PRINT FAMILY NAME:____________________________

Examination 1
COP 5615 (CSR 740-R) - Operating System Principles
4 March 1999

Instructions:

- **Read all these instructions; failure to follow instructions will result in loss of points.**
- Do not start the examination until instructed to do so.
- This test is open book, open notes.
- **Answer any five questions below.**
- **Start the answer to each question on a new page** (i.e., do NOT put the answer to more than one question on the same page).
- Show your work.
- **Assemble your answers in numerical order of the questions when you submit them.**
- **Leave a 1" square in the upper left corner for a staple.**
- Be sure to include your name on your answer sheets.
- You have 50 minutes to complete this examination.
- **Read and sign the following statement.** You may write this on your exam and sign it there if you wish to take the exam questions home with you today. Do not discuss this exam with anyone in this course who has not yet taken this exam.

On my honor, I have neither given nor received unauthorized aid on this examination, and I will not discuss the contents of this examination with any student who has not yet taken this examination.

Signed:

1. (a) (2) What is meant by "software bus" in the description of cooperative autonomous systems? How is it implemented?
   (b) (2) How is the software bus used? Give an example. I/O?

2. (a) (4) Define and relate the terms directory information base (DIB), directory information tree (DIT), naming domains and naming contexts. Give examples that relate the terms.
   (b) (1) How does the DIT relate to network structure?

3. (5) Develop a solution to the strong reader preference version of the Reader/Writers problem using one of CCR, Linda, or monitors.

4. (a) (2) Give an example of when a total ordering of messages is required as opposed to a causal ordering. Justify your answer.
   (b) (3) Describe, compare and contrast the vector logical clock and the vector sequence number.

5. Consider an instance of the logical tree structure for token-based mutual exclusion with seven nodes, labeled in BFS order from the root (node 1) to the right-most leaf (node 7) (see Fig. 1). Node 1 holds the token for the critical section and all request queues are empty. Requests to claim the token arrive every one unit time apart in the order (7,5,4,-,6,1,2,3) (i.e., node 7 generates a request at time 1 and node 6 generates a request at time 5). Assume that the communication delay between each pair of nodes is one unit time, and that each node first receives incoming messages (if two messages arrive at the same time, they are processed in the order of the sender's node number), then performs local processing instantly, and then sends messages as needed (e.g, if a node is sent a request at time t, then it receives the request, updates its local structures and sends messages if required at time t+1.)
   (a) (3) Show the state of the request queue at each node after node 7 has obtained the token and has been in its critical section for 4 units of time.
   (b) (2) In what order will the nodes enter their critical section?

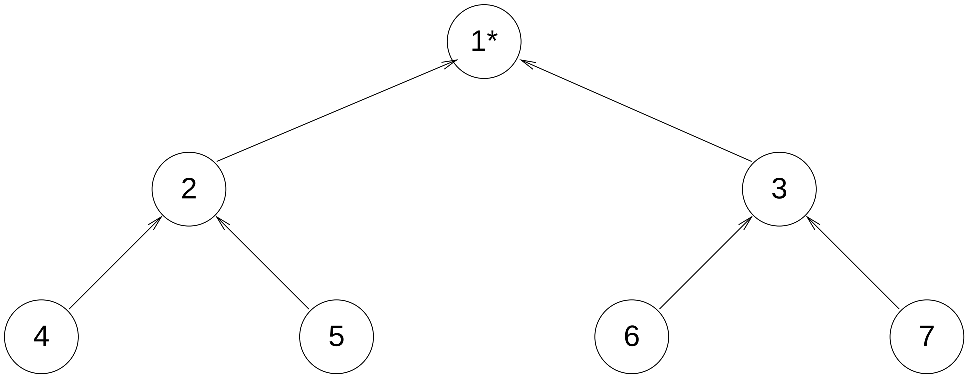

Figure 1: Logical tree for token-based mutual exclusion

6. (a) (3) Paramenter passing in RPC is typically done by value or by copy-and-restore. Why is call by reference typically not done? How is call by copy-and-restore different in terms of its effect? When would call by reference be most desirable?
   (b) (2) How could call by reference be implemented in RPC?

7. (a) (5) Can Leader Election be used for distributed mutual exclusion? If so, how? If not, why not? Be specific, and consider the case in which k of n processes contend for the critical section. Note any deficiencies of this approach.

8. (a) (2) What is makespan? Why is it popular for static scheduling?
   (b) (3) Why is makespan less appropriate for dynamic scheduling? What is a more appropriate metric for dynamic scheduling? Why?

9. (a) (3) What is priority inversion? Give two methods to handle it and state how they accomplish their goals.
   (b) (2) Does the following periodic realtime task set have a feasible static priority assignment: ((1,4),(2,5),(3,10))? Prove your answer. (You may use the timeline for scheduling provided.)

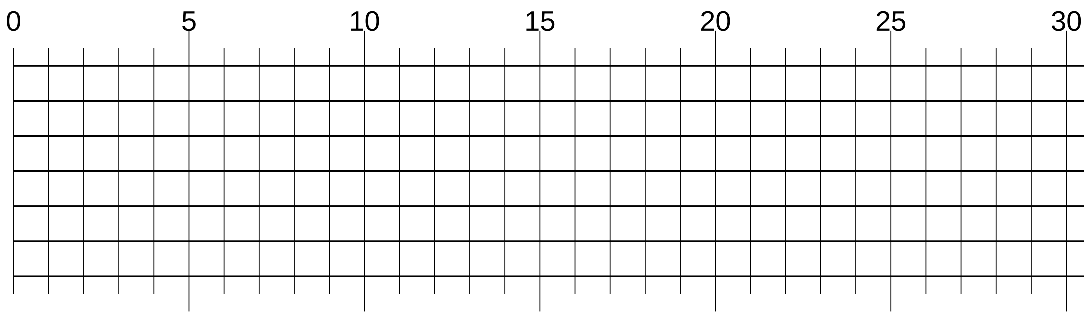

Figure 2: Timeline for scheduling