Mohit Joshi
msj696

HW #3

1. Currently, Lamport's algorithm states that in order for a process to enter its critical section, its request must be at the head of the queue and it must have received acknowledgements from every process with a larger timestamp t.

In order to allow concurrent reads and writes, we can modify the first requirement for processes wanting to read. The new requirements are:

Its request must be ahead of all write requests in the queue and it must have received acknowledgements from every process with a larger timestamp t.

This allows for concurrent read and write accesses. The reason we changed the algorithm for read requests and not for write requests is that writes cannot happen concurrently with other writes.

2. a) In order to allow at most k processes, we can again modify the requirements for a process wanting to enter its critical section. The new requirements are:

The number of processes with timestamp < t plus the number of processes that have not sent the current process an acknowledgement must be less than k.

This will guarantee that at most k processes can be in the critical section concurrently

b) In order for a process using the Ricart/Argawala algorithm to gain access to the critical section, it must request the resource and receive the "okay" message from every other process.

To extend this algorithm to satisfy the k-mutual exclusion problem, we only have to modify the condition for entering the critical section. The new requirements are:

The process requested the resource and received the "okay" message from all but (k-1) processes.

In other words, the process must receive (Total#Processes - (K - 1)) messages before it is allowed access to the critical section.