

Final Outputs

FCFS(First Come First Serve):

Average Turnaround Time (TAT): 21.64 quanta

Average Wait Time (WT): 16.26 quanta

Average Response Time (RT): 16.26 quanta

Average Throughput: 0.1871 processes/quantum

SJF(Shortest Job First):

Average Turnaround Time (TAT): 10.71 quanta

Average Wait Time (WT): 6.04 quanta

Average Response Time (RT): 6.04 quanta

Average Throughput: 0.2264 processes/quantum

SRT(Shortest Remaining Time):

Average Turnaround Time (TAT): 10.43 quanta

Average Wait Time (WT): 5.77 quanta

Average Response Time (RT): 5.14 quanta

Average Throughput: 0.2265 processes/quantum

RR(Round Robin):

Average Turnaround Time (TAT): 41.58 quanta

Average Wait Time (WT): 36.01 quanta

Average Response Time (RT): 6.43 quanta

Average Throughput: 0.1810 processes/quantum

HPF(Highest Priority First - Preemptive):

Average Turnaround Time (TAT): 27.01 quanta

Average Wait Time (WT): 21.42 quanta

Average Response Time (RT): 6.53 quanta

Average Throughput: 0.1783 processes/quantum

Priority-Level Breakdown:

Priority	Avg Turnaround (TAT)	Avg Wait (WT)	Avg Response (RT)	Throughput
1	8.61	3.9	0.65	0.0609
2	27.49	21.14	5.41	0.0622
3	27.61	22.95	5.83	0.0378
4	80.03	75.27	14.17	0.0174

HPF(Highest Priority First - Non-Preemptive):

Average Turnaround Time (TAT): 16.09 quanta

Average Wait Time (WT): 10.54 quanta

Average Response Time (RT): 10.54 quanta

Average Throughput: 0.1802 processes/quantum

Priority-Level Breakdown:

Priority	Avg Turnaround (TAT)	Avg Wait (WT)	Avg Response (RT)	Throughput
1	9.24	4.72	4.72	0.0694
2	18.91	12.57	12.57	0.0594
3	15	10.04	10.04	0.0316
4	20.1	15.33	15.33	0.0198

Summary Report

This report analyzes 6 different scheduling algorithms and looks at how they perform in wait time, turnaround time, response time, and throughput. Each process arrives between 0-99 quanta, has a process length of 1-10 quanta and a priority between 1-4. All these values are uniform distributions.

For response times, SRT performed the best with RR and SJF also doing really well. SRT and SJF perform well in these categories due to the fact neither of those algorithms have to worry about long tasks causing another task to wait. RR performs good for this due to the fact that since each process only got a small equal slice of time on the CPU, a long task can't cause a delay in waiting for the first response. We expected RR to be the best for this category but that seemingly didn't happen due to the fact that a new arrived task goes to the end of the ready queue while for the other 2 algorithms that might not be the case. The HPF version of RR was slightly worse than regular RR and that is due to sometimes the lower priority processes having to wait longer for the higher priority processes to finish. FCFS was by far shown to be the worst and that is because a shorter process can easily have to wait for a bunch of longer processes to finish.

The analysis for turnaround time and wait time is the same as turnaround time just also includes the time a process spent in the CPU. Round Robin is the worst at this as since you have to wait for each process to get a turn and the slice causes one turn to often not be enough to finish the process. All the processes usually tend to all complete towards the end. The reason regular RR performed better than HPF RR is due to the fact that HPF RR cause the lower priority processes to not even make it to the queue by time 100 so with less processes in the queue there would be less wait and turnaround. SRT performed the best in this with SJF performing similarly but slightly worse. A short wait time helped for both algorithms and once again there is no worry about a long process causing a delay for a short process to finish. And once again for a

long process, it is not a big deal to let a shorter process finish. Conversely, that is the exact reason that once again FCFS performed poorly.

For throughput, SRT and SJF were the best as they prioritize shorter processes and that in turn allows you to run more processes. The other 3 algorithms were worse and had around the same throughput as they don't discriminate by length of a process.

Finally, HPF in both versions was really good for the high priority processes but worse for the low priority processes.