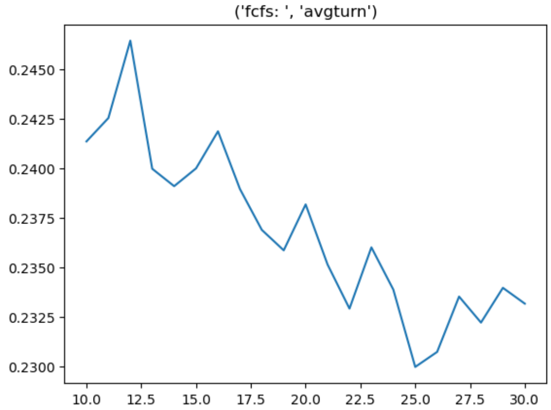
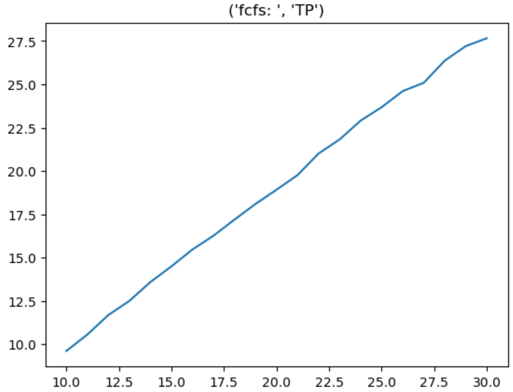
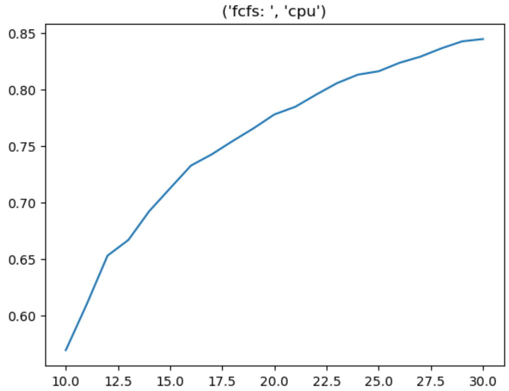
Report: bellow is graphs for the simulations run along with an interpretation of the data for lambda 10 to 30 for process arrival all with an average service time of .04. X axis is lamda for all and the y-axis is the title of the graph
Avgturn: average turnaround time
TP: throughput
Cpu: CPU utilization
abgReady: average number in the ready queue

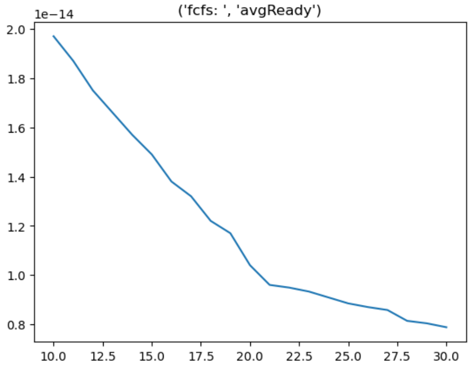
First come first serve

This would complete processes about as fast as they would come in due to the fast service time this meant as the number of processes increased the CPU usage went up and throughput matched the arrival rate.



