

Assignment 1

SYSC 4001A (L3)

Student 1: Nicholas Dorscht (101298132)

Student 2: Eshal Kashif (101297950)

Part II - Design and Implementation of an Interrupts Simulator

Introduction

This assignment served as a way to discover and explore the concept of hardware interrupts by going through the steps a short program goes through when it is executed. It goes through the different states of executing a system call, including switching execution modes, saving context, and processing values from the ISR table. Additionally, it explores how each of these steps affects the runtime of the program in different ways, and how changing some values may have a more dramatic effect than others.

Discussion

By varying the values of certain variables, such as the length of time the ISR takes to run, as well as the amount of time the CPU takes to save/restore context, we were able to gather data on how these changes affect the overall runtime in different ways. Additionally, we asked ourselves some questions as to how different factors that are not necessarily software related can also affect execution time.

One of the first changes we explored was the effect of changing the time it takes to save and restore context. While this may seem trivial, as its execution time is short compared to other delays such as device drivers, for every interrupt context must be saved and restored. This means that the effect of doubling the time is seen twice which can lengthen execution of a program by entire seconds depending on the length of the program. Another change we tested was the time it takes for the processor to execute the ISR. While this change is only seen once per interrupt, a much longer ISR execution time delays the entire system as the processor has to wait to execute the ISR before it can move onto the next interrupt. This means that having longer execution times can result in more and more interrupts stacking up. Overall, we found that these changes both lengthened the amount of time it took for the program to execute then they were increased.

Examples of a test case where we changed the context save/restore time can be seen below. In Figure 1, we use a context save/restore time of 10ms, which results in a total time of 524ms at the end of the screenshot. In Figure 2, a context save/restore time of 30ms is used resulting in a total time at the end of the image of 584ms (60ms more, since $30-10=20$ and we save twice and restore once here), causing longer execution of time.

```

0, 1, switch to kernel mode
1, 10, context saved
11, 1, find vector 18 in memory position 0x0024
12, 1, load address 0X060A
into the PC
13, 123, SYSCALL ISR: call device driver
136, 200, handle ISR (activity time)
336, 1, IRET
337, 10, Restore context
347, 1, Switch to user mode
348, 164, CPU
512, 1, switch to kernel mode
513, 10, context saved
523, 1, find vector 18 in memory position 0x0024
524, 1, load address 0X060A

```

Fig. 1: Trace file showing longer ISR and context times (context time=10ms).

```

0, 1, switch to kernel mode
1, 30, context saved
31, 1, find vector 18 in memory position 0x0024
32, 1, load address 0X060A
into the PC
33, 123, call device driver
156, 200, handle ISR
356, 1, IRET
357, 30, Restore context
387, 1, Switch to user mode
388, 164, CPU
552, 1, switch to kernel mode
553, 30, context saved
583, 1, find vector 18 in memory position 0x0024
584, 1, load address 0X060A
into the PC

```

Fig. 2: Trace file showing longer ISR and context times (context time=30ms).

```

0, 1, switch to kernel mode
1, 10, context saved
11, 1, find vector 18 in memory position 0x0024
12, 1, load address 0X060A
into the PC
13, 123, SYSCALL ISR: call device driver
136, 200, handle ISR (activity time)
336, 1, IRET
337, 10, Restore context
347, 1, Switch to user mode
348, 164, CPU
512, 1, switch to kernel mode
513, 10, context saved
523, 1, find vector 18 in memory position 0x0024
524, 1, load address 0X060A

```

Fig. 1: Trace file showing longer ISR and context times (context time=10ms).

```

0, 1, switch to kernel mode
1, 30, context saved
31, 1, find vector 18 in memory position 0x0024
32, 1, load address 0X060A
into the PC
33, 123, call device driver
156, 200, handle ISR
356, 1, IRET
357, 30, Restore context
387, 1, Switch to user mode
388, 164, CPU
552, 1, switch to kernel mode
553, 30, context saved
583, 1, find vector 18 in memory position 0x0024
584, 1, load address 0X060A
into the PC

```

Fig. 2: Trace file showing longer ISR and context times (context time=30ms).

We achieve a similar result by changing ISR activity times. In some cases, long ISRs can cause a timeout, potentially causing data loss or crashes.

Another non-software factor that could affect execution time is the speed of the CPU being used. Having a faster CPU could shorten execution time, but not in the same way as the software. While better software can decrease the amount of clock cycles needed to run the program, a

faster CPU will run the same amount of cycles as a slower one. Instead, it shortens execution time with a faster clock, allowing it to process commands at a faster rate.

Github

https://github.com/nicdorscht/SYSC4001_A1