Report - Part II

Yousif Al-Satari 101317551
Hamza Mohammed Salih 101311021

Introduction:

This experiment uses our interrupt simulation system that we built. The simulator reads three input files - vector_table.txt, device_table.txt, and trace.txt and produces an output file, execution.txt. Each line in execution.txt records an event along with its start time and duration in milliseconds.

The goal is to analyze how different factors affect total execution time, the factors we will be looking at are:

- Changing context save/restore time (10ms, 20ms, 30ms)
- Modifying device speeds in the device table
- Varying the number of interrupts and CPU workload patterns

By running several workloads and comparing their total durations, we can clearly see how context switch and device speed affects system performance.

Test cases:

We will be analyzing the performance characteristics of the interrupt simulation system we built by taking a look at how different factors affect overall execution time.
Some of these factors include:

- Context switching time variations (10ms, 20ms, 30ms)
- Device table timing
- How many interrupts are in the scenario
- Time spent in interrupt service routines

I have 5 trace files with different scenarios and workload patterns such as having a light workload (short CPU bursts) or heavy workload (long CPU bursts) and I/O intensive (6 interrupts) and then changed the context switch time and the device table i will explain the different workloads and how it got affected

1.light workload

We have short cpu bursts and only 2 interrupts
CPU, 30
SYSCALL, 7
CPU, 20
END_IO, 7

It is a simple sequence with I/O operations on device 7 when context time changes from 10ms to 30ms the execution time increases by 40ms because of the 20 ms increase in the context time. This is predictable linear scaling with context time changes. But with the device speed change because there is only one device involved, making the speed faster will reduce the waiting time during I/O operations, but the impact is relatively small due to low interrupt frequency and these can be clearly observed in the different execution files.

2. Heavy workload

Here, there are long CPU bursts with 4 interrupts using device 14 and 19.

CPU, 100
SYSCALL, 14
CPU, 80
END_IO, 14
CPU, 60
SYSCALL, 19
CPU, 40
END_IO, 19

Simulates computational work with moderate I/O activity. When we increase the context time from 10ms to 30ms, we observe an 80ms increase because we have 4 interrupts. This becomes noticeable, but it's still proportionally small to the total CPU work. For the device speed, using faster devices in execution 17 shows improved performance because there are multiple I/O operations. Having slower devices will create longer waiting times between computing phases.

3. Multi-device workload

Here we have medium CPU bursts and 4 interrupts with two devices 7 and 14

CPU, 40
SYSCALL, 7
CPU, 30
SYSCALL, 14
CPU, 25
END_IO, 7
CPU, 35
END_IO, 14

We see a similar 80ms increase as the heavy workload, but here its way more significant, as the cpu bursts are shorter, so context switching weighs more in the total percent. Increasing the device speeds here significantly benefits, resulting in improved performance.

4. CPU-Intensive workload

Here we have very long CPU bursts with 2 interrupts

CPU, 200
SYSCALL, 14
CPU, 150
END_IO, 14

CPU, 180
The context switch increase shows the smallest relative impact 40ms because interrupt overhead is considered nothing by the 530ms of total cpu work. It also shows minimal sensitivity to device speed changes since I/O operations are a tiny fraction of the total time.

5. I/O intensive workload
Here we have very short cpu bursts with 6 interrupts across 3 devices
CPU, 10
SYSCALL, 19
CPU, 15
SYSCALL, 7
CPU, 12
SYSCALL, 14
CPU, 8
END_IO, 19
CPU, 5
END_IO, 7
CPU, 10
END_IO, 14
Context time increase shows the largest impact (120ms) because it has the highest amount of interrupts. Switching context dominates the execution time when CPU bursts are very short and when we have a lot of interrupts. Here also device speed also really affects performance; fast devices improve performance because I/O operations happen frequently. Slow devices create significant problems that compound across multiple interrupts.

Conclusion:

Workloads that have more interrupts show higher sensitivity to context switching time. The I/O workload suffers most from an increase in context time, while the CPU-intensive workload doesn't get affected or is barely affected. The relationship is linear: doubling context time roughly doubles the interrupt-related overhead.
For device speed, I/O-intensive and multi-device workloads have the most benefit from faster devices, while CPU-intensive workloads show small improvement. When devices are very fast compared to context switching time, context overhead becomes the dominant factor. When devices are slow, I/O wait time dominates.
In the combined tests (tests 16-20), workloads with both high interrupt frequency and slower devices show compounded performance degradation. Fast devices can partially offset the impact of slower context switching, and vice versa.