Average-Response-Time	Average-Wait-Time	Average-Turnaround	Average-Throughput
9.5	9.5	15.8	16.0

sowmya@Sowmyas-MacBook-Pro SourceCode %

Output for Aging

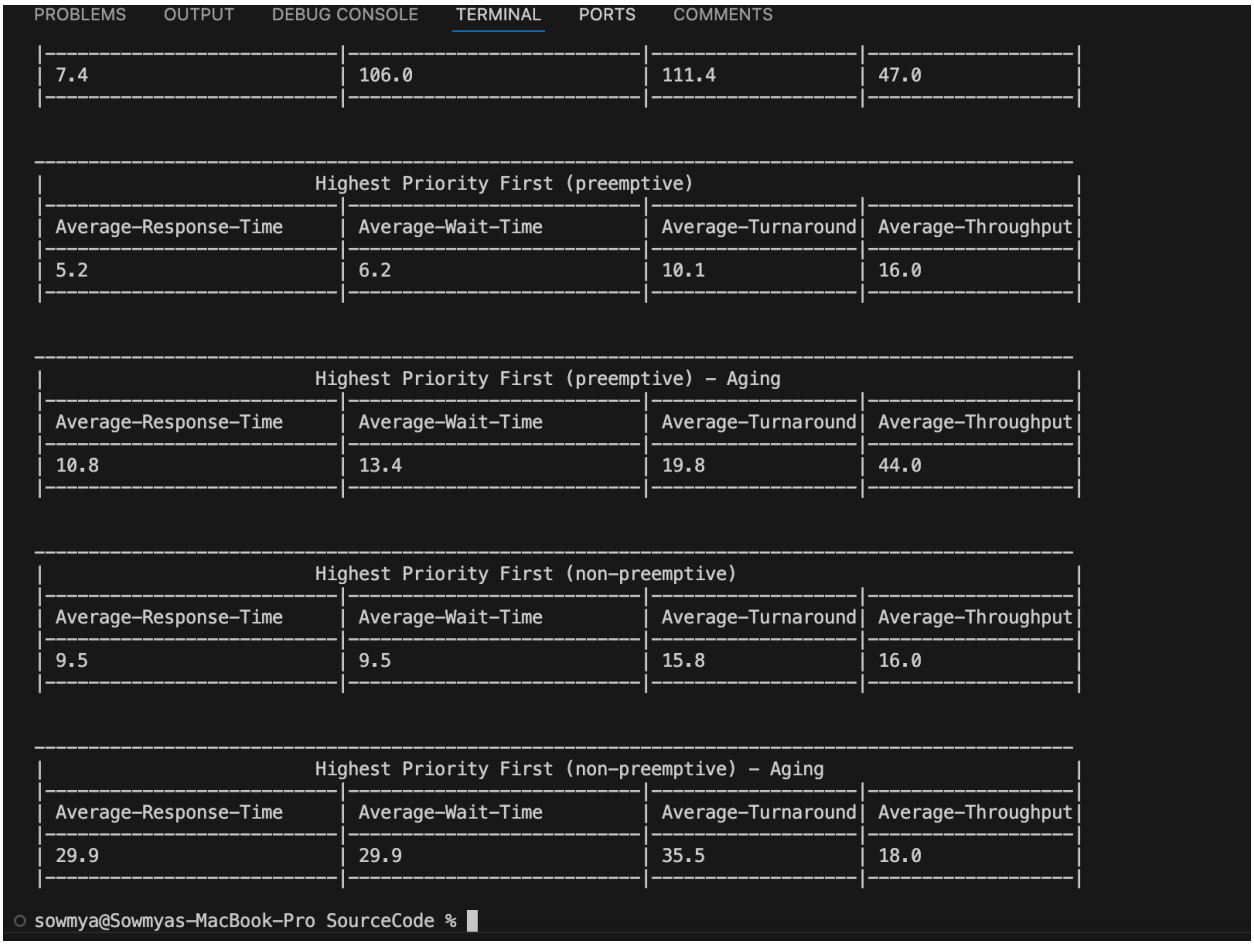
```
The average metrics obtained from 5 runs for each algorithm:


-------------------------------------------------------------------------------------------------
|                        First Come First Served (non-preemptive)                               |
|--------------------------|--------------------------|-------------------|-------------------|
| Average-Response-Time    | Average-Wait-Time        | Average-Turnaround| Average-Throughput|
|--------------------------|--------------------------|-------------------|-------------------|
| 28.8                     | 28.8                     | 34.2              | 19.0              |
|--------------------------|--------------------------|-------------------|-------------------|


-------------------------------------------------------------------------------------------------
|                        Shortest Job First (non-preemptive)                                    |
|--------------------------|--------------------------|-------------------|-------------------|
| Average-Response-Time    | Average-Wait-Time        | Average-Turnaround| Average-Throughput|
|--------------------------|--------------------------|-------------------|-------------------|
| 3.8                      | 3.8                      | 7.1               | 30.0              |
|--------------------------|--------------------------|-------------------|-------------------|


-------------------------------------------------------------------------------------------------
|                        Shortest Remaining Time (preemptive)                                   |
|--------------------------|--------------------------|-------------------|-------------------|
| Average-Response-Time    | Average-Wait-Time        | Average-Turnaround| Average-Throughput|
|--------------------------|--------------------------|-------------------|-------------------|
| 2.2                      | 3.1                      | 6.4               | 30.0              |
|--------------------------|--------------------------|-------------------|-------------------|


-------------------------------------------------------------------------------------------------
|                        Round Robin (preemptive)                                               |
|--------------------------|--------------------------|-------------------|-------------------|
| Average-Response-Time    | Average-Wait-Time        | Average-Turnaround| Average-Throughput|
|--------------------------|--------------------------|-------------------|-------------------|
| 7.4                      | 106.0                    | 111.4             | 47.0              |
|--------------------------|--------------------------|-------------------|-------------------|
```



Conclusion

The algorithm with the highest throughput among those considered is the round-robin (Preemptive).

For the Shortest Remaining Time First (SRTF) Preemptive the response time, wait time, and turnaround time are the lowest.

Conversely, the Highest Priority First exhibits the lowest throughput, while Round Robin records the highest wait time for process execution.