CSCI 4320 (Principles of Operating Systems), Fall 2003

Homework 5

Assigned: November 14, 2003.

Due: November 20, 2003, at 5pm.

Credit: 30 points.

> *Note:* The HTML version of this document may contain hyperlinks. In this version, hyperlinks are represented by showing both the link text, formatted like this, and the full URL as a footnote.

Be sure you have read chapter 5.

1 Problems

Do the following problems. 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? (*Hint:* How much time is required for interrupt processing if the printer is printing at its maximum rate?)

   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. (Assume that detecting an invalid buffer address can only be done in supervisor mode.)

(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 system that uses its local area network as follows. An application program makes a system call to write data packets (each 1024 bytes, ignoring headers) to the network. The operating system first copies the data to be sent to a kernel buffer. Working on one packet at a time, it then copies the data to the network controller. When all 1024 bytes have been copied to the network controller, it sends them over the network at a rate of 10 megabits per second. The receiving controller receives each bit a microsecond after it is sent. When the last bit in the packet is received, the destination CPU is interrupted, and its operating system copies the packet into a kernel buffer, inspects it, and copies it into a buffer owned by the application program that should receive it. It then sends back an acknowledgment (assume one bit) to the sending computer, which interrupts the sending CPU, and work can begin on the next packet. How long does it take to send each packet, if it takes one millisecond to process an interrupt (on either CPU) and one microsecond to copy a byte? Assume that the time taken for the receiving CPU to inspect the packet is negligible. What is the effective transfer rate (in bits per second) over this connection?

   (*Hints:* Notice that some times are per bit and some are per byte. If you think you need to make additional assumptions, do so and explain them. If you show your calculations and briefly explain what you are doing, your odds of getting partial credit are better.)

5. (5 points) Suppose at a given point in time a disk driver has in its queue requests to read cylinders 10, 22, 20, 2, 40, 6, and 38, received in that order. If a seek takes 6 milliseconds per cylinder moved, and the arm is initially at cylinder 20, how much seek time is needed to process these requests using each of the three scheduling algorithms discussed (FCFS, SSF, and elevator)? Assume no other requests arrive while these are being processed.

6. (5 points) Student H. Hacker installs a new disk driver that its author claims improves performance by using the elevator algorithm and also processing requests for multiple sectors within a cylinder in sector order. Hacker, very impressed with this claim, writes a program to test the new driver's performance by reading 10,000 blocks spread randomly across the disk. The observed performance, however, is no better than what would be expected if the driver used a first-come first-served algorithm. Why? What would be a better test of whether the new driver is faster? (*Hint:* The test program reads the blocks one at a time. Think about how many requests will be on the disk driver's queue at any one time.)