CSCI 4320 (Principles of Operating Systems), Fall 2001

Homework 4

Assigned: November 15, 2001.

Due: November 27, 2001, at 5pm. *Not accepted later than November 29 at 5pm.*

Credit: 20 points.

1 Reading

Be sure you have read (or at least skimmed) chapter 5.

2 Problems

Answer the following questions. You may write out your answers by hand or using a word processor or other program, but please submit hard copy, either in class or in my mailbox in the department office.

1. (5 points) Consider a computer system with the following characteristics: Reading or writing a memory word takes up to 10 nsec. It has 16 CPU registers, and when an interrupt occurs, all of them, plus the program counter and the PSW are pushed onto the stack (in memory). What is the maximum number of interrupts per second this machine can process? (*Hint:* Observe that after an interrupt is processed, the contents of CPU registers, program counter, and PSW must be restored to their pre-interrupt values by popping them back off the stack.)

2. (5 points) Consider a printer that prints at a maximum rate of 400 characters per second, connected to a computer system in which writing to the printer's output register takes essentially no time. If each character printed requires an interrupt that takes a total of 50 microseconds to process, would it make sense to use interrupt-driven I/O to write to this printer, or would it be better to use programmed I/O? Why?

   Now consider a system with a memory-mapped terminal and suppose that interrupts take a minimum of 100 nsec to process and copying a byte into the terminal's video RAM takes 10 nsec. Would it make sense to use interrupt-driver I/O to write to the terminal, or would it be better to use programmed I/O? Why?

3. (5 points) The textbook divides the many routines that make up an operating system's I/O software into four layers, as shown in Figure 5-10. In which of these layers should each of the following be done?

   (a) Writing commands to a printer controller's device registers.

   (b) Detecting that an application program is attempting to write data from an invalid buffer address.

   (c) Converting floating-point numbers to ASCII for printing.

   (d) Computing the track, sector, and head for a disk read operation.

4. (5 points) Consider a computer system that maintains date and time using a 32-bit unsigned integer whose value represents a number of seconds since January 1, 1970. (So, a value of 362 would represent 12:06:02am, January 1, 1970.) In what year will this scheme become unworkable because the 32-bit integer is not big enough? What if instead the system uses a signed 32-bit integer, allowing negative values to represent dates and times before January 1, 1970? (Ignore leap-year complications and assume that the average year has 365.25 days.)