

Concurrency: State Models & Design Patterns

Practical Session

Week 01

Exercises 01

Preview

Exercise 01 - Task 1

- a) Do recent central processing units (CPUs) of desktop PCs support concurrency? Why became concurrency for many software applications very important these days?**
- b) What is safety? Give one concrete example of a safety violation.**
- c) What is liveness? Give a concrete example of a liveness violation.**
- d) 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.**
- e) Why do we need synchronization mechanisms in concurrent programs?**
- f) How do monitors differ from semaphores? Please provide a precise answer.**
- g) How are monitors and message passing similar? And how are they different?**

Exercise 01 - Task 2

$x := 1$

Thread 1 $\rightarrow x := x + 7.$

Thread 2 $\rightarrow x := x * 5.$

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

Exercise 01 - Task 3

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