Assignment 1 Report

SYSC 4001: Operating Systems

Jonathan Gitej (101294584)
Atik Mahmud(101318070)

GitHub Repo: https://github.com/atikmahmud24/Sysc4001-Assignment1/tree/main

1. INTRODUCTION

The goal of this assignment is to design and implement an Interrupts Simulator. This interrupt simulator is meant to replicate the interaction between the CPU, system calls, and I/O devices, demonstrating timing and performance effects of different stages in the interrupt handling process. By using an input trace that represents a program’s sequence of CPU bursts and I/O operations, the simulator reproduces the transitions between user mode and kernel mode, context saving and restoring, ISR (Interrupt Service Routine) execution, and IRET completion. The implementation was completed in C++, based on the provided template files. A series of at least twenty test cases were executed, using different combinations of device delays, ISR activity durations, and context-switch times.

2. DISCUSSION

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

When changing the value of the context time from 10 to 20 to 30 it was observed that the higher context time would lead to a higher execution time. This is because when you run an interrupt it adds the context time for each interrupt since it has to save before entering the interrupt and once it is finished with the interrupt it must restore. If you have lots of interrupts this time would quickly add up.

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

When changing the ISR activity time it was found that the higher value would increase execution time. This is because the ISR activity time is added whenever an interrupt occurs. If the ISR execution takes too long it stops the CPU from performing other user tasks since it must first complete the ISR.

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

Overall these steps should be as fast as possible since they both greatly affect the speed of the process. The context time should be fast since it must save and restore all data before and after an interrupt and the ISR activity must be fast since you do not want to be stuck doing an interrupt and halting your CPU from continuing with the user program. Therefore these factors have a direct correlation to the process. Where the faster they are the faster and process and the slower they are the slower the process.

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

If we change addresses to 4 bytes instead of 2 it will double each entry in the vector table. This now means the interrupt controller must read 4 bytes per vector instead of 2, otherwise it will cause errors leading to incorrect jumps.

What if we have a faster CPU?

With a faster CPU it was found that it only affects the execution of normal tasks that do not involve I/O. This is because the interrupt handler and ISR run independently from the CPU. But normal tasks get completed faster with a faster CPU.

3. CONCLUSION

The interrupt simulator demonstrates how various factors in the interrupt handling process impact overall system performance. Increasing the context save/restore time directly increases execution time, as each interrupt requires additional overhead. Similarly, longer ISR activity times delay the CPU from executing user tasks, further slowing the process making efficient interrupt handling a key aspect in increasing system performance.