MCS-378 Final Exam

Serial #:

This exam is closed-book and mostly closed-notes. You may, however, use a single 8 1/2 by 11 sheet of paper with *hand-written* notes for reference. (Both sides of the sheet are OK.)

Please write your name only on this page. Do not turn the page until instructed, in order that everyone may have the same time. Then, be sure to look at all problems before deciding which one to do first. Some problems are easier than others, so plan your time accordingly. You have 120 minutes to work.

Write the answer to each problem on the page on which that problem appears. You may also request additional paper, which should be labeled with your test number and the problem number.

Printed name: _____

On my honor, I pledge that I have not given, received, nor tolerated others' use of unauthorized aid in completing any work for this course.

Signature for above honor pledge: _____

Problem	Page	Possible	Score
1	1	10	
2	1	10	
3	2	10	
4	2	10	
5	2	10	
6	3	10	
7	3	10	
8	4	10	
9	5	10	
10	5	10	
Total		100	

1. [**10 Points**] Consider a uniprocessor machine running a Unix-family system such as Mac OS X or Linux. Two single-threaded processes are running with different niceness levels. Both are carrying out long running, purely CPU-bound computations.

   (a) Which receives a greater proportion of the CPU time: the one with a numerically greater niceness or the one with the lesser niceness?

   (b) On the OS X system, each process has a priority as well as a niceness. Briefly explain why the priority is not the same as the niceness.

   (c) The relative proportion of CPU time each receives can adjusted by adjusting the difference in niceness levels. Suppose someone has a target ratio of CPU shares they want to achieve. Explain why the corresponding difference in niceness levels would be simpler to calculate for Linux than for Mac OS X.

   (d) Continuing the prior question, why would the difference in niceness levels be more straightforward to calculate in very recent versions of Linux (such as we discussed in class and I described in an update on the web) than in somewhat older versions of Linux (such as are described in the book)?

2. [**10 Points**]

(a) A semaphore can be used as a mutex. Which of the two semaphore operations, `down` and `up`, would be used to lock the mutex, and which would be used to unlock it?

(b) If each account at a bank is protected by a mutex, the code to transfer money from one account to the other might look something like this:

- lock the source account's mutex
- lock the destination account's mutex
- update the source account
- update the destination account
- unlock the source account's mutex
- unlock the destination account's mutex

However, with this approach, concurrent transfers could deadlock. How should the code be changed to prevent deadlock?

(c) Instead of preventing deadlock, it is possible to detect it once it occurs, abort one of the transfers, and later retry it. If transfers are done using the steps listed above, a deadlocked transfer that is aborted would not yet have updated either account. Many database systems can cleanly handle the more general case of a transaction that needs to be aborted after performing some updates. What is a common mechanism they use to support this ability?

3. [**10 Points**] Write down five different two-phase histories that would be possible for a transaction, T_1, that first reads x, then writes y, then writes x. In at least two of your five histories, avoid grouping all the unlock operations together at the end of the history.

4. [**10 Points**] Give at least two situations in which COW is used and explain in each case why it is valuable. Your explanations should be specific to each of the situations you list; do not just give one generic explanation of what makes COW valuable.

5. [**10 Points**] On a POSIX system, a file and a directory are both owned by user 42 and group 53, and both have permissions `-wxr----x`; that is, `-wx` for the owner, `r--` for the group, and `--x` for others. The members of group 53 are users 37, 42, and 71. The only other user on the system has user ID 85. (You may disregard the superuser, `root`, who has user ID 0.)

   (a) Which user(s) may read the file?

   (b) Which user(s) may write the file?

   (c) Which user(s) may execute the file?

   (d) When the file is executed by user 85, what are the two possible numeric values for the effective user ID?

   (e) What determines which of these two possible user IDs is used?

   (f) Which of the following are true?

      i. User 37 may list the contents of the directory.

      ii. User 37 may use the directory in a pathname to access files under it, subject to those files' permissions.

      iii. User 42 may list the contents of the directory.

      iv. User 42 may use the directory in a pathname to access files under it, subject to those files' permissions.

      v. User 85 may list the contents of the directory.

      vi. User 85 may use the directory in a pathname to access files under it, subject to those files' permissions.

6. [**10 Points**]

   (a) In disk space allocation, what is the difference between internal fragmentation and external fragmentation? What causes each?

   (b) If the size of a file system's disk blocks is increased, will internal fragmentation increase or decrease? Why?

   (c) If the size of a file system's disk blocks is increased, will external fragmentation increase or decrease? Why?

7. [**10 Points**] For each of the protocols, technologies, or functions listed below, identify which of the following layers it corresponds to: application, transport, network, or link.

   (a) UDP
   (b) retrieving email
   (c) Ethernet and Wi-Fi MAC addresses
   (d) IP
   (e) congestion control
   (f) TCP
   (g) routers
   (h) delivering bytes in their proper sequence
   (i) DNS
   (j) end-to-end flow-control
   (k) HTTP
   (l) retransmitting data for which no acknowledgment is received
   (m) verifying that a cached web page is up to date
   (n) port numbers
   (o) NFS

.

8. **[10 Points]**

   (a) What are the two forms of messaging supported by messaging systems (that is, by message-oriented middleware)? How do they differ from one another?

   (b) What are RPC and RMI? How do these forms of communication middleware differ from messaging systems?

   (c) What are web services? Given that the communication pattern for a typical web service is either messaging or RPC, what advantages do web services offer over prior approaches to messaging and RPC?

9. [**10 Points**]

(a) List five different ways in which an adversary could determine your password for a computer system.

(b) What are the two other major techniques for user authentication, beyond passwords? (Do not include techniques, such as two-factor authentication, that are just combinations of others.)

(c) When I try doing anything particularly security-critical on my Mac, it demands that I type in my password first: the very same password as I typed to login. What security benefit is achieved by making me enter the same password a second time?

10. [**10 Points**] Briefly define each of the following terms.

(a) Covert channel

(b) Virus

(c) Worm

(d) Trojan horse

(e) Stack smashing