UMEÅ UNIVERSITET
Institutionen för Datavetenskap

UMEÅ UNIVERISTY
Dept of Computing Science
Unique code: 20

EXAMINATION

Course: **5DV020/Distributed Systems**

Teacher in charge: Yvonne Löwstedt/Lars Larsson

Semester: HT-09
Date: 2009-11-02
Time: 09.00–14.00

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 University, 901 87 Umeå. Ph: 090 786 6131. Fax: 090 786 6126. Email:

EXAMINATION

Course: **5DV020/Distributed Systems**

Teacher in charge: Yvonne Löwstedt/Lars Larsson

Semester: HT-09

Date: 2009-11-02

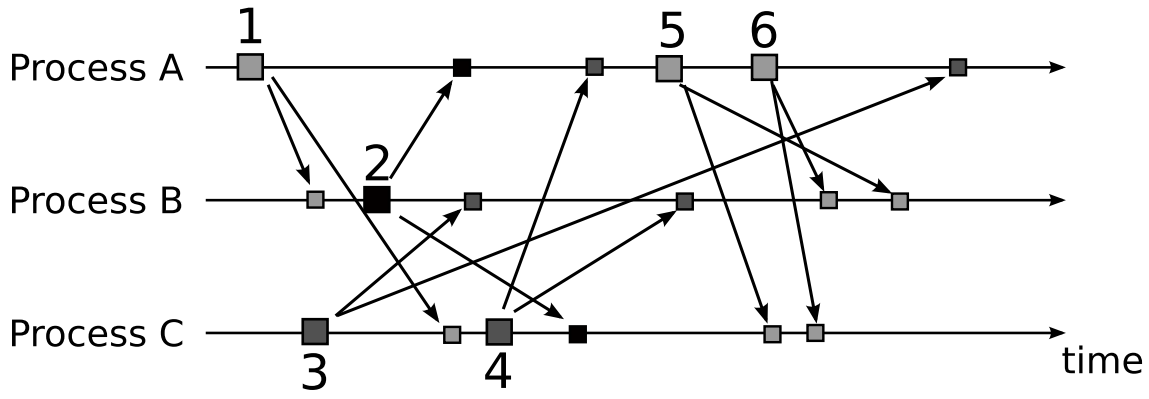
Time: 09.00–14.00

Unique code for this examination: **20**

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

Question 1 (8x1 + 2 points)

Given the sequence of messages shown:



a. State clearly the order in which the messages are delivered **in each node** using the message orderings shown below. For Total ordering, assume that node A is the sequencer and that sequence messages are transferred **instantly** to both B and C. Note that you should write the exact order of messages as they are delivered at each node, and providing *any* valid sequence if messages is **not** sufficient. The arrows in the graph indicate message transfer and reception.

(i) FIFO
(ii) Causal
(iii) Total
(iv) Causal-Total
(v) Total-Causal
(vi) FIFO-Total
(vii) Causal-FIFO
(viii) Unordered-FIFO-Causal-Total

b. Analyze your answers and motivate why some answers are similar.

Note that you have an answering matrix for *a* on the following page.

Answers for *a*.

	Process A	Process B	Process C
i			
ii			
iii			
iv			
v			
vi			
vii			
viii			

Answer for *b*:

Question 2 (1 + 2 + 3 points)

The people at CERN are having problems with the Large Hadron Collider (again!). This time, it is related to time synchronization issues between their machines controlling the collider. They need a way to solve the problem, and they need it *now*. Obviously, they have a UTC time source on site, which should be used to synchronize their equipment.

a. The technicians have heard of a protocol called NTP, but are a bit wary of using it, since they know that it is used to synchronize all kinds of consumer products such as their laptops as well. Set their mind at ease by giving a rough sketch of how NTP networks are constructed!

