SYSC4001 Report

Yuvraj Bains
Jared St.Louis
https://github.com/Yuvi-Pain/SYSC4001_Ass1/tree/main

Change the value of the save/restore context time from 10, to 20, to 30ms. What do you observe?

The entire runtime grows roughly linearly with the number of interrupts when context save/restore is increased from 10 to 20 to 30. The fact that context overhead directly grows with interrupt frequency is confirmed by the bigger absolute and percentage increases seen in I/O-bound traces.

Vary the ISR activity time from between 40 and 200, what happens when the ISR execution takes too long?

Additional interrupts may be served more slowly or not at all if an ISR takes too long. This could lead to missed interruptions, higher interrupt latency, lower real-time performance, and system unresponsiveness.

How does the difference in speed of these steps affect the overall execution time of the process?

The CPU first saves the current process's status in the Process Control Block before executing an interrupt service routine. The less time that passes before the ISR starts, the faster the context is preserved. The ISR then handles the interruption; the longer this process takes, the more delays there are in other processes or interruptions. After the ISR is finished, the CPU restores the process state from the PCB, enabling the paused process to continue. Faster context saving and ISR execution durations while preserving system responsiveness.

What happens if we have addresses of 4 bytes instead of 2?

The vector table's size doubles when the address size is doubled from two to four bytes because each address requires 32 bits of memory instead of just 16 bits. The address lookup vector table needs a larger handling address for the overhead, which could make memory-related operations take longer.