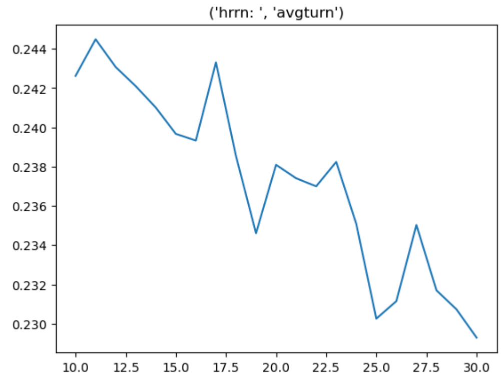


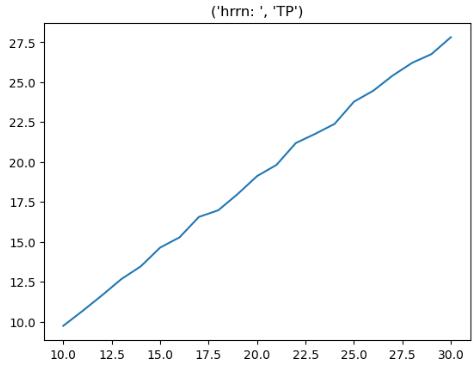


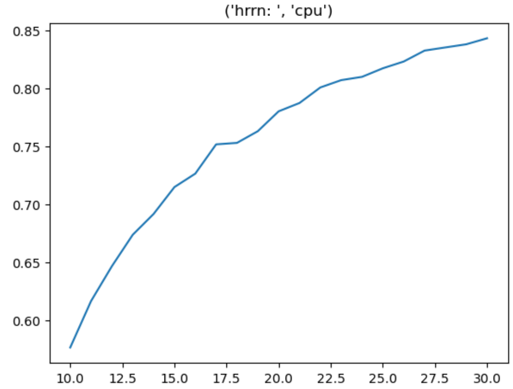


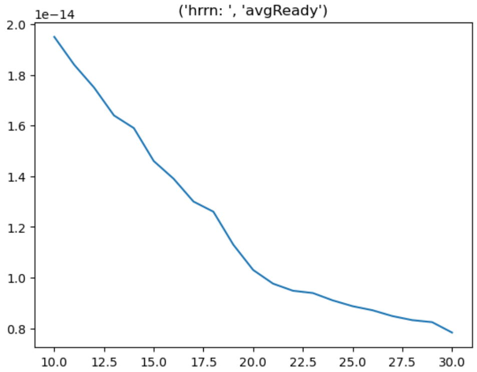
HHRN

This performed a little better than the first come first serve however it would reorder the ready queue in order to finish the processes in a more fair order this caused there to be some variation in average turnaround time. it still would have a throughput that was close to the arrival rate for this service time. CPU usage went up as more processes arrived.



