Carleton University
Department of Systems and Computer Engineering
SYSC 4001 Operating Systems Fall 2025

Assignment 1 – Section L3 Group 29

Brodie Macleod 101302779 **Basil Thotapilly (101313150)**

Github: https://github.com/auriza-jpg/Assignment-1_L3_29

PURPOSE AND INTRODUCTION

The objectives of the report are to analyze the performance of a small, simulated interrupt system. The simulator is given as an input, the trace of a single program. The trace follows the structure

Activity, Duration/Device Number

- CPU, X(ms)
- SYSCALL, Y
- CPU, Z(ms)
- IO_END Y
- …

Each activity in the trace is executed by calculating the time required and then incrementing the current system time.

PREFORMANCE

Input Files

To begin assing the performance of the interrupt system, Input files are needed. These files were divided into three categories, IO-Bound, CPU Bound and Balanced. 3 Input files were created for each. The 3 test files for bounded programs range from varying extremes regarding Device Delay / CPU ratio. Balanced Input files consist of shorter, moderate, and long device delay and CPU bursts respectively.

Category	Input File	CPU Burst (ms)	Device (No [Time ms])
IO-Bound	IO_1	100	1 [110]
IO-Bound	IO_2	100	2 [300]
IO-Bound	IO_3	100	3 [1000]
CPU-Bound	CPU_1	300	6 [152]
CPU-Bound	CPU_2	300	7 [265]
CPU-Bound	CPU_3	300	9 [1000]
Balanced	BAL_1	200	11 [145]
Balanced	BAL_2	200	12 [523]
Balanced	BAL_3	400	14 [636]

Files following the outlined device numbers following the Trace Format

Activity, Duration/Device Number

- CPU, X(ms)
- SYSCALL, Y
- CPU, Z(ms)

- IO_END Y
- … (loop 5 times)

A script was written to complete this.

Simulation Cases

To aid in understanding how the individual sections of the interrupt system affect execution time, all input files were executed across multiple system parameters. The combination of Input File and parameter is a Simulation Case. The parameters targeted were ISR activity time and context save time.

Summary and Analysis

From the Data Summary Document on GitHub

Context Time	CPU Total (≈)	IO Total (≈)	BAL Total (≈)
10 ms	4660–5790	2140–4140	3590–8700
20 ms	4760–5890	2240–4240	3690–8800
30 ms	4860–5990	2340–4340	3790–8900

All programs time increased with a flat 100ms for every 10ms of Context Time. The consistency across workload types shows how the time added by a context switch is unrelated to the duration of CPU or IO events, but the number of them.

The first noticeable aspect of ISR time is between the 50ms and 100ms jumps, the slowest tasks don't change, as device delays still take longer than the ISR. Though, eventually the ISR becomes so large it bounds the length of both previously IO bound, and CPU bound tasks. The fastest balanced tasks increase by nearly 4.5kms.

ISR Time	CPU Total (≈)	IO Total (≈)	BAL Total (≈)
50 ms	4660–5790	2140–4140	3590–8700
100 ms	4900–5790	2640–4140	3865–8700
390 ms	8990	6990	7990–10320