CSCE 351 Operating System Kernels

Fall 2001
Steve Goddard

Homework 3/Study Aid, December 4

Due: never; this assignment need not be handed in.

However, your final will include at least one question from this assignment.

---

1) The clock interrupt handler on a certain computer requires 2 msec (including process switching overhead) per tick. The clock runs at 60 Hz. What fraction of the CPU is devoted to the clock?

2) Give three examples of watchdog timers (i.e., three uses for watchdog timers).

3) Consider the use of a delta list to keep track of pending alarms (timers), as shown in Figure 3-25 of the text.

   Construct the delta list that would be created if the following alarm requests all arrived at time 30: 19, 15, 7, 42, 25, 19, and 24.

   Insert into your delta list an alarm request for 15 that arrives at time 32.

   At time 40, the alarm request for 15 time units is canceled. Explain how this should be handled, the delta list after the alarm is canceled, and the time at which the next alarm will expire.

4) Is it better to use a delta list rather than simply creating a linked list of timers with absolute timeout values (sorted in monotonically non-decreasing order)? Explain.

5) A bit-map terminal contains 1200 by 800 pixels. To scroll a window, the CPU (or the controller) must move all the lines of text upward by copying their bits from one part of the video RAM to another.

   If a particular window is 66 lines high by 80 characters wide (5280 characters, total), and a character's box is 8 pixels wide by 12 pixels high, how long does it take to scroll the whole window at a copying rate of 500 nsec per byte?

   If all lines are 80 characters long, what is the equivalent baud rate of the terminal when it takes 50 microsec to put a character on the screen?

   Now compute the baud rate for the same terminal in color, with 4 bits/pixel. (Putting a character on the screen now takes 200 microsec.)

6) After receiving a DEL (SIGINT) character, the MINIX driver discards all output currently queued for that terminal. Why?

7) Many RS232 terminals have escape sequences for deleting the current line and moving al the lines below it up one line. How do you think the feature is implemented inside the terminal?

8) MINIX postpones driver initialization until as late as possible. Why does it do this?

9) You are writing a new disk driver for a hard disk that supports Linear Block Addressing (LBA). How does this affect your driver? Does it make it easier or harder? Why?

10) How do reads and writes to I/O ports differ from reads and writes to memory addresses? For example, reading from a memory address is a non-volatile operation. That is, reading a memory address does not change the value. Is this true for I/O ports?

11) The routine `kbd_hw_int` in MINIX does not send an interrupt to the terminal task when the keyboard generates an interrupt. Why? How is the terminal task notified when data is entered and how does it get to the user in MINIX?

12) Explain the difference between software scrolling and hardware scrolling. Why would the driver ever use software scrolling rather than hardware scrolling if it is available?

13) What does the MINIX system task do?

14) Explain the role of the MINIX system task in file reads.