Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033 Computer Systems Engineering: Spring 2007

Quiz I

There are 12 questions and 10 pages in this quiz booklet. Answer each question according to the instructions given. You have **50 minutes** to answer the questions.

Most questions are multiple-choice questions. Next to each choice, circle the word **True** or **False**, as appropriate. A correct choice will earn positive points, a wrong choice may earn negative points, and not picking a choice will score 0. The exact number of positive and negative points for each choice in a question depends on the question and choice. The maximum score for each question is given near each question; the minimum for each question is 0. Some questions are harder than others and some questions earn more points than others—you may want to skim all questions before starting.

If you find a question ambiguous, be sure to write down any assumptions you make. **Be neat and legible.** If we can't understand your answer, we can't give you credit!

Write your name in the space below AND at the bottom of each page of this booklet.

THIS IS AN OPEN BOOK, OPEN NOTES QUIZ.
NO PHONES, NO COMPUTERS, NO LAPTOPS, NO PDAS, ETC.

CIRCLE your recitation section number:

10:00	1. Madden/Komal		
11:00	2. Madden/Zhang	3. Katabi/Komal	10. Yip/Chachulski
12:00	4. Yip/Zhang	5. Katabi/Chachulski	
1:00	6. Ward/Shih	7. Girod/Schultz	
2:00	8. Ward/Schultz	9. Girod/Shih	

Do not write in the boxes below

1-4 (xx/30)	5-7 (xx/26)	8-10 (xx/22)	11-12 (xx/22)	Total (xx/100)

Name:

I Reading Questions

1. [6 points]: Which of the following statements are true of the X Windows System (as described in the X Windows paper, reading # 5)?

(Circle True or False for each choice.)

A. True / False When a program such as Xterm is run remotely and displays its window on your local workstation, that *remote* machine is considered an X server.

B. True / False In X Windows, interactive widgets such as pulldown menus and clickable buttons are implemented by the client.

C. True / False Like most client/server protocols, the X Windows protocol exclusively uses big-endian byte ordering for all network communication.

2. [8 points]: The UNIX designers chose to have the process created by a `fork()` share (inherit) the open file descriptors of its parent. Which of the following would become more complicated had they decided on an alternate `fork()` semantics, in which the child process is created with no open file descriptors?

(Circle True or False for each choice.)

A. True / False The management of standard I/O streams of child processes.

B. True / False Structure of inodes as stored on the disk.

C. True / False The format of directories as stored on the disk.

D. True / False The creation of pipes between processes.

3. [8 points]: Ben Bitdiddle is using a MapReduce cluster (as described in reading #8) to process some data. He finds that some of his programs are running slowly and needs your help to understand which performance enhancements to try. For each of the following MapReduce configurations and suggested enhancements, circle **True** if the statement about the enhancement is true, and **False** otherwise.

A. **True / False** If the performance of the system is not limited by network throughput or the rate at which map tasks can read from or write to the disk, then doubling the number of nodes in the system will probably improve performance if the number of map and reduce tasks is greater than the number of nodes.

B. **True / False** If some tasks on some workers take much longer than others to complete, and these "stragglers" are the bottleneck in the system, then allocating additional nodes to these straggler tasks will definitely improve performance.

C. **True / False** Assume that there are 1000 machines, 1000 map tasks and 100 reduce tasks, and that the CPU processing time for each map or reduce task to execute on a single machine is 1 second. If the entire 1000-node cluster can complete these MapReduce operations in 100 seconds then doubling the CPU performance (task processing rate) of each machine in the cluster may improve performance somewhat but not significantly.

D. **True / False** Suppose local disk reads and writes are 100 times as fast as network transfers and the coordinator is never able to achieve locality by assigning map workers to map input tasks stored on their local disks. In that case, if the system has an I/O bottleneck, then doubling network throughput will substantially improve performance.

4. [8 points]: Which of the following statements about operating system kernels are true?
(Circle True or False for each choice.)

A. **True / False** Preemptive scheduling allows the kernel's thread manager to run applications in a way that helps avoid fate sharing.

B. **True / False** The kernel serves as a trusted intermediary between programs running on the same computer.

C. **True / False** In an operating system that provides virtual memory, the kernel must be invoked to resolve every memory reference.

D. **True / False** When a kernel switches a processor from one application to another application, the target application sets the PMAR register appropriately after it is running in user space.

