Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	120308
Sal	TER4
Tid	14-18
Kurskod	TDDD25
Provkod	
Kursnamn/benämning	Distribuerade system14
Institution	*IDA*
Antal uppgifter som ingår i tentamen	14
Antal sidor på tentamen (inkl. försättsbladet)	5
Jour/Kursansvarig	Petru Eles
Telefon under skrivtid	0703681396, 281396
Besöker salen ca kl.	16
Kursadministratör (namn + tfnnr + mailadress)	Gunilla Mellheden, 282297, gunilla.mellheden@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

2012-03-08, kl. 14-18

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 !!!

Tentamen i kursen Distribuerade System -TDDD25, 2012-03-08, kl. 14-18
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. 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

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

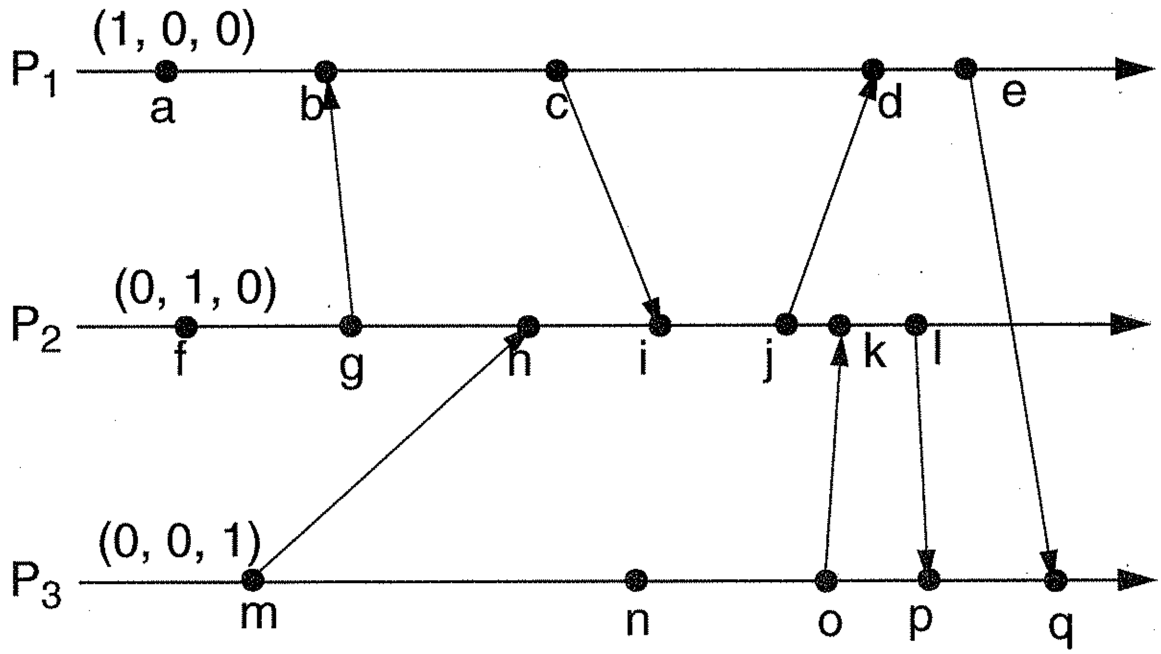
(2p)

4. 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)

Tentamen i kursen Distribuerade System -TDDD25, 2012-03-08, kl. 14-18
Du kan skriva på svenska eller engelska!

5. Consider the following set of events:



Assign the missing vector clock values to the events.

(3p)

6. 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?

(3p)

7. 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)

8. Consider mutual exclusion with the Ricart-Agrawala algorithm (the first algorithm, not using a token). Imagine three processes: P_0, P_1, and P_2. P_1 and P_2 are requesting the same resource, and the timestamp of the requests is (6, 1) and (5, 2) respectively. Illustrate the sequence of messages exchanged (use figures). Who gets the resource fist?

(3p)

Tentamen i kursen Distribuerade System -TDDD25, 2012-03-08, kl. 14-18
Du kan skriva på svenska eller engelska!

9.

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)

10. 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)

11. The Byzantine Generals Problem: show how agreement is not or is possible for three and for four generals respectively, in the case one of the generals (not the commander) is a traitor (illustrate the exchange of messages with figures).

(3p)

12. 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)

13. Adjusting drifted clocks: T_{curr} is the time shown by the clock and T_{new} is the value we have to change the clock to.
a) What is the main problem and how is it solved in principle?
b) Concrete solution with mathematical discussion.

(3p)

14. Compare the Ethernet protocol and the CAN protocol from the point of view of predictability. Explain.

(2p)