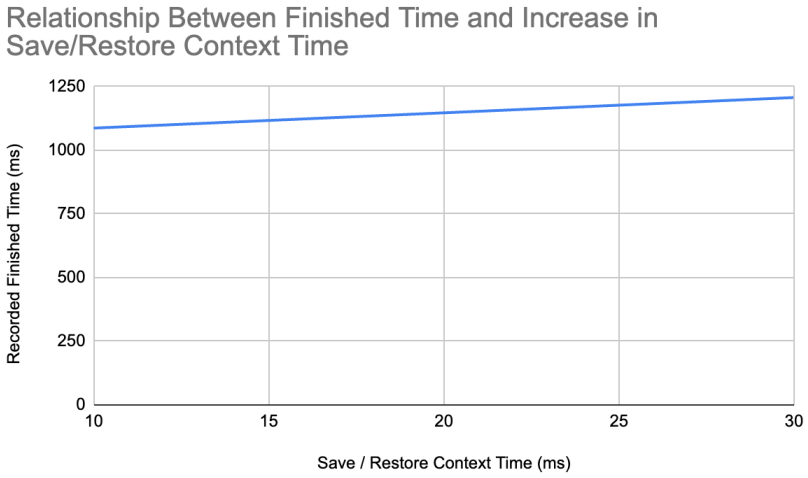
Assignment 1 - Part II - Report
SYSC 4001

Samy Kaddour - 101258499
Mithushan Ravichandramohan - 101262467

GitHub Repository:
https://github.com/mith3x/SYSC4001_A1

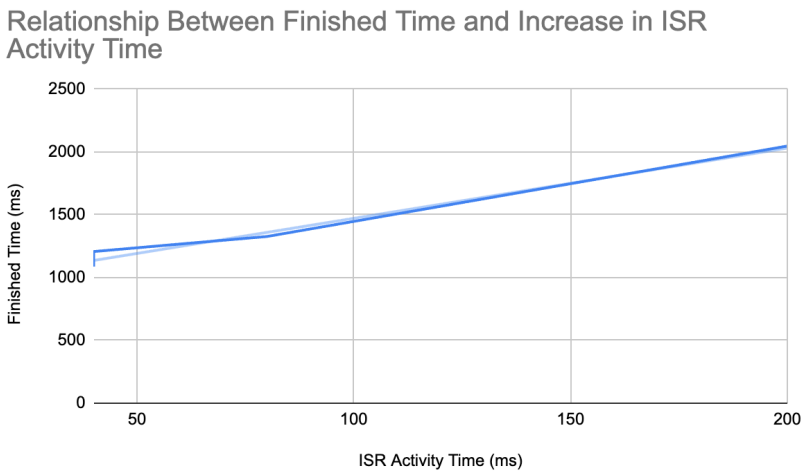
Throughout this simulation, we had run 20 test cases that helped us conclude the effects and influences of changing around various variables. Our first first few tests were centered around changing the value of the save/restore context time (*CSR_var in program*), and observing the influence it had on execution time. We had used a constant trace file, along with constant ISR values (*40 ms*). Below is a chart ***[Chart 1]*** that represents our obtained results.



[Chart 1]

When changing the save/restore context time, we noticed that the increase had a direct linear impact on the execution time. The first few tests gave us a good base of knowledge of just how linear the effect is. It was through later tests when we noticed that the more interrupts we had, the more save/restore context time had an impact on the execution time. This makes sense, as whenever an interrupt occurs, a switch is made to kernel mode. This means that a save/restore must be made every time the switch happens. Therefore we concluded that changing the save/restore context time had a more noticeable impact when many interrupts occur, it is minimal otherwise.

The next few tests were made to set a base analysis for what the impacts of changing the value of ISR activity time had on the execution time. We used the same trace files used in the previous tests, and used a constant CSR value (*10ms*). Below is a chart ***[Chart 2]*** that represents the results.



[Chart 2]

The first thing that was noticed after these results, was that there was a skyrocket of a difference in the final execution time when changing the ISR value. This was expected though, as this is the handling time of the interrupts. ISRs are very frequent, thus when there is a high value such as 200 ms, the CPU is being occupied for much longer thus causing a 'bottleneck'. Similar to the CSR, we had also concluded after later tests that the more interrupts that occur, the more the execution time is. The difference between the changing the ISR and changing the CSR values, was that the ISR never had an initial 'minimal impact'. This can be seen, especially when comparing the peak values of Finished time found in chart 1 and chart 2. The peak, along with the slope in chart 2 were much more noticeable then they were in chart 1.

Now at first we thought, is this because the change from 10ms to 30 ms is much less than the change of 40 ms to 200 ms. At initial glance yes, but this simulation is supposed to represent a real OS design. The increase of ISR going from 40 - 200 ms is realistic, and similar to CSR going from 10 - 30 ms. So yes the scale of variation chosen for the two values may seem unfair, but they are realistic thus the takeaways and conclusions are valid and reasonable.

We had also run tests, to see what difference changing the addresses from 2 bytes to 4 bytes did. This basically just means that there was a double in memory positions. We had made a fixed lookup time (1 ms) thus there was no impact on the execution time when it came to changing the vector. The only change that was made was the memory positions in which the values were logged at (they doubled as mentioned before).