**Lab 1**

**Does my Code Work?**

My code is working.

**Introduction**

Determining which process will own the CPU while another process in on hold is called CPU Scheduling. There are many different algorithms that determine when to move these processes into the CPU. These algorithms can be non-preemptive or preemptive. A non-preemptive algorithm cannot leave its running state until it completes its task while a preemptive algorithm can switch to a new task before it is completed. The first come first serve algorithm is a non-preemptive algorithm that allocates the CPU to the first process that arrives. The shortest job first algorithm is a non-preemptive algorithm that allocates the CPU based on the shortest completion time. The round robin algorithm is a preemptive algorithm where the CPU is assigned for a limited amount of time (quantum), the quantum’s that we will be assigning will be 1ms, 5ms, 10ms, 15ms, 20ms, 25ms, and 50ms. To analyze the efficiency of the algorithms we will be analyzing the difference in average turnaround time (TAT), average response time (RT), CPU busy time (CBT), Throughput (T), and average waiting time (AWT).

The Results of all three algorithms are listed below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Policy** | **Policy #** | **TAT** | **RT** | **CBT** | **T** | **AWT** |
| FCFS | 1 | 12.91 | 2.67 | 0.904 | 0.89 | 11.02 |
| SJF | 2 | 4.83 | 2.05 | 0.77 | 0.97 | 2.49 |
| RR (Q=1ms) | 3 | 24.67 | 0.002 | 0.557 | 0.75 | 0.709 |
| RR (Q= 5ms) | 3 | 15.91 | 0.028 | 0.796 | 0.89 | 3.39 |
| RR (Q= 10ms) | 3 | 11.304 | 0.06 | 0.85 | 0.94 | 3.53 |
| RR (Q= 15ms) | 3 | 10.82 | 0.099 | 0.87 | 0.94 | 4.3 |
| RR (Q= 20ms) | 3 | 10.39 | 0.134 | 0.88 | 0.96 | 4.95 |
| RR (Q= 25ms) | 3 | 9.82 | 0.16 | 0.89 | 0.97 | 5.06 |
| RR (Q= 50ms) | 3 | 8.6 | 0.34 | 0.89 | 0.97 | 5.62 |

For average turnaround time, RR round with a quantun of 1ms performed the highest and shortest job first performed the lowest. With a large quantum RR behaves similarly to FCFS. This makes sense because non-preemtive tasks will generally have a lower turnaround time.

For average response time round robin with a 1ms response time performed much better than any other algorithm which is to be expected. FCFS performed the worst which is expected. FCFS and SJF performed badly because they are non-preemptive.

The CPU was used the most for first come first serve which makes sense because it is non-preemptive. Round Robin with quantum of 50ms performed very high as well which makes sense because it does not switch tasks as much.

For throughput, round robin with a low quantum performed the best while SJF and round robin with high quantum performed the worst. It makes sense that round robin with quantum = 1ms would perform badly sense it is constantly switching tasks.

For average waiting time, FCFS performed the worst and RR with a quantum = 1ms performed the best. Since AWT is dependent on time spent in the ready queue, it makes since that SJF would be low because is selects the process that needs the least amount of time to complete.

**Round Robin**

The ATAT for the RR algorithm increases as the quantum increases. If the quantum is large enough it will most likely behave like the FCFS algorithm.

Average response time increases as Quantum increases which makes sense since tasks will spend longer in the CPU

The CPU busy time increases as the quantum increases which makes sense since it will not have to switch tasks as much

Throughput does not change much which makes sense since T is a measure of the number of processes that complete their execution per unit of time, so increasing the quantum should not change this much

The average waiting time increases with the quantum which makes sense because processes will have to wait longer to get the CPU when the quantum is higher.

**Conclusion**

It is difficult to declare any one algorithm to be the best. Each one has its own strengths and weaknesses and should be used for appropriate instances.