TEKNISKA HÖGSKOLAN I LINKÖPING
Institutionen för datavetenskap
Petru Eles

Tentamen i kursen

Distribuerade System- TDDB 37

2000-04-26, kl. 08-12

Hjälpmedel:

Inga.

Poänggränser:

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

Resultat anslås:

Senast 2000-05-11 på IDAs anslagstavla för tentamensresultat.

Jourhavande lärare:

Petru Eles, tel 28 13 96

Good luck !!!

Tentamen i kursen Distribuerade System - TDDB 37, 2000-04-26, kl. 08-12
Du kan skriva på svenska eller engelska!

1. Illustrate by a figure a typical software structure in a distributed system.
   What is typical for a microkernel, as used in a distributed operating system, compared to a traditional monolithic kernel?

   (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)

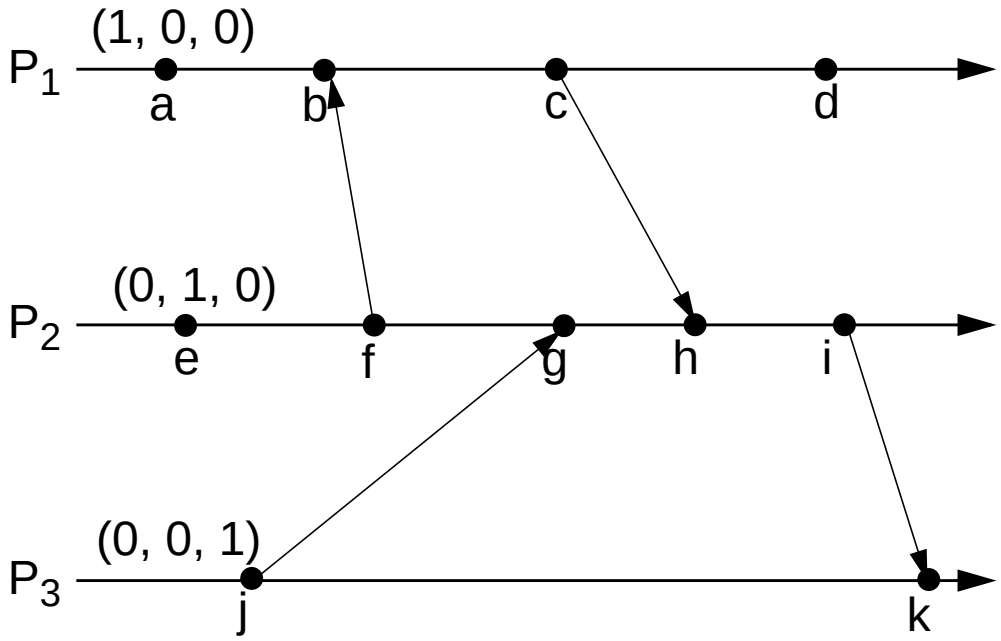
3. Static and dynamic invocation in CORBA:
   How do they work? Compare.

   (3p)

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

   (2p)

5. Consider the following set of events:



Assign the missing vector clock values to the events.

(3p)

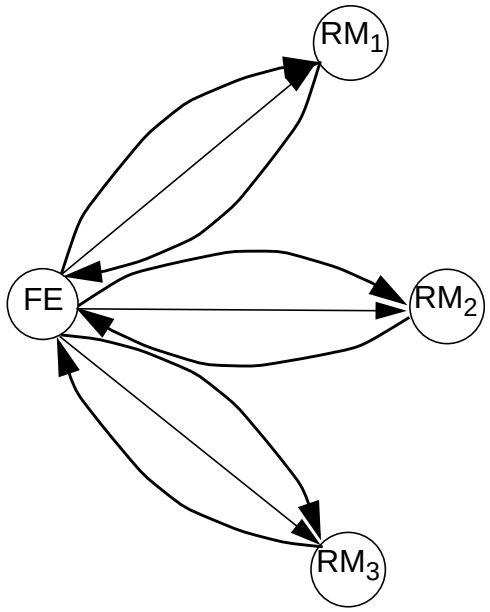
Tentamen i kursen Distribuerade System - TDDB 37, 2000-04-26, kl. 08-12
Du kan skriva på svenska eller engelska!

6. Illustrate by an example that it is not possible to derive the causal ordering of events using Lamport Logic Clocks. Show, using the same example, that with vector clocks the problem can be solved.

(3p)

7.

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 with one front end and several replication managers, like below:



Three messages have to be exchanged by the FE with each RM. Explain. What do this messages contain?

(4p)

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 (8, 1) and (7, 2) respectively. Illustrate the sequence of messages exchanged (use figures). Who gets the resource fist?

(3p)

Tentamen i kursen Distribuerade System - TDDB 37, 2000-04-26, kl. 08-12
Du kan skriva på svenska eller engelska!

9. Define a *k*-fault tolerant system.
   How many components do you need in order to achieve *k*-fault tolerance with byzantine faults:
   - for agreement?
   - with a majority voting scheme?

   (3p)

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

    (3p)

11. Predictability of real-time systems:
    Why is it important?
    What are the main problems?

    (2p)

12. Clock synchronization. What is the problem? Why is this important for real-time systems?
    Show situations (at least two) in which clock synchronization is needed.

    (2p)

13. When drifted clocks have to be adjusted in a distributed real-time system, there are two possible situations (T_{curr} is the current time shown by the clock, and T_{new} is the new time to which the clock has to be changed):
    1) $T_{new} > T_{curr}$
    2) $T_{new} < T_{curr}$
    Comment on these two alternatives and on possible difficulties which can arise.
    How can these difficulties be solved?

    (3p)

14. Compare the CAN protocol and the TDMA protocol from the point of view of how collisions are avoided.

    (3p)