

Comparison of Scheduling Algorithms: FIFO, STCF and RR

By Ahmed Ikram

First In, First Out (FIFO) processes jobs in the order they arrive, offering simplicity and fairness as its primary benefits. It is easy to implement, but this algorithm does not consider the length of each process. As a result, it can lead to the convoy effect – where longer processes delay the execution of shorter ones – leading to increased waiting and turnaround times. FIFO is most effective in batch-processing systems, where jobs are long-running and fairness in the order of execution is prioritised over speed and responsiveness.

Shortest Time-to-Completion First (STCF), also known as pre-emptive Shortest Job First (SJF), selects the process with the shortest remaining execution time. This algorithm excels at minimizing average turnaround time and ensuring faster response for shorter jobs. However, it suffers from starvation – as longer processes may never get a chance to execute if short jobs keep arriving. STCF also requires accurate knowledge of process execution times, which may not always be feasible. It is ideal for batch-processing environments where job lengths can be predicted but less suitable for interactive or real-time systems due to its lack of fairness.

Round Robin (RR) assigns a fixed time slice – or quantum – to each process, rotating between them to ensure fairness. This method is highly effective in time-sharing systems where rapid response and equitable CPU time distribution are critical. The key advantage is that every process gets a chance to run within a specific time frame – preventing starvation. However, frequent context switches, especially when the time quantum is too large, mean round robin is best suited for interactive systems such as modern operating systems, where user responsiveness and fairness are key factors.

FIFO is most effective for simple batch systems, STCF excels in minimising turnaround in predictable environments and RR is ideal for time-sharing systems, balancing fairness and responsiveness. The choice of algorithm depends largely on system goals – whether favouring simplicity, speed or fairness.