Programming Assignment 2 – Multi-threaded Matrix Multiplication

Assigned: Tuesday, February 22, 2011
Due: Tuesday, March 8, 2011, 1pm (as instructed below)

Assignment Description

You are to complete the "Matrix Multiplication Project" which is described in the required textbook *Operating System Concepts (8th Edition)* by Silberschatz et al. on pages 178-181. The project asks you to implement a multi-threaded way of computing multiplication of two matrices.

Important Notes

The followings requirements in addition or in replacement of what is stated in the description in the textbook:

- Though the project description in the textbook provides Win32 API in addition to the pthread API, you are not allowed to use the Win32 API to develop your code.
- Your code will be tested and graded on the UNIX machines at the ECC Lab.
- Name your executable program as "`multiplier`"
- Your program will be given two input text files as arguments. In each text file there will be a matrix. You program will perform the multi-threaded multiplication (as detail in the description in the textbook), and then print out the multiplication of the two matrices to the screen. The format of the command line arguments will be as follows:

  ```
  "$myprompt> ./multiplier matrix1.txt matrix2.txt"
  ```
- The format of an input text file containing a matrix with M rows and N columns will be as follows:

  ```
  <M>
  <N>
  <1st row, 1st col> <1st row, 2nd col> <1st row, 3rd col> ... <1st row, Nth col>
  <2nd row, 1st col> <2nd row, 2nd col> <2nd row, 3rd col> ... <2nd row, Nth col>
  ...
  ...
  ...
  <Mth row, 1st col> <Mth row, 2nd col> <Mth row, 3rd col> ... <Mth row, Nth col>
  ```

  For example, a 3X4 matrix with increasing element values will be expressed as follows in the input files:

  ```
  3
  4
  0 1 2 3
  4 5 6 7
  8 9 10 11
  ```

  Your program must print out the result of the multiplication in the above format as well.

Submission Requirements

A **`makefile`** is required. All files in your submission will be copied to the same directory, therefore do not include any paths in your **`makefile`**. The **`makefile`** should include all dependencies that build your program. If a library is included, your **`makefile`** should also build the library.

To make this clear: ***do not hand in any binary or object code files***. All that is required is your source code, a **`makefile`**, and other necessary files as stated in the assignment description. Do test your project by copying the source code only into an empty directory and then compile it by entering the command **`make`**. Your code *must* compile without any errors or warnings and run properly under the Linux system used in the ECC lab (check their Web site for hours). You may develop and test your programs on your own UNIX machine, but it is your responsibility to ensure that they also work properly on the Linux installation of ECC, under the default ECC Lab shell. There will be a heavy penalty if they don't. The specific programming language you use is your choice, but your **`makefile`** has to be compiling your source code and producing a program that is executable in the Linux installation of ECC.

Your source code will be assessed and marked. Commenting is definitely required in your source code.

Go through the following steps to make your submission file ready to submit:

1. Go to your home directory and make a folder named with your name in format LastName_FirstName.
2. Put your source code and all other necessary files in this folder
3. In your home directory type and enter:
   `tar -cvf <`LastName_FirstName**`>.tar <`**LastName_FirstName**`>`**
4. Finally, type and enter:
   `gzip <`LastName_FirstName**`>.tar`**

These above steps will yield a new file named **`<`**LastName_FirstName**`>.tar.gz`** in your home directory.
To submit your file, log in to WebCampus and select the CS 446/646 course. From the "Course Tools" panel on left go to 'Assignments" and choose "Programming Assignment 2". Attach your submission file and submit.

If you have any problems please e-mail the instructor or the TA. ***Submissions including multiple files or sent by e-mail will NOT be evaluated.*** Do not turn in printouts in any form. Late assignments will be marked down according to the late policy published on the course Web page.

Good luck!