Name: _____

CS 6/73201 **Exam #2** Advanced OS

Monday 7 November 2005

1. **Consider Lamport's algorithm for mutual exclusion in a distributed environment.**

   a. **Under what conditions does a process enter the critical section? (10 points)**

      When its own request is at the top of its own request queue and it has received a reply message (with a timestamp larger than its request) from all the other processes in the request set.

   b. **Since every process decides on its own when to enter the critical section (i.e., no central coordinator is making this decision for it), how do the conditions above prevent multiple processes from deciding independently to enter the critical section at the same time? (10 points)**

      A process only sends the reply message after it adds the request to its own request queue, so sending the reply message acknowledges that it has received the request and has properly inserted it into its request queue. Once reply messages have been received from all the other processes, those processes must have the requesting processes at the same position in their request queues, so if it is at the head of the requester's request queue, it must also be at the head of the other processes' request queues.

2. **Many of the algorithms for mutual exclusion in a distributed environment, including Lamport's algorithm, Ricart and Agrawala's algorithm, and even Suzuki and Kasimi's algorithm, share some common disadvantages. What are those disadvantages, and how are they overcome by Raymond's tree algorithm? (10 points)**

   They all have multiple points of failure and multiple bottlenecks. With the structure in Raymond's tree algorithm, only processes in the path back to the common ancestor between a requesting process and the process holding the token are involved in the algorithm, so if processes outside that path fail or are overloaded the algorithm is not affected. Further, Raymond's tree algorithm has less message traffic since messages do not need to go to every process in the request set.

3. **Briefly define false deadlock, and then explain why reporting false deadlock is a problem. (10 points)**

   False deadlock, or phantom deadlock, is deadlock that does not really exist, usually due to some out of date information being used to make the decision of whether or not deadlock is

present. Reporting false deadlock is a problem, because it may result in a process being unnecessarily terminated to recover from this non-existent deadlock.

4. Summarize the Chandy, Misra, and Haas Edge-Chasing algorithm for deadlock detection in a distributed environment. (20 points)

When a process has to wait for a process and blocks, it sends a probe message to the process holding the resource; that probe message contains the ID of the process that blocked, the ID of the process sending the message, and the ID of the process the message was sent to. When a blocked process receives a probe, it propagates it to the process(es) holding resources that it has requested, updating the last two IDs in the probe. If the original blocked process receives its own probe, there path that probe followed is a deadlock.

5. One method for deadlock prevention tries to avoid the "hold and wait" condition by forcing a process that wants a new resource to release all current resources and then request the new one plus all the old ones at the same time. What are the problems with this method? (8 points)

Other processes may also be waiting on the released resources, so even if the new resource is available, the process may not be able to immediately get its previous resources back because another process may get them first. If this situation continues indefinitely, the process may starve (never get the requested resources). Further, if all resources acquired are held until the end of the program's execution, the algorithm wastefully ties up resources and reduces concurrency.

6. One implementation method for atomic transactions uses a writeahead log with immediate update.

a. In this method, what is written in the log, and when is the log written with respect to when the data is updated? (10 points)

Written in the log: ID of transaction making the change, what is being changed, and old and new data values. The log is written and then the actual data is updated immediately.

b. In this method, what happens if the transaction commits? What happens if the transaction aborts? (6 points)

If the transaction commits, nothing else need be done since the data has already been written. If the transaction aborts, the log must be used to roll the data back to its original value(s).

7. One method for concurrency control is two-phase locking.

Name: _____

a. Briefly explain the two phases of this method. (10 points)

In the growing phase, the transaction requests new locks but doesn't release any of its existing locks. In the shrinking phase, the transaction releases its existing locks but doesn't request any new locks.

b. Explain why this method increases concurrency over static locking. (6 points)

Two-phase locking acquires and releases locks gradually whereas static locking acquires all locks at once (e.g., when the first lock is needed) and releases all locks at once (e.g., when the last lock is no longer needed), so two-phase locking permits concurrency during these growing and shrinking phases whereas static locking keeps the locks fully utilized during that same period and permits no concurrency.