Julian Domingo : jad5348

Alec Bargas : apb873

1. To satisfy allowing a read access to happen concurrently with either a read access or another write access, the read request would have to ensure it has a timestamp that is less than all other write request timestamps, in addition to receiving an acknowledgement from all other processes in the queue.
   1. Justification: allows multiple reads to happen at the same time, while the write algorithm remains untouched, meaning it cannot do any concurrent accesses with another read or write access.
2. Extending mutual exclusion algorithms:
   1. A process **p’** with a request timestamp **t’** can enter the critical section if there are either less than **k** number of processes which all have smaller request timestamps than **t’**.
      1. Justification: if a process is in the first **k** in the queue, it is guaranteed that no more than the first **k** processes will enter the critical section.
   2. With at most **k** processes in the critical section, a process that requests the critical section must wait for at least **N – k** “okay” responses before it enters the critical section, which are enough to determine that not all of the **k** units of resources are in use where **N** is the total number of processes in the distributed system.
      1. Justification: **N – k** “okay” responses are enough to determine that not all of the **k** units of resources are in use where **N** is the total number of processes in the distributed system.