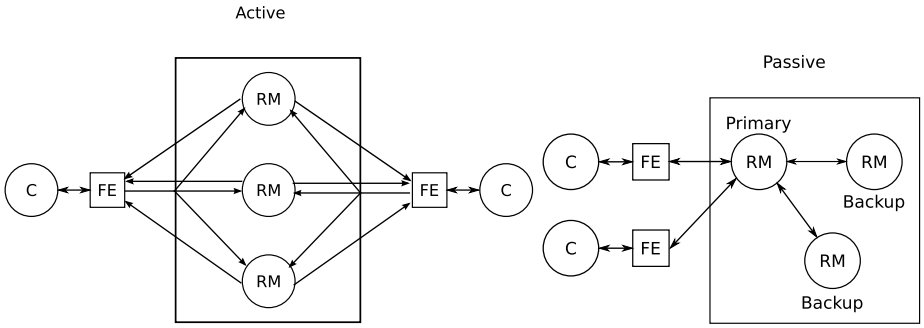
b. Given that there is a UTC time source on-site, what NTP mode would be used? How does it work, on the level of messages being sent and what they contain? Note: you do not have to show how the messages are mathematically used to calculate offsets and delays.

c. Correct and reliable time data is of utmost importance in this — what security measures would you recommend, and for what reason? Motivate your recommendations clearly.

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

Question 3 (2 + 2 + 2 points)

These two pictures show the flow of communication under the active and passive replication schemes:



a. Which scheme would you recommend for an application that is truly time-critical? Why?

b. Which scheme would you recommend for an application that needs to remain useful even if arbitrary errors may occur? Why?

c. If you optimize these schemes by issuing read requests to either just one replica manager (in active replication) or to a backup replica manager (in passive replication), do you lose any of these properties? Why?

Question 4 (1 + 2 + 3 points)

Transactions require concurrency control. During the course, we have studied pessimistic and optimistic concurrency control.

a. Give an example of pessimistic concurrency control!

b. Why are the types of concurrency control called optimistic and pessimistic? Describe the scenarios when each approch is most beneficial!

c. In optimistic concurrency control, once a transaction is complete, it gets validated against other active transactions to investigate whether it can be committed or must be aborted. Use a figure to explain how forward- and backward validation of transactions work.

Write on this page too, we don't like trees anyway, stading there all smug, who do they think they are?...

Question 5 (3 + 5x1 points)

Figure 1 below shows events that happend in a system with two different processes, P_A and P_B. Arrows indicate a transfer of messages. Figures 2 to 4 are all copies of Figure 1.

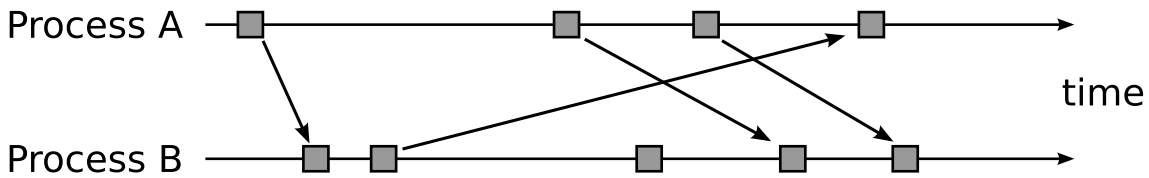


Figure 1.

a. Draw a diagram that shows all consistent states of the system, including the transfers between those states. Begin in state (0,0), before anything happend in the system. Clearly indicate which events that has happened in each of the processes in every consistent state. From the lectures and the book, you might recall that the lattice (crystalline structure) is very suitable for this.

b. Using Figure 2 below, show an example of a consistent cut and an example of an inconsistent cut. Also describe how to tell the difference between a consistent cut and an inconsistent one.

c. Add simple logical (Lamport) clocks to Figure 3.

d. Add Vector clocks to Figure 4.

e. Give an example of a *linearization* for all the events in the system. Use the following notation: e_{xy}, where x is the process identity and y is the event number.

f. Give an example of a *run* that is not also a *linearization*, for all the events in the system. Use the same notation as in question e.