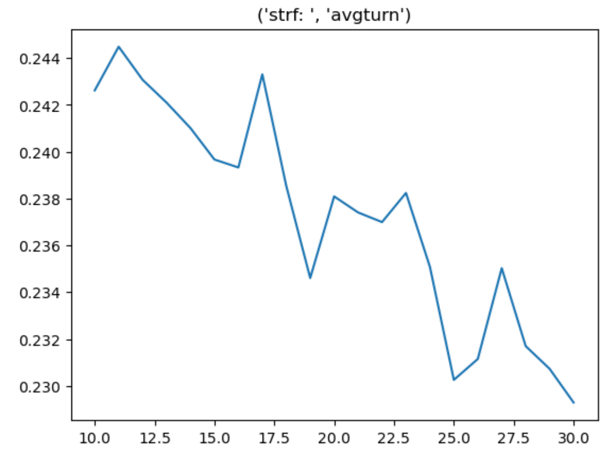


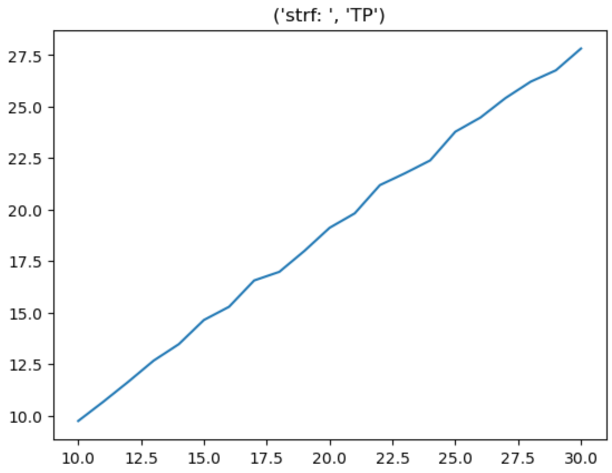


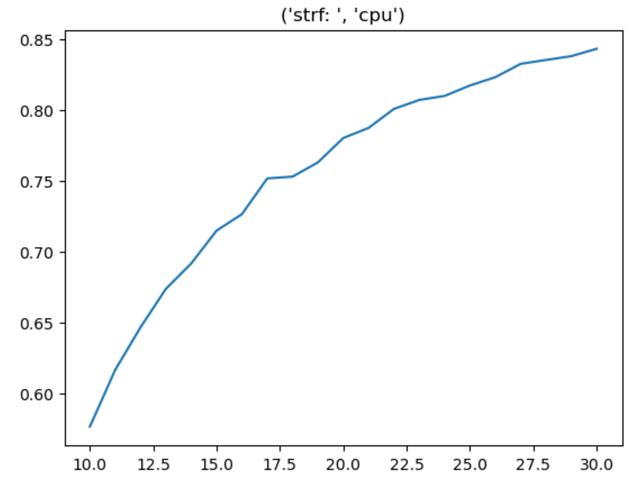


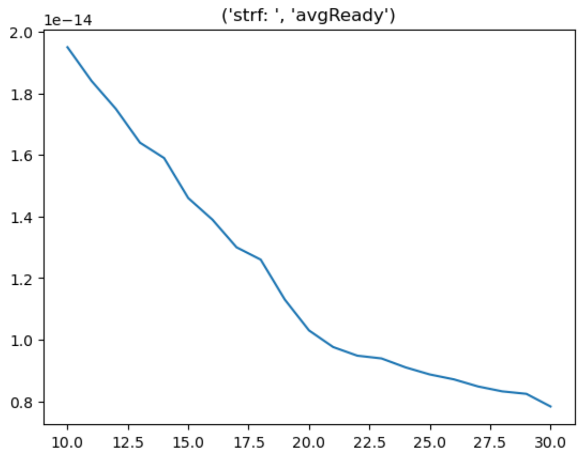
STRF

This one struggled with average turnaround time because sometimes a long process can mess up the average but it still is low. but it improved the throughput this is because some long processes would have to wait a long time to get processed thus the avg turnaround time went up, while it was able to finish a lot of processes faster because it always picked the shortest one and would stop a long process even if it started running. this could lead to starvation for a long process. This could also lead to more processed being in the ready queue because some would begin to starve.



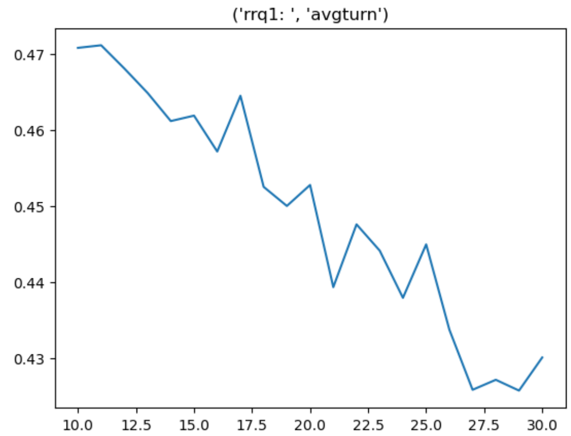


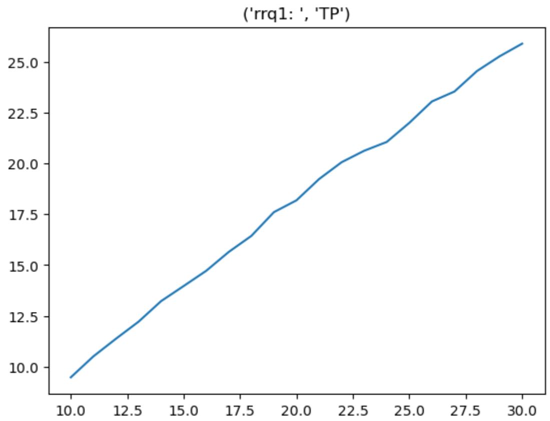


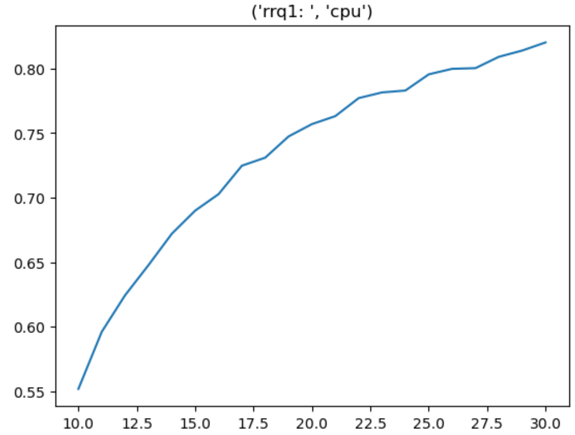


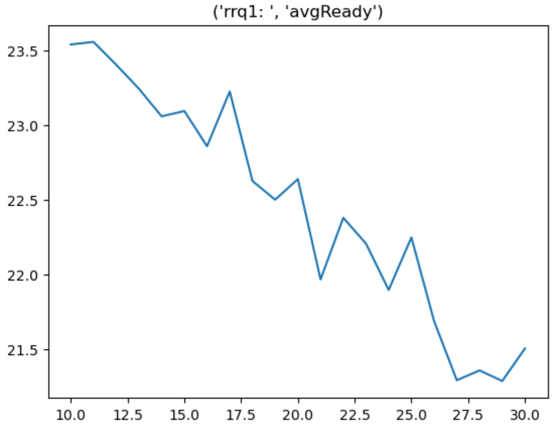
RR .01quantum

This one struggled the most when lots of processes came in fast because the quantum was so short it was not able to make a lot of progress on a process and the ready queue would fill up fast. The turnaround time would all be similar for each process bringing the average down over time.



