Operating System **Hsu PO-CHUN**
10610 CS342302 Student ID: s105062803
Prof. Pai H. Chou Submitted: Sept. 22, 2017
TA: Due Date: Sept. 24, 2017

Assignment 1

1 Problem 1

[25 points] Which of the following instructions should be privileged for an OS to function properly? Explain.

a. Set value of timer.

b. Read the clock.

c. Issue a trap instruction.

d. Turn off interrupts.

e. Access I/O device.

1.1 Answer

a. Set value of timer, d. Turn off interrupts and e. Access I/O device should be privileged for an OS to function properly. Explain:

a. Set value of timer should be privileged because the user processor may affect the OS scheduling policy that use the value of timer to proceed.

b. Read the clock is OK for user program that just to read the value.

c. Issue a trap instruction is used to let user processes request a software interrupt for CPU time to process specific jobs

d. Turn off interrupts should be privileged or user processes can turn off the interrupts to effect the control and the behavior progress.

e. Access I/O device should be privileged because I/O resources need to be maintained by the OS or it is prone to cause failure by user processes.

2 Problem 2

[25 points] (Problem 1.8) What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

2.1 Answer

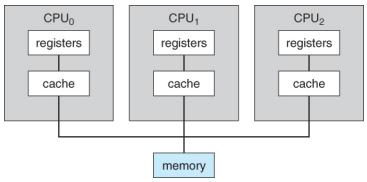
Interrupts are generated by the hardwares to change the flow of the system. There is an interrupt handler to handle the interrupts with different interrupt service routines(ISR). The control will return back to the context and instruction after the ISR finished. An interrupt can be used to signal the CPU when the I/O operation is finished. And a trap is an interrupt generated by the software. Yes, traps can be generated intentionally by a user program, and a trap can be used to call OS routines or catch the exceptions such as divide by zero.

3 Problem 3

[25 points] (Problem 1.12) Consider an SMP system similar to the one shown in Fig. 1.6. Illustrate with an example how data residing in memory could in fact have a different value in each of the local caches. [Hint: all you need to do is to explain an example of how this can happen]

3.1 Answer

In SMP system, each processor has its register set and local cache, and these CPU share the same physical memory:



Consider an example: A process running in CPU_0 read a data from the main memory and some operations are performed. Before the data is write back to the memory, another process running in CPU_1 also read the data from the same main memory, and also performing some operations to change the data. The data in the local cache of CPU_0 is obtained from a process, and the data in the local cache of CPU_1 is obtained from another process. Both of the processors change the value of the data in different operations. The data will be different. From this example we can tell that the data residing in the memory could in fact have two different values in each local caches.