Carleton University
Department of Systems and Computer Engineering
SYSC 3303 - Real-Time Concurrent Systems – Winter 2014
Assignment 1

Assignments are to be completed individually, not in your project teams. This document is descriptive not prescriptive – ask questions in class or in a discussion forum if you need more information.

Objectives:

- Further practice in basic network programming and threading
- Learning how to construct an experiment for testing concurrent systems

Part A Network Programming

There are many intricacies and exceptions in network programming. In this part, we are going to explore queuing at the UDP level. The answers can be found by reading.

Example: 2buntu.com/1209/udp-untangled-overview-of-how-udp-works/

In this assignment you will find your answers heuristically. Your task is to construct an experiment to demonstrate that the UDP communication is *asynchronous* :-

1. The sender does not need to be ready-and-waiting when a client sends a datagram,
2. The client continues execution concurrently.
3. Consequently, the UDP layer provides queuing of datagrams (under one condition).

Requirements:

1. You must write a rationale and method for your experiment
2. You must write the code in Xenomai/C.
3. You must provide your experimental data as well as a summary/analysis of the data showing how your experiment proved that UDP communication is asynchronous.
4. For full marks, you must identify the condition in which datagrams are not queued, and your experiment must also include that condition.

Notes:

1. Your experiment may be run on one machine (i.e. you can work at home on your laptop) but your code should be written so that it would be easy to run the experiment on two programs.

Part B Threading

Threads can share data, yet their concurrency opens up the danger of data collisions. We will spend the rest of the term learning to write code that is “thread-safe”. In this assignment, you will write code that is **not** thread-safe so that you can prove that data collisions do occur (hopefully!)

Advanced Reading: For this assignment, ignore the synchronization solutions. Make sure you understand the problem of non-atomic operations and visibility: http://www.baptiste-wicht.com/2010/08/java-concurrrency-synchronization-locks/

Your program will:

2. Be written in Java
3. Be multi-threaded. You may use either Thread or Runnable.
4. Be able to create various numbers of threads, but always in pairs. The goal will be to run the program with more threads than you have cores on your (lab or home) machine.
5. Not use any thread synchronization.
6. Share a counter between threads:
   a. The counter must be double or long (Rationale:Stack Overflow: What operations are considered atomic)
   b. To maximize the chance of a collision, the thread shall ***read-test-and-write*** the counter
   c. The final state of the counter will be compared against its expected value to test whether a collision occurred during the program execution.
7. Will execute a ***statistically significant number*** of times, so as to maximize the chance of collisions occurring
8. Not print data during the execution (during initialization and termination is fine)

Requirements:

1. You must write a rationale and method for your experiment
2. You must provide your experimental data as well as a summary/analysis of the data showing how your experiment proved that data corruption occurs.

Common Instructions

1. **Rationale:** A rationale begins with a definition of the topic-under-study (e.g. asynchronous communication) followed by the experiment's hypothesis and a broad description of the experiment's goal. (5-10 sentences, possibly with a diagram, depending on the topic).
2. **Method:** An explanation of the code that was written to (dis)prove the hypothesis. A few sentences about the components, their role, and their expected behavior. Finally, something should be said about what data will be collected/measured/observed to arrive at the experiment's conclusion.
3. **Results:** A data printout – or an excerpt – or a calculation (e.g. average, min, max) of the data, even the presence or absence of data. Whatever representation, the data should support a conclusion about the experiment
4. **Analysis:** A statement that makes a conclusion based on the results.

Work Products (Deliverables)

<u>For each Part,</u>

1. README.txt: <u>DETAILED</u> instructions on how to run your code. If the TA can't easily run your system, you don't get any marks!
2. A PDF containing the experiment's rationale, method, results and analysis.

   Note: Be concise**. A long report is not necessary and is not the goal of this assignment**. Your document should simply convey the notion that the code that you submit was methodically built to prove a supposition and help the TA navigate your code. One page, at most 2 pages.

3. The code, zipped.
4. If applicable, the data results.

Reminder: The TAs will mark your assignments **in the lab**. It is your responsibility to ensure that your code works in that environment, and that any software required for viewing any text/diagrams is also present in that lab.

Assignments are to be submitted electronically on CULearn Course page. Emailed submissions will not be accepted. See the course outline for the procedure to follow if illness causes you to miss the deadline.

Marking Scheme

Category	Criterion	Limit	Points	Comment
Part A				
Experiment Rationale	A clear expression of the purpose of the experiment	5		
Experiment Method	A structured methodical applicable procedure for producing (a) scenario(s) that (is) are relevant to the purpose of the experiment.	10		
Experiment Code	• Successful execution of the code. • Legibility and completeness of README • Quality of Code (correct, concise)	10		
Experimental Results	• Actual experimental data presented. • Data is minimal, relevant, digestible e	10		
Experiment Analysis	A logical conclusion derived by relating the experimental results to the experiment's method and code.	5		
Part B				
Experiment Rationale	A clear expression of the purpose of the experiment	5		
Experiment Method	A structured methodical applicable procedure for producing (a) scenario(s) that (is) are relevant to the purpose of the experiment.	10		
Experiment Code	• Successful execution of the code. • Legibility and completeness of README • Code correctly implements the experimental method as described	10		
Experimental Results	• Actual experimental data presented. • Data is minimal, relevant, digestible	10		
Experiment Analysis	A logical conclusion derived by relating the experimental results to the experiment's method and code.	5		
Overall Code Evaluation	Using of coding standards (indentation, naming,documentation); Code Maturity & Quality (minimal,clean,robust)	10		
Overall Written Evaluation	Writing is organized, has correct spelling and grammar, makes good use of diagrams, is coherent and concise.	10		
Total		/100		