

# Homework 6

Tom Kelliher, CS 325

30 points, due Apr. 6, 2012

## The Assignment

*“It’s déjà vu all over again”* — Yogi Berra

The objective of this assignment is to use Pthreads to parallelize the serial n-body solver accompanying assignment 3. Remember, sample data was provided with that assignment.

Your solution should work for up to 8 bodies. The main thread will read the data file, initialize the `current` array, and print out the initial state of the system. The main thread will then create `n` worker threads, one per body. The worker threads will perform the solver’s work. Each worker will use a local scalar `next` to compute the next state information for its body. After performing this computation, the worker threads will wait at a barrier before updating the global `current` array with the `next` data for their bodies. This update will have to be serialized by a mutex.

Using `n join()`s, the main thread will wait for the worker threads to perform all the computations and exit. Once all the worker threads have exited, the main thread prints out the final state of the system.

## 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 pnmtjpeg (on a Linux system)
* and redirect the filter’s output to a file:
*
*   ./mandelbrot | pnmtjpeg > 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

Due to the use of `sqrt()`, the math library will need to be linked into your program:

```
gcc -lm -lpthread -o parNbody parNbody.c
```

You will also need to include the math library's header file, `math.h`. The program will expect a command line argument corresponding to the name of the data file:

```
./parNbody nbodyData
```

The format of this data file is documented in the serial n-body program. Here's an example of a complete data file:

```
6 100000 0.01
1.0E4 0.0 0.0 0.0 0.0
1.0E4 100.0 0.0 0.0 0.0
1.0E4 -100.0 0.0 0.0 0.0
1.0E4 0.0 100.0 0.0 0.0
1.0E4 0.0 -100.0 0.0 0.0
1.0E4 100.0 100.0 0.0 0.0
```

## Barrier Pseudo-Code

```
/* Global, shared variables. */
num_threads = n;
waiting_threads = 0;
phase = 0;
mutex m;
condition c;

lock(m);
lphase = phase; /* lphase is local, private. */
waiting_threads++;
if (waiting_threads == num_threads)
{
    waiting_threads = 0;
    phase = 1 - phase;
    broadcast(c);
}
else
    while (lphase == phase)
        wait(c);
unlock(m);
```

## 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 6th, send me your source code. Your parallel n-body program should produce **exactly** the same output as the serial n-body program. You should anticipate that I will test your program with several different data files. 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.)