Name _____ Email _____ 3-digit ID _____

Systems Breadth Exam

10 February 2004

Please read **all** of the following instructions carefully.

This is a **closed-book**, **closed-notes**, written examination. You must answer *all* questions in the "Core" section. You will then answer the questions in the other two sections you have selected. Unless otherwise instructed, you must answer **all** of the questions in your chosen section. You have **three hours**.

Do not write any answers on this paper. Write **only** your *name* and *email address* on this paper. Write your answers on the blank sheets provided. Use only one side of each answer sheet. Choose a **3-digit ID code** and write it at the top of this page, next to your name. The also **write it, along with the page number, at the top of each answer sheet.** In particular, **do not put your name on your answer sheets**.

1 Core

1. Page tables can easily consume large amounts of memory. For example, consider the use of 8Kbyte pages on a machine with a 64-bit virtual address space. What technique(s) do OS designers use to keep page tables at manageable sizes?

2. Suppose you need to write code to initialize a large two-dimensional array—say, one with several thousand rows and columns. Explain why it might be important to know whether the array is stored in row-major (consecutive memory addresses go along the row) versus column-major (consecutive addresses go along the column) order. State any assumptions you make.

3. A popular program to compute horoscopes runs in 8 seconds on Whizzy Machines Incorporated's Whizmaster II, which has a 600 MHz clock speed. You have been hired by WMI to help design a new machine, the Whizmaster III, which will be capable of running the program in 4 seconds. It is possible to increase the clock speed by a substantial amount, but WMI's chip designers have told you that doing so will affect the rest of the architecture, so that the Whizmaster III will require 1.2 times as many clock cycles for this program as the Whizmaster II. For what clock rate should the designers aim in order to achieve the performance goal for the horoscope program?

4. The Whizmaster II did not include any machine instructions for setting and clearing bits in registers. Since those operations are rarely needed for computing horoscopes, the designers decided it would be better to simply simulate them in software, using other machine instructions. Show how this can be done by writing assembly code fragments that have the same effect as the instructions "`bis #i,%r`" (which sets the ith bit of register r) and "`bic #i,%r`" (which clears the ith bit of register r). You may assume the existence of the machine instructions to load a constant into a register, to copy the contents of one register into another, to add one register to another, to subtract one register from another, to shift the contents of a register in either direction (by any amount), and to conditionally branch based on the sign bit of a register. (Include comments to indicate what each instruction is doing.)

5. List as many different levels of storage hierarchy as you can think of that would be found in a typical home computer system today (say, a 1GHz processor with a 433MHz bus). For each level, give a rough indication of approximate access latency (i.e. the time required to retrieve an arbitrary bit from the store at that level).

6. (a) Explain the meaning of the term *critical section*. (b) Explain what makes a code fragment a critical section (e.g., what kind of operations or function it contains). (c) Sketch (i.e. describe briefly) a way of dealing with critical sections satisfactorily.

2 Programming Languages

1. Show how to use call-by-name to code a single procedure that can be used to (1) add elements 1 . . . 10 of an array, (2) add elements 3 . . . 12 of an array, (3) add 1^2, 2^2, . . ., 10^2, all by just changing the actual parameters. (This trick is called *Jensen's Device*.)

2. (a) Write a Lisp function that computes the nth Fibonnaci number. In other words, fill in the body for this code:

```
(function Fibonnaci (n)
   -- body goes here
 )
```

   (b) What is the complexity of your program?

3. Prolog is especially good for writing backtracking algorithms. Code the following algorithm for generating all binary trees of with n nodes in Prolog. If n is 0, there is only one binary tree: `nil`. Otherwise, for every choice of root r in $1 \ldots n$, for every binary tree b_1 with $r-1$ nodes, for every binary tree b_2 with $n-r$ nodes, generate a binary tree with b_1 and b_2 as children. You may assume a predicate `upto(A,B,C)` that generates all integers `C` between `A` and `B`, inclusive, but if you can code `upto` as well, that is even better.

