Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Prof. Dr. W.-P. de Roever
Heiko Schmidt, Tim Fenten, Christian Motika

Verteilte Algorithmen

Wintersemester 2007/08 — **Serie 7 (MST)** — Montag, 3. Dezember 2007

Thema: Mittsemestertest

Ausgabetermin: Montag, 3. Dezember 2007

Abgabe: **Montag, 17. Dezember 2007 (12:00) Strikte Deadline!**

> **Mittsemestertest:** Diese Aufgabenserie ist von jedem *allein* zu bearbeiten! Die Aufgaben sind etwas breiter gestreut und die erreichten Punkte in dieser Serie gehen verstärkt in die Endnote ein. Bitte beachten Sie den Abgabetermin, der vom üblichen Turnus abweicht. Viel Erfolg!

Aufgabe 1 (6 Punkte) Design a consensus algorithm for four processes in a completely connected graph that tolerates *either* one Byzantine fault *or* three stopping faults. Try to minimize the number of rounds. (You do not need to give the formal code, if your informal description is clear and unambiguous, and of course fits the considered model of execution.)

Aufgabe 2 (6 Punkte)

(a) Give code for the *ThreePhaseCommit* algorithm (including the termination protocol).

(b) Modify the code such that the algorithm permits processes to decide and halt quickly in the failure-free case. Your algorithm should use a small constant number of rounds and $O(n)$ messages, in the failure-free case.

(c) Prove the correctness of the modified algorithm.

Aufgabe 3 (4 Punkte) Prove that the *EIGStop* algorithm, modified so that all messages are signed and only correctly signed messages are accepted, solves the consensus problem for the authenticated Byzantine failure model (cf. Section 6.2.4). For the proof, first show that the following assertion (analogous to the statement of Lemma 6.12) is true:

Assertion 1 After $f + 1$ rounds:

(a) If i and j are nonfaulty processes, $val(y)_i = v \in V$, and xj is a prefix of y, then $val(x)_j = v$.

(b) If v is in the set of vals at any nonfaulty process, then v is an initial value of some process.

(c) If i is a nonfaulty process, and $v \in V$ is in the set of vals at process i, then there is some label y that does not contain i such that $v = val(y)_i$.