CSC 325 Lab 10

threads

Table of Contents

This is an **individual** lab which may be completed on **either** VM.

1. Lab Goals

1. Gain more experience in C system programming.
2. Understand basic usage of POSIX pthreads.

2. Lab Overview

Processes are necessary abstractions in multiprogramming systems, which support multiple processes existing in the system at the same time. The process abstraction is used by the OS to keep track of individual instances of programs running in the system, and to manage their use of system resources.

One way to speed up the execution of a single process is to decompose it into lightweight, independent execution flows called threads. While each thread has its own private allocation of call stack memory, all threads share the program data, instructions, and the heap allocated to the multithreaded process.

3. Handy References

- Read and understand textbook chapter 14.2 Threads
- View the libc manual pages from the command-line:

```
$ man 7 pthreads
$ man 3 pthread_create
$ man 3 pthread_join
```

4. Lab Starting Point Code

4.1. Getting the Starting Point Code

Download the starter files from Canvas:

```
Makefile threads.c
```

4.2. Starting Point Code

The files included are:

- `Makefile`: a complete Makefile to build the threads application
- `threads.c`: starter code for the threads application

5. Lab Requirements

- Your application shall require 2 command line parameters:
    - n – the highest candidate prime to search for
    - t – the number of threads to use concurrently
- Your thread function takes a single int parameter n and determines if n is prime. If n is prime, output using the following format:

    2 is prime
    3 is prime
    5 is prime
    7 is prime
    11 is prime
    …

- Your application shall create t threads and wait for all t threads to complete, then it should create another t threads and wait for all of them to complete, etc.
- Compile and link your source file by typing
    - `make`
- Run your application by typing
    - `./threads <highest_candidate> <num_threads>`
- Select a value of highest_candidate that produces a runtime of a few seconds. Then try different values of num_threads. What is the optimal value of num_threads? Submit a brief discussion of your results.

6. Tips

Read Chapter 14.2.

7. Submitting your Lab

7.1 What to submit

A folder containing your completed `threads.c` source file, the Makefile, a brief discussion of your results, and a screenshot.

The folder must also include a screenshot of the output from:

```
make clean; make
./threads 100 4
```

Your application must compile, assemble, and link without warnings or errors.

Zip your folder and submit it to Canvas.

7. Example usage

```
$ ./threads 11 5
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
time is 0.000194 seconds
```