SYSC 4001 - OPERATING SYSTEMS

ASSIGNMENT 1

GROUP - 22

Student 1 - **Peter Tourkoyiannis**

- 101262960

Student 2 - **Akshwin Sellathurai**

- 101225500

Introduction

The goal of this assignment is to simulate and collect and analyze data from how interrupts affect general scheduling. The tests simulate different save times and ISR activity times from 10,20,30 and 40, 60, 80, 100, 120, 160, 200 respectively.

Table 1: Data collected from the saving times and the ISR chunk times.

Test Number	Save Time (ms)	ISR Chunk Time (ms)	Total Time (ms)
1	10	40	30965
2	10	60	31125
3	10	80	31405
4	10	100	31765
5	10	120	32085
6	10	160	32205
7	10	200	32965
8	20	40	31565
9	20	60	31725
10	20	80	32005
11	20	100	32365
12	20	120	32685
13	20	160	32805
14	20	200	33565
15	30	40	32165
16	30	60	32325
17	30	80	32605
18	30	100	32965
19	30	120	33285
20	30	160	33405
21	30	200	33405

Github Link To Code

https://github.com/Peter-Tourkoyiannis/SYSC4001_A1.git

Observations

After observing the data, it is clear that increasing the overhead and ISR time increases the total time. It seems that increasing the overhead increases it uniformly while increasing the ISR time increases it in an exponential manner.

Increasing the address bytes from 2 to 4 should not change the time in the system created as it is dealt with in the boilerplate. In practice, assuming the system can handle a larger address, would take longer as the hardware would need to take more time to get to the address (Think bigger but same speed memory etc.) It should be noted though that it would not increase the time significantly.

A faster CPU would allow for more instructions to pass through making it faster. However, this would require that it is not constantly waiting on external factors, such as I/O, as that is where the main bottleneck would be.

Note: The way the code was originally set up (NOT THE FINAL VERSION), I had used a function to stop the ISR when it was done (I believe the min, can't remember), even when the ISR chunk was not completed yet. However, the final version was updated to run the entire chunk even if the SYSCALL duration had expired. This was done as the same value was outputted for each save time. This led to the chuck time having no impact, hence the change.

Conclusion

The program successfully processes from an input file, simulate SYSCALLS, I/O and CPU Burst cycles. With the data we can prove that the more overhead and the longer chunk times result in a more inefficient system