UMEÅ UNIVERSITET
Institutionen för Datavetenskap

UMEÅ UNIVERISTY
Dept of Computing Science

EXAMINATION

Course: **5DV020/Distributed Systems**

Teacher in charge: Yvonne Löwstedt/Lars Larsson

Semester: HT-08
Date: 2008-11-03
Time: 09.00–15.00

Name: ____

Personal ID number: ________________________________

Unique code for this examination: **20**

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 your code and the question number in the **top right corner** of every paper.
- 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.

UMEÅ UNIVERSITET
Institutionen för Datavetenskap

UMEÅ UNIVERISTY
Dept of Computing Science

EXAMINATION

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

Semester: HT-08
Date: 2008-11-03
Time: 09.00–15.00

Unique code for this examination: **20**

Problem	**Solved**	**Points**
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
Sum		
Grade		

Question 1 (3 + 1 + 1 points)

In a distributed system such as GCom that deals with groups, the group membership service is of great importance.

a. Mention at least three of the main responsibilities of a group management service that handles dynamic groups?

b. Group membership services may be either *primary-partition* or *partitionable.* Describe the difference between these two types.

c. Without going into too much detail, give a brief description of what a *view-synchronous* group membership service offers.

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

The Network Time Protocol (NTP) is used for synchronization to an external time source. Servers are organized in a heirarchical system of *strata* (levels). Servers closer to the time source are have the most reliable time values. As the time values are transferred in the system, errors are introduced for each new stratum.

a. NTP was developed to handle several problems, and has done so with great success (all modern mainstream operating systems have some NTP service running by default). Which were these problems?

b. Show using a figure how two NTP peers communicate by message passing in procedure-call and symmetric mode (hint: they work the same).

c. Show how the NTP peers above calculate the *offset* and the *delay* values, and state what these values are used for.

d. Many services depend on NTP providing correct time. Thus an intruder supplying erroneous time information could in theory be devastating to many systems. Reason about how computers can validate the identity of an NTP peer to solve this security-related issue.

Question 3 (2 + 2 + 2 points)

Failure handling in distributed systems is hard. Fischer *et al.* (1985) showed that asynchronous systems cannot reach consensus, even with just a single crash failure. Synchronous systems are slightly easier from that point of view, but even those have problems. One such problem is the Byzantine generals problem, where arbitrary failures must be taken into account.

a. What makes synchronous systems generally easier to deal with than asynchronous from a failure-handling point of view?

b. Briefly describe the Byzantine generals problem.

c. Show, preferrably with two figures, why the Byzantine generals problem cannot be solved for $N \leq 3f$ processes (where N is the amount of processes and f is the amount of failing processes). Hint: show why 3 processes cannot handle 1 process exibiting arbitrary failures, but why 4 processes can.

Question 4 (2 + 4 points)

Transactions is a very important concept in distributed systems — in particular ones that deal with databases of various kinds. The idea is to define not mere operations, but operation "groups" that can be applied to a distributed object. In that context, ACID is an acronym that is well-known and used in distributed systems as well as in databases. It describes properties that one requires from a transaction-aware system for the system to be useful.

a. What does ACID stand for? Name the four properties that make up the acronym, and give a brief description of them in the context of distributed systems.

b. There are some problems related to distributed objects that can be accessed using transactions. Three classical problems are the *dirty read*, *lost update*, and *inconsistent retrievals* problems. Describe **two** of these (you may choose which two of the three), and give simple and clear examples that illustrate what goes wrong.

Question 5 (4 + 1 + 1 points)

Data replication is a very interesting and quite profitable area of distributed systems. Replicating data may boost performance and may also be used for masking failures. There are two common schemes of data replication: *active* and *passive*.

a. Describe both the active and the passive replication scheme, including the difference between them.

b. Briefly compare the two schemes from a performance point of view.

c. Which replication scheme is more appropriate for a database system containing highly sensitive data such as medical records, where we must assume that arbitrary failures may occur? Motivate your answer.

Question 6 (2 + 3 + 2 points)

In distributed systems that use locks to ensure correctness in spite of concurrency, the biggest problem is the dreaded *deadlock*. Deadlocks arise e.g. when a process A waits for another process B, and B waits for A at the same time. The processes may not wait directly for each other, in fact, as long as there is a cycle in a *wait-for graph* of processes, there is a deadlock.

a. Deadlocks can either be prevented or detected (and then handled). How do these approaches work, and what are their properties? Compare the approaches from a performance point of view.

b. Centralized deadlock detection is not good. Describe the three steps of the distributed algorithm for *edge chasing* instead.

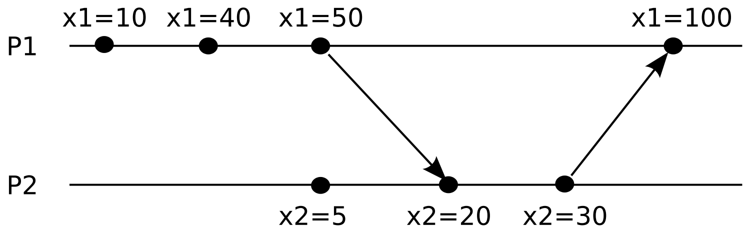
c. Briefly explain the concepts *vulnerable locks* and *starvation* that are both related to concurrency control using locks.

Question 7 (4 points)

Identify and explain (motivate) a situation where Web Services would be a better choice for distributed computing than Java RMI.

Question 8 (1 + 3 points)

Distributed debugging involves evaluating whether some global state predicate ϕ was definitely true at some point in the execution, and cases when it was possibly true.



a. Using your own terms, clearly explain *possibly* ϕ and *definitely* ϕ and the difference between them.

b. For the figure above, construct the lattice of global states and give the truth value for *possibly* ϕ, where $\phi = x_1 \geq x_2$.

Question 9 (4 + 2 points)

Proper access control is important to obtain a high level of security in a system. Two ways of implementing access control are *access control lists* and *capabilities*.

a. Explain each concept, how they are applied in practice, and their differences.

b. Discuss pros and cons with each approach.

Question 10 (5 × 1 points)

Understanding the terminology used within a subject can sometimes (as within computer security) be fundamental. Not only does the proper use of terminology make someone sound trustworthy and eductated, it is also central in understanding new material concerning the subject, and relate this to earlier experiences.
For each pair of security related terms below:

- briefly explain *both* terms and;
- explicitly state if and how they relate to each other.

The terms are:

a. TLS and RSA

b. Integrity policy and Chineese Wall Model

c. Cipher Block Chaining and PKI

d. Simple Security Condition and *-property

e. Certification Authority and Kerberos

Question 11 (-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.

Sequential consistency implies linearizability	
For logical clocks, $L(e) < L(e') \Rightarrow e \rightarrow e'$	
Phantom deadlocks cannot occur using two-phase locks	
Causal-total ordering is the same as total ordering	
Starvation of transactions cannot happen if forward validation is used	
Selecting a new leader process in a set of processes by choosing it from a sorted list of processes, but without coordination between the processes, is a correct election algorithm	