UMEÅ UNIVERSITET
Institutionen för Datavetenskap

UMEÅ UNIVERISTY
Dept of Computing Science
Unique code: 1

EXAMINATION

Course: **5DV020/Distributed Systems**

Teacher in charge: Yvonne Löwstedt/Daniel Henriksson

Semester: HT-09
Date: 2009-04-07
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/Daniel Henriksson

Semester: HT-09
Date: 2009-04-07
Time: 09.00–14.00

Unique code for this examination: **1**

Problem	**Solved**	**Points**
1		
2		
3		
4		
5		
6		
7		
8		
9		
Sum		
Grade		

Question 1 (3x1 + 3x1 + 3x1 points)

As you are well aware, many algorithms in distributed systems require that messages are ordered according to some specific scheme. However, *which* scheme is the required one, and how it will affect system performance, is an interesting topic of study. In particular when we couple message ordering algorithms with reliable multicasting.

For each of the following message ordering schemes:

- FIFO
- Causal
- Total

please answer the following questions:

a. what is the definition of the scheme (1 point per scheme)?

b. state a system where the scheme should be used due to its particular ordering properties, and motivate your response clearly (1 point per scheme)?

c. given your choice of *system* in the previous point, evaluate its performance characteristics when combined with naive reliable multicast, i.e. the one you implemented for the assignment? Is it still suitable? Why/why not? (1 point per scheme)

This question is worth 9 whole points, surely you want to use this page too to make sure you get as many as possible!

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

Clock synchronization in a distributed system is *hard.* There are many different approaches to this problem, e.g. the Berkeley algorithm where there's a master node that internally synchronizes a group by sending out offsets to each client, or the Internet-scale NTP system. However, the in many ways simplest algorithm is Cristian's method. This is the one we shall devote our attention to this time.

a. Please describe what messages are passed (and who the actors are) in Cristian's method using a simple illustration.

b. Describe the requirements and assumptions Cristian's method makes on the environment it is deployed in, e.g. on the network.

c. Show the math involved! Show how synchronization is acheived and pay particular attention to showing the bounds on the accuracy of the synchronization.

d. Discuss Cristian's method from a security and fault-tolerance point of view!

You might want to write on this page too...

Question 3 (2 + 2 + 2 points)

The Byzantine Generals Problem is well-studied and very useful for discussing consensus, agreement, and fault-tolerance. As you know, there is a Commander and a number of Lieutenants who need to coordinate whether they should simultaneously attack or retreat by means of message passing/forwarding. Some of these generals may be treacherous, and intentionally attempt to give wrong or conflicting information to the others.

a. Illustrate why the Byzantine Generals Problem can be solved for $N > 3f$ processes, where f is the number of failing processes.

b. Illustrate why the problem is impossible with $N \leq 3f$ processes.

c. While interesting in theory, the problem can be solved quite easily if more attention is paid to rigorous security (rather than just blind trust among the generals). Discuss why this is the case, and provide a solution!

Illustrations can take up quite some space, have another page, why don't you?

Question 4 (2 + 2 + 4 points)

Optimism is according to Merriam-Webster's Online Dictionary "an inclination to put the most favorable construction upon actions and events or to anticipate the best possible outcome"[1]. Based on the practical assignment, would you say that such an approach works for distributed systems? Really?

Let's discuss optimistic concurrency control, that presumably works really well in the land of unicorns and chocolate[2] (and in reality too, actually, just look at MediaWiki that powers Wikipedia).

a. Draw an illustration to show *forward validation* of transactions. Explain (using e.g. pseudo code) how the validation scheme works.

b. Draw an illustration to show *backward validation* of transactions. Explain (using e.g. pseudo code) how the validation scheme works.

c. Compare the two schemes:

- How may we determine which transaction to abort?
- Based on our choice of transaction to abort, is either scheme more susceptible to transaction *starvation*?
- Which information is saved in either scheme, and for how does it have to be kept?
- If processing a validation phase takes very long time, which scheme is better? Why?

[1]`http://www.merriam-webster.com/dictionary/optimism`, fetched March 26, 2010
[2]Or chocolate unicorns. Mmmh.

Perhaps you need more space?

Question 5 (4 + 4 + 2 points)

Replication can give us increased speed as well as improved fault-tolerance, so it is a Good Thing. However, one size does not fit all, and different replication schemes exists. Choosing the right one for the job is a task for people such as yourself, so let's get to it!

a. Draw a figure and explain how *passive replication* works, and discuss its properties.

b. Draw a figure and explain how *active replication* works, and discuss its properties.

c. The general replication phases are *Request*, *Coordination*, *Execution*, *Agreement*, and *Response*. In the schemes you have just described are there any optimizations you can make in either of these phases for *read-only* operations? If so, state them!

Hint: note that you can get up to 4 whole points for your explanations of how the schemes work. Really show why you deserve them!

Hint 2: we can't really give any better hint than listing the phases in the third point, can we?

This page is intentionally left blank-ish.

Question 6 (3x2 points)

Let us assume that you are to be hired as a security consultant for a medium-sized bank, working on a project basis to secure their current system. During the interview they ask the questions listed below, and we are very interested in seeing what you respond!

Please provide reasonably short answers (maybe a couple of sentences per question) for the following general security questions:

a. What do you percieve as the largest threat(s) to this kind of system?

b. What's your take on security policies? We feel that our current policy, "Don't do stupid stuff" covers most of the issues.

c. What would your strategy be in order to make the system completely safe?

Perhaps you need another page?

Question 7 (2 points)

Here is a quick one for you:

a. What constitutes a *computationally secure* cryptographic system?

Question 8 (8 points)

There are several tools in the security toolbox. Briefly describe the following tools, including their strengths and weaknesses, and also describe *when* these tools are used.

a. Symmetric encryption algorithms.

b. One-way hash functions.

c. Asymmectric encryption algorithms.

d. Digital signatures.

Question 9 (-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.

For transactions, read and read operation for the same object are *never* in conflict	
Kerberos authorization is a suitable solution for sites with many servers	
Deadlocks can *never* occur in optimistic concurrence control	
In nested transactions, if a subtransaction aborts, the parent transaction must also abort	
The *inconsistent retrievals problem* can be solved using SSL encryption of the network channel	
Correctness of a clock implies that it is correctly synchronized to UTC within a certain margin of error	

"Success is not the key to happiness. Happiness is the key to success. If you love what you are doing, you will be successful." — Herman Cain, American business man, author and speaker