Homework 5

Tom Kelliher, CS 325

30 points, due Mar. 30, 2012

The Assignment

The "Selling Cells" problem from the Association for Computing Machinery's 2010 Mid-Atlantic Regional Programming Contest accompanies this assignment sheet, along with the test data used by the contest judges. `sellingCells.in` was the test input data and `sellingCells.out` was the expected output.

You are to write a Pthreads program in C that solves the problem using the Monte Carlo method. You will need to add one line to the beginning of the test input data file. This line will contain two integer values. The first value will specify the number of worker threads, w, to create, with $1 \leq w \leq 8$. The second value will specify the *total* number of random points, p, to be generated. You may assume that $w|p$. For example:

```
4 1000000
```

Each of the four worker threads will generate and test 250,000 random points.

The main thread should read its input data from `stdin`, initialize state for the first simulation, and create the worker threads. This set of worker threads is to remain in existence for the entire set of simulations. After each simulation, the main thread will print the result to `stdout`. Mutexes and condition variables will be necessary to coordinate access to shared variables and to coordinate the alternating operation of the main thread and the worker threads — the main thread sets up a simulation, gets the worker threads going, and goes to sleep. Once each worker thread completes its tasks, it goes to sleep, with the last worker thread going to sleep waking the main thread. This alternation is repeated for each simulation.

Note that `rand()` is not thread-safe (refer to the man page for `rand()`. Instead, you should use `drand48_r`, with each thread using its own `buffer` (refer to the man page for `drand48_r`).

Documentation Requirements

Minimally, your program must contain your name and an overview at its top similar to this:

```
/**********************************************************************
 * Tom Kelliher, Goucher College.
 *
 * mandelbrot.c --- Serial program to generate a PPM image
 *                  representation of the Mandelbrot set for some
 *                  rectangular portion of the complex plane.
 *
 * Due to the use of sqrtf(), the math library needs to be compiled-in:
 *
 *    gcc -lm -o mandelbrot mandelbrot.c
 *
 * The PPM image is written to stdout.  It will be HUGE, so it is
 * advisable to pipe the output to pnmtojpeg (on a Linux system)
 * and redirect the filter's output to a file:
 *
 *    ./mandelbrot | pnmtojpeg > image.jpg
 *
 * The pack()/unpack() routines, unnecessary in a serial environment,
 * _might_ be useful in a message passing environment, but that is a
 * claim that ought to be tested.
 ********************************************************************/
```

If the program you submit for a grade does not work, I will need further documentation within the program itself in order to assign partial credit. (You risk receiving **no** partial credit if you do not provide sufficient documentation.) One component of this documentation should be an analysis, to the best of your ability, of why the program is not working.

Compiling and Running Your Program

This problem requires the use of the Pthreads library:

```
gcc -o sellingCells -lpthread sellingCells.c
```

Your program should take its input data from `stdin`:

```
./sellingCells < sellingCells.in
```

Seeking Assistance

I strongly encourage you to begin this assignment as soon as it is issued and to come to my office if you need assistance. If you email me to describe a problem, describe the problem carefully and attach your source code. If extensive debugging will be required to determine the cause of the problem, I will let you know that you will need to visit me in person to resolve the problem. (In other words, I will not debug your program for you.)

Submitting Your Assignment

By the start of class on the 30th, send me your source code. Debugging messages in your source code should be commented-out. Late work will be penalized 15% per day; Saturday does not count, but Sunday does. (An unforeseen circumstance preventing you from finishing an assignment on time, while rare, may warrant a request for a deadline extension. Any such request must be made in writing, state the length of the extension requested, explain the nature of the unforeseen circumstance, include a strong justification for the requested extension, and be made at least 24 hours in advance of the assignment deadline. The strength of your justification and the nature of your unforeseen circumstance will determine whether or not the request is granted.)