Dept. Comp. Sc. **Distributed Systems**
Vrije Universiteit **20.12.2005**

Part I

This part covers the same material as the midterm exam.

1a Explain the principal working of a remote procedure call (RPC). *5pt*

1b Give two compelling arguments why an RPC can never provide the same semantics as a local procedure call. *5pt*

1c What is the main difference between a remote method invocation (RMI) and an RPC? *5pt*

2a Imagine a file server keeping a table of (client,file)-pairs, identifying which clients accessed which files. Is this a stateful server? Morivate your answer. *5pt*

2b Explain what a message-queuing server is and whether it can be designed to be stateless. *5pt*

2c The X Window system refers to the X kernel running on the client machine as the "X server", and the application running on the compute server, as the "X client." Why is this actually a correct usage of client/server terminology? *5pt*

3a Consider a client that attempts to synchronize its clock with that of a time server once every minute. It sends a number of requests, with the results shown below. How will the client adjust its clock after receiving a response? Time is given in (hr:min:sec:msec). Processing time at the server is zero. *5pt*

Sent at (local time)	Round-trip delay	Response
10:54:00:00	18 ms	10:54:00:10
10:55:00:00	24 ms	10:55:00:12
10:56:00:00	22 ms	10:55:00:10

3b Explain the principle of the Berkeley algorithm for adjusting the clocks in a distributed system. *5pt*

3c The communication layer in distributed systems can keep track of causally related messages. What are two major objections against this functionality? *5pt*

Part II

4a Is the following sequence of events allowed with a sequentially-consistent store? What about a causally-consistent data store? Explain your answer. *5pt*

P1:	W(x)a		W(x)c		
P2:	R(x)a	W(x)b			
P3:	R(x)a			R(x)c	R(x)b
P4:	R(x)a			R(x)b	R(x)c

4b Explain why *writes-follow-reads* consistency guarantees that causal relations are maintained when used for the implementation of a distributed news system. *5pt*

4c What are the conditions to prevent read-write and write-write conflicts in a quorum-based system. *5pt*

5a Show that with 4 processes of which one is faulty, that 3 processes can come to an agreement irrespective of the message communicated by the faulty process. *5pt*

5b Consider a print server with three possible events: (M) notify the client that printing is done; (P) print; (C) crash. When a client is notified that the print server has just recovered from a crash, it can follow 4 different strategies: (1) Never reissue the print request, (2) Always reissue the print request; (3) Reissue only if the delivery of the print request has not been acknowledged; (4) Reissue only when delivery of the request has been acknowledged. Show that if the server always notifies the client after printing, it is impossible to devise a scheme in which the print job is never lost or never duplicated. *10pt*

6a NFS is arguably not a file system. Explain why. *5pt*

6b Coda allows clients to cache files, but sends invalidation messages when a file is modified. What is the main reason for doing these callbacks in parallel? *5pt*

6c Coda allows a reading client to continue to operate on its local copy, even if a concurrent write takes place at the server. Why is this behavior considered correct? *5pt*

Final grade: *(1) Add, per part, the total points. (2) Let T denote the total points for the midterm exam ($0 \leq T \leq 45$); $D1$ the total points for part I; $D2$ the total points for part II. The final number of points E is equal to* $\max\{T, D1\} + D2 + 10$.