CS311 - FA13: Happy Primes

Trevor Bramwell

November 24, 2013

1 Design

The first step I will take in this assignment will be to convert my sieve of eratosthenes to use a bitmap instead of an array. I will then refactor the main loop to be a single function which will become the thread function. I will then write functions for dealing with a bitmap, specifically set_bit, and clear_bit.

I will be reusing as much code from the first implementation of the sieve as possible, and will also try to factor out as much shared functionality as possible to a shared library (ex. compute happiness).

After each process or thread has computed the primailty of their section, they will increment a integer. Once that integer reaches the number of threads or process, the bitmap will be copied, and they will be allowed to continue onto computing happiness. This synchronization is needed due to happy numbers using the same bitmap.

1.1 Threads & Implicit Sharing

Because a bitmap backed is a single number, it would be extremely wasteful to lock the number. Instead, I will setup *MAX_INT (or MAX_LONG)/num threads* , partitions across the number. Two locked partions will be needed for working on the section of the map they represent. This includes the extra two partions at 0, and MAX_INT (or MAX_LONG). Each *num threads* section will have a condition variable to signal sections adjacent.

1.2 Processes & Shared Memory

The same approach will be used in this part, but with the exception of mutexs and condition variables being replaced by semaphors of size 2. The implicit memory, or global variable, will be replaced by the bitmap being stored in the shared memory.

1.3 Happy Numbers

Before moving onto *Happy Numbers* I will check to make sure I have all the primes with multiple numbers of threads and processes. Once I am confident with my output of primes, I will move on to checking for happy numbers. Checking for number happiness will be the same as generating primes, with the exception that if a number is not happy, it is marked as not prime. Since there is a distinct transition point between prime generation and happiness, a copy of the bitmap will be made and saved for later output.

1.4 Deviation

2 Questions

What do you think the main point of this assignment is?

> The main point of this assignment is for us to gain experience with race conditions, use pthreads, shared memory, and synchronization constructs. Basically have first hand experience writing concurrent code.

How did you ensure your solution was correct? Testing details, for instance. This section should be very thorough.

What did you learn?

3 Work Log