Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2010-03-10
Sal	KÅRA, T2
Tid	8-12
Kurskod	TDD25
Provkod	
Kursnamn/benämning	Distribuerade system
Institution	*IDA*
Antal uppgifter som ingår i tentamen	14
Antal sidor på tentamen (inkl. försättsbladet)	5
Jour/Kursansvarig	Jakob Rosen
Telefon under skrivtid	284046, 0768244344
Besöker salen ca kl.	10
Kursadministratör (namn + tfnnr + mailadress)	Gunilla Mellheden, 282297, gunme@ida.liu.se
Tillåtna hjälpmedel	Ordbok
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	
Vilken typ av papper ska användas, rutigt eller linjerat	
Antal exemplar i påsen	

LINKÖPINGS TEKNISKA HÖGSKOLA
Institutionen för datavetenskap
Petru Eles

Tentamen i kursen

Distribuerade System- TDDD25/TDDB37

2010-03-10, 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:

Jakob Rosén, tel. 284046, 0768244344

Good luck !!!

Tentamen i kursen Distribuerade System -TDDD25/TDDB37, 2010-03-10, kl. 8-12
Du kan skriva på svenska eller engelska!

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. Define the following three possible semantics for remote procedure calls:
   a. At least once semantics
   b. At most once semantics
   c. Exactly once semantics.

   Is it possible to achieve *exactly once semantics* in the case of lost messages? But in the case of server crashes? Explain.

(3p)

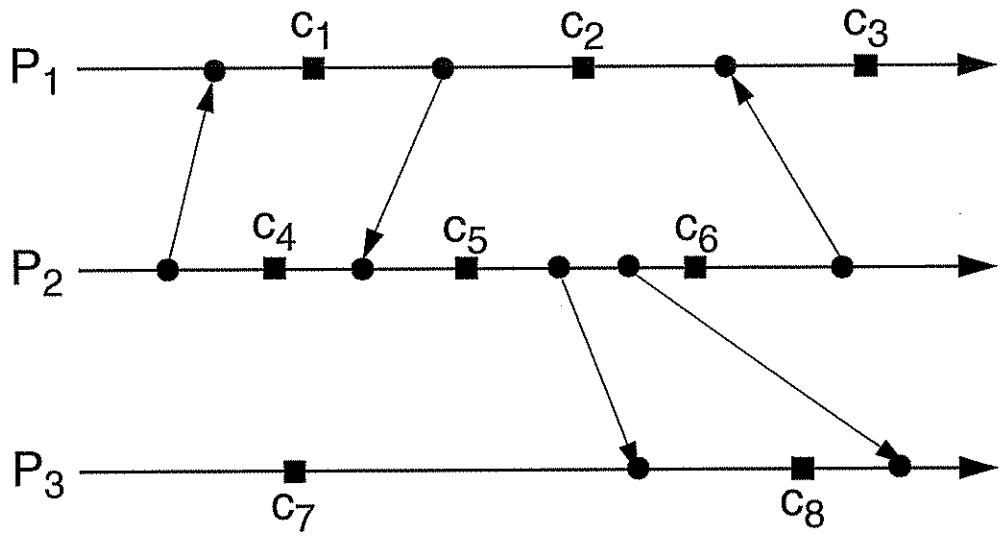
4. What is an Interface Definition Language. What is its function in the context of Middleware.

(2p)

5. 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)

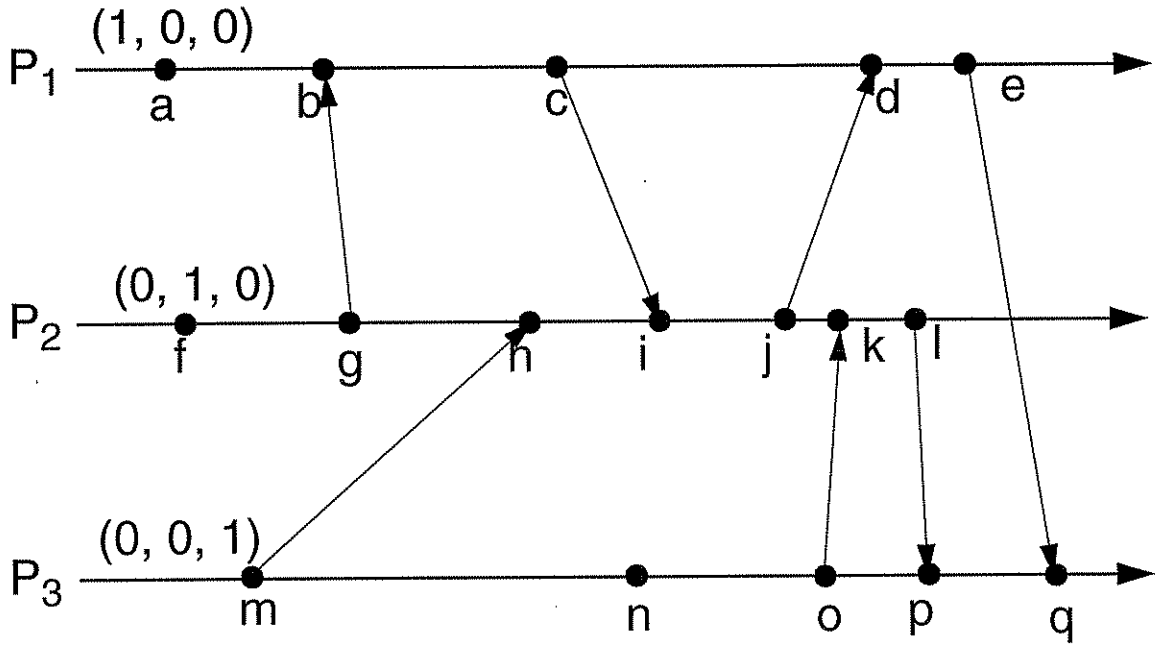
6. What is a cut of a distributed computation? What means a consistent and a strongly consistent cut? Consider the following set of events:



Determine for each of the following cuts if it inconsistent, consistent or strongly consistent: $\{c_2, c_6, c_8\}$, $\{c_1, c_4, c_7\}$, $\{c_1, c_5, c_7\}$, $\{c_1, c_6, c_8\}$, $\{c_1, c_6, c_7\}$, $\{c_3, c_6, c_8\}$, $\{c_2, c_5, c_8\}$.

(3p)

7. Consider the following set of events:



Assign the missing vector clock values to the events.

(3p)

8. 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)

9. 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)

Tentamen i kursen Distribuerade System -TDDD25/TDDB37, 2010-03-10, 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 mangers, 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. Explain the following types of redundancy:
    - Time redundancy
    - Hardware redundancy
    - Software redundancy
    - Information redundancy

(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. You know the maximum drift rate of the clocks on two processors and the maximal allowed skew between them. How do you determine the maximum interval between two successive synchronizations between the clocks?

(2p)

14. Cristian's algorithm for clock synchronization. Describe how it works. How does it estimate the time at the receiver? What is the accuracy of this estimation?

(3p)