SYSTEMS

Qualifying Exam — 1996	Systems	Tuesday, January 16
Total Points: 90		2:00pm to 4:00pm

1 Virtual Memory (15 points)

Describe the key characteristics and the problem solved by each of the following *virtual memory* schemes:

1. Segmentation (5 points).
2. (Pure) paging (5 points).
3. Paged segments (5 points).

2 Operating Systems (15 points)

Hoare's CSP and other concurrent programming notations support *synchronous* message-passing statements. A send statement naming channel C

$$C \,!\, expr$$

delays until a receive statement naming channel C

$$C \,?\, v$$

is executed. Then, the assignment $v := expr$ derived from the statements is executed and the processes containing the send and receive continue.

Thus, the following program first sets x to 1 and then sets y to 2, or else it first sets x to 2 and then sets y to 1:

```
(*) cobegin
          C ! 1
      ||
          C ! 2
      ||
          C ? x ; C ? y
    coend
```

Complete the outline below of a monitor that implements (on a uni-processor) synchronous message-passing statements naming channel C. Using your monitor, we might recode (*) as follows:

```
cobegin
        C.bang(1)
    ||
        C.bang(2)
    ||
        C.query(x) ; C.query(y)
coend
```

Here is the outline you should complete:

```
C = monitor
    var

    bang: procedure entry (val)
          begin



          end bang

    query: procedure entry (var target)
           begin


           target :=


           end query

    begin




    end C
```

3 Architecture: Memory system design and performance (30 points)

1. **Cache performance (15 points)**

   Consider three machines with different cache configurations (the machines are otherwise identical):

   Machine 1: direct mapped cache with one-word blocks
   Machine 2: direct mapped cache with four-word blocks
   Machine 3: 2-way set associative cache with four-word blocks

   (notes: all caches are unified data/instruction caches; a cache block is the minimal unit of transfer between the cache and main memory)

   The following set of miss rate measurements have been made on a set of benchmark programs:

   Machine 1: instruction miss rate is 4%, data miss rate is 8%
   Machine 2: instruction miss rate is 2%, data miss rate is 5%
   Machine 3: instruction miss rate is 2%, data miss rate is 4%

   In the benchmark programs one-half the instructions contain a data reference. Assume that the cache miss penalty, measured in cycles, is (6 + block_size), where the block size is measured in words. The CPI (average number of cycles per instruction) including cache misses for this workload was measured on Machine 1 and was found to be 2.0.

   Question: Determine the CPI of machines 2 and 3. Show the intermediate steps in your calculation.

2. **Cache design (15 points)**

   You are to choose a cache organization for a new workstation you are building. Your alternatives are:

   Design 1: a 1-level cache design:
   level 1 cache: 64Kbytes, direct mapped, 1 cycle access time,
   Main memory: 20 cycle access time

   Design 2: a 2-level cache design:
   level 1 cache (fastest): 64Kbytes, direct mapped, 1 cycle access time
   level 2 cache (slowest): 1Mbytes, 4-way set associative, 4 cycle access time,
   Main memory: 20 cycle access time

   Notes:

   - The cache access times are equal for hits and misses.
   - In a 2-level design, a miss at level 1 gets filled by an access to level 2; a miss at level 2 gets filled by an access to main memory.
   - Both designs use the same block size, the same write policy, and are built using the same technology.

Question: How do you trade-off between the two designs? Design 2 seems clearly better than design 1, given that both use the same first-level cache and design 2 has an additional 2nd level cache. Are there any disadvantages to design 2? Explain which and relate your answer to measurable quantities such as miss rates, access times, circuit speed, circuit cost, etc... Be brief.

