Concepts of Programming Languages: Parallel Programming in Java

September 20, 2007

1 Introduction

For this assignment two programs, written in Java, must be compiled and run, studied and analyzed, and then the student is expected to write another two programs in Java and run them. The purpose of this assignment for the student is to get some experience implementing concurrent programs in Java.

To get the Java software available to you on the computers of our department you should add `javac` to `wi_packages` into the `.tcshrc` file in your home directory:
`"set wi_packages=(...,javac,...)"`.

This assignment has three parts:

1. A simple Java applet is studied.
2. Another simple Java program is studied, which uses threads to implement a solution to a problem involving several beer drinkers and a toilet.
3. Two simple Java programs must be written, which use threads to implement the readers-writers problem.

All examples can be found on the Course Documents section on blackboard.

2 A Simple Java Applet

Look at `"Ogen.html"` in `"Java.zip"`. This will show an applet with two eyes following the cursor movements within a specified range. Notice how each pupil moves within its own eyehole to follow the cursor movements.

You can find the source for this program in the zipfile in `"Ogen.java"` together with a copy of the file `"Ogen.html"` and a makefile `"Makefile"`.

Copy these three files to a directory of your own and study the makefile to find out how you can compile the program yourself and execute it with the help of a web-browser or appletviewer.

Then study the java source and try to find out why the eye-pupils never come out of their eyeholes; you do not have to submit an answer for this.

3 A Simple Example Java Program with Threads

Copy the files of `"Beer_drinkers"` to a directory of your own and study the makefile to find out how you can compile the program yourself and execute it.

This time the program runs directly as a java application program, and not as an applet within a web-browser or appletviewer.

Analyze the java source code for this program and find out how this program implements a solution for the problem explained in the source file. Once again, there is no need to submit a report for this assignment.

4 Write a Simple Java Program with Threads Yourself

Inspired by the programs above you are expected to write two simple Java program yourself.

The Readers/Writers Problem

Two kinds of processes – readers and writers – share a database. Writers execute transactions that update the database; reader transactions access the database without modifying it. The database is assumed initially to be in a consistent state (i.e. one in which relations between data are meaningful). Each transaction, if executed in isolation, transforms the database from one consistent state to another. To preclude interference between transactions, a writer process must have exclusive access to the the database. Assuming no writer is accessing the database, any number of readers may concurrently execute transactions.

First Program

Implement the Readers/Writers problem (described above) using **monitors**.

Second Program

Implement the Readers/Writers problem (described above) using **monitors** giving priority to the writers; i.e. if a writer wants to enter the database but has to wait then no reader may enter, and of course the writer still has to wait for the current readers in the database to be finished.

Monitors and Condition Variables in Java

Every object in Java with *synchronized* methods acts as a **monitor**. Unfortunately Java does not allow the use of more than one **condition variable** per object.

You can use the following synchronization methods:

1. **wait()**: blocks the invoking thread.
2. **notify()**: reactivates one thread blocked in a **wait()**.
3. **notifyAll()**: reactivates all threads blocked in a **wait()**.

5 How to submit

Your programs should be submitted together with a makefile and a written report in which you explain your programs, to Thijs van Ommen (mvommen@liacs.nl).

6 Some Hints

To stop an endless Java program, use `"CTRL-C"`.

If you want to know more about the Java language, just visit `http://java.sun.com`. Among other things you may find there an on-line collection of Java tutorials at `http://java.sun.com/docs/books/tutorial/index.html`.