Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2014-06-11
Sal	TER1
Tid	14 - 18
Kurskod	TDDD25
Provkod	TEN1
Kursnamn/benämning	Distribuerade system
Institution	*IDA*
Antal uppgifter som ingår i tentamen	14
Antal sidor på tentamen (inkl. försättsbladet)	
Jour/Kursansvarig	Petru Eles
Telefon under skrivtid	0703681396
Besöker salen ca kl.	15:15
Kursadministratör (namn + tfnnr + mailadress)	Carita Lilja , 1463, carita.lilja@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	12

LINKÖPINGS TEKNISKA HÖGSKOLA
Institutionen för datavetenskap
Petru Eles

Tentamen i kursen

Distribuerade System- TDDD25

2014-06-11, kl. 14-18

Hjälpmedel:

Engelsk ordbok.

Poänggränser:

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

Supporting material:

English dictionary.

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, 2014-06-11 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. What means transparency in a distributed system? We have defined seven aspects of transparency. Enumerate and explain at least five of them.

(3p)

3. Publish-Subscribe systems:
   a) Draw a figure in which you illustrate the three players (publishers, subscribers, and notification service) and their interaction.
   b) Explain the filtering function and illustrate by an example.

(3p)

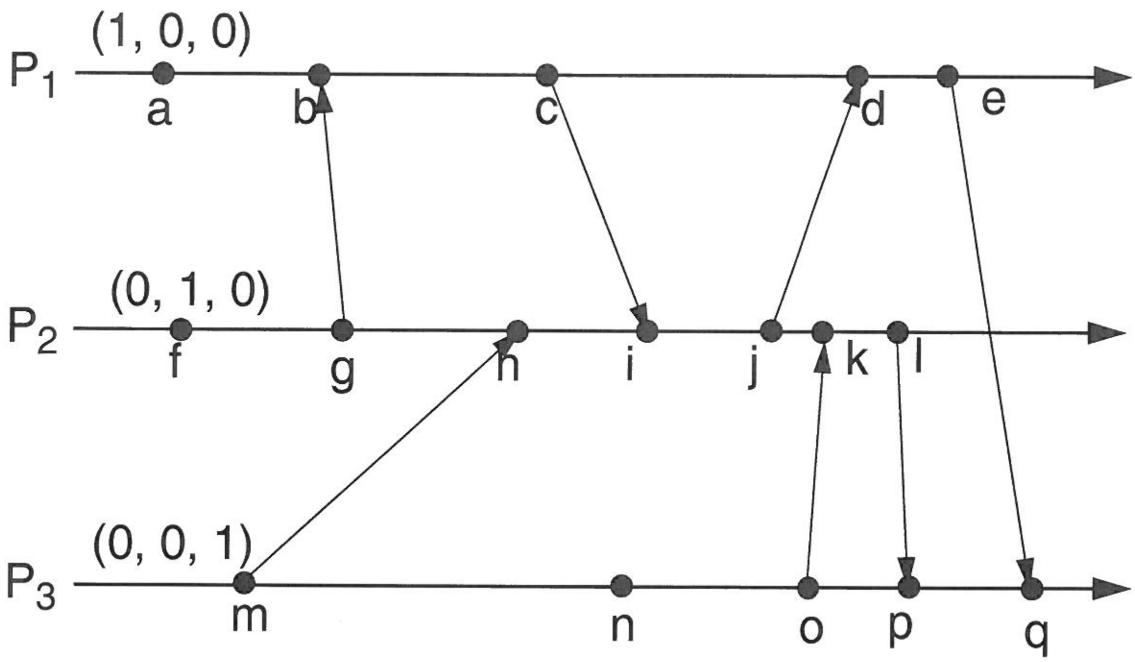
4. What are potential problems with client-server systems?
   How are they solved with peer-to-peer systems?
   What are key issues and problems with peer-to-peer systems?

(2p)

5. We have identified an important limitation of Lamport's logical clocks (NOT the one related to the lack of total ordering, which is not so important).
   a) What is that limitation? Illustrate by an example.
   b) Show, using the same example, how vector clocks solve that problem.

(3p)

6. Consider the following set of events:



Assign the missing vector clock values to the events.

(3p)

7. 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)

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

(3p)

Tentamen i kursen Distribuerade System -TDDD25, 2014-06-11 kl. 14-18
Du kan skriva på svenska eller engelska!

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

(3p)

11.

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)

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. 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? Consider both the case when after synchronisation the clocks are perfectly aligned and the case when after synchronisation there exists an offset Φ between the clocks.

(2p)