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)

22 january 2007, 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) **(8) Give** in words or in pseudo-code the algorithm of Chandy and Lamport for detecting the global state of a distributed system.

   (b) **(2)** What are possible uses of this algorithm?

   Consider a system consisting of three bank accounts (processes) P_1, P_2, and P_3 with some initial values and with a separate channel each way between every pair of accounts. Processes can transfer amounts of money to other processes by sending messages.

   (c) **(8)** Present a scenario in which every process sends at least one message to another process and an execution of the algorithm in that scenario that detects a global state that has not actually occurred in the system. Clearly specify the global state that has been detected, and argue that this state indeed has not occurred.

   (d) **(4)** Show a different order of the events in the system such that the detected state would have occurred.

2. (a) (7) Give in words or in pseudo-code the *candidate process* of Afek's and Gafni's election algorithm in a synchronous complete network.

   (b) (4) Give in words or in pseudo-code the *ordinary process* of this algorithm.

   (c) **(4)** Assume that there are 8 processes and that all processes spontaneously start the algorithm at the same time. Show that it can take 3 rounds before only one process is still active.

   (d) **(7)** Argue that in the general case with n processes, and with all processes spontaneously starting the algorithm at the same time, it can take $\log_2 n$ rounds before only one process is still active.

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

Assume in the rest of this exercise that no authentication is used, and that n is the total number of generals and f the number of traitors.

(b) (6) Show that there is no solution to Byzantine agreement with $n = 3$ and $f = 1$.

(c) (6) Give in words or in pseudo-code the algorithm OM(f) for Byzantine agreement.

(d) (6) Assume that $n = 7$ and $f = 2$, and that the commander does not exhibit failures (is loyal). List in a systematic way all the messages a loyal lieutenant receives in every round of the algorithm, and show in detail how he deduces his final decision from these messages.

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) (5) Explain the *finger table* data structure in Chord.

(c) (5) Describe how a node that wants to join a Chord ring can obtain the values of the entries in its finger table.

(d) (6) Explain in detail when a new node joins a Chord ring, how the *predecessor* and *successor* pointers in the new node and in its predecessor and successor can be set correctly.