EC440, Fall 2025

Homework #1

Yigong Hu, yigongh@bu.edu

The [name] tags give scholarly attribution to authors of other OS textbooks who wrote the question (you do not need the textbook to be able to answer the question).

Problem 1: [Anderson]

For each of the three mechanisms that supports dual-mode operation–privileged instructions, memory protection, and timer interrupts–explain what might go wrong without that mechanism, assuming the system only had the other two. (In other words, if we just had memory protection and timer interrupts but not privileged instructions, what could go wrong; if we just had privileged instructions and timer interrupts, what could go wrong, etc.)

Problem 2: [Silberschatz]

What are the differences between a trap and an interrupt? What is the use of each function?

Problem 3: [Silberschatz]

Which of the following instructions should be privileged?

  (i) Set value of timer

 (ii) Read the clock

(iii) Clear memory

 (iv) Turn off interrupts

  (v) Switch from user to monitor (kernel) mode

Problem 4: [Silberschatz]

Protecting the operating system is crucial to ensuring that the computer system operates correctly. Provision of this protection is the reason for dual mode operation, memory protection, and the timer. To allow maximum flexibility, however, you should also place minimal constraints on the use.
The following is a list of instructions that are normally protected. What is the minimal set of instructions from this list that must be protected?

  1. Change to user mode

  2. Change to monitor (kernel) mode

  3. Read from monitor memory

4. Write into monitor memory

5. Fetch an instruction from monitor memory

6. Turn on timer interrupt

7. Turn off timer interrupt

Problem 5: [Anderson]

Explain the steps that an operating system goes through when the CPU receives an interrupt.

Problem 6: [Anderson]

Suppose you have to implement an operating system on hardware that supports exceptions and traps but does not have interrupts. Can you devise a satisfactory substitute for interrupts using exceptions and/or traps? If so, explain how. If not, explain why.

Problem 7: [Silberschatz]

Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.

- How does the CPU interface with the device to coordinate the transfer?

- How does the CPU know when the memory operations are complete?

- The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of user programs? If so, describe what forms of interference are caused.

Problem 8: [Tanenbaum]

For each of the following system calls, give a condition that causes it to fail: open, fork, exec, unlink. (Hint: You can explore the semantics of these system calls using man, e.g., man 2 fork.)

Problem 9

he Java runtime provides a set of standard system libraries for use by programs. To what extent are these libraries similar to the system calls of an operating system, and to what extent are they different?

Problem 10

List two challenges an operating system faces when passing parameters between user and kernel mode. Describe how an operating system can overcome them.