Lab 1 Analysis

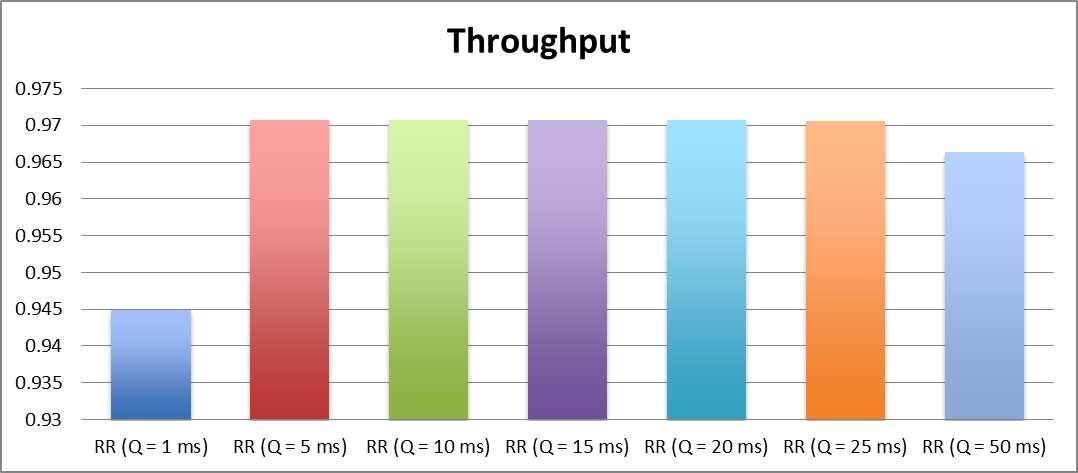
COMP 3500

Walter Matthews, Adam Brown

**DATA**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Policy | Policy Number | TAT | RT | CBT | THGT | AWT |
| FCFS | 1 | 12.45134 | 2.809881 | 0.90167 | 0.89946 | 43.882019 |
| SJF | 2 | 2.724357 | 1.023691 | 0.904827 | 1.039275 | 2.425883 |
| RR (Q = 1 ms) | 3 | 8.420236 | 0.011073 | 0.870335 | 0.944933 | 39.273079 |
| RR (Q = 5 ms) | 3 | 7.339706 | 0.043723 | 0.897605 | 0.970775 | 38.640359 |
| RR (Q = 10 ms) | 3 | 7.109464 | 0.083785 | 0.90404 | 0.970754 | 33.192927 |
| RR (Q = 15 ms) | 3 | 7.144411 | 0.125135 | 0.907285 | 0.97071 | 37.148409 |
| RR (Q = 20 ms) | 3 | 7.183261 | 0.166104 | 0.908512 | 0.970721 | 11.307943 |
| RR (Q = 25 ms) | 3 | 7.282563 | 0.207657 | 0.910483 | 0.970643 | 32.234798 |
| RR (Q = 50 ms) | 3 | 7.74806 | 0.419265 | 0.918294 | 0.96631 | 26.469879 |

Plotted below are the values for various metrics vs the RR quanta. From these we can draw conclusions about how changes in quantum size affect performance.



Here we can see a big jump in throughput between quanta 1ms and 5ms, but not much difference between 5ms-25ms. Finally, we observed a slight drop between 25ms and 50ms. This suggests that the “sweet spot” for quantum selection is somewhere in the middle. The throughput associated with the 1ms quantum is small because small quanta require considerably more context switches and therefore more overhead cost.

Average turnaround time seems to dip around 10ms-20ms but then begins to increase again. This could suggest that very low and very high quanta have worse turnaround time than those in the median.

We have a very clear positive relationship here between quantum and response time. A lower quantum means that each of the processes’ “turns” on the CPU will be shorter, and therefore a process doesn’t have to wait as long for its turn. This lowers response time.

We can also see a positive relationship between quantum and CPU busy time. This is expected, because with higher quanta, the CPU can spend more time computing and less time switching contexts.

Finally, we could not draw a relationship between quantum and average waiting time. The data we collected for this metric seemed to vary unpredictably, possibly due to sample size and/or variance.

**ANALYSIS**

After implementing and testing these policies, we were able to determine that shortest job first (SJF) has the greatest throughput among the policies at 1.039. This was expected as we had discussed in class that SJF was optimal for throughput. Round robin (RR) comes in second for throughput at around 0.970, varying slightly with the selected quantum. Finally, first-come first-served (FCFS) performed the worst for throughput at 0.899.

For turnaround time, we found SJF was still optimal 2.724. This was also expected as good throughput is an indicator of good turnaround time as they are correlated. Similarly to throughput, RR comes in second at around 7.183 (only varying slightly with quantum) followed far behind by first come first serve at 12.451.

Regarding response time, RR with the lowest quantum (1ms) comes in at the top at 0.011. Followed closely by the other RRs with increasing quanta. This is expected as RR, being the only preemptive policy, will minimize response time because each process gets a shorter turn on the CPU. Following RR, SJF had the second best response time at 1.024. Finally, FCFS comes in last for response time at 2.810.

For CPU busy time, we found that round robin with a high quantum is optimal at 0.918. First come first served and shortest job first came out around the same at 0.902 and 0.905 respectively. Finally, round robin with the lowest quantum (1ms) had the lowest CPU busy time at 0.870. This makes sense if you consider that a low quantum implies more context switches.

For average wait time, SJF was optimal by far at around 2.426. Round robin, in second, varied in an unpredictable way between 11 and 39. No correlation could be drawn from this data between wait time and quantum. This could be due to small sample size and/or variance in the set of generated processes. FCFS came in last again at 43.882.