

SYSC4001A (L1 - 6) Assignment 1

Part 2: Design and Implementation of an Interrupts Simulator

Student 1: Wenxuan Han - 101256669

Student 2: Tony Yao - 101298080

2025-10-06

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1			Time	Activity	Time if CPU	Time if IO	Time if overhead							
2		0	51	CPU Burst	51	0	0		Total running time:	2458				
3		51	1	switch to kernel mode	0	0	1		total running time if CPU:	190				
4		52	10	context saved	0	0	10		total running time if IO:	2216				
5		62	1	find vector 14 in memory position 0x001C	0	0	1		total running time if overhead:	52				
6		63	1	load address 0X0165 into the PC	0	0	1							
7		64	40	SYSCALL: run the ISR (device driver)	0	40	0		Percentage of usage(CPU)	8%				
8		104	40	transfer data from device to memory	0	40	0		Percentage of usage(IO)	90.15%				
9		144	376	check for errors	0	376	0		Percentage of usage(overhead)	2.12%				
10		520	39	CPU Burst	39	0	0							
11		559	1	switch to kernel mode	0	0	1							
12		560	10	context saved	0	0	10							
13		570	1	find vector 14 in memory position 0x001C	0	0	1							
14		571	1	load address 0X0165 into the PC	0	0	1							
15		572	40	END_IO: run the ISR(device driver)	0	40	0							
16		612	416	check device status	0	416	0							
17		1028	72	CPU Burst	72	0	0							
18		1100	1	switch to kernel mode	0	0	1							
19		1101	10	context saved	0	0	10							
20		1111	1	find vector 19 in memory position 0x0026	0	0	1							
21		1112	1	load address 0X0765 into the PC	0	0	1							
22		1113	40	SYSCALL: run the ISR (device driver)	0	40	0							
23		1153	40	transfer data from device to memory	0	40	0							
24		1193	572	check for errors	0	572	0							
25		1765	28	CPU Burst	28	0	0							
26		1793	1	switch to kernel mode	0	0	1							
27		1794	10	context saved	0	0	10							
28		1804	1	find vector 19 in memory position 0x0026	0	0	1							
29		1805	1	load address 0X0765 into the PC	0	0	1							
30		1806	40	END_IO: run the ISR(device driver)	0	40	0							
31		1846	612	check device status	0	612	0							
32														
33		save/restore context switch:	10											
34		ISR per activity time:	40											
35		vector number at index 19	652											
36		vector number at index 14	456											
37														
38														
39														

[illegible]

- If we change the value of save/restore context time from 10, to 20, and to 30 (the image of excel is very similar to simulation outputs on the GitHub), the total running time of system will be increased from 2458 to 2498, and to 2538. However, the percentage of time of overhead is also increased steadily from 2.12% to 5.20%. That means that increasing save/restore context time will increase the overhead time in the system. Also, overhead time is always managed by the operating system, and they always do useless work in the process of system while OS keeps manage them.
- When we tried to increase the ISR activity time from 40 to 130, the total interrupts handling time became huge due to post-processing of interrupt. It causes some delays in the CPU execution, and it also delays further processing. Also, the time of checking errors will be decreased as there was an increase in some activities within the process of ISR. Thus, there will be limited time for hardware device to check errors before executing the next instruction via the CPU. However, when the total time of activities time is larger than delays time of device on vector table, it will delay the whole execution time and stops the process of checking the errors by hardware device.
- If we tried to vary the value of the save/restore context time from 10ms to 30ms, it will delay the overall execution time while the total time of running process is increased due to increased save/restore context time. Also, save/restore context is also a part of context switch and takes up the time of overhead in the whole process. Time of overhead is also useless in the overall execution time of the process as operating system just manages it in the whole process. On the other hand, when we tried to vary the ISR activity time from 60 ms to 130ms, the whole execution time is unchanged while it only causes some delays in activities within the process of ISR. But the whole process will delay if the total activity time within ISR is larger than time of delays in vector device time (mainly based on the preset value of ISR activity time).

[illegible]

- When we have a faster CPU in the process, the whole execution time can be reduced as the CPU finished executing job earlier and other process can start earlier as well. Also, the percentage of usage of overhead time is increased when CPU in the process becomes faster (can be seen from the above two tables). That is because the process with overhead time is not related to the process of CPU execution while the process with overhead time always did something that is not useful for system functions. When the running time of overhead is unchanged, the whole execution time is reduced, and the value of percentage becomes larger.

GitHub Repository:

https://github.com/toast23/SYSC4001_A1.git