Test case and Results

Category 1: Context Save Time

Objective: Measure effect of context save time on total execution time. **Input file:** `trace_01.txt`

Test	Output file	Context Save Time (ms)	Total Execution Time (s)
1	Execution_01.0.txt	10	38.12
2	Execution_01.1.txt	20	39.12
3	Execution_01.2.txt	30	40.12

Observation: As context save time increases, total execution time grows proportionally, indicating that context switching overhead directly impacts the performance.

Category 2: ISR Activity Time Variation (CPU_SPEED 100)

Objective: Analyze effect of ISR delay time on runtime. **Input file:** `trace_02.txt`

Test	Output file	ISR Delay (ms)	Total Execution Time (s)	# of ISR log steps
4	Execution_02.0.txt	40	24.1	Many (5–6 per device event)
5	Execution_02.1.txt	100	24.1	Moderate (2–3 per device event)
6	Execution_02.2.txt	200	24.1	Few (1–2 per device event)

Observation: Changing the ISR delay time did not affect the overall runtime, which stayed constant at 24.1 seconds. This is because ISR activity time was given to us in the device table. The program's speed did not change because the device table already accounted for the ISR delays. The way we interpreted this was: if the delay time is longer than the device average time, then the ISR delays will be split evenly in between, "SYSCALL: run the ISR (device driver)" and "transfer data from device to memory" assuming there were no other ISR tasks.

Category 3: Vector size (CPU_SPEED 100)

Objective: Measure effect of vector size on execution time. **Input file:** `trace_03.txt`

Test	Output file	Vector Size (bytes)	Total Execution Time (s)
7	Execution_03.0.txt	2	24.1
8	Execution_03.1.txt	4	24.1

Observation: Changing the vector size from 2 bytes to 4 bytes did not affect execution time because the lookup in the vector table and the calculations is still the same, the only difference is there is more memory space so there can be more routines.

Category 4: CPU speed

Objective: Measure effect of CPU frequency on execution time. **Input file:** `trace_05.txt`

Test	Output file	CPU Speed	Total Execution Time (s)
9	execution_05.0.txt	100Hz	317.74
10	execution_05.1.txt	1000 Hz	31.774
11	execution_05.2.txt	1000000Hz (1MHz)	0.031774
12	execution_05.3.txt	1000000000Hz (1Ghz)	0.000032

Observation: Here we test the speed of the CPU and how it affects the number of operations it is able to compute. The first test used was the value of 100Hz which was already inputted on the header file for us. We then upped the speed 10x which reflected in the execution time dropping it proportionally (it was 10x faster). The last CPU speed we tested was 1GHz as CPU's nowadays are running at those speeds. Comparing those results shows how much technology has changed in the last 80 years. The trace 5 program would've taken a little over 5 minutes back then but today that task would be accomplished in less than a millisecond.

Category 5: CPU heavy work (CPU_SPEED 100)

Objective: Analyze performance for CPU-intensive workloads.

Test	Input file	Output file	Total Execution Time (s)
13	trace_06.txt	execution_06.0.txt	12
14	trace_07.txt	execution_07.0.txt	10
15	trace_08.txt	execution_08.0.txt	10

Observation: For CPU-heavy workloads, the execution time was primarily determined by the length of CPU bursts. Interrupt overhead was negligible because there were few or no I/O operations. This shows that in compute-intensive programs, optimizing CPU speed has the greatest impact, while interrupt handling plays a minor role.

Category 6: I/O heavy work (CPU_SPEED 100)

Objective: Analyze performance for I/O-intensive workloads.

Test	Input file	Output file	Total Execution Time (s)
16	trace_09.txt	execution_09.0.txt	7.04
17	trace_10.txt	execution_10.0.txt	9.5
18	trace_11.txt	execution_11.0.txt	6.81

Observation: I/O-heavy workloads had shorter CPU bursts but frequent device interrupts, which introduced significant ISR overhead. The total execution time was dominated by device delays rather than CPU processing. This reflects real-world systems where slow peripherals can bottleneck performance even if the CPU is fast.

Category 7: Mixed workload (CPU_SPEED 100)

Objective: Analyze performance for workloads with balanced CPU and I/O.

Test	Input file	Output file	Total Execution Time (s)
19	trace_12.txt	execution_12.0.txt	8.5
20	trace_13.txt	execution_13.0.txt	6.78
21	trace_14.txt	execution_14.0.txt	12.86

Observation: Mixed workloads exhibited a balance between CPU bursts and I/O operations. Execution time varied depending on interrupt frequency and device delay. This scenario is closer to real applications, where both compute and I/O tasks coexist, making interrupt handling efficiency critical for overall performance.

Link to repository: https://github.com/Hexax/SYSC4001-Assignment-1