Project 3
Concurrent Programming

21st November 2002

This project is for all graduate students and for undergraduates who wish to receive extra credit.

Important dates

- 11/21 – Project Assigned
- 12/3 – turn in 1 page writeup of what you intend to do
- 12/10 and 12/12 - 10 minute talks about projects in class
- 12/17 - turn in project (this Tuesday of finals week)

Project summary

The purpose of this project is to give you the chance to explore some aspect of concurrent programming in more depth or to look at some aspect that we haven't discussed in class.

You are to do one of the following:

- design and implement a program that makes creative use of concurrency. You may use Java, C, C++, or another language (but talk to me first about which language). Your program should have a rich set of process interactions. Your program must be accompanied by a paper that describes the problem you set out to solve, research you did into previous solutions, how you solved it, and an assessment of how well your program solves the problem. If you consulted researchers papers or books, include a bibliography.

- Write a paper that analyses some aspect of concurrent programming. You might include a short demonstration program along with your paper. Your paper should be 15-20 pages in length and be your own, original writing. It must contain a bibliography of the papers and books that you consulted in writing the paper.

You may work together with someone, but in that case I expect that the project will be significantly (a factor of 2) more ambitious than what I expect of a single-person project.

One guideline on what I expect would be that this project should be roughly comparable in effort to the other two.

During the last week of class you will make a short presentation (10 minutes) to the class about your project.

If you do a programming project you will need to demonstrate it to me before the final exam – I'll be here all of finals week so e-mail me for an appointment.

Grading

Programming projects:

- Creative use of concurrency
- Appropriate use of concurrency mechanisms, for example, I would not expect to see any busy waiting.
- Statement of problem
- Summary of relevant research
- Description of your solution, including documentation of the structure of the program itself
- Assessment of how well your solution works, including a description of how you developed evidence to support your conclusions

Papers:

- Statement of problem or issue
- presentation of evidence
- presentation of others' perspective and research on the problem
- presentation of your own perspective on the problem (e.g. assessment of when techniques you've researched are appropriate)

- conclusion based on evidence presented

Presentations: your presentation will be graded on the same criteria as your paper or program writeup; you will have to condense your material into 10 minutes. I *strongly suggest* that you rehearse your presentation at least twice before giving it. Ten minutes is not very long and you will need to make the best possible use of the time. Your papers and writeups need not be finished at the time of your presentation. Indeed, you should expect to take feedback from the class and incorporate your reactions to it into your final product.

More detailed information about grading will be provided after break.

A few ideas for projects

Here are some ideas for projects:

- Problem 7.26 suggests several programming project ideas.
- Describe in detail and provide examples of the use of a concurrent programming language that we have not discussed in class.
- Research nonblocking and wait-free algorithms (see p. 138 of the text for references).
- In general, the Historical Notes at the end of each chapter are a rich source of ideas for papers.