Assignment 2A: Processor Scheduling with C Programming

Task 4

Ray Krishardi Layadi - 26445549

First-Come-First-Served (FCFS) Scheduling

In the case of First-Come-First-Served (FCFS) scheduling algorithm, it appears that for the given input file “processes1.txt”, the average turnaround time (7.25) and waiting time (3.5) for this scheduling algorithm is the **2nd shortest** compared to the Round Robin (RR) and Shortest Remaining Time (SRT) scheduling algorithm. This is mainly because **FCFS favors processes with longer processing time (i.e. longer processes)** than processes with shorter processing time (i.e. shorter processes). The reason behind this is because FCFS is a non-preemptive scheduling algorithm which means that the process that is currently being executed by the processor cannot be interrupted/preempted (i.e. cannot be send back to the ready queue). As a result, longer processes will benefit more than the shorter processes. In addition, the overall throughput of the system is the same for all three scheduling algorithms (throughput = 0.27 jobs/second) as all three algorithms have the same number of processes (i.e. 4 processes) and completed the execution in 15 seconds. Therefore, we can conclude that the overall system performance when using this algorithm is better than the RR but it is still inferior compared to the SRT scheduling algorithm for the input file “processes1.txt”.

On the other hand, for the given input file “processes2.txt”, the average turnaround time (12.2) and waiting time (7.4) for this scheduling algorithm is the **2nd shortest**. Interestingly, in this case, both average turnaround and waiting time for this algorithm are the same as the Round Robin (RR) algorithm’s average turnaround and waiting time which means that their performance is the same for input file “processes2.txt”. In addition, the overall throughput of the system is the same for all three scheduling algorithms (throughput = 0.21 jobs/second) as all three algorithms have the same number of processes (i.e. 5 processes) and completed the execution in 24 seconds. Therefore, we can conclude that the overall system performance when using this algorithm is the same as the RR but it is still inferior compared to the SRT scheduling algorithm for the input file "processes2.txt".

Round Robin Scheduling

In the case of Round Robin (RR) scheduling algorithm with quantum time of 2 seconds, it appears that for the given input file "processes1.txt", the average turnaround time (8.25) and waiting time (4.5) for this scheduling algorithm is the **3rd shortest** compared to the First-Come-First-Served (FCFS) and Shortest Remaining Time (SRT) scheduling algorithm. This is mainly because RR with quantum time of 2 seconds provides a fair/equal treatment for both longer and shorter processes. The reason behind this is because RR is a preemptive scheduling algorithm which means that the process that is currently being executed by the processor can be interrupted/preempted (i.e. can be send back to the ready queue) depending on the selected time quantum (2 seconds in this case). As a result, the performance of the system using this algorithm depends on the length of the quantum time (i.e. short quantum time could result in frequent unnecessary context switching while long quantum time could result in the algorithm behaving the same way as FCFS). In addition, the overall throughput of the system is the same for all three scheduling algorithms (throughput = 0.27 jobs/second) as all three algorithms have the same number of processes (i.e. 4 processes) and completed the execution in 15 seconds. Therefore, we can conclude that the overall system performance when using this algorithm is inferior compared to the FCFS and SRT scheduling algorithm for the input file "processes1.txt".

On the other hand, for the given input file "processes2.txt", the average turnaround time (12.2) and waiting time (7.4) for this scheduling algorithm is the **2nd shortest**. Interestingly, in this case, both average turnaround and waiting time for this algorithm are the same as the FCFS algorithm's average turnaround and waiting time which means that their performance is the same for input file "processes2.txt". In addition, the overall throughput of the system is the same for all three scheduling algorithms (throughput = 0.21 jobs/second) as all three algorithms have the same number of processes (i.e. 5 processes) and completed the execution in 24 seconds. Therefore, we can conclude that the overall system performance when using this algorithm is the same as the FCFS but it is still inferior compared to the SRT scheduling algorithm for the input file “processes2.txt”.

Shortest Remaining Time Scheduling

In the case of Shortest Remaining Time (SRT) scheduling algorithm, it appears that for the given input file “processes1.txt”, the average turnaround time (6.25) and waiting time (2.5) for this scheduling algorithm is the **1st shortest** compared to the First-Come-First-Served (FCFS) and Round Robin (RR) scheduling algorithm. This is mainly because **SRT favors processes with shorter processing time (i.e. shorter processes)** than processes with longer processing time (i.e. longer processes). The reason behind this is because SRT is a preemptive scheduling algorithm which means that the process that is currently being executed by the processor can be interrupted/preempted (i.e. can be send back to the ready queue) depending on the shortest expected remaining processing time (i.e. processes with shorter expected remaining processing time will always get executed first). As a result, shorter processes will definitely benefit more than the longer processes. In fact, this algorithm could result in starvation for longer processes if there is a constant amount of incoming short processes. In addition, the overall throughput of the system is the same for all three scheduling algorithms (throughput = 0.27 jobs/second) as all three algorithms have the same number of processes (i.e. 4 processes) and completed the execution in 15 seconds. Therefore, we can conclude that the overall system performance when using this algorithm is the most superior compared to the FCFS and RR scheduling algorithm for the input file “processes1.txt”.

On the other hand, for the given input file “processes2.txt”, the average turnaround time (8.6) and waiting time (3.8) for this scheduling algorithm is the **1st shortest** compared to the First-Come-First-Served (FCFS) and Round Robin (RR) scheduling algorithm. In addition, the overall throughput of the system is the same for all three scheduling algorithms (throughput = 0.21 jobs/second) as all three algorithms have the same number of processes (i.e. 5 processes) and completed the execution in 24 seconds. Therefore, we can conclude that the overall system performance when using this algorithm is the most superior compared to the FCFS and RR scheduling algorithm for the input file “processes2.txt”.