

Series 01 — 20.09.2017 – v1.0a

Introduction to Concurrency

Exercise 1 (7 Points)

Answer the following questions (1 point each):

- Do recent central processing units (CPUs) of desktop PCs support concurrency? Why became concurrency for many software applications very important these days?
- What is safety? Give one concrete example of a safety violation.
- What is liveness? Give a concrete example of a liveness violation.
- Using the implementation in the slides, can a binary semaphore lead to a deadlock? Can it lead to starvation? Explain with the aid of an example.
- Why do we need synchronization mechanisms in concurrent programs?
- How do monitors differ from semaphores? Please provide a precise answer.
- How are monitors and message passing similar? And how are they different?

Exercise 2 (2 points)

```
x := 1
Thread 1 -> x := x + 7.
Thread 2 -> x := x * 5.
```

Considering the previous code, give all possible values of x at the end of the execution of both threads with their corresponding execution traces.

Hint: You should be able to perceive 6 different execution flows in total, however, some of them could may lead to the same x .

Exercise 3 (1 points)

Implement a monitor using semaphores. Use pseudo-code and comment it.
