

Introduction

This report details the implementation and performance analysis of Graduate Systems (CSE638) PA01: Processes and Threads. The assignment explores process and thread creation in C, along with CPU, memory, and I/O intensive workloads. Measurements were taken using system tools like top and execution time, with automation via bash scripts. All code adheres to the 2000-loop requirement for worker functions.

Part C Analysis

Part C evaluates the six combinations of programs (A: processes, B: threads) and workers (cpu, mem, io) under default 2 workers. Metrics include CPU%, Memory usage (from top), and execution time (as IO proxy).

Table: Part C Measurements

Program+Function	CPU%	Mem	IO (Execution Time, s)
A+cpu	0	0	0.02
A+mem	0	0	0.54
A+io	0	0	199.73
B+cpu	0	0	0.01
B+mem	73.30	0	0.20
B+io	0.39	0	199.26

Analysis

- CPU and Memory Metrics: Low values indicate programs finish quickly; top may not capture short runs. B+mem shows higher CPU% (73.30%), possibly due to thread contention.
- Execution Time (IO): CPU/mem functions are fast (<1s), while I/O is slow (~200s) due to fsync forcing disk writes.
- Processes vs. Threads: Similar performance; threads may have slight overhead but better for shared resources like memory.

[Screenshot: Terminal output of script running Part C]

Part D Analysis

Part D scales workers (A: 2-5 processes, B: 2-8 threads) and plots metrics vs. worker count.

Plots Description

- CPU Plot: Time remains low and flat, as tasks are parallelizable.
- Memory Plot: Minimal increase, efficient allocation/deallocation.
- I/O Plot: Linear increase with workers, due to sequential disk access.

Analysis

Increasing workers shows CPU/mem saturation limits, while I/O highlights bottlenecks. Threads (B) scale better for CPU but similar for I/O.

[Insert Plots: MT25022_Part_D_cpu_plot.png, etc.]

[Screenshot: CSV data tables]

AI Declaration

The code was developed with assistance from GitHub Copilot for boilerplate, logic, and comments. All parts are understood and modified as needed.

GitHub Link

[Insert your GitHub repo URL here, e.g.,
https://github.com/jcharsha2003/GRS_PA01]