

CECS 327 Assignment 6 - Coordination

20 points

Assignment Description. Answer the following questions from the Chapter 6 reading from your textbook. Be thorough and complete with your answers. You *may* work on these questions with a partner (no more than two working together), but **both** students must submit the document individually on Beachboard Dropbox along with both students' names on each submission.

1. Consider the behavior of two machines in a distributed system. Both have clocks that are supposed to tick 1000 times per millisecond. One of them actually does, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?
2. One of the modern devices that have (silently) crept into distributed systems are GPS receivers. Give a few examples of distributed applications that can use GPS information.
3. When a node synchronizes its clock to that of another node, it is generally a good idea to take previous measurements into account as well. Why? Also, give an example of how such past readings could be taken into account.
4. Consider a communication layer in which messages are delivered only in the order that they were sent. Give an example in which even this ordering is unnecessarily restrictive.
5. To achieve totally-ordered multicasting with Lamport timestamps, is it strictly necessary that each message is acknowledged? Why?
6. Many distributed algorithms require the use of a coordinating process. To what extent can such algorithms actually be considered distributed? Explain and discuss.
7. In the centralized approach to mutual exclusion (**Fig. 6-15** in *DS:PEP*, 3rd Ed.), upon receiving a message from a process releasing its exclusive access to the resources it was using, the coordinator normally grants permission to the first process on the queue. Give another possible algorithm for the coordinator and explain.
8. Consider **Fig. 6-15** again. Suppose that the coordinator crashes. Does this always bring the system down? If not, under what circumstances does this happen? Is there any way to avoid the problem and make the system able to tolerate coordinator crashes?
9. A distributed system may have multiple, independent resources. Imagine that process 0 wants to access resource *A* and process 1 wants to access resource *B*. Can Ricart and Agrawala's algorithm lead to deadlocks? Explain your answer.
10. Suppose that two processes detect the demise of the coordinator simultaneously and both decide to hold an election using the bully algorithm. What happens?

Deliverables. Submit the answers to the questions on **Beachboard Dropbox** by the indicated due date and time. Acceptable file submission formats are: .txt, .rtf, .odt, .doc, .docx, or .pdf.