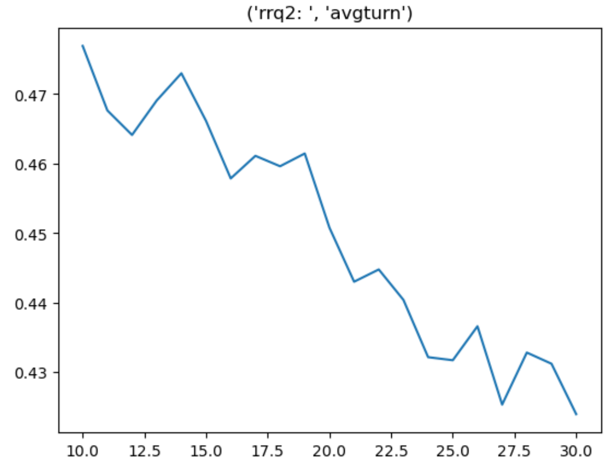


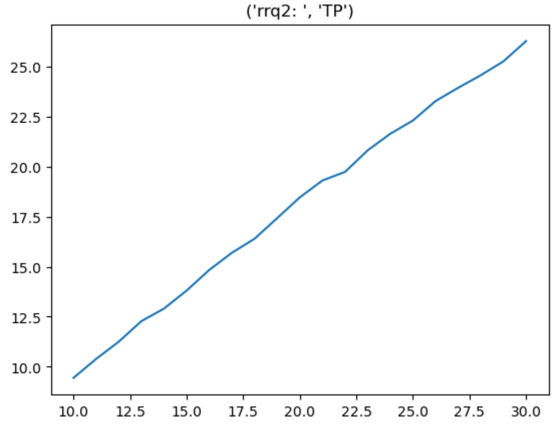


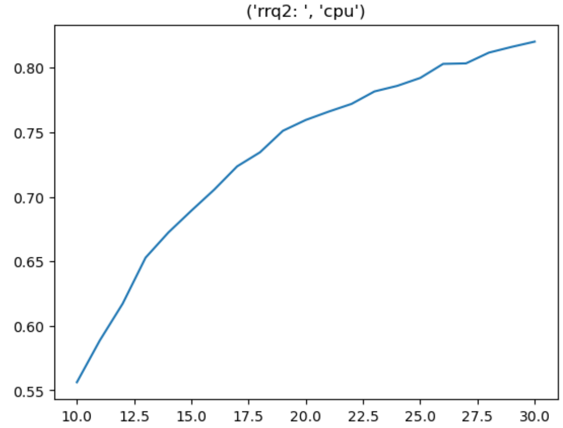


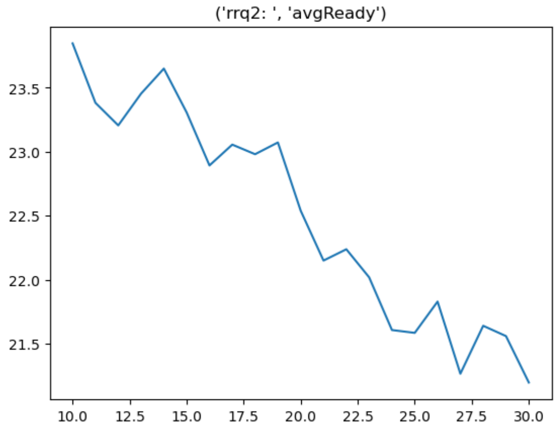
RR .2quantum

This improved on the previous round-robin in the sense it could finish processes faster because the quantum was closer to the average service time. Making the turnaround time graph a little steeper and smoother as a process could not get stuck in the loop as often as it would finish.The turnaround time would all be similar for each process


('rrq2: ', 'avgturn')
0.47
0.46
0.45
0.44
0.43
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0


('rrq2: ', 'TP')
25.0
22.5
20.0
17.5
15.0
12.5
10.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0


('rrq2: ', 'cpu')
0.80
0.75
0.70
0.65
0.60
0.55
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0


('rrq2: ', 'avgReady')
23.5
23.0
22.5
22.0
21.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0