Performance Comparison of Scheduling Algorithms

Run	Average Waiting Time	Average Response Time	Average Turnaround Time
FIFO	1998.3	1998.3	2079
SJF	1489.7	1489.7	1570.6
RR - 10	2967.2	245	3047.9
RR - 100	2488.3	1821.6	2614.5

First-In, First-Out (FIFO)

Experimental Results: The typical wait and response durations are 1998.3 milliseconds, while the typical turnaround time is 2079 milliseconds.

Discussion: Even though FIFO is straightforward and quick to construct, it has longer waiting periods than SJF. Additionally, it experiences the "convoy effect," which increases the average waiting time since short operations must wait for long processes to finish.

Shortest Job First (SJF)

Experimental Results: Average wait and response times are 1489.7 milliseconds, while the average turnaround time is 1570.6 milliseconds.

Discussion: SJF excels at reducing wait times and has the fastest turnaround times on average. These measurements, however, imply that it can be vulnerable to process famine because lengthy activities might be routinely put off in favour of lesser ones.

Round Robin (RR)

Experimental Results: The average response time is 245 ms, whereas the average waiting time is 2967.2 ms for quantum size 10. The waiting time drops to 2488.3 ms for quantum size 100, whereas the response time rises to 1821.6 ms.

Discussion: The significant performance impact of context switching, and the quantum size is demonstrated by the large disparity in response times between the two quantum sizes. The context switching overhead increases with decreasing quantum size while reaction time decreases. The quantum size has a direct correlation with waiting time but an inverse relationship with response time. The data demonstrates a trade-off between throughput optimisation, which favours larger quantum sizes, and low latency optimisation, which favours smaller quantum sizes.

Favouring Interactive Processes

Shortest Remaining Time First: One could modify the SJF algorithm to become pre-emptive. If an arriving process has a burst time less than the remainder of the current process, the current process will be pre-empted. Interactive tasks usually have short burst times, so giving them priority would result in shorter waiting and response times for these tasks.