3 Databases

The *Sunny Hill Valley Two-on-Two Basketbal League* holds its annual tournament just after the New Year's Day. Each basketball team consists of two players. After a complex playoff system, four best teams in the league are determined and play a round robin tournament. This year, the League officials decided to keep track of all the games in the final tournament using a specially designed database. Unfortunately, they did not really think too much about the structure of their database, and so, when the dust settled, they wound up with the following database schema:

```
Players(Name String(10), team String(10));
Games(GameId int, team String(10));
Stats(GameId int, Name String(10), points int, rebounds int, assists int,
shots int, shotsMade int);
```

Here, the primary keys are underlined. In addition Games.Name is a foreign key on Players. While no other foreign keys exist in the database, we also note that Games.GameId and Stats.GameId store the same information; similarly, Players.team and Games.team store the same information.

After the tournament was over, the database looked like this:

Players

Name	team
Alice	Bashers
Ashley	Slashers
Anne	Crushers
Amy	Blasters
Bob	Bashers
Bill	Slashers
Bruce	Crushers
Ben	Blasters

Games

GameId	team
1	Bashers
1	Slashers
2	Crushers
2	Blasters
3	Bashers
3	Crushers
4	Slashers
4	Blasters
5	Bashers
5	Blasters
6	Slashers
6	Crushers

Stats

GameId	Name	points	rebounds	assists	shots	shotsMade
1	Alice	13	12	5	16	6
1	Ashley	21	16	4	20	9
1	Bob	32	14	4	36	14
1	Bill	22	10	6	20	10
2	Anne	8	6	3	10	4
2	Amy	15	7	5	20	7
2	Bruce	26	7	3	17	11
2	Ben	22	6	5	21	10
3	Alice	20	10	8	12	10
3	Anne	15	9	6	22	7
3	Bob	16	9	7	20	8
3	Bruce	17	8	5	15	7
4	Ashley	20	19	5	20	10
4	Amy	17	6	8	19	8
4	Bill	25	27	6	34	11
4	Ben	22	23	4	46	11
5	Alice	9	17	9	30	4
5	Amy	26	15	5	22	12
5	Bob	24	20	2	32	12
5	Ben	19	15	7	20	8
6	Ashley	33	10	7	32	15
6	Anne	27	12	10	35	13
6	Bill	21	19	10	20	9
6	Bruce	30	9	9	17	15

Relation `Players` identifies the team of each participant of the final tournament. Relation `Games` specifies all the games that took place in the tournament. With each game, two participating teams are associated (as two records). Finally, relation `Stats` records for each player and each game, points scored, rebounds, assists, number of shots taken at the basket and number of shots made.

In addition to this, the rules of basketball are fairly straightforward: the team with the highest score wins the game and there **must** be a winner of each game, i.e., ties are not allowed. The League rules for determining

the champion are: (i) the team with most wins, (ii) the team with the best differential between points for and points against if two or more teams are tied in the number of wins and (iii) the team with the highest number of points scored if two or more teams coincide on (i) and (ii).

1. The designers of the basketball database did not bother with an E-R model. Restore/construct this model for them. You need to draw the E-R diagram, and explain for each relationship set what its constraints are.

2. How would you redesign this database? (You may want to work on the next problem first, and then return to answering this question).

3. You have decided to improve the database step by step. First, you want to add a new attribute, Points, to the Games table. This attribute needs to store information about the total number of points scored by the given team in the given game. Write in SQL the following commands:

   (a) The command for adding the new attribute to the relation Games.

   (b) The command that computes the number of points scored by the Slashers in each game and sets the new attribute value in the Games table accordingly.

4. Write a relational algebra expression (use P, G, S as table names) and show the answer based on the given database for the following query: Find all teams against which Alice scored 10 or more points. Return the names of the teams.