II RPC SEMANTICS

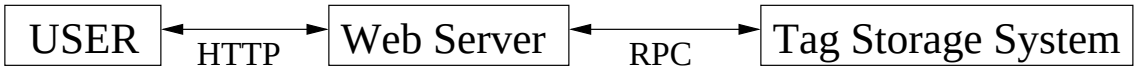
Ben is in charge of system design for Stickr, a new website for posting pictures of bumper stickers and tagging them. Luckily for him, Alyssa had recently completed implementing the Tag Storage System from design project 1. Alyssa's design supports the interface suggested by the design project, and has the following specification:

procedure FIND$(subj, rel, obj, start, count)$
// returns ok + array of matching tags

procedure INSERT$(subj, rel, obj)$
// returns ok (duplicate tags ignored)

procedure DELETE$(subj, rel, obj)$
// returns boolean, true if tag existed, false otherwise

Ben comes up with the following design:



To implement this, Ben uses an RPC interface to allow the web server to interact with the tag system. Ben chooses *at-least-once* RPC semantics. You may assume that the tag storage system does not crash (which would result in unspecified behavior), but the network between the web server and tag storage system is unreliable and may drop messages.

5. [8 points]: Suppose only a single thread on Ben's Web server is using the tag storage system and that this thread issues just one RPC at a time. What types of incorrect behavior can the Web server observe?

(Circle True or False for each choice.)

A. True / False The FIND RPC stub on the web server sometimes returns no results when matching tags exist in the storage system.

B. True / False The INSERT RPC stub on the web server sometimes returns ok without inserting the tag into the storage system.

C. True / False The DELETE RPC stub on the web server sometimes returns false when it actually deleted a tag.

D. True / False The FIND RPC stub on the web server sometimes returns tags that have been deleted.

6. [8 points]: Suppose Ben switches to *at-most-once* RPC; if no reply is received after some time, the RPC stub on the web server terminates with a timeout error code. Assume again that only a single thread on Ben's Web server is using the tag storage system and that this thread issues just one RPC at a time. What types of incorrect behavior can the Web server observe?

(Circle True or False for each choice.)

A. True / False Assuming it does not timeout, the FIND RPC stub on the web server can sometimes return no results when matching tags exist in the storage system.

B. True / False Assuming it does not timeout, the INSERT RPC stub on the web server can sometimes return ok without inserting the tag into the storage system.

C. True / False Assuming it does not timeout, the DELETE RPC stub on the web server can sometimes return false when it actually deleted a tag.

D. True / False Assuming it does not timeout, the FIND RPC stub on the web server can sometimes return tags that have been deleted.

III NAMING

Bob and Alice are using a file UNIX system as described in Appendix 2A. The file system has two disks, mounted as `/disk1` and `/disk2`. A system administrator creates a "`home`" directory containing symbolic links to the home directories of Bob and Alice via the commands:

```
mkdir /home
ln -s /disk1/alice /home/alice
ln -s /disk2/bob /home/bob
```

Subsequently, Bob types the following to his shell:

```
cd /home/alice
cd ../bob
```

and gets an error.

7. [10 points]: Which of the following best explains the problem?

(Circle the BEST answer)

A. UNIX forbids the use of "`..`" in a cd command when the current working directory contains a symbolic link.

B. Since Alice's home directory now has two parents, UNIX complains that "`..`" is ambiguous in that directory.

C. In Alice's home directory, "`..`" is a link to `/disk1`, while the directory "`bob`" is in `/disk2`.

D. Symbolic links to directories on other disks are not supported in UNIX; their call-by-name semantics allows their creation, but causes an error when they are used.

E. None of the above.

IV GOOMBLE

Observing that US legal restrictions have curtailed the booming online gambling industry, a group of former Therac programmers has launched a new venture called Goomble. Goomble's web server allows customers to establish an account, deposit funds using a credit card, and then play the Goomble game by clicking a button labeled **I FEEL LUCKY**. Every such button click debits their account by $1, until it reaches zero.

Goomble lawyers have successfully defended their game against legal challenges by arguing that there's no gambling involved: the Goomble "service" is entirely deterministic.

The initial implementation of the Goomble server uses a single thread, which causes all customer requests to be executed in some serial order. Each click on the **I FEEL LUCKY** button results in a procedure call `Lucky(account)`, where account is a pointer to a data structure representing the user's Goomble account. Among other data, the account structure includes an unsigned 32-bit integer Balance, representing the customer's current balance in dollars.

