Prof. Dr. Oscar Nierstrasz
Pascal Gadient, Mohammadreza Hazhirpasand

Assignment 01 — 22.09.2021 – v1.0
Introduction to Concurrency

Exercise 1 (7 pts)

Answer the following questions:

a) Do recent central processing units (CPUs) of desktop PCs support concurrency? (1 pt)
b) Why do we need synchronization mechanisms in concurrent programs? (1 pt)
c) What is safety in the context of concurrent programs? (0.5 pts)
d) Give one concrete example of a safety violation. (0.5 pts)
e) What is liveness in the context of concurrent programs? (0.5 pts)
f) Give a concrete example of a liveness violation. (0.5 pts)
g) Why or why not can a binary semaphore lead to a deadlock? (0.5 pts)
h) Why or why not can a binary semaphore lead to starvation? (0.5 pts)
i) How do monitors differ from semaphores? Please provide a precise answer. (1 pt)
j) What are similarities between monitors and message passing? (0.5 pts)
k) What are differences between monitors and message passing? (0.5 pts)

Exercise 2 (2 pts)

```
x := 1
Thread 1 -> x := x + 3.
Thread 2 -> x := x * 2.
```

Considering the code above: Give all possible values of `x` at the end of the execution of both threads together with their corresponding execution traces.

Hint: You should be able to identify six different execution flows, however some of them could lead to the same `x`.

Exercise 3 (1 pts)

Implement a monitor using a binary semaphore, *i.e.*, a semaphore that can contain at most one token. Use pseudo-code and comment it.