Concurrent Programming

COMP 409, Winter 2008

Assignment 4

Due date: Friday, December 3, 2010

Note: Assignments must be handed in **in class**. Late assignments will only be accepted with prior **written** permission of the instructor. You must **explain** all answers and **show all work** to get full marks!

Assignment Questions

1. In the simulation used in the programming section the `CircuitLoop` (default) simulation requires only a capacity of 1 on each buffer. The `CircuitSquares` example, however, requires 2 in order to operate. This may be due to the fact that in the implementations multiple channels are connected together to form longer channels, which also gives more capacity. Suppose, instead, that all connections are made by single channels. What is the minimum buffer capacity for the two circuits? Why? **5**

Programming

You must use Java for the following implementation. In all cases your code must be in a professional style: **well-commented,** properly structured, and appropriate symbol names. Marks will be very generously deducted if not! All programs should include demonstrative, but not excessive, output. Your programs should not have race conditions and should maximize the ability of threads to execute concurrently.

Code is provided on the course website for a graphical *dataflow* simulator. A "circuit" is formed of *functions* and *channels*, with channels providing bounded FIFO connectivity between functions. Each function operates as a separate thread, blocking until data is available on all input channels and there is sufficient capacity on all output lines. Once enabled by the availability of all input data and output resources, it consumes data from the input lines, does its functional behaviour, and writes the result to its output lines. This is not necessarily atomic, although channels should stay in consistent states. There are two circuits you can simulate; see the `readme.txt` with the text for code details.

The code provided, however, is incomplete and has a number of important flaws you must fix. In particular, there is no direct synchronization, and as given it will probably not run without ConcurrentModificationExceptions or other errors. Note that as well as a thread for each Function (and this includes some Channels when they are chained together to make longer channels—see the code) there is a GUI event handler thread to consider as well.

1. Channels are read, written, and drawn by different threads. This needs to be made thread-safe. **3**
2. Functions checking for the availability of input/output channels need to block properly. **3**
3. The GUI provides start, pause and resume buttons. These need to be implemented properly. Buttons should have sane behaviour—pressing start more than once should have no effect, and pressing pause repeatedly or continue when not paused should have no effect. **4**
4. The GUI provides a button to increase the overall capacity (bound) of channels. This needs to be safely implemented. Note that if a thread is blocked waiting for a channel to empty out it may no longer need to block if capacity is increased. **5**
5. The GUI needs a button to decrease the overall capacity (bound) of channels. This must also be safely implemented. Note that this can make channels overcapacity; they should behave correctly (accept receives but not sends). Note that this can also put a function in a state where it is has verified capacity for at least some channels, but now capacity has changed. Your code should behave sensibly in all these cases. **10**

What to hand in

For assignment submission you will use `moodle`:

`http://moodle.cs.mcgill.ca/moodle`

If you did not use this for assignment 1 be sure that you login and familiarize yourself with the system before the deadline. Clock accuracy varies, and late assignments will not be accepted without a medical note: you have ample time, **do not wait until the last minute** to do the assignment! Assignments must be submitted on the due date **before 6pm**.

Where possible hand in only **source code** files containing code you write. Do not submit compiled binaries or .class files. For the written answer questions submit either an ASCII text document or a .pdf file *with all fonts embedded*. Do not submit .doc or .docx files.

Note that for written answers you must show all intermediate work to receive full marks.

This assignment is worth 10% of your final grade. **30**