

CSCI 4320 (Principles of Operating Systems), Fall 2004

Homework X

Assigned: December 7, 2004.

Due: December 14, 2004, at 5pm. *Not accepted late.*

Credit: Up to 20 extra-credit 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.

1 General instructions

Answer as many (or few) of the following questions as you like. (Notice, however, that you can receive at most 20 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 Problems from Chapter 9 (Security)

For these problems, please submit hard copy (in my mailbox in the department office or under my door).

1. (2 points) Answer question 3 on p. 667 of the textbook. (*Clarification:* i, j, k , etc., represent integers in the range from 0 through $n - 1$ inclusive.)
2. (2 points) Answer question 6 on p. 667 of the textbook. (*Hint:* What are the odds of being able to guess the password if you know its length? if you don't?)
3. (2 points) Answer question 7 on p. 667 of the textbook.
4. (2 points) Answer question 11 on p. 668 of the textbook.
5. (2 points) Answer question 16 on p. 668 of the textbook.
6. (2 points) Answer question 28 on p. 669 of the textbook.

3 Problems from Chapter 10 (UNIX and Linux)

For these problems, please submit hard copy (in my mailbox in the department office or under my door).

1. (2 points) Answer question 1 on p. 758 of the textbook.
2. (2 points) Answer question 3 on p. 758 of the textbook.
3. (2 points) Answer question 6 on p. 758 of the textbook. (Assume that commands **a**, **b**, **c**, **d**, **e**, and **f** all run for a long time.)
4. (2 points) Answer question 8 on p. 758 of the textbook.
5. (2 points) Answer question 15 on p. 759 of the textbook.
6. (2 points) Answer question 22 on p. 760 of the textbook.
7. (2 points) Answer question 23 on p. 760 of the textbook.
8. (2 points) Answer question 35 on p. 760 of the textbook.
9. (2 points) Answer question 40 on p. 761 of the textbook.
10. (2 points) Answer question 41 on p. 761 of the textbook.

4 Programming Problems

For these problems, turn in your code by sending mail to `csci4320-homework@cs.trinity.edu`, with each of your code files as an attachment. If there's any question of which file(s) correspond to which problems, explain in the body of the mail message. Please use a subject line such as "homework X" or "hwx". You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department's Fedora Core 2 Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (2 points) Write a simple command shell, as described in optional programming problem 1 in [Homework 2](#)¹. (If you submitted a version of this program as part of Homework 2, you will only get credit for improvements/additions to what you submitted before.)
2. (2 points) Write/revise the scheduler program from optional programming problem 3 in [Homework 2](#)². (If you a version of submitted this program as part of Homework 2, you will only get credit for improvements/additions to what you submitted before.)
3. (2 points) Write a program to count files of varying sizes, as described in the optional programming problem in [Homework 4](#)³. (If you a version of submitted this program as part of Homework 4, you will only get credit for improvements/additions to what you submitted before.)

¹http://www.cs.trinity.edu/~bmassing/Classes/CS4320_2004fall/HW02/Problems/homework/homework.html

²http://www.cs.trinity.edu/~bmassing/Classes/CS4320_2004fall/HW02/Problems/homework/homework.html

³http://www.cs.trinity.edu/~bmassing/Classes/CS4320_2004fall/HW04/Problems/homework/homework.html

4. (2 points) Program `thing.c`⁴ performs some calculations on an array of doubles of size N (where N is specified by a command-line argument, e.g., `thing 10000`) and prints the result and how it took to compute. Run this program for varying values of N , until you find values for which the time seems to increase more than you would expect (e.g., you double N but the time increases by a factor of four). You should probably run the program at least twice for each value of N to be sure time is fairly consistent across multiple executions. (What I noticed on my desktop machine was that the first execution for a given value of N was sometimes much slower than subsequent executions — probably because Linux is doing some kind of caching.)

(Advice for this phase: Start with $N = 10^6$ and increase by factors of 10 until you notice a dramatic increase in time, or the program tells you it can't allocate memory. Then back off to the previous value and increase by smaller amounts. Something I found interesting to do was run the command `top` in a different `xterm` window; this command shows the processes using the most CPU time. It was interesting to notice that sometimes `kswapd` or `kswapd0` was using a lot of CPU time — I believe this is a daemon process involved in Linux's memory management scheme.)

Now try to speed up the program: Make any changes you think will help, and experiment with the changed program to see whether it is faster than the original, less likely to slow down for large N , etc. Again record program output in a text file.

Submit the changed program and the text file(s) containing the recorded program output. Also include in the text file(s) some information about what computer you ran it on — if it's a department lab machine, just its name; otherwise the processor speed, cache size, and amount of memory.

⁴http://www.cs.trinity.edu/~bmassing/Classes/CS4320_2004fall/Homeworks/HW0X/Problems/thing.c