TUM I 13
Chair of Application and Middleware Systems
Prof. Dr. H.- A. Jacobsen

Tutorial
Distributed Systems (IN2259)
WS 2020/21

EXERCISES ON COORDINATION & AGREEMENT

EXERCISE 1 Multicast-based Mutual Exclusion

Does the following algorithm solve the mutual exclusion problem? Discuss your answer based on the requirements for the solutions of the mutual exclusion problem, as discussed in the lecture.

```
On initialization
        state := RELEASED;

To enter the section
        state := WANTED;
        Multicast request to all processes;
        Wait until (number of replies received = (N - 1));
        state := HELD;

On receipt of a request Pi at Pj (i!=j)
        if (state = HELD or (state = WANTED and Pj < Pi))
        then
                        queue request from Pi without replying;
        else
                        reply immediately to Pi;

        end if

To exit the critical section
        state := RELEASED;
        reply to any queued requests according to the current status of process;
```

EXERCISE 2 Ring-based Mutual Exclusion

The original algorithm requires the token to be passed continuously even when no process desires access to the critical section, thereby wasting communication resources. Propose a new algorithm where Chang & Roberts algorithm is integrated with the ring-based approach in a way that communication cost is only required when there is at least one process that requires access to the critical section. Assume an asynchronous model with reliable communication, FIFO channels, and no process failure. Discuss if your proposed solution is safe?

EXERCISE 3 Leader Election

(a) For leader election, assume that two processes detect the demise of the coordinator simultaneously and both decide to hold an election using the Bully algorithm. What happens?

(b) How is the Bully algorithm dealing with (i) temporary network partitions and (ii) slow processes. What happens?

Suggest an adaptation if necessary.