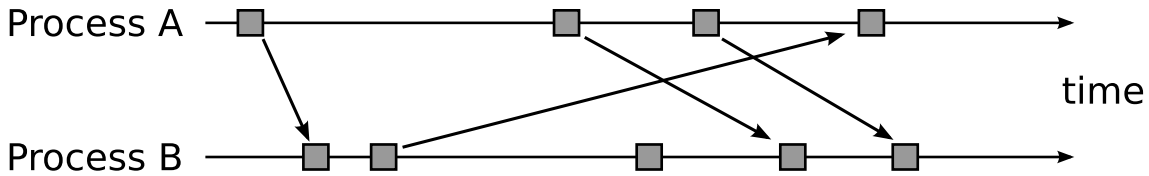


Figure 2.

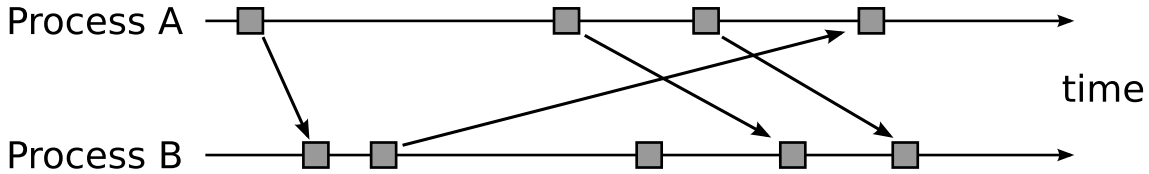


Figure 3.

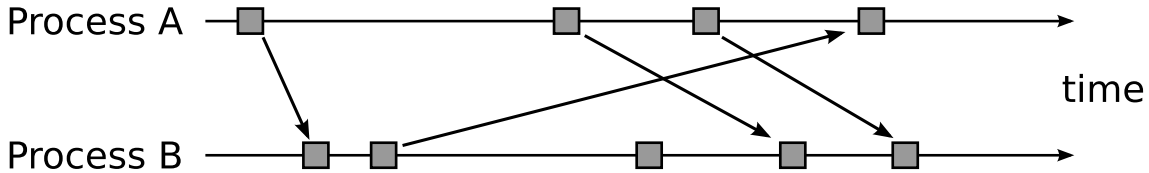


Figure 4.

This page is intentionally left blank-ish.

Question 6 (1 + 2 + 3 points)

In real-world systems, errors occur. Let's talk errors and failure handling!

a. What is the difference between unreliable and reliable failure detectors?

b. Why are synchronous systems easier in some regards to work with than asynchronous ones, from a failure handling point of view?

c. List and briefly explain three strategies for masking errors in asynchronous distributed systems, where crashes may occur!

Don't stop writing now!

Question 7 (3+4+2 points)

One of the most commonly used security mechanisms is cryptography, and the problems with cryptography are often associated with key management.

For the questions below, explain the basics of the algorithms / procedures, preferrably using images. Note that you can get high points on this questions even if you cannot recall the exact contents of each message, as long as you give a clear description of the overall algorithm.

a. Needham-Shroeder (Algorithm for exchange of established keys)

b. Kerberos

c. Diffie-Hellman (Message exchange that require no previously shared secrets / keys)

Question 8 (2x2 + 2 points)

Briefly explain the following:

a. Biba security model

b. Bell-LaPadula security model

Would any of these models be suitable to use at a research facility? Clearly motivate your answer!

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.

Once a transaction has been committed using two-phase commit, it cannot be rolled back.	
In a distributed system, it is impossible to tell a faulty process from a slow one.	
Data encoded with one-time pads cannot be cryptoanalyzed (broken).	
If a subtransaction aborts, the parent transaction must abort also.	
The Byzantine Generals problem cannot be solved if two processes are faulty.	
Client-server solutions are not distributed systems.	