CSc 422/522 — Homework 4

due Tuesday, April 18

The first four problems are worth 10 points each. The last problem is worth 20 points. As usual, graduate students are to solve all problems (60 points), and undergraduates are to solve any combination that adds to 40 points.

1. MPD book, exercise 7.7 — a time-server process.

2. MPD book, exercise 7.12, part (a) — the Roller Coaster problem. If you wish, you may do part (c) instead, using C and MPI, Java and sockets, or SR.

3. MPD book, exercise 7.15 — the Stable Marriage problem. If you wish, you may do part (b) instead, using C and MPI, Java and sockets, or SR.

4. MPD book, exercise 7.16. (This problem is sometimes called the Welfare Crook. Assume that the arrays are strings instead of integers. If the first array is people working at IBM Yorktown, the second is students at Columbia, and the third is people on welfare in New York City, something peculiar is going on if a person is on all three lists!)

5. Write a distributed parallel program to solve *either* of the problems below. You may write your program in SR, in C using the MPI library, or in Java using sockets. Develop your program on Lectura, and then run your program on Par with 4 worker processes to see what kind of speedup it gets. You do not have to beat on your program to get the best possible speedup (although you are welcome to do so :-). It is sufficient to write what you think is a good program.

Hand in a commented listing of your program together with a brief (one page) overview of your approach and summary of the performance data. We are not going to schedule time on Par for this program. Just run a few experiments to get a feel for how your program performs.

Also use the `turnin` program to submit a copy of your program. The assignment name is `hw4.problem5`. Just turn in the source file(s).

Problem 1. Determine the number of words in `/usr/dict/words` that have unique letters, namely the number of words in which no letter appears more than once. Print the number of such words. Also print all the longest words. The dictionary file contains 25143 words, but consider only those that start with a *lower-case* letter. Ignore words that begin with numbers or capitals.

Use the manager/workers paradigm (distributed bag of tasks) to solve this problem. I suggest that you divide the work into 26 tasks, one for each letter of the alphabet.

The process may share only the values of command-line arguments or other information that is not changed by the workers. You may read the dictionary file into shared variables at the start of the program. (If you don't, it will be hard to get any speedup.)

Problem 2. Implement a distributed parallel version of your N^2 program for solving the N-body problem. Section 11.2.3 describes three different approaches you could use. Pick one of them. The process may share only the values of command-line arguments or other information that is not changed by the workers. They must use message passing to exchange information with each other.