CSCI 3323 (Principles of Operating Systems), Fall 2017

Homework X

Credit: Up to 50 extra-credit points.

1 General Instructions

Answer as many (or few) of the following questions as you like. (Notice, however, that you can receive at most 50 extra-credit points.)

I am also open to the possibility of giving extra credit for other work — other problems from the textbook, a report on something course-related, etc. If you have an idea for such a project, let's negotiate (by e-mail or in person).

For this assignment, please work individually, without discussing the problems with other students. If you want to discuss problems with someone, talk to me.

2 Honor Code Statement

Please include with each part of the assignment the Honor Code pledge or just the word "pledged", plus one of the following statements, whichever applies:

- "This assignment is entirely my own work".
- "This assignment is entirely my own work, except that I also consulted *outside course — a book other than the textbook (give title and author), a Web site (give its URL), etc.*."

(As before, "entirely my own work" means that it's your own work except for anything you got from the assignment itself — some programming assignments include "starter code", for example — or from the course Web site.)

3 Problems

Answer as many of the following questions as you like. 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. Problems related to chapter 5 (I/O):

   (a) (Up to 2 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 12:06:02 am, 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.)

(b) (Up to 8 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 (10×10^6 bits) 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.)

2. Problems from chapter 9 (security):

   (a) (Up to 2 points) Answer question 26 on p. 708 of the textbook. (*Hint:* What are the odds of being able to guess the password if you know its length? if you don't?)

   (b) (Up to 2 points) Answer question 34 on p. 709 of the textbook.

   (c) (Up to 2 points) Answer question 37 on p. 709 of the textbook.

   (d) (Up to 2 points) Answer question 46 on p. 710 of the textbook.

   (e) (Up to 2 points) Answer question 48 on p. 710 of the textbook.

3. Essay problems (please include in your answer an informal bibliography listing sources on which it is based — Web sites, books, etc.):

   (a) (Up to 10 points) Most of the memory-management schemes discussed in the textbook are based on the idea that each process has its own "address space", each of which uses the same range of virtual addresses ranging from 0 to some large number (often the maximum possible based on the number of bits in an address). However, some of the older mainframe operating systems instead defined a single address space shared by all processes, with each process having a different range of virtual addresses. There have been indications in the not-so-dim past that this idea might be considered again. Speculate on how it might be done, what advantages there might be, what disadvantages there might be, and so forth. (In particular it might revive the program-relocation problem.)

4 Programming Problems

Do as many of the following programming problems as you like. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu` with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., "csci 3323 hw X" or "O/S hw X"). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department's Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (Up to 10 points) Add one or more features to the simple shell program you wrote for Homework 1. How much credit you get will depend on the level of difficulty involved. A not-too-difficult choice involves adding a command history; the `man` page for `readline` and associated reading is a good starting point.

2. (Credit will vary.) Do any or all of the extra-credit parts of the programming problems for Homeworks 2, 3, and 5. (Credit will be whatever you would have received if you had turned in the extra-credit parts with the rest of the assignment.)

3. (Up to 10 points) The Linux lab machines have special files `/dev/random` and `/dev/urandom` that generate sequences of "random" bytes. (Read the man page for `urandom` for an explanation of the difference between them.) Write a program that compares the results of generating N integers using one or both of these special files to the results of generating N integers using function `rand()`. (It's up to you to decide how to compare them. A simple test might be to count how many are even and how many are odd. You may have a better idea!) Submit your source code and a text file containing output of one or more executions. (*Hint:* You will probably need to use `open` and `read` rather than `fopen` and `fscanf` to read from the special file. `man` pages for these two functions can be found via `man 2 open` and `man 2 read`.)