SYSC 4001 – Assignment Part 1 – Report

Group Members:

Rigel Landmeter-Blondin, 10197902

Abdullah Khan, 101305235

Repository Links:

Part 1:

https://github.com/RigelLB/SYSC4001_A3_P1

Part 2:

https://github.com/AbdullahKhan28CU/SYSC4001_A3P2

Simulator Results:

To preface this report, midway through doing the test cases and analyzing them, I have noticed that my choice of test for the respective cases of I/O bound, CPU bound, and similar bursts are not ideal for a real test, however, I ran out of time to modify the whole report and test to fix this issue, so, the rest of this report is based on the tests previously made. I also realize that the python script I used to gather the metrics my be slightly faulty as some of the values collected look "off".

The metrics for the throughput, average wait time, average turnaround time, and average response time were collected as follows.

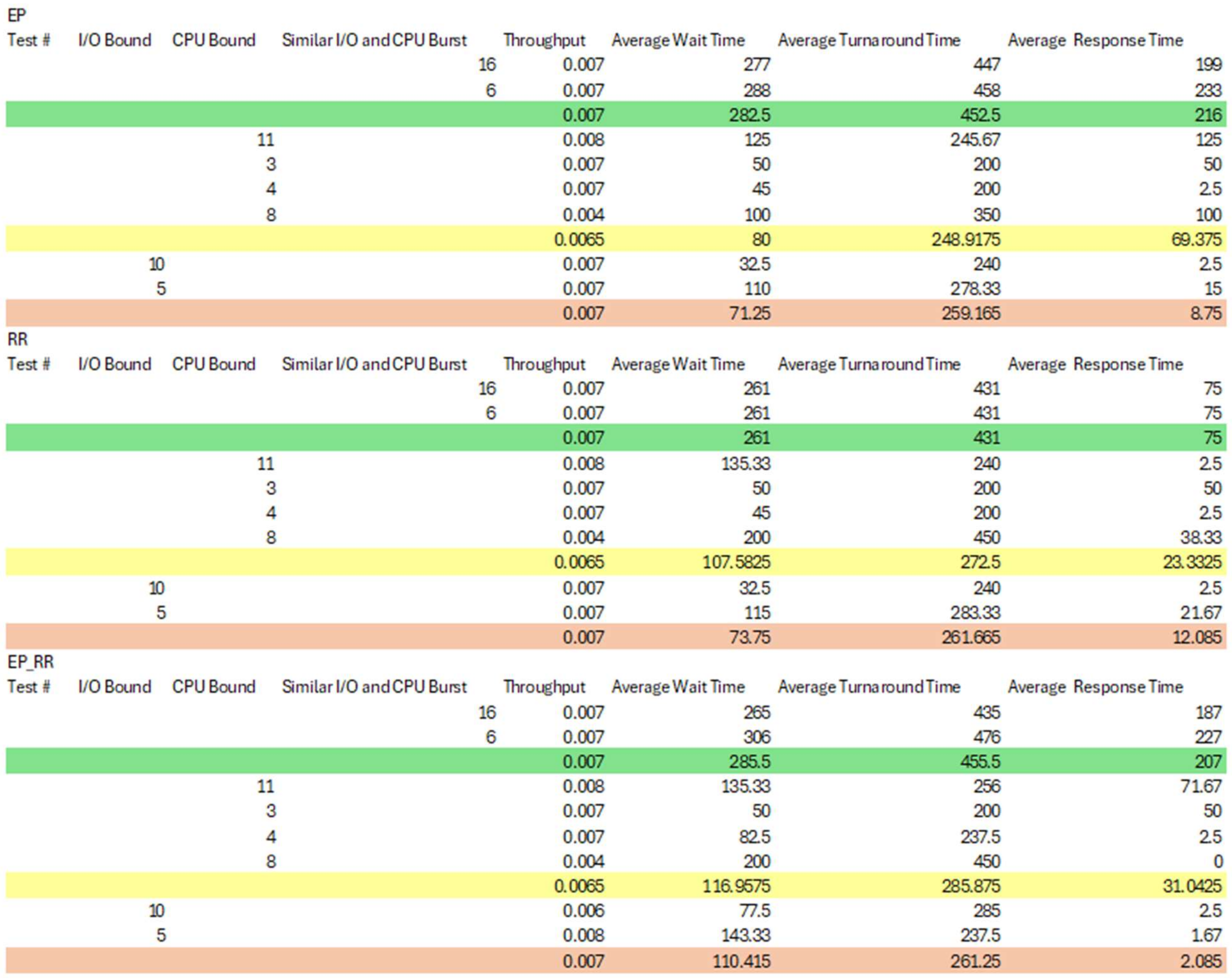
Throughput: $\frac{\#\ processes}{Total\ Time}$

Average Wait Time: Sum of time spent in the READY state

Average Turnaround Time: The time spent between the NEW state to the TERMINATED state

Average Response Time: The time spent between the NEW state to the first RUNNING state.

All the values were then summed for all the test cases in the same category is divided by the number of respective test cases. The different algorithms result in the following data table:



EP

Test #	I/O Bound	CPU Bound	Similar I/O and CPU Burst	Throughput	Average Wait Time	Average Turnaround Time	Average Response Time
			16	0.007	277	447	199
			6	0.007	288	458	233
				0.007	282.5	452.5	216
		11		0.008	125	245.67	125
		3		0.007	50	200	50
		4		0.007	45	200	2.5
		8		0.004	100	350	100
				0.0065	80	248.9175	69.375
	10			0.007	32.5	240	2.5
	5			0.007	110	278.33	15
				0.007	71.25	259.165	8.75

RR

Test #	I/O Bound	CPU Bound	Similar I/O and CPU Burst	Throughput	Average Wait Time	Average Turnaround Time	Average Response Time
			16	0.007	261	431	75
			6	0.007	261	431	75
				0.007	261	431	75
		11		0.008	135.33	240	2.5
		3		0.007	50	200	50
		4		0.007	45	200	2.5
		8		0.004	200	450	38.33
				0.0065	107.5825	272.5	23.3325
	10			0.007	32.5	240	2.5
	5			0.007	115	283.33	21.67
				0.007	73.75	261.665	12.085

EP_RR

Test #	I/O Bound	CPU Bound	Similar I/O and CPU Burst	Throughput	Average Wait Time	Average Turnaround Time	Average Response Time
			16	0.007	265	435	187
			6	0.007	306	476	227
				0.007	285.5	455.5	207
		11		0.008	135.33	256	71.67
		3		0.007	50	200	50
		4		0.007	82.5	237.5	2.5
		8		0.004	200	450	0
				0.0065	116.9575	285.875	31.0425
	10			0.006	77.5	285	2.5
	5			0.008	143.33	237.5	1.67
				0.007	110.415	261.25	2.085

The different colors are used to separate different groups of tests from each other. For example, the first set of green values are the collective averages of the Similar I/O and CPU Burst test group using the External Priorities algorithm.

From the collected data, the waiting time for similar IO/CPU bursts are similar, with the slight edge to the round robin algorithm. In terms of response times, the round-robin algorithm beats out the two other algorithms by a wide margin. In theory, the response time for the external priority algorithm should be worse than the two other algorithms since processes run until completion before handing over the CPU, which we do see in the simulation results. The hybrid external priority w/ the round-robin timeout should have slightly better response times since it gives up the CPU with higher priority processes when they arrive and are forced out by the quantum timer, which we do see in the data. In general, RR will be very good at reducing the waiting time and response time in I/O bound situations since it gives the opportunity to all processes to use the CPU equally. However, in this simulation it is not observed. This could be due to the test cases and the relatively large quantum. Since most of the processes don't have CPU processing times that are very large, they will very rarely be handled by the round robin style timeout. In practice, however, the hybrid approach can be very powerful and outshine the other algorithm in most cases. The ability to attribute priorities can be very important since some processes will be more important than others. The downsides such as the response time can be attributed somewhat by modifying the quantum to obtain the best performance.