CSCI 3323 (Principles of Operating Systems), Fall 2016

Homework 7

Credit: 30 points.

1 Reading

Be sure you have read, or at least skimmed, Chapter 5.

2 Honor Code Statement

Please include with each part of the assignment the Honor Code pledge or just the word "pledged", plus one or more of the following about collaboration and help (as many as apply).[1] Text *in italics* is explanatory or something for you to fill in. For written assignments, it should go right after your name and the assignment number; for programming assignments, it should go in comments at the start of your program.

- This assignment is entirely my own work.
- This assignment is entirely my own work, except for portions I got from the assignment itself *(some programming assignments include "starter code")* or sample programs for the course *(from which you can borrow freely — that's what they're for)*.
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc.*
- I got significant help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. ("Significant" here means more than just a little assistance with tools — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*
- I provided significant help to *names of students* on this assignment. *("Significant" here means more than just a little assistance with tools — you don't need to tell me about helping other students decipher compiler error messages, but beyond that, do tell me.)*

3 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 one of my mailboxes (outside my office or in the ASO).

[1] Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM's Special Interest Group on CS Education.

1. (10 points) Consider the following two I/O devices. For each device, say whether you think programmed I/O or interrupt-driven I/O makes the most sense, and justify your answer. (*Hint:* Consider the time required for interrupt processing versus the time needed for the actual input/output operation. You will get more credit if you give actual numbers for these times.)

   (a) 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, and using interrupt-driven I/O means that each character printed requires an interrupt that takes a total of 50 microseconds (i.e., 50×10^{-6} seconds) to process.

   (b) A simple memory-mapped video terminal (output only), connected to a system where interrupts take a minimum of 100 nsec to process and copying a byte into the terminal's video RAM takes 10 nsec. (It's probably best to think of this as a somewhat hypothetical problem, using only the description supplied, rather than trying to extrapolate from what you know or can read about typical actual hardware.)

2. (10 points) The textbook divides the many routines that make up an operating system's I/O software into four layers. In which of these layers should each of the following be done? Why? (Assume that in general functionality should be provided at the highest level at which it makes sense — e.g., in user-level software rather than device-independent software.)

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

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

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

   (d) 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.)

3. (10 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 5 milliseconds (i.e., 5×10^{-3} seconds) 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 that no other requests arrive while these are being processed and that for the elevator algorithm the initial direction of movement is outward (toward larger cylinder numbers).