The Lucky procedure is coded as follows:

```
procedure LUCKY(Account account)
  if account.Balance > 0  then {              // line 1: test
   account.Balance ← account.Balance − 1;     // line 2: update
  }
```

The Goomble software quality control expert, Nellie Nervous, runs Eraser (as described in the Eraser paper, reading #7) on the single-threaded Goomble server code to check for race conditions; Eraser reports no problems.

8. [6 points]: Which conclusions might Nellie draw?

(Circle True or False for each choice.)

A. True / False Eraser reports no problems because it found no shared variables; each variable is accessed by a single thread.

B. True / False Eraser reports no problems because the code contains no locks.

C. True / False The single-threaded server has potential race conditions that Eraser missed because the timing details of Nellie's test run didn't expose them.

The success of the Goomble site quickly swamps their single threaded server, limiting Goomble's profits. Goomble hires an MIT-educated server performance expert, Threads Galore, to improve server throughput.

Threads modifies the server as follows: Each **I FEEL LUCKY** click request spawns a new thread, which calls `Lucky(account)` and then exits. All other requests (e.g. setting up an account, depositing, etc) are

Name:

served by a single thread. Threads argues that the bulk of the server traffic consists of player's clicking **I FEEL LUCKY**, so that his solution addresses the main performance problem.

Unfortunately, Nellie doesn't run Eraser on the multi-threaded version of the server. She is busy with development of a follow-on product: the Goomba, which simultaneously cleans out your bank account and washes your kitchen floor.

Suppose Nellie had run a brief test using Eraser with Goomble's multi-threaded server (and their million-customer audience), sufficient to run each piece of code (e.g., Lucky) many times, and to identify all shared variables.

9. [6 points]: If that test had been run, which of the following would be true?
(Circle the BEST answer)

A. Eraser would definitely find a problem.

B. Eraser would definitely find no problem.

C. Eraser might report a problem, depending on the details of the run-time scheduling of the test run.

Willie Windfall, a compulsive Goomble player, has two computers and plays Goomble simultaneously on both (using the same Goomble account). He has mortgaged his house, depleted his retirement fund and the money saved for his kid's education, and his Goomble account is nearly at zero. One morning, clicking furiously on **I FEEL LUCKY** buttons on both screens, he notices that his Goomble balance has jumped to something over four billion dollars.

10. [10 points]: Explain the source of Willie's good fortune. Give a simple scenario involving two threads (T1 and T2) with interleaved execution of lines 1 and 2 in calls to `Lucky(account)`, detailing the timing that would result in a huge *account.Balance*. We have given the first step of the scenario; fill in the subsequent steps (you may not need to use all steps.)

- Step 1. T1 evaluates "if account.Balance $>$ 0", find statement is true
- Step 2.
- Step 3.
- Step 4.
- Step 5.
- Step 6.

Name:

```
// VERSION 1
procedure LUCKY(Account account)
  ACQUIRE(global_lock);
  if account.Balance > 0 then {
   account.Balance ← account.Balance − 1;
  }
  RELEASE(global_lock);


// VERSION 2
procedure LUCKY(Account account)
  integer temp;
  ACQUIRE(account.lock);
  temp ← account.Balance;
  RELEASE(account.lock);
  if temp > 0 then {
   ACQUIRE(account.lock);
   account.Balance ← account.Balance − 1;
   RELEASE(account.lock);
  }

// VERSION 3
procedure LUCKY(Account account)
  ACQUIRE(account.lock);
  if account.Balance > 0 then {
   account.Balance ← account.Balance − 1;
  }
  RELEASE(account.lock);
```

Figure 1: Possible rewrites

Word of Willie's big win spreads rapidly, and Goomble billionaires proliferate. In a state of panic, the Goomble board calls you in as a consultant to review three possible fixes to the server code to prevent further "gifts" to Goomble customers. Each of the proposals involves use of a lock—either global or specific to an account—to rule out the unfortunate timing bug. Each of the possible rewrites of Lucky is shown in figure 1.

Name:

11. [12 points]: Which of the proposed versions eliminates the timing bug that caused Willie's big win?

(Circle True or False for each choice.)

A. True / False VERSION 1

B. True / False VERSION 2

C. True / False VERSION 3

12. [10 points]: Which version would you recommend, considering both reliability and performance goals?

(Circle the BEST answer)

A. VERSION 1

B. VERSION 2

C. VERSION 3

End of Quiz I

Name: