EXAMINATION

Course: **5DV020/Distributed Systems**

Teacher in charge: Yvonne Löwstedt/Lars Larsson

Semester: HT-09
Date: 2010-01-08
Time: 09.00–14.00

Name: ____

Personal ID number: ______________________________

Unique code for this examination: **1**

Note!

This examination will be graded anonymously. This sheet will be removed before the teacher receives the rest of the examination. The above code must therefore be on all other pages when you submit the examination to the examination supervisory staff. **Memorize** your code since it will be used as reference when the results are published.
Furthermore,

- Write the answers on the answers on the same paper as the question (the back of the paper may also be used).
- Mark the questions you have solved with a cross on the next page.
- The solutions should be neatly written. The train of thought should be easy to follow. All non-obvious assumptions must be explicitly stated.

Till skrivningsbevakaren: Avskilj detta försättsblad och stoppa i kuvert som skickas till Yvonne Löwstedt, Datavetenskap.

EXAMINATION

Course: **5DV020/Distributed Systems**
Teacher in charge: Yvonne Löwstedt/Lars Larsson

Semester: HT-09
Date: 2010-01-08
Time: 09.00–14.00

Unique code for this examination: **1**

Problem	**Solved**	**Points**	**Max points**
1			7
2			4
3			4
4			8
5			8
6			6
7			7
8			6
9			7
11			3
Sum			60
Grade			

Question 1 (1 + 3 + 3 points)

Message ordering is an important topic in distributed systems, since it gives us power to reason about the messages that are sent and delivered in the system, and prove correctness and applicability of algorithms. The course book covers two algorithms for implementing total ordering of messages.

a. Give a concise but accurate description of total ordering for messages delivered to a group of processes in a distributed system.

b. Draw a figure and explain how the *sequencer* algorithm for implementing total ordering works.

c. Draw a figure and explain how the *voting* algorithm (used in the ISIS system) for implementing total ordering works.

We figure that figures may take up a lot of space, figuratively speaking.

Question 2 (4 points)

Assume that you have a system with the following properties:

a. total message ordering, implemented with the algorithm of your choice;

b. reliable multicast implemented using the naive algorithm from the book (repeated basic multicast).

On top of this system, you need to implement an algorithm that sends (in total) five (5) **data** messages per iteration.

Briefly analyze one such iteration from a complexity point of view in $\mathcal{O}(n)$ notation, where n is the number of processes in the system **and** roughly estimate the number of messages sent in this system with four (4) processes. Also, briefly reflect on the suitability of choosing this approach (it should be quite obvious!).

You might want to use this page too.

Question 3 (4 points)

Let T and U be transactions defined as follows:
T:

- T_1 x=read(i);
- T_2 write(i, x+30);
- T_3 write(j, 20);

U:

- U_1 write(i, 100);
- U_2 write(j, -50);

Which of the following interleavings are serially equivalent and which can happen if two-phase locking is used?

a. U_1, T_1, T_2, T_3, U_2

b. T_1, U_1, T_2, T_3, U_2

c. T_1, T_2, U_1, T_3, U_2

d. T_1, T_2, T_3, U_1, U_2

Question 4 (2 + 2 + 2 + 2 points)

Logical time uses "event counters" rather than physical time to order events. Recall that events may be either internal events (i.e. state changes), send events, or receive events.

a. Make a figure with 3 processes that send a few (minimum 5, and each process has to send at least once) messages back and forth and have some internal events. Write the Lamport logical clock values for each event. Include concurrent events!

b. Make a figure with 3 processes that send a few messages (same restriction as above) back and forth and have some internal events. Write the vector clock values for each event. Include concurrent events!

c. From a theoretical point of view, values from Lamport logical clocks provide us with less reasoning power than values from vector clocks, with regard to which events took place before which in real time. Use your figures to show this!

d. Lamport logical clocks are not without use, though! Show rules for comparing Lamport logical clocks that have been extended with totally ordered **process identifiers** that make it possible to get a total ordering of events that occur at the processes!

Question 5 (2 + 2 + 4 points)

Chandy and Lamport invented the Snapshot algorithm. It is used to determine the global state of a distributed system in a distributed manner using message passing.

a. What assumptions does the algorithm make?

b. What two purposes do marker messages serve?

c. Roughly explain how the algorithm works (using an example, if you wish)!

How did this empty page get here? Oh well, you could use it to provide a more lengthy answer to the question, now that it is here...

Question 6 (2 + 2 + 2 points)

One thing we've talked a lot about during the course is *Security policies.* But what was all that stuff about anyway? Feel free to enlighten us!

a. What is a security policy?

b. Mention different kinds of security policies, and very briefly explain the differences.

c. How are these security policies related to the big security picture?

You may need at least one more page for that last one!

Question 7 (2 + 0.5 × 6 + 2 points)

Security is often divided into multi-lateral and multi-level security. **For the first two points**, state the difference and reason about how they complement each other here:

Assume the following order of security levels:

a. Top secret (TS)

b. Secret (S)

c. Confidential (C)

where TS is the highest level. Also assume that we have the following categories:

- Research and Development (RD)
- Economics (E)
- Human Resources (HR)

Subject	**(TS)**	**(S)**	**(C)**	**RD**	**E**	**HR**
John Doe		X		X		
Richard Roe		X			X	X
Terry Tate	X					X
Joe Bloggs		X		X	X	X

Object	**(TS)**	**(S)**	**(C)**	**RD**	**E**	**HR**
Lab staff wages		X			X	X
Drug test results	X					X
Nifty new weapon systems	X			X		
Staff list			X	X	X	X
Lab inventory			X	X		

Determine whether the following six statements are true of false based on the matrix on the previous page.

a. Terry Tate *dominates* Drug test results

b. Richard Roe *dominates* Staff list

c. John Doe *dominates* Lab inventory

d. Terry Tate *dominates* Lab staff wages

e. Joe Bloggs *dominates* Staff list

f. John Doe *dominates* Nifty new weapon systems

For the final two points: in this question, you have access to the information on security levels, categories, and classification. What is the security classification for this information? Motivate your answer!

Question 8 (1 + 2 + 3 points)

Election algorithms are very useful in distributed systems, since many algorithms require that a certain process is “special” in some way (coordinator, master, etc.).

a. What are the two requirements we expect any reasonable election algorithm to fulfill?

b. Explain how the ring-based election algorithm works!

c. How does the ring-based election algorithm handle crash failures? Does it still, under some circumstances, fulfill the requirements we asked about earlier?

Question 9 (2 + 1 + 1 + 3 points)

Deadlocks ("I wait for you, while you wait for me — neither of us can continue") are an unfortunate reality for the locking concurrency control scheme unless we cautiously avoid them. If our transactions are distributed, detecting deadlock gets even trickier.

a. Deadlocks can either be detected, or prevented from happening in the first place. A primitive way of prevention is to lock all resources that the transaction will require. Why is this a bad or even (for some systems) impossible way of handling the issue?

b. What is a phantom deadlock?

c. What are vulnerable locks? How do they work?

d. Explain the rough steps of the edge chasing algorithm for detecting deadlock in a distributed system. You do not have to explain about the modifications and optimizations we talked about during the lectures, just the pure version.

Question 10 (-3 to 3 points)

The following questions require only a true or false answer. Correct answers give 0.5 points, whereas incorrect answers are penalized with -0.5 points. Note that the total from the question may be negative, and this will impact your final score. No answer is the safest option, and counts as 0 points. Any text besides "true" or "false" will not be taken into consideration.

The computational cost to break encryption is exponential to the key length.	
Cristian's method for synchronizing clocks requires a well-performing and stable network connection.	
SSL 3.0 is vulnerable to man-in-the-middle attacks.	
Active replication is used to gain fault tolerance against arbitrary failures.	
There is no guaranteed solution to the byzantine generals problem in an asynchronous system.	
When a participant process in the two-phase commit protocol is prepared to commit, it has already written its changes to permanent storage.	