

**Tentamen i kursen**  
**Distribuerade System- TDDD25**  
**2013-08-21, kl. 8-12**

**Hjälpmedel:**

Engelsk ordbok.

**Supporting material:**

English dictionary.

**Poänggränser:**

Maximal poäng är 40.  
För godkänt krävs sammanlagt  
21 poäng.

**Points:**

Maximum points: 40.  
In order to pass the exam you need a  
total of minimum 21 points.

**Jourhavande lärare:**

Petru Eles, tel. 0703681396

**Good luck !!!**

1. Synchronous and asynchronous distributed systems. What are their main features and what are the consequences of these features?

(3p)

2. What means transparency in a distributed system? We have defined seven aspects of transparency. Enumerate and explain at least five of them.

(3p)

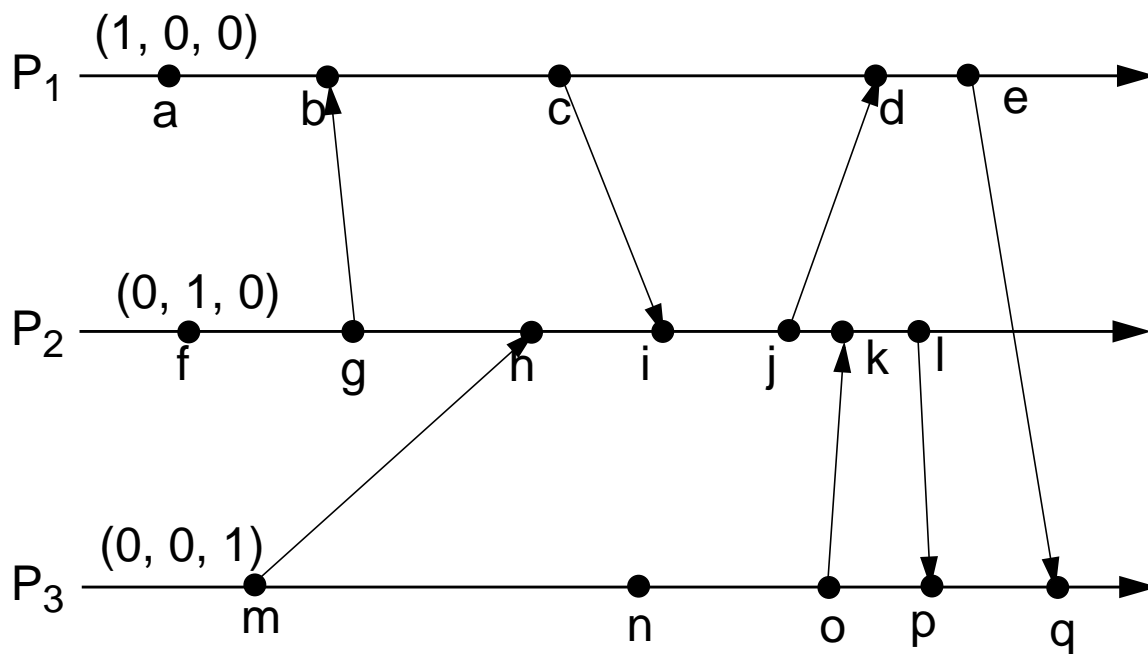
3. What is an Interface Definition Language. What is its function in the context of Middleware.

(2p)

4. What are potential problems with client-server systems?  
How are they solved with peer-to-peer systems?  
What are key issues and problems with peer-to-peer systems?

(2p)

5. Consider the following set of events:



Assign the missing vector clock values to the events.

(3p)

6. Consider a system of four processes  $P_1, P_2, P_3, P_4$ . Consider the events  $a$  in  $P_1$ ,  $b$  in  $P_2$ ,  $c$  in  $P_3$ , and  $d$  in  $P_4$ .
- a) Let us consider a case such that the Lamport's logical clock timestamps associated to the events are the following:  
 $C(a) = 1; C(b) = 3; C(c) = 2; C(d) = 2$ ;  
What can you say regarding the happened before relation between events  $a, b, c, d$  (consider each pair of events)?
- b) Let us consider a case such that the vector clock timestamps associated to the events are the following:  
 $C^V(a) = (2,0,0,1); C^V(b) = (2,3,1,1); C^V(c) = (3,2,2,1); C^V(d) = (2,3,1,2)$ ;  
What can you say regarding the happened before relation between events  $a, b, c, d$  (consider each pair of events)?

(3p)

7. Explain the following types of redundancy:
- Time redundancy
  - Hardware redundancy
  - Software redundancy
  - Information redundancy

(3p)

8. What is the basic idea behind the token based distributed mutual exclusion algorithm by Ricart-Agrawala (the second algorithm)? Consider how mutual exclusion is guaranteed and how the token is passed after a process has left the critical section. How many messages are passed in order a process to get permission to a critical section? Compare to the first algorithm by Ricart-Agrawala (which is not using a token).

(3p)

9. Remote Method Invocation: trace the way of a request and of the reply from the client to a remote server and back. Illustrate with a figure.

(3p)

**Tentamen i kursen Distribuerade System -TDDD25, 2013-08-21 kl. 8-12**  
**Du kan skriva på svenska eller engelska!**

10.

- a. Define total and causal ordering of requests. Illustrate by an example.
- b. How can total ordering be implemented using a central sequencer?
- c. Consider total ordering based on distributed agreement (no central sequencer); consider one front end and several replica managers.

In this case, the replica managers, after receiving a request, send back to the front end a *cuid*. What does the front end send back to the replica managers after receiving the *cuid* from each replica manager? How does the front end calculate the value it sends back?

- d. What happens if a replica manager crashes before sending to the front end the *cuid* for a request it received?

(4p)

11. Consider a bully election with 6 processes,  $P_1, \dots, P_6$ .  $P_6$ , the current coordinator, fails and  $P_3$  starts the election. Illustrate the sequence of messages exchanged (use figures).

(3p)

12. What is the basic idea with voting protocols for updating replicated data? How do they work? Consider a set of 12 replica managers. Define two voting protocols. One for a situation when the number of writes is relatively large compared to that of reads, and the other for the reverse situation. Give examples of read and write quorums (use figures).

(3p)

13. What does it mean by external and internal synchronization of physical clocks?  
What does it mean by centralised and distributed synchronisation algorithms?

(2p)

14. For clock synchronisation with the Precision Time Protocol the communication delays on the way master to slave and slave to master have to be considered. The calculations for clock synchronisation assume that the delays in both directions are equal. This, however, should not be necessarily true, in general. How is this particular problem solved? Explain and illustrate your explanation with a figure?

(3p)