Carleton University

Department of Systems and Computer Engineering

SYSC 4001 Operating Systems Fall 2025

Assignment 1

Part II - Interrupt Handling Simulation: Analysis of System Performance and Overhead

Student 1 - Tamunoibuomi Okari Student ID - 101287549
Student 2 – Rounak Mukherjee Student ID - 101116888
Course – SYSC 4001 Operating Systems Professor - Gabriel Wainer
Deadline – October 6th, 8:00 AM
GitHub Repository: https://github.com/ibuomi/SYSC4001_A1

INTRODUCTION

Interrupt handling is fundamental to operating systems, allowing CPUs to respond to asynchronous hardware events. When interrupts occur, the system must suspend current execution, save context, execute an Interrupt Service Routine (ISR), and restore the interrupted process. This overhead significantly impacts performance. This report analyzes how context switching time, ISR duration, and vector table size affect system efficiency through comprehensive simulation testing.

METHODOLOGY

We implemented a C++ interrupt simulator modeling a single-CPU system processing trace files with CPU bursts, system calls (SYSCALL), and I/O completions (END_IO). Each interrupt executes: mode switch (1ms), context save (variable), vector lookup (1ms), ISR address load (1ms), ISR execution (device-dependent), and IRET (1ms). We conducted 20 experiments testing context times (10-50ms), ISR scaling (0.5x-2.5x), vector sizes (2, 4, 8 bytes), and combined effects. Five custom trace files (trace_1.txt through trace_5.txt) with varying interrupt patterns validated simulator behavior under different workload conditions.

RESULTS AND ANALYSIS

Context Save Time Impact (Experiments 1-5)

Experiment #	Context Time	ISR Scale	Total Time	CPU %	Overhead %	ISR %
1	10ms	1.0x	31,914ms	9.31%	2.63%	88.06 %

2	20ms	1.0x	32,514ms	9.14%	4.43%	86.43%
3	30ms	1.0x	33,114ms	8.98%	6.16%	84.86%
4	40ms	1.0x	33,714ms	8.82%	7.83%	83.35%
5	50ms	1.0x	34,314ms	8.66%	9.44%	81.90%

Key Findings: Context switching shows linear scaling—each 10ms increase adds 600ms total time (60 interrupts × 10ms). From 10ms to 50ms, total time increased 7.5% while overhead grew from 2.63% to 9.44% (3.6x increase). CPU utilization declined from 9.31% to 8.66% as overhead consumed more time. This demonstrates why hardware-assisted context switching is critical in modern systems.

ISR Duration Impact (Experiments 6-10)

Experiment #	Context Time	ISR Scale	Total Time	CPU %	Overhead %	ISR %
6	10ms	1.0x	31,914ms	9.31%	2.63%	88.06%
7	10ms	0.5x	17,854ms	16.65%	4.70%	78.65%
8	10ms	1.0x	31,914ms	9.31%	2.63%	88.06%
9	10ms	1.5x	45,974ms	6.46%	1.83%	91.71%
10	10ms	2.0x	60,016ms	4.95%	1.40%	93.65%

Key Findings: ISR duration has dramatic impact. Doubling ISR time (2.0x) nearly doubled total execution (88% increase) with ISR consuming 93.65% of system time and CPU only 4.95%. The relationship is nearly linear—ISR time directly determines total time. At 2.0x scale with 950ms average per interrupt, the system became interrupt-bound, spending almost all-time handling interrupts rather than productive work. This validates the OS design principle of keeping ISRs minimal.

TEST CASE VALIDATION

Test Case	Trace File	Interrupts	Total Time	CPU %	Overhead %	ISR %
1	trace_1.txt	10	3812 ms	20.67%	3.67%	75.66%

2	trace_2.txt	6	2410 ms	17.18%	3.49%	79.34%
3	trace_3.txt	66	28455 ms	12.27%	3.25%	84.48%
4	trace_4.txt	68	28377 ms	13.63%	3.35%	83.02%
5	trace_5.txt	72	31770 ms	12.46%	3.18%	84.37%
6	trace_6.txt	10	3940ms	16.95%	3.55%	79.49%
7	trace_7.txt	10	3181ms	19.71%	4.40%	75.89%
8	trace_8.txt	10	3943ms	13.67%	3.55%	82.78%
9	trace_9.txt	10	5572ms	12.96%	2.51%	84.53%
10	trace_10.txt	10	2172ms	30.29%	6.45%	63.26%
11	my_trace_intensive.txt	10(5 SYSCALL + 5 END_IO)	2880ms	8.7%	5.6%	85.7%
12	my_trace_sparse.txt	4(2 SYSCALL + 2 END_IO)	1860ms	29.0%	2.7%	68.3%

Key Findings: Test cases 1-10 validate the simulator across workloads ranging from 6 to 72 interrupts, with ISR execution consistently dominating (63-86% of total time). Custom test cases 11-12 demonstrate that sparse interrupts achieve 29% CPU utilization with 68.3% ISR overhead.

DISCUSSION

Three critical findings emerged: **(1) Context switching overhead is linear and predictable**—40ms context increase added 2,400ms total time with 60 interrupts. **(2) ISR duration has a dominant impact**—at 2.0x scale, 93% of time spent in handlers versus 5% productive work. **(3) Combined high overhead causes collapse**

CONCLUSION

Interrupt handling overhead significantly impacts system performance with ISR duration as the dominant factor. Context switching scales linearly (7.5% increase for 5x time), ISR duration shows dramatic impact (88% increase at 2x scale), combined factors reduce CPU utilization below 4%, and vector size has negligible effect. These findings validate that efficient interrupt handling requires both architectural support (fast context switching) and software discipline (minimal ISR execution).