EE 382N, Sample Test 1, Vijay K. Garg, Fall 2004

NAME:

Q. 1 **(12 points)** Please be concise in your answers.
(a) Define the Lamport's happened before relation on the set of events in a distributed computation.

(b) What are the main differences between the UDP protocol and the TCP protocol?

(c) What is the role of a client stub object in Java RMI? How are non-remote objects passed as parameters for java remote method invocation?

(d) Let E be the set of events timestamped using Lamport's clock C. What is the property guaranteed by C? Let V be the vector clock function on E. What is the property guaranteed by V?

(e) What is the message complexity of Lamport's and Ricart and Agrawala's algorithms per citical section invocation? Assume that there are N processes in the system. Justify your answer.

(f) What does a process do on receiving a marker in Chandy and Lamport's algorithm.

Q. 2 **(8 points)** (a) What is a stable predicate?

Which of the following predicates are stable: (Justify your answer)
(b) The token is lost in a token ring system.

(c) There are two processes in the critical section.

(d) There are two messages in the channel from P_1 to P_2.

Q. 3 **(6 points)**
Consider the predicate B equal to "None of the processes has the token."
in a distributed system. Answer the following questions with justification.
(a, 3 points) Is this predicate stable?

(b, 3 points) Can it be expressed as a conjunction of local predicates?

Q. 4 **(4 points)** Why is it useful to model the behavior of a distributed system as a partial order of events rather than a total order of events?

Q.5 **(6 points)** (a) What is the requirement on fork placement in the dining philosopher algorithm?

(b) Under what condition, a philosopher releases a fork that it is holding when it receives a request for that fork.

Q.6 **(10 points)** Consider the following algorithm for mutual exclusion in which we only require safety and liveness. There is a token in the system. Whoever has the token can enter the critical section. The token keeps a queue of all pending requests. Whenever the process holding the token finishes the critical section, it forwards the token (with the associated request queue) to the first process in the request queue and deletes that request from the queue. If the queue is empty it keeps the token till it receives a request. A process requesting the critical section is required to send its request message to all processes. On receiving a request message, if a process has the token it inserts the request in the queue. If the process does not have the token, then it simply discards the request. Does the algorithm satisfy
(a, 3 points) safety property:mutual exclusion is not violated
(b, 7 points) liveness property:every request is eventually served.
Justify your answers. Give "informal" reasoning if the property is satisfied and a counter-example if it is not. Assume that all channels are reliable and FIFO and processes do not crash.

Q.7 **(15 points)** Assume that you want to maintain a replicated queue at n processes in a distributed system. There are two operations allowed on a queue:`enqueue` and `dequeue`. The `enqueue` operation inserts a given item at the tail of the queue. The `dequeue` operation removes the item from the head of the queue and returns it. If the queue is empty, the `dequeue` operation returns null. A client can connect to any of the processes and request it to perform that operation. Different clients may issue commands to different processes. How will you ensure that queues are identical at all sites (all operations are performed at all sites in the same order). Just give the main ideas of the algorithm.