Carleton University
Department of Systems and Computer Engineering
SYSC 3303 - Real-Time Concurrent Systems - Summer 2013

Assignment 2

The Sandwich-Making Chefs Problem. (This problem was first published as the cigarette-smokers problem by S. Patil in 1971, and is one of several classic process-coordination problems that are used to evaluate facilities for synchronizing concurrent threads and processes.)

Consider a system with three *chef* threads and one *agent* thread. Each chef continuously makes a sandwich and then eats it. But to make and eat a sandwich, the chef needs three ingredients: bread, peanut butter, and jam. One of the chef threads has an infinite supply of bread, another has peanut butter, and the third has jam. The agent has an infinite supply of all three ingredients. The agent randomly selects two of the ingredients and places them on a table. The chef who has the remaining ingredient then makes and eats a sandwich, signalling the agent on completion. The agent then puts out another two of the three ingredients, and the cycle repeats.

Follow the design process outlined in class to develop a Java monitor that synchronizes the agent and the chefs. Produce UCM(s), UML collaboration diagrams and a UML class diagram for your design. Then write a program (following your design) to simulate the agent and the chefs. The program should run until 20 sandwiches have been made and consumed.

Work Products

Put these in a folder called *part1*.

1. A “README.txt” file explaining the names of your files, IDE used, set up instructions, etc.
2. Your design diagrams including:
   - at least one UCM of the system
   - a UML collaboration diagram for each active object
   - a UML class diagram for your system

   Hand-drawn (and then scanned or photographed) diagrams are acceptable, as long as they are neatly drawn and your handwriting is legible.
3. The Java source code for your classes, as well as any files required to run these files in the Java IDE chosen. Your code should demonstrate good programming style, be well documented, etc. For examples of "industrial quality" Java code, have a look at Sun's Java coding conventions, which can be found on our Java resources Web site. (This site can be reached by a link from the SYSC 3303 Web site.)

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. You may use any Java IDE available in the lab (e.g. JCreator or Eclipse).

Submitting Assignments

Assignments are to be submitted electronically using the assignment “submit” program. Emailed submissions will not be accepted. See the course outline for the procedure to follow if illness causes you to miss the deadline.

Due: Tuesday, May 21st at 8pm SHARP!