Exam in Distributed Systems

Justin Pearson

June, 2011

Cover Sheet

This sheet should be handed in together with the exam.

Each problem must be solved on a separate sheet. Write your name on each sheet. Indicate below which questions you have answered.

Problem no.	Solution provided	Max	Your points
1		7	
2		15	
3		12	
4		9	
5		20	
6		10	
	Total:	73	

Name : .....

Anonymous Code : .....

Exam Rubric

A mark of 50% is required for a pass, a 4 and 5 are distributed evenly. Answers can be in English or Swedish.

Each full answer should be started on a separate sheet. Please write you name and personal number on each sheet.

I will not be able to come to the exam. If you are unsure about how to answer a question. Make some reasonable assumptions, state your assumptions and answer the exam.

Hjälpmedel:

Pen, pencil, ruler, rubber (eraser for people from the Antipodes or from across the pond) calculator.

1. General Questions on Distributed Systems.

   (a) What problems to networks cause for synchronisation in a distributed system? (**2 points)**

   (b) Explain why it might be a problem to detect failure in a distributed system. (**1 points)**

   (c) Give two examples of where replication could be used in a distributed system, explain in each case why replication is a good thing. (**4 points)**

   (d) What is middle-ware in a distributed system and why is it used? (**2 points)**

   (e) Explain the different types of transparency that are desirable in a distributed system. (**2 points)**

2. Distributed Programming Patterns.

   (a) Describe the client server paradigm. (**1 points)** Give at least one example. (**1 points)**

   (b) Describe what a remote procedure call is.(**1 points)** Further, describe in detail the implementation on both the client and server side of a client-server system. (**3 points)**

   (c) When implementing distributed system middleware there are choices of representation for data marshaling. Describe two approaches, and give scenarios where each of your approaches are appropriate. (**4 points)**

   (d) When designing a distributed system why is it often a good idea to avoid central servers? Give examples. (**2 points)**

   (e) Describe the public subscribe paradigm. (**1 points)** Give at least one example. (**1 points)**.

   (f) How does the implementation of a remote object and a remote procedure call differ, both on the client side and on the server side. (**2 points)**

3. Global State.

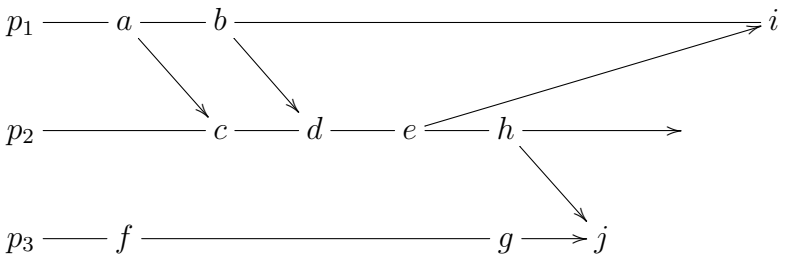
   (a) Why is it hard to record the state of a distributed system. Give examples. (**4 points)**

   (b) Define a consistent cut of a distributed system. (**2 points)** Make sure that you give examples. (**1 points)**. How does a consistent cut help when designing state recording algorithms.(**1 points)**.

4. Clock Synchronization and Timestamps.

   (a) Suppose that you have two clocks both with drift $\rho = 10^{-3} secounds/seconds$. If you want the clocks synchronized within 0.4 of a second, then how often should the clocks be resynchronized to achieve this? (**2 points)** (Obs. Show your workings, no workings no points for a correct or incorrect answer)

   (b) Consider the following three processes p_1,p_2 and p_3 with the following pattern of communication:



   Label the each event with a normal scaler value lamport time stamp (**2 points)** and a vector time stamp (**2 points)**

   (c) What does it mean for two events to be concurrent and what is the relation of the lamport timestamps of the two events. (**2 points)**.

   (d) Lamport timestamps can be used to implement mutual exclusion in a distributed system. Describe an algorithm that does this (using Lamport timestamps) and argue for the correctness of the algorithm. (**5 points)**

5. Transaction and Concurrency Control.

   (a) Define what a transaction is. Be sure to give an example (not an example from my slides) (**2 points)**

   (b) A transaction is supposed to satisfy the so called ACID properties. Explain what the ACID properties are. (**4 points)**.

   (c) Define what an interleaving of two transactions is. (**1 points)**.

   (d) Define what it means for the interleaving of two transaction to be serially equivalent. (**2 points)**.

   (e) Give examples (not examples taken from my slides), and explain why it is necessary to require serially equivalent interleaving of transactions. (**2 points)**

   (f) Lamport timestamps can be used for concurrency control of parallel transactions. Describe an algorithm and argue for its correctness. (**5 points)**