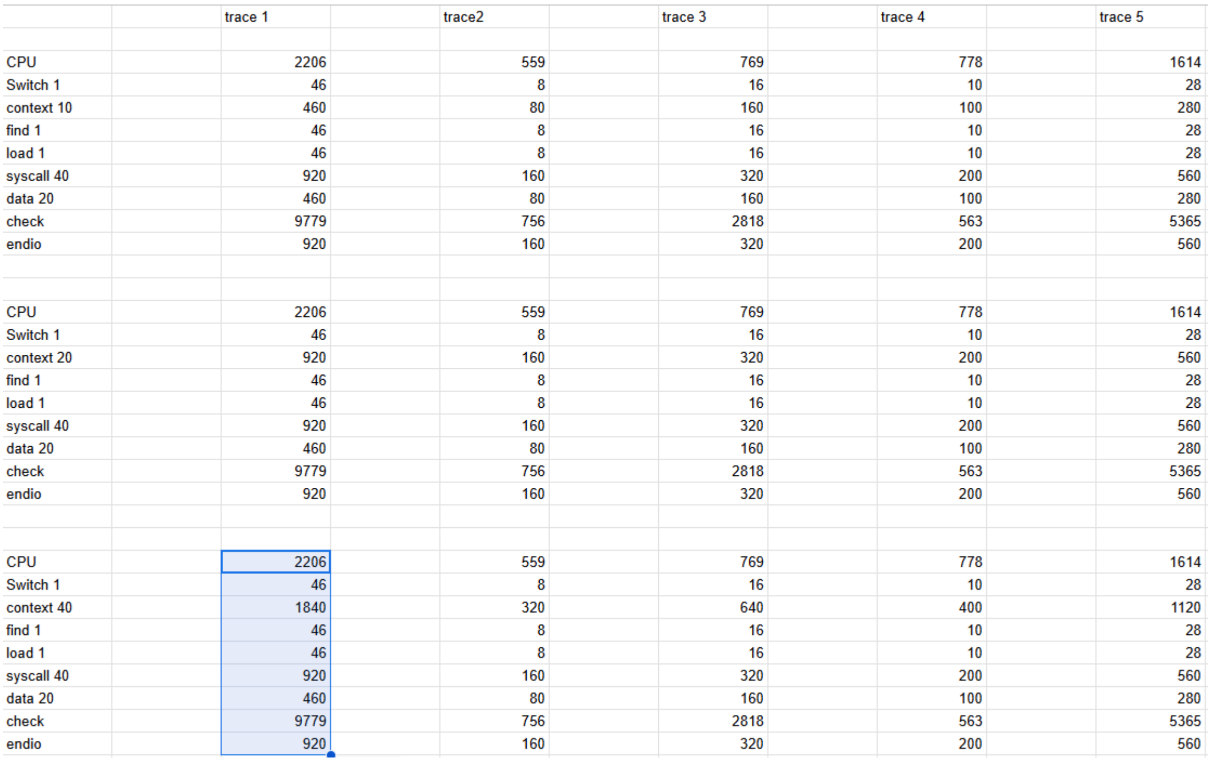
Part 2 - Report

Interrupt simulation scheduling variation analysis discussion

Github link: https://github.com/aiden3218/SYSC4001_A1

TA_testcase_execution_# is execution from the TA test case files (trace_#)
Execution_# is for the student made trace files (student_trace_#)

Context switch time changes from 10,20,40



	trace 1	trace2	trace 3	trace 4	trace 5
CPU	2206	559	769	778	1614
Switch 1	46	8	16	10	28
context 10	460	80	160	100	280
find 1	46	8	16	10	28
load 1	46	8	16	10	28
syscall 40	920	160	320	200	560
data 20	460	80	160	100	280
check	9779	756	2818	563	5365
endio	920	160	320	200	560
CPU	2206	559	769	778	1614
Switch 1	46	8	16	10	28
context 20	920	160	320	200	560
find 1	46	8	16	10	28
load 1	46	8	16	10	28
syscall 40	920	160	320	200	560
data 20	460	80	160	100	280
check	9779	756	2818	563	5365
endio	920	160	320	200	560
CPU	2206	559	769	778	1614
Switch 1	46	8	16	10	28
context 40	1840	320	640	400	1120
find 1	46	8	16	10	28
load 1	46	8	16	10	28
syscall 40	920	160	320	200	560
data 20	460	80	160	100	280
check	9779	756	2818	563	5365
endio	920	160	320	200	560

Switch modes, Finding addresses and loading addresses calls all take 1 millisecond. These times are bordering on negligible. When looking at room for improvement through scheduling or even hardware performance there is very little to change. As well, focusing on improving hardware to speed up any of these processes will also better benefit the other processes needed in interrupts. Focusing on creating software to improve these speeds will also most likely decrease other processes' speed. There would be little to no reason to look into shortening these time frames.

A context switch seems to be the only standard predictable step with a large enough scheduling time. Context switches can happen often as interrupts, system calls and the scheduler can all cause a context switch. In a more advanced operating system simulation registers are designed with context switch timing in mind as the registers are what hold the information from the context switches.

The most time consuming parts of the simulation come from the CPU, ISR and ENDIO processes. They are also the most mandatory as the CPU computations are what the user wants and the drivers are what communicate the information. Although the CPU takes the most significant amount of time this is the best case as the goal is to have the CPU have the least downtime; more CPU time means more jobs will get done, as long as the CPU is not waiting. The scheduler needs to be designed properly to effectively manage the jobs for the CPU. A poorly designed scheduler although could have the same CPU time could also cause starvation and longer wait times for the user. Of course the main part about improving computers is creating a faster and cheaper CPU as they are the most important part of a computer.

The ISR and ENDIO also take significant time during the interrupt process. The length of the processes vary by SYSCALL as they call different drivers that can all take different times. These are the biggest concerns for holding up the CPUs computing time, as they take up a lot of time and poor design can heavily impact the overall computers performance. ISRs are generally recommended to be as short as possible as too long of ISRs can cause latency and also so it will not cause errors in the OS; the OS has critical timing required for events as well as other lower priority interrupts that can be missed if the ISR is still running. A long ISR time can cause severe issues in the OS, missing critical information and ultimately not functioning properly for the user. As asked in the assignment, if the ISR were to take 200ms it would exceed the expected time period of the interrupt itself and the OS would ultimately miss interrupts and important data resulting in many possible errors.

If we were to increase the length of something else such as the size of each element in the vector table there would be several implications. Larger element sizes means more space required for the vector table. Finding the vector address and loading it would also take longer requiring a longer expected interrupt time. Although larger data spaces would also allow for more information to be stored in the vector table. With a well designed OS the ISR time could be decreased under certain circumstances.

In conclusion, operating systems can have many things changed in terms of space or expected timing that can all impact the performance of the overall computer.