Assignment 1 Report - Lucas Baker & Amit Kunkulol

The following report summarizes the results of tests run on our interrupt system simulator, demonstrating the interrupt process. The following are 25 executed test scenarios (including the 5 given in the sample). The 20 created have been done using 5 different input files “trace_test[1-5].txt”, which are visible in our repository. **In the files, every 5 lines of the trace file corresponded to one individual test**, representing a sequence of CPU bursts, system calls (SYSCALL), and END_IO events. The original comments marking each test were removed for submission, but the structure remained consistent. Modifications to context save durations and ISR times were applied to assess their impact.

Figure 1: Overview of Tests Run

Trace File	Tests Covered	Context Save (ms)	ISR Duration
1	1-4	10	default
2	5-8	20	default
3	9-12	10	200ms
4	13-19	30	default
5	17-20 (CPU-Only)	N/A	N/A
Sample Tests Given (1-5)	N/A	10	default

Using a python script to process the outputted text files, we were able to obtain the following statistics:

Figure 2: Total Group Test Values

Test Group	Context Save (ms)	Total Execution Time (ms)	Total CPU Time (ms)	Total ISR Time (ms)	Total Overhead Time (ms)	CPU Utilization
1-4	10	2269	530	1661	60	23.36%
5-8	20	1416	525	771	100	37.08%
9-12	10	1797	485	1227	60	26.99%
13-19	30	2142	470	1515	140	21.94%
17-20	N/A	470	470	0	0	100.00%
Sample Test 1 (trace_1)	10	2262	713	1442	75	31.52%

Sample Test 2 (trace_2)	10	1357	319	956	45	23.51%
Sample Test 3 (trace_3)	10	16292	3443	12020	495	21.13%
Sample Test 4 (trace_4)	10	16469	3781	11779	510	22.96%
Sample Test 5 (trace_5)	10	18266	3944	13402	540	21.59%

Note: CPU Utilization = 100 x (Total CPU time)/(Total Execution time)

Analysis:

- Effect of Context Save
    - Increasing from 10 ms to 20 ms to 30 ms increases total execution time due to more time spent saving and restoring CPU context during each interrupt.
    - CPU utilization drops with higher context save times because more time is spent in overhead rather than productive CPU bursts.

- Effect of ISR Execution Time
    - Devices with higher ISR delays filled up the total execution time the most, delaying CPU execution and I/O completion. Essentially, longer ISR durations block CPU progress

- Impact on Overall Execution
    - Overhead (switching kernel mode, context save/restore, finding vector, loading address, IRET) is small compared to ISR times but increases a lot with frequent interrupts.
    - Small CPU bursts can reduce CPU utilization.

- Overall:
    - System design should minimize ISR duration and context save time.
    - Devices with extremely long ISR times can be a bottleneck.
    - The final set of tests (CPU-Only) gave a good baseline for comparison. With this, we were able to notice how interrupt handling introduces a significant overhead.

In conclusion, the simulator paired with the tests written have shown us that the CPU utilization ranges from 23% to 100%, depending on ISR load and context switch time. Additionally, it is the ISR execution time that dominates most of the total time, so reducing that is the key.