CS 318/418/618, Fall 2017

Homework #1

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).

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.)

2. [Silberschatz] What are the differences between a trap and an interrupt? What is the use of each function?

3. [Silberschatz] Which of the following instructions should be privileged?

   (a) Set value of timer

   (b) Read the clock

   (c) Clear memory

   (d) Turn off interrupts

   (e) Switch from user to monitor mode

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?

   (a) Change to user mode

(b) Change to monitor mode

(c) Read from monitor memory

(d) Write into monitor memory

(e) Fetch an instruction from monitor memory

(f) Turn on timer interrupt

(g) Turn off timer interrupt

5. [Crowley] Suppose the hardware interval timer only counts down to zero before signaling an interrupt. How could an OS use the interval timer to keep track of the time of day?

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.

7. [Anderson] Explain the steps that an operating system goes through when the CPU receives an interrupt.

8. [Tanenbaum] For each of the following system calls, give a condition that causes it to fail: `open`, `fork`, `exec`, `unlink`. (Hint: We discussed some in lecture, and you can also explore the error semantics of these system calls using `man`, e.g., `man 2 fork`.)

9. The 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?

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