CSCE 410/611 Operating Systems Fall 2022

Homework for Week 2

(Due Date: Check Canvas)

1. A typical hardware architecture provides an instruction called “return from interrupt”, and abbreviated by something like rti. This instruction switches the mode of operation from supervisor mode to user mode. This instruction is usually only available while the machine is running in supervisor mode.

(a) Explain where in the operating system this instruction would be used.

This instruction occurs when the OS would like to return from a context switch or handling a routine interrupt.

(b) What happens if an application program executes this instruction?

The application program since it is not in supervisor mode the program will cause a hardware exception

2. A hardware designer argues that there are enough transistors on the chip to provide 1024 integer registers and 512 floating point registers. You have been invited as the operating system guru to give opinion about the new design.

(a) What is the effect of having such a large number of registers on the operating system?

This increases the amount of memory significantly, but also makes context switching, interrupt handling more expensive. Now this spirals to make the cost of creating the structures to support these operations expensive and increases the cost of lightweight threading.

(b) What happens if the hardware designer also wants to add a 16-station pipeline into the CPU. How would that affect the context switching overhead?

This would increase the cost of handling interrupts because it must flush an increased pipeline, overall increasing the cost of context switching and its necessary overhead.

3. Consider a uniprocessor kernel that user programs can trap into using system calls. The kernel receives and handles interrupt requests from I/O devices. Would there be any need for critical sections within that kernel?

Yes. The user program can enter a trap state and the single kernel thread that will not be rejoined as it could be operating on unprotected global data. What can operate on the global data structures is the interrupt handler.

References

[1] A. Silberschatz, P. Galvin, and G. Gagne, *Applied Operating Systems Concepts*, John Wiley & Sons, Inc., New York, NY, 2000.

[2] Deitel, Deitel, and Choffnes, *Operating Systems*, Pearson / Prentice Hall, 2004. [3] A. S. Tanenbaum, *Modern Operating Systems*, Pearson / Prentice Hall, 2008. [4] L. F. Bic, A. C. Shaw, *Operating Systems Principles*, Prentice Hall 2003. [5] C. Crowley, *Operating Systems, A Design-Oriented Approach*, Irwin 1997. [6] M. Herlihy, N. Shavit, *The Art of Multiprocessor Programming*, Elsevier, 2008

1