ECE428/CS425/CSE424: Distributed Systems **Spring 2011**

Homework 3 **Due: by 2 pm on 03/15/2011**

Total points: 70

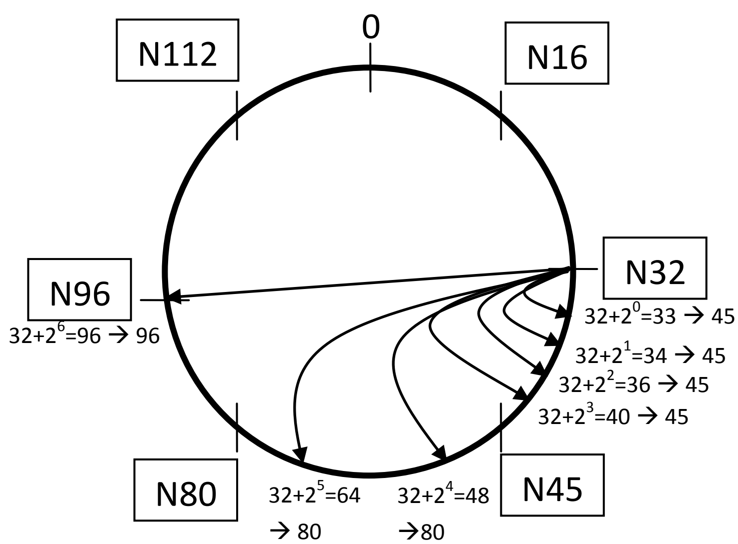
1. (20 points) Suggest a message-passing mutual exclusion algorithm in which nodes enter critical section in the happened-before order. That is, if a node A's entry operation happens-before that of node B, then node A must enter critical section first. Explain why you believe that your algorithm is correct. Assume that the system is asynchronous, but there are no failures. Also assume that all channels are reliable and FIFO.

   If you believe that such an algorithm does not exist, explain why.

   We can use the Ricart & Agrawala's algorithm described in the textbook. The happened-before order is ensured by the Lamport's logical timestamp. This is because when event *a* happens-before event *b*, the logical timestamp(*a*) < timestamp(*b*). Therefore, if a node A's entry operation happens-before node B, the timestamp(A) < timestamp(B), implying that A will get access to the critical section before B.

2. (15 points) Consider slide 4 in the set of slides used for Chord (distributed hash tables) in lecture 11.

   a. Pictorially show all the fingers at node with id 32.

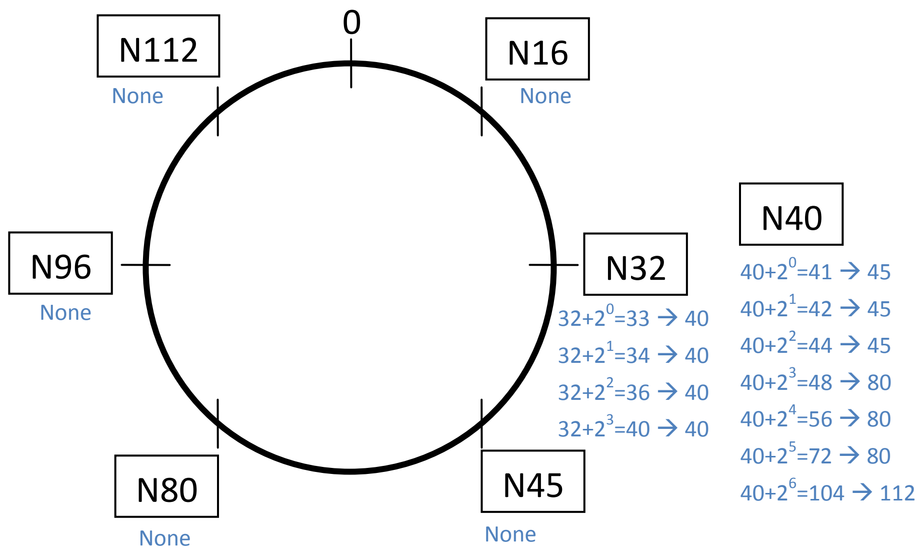


   b. Determine where files with the following keys will be stored: 3, 43, 88, 100.

   The file with key 3 →N16, The file with key 43 → N45 , The file with key 88 → N96,
   The file with key 100 → 112

   c. Suppose that node with id 40 is added to the network. Show the fingers that need to be modified due to this addition (that is, the fingers that will point to node 40 in the modified networks, and the fingers at node 40).

   Blue font indicates the modified fingers.



3. (15 points) Consider the following code to be executed by every processor in a shared memory system.

```
Initially, shared memory variable V equals 0
    <Entry>:
    1: wait until V=0
    2: V := V + 1
    3: wait until V < 2

    <Critical Section>

    <Exit>:
    4: V := 0
```

Will the above code for Entry and Exit ensure mutual exclusion? *Explain your answer*. Assume that no failures occur, and that the system is asynchronous.

No. Consider two processes, p1 and p2 executing line 1 concurrently. Because V is initially 0, both p1 and p2 will execute line 2. Let p1 succeed first in updating the variable V in line 2, and enter the critical section (value of V will now be 1). In the meantime, p2 will also increment V in line 2 (V becomes 2) and will be waiting in line 3 for the condition (V < 2) to become true. When p1 exits the critical section and resets the value of V to be 0 at line 4, p2 will enter the critical section without modifying the variable V. In the meantime, consider another process p3 executing line 1. Because V = 0, p3 (after executing lines 2 and 3) will also enter the critical section. Because p2 is already in critical section, this violates mutual exclusion.

4. (10 points) In the ring-based leader election algorithm discussed in class, the node with the largest identifier (or attribute) becomes the leader. How can this algorithm be modified such that the probability that a given node will become a leader is identical for all nodes. Assume that no failures occur.

1) A node sends an "election" message along with its id to initiate the leader election process.

2) Each node inserts its own id to the "election" message and forwards the message to the next node.

3) When the node that initiated the leader election receives the "election" message, the node chooses a node id randomly from the ids in the "election" message, and sends out an "elected" message with the chosen id.

4) Upon receiving the "elected" message, a node chooses itself as the leader if its id is equal to the id in the received "elected" message and then forwards the message. If not, the node simply forwards the message.

5) When the node that initiated the leader election gets back the "elected" message, it simply drops it.

5. (10 points) *This question is based on material to be covered in the lecture on March 3, 2011.*
Briefly discuss two differences between shared memory in a bus-based shared memory multiprocessor and a distributed shared memory

1) **In a distributed shared memory system, the shared address space is spread over all participating computers, while in a bus-based shared memory system, the shared address space is centralized in a shared memory module.**

2) **In a distributed shared memory system, message exchanges may occur over a wide area network.**