SYSC 4001-A

Lab Section: L1

Assignment 1 Report – Group Submission

Link To Repository: https://github.com/aliBundook/SYSC4001_A1

Student 1: Ali Asghar Aqeel Bundookwalla (101299213)

Student 2: Mohamed Gomaa (101309418)

Date: October 6, 2025

Overview

For every simulation, the same process always occurs. CPU, system call, and I/O. The difference comes in the time it takes for each of those activities to execute. The time it takes the CPU is always random, because the CPU execution can either be long or short depending on the program fed to it. After that finishes, switching to kernel mode takes 1 ms, saving and restoring context takes 10 ms, and finding the vector address and loading it takes 2 ms (1 ms each). With system calls, the time it takes for a single ISR activity is constant at 40 ms. The reason why some system calls for the devices would have more ISR activities being executed is because every device has its own delay time, which is the time it should take for that device to finish running its execution. This is why this is different for all devices. END I/O always has a time of 1 ms, and this comes from the fact that after the I/O device finishes running its ISR, it must send an interrupt to the CPU to inform it of its completion. In our output table, this is where the IRET execution comes from. IRET stands for Interrupt Return. After the ISR activities run for the system calls, sometimes we get the remaining delay time for that specific device. The reason this happens is because every ISR activity takes 40 ms to complete; when a device has time left that is less than 40 ms, it cannot run another ISR activity; therefore, that time is known as our remaining delay time for that device.

What happens when we change the context time from 10 to 20 to 30 ms:

What we observe is that longer context time means longer duration of the interrupt handling, which means more overhead for the whole program execution. The whole program duration increases more when you go from 10 to 20 to 30 ms.

What happens when the ISR activity time from between 40 and 200 ms:

After observing multiple scenarios where we changed the time from 40 to 100 to 150 to 200 ms. We can clearly see that the devices perform fewer ISR activities because each activity now would take much more time to execute and therefore as a result, we would have a much larger remaining delay time available. Another setback is the fact that longer ISR activity time means slower interrupt handling and more time wasted in kernel mode which generally makes the system performance worse.

For the purposes of the report length, below is the analysis of the first 10/20 simulation cases.

Analysis of results of Simulation 1:

For Simulation 1, We are using devices 12 (523 ms), 9 (1000 ms), 19 (123 ms), 14 (636 ms), 8 (152 ms), 5 (250 ms), 13 (145 ms), 7 (265 ms), 4 (300 ms), 11 (564 ms), and 17 (956 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 1, we can see that device 12 ran 13 ISR activities and has a remaining delay time of 3. That is because each activity is 40 ms and we had 13 different activities within the ISR body so that takes a total of 520 ms, then we have an extra 3 ms remaining delay time at the end accounting for a total 523 ms. This methodology applies for the rest of the device's activities within the Simulation 1 trace input file.

Analysis of results of Simulation 2:

For Simulation 2, we are using devices 1 (110 ms), 4 (300 ms), 18 (235 ms), 11 (564 ms), 12 (523 ms), and 19 (123 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 2, we can see that device 1 ran 2 ISR activities and has a remaining delay time of 30. That is because each activity is 40 ms and we have 2 different activities within the ISR body so that takes a total of 80 ms, then we have an extra 30 ms remaining delay time at the end accounting for a total 110 ms. This methodology applies for the rest of the device's activities within the Simulation 2 trace input file.

Analysis of results of Simulation 3:

For Simulation 3, we are using devices 17 (956 ms) and 1 (110 ms), and 13 (145 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 3, we can see that device 17 ran 23 ISR activities and has a remaining delay time of 36. That is because each activity is 40 ms and we had 23 different activities within the ISR body so that takes a total of 920 ms, then we have an extra 36 ms remaining delay time at the end accounting for a total 956 ms. This methodology applies for the rest of the device's activities within the Simulation 3 trace input file.

Analysis of results of Simulation 4:

For Simulation 4, we are using devices 20 (652 ms), 9 (1000 ms), 3 (150 ms), and 16 (68 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 4, we can see that device 20 ran 16 ISR activities and has a remaining delay time of 12. That is because each activity is 40 ms and we had 16 different activities within the ISR body so that takes a total of 640 ms, then we have an extra 12 ms remaining delay time at the end accounting for a total 652 ms. This methodology applies for the rest of the device's activities within the Simulation 4 trace input file.

Analysis of results of Simulation 5:

For Simulation 5, we are using devices 7 (265 ms), 11 (564 ms), 18 (235 ms), and 16 (68 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 5, we can see that device 7 ran 6 ISR activities and has a remaining delay time of 25. That is because each activity is 40 ms and we have 6 different activities within the ISR body so that takes a total of 240 ms, then we have an extra 25 ms remaining delay time at the end accounting for a total 265 ms. This methodology applies for the rest of the device's activities within the Simulation 5 trace input file.

Analysis of results of Simulation 6:

For Simulation 6, we are using devices 9 (1000 ms), 12 (523 ms), and 15 (456 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 6, we can see that device 9 ran 25 ISR activities and has a remaining

delay time of 0. That is because each activity is 40 ms and we had 25 different activities within the ISR body so that takes a total of 1000 ms, then we have an extra 0 ms remaining delay time at the end accounting for a total 1000 ms. This methodology applies for the rest of the device's activities within the Simulation 6 trace input file.

Analysis of results of Simulation 7:

For Simulation 7, we are using devices 1 (110 ms), 8 (152 ms), 5 (250 ms), and 3 (150 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 7, we can see that device 1 ran 2 ISR activities and has a remaining delay time of 30. That is because each activity is 40 ms and we have 2 different activities within the ISR body so that takes a total of 80 ms, then we have an extra 30 ms remaining delay time at the end accounting for a total 110 ms. This methodology applies for the rest of the device's activities within the Simulation 7 trace input file.

Analysis of results of Simulation 8:

For Simulation 8, we are using devices 19 (123 ms), 9 (1000 ms), and 14 (636 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 8, we can see that device 19 ran 3 ISR activities and has a remaining delay time of 3. That is because each activity is 40 ms and we had 3 different activities within the ISR body so that takes a total of 120 ms, then we have an extra 3 ms remaining delay time at the end accounting for a total 123 ms. This methodology applies for the rest of the device's activities within the Simulation 8 trace input file.

Analysis of results of Simulation 9:

For Simulation 9, we are using devices 16 (68 ms), 6 (211 ms), 4 (300 ms), 15 (456 ms), 9 (1000 ms), 11 (564 ms), and 18 (235 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 9, we can see that device 16 ran 1 ISR activity and has a remaining delay time of 28. That is because each activity is 40 ms and we had 1 different activity within the ISR body so that takes a total of 40 ms, then we have an extra 28 ms remaining delay time at the end accounting for a total of 68 ms. This methodology applies for the rest of the device's activities within the Simulation 9 trace input file.

Analysis of results of Simulation 10:

For Simulation 10, we are using devices 20 (652 ms), 5 (250 ms), 18 (235 ms), 9 (1000 ms), 2 (100 ms), 8 (152 ms), 3 (150 ms), 15 (456 ms), and 8 (152 ms). What we have in brackets is the time it takes for each device to complete its execution of the ISR routine. In our output table for simulation 10, we can see that device 20 ran 16 ISR activities and has a remaining delay time of 12. That is because each activity is 40 ms and we had 16 different activities within the ISR body so that takes a total of 640 ms, then we have an extra 12 ms remaining delay time at the end accounting for a total 652 ms. This methodology applies for the rest of the device's activities within the Simulation 10 trace input file.