OPERATING SYSTEMS ASSIGNMENT WEEK 5

The following is my reflection on the Operating systems assignment and comparison of FCFS and SJF (the two scheduling algorithms I coded a process simulator of).

In my recent assignment on operating systems, I coded a process simulator that focused on two scheduling algorithms: First-Come, First-Served (FCFS) and Shortest Job First (SJF). Both algorithms.FCFS was straightforward and easy to implement because processes are executed in the order they arrive in the process queue. Using this algorithm, the earlier processes will be implemented before the others, similar to how people in a queue at a shop will be attended to. The benefit of this compared to other algorithms is that each process gets a chance without being preempted. This is key when it comes to user interface systems with multiple people of similar processes like ticket-buying websites, where it is fair that the earlier people get their desired tickets first. However, its major drawback lies in the “convoy effect,” where shorter processes must wait for longer processes to complete, leading to increased turnaround time and inefficiencies, and could lead to deadlocks in important processes (e.g operating system of a laptop) so it is not suitable for these types of operating systems. It is also one of the least efficient in terms of total waiting time.

On the other hand, SJF aims to minimise waiting time by selecting the shortest job available next. This algorithm can significantly reduce average turnaround and waiting times because it prioritises smaller jobs. However, SJF has its drawbacks, as it can lead to starvation for longer processes if shorter jobs continuously arrive. Additionally, it is not always practical because estimating the length of jobs can be challenging and takes up memory continually preempting new processes.

For instance, if 3 processes have burst times of 8, 5, and 2 milliseconds, FCFS would lead to a total waiting time of 15 milliseconds, whereas SJF would effectively complete the shortest job first, significantly reducing the total waiting time. For response time, SJF also holds an advantage. Important processes queued in FCFS may experience delayed response if they are not among the first to arrive, whereas SJF prioritizes shorter tasks, improving the responsiveness of the system.

Therefore, while both are different, FCFS provides a fair and straightforward approach, and SJF offers enhanced efficiency at the potential risk of starvation. Each is suitable for different types of systems based on specific operational needs. FCFS is good for batch processing or simple tasks where a small proportion of resources are used. SJF is good for user-interactive systems, where speed is prioritised, or systems with short tasks.