CSC201, SECTION 002, Fall 2000: Homework Assignment #5 Problems

---

DUE DATE

December 6 for the homework problems, in class.

INSTRUCTIONS FOR HOMEWORK PROBLEMS

- Neat, in order, answers easy to find.
- Staple the pages together at the upper left corner.
- Fold lengthwise. On the outside write the course number, the assignment number, the date, and your name.

Thanks for your help.

PROBLEMS

1. (3)
   Why does the operating system have to be loaded in stages during boot up? Why does part of the operating system have to be stored in ROM (vs. storing in the regular memory, or storing all of the operating system on the hard drive)?

2. (4)
   A CPU outputs 1000 characters. The CPU takes 100 ns to execute the code required to transfer a single character to a device for output. The device can output characters at a maximum rate of 5000 characters per second (it's a slow device). The CPU uses a busy wait loop between outputting each character. How much time does it take to output the 1000 characters, and what fraction of the time is the CPU doing useful work (i.e., transferring a character)?

3. (4)
   Suppose instead of using busy waiting, the CPU

- outputs a character;
- immediately switches to another task;
- is then interrupted when the printer is again ready;
- outputs another character;
- immediately switches to another task;
- etc.

Assume it takes 100ns to transfer the character, as before. It requires 500ns to switch to another task, and 200ns to interrupt the CPU. How much time does it take to output the 1000 characters, and what fraction of the time is the CPU doing useful work (i.e., either transferring a character or executing the other task; the time for switching, and the time for interrupts, is not counted as useful work).

4. (1)
   If the user presses the CTRL key, presses the SHIFT key, releases the CTRL key, presses the 'a' key, releases the 'a' key, and releases the SHIFT key, what character or characters does the CPU interpret as being typed by the user?

5. (2)
   When an interrupt occurs, the hardware does several things automatically before the next instruction is executed. One thing it does is disable interrupts by clearing the interrupt flag. Why does the hardware disable interrupts automatically, before transferring control to the interrupt service routine?

6. (2)
   A modem transmits at a rate of 40000 bits per second. If each character is preceded by 2 start bits and ended by 1 stop bit, what is the data transfer rate in characters per second?

7. (5)
   Show the assembly language code for the CPU to receive a character from the keyboard, print it, then receive another character from the keyboard, and print it, in that order. If the printer is extremely slow and the user is a fast typist, will there be any problems with your code? If so, can you suggest any way to overcome this problem?

8. (2)
   Why is DMA useful, i.e., what benefit does it provide over letting the CPU handle all data transfers between devices and memory?

9. (2)
   Why are vectored interrupts preferred over non-vectored interrupts?

10. (2)

What is the task state that must be saved when an interrupt occurs?

11. (3)
    If a user program gets in an infinite loop, how is it possible for the operating system to ever get control of the CPU again, so that, for instance, the user program can be terminated?

12. (4)
    Suppose a hard disk read/write head is reading track #11, and requests arrive to read tracks 1, 36, 16, 34, 9, and 12. If the head moves to these tracks in this sequence, what is the total amount of motion of the head? (Note: to move from track 11 to track 1 is a movement of 10 tracks, etc.) Is there any way to schedule the sequence of tracks being read so that the total amount of motion is minimized? What is this schedule, and what is the total motion? (Note: one schedule, for instance, would be to access the tracks in numerical order: 1, 9, 12, 16, 34, 36.)

13. (2)
    Why do traps have to be detected in hardware, and calls to the operating system do not?

14. (3)
    What factors determine how long it takes to respond to an interrupt? (a 3 or 4 sentence answer is sufficient)

15. (2)
    Why does the CPU need to acknowledge an interrupt request, i.e., what happens if it services the interrupt but does not acknowledge it?

16. (3)
    Why should interrupts ever be disabled? Why is it important to disable them for as little time as possible?

17. (3)
    If an interrupt service routine is itself interrupted, will it ever resume execution, or is it just terminated? Explain!

18. (5)
    Write the code to read characters from the keyboard and store them in a buffer. Your code should use busy-waiting. When the end-of-line character (0Ah) is encountered, your code should branch to a label called endofinput.

19. (2)
    How can a user program request operating system services if the operating

systems memory has a higher privilege level than the user program memory?

20. (4)
    A DVD-ROM can store 4.7 GB of data (consisting of sound, images or frames, and text). Suppose a 133-minute video consists of frames that are 720x480 in size, and each pixel requires 3 bytes of storage. The video frame rate is 30 frames a second. How much storage do the frames of this video require altogether if the video is not compressed?

GRADING

This homework is graded on a scale of 100 points; the points for each problem are shown above. The homework problems are worth 60 points, and the program is worth 40 points. This homework will be weighted to contribute 5% of your course grade.