Prof. O. Nierstrasz
Jorge Ressia, Jan Kurš

Serie 1 - Introduction to Concurrent Programming

Don't forget to register to the mailing list at https://www.iam.unibe.ch/mailman/listinfo/cp-course

Exercise 1

Answer the following questions (be clear and precise):

- What is the difference between concurrency, concurrent programming, and parallelism?
- What are safety and liveness?
- What is the difference between deadlock and starvation?
- Give an example of deadlock. (Technical example or invented situation in the real life)
- Give an example of starvation. (Technical example or invented situation in the real life)
- Why do we need synchronization mechanisms in concurrent programs?
- How exactly do monitors differ from semaphores?
- When does it make sense to use busy-waiting?
- Are binary semaphores as good as counting semaphores? Explain your answer.
- What problems could nested monitors cause?

Exercise 2

```
x := 1
Thread 1 -> x := x + 5.
Thread 2 -> x := x * 3.
```

Considering the previous code, give all the possible value of x at the end of the execution of both threads in case the safety has been respected or not (Hint: 4 result expected).

Exercise 3

Implement a semaphore using monitors. Use pseudo-code and comment it.

Exercise 4

Implement monitors using semaphores. Again, use pseudo-code and comment it.