Name _____

CS Systems Breadth Exam
February 11, 2005

The first part of the exam covers **Core** topics in basic operating system concepts and machine architecture/organization. You must answer all questions in the **Core** part of the exam. In addition to the **Core** area, you must select two areas from the list of **Databases**, **Distributed/Advanced Operating Systems**, **Compilers**, and **Programming Languages**. You must answer all questions in the two areas that you select.

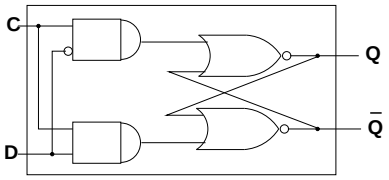
You have 3 hours to complete the exam. The exam is closed note and closed book. Similarly, calculators and other electronic devices (PDAs, cell phones, etc) are not allowed.

1 Core Questions

1. Page Replacement

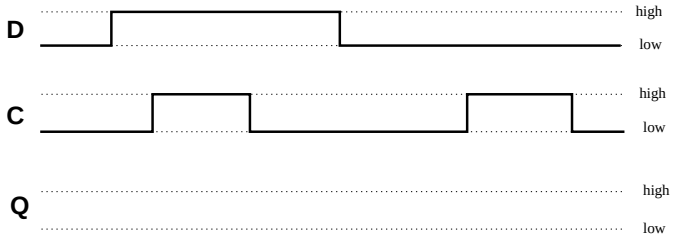
   - Describe the optimal page-replacement algorithm.
   - Explain why LRU and LFU are NOT optimal by giving an example reference string (sequence of page numbers referenced).

2. Consider the D latch implemented with NOR gates below.



   (a) What is the purpose of a D latch? What is it used for?

   (b) Given the values for C and D below, show how the value of Q changes over time.



3. Consider an application that consists of multiple **producer** processes and multiple **consumer** processes. The code for the producers and consumers (without synchronization) is shown below. Rewrite the code using **semphores** to synchronize the producers and consumers.

```
Producer
--------
do {
  p_item = GetNewItem();
  BoundedBuffer[NextFree++] = p_item;
  if (NextFree>=BUFSIZE) NextFree = 0;
} while (1);
```

```
Consumer
--------
do {
  c_item = BoundedBuffer[NextUsed++];
  printf("Item = %d\n", c_item);
  if (NextUsed>=BUFSIZE) NextUsed = 0;
} while (1);
```

2 Distributed/Advanced Operating Systems

1. State whether each of the following statements is true or false. True means ”always true”, false means ”sometimes true or never true”. Justify your answer.

   (a) Suppose a distributed shared memory system DSM1 implements a stronger consistency model than the consistency model implemented by another distributed shared memory system DSM2, then any program written for DSM1 would work correctly if run under DSM2.

   (b) Timestamping events that occur in a distributed computation using Lamport’s clock helps in ordering those events in a unique way.

   (c) Sender-initiated load-sharing algorithms cause preemptive task transfers.

2. Describe the Chandy-Lamport Global State Recording Algorithm. What good is the global state recorded by this algorithm when the distributed computation might not have passed through this recorded global state?

3 Databases

1. Consider the following database schema used by a software manufacturer:

```
Aircraft (Manufacturer, Model, NumSeats)
Owns (Airline, AircraftId, Model, Cost, PurchaseDate)
Flys (PNum, PilotName, Age, SS#, Model)


Model -> Manufacturer, NumSeats
AircraftId -> Model, Cost, PurchaseDate, Airline
PNum -> Pilotname, Age, SS#
SS# -> PilotName, Age
```

   (a) Is the database schema given above a good design? If not describe the alternative schema that you would use and briefly discuss why your design is an improvement.

   (b) Using either the given database or your “improved” design formulate answers to the following into any query language (e.g., SQL, QBE or Quel, relational algebra, or relational calculus). It is acceptable to formulate several simple expressions to produce the requested result.

      i. List every manufacturer of planes flown by “Delta” airlines.

      ii. For each pilot list the number of planes (Models) that he or she flies.

      iii. American Airlines has decided to shorten the aisles and to reduce the leg room. This will give each plane 10% more seats. Write a command to increase the number of seats on all their planes by 10%.

      iv. Finally SouthWest airlines has a number of job openings so they want a list of all Pilots who are candidates for flying their planes. To make their scheduling easier they only want to consider pilots who can fly all of their models. Please generate their list.

2. Consider the following transaction log.

```
10 begin_checkpoint
20 end_checkpoint
30  T1 update P1
40  T2 update P1
50  T3 update P2
60  T4 update P3
70  T1 update P4
80  T2 update P5
90  T1 update P3
100 begin_checkpoint
110 end_checkpoint
120 T5 update P3
130 T6 update P2
140 T3 update P6
150 T2 abort
160 T4 update P5
170 T1 commit
180 T5 update P4
190 T6 update P2
200 begin_checkpoint
210 end_checkpoint
220 T2 undo
230 T1 end
240 T3 commit
250 T4 commit
260 T6 update P1
       CRASH!!!!!!
```

Answer the following questions.

(a) Restore the values of the $PrevLSN$ field for all log entries. (use schedule above to show your work) Restore the values of $NextUndoLSN$ where required.

(b) Suppose that transaction table and DPT stored in the `end_checkpoint` record with LSN 20 are both empty. Assume also that between records with LSNs 10 and 260, no buffer pages had been flushed to disk. Show the contents of the transaction table and the dirty page table stored in the records with LSNs 110 and 210.

(c) Suppose that the DPT stored in record with LSN 210 has the following contents:

PageID	RecLSN
P2	130
P3	60
P4	70
P6	140

Construct the DPT and the transaction table in at the moment of the crash (assuming no more buffer flushes until the crash). Which record shall the ReDo stage of the AIRIES algorithm commence with? Which updates will be redone? For each update that will NOT be redone, briefly specify the reason.

(d) Which transactions will be rolled-back by the Undo stage of the AIRIES algorithm? Which updates will be undone? Write the first three Compensation Log Records (including $PrevLSN$ and $NextUndoLSN$ fields).

4 Compilers

1. Provide complete discussion and concrete examples that answer the following questions:

   (a) Define the concepts of dynamic scoping and static scoping.

   (b) Give a small program (pseudocode rather that a complete program in a specific programming language) illustrating the difference between the dynamic and static scoping.

   (c) Discuss in the context of compilers, focusing on the run-time memory management, how to implement static and how to implement dynamic scoping. Discuss differences and similarities, pros and cons.

   (d) List example programming languages indicating the type of scoping assumed by the semantics of the language.

2. Provide complete discussion and concrete examples that answer the following questions:

   (a) Define the concept of ambiguity for Context Free Grammars and Context Free Languages. Discuss these concepts from the perspective of parsing and compiling programming languages.

   (b) Provide two grammars for arithmetic expressions, one ambiguous and one unambiguous. For each grammar show derivations and parse trees for $id + id * (id - num) + num$. Justify that one of your grammars is indeed ambiguous.

5 Programming Languages

1. Write a Prolog program that outputs all the integers between 1 and *n*, inclusive. You may use a unary `write` predicate.

2. Procedures

   (a) Define the concept of a **first-class procedure**. Your definition should apply both to imperative and to functional languages.

   (b) First class procedures, in the presence of nested procedure declarations and deep binding, lead to what problem? Describe precisely what the problem is; just giving it a name is not sufficient.

   (c) Why does this problem not occur in C?

   (d) How can a language solve this problem but still allow any procedure to be first-class, allow procedures to nest, and employ deep binding?