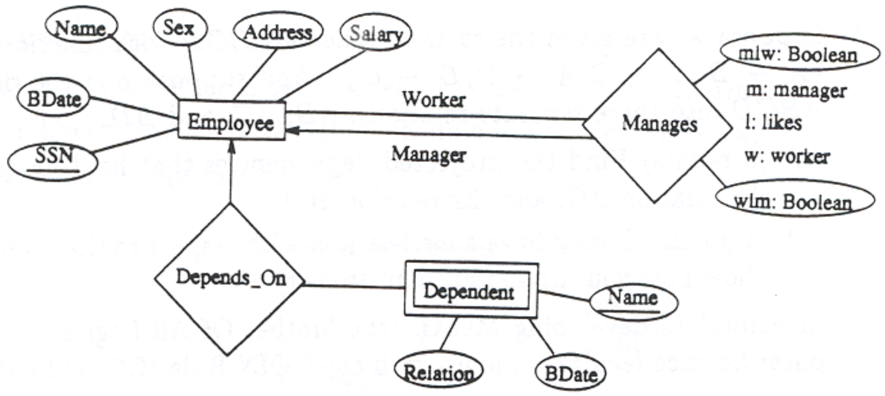

Figure 1: ER-diagram for problem DB-1.

4 Databases (30 points)

DB-1 The entity-relationship (ER) model is used in initial database design. Based on a perception of the real world, the ER-model contains *entities* which are basic objects in our world, and *relationships* between entities. Entities have attributes. Relationships can also have attributes.

A database design in the ER-model is depicted by an ER-diagram. In an ER-diagram, entities are represented by a rectangle, relationships by a diamond and ellipses represent attributes. The underlined attributes form the *key* for an entity/relationship. *Weak entities* are represented by double rectangles.

Each entity/relationship in an ER-diagram can be converted into a table, where each table is a bunch of (unordered) attributes. A table can be depicted by

Attribute-1	Attribute-2	...	Attribute-n

.

Using the ER-diagram in Figure 1,

1. (3 points) Write tables for entity sets *Employee* and *Dependent*.
2. (2 points) Write table for relationship set *Manages*. Invent any attribute names that you need.
3. Express the following queries in one of the following languages: relational algebra, tuple relational calculus, domain relational calculus, or SQL. Use the tables (relations) developed in parts one and two.
   (a) (2 points) Find names of all employees who earn more that $60,000.
   (b) (3 points) Find names and salaries of managers who are not liked by at least one of his/her workers.

DB-2

1. Suppose we are given the relation scheme $ABCD$ with functional dependencies $\{A \rightarrow B, B \rightarrow C, A \rightarrow D, D \rightarrow C\}$. And suppose ρ is the decomposition of $ABCD$ into three separate relations, AB, AC, and BD.

   (a) (3 points) Find the projected dependencies that hold on the relation AB, the relation AC, and the relation BD.

   (b) (4 points) Does ρ have a lossless join with respect to the given dependencies? Show how you arrived at your answer.

2. (3 points) In developing MOAL (the Mother Of All Logics), the Cornell Computer Science faculty came up with the COIN Rule (COrnell INference Rule)

$$\text{if } X \rightarrow Y \text{ and } Z \rightarrow Y \text{ then } X \rightarrow Z$$

   Show that you are smarter than the faculty by coming up with a simple counter-example, *i.e.*, a relation R (with tuples to illustrate your point) that satisfies $X \rightarrow Y$ and $Z \rightarrow Y$, but not $X \rightarrow Z$.

DB-3 1. (2 points) True or False

(a) Every view serializable schedule is also conflict serializable.

(b) Every conflict serializable schedule is also view serializable.

2. Consider the following schedule:

T_0	T_1	T_2
		read(D)
		read(C)
read(A)		
write(A)		
write(B)		
		write(B)
		write(C)
		write(D)
	read(C)	
read(D)		
write(D)		
	read(D)	
	read(A)	
	write(A)	
	write(B)	
	write(D)	

(a) (4 points) Is this schedule conflict serializable? If yes, give an equivalent serial schedule ($T_i \rightarrow T_j \rightarrow T_k$). If not, why not?

(b) (4 points) Is this schedule view serializable? If yes, give an equivalent serial schedule ($T_i \rightarrow T_j \rightarrow T_k$). If not, why not?