Parallel Programming WS 2014/2015 - Assignment 3

Kastens, Pfahler
Institut für Informatik, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn
Nov 24, 2014

Exercise 1 (Monitor Development: The one-lane bridge)

Cars coming from the north and the south arrive at a one-lane bridge. Cars heading in the same direction can cross the bridge at the same time, but cars heading in the opposite direction cannot.

Develop a solution for this problem. Model the cars as two different kinds of processes and design a monitor for synchronization using the 5 steps procedure explained in the lecture.

a) Define the monitor state and the entry procedures.
b) Specify the monitor invariant.
c) Insert conditional waits.
d) Insert notifications.
e) Eliminate unnecessary notifications.

You don’t have to worry about fairness.

Exercise 2 (Monitors in Java: The bounded buffer)

Directory `blatt3/buffer` contains a complete simulation of the bounded buffer problem using a monitor in Java.

a) Compile and run the simulation.
b) Integrate output statements that print the average number of elements in the queue whenever an element is queued or dequeued. Make experiments to show how the number of producers and consumers influences the average queue length.

You can use the bounded buffer implementation as a template for the following exercise.

Exercise 3 (Monitors in Java: The one-lane bridge)

Develop a Java implementation of a one-lane bridge simulation using the monitor designed in assignment 1.

The main simulation loop could be implemented by the following class:

```
public class OneLaneTest {

    public static void main(String[] args) {
        OneLane lane = new OneLane();

        for (int i = 0; i < 100; i++) {
            if (Math.random() < 0.5) {
               new SouthBound(lane).start();
            }
            else {
               new NorthBound(lane).start();
            }
            try {
                Thread.sleep((int)(Math.random() * 1500));
            } catch (InterruptedException e) {}
        }
    }
}
```

The monitor is provided by the object of class `OneLane`. The two kinds of car processes are called `NorthBound` and `SouthBound`. After entering the bridge a simulated car should leave it after 1 second.

You will need the following classes:

- `OneLaneTest` (given above)
- `SouthBound`, processes for southbound cars
- `NorthBound`, processes for northbound cars
- `OneLane`, the monitor implementation

Please provide log output such that you can observe what happens on the one-lane bridge. This output could for instance look as follows:

```
    ...
    3 southbound cars on lane
    2 southbound cars on lane
    1 southbound cars on lane
    0 southbound cars on lane
    1 northbound cars on lane
    0 northbound cars on lane
    1 southbound cars on lane
    0 southbound cars on lane
    ...
```