# CSCI 604 Distributed Computer Systems Architecture

# Unit 6

|  |  |  |
| --- | --- | --- |
| Teacher: | George Rudolph |  |
| Subject: | Coordination and agreement among distributed processes |  |
| Objective 1: | Analyze the impossibility of consensus theorems and their practical implications |  |
| Objective 2: | Introduce algorithms for distributed mutual exclusion and locks |  |
| Objective 3: | Discuss elections and voting algorithms |  |
| Objective 4: | Discuss coordination and agreement among members of a group |  |
| Objective 5: | Extend transactions to distributed systems |  |
| Objective 6: | Introduce schemes for distributed concurrency control and timestamp ordering |  |
| Due Date: | Oct 30 |  |

## C Level Maximum 50 points

Your written answers may include pictures, drawings, short phrases instead of sentences, as long as the answer is clear and says what you mean.

1. Make flashcards for the terms and concepts. (10pts)
2. List the failure assumptions and failure detectors used in Chapter 15. For each, give a 1 or 2 sentence description of what it is, or its impact. (10 pts)
3. What issues make distributed mutual exclusion more complex than single-machine mutual exclusion? (5pts)
4. Create a flyer illustrating the different mutual exclusion algorithms and highlight strengths and weaknesses. (10 pts)
5. Create a flyer illustrating the different election algorithms and highlight strengths and weaknesses. (10 pts)
6. (You cannot do this one and #7) List and define the three main types of multicast ordering requirements. For each, give an example application that requires that type of ordering. (10 pts)
7. (You cannot do this one and #6) Create a flyer showing 3 algorithms for ordered multicast. (10 pts)
8. Create a flyer advertising 3 of the possible variations of the consensus problem from Chapter 15 as if they were flavors of Baskin-Robbins ice cream. (10 pts)
9. Create a flyer describing Lampson’s failure model for transactions and highlight its features. (5 pts)
10. List and define the ACID properties for a transaction. Which properties are typically relaxed for distributed transactions, and how are they relaxed. Give an example database with relaxed properties. (10 pts)
11. Make a flyer illustrating the concepts of recovery and recoverability as described in Chapters 15 and 16 of your text. (10 pts)
12. Pick one algorithm for deadlock prevention and describe how it works. (10 pts)
13. How does optimistic concurrency control work? (10 pts)
14. What are the advantages and drawbacks of multiversion timestamp ordering as compared to ordinary timestamp ordering? (10 pts)

## B Level 30 points—Choose Two

You may use, Java, Mule or any other tools to implement the code for the problems that specify implementations,

as long as you specify what code you wrote and you can demonstrate it working. For these purposes, XML descriptors and other “metacode” or configuration stuff you write count as code.

1. The TaskBag is a service whose functionality is to provide a repository for “task descriptions”. It enables clients running in several computers to carry out parts of a computation in parallel. A *master process* places descriptions of subtasks of a computation in the TaskBag, and *worker* processes select tasks from the TaskBag and execute them, returning descriptions of the results to the TaskBag. The *master* collects and combines the results to produce the final result.

The TaskBag service provides the following operations:

putTask allows a client to add a task description to the bag;

takeTask allows a client to take a task description out of the bag.

Assume a client makes a request *takeTask* when a task is not available, but may be soon. Implement and demonstrate the server and a test client with the following alternative policies:

* 1. The server replies immediately, telling the client to try again later.
  2. The and client must wait until a task is available.
  3. Callbacks are used. (may be difficult)

1. A server manages objects a1,a2,…an. The server provides two operations for its clients:

read(i) returns the value of ai;

write(i,Value) assigns Value to ai.

Transactions T and U are as follows:

T: x = read(j); y= read(i); write(j,44); write(i:33);

U: x = read(k); write(i,55); y=read(j); write(k,66).

How many interleavings of transactions T and U are serially equivalent? Give 3 serially equivalent interleavings.

1. Do exercise 16.18 on page 725 of your text.
2. Do Exercise 16.19 on page 725 of your text.

## A Level 20 points—Choose One

1. Compare and contrast *the approach* used by Dynamo storage service with a traditional transaction-based approach. Be sure to explicitly state your criteria for comparison. Your textbook is a starting point, but cannot be one of your three sources. How can we claim that a weaker form of consistency is acceptable?
2. Discuss two or three algorithms that allow asynchronous systems to reach agreement even though agreement cannot always be guaranteed. Is there a possible next step toward a guarantee?
3. What conditions are necessary and sufficient to implement distributed transactions on shared items without Locks? If any exist today, briefly describe how it handles transactional concurrency and what its advantages/disadvantages are.

F: < 60 D: < 70 C: < 80 B: < 90 A: >= 90