5. Write the following queries in SQL and specify their answers.

   (a) Find the field goal percentage totals for the members of the Bashers (that is: total number of shots made over total number of shots attempted). Output the name of each player and his/her field goal percentage.

   (b) Find the team with the most rebounds in the tournament. Output the team name and the total number of rebounds.

6. (This problem concerns the transaction execution schedule shown below.) Consider a database management system trying to execute concurrently transactions $T1$, $T2$, $T3$ and $T4$.

	T1	T2	T3	
1	W(B)			
2		R(A)		
3			R(B)	
4		R(C)		
5				R(C)
6	R(A)			
7		W(B)		
8	W(A)			
9			W(C)	
10				W(B)
11				W(C)
10	commit			
11		commit		
12			commit	
13				commit

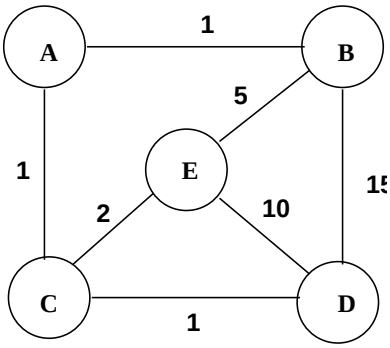
(a) Indicate all conflicts that occur in this schedule. For each conflict, indicate its type, actions and transactions affected.

(b) Indicate **all** blind writes in the schedule.

(c) Is this schedule conflict-serializable? Explain your answer in detail.

4 Networks

1. Consider the network topology shown below.

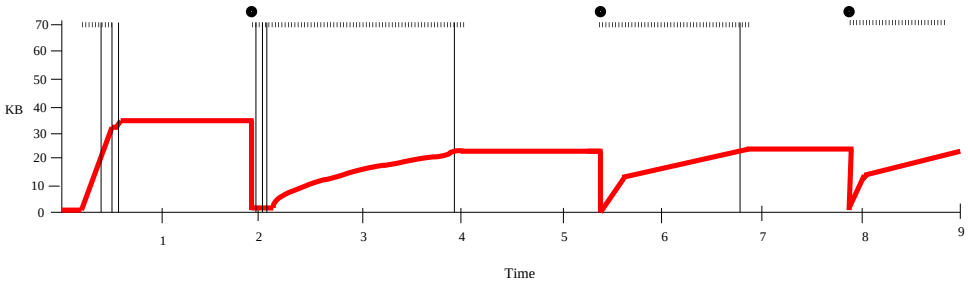


   (a) If distance vector routing is used, show the distance routing table at E after the routes have stabilized.

   (b) Suppose the link from node A to C fails. Describe the update messages that result from that failure (i.e., state the link the message goes over, and say what information is carried in the message).

   (c) If link state routing were being used instead of distance vector routing, show all the link state messages that would result from the link failure between A and C (again indicating the link traversed and message content).

2. TCP Behavior: The figure below shows the behavior of TCP's congestion window over time on a certain connection. The thick solid line is the size of the congestion window (i.e. value of *cwnd*). The hash marks at the top of the graph show when each packet is transmitted. Long vertical hash marks represent packets that were later retransmitted (i.e., were lost). The bullets at the top of the graph represent timeouts. (Note that TCP's fast retransmit/fast recover mechanism was intentionally disabled during this trace.)



   For each of the following questions, provide a **short discussion giving the reasoning behind your answer**.

   (a) Identify the interval(s) of time in which TCP is operating in slow start.

   (b) Identify the interval(s) of time in which additive increase is in effect.

   (c) What causes the change from slow start to additive increase?

(d) Why does the congestion window become constant ("flatten out") during (roughly) times 0.5–1.9, 3.9–5.3, and 6.8–7.8?

(e) If the receiver had advertised a larger initial window, what would change in this graph?

3. Suppose that N stations sharing a broadcast channel use a protocol that requires, on the average, $N/2$ slot times to sort out who transmits next. Assuming the average packet size is 8 slot times, express the fraction of the channel capacity available for transmitting data as a function of N.