Assignment 3 Part 1

Scheduler Simulation Report

Note: This report will use milliseconds (ms) as a unit of time to keep consistent with the Round Robin "100ms" quantum, but the time units are not in milliseconds.

Table 1: EP Only Scheduler:

Test File Used	Throughput (processes/ms)	Average Wait Time (ms)	Average Turnaround Time (ms)	Average Response Time (ms)
Test1	0.1	0	10	0
Test2	0.0909	0	11	0
Test3	0.1333	3.5	11	3.5
Test4	0.1429	1.5	9.5	0
Test5	0.005	99.5	299.5	99.5
Test6	0.0048	122.5	387.5	25
Test7	0.0039	288.33	615	205.33
Test8	0.005	225.5	485.5	104.5
Test9	0.0023	453.33	886.67	453.33
Test10	0.0026	348.75	1278.75	41.5
Test11	0.0059	83.67	270.33	69.33
Test12	0.0034	219.33	602.67	182.33
Test13	0.0034	255.75	608.25	71.5
Test14	0.0104	231.8	375.8	104.6
Test15	0.0021	563.33	1036.67	563.33
Test16	0.0035	187.75	1085.25	42.75
Test17	0.0044	363	658	303.25
Test18	0.0042	407.4	701.4	334.2
Test19	0.0025	141	1151	34.33
Test20	0.0048	190	396.67	190

Average Throughput: 0.0267

Average Wait: 209.3

Average Turnaround Time: 544.0

Average Response Time: 141.4

Table 2: RR Only Scheduler:

Test File Used	Throughput (processes/ms)	Average Wait Time (ms)	Average Turnaround Time (ms)	Average Response Time (ms)
Test1	0.1	0	10	0
Test2	0.0909	0	11	0
Test3	0.1333	3.5	11	3.5
Test4	0.1429	1.5	9.5	0
Test5	0.005	150	350	50
Test6	0.0048	122.5	387.5	25
Test7	0.0041	362.67	689.33	81
Test8	0.005	344.5	604.5	77
Test9	0.0024	657.67	1091	71
Test10	0.0024	356.25	1286.25	49
Test11	0.0069	158.67	345.33	56
Test12	0.0036	347.67	731	57.67
Test13	0.0035	289.75	642.25	56.75
Test14	0.0104	260.4	404.4	66
Test15	0.0023	697.33	1170.67	97.67
Test16	0.0035	187.75	1085.25	42.75
Test17	0.0041	331.5	626.5	115.25
Test18	0.0045	625.6	919.6	166
Test19	0.0025	141	1151	34.33
Test20	0.0048	324.67	531.33	97.67

Average Throughput: 0.0268

Average Wait: 268.1

Average Turnaround Time: 602.9

Average Response Time: 57.33

Table 3: EP and RR Combination Scheduler:

Test File Used	Throughput (processes/ms)	Average Wait Time (ms)	Average Turnaround Time (ms)	Average Response Time (ms)
Test1	0.1	0	10	0
Test2	0.0909	0	11	0
Test3	0.1333	2.5	10	0
Test4	0.1429	1.5	9.5	0
Test5	0.005	99.5	299.5	99.5
Test6	0.0048	122.5	387.5	25
Test7	0.0039	288.33	615	205.33
Test8	0.005	249.5	509.5	38
Test9	0.0023	453.33	886.67	453.33
Test10	0.0026	366.75	1296.75	10.5
Test11	0.0059	87	273.67	59
Test12	0.0034	219.33	602.67	182.33
Test13	0.0034	256	608.5	36.25
Test14	0.01	218	362	0
Test15	0.0021	563.33	1036.67	563.33
Test16	0.0035	187.75	1085.25	42.75
Test17	0.0044	363	658	303.25
Test18	0.0042	407.4	701.4	334.2
Test19	0.0023	146.33	1156.33	13.67
Test20	0.0048	190	396.67	190

Average Throughput: 0.0267

Average Wait: 211.1

Average Turnaround Time: 545.8

Average Response Time: 127.8

Analysis:

As you can see from these results, the average throughput for each algorithm is incredibly similar with EP and EP + RR being 0.0267ms and RR being 0.0268ms. Next, the average wait time where EP and EP + RR are also similar being 209.3ms and 211.1ms respectively, whereas RR had a longer wait time of 268.1ms. Then, we have average turnaround time where again EP and EP + RR were similar being 544.0ms and 545.8ms respectively and RR was a little bit longer at 602.9. Finally, there's average response time where this time RR was the fastest by far at 57.33ms, then EP + RR at 127.8ms, and lastly, EP at 141.4ms. From these results it is clear that the EP and EP + RR algorithm both have much faster throughput, wait and turnaround times when compared to the RR algorithm, however the RR algorithm has a much faster response time compared to EP and EP + RR. This indicates that the RR algorithm would be much better suited for systems that have processes requiring quick response times, such as interactive processes, whereas the EP algorithm would be best suited for systems that have processes with little interaction and focus more on throughput and turnaround time, and the EP + RR algorithm would be best suited for systems where starvation is an issue. In conclusion, there is not a definitive best or worst scheduling algorithm, and each system should have an algorithm that best suits its needs, so that it can perform as optimal as possible.