Universität Heidelberg
Wintersemester 18/19

Group Parallel and Distributed Systems (PVS)
Artur Andrzejak, Diego Costa

Problem Set 9 for lecture Distributed Systems I (IVS1)

Due: 08.01.2019, 14:00 Uhr

Exercise 1 (2 Points)

Take a look at the system S presented in the slide „Pseudo-Stabilizing Systems“ from Lecture 10. It has been showed that S is a pseudo-stabilizing system. Find out and substantiate whether S is also a stabilizing system.

Exercise 2 (6 Points)

In this exercise you will implement a simulation of the Ricart and Agrawala algorithm. First, create an object `Process`, which should contain the following fields:

- an unique ID,
- a Lamport-Clock, which can be simulated with a numeric variable,
- a state which can assume one of the following values {RELEASED, WANTED, HELD},
- a queue of *received* messages and *queued* messages.

Your simulation should work with an array of Nprocesses. At each step your `Process` should go over the procedures depicted in slide 25 („Algorithm of Ricart und Agrawala “). Your program should:

- never block at a specific `Process`, i.e., if there is still messages to be received, your `Process` should go over the next procedure without entering the critical section,
- never violate the mutual-exclusion constraint, only one `Process` can be at the critical section,
- simulate the multicast by adding a message into the *received* queue of the other `Process`
- a process that enters the critical section should print in the console „Processor [ID] is in the critial section!“,

Run your simulation in a for loop, allowing each `Process` to execute its action sequentially. Submit your code and the output of your program for $N = 2, 4, 8, 16$.

Exercise 3 (Bonus, 4 Points)

Improve upon the proof presented on slides „Lemma: Dijkstra‘s Token Ring Converges“ from Lecture 10. Show that for every possible starting configuration, the system will reach a legitimate configuration in L, a state where only 1 process is privileged, within $O(n^2)$ steps. Hint: Use should use the step definition shown in class.