Interrupt Simulator Test Report
SYSC 4001 – Operating Systems – Assignment 1
Student 1: Faiaz Ahsan 101120268
Student 2: Nicky Fang 101304731
10/06/2025
Gibhub Repository: https://github.com/faiazahsan/SYSC4001_A1

File	Description
Trace	Basic simulation with 3 CPU bursts and 2 I/O operations
Trace_01	Moderate workload with 5 I/O cycles
Trace_02	Shorter simulation with 3 CPU bursts interrupted by I/O operations
Trace_03	Extensive stress test with 32 CPU I/O cycles, testing high frequency interrupts
Trace_04	Heavy I/O workload with 30 interrupt cycles
Trace_05	Complex mixed workload with 28 interrupt cycles
Trace_06	System with short CPU bursts interrupted frequently by I/O operations
Trace_07	CPU intensive system with minimal I/O requirements
Trace_08	Tests concurrent SYSCALL handling with frequent system calls
Trace_09	Simulates scenario with both a fast and slow device with moderate CPU bursts
Trace_10	Repeated device testing with 5 consecutive operations on same device
Trace_11	Alternating device pattern using 2 device
Trace_12	Very short burst simulation with minimal CPU work
Trace_13	Large scale processing simulation with substantial CPU bursts
Trace_14	Incremental workload with CPU burst durations that increase by 10ms each cycle and I/O on sequential devices
Trace_15	Descending CPU workloads with bursts decrementing by 10ms each cycle
Trace_16	Randomized pattern with irregular CPU bursts
Trace_17	Multiple concurrent interrupts on a single device to test how system handles multiple interrupts
Trace_18	Uniform workload with consistent CPU bursts and sequential I/O operations
Trace_19	Overlapping interrupts with SYSCALLs
Trace_20	Minimal burst stress test with short bursts between I/O operations
Trace_21	Long CPU bursts and minimal I/O interruptions
Trace_22	Multiple interrupts in quick succession

Trace_23	Irregular pattern with varying CPU burst duration
Trace_24	High frequency of interrupts with short but consistent CPU bursts
Trace_25	Standard balanced simulation with large CPU bursts

Summary:

This report analyzes the performance of interrupt handling through simulation testing written in C. The simulator processes trace files representing various program execution patterns with CPU bursts and interrupts, measuring the impact of different interrupts on overall system performance.

Impact of save/restore context time:

With context time varying from 10-30ms, this results in a linear relationship with overall execution times. For high frequency interrupt test case such as trace_03.txt, increasing context time from 10 to 30ms resulted in a total increase of interrupt handling time.

ISR activity time impact:

ISR execution time becomes the main increasing factor of execution times when varied between 40 and 200ms. When ISR time exceeds the device I/O delays it becomes the dominant bottle and correlates to increasing execution times in complex test cases such as trace_21.txt.

Additional Questions:

Difference in speed and execution times:

The difference in speed in interrupt steps result in uneven impacts. Save/restore hurts performance the most and ISR execution time matters mainly for I/O heavy workloads, while mode switching, and vector operations have negligible effects.

Impacts of multiple concurrent interrupts:

Multiple SYSCALLs on a single device resulted in slower execution times due to additional processing. Overlapping SYSCALLs on multiple devices resulted in quicker execution.

What if we vary I/O times significantly?

By increasing the delay of devices, this resulted in the system waiting for slower devices and making interrupts more impactful on overall execution times. The bottleneck shifted from CPU to the devices.