SYSC 4001 - Assignment 3 Report

Student 1: Radhe Patel 101262847
Student 2: Anvita Ala 101301514
Date: Friday, November 28th, 2025

1. Introduction

The goal of this part of assignment 3 was to build a small-scale scheduler simulator. The code built off the code from the previous two assignments and effectively simulated the scheduler's CPU allocation of three different scheduling algorithms: External Priorities, Round Robin with a 100ms time quantum, and a combined preemptive EP/RR.

The objective of this analysis is to determine how each algorithm behaves if the processes are mostly I/O bound, mostly CPU-bound processes, or if they have similar I/O and CPU bursts. The performance is measured and compared using throughput, average wait time, average turnaround time, and average response time.

The following section shows the simulation details from 20 test cases for each of the three aforementioned algorithms.

2. Analysis

a. Methodology

In this assignment, three different schedulers were simulated in C++. There was the External Priority scheduler (EP), the Round Robin scheduler (RR), and finally the External Priority with 100 ms timeout Round Robin scheduler (EP+RR). For each scheduling simulation, 20 test cases are used to evaluate the functionality and effectiveness of each algorithm. The results were placed in a table, which included the Time of Transition, the Process ID (PID), the Old State, and the New State. In this report, we calculated the Throughput, Average Wait Time, Average Turnaround Time, and Average Response Time for each test case to analyze the results from each scheduling algorithm. Below are the results and conclusions that were drawn from the data collected.

Results from calculations (only first 5 test cases for each type of scheduling algorithm):

Scheduler	**Avg Turnaround**	**Avg Waiting**
EP 1	10.0	0.0
EP 2	12.0	4.5
EP 3	11.0	3.5
EP 4	14.0	6.5

EP 5	4.67	1.67
RR 1	100.0	50.0
RR 2	35.0	0.0
RR 3	200.0	75.0
RR 4	105.0	35.0
RR 5	22.5	12.5
EP+RR 1	300.0	0.0
EP+RR 2	325.0	150.0
EP+RR 3	345.0	145.0
EP+RR 4	475.0	225.0
EP+RR 5	310.0	176.67

b. Results from External Priority (EP)

The External Priority (EP) scheduling algorithm schedules processes based on their PID. Once they start running on the CPU, the next process can begin as soon as the current process has released the CPU (either I/O or until the process finishes executing).

CPU-Bound → When lots of processes are CPU-bound, meaning they have long burst times and few to no I/O operations, the EP scheduling algorithm always gives higher priority tasks first, and if that happens to be for a CPU-bound process, then this leads to long waiting and turnaround times for the jobs that are shorter and don't require as much CPU time (lower-priority). The throughput was not really affected by CPU-bound processes, but some starving delays can be seen in the shorter processes.

I/O-Bound → When lots of processes are I/O-bound, meaning they have short burst times and lots of I/O operations, the EP scheduling algorithm usually gets quick access to the CPU. Lower-priority processes might have to wait some time before using the CPU, making their waiting times and turnaround times longer.

When there is a mixture of both CPU-bound and I/O-bound processes, the higher-priority tasks will always run first. Overall, the EP scheduling algorithm has a good response time, but it can cause the lower-priority processes to have longer wait times and turnaround times.

c. Results from Round Robin (RR)

The Round Robin (RR) scheduling algorithm has a quantum (100 ms) and runs through all the tasks. Once a task takes longer than the quantum value to complete, it is returned to the ready queue at the end, making the RR scheduling algorithm preemptive.

CPU-Bound → When lots of processes are CPU-bound, the RR scheduling algorithm shares the CPU time between those processes evenly, which can cause the wait times to increase. Overhead is also increased in CPU-bound processes because there is a lot of changing processes to share the CPU evenly (100 ms per process). The throughput and the turnaround time are normal for RR scheduling.

I/O-Bound → When lots of processes are I/O-bound, the RR scheduling algorithm is more effective. Shorter processes do not have to wait after a long process to complete their execution. This algorithm benefits the I/O-bound processes and decreases their turnaround times compared to the other algorithms.

When there is a mixture of both CPU-bound and I/O-bound processes, the shorter processes do not have to wait for the CPU-bound process to complete its execution in order to finish. This algorithm gives fairness to all processes at the cost of increasing overhead if CPU-bound processes are commonly found.

d. Results from External Priority with Round Robin (EP + RR)

The External Priority with Round Robin (100ms) scheduling algorithm runs similarly to EP, but after quantum (100 ms), the process is returned to the ready queue and has to wait its turn again to use the CPU.

CPU-Bound → When lots of processes are CPU-bound, the EP + RR scheduling alogrthms lets the high priority process run first, but still time slices then at the quantum time (100 ms), therefore the lower priority processes have a chance to finish with a responsible waiting and turnaround time.

I/O-Bound → When lots of processes are I/O-bound, the high priority process get the CPu quickly after the I/O completion and the lower priority processes also get a chance to complete quicker than EP scheduling algorithm and no starvation.

When there is a mixture of both CPU-bound and I/O-bound processes, the processes all get a fair chance to use the CPU, higher priority tasks do get to go first, followed by the medium and lower priority tasks, to execute. Overall the EP + RR scheduling algorithms results are similar to the RR scheduling algorithms results.

e. Comparison

Throughput → similar for all three scheduling algorithms
Average waiting and turnaround times → best for EP (high priority) and EP + RR and RR (low priority)

Response time between I/O → best for EP (high priority) and EP + RR

3. Conclusion

The simulation was able to successfully show the performance of the External Priorities, Round Robin with a 100ms time quantum, and combined preemptive EP/RR algorithms under different workloads. The analysis of the throughput, average wait time, average turnaround time, and average response time was calculated and shown in the report above.

The Round Robin algorithm used a quantum to time slice the CPU process to 100 ms, making it most effective for the combination of CPU-bound and I/O-bound processes. The External Priorities algorithm on the other hand, always letting the higher priority processes run without preemption from any other processes made it most effective for CPU-bound processes. The combined EP/RR algorithm worked very similar to the EP scheduling algorithm, except there was preemption every 100 ms.

In conclusion, the findings from the analysis of the three algorithms showed the importance of selecting the most optimal algorithm for the type of processes in the system.