CSc 422/522 — Homework 3

due Tuesday, March 6, 2001

Problems 1 through 4 are worth 5 points each. Problem 5 through 7 are worth 10 points each. Again, graduate students are to solve all problems (50 points), and undergraduates are to solve any combination of problems that adds to 40 points.

Hand in written answers to problems 1 through 5, commented listings of your programs for problems 6 and 7, and sample output from your programs. *In addition*, use the `turnin` program to submit the source code for your programs; see the details at the end of the assignment.

Be sure to explain your answers and to explain any assumptions you make.

1. MPD book, Exercise 4.6.

2. MPD book, Exercise 4.20.

3. MPD book, Exercise 5.5.

4. MPD book, Exercise 5.10.

5. MPD book, Exercise 4.30.

6. Write a program to simulate the Water Molecule Problem described in Exercise 4.25. Each atom should be a process that "bounces around" for a random amount of time, then calls an `Hready` or `Oready` procedure. After the atom has synchronized with other atoms to form a water molecule—and hence `Hready` or `Oready` has returned—then the atom should terminate. The program should terminate when either all atoms have terminated or there are not enough left to form any more water molecules.

Write your program either in MPD or in C plus Pthreads. Your program should have two command-line arguments: `numH`, the number of hydrogen atoms, and `numO`, the number of oxygen atoms. The `makeWater` procedure in your program should write a line to the standard output file every time a new water molecule is formed. You might want to include "time stamps" in these lines, so that you can see when water molecules are formed. At the end of your simulation, write out how many water molecules were formed and how many atoms of each type remain.

The MPD program `rw.simulation.mpd` shows how to simulate the actions of processes. This program was handed out in class. It is described in the MPD tutorial and is also located in `/home/cs522/Programs/mpd`.

7. Write a program to simulate the Roller Coaster Problem described in Exercise 5.17. Your program will contain passenger processes, one car process, and a monitor to synchronize them. Each passenger process should repeatedly "do other things" for a random amount of time, then take a roller coaster ride. The car process should repeatedly wait for `C` passengers to want to take a ride, go around the track, then let the passengers unload. Assume that it takes `rideTime` seconds for the car to go around the track. The program should terminate after each passenger has taken `numRides` trips on the roller coaster.

Write your program in MPD, in Java, or in C plus Pthreads. If you use MPD, structure your program as in `rw.simulation.mpd` and simulate monitor entry, exit, wait, and signal using semaphores as shown in Figure 6.7. If you use Java, see Section 5.4 for how to program threads and synchronized methods. If you use Pthreads, see Section 5.5 for how to program monitors.

Your program should have four command-line arguments: `C`, the capacity of the roller coaster car; `n`, the number of passenger processes; `numRides`, the number of rides each passenger takes; and `rideTime`, the number of seconds it takes for the car to go around the track.

The output of your program should be a trace of the significant events that occur: passengers calling `takeRide`, the car calling `load`, the car starting around the track (return from `load`), the car calling `unload`, and passengers getting off the car (return from `takeRide`). Each line of output should contain a time stamp and a short descriptive message, including the passenger identity where appropriate. Write the trace to the standard output file.

Electronic Turnin.

Use the `turnin` program on Lectura to turn in copies of your programs. For problem 6, the assignment name is `hw3.prob6`. The file name should be `water.mpd` or `water.c`, depending on whether you use MPD or C/Pthreads.

For problem 7, the assignment name is `hw3.prob7`. The file name should be `coaster.mpd`, `coaster.java`, or `coaster.c`, depending on whether you use MPD, Java, or C/Pthreads.