SYSC4001 A3 P1 Report

Mark Bowerman 10127201
Joshua Heinze 101272848
2025-12-01

Introduction

This report analyzes the performance of three CPU scheduling algorithms implemented as part of an operating system simulator. External Priorities (non preemptive), Round Robin, and a hybrid of both.

It models realistic behavior through fixed CPU burst times, periodic I/O operations, a memory allocation subsystem using fixed partitions, and state transitions between new, ready, running, waiting, and terminated.

To test each algorithm, 20 simulation scenarios were executed per scheduler.Performance is measured using four metrics:

- Throughput
- Average Waiting Time
- Average Turnaround Time
- Average Response Time
- Memory Usage (optionally)

External Priorities (no preemption)

In this scheduling method, each process has a priority assigned from outside the system. The CPU always runs the highest priority ready process, and lower priority processes wait. This version does not preempt a running process so once one is running, it keeps the CPU until it finishes or performs I/O. EP works best for systems where important tasks must be completed quickly and predictable like real time systems.

Round Robin with a 100 ms timeout

In round robin scheduling, all ready processes share the CPU equally by taking turns in a fixed time slice or quantum. When a process uses up its time slice, it moves to the back of the ready queue, and the next process runs. RR is fair because no process can hog the CPU and it works best for interactive systems or situations where many processes need regular CPU attention. Its less effective for tasks that require long uninterrupted execution since they may be frequently paused.

A combination of both external Priorities and Round-Robin

This method combines priority scheduling with a time slice (quantum). The highest priority process runs first, but if it uses up a timeslice, it moves to the back of the ready queue. Also if a higher priority process becomes ready, the CPU is preempted and given to that process. EP RR benefits systems where you want both responsiveness for high priority tasks and fairness, so lower priority processes still get CPU time.