Parallel and Distributed Systems Group
Department of Software Technology
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology

Examination Distributed Algorithms (IN4150)

18 juni 2009, 9-12 AM

Notes:

1. The number of exercises is 4, and the number of pages is 2.
2. The solutions to the exercises can be either in Dutch or in English.
3. Try to give **short, concise, and precise answers.**
4. The maximum number of points to be obtained for each part of each exercise is indicated between parentheses. The final grade is computed as 12 plus the total number of points obtained, divided by 10 and rounded to the nearest integer.

1. (a) (4) Give the definition of causal ordering of point-to-point messages.
   (b) (6) Give an example of a system with three processes in which causal ordering of point-to-point messages is violated.
   (c) (7) Give in words or in pseudo-code the algorithm of Schiper-Eggli-Sandoz for causal ordering of point-to-point messages.
   (d) (5) Show how the algorithm of (c) would have enforced causal message ordering in your example of (b).

2. (a) (8) Give in words or in pseudo-code Suzuki's and Kasami's algorithm for mutual exclusion.
   (b) (6) Argue that the algorithm is free of deadlock and starvation.
   (c) (8) Show in detail the execution of the algorithm in a system of four processors three of which request entry to their critical sections, until all requests have been granted.

3. (a) (4) Formulate the Byzantine agreement problem. In particular, state the conditions for *agreement* and *validity*.

   (b) (8) Give in words or in pseudocode the algorithm for randomized Byzantine agreement.

   (c) (4) Argue why this algorithm is also suitable for asynchronous systems.

   (d) (6) Explain the execution of the algorithm when there are in total 11 processes, two of which are faulty and do not send any message, while of the non-faulty processes two start with the value 0 and seven with the value 1.

4. (a) (6) Explain how files and nodes are organized, and in which way files are assigned to nodes, in the Distributed-Hash-Table-based P2P system Chord.

   (b) (4) What is the *basic* way of searching for files in Chord?

   (c) (6) Explain the `fingertable` data structure in Chord, and argue why in a real system many of the first elements of the finger table in a node will coincide. Show how searching can be optimized with the help of this data structure.

   (d) (6) What is the complexity of searching for files with the basic and with the optimized methods of (b) and (c), respectively? Explain your answer in detail.

Parallel and Distributed Systems Group
Department Software Technology
Faculty EEMCS
DELFT UNIVERSITY OF TECHNOLOGY

DISTRIBUTED ALGORITHMS (IN4150)

List of algorithms

D.H.J. Epema

april 2009

Below is a list of the algorithms in the order of treatment in the course, with the names of the original authors, if applicable.

Chapter 3: Synchronization

1. Alpha-, beta-, and gamma-synchronizer: Awerbuch
2. Causal message ordering (broadcast): Birman-Schiper-Stephenson
3. Causal message ordering (point-to-point): Schiper-Eggli-Sandoz
4. Total message ordering
5. Determining global states: Chandy-Lamport
6. Termination detection in a unidirectional ring
7. Termination detection in a general network
8. Deadlock detection (with/without instantaneous communication): Bracha-Toueg

Chapter 4: Coordination

1. Assertion-based mutual exclusion: Lamport
2. Assertion-based mutual exclusion: Ricart-Agrawala
3. Assertion-based mutual exclusion: Maekawa
4. Generalized assertion-based mutual exclusion
5. Token-based mutual exclusion: Suzuki-Kasami T/TN
6. Token-based mutual exclusion: Singhal
7. Detection of loss and regeneration of a token

8. Election in a synchronous unidirectional ring (non-comparison-based)
9. Election in a bidirectional ring: Hirschberg-Sinclair
10. Election in a bidirectional ring (enhanced version)
11. Election in a unidirectional ring: Chang-Roberts
12. Election in a unidirectional ring: Peterson
13. Election in a synchronous complete network: Afek-Gafni
14. Election in an asynchronous complete network: Afek-Gafni
15. Minimum-weight spanning trees: Gallager-Humblet-Spira

Chapter 5: Peer-to-peer Systems

1. Unstructured: Freenet and Gnutella
2. DHT: Chord, CAN, Pastry
3. BitTorrent
4. Epidemic protocols: anti-entropy and rumor mongering

Chapter 6: Consensus

1. Agreement with stopping failures
2. Byzantine agreement with oral messages: Lamport-Pease-Shostak
3. Byzantine agreement with authentication: Lamport-Pease-Shostak
4. Randomized Byzantine agreement
5. Stabilizing mutual exclusion: Dijkstra
6. Stabilizing stop-and-wait datalink algorithm
7. Stabilizing sliding-window datalink algorithm