

Programming Assignment I: Pthreads Programming

by Prof. Yi-Ping You

The purpose of this assignment is to familiarize yourself with Pthreads programming.

1 Problem Statement

Suppose we toss darts randomly at a square dartboard, whose bullseye is at the origin, and whose sides are two feet in length. Suppose also that there is a circle inscribed in the square dartboard. The radius of the circle is one foot, and its area is π square feet. If the points that are hit by the darts are uniformly distributed (and we always hit the square), then the number of darts that hit inside the circle should approximately satisfy the equation

$$\frac{\text{number in circle}}{\text{total number of tosses}} = \frac{\pi}{4},$$

since the ratio of the area of the circle to the area of the square is $\pi/4$.

We can use this formula to estimate the value of π with a random number generator:

```
number_in_circle = 0;
for (toss = 0; toss < number_of_tosses; toss++) {
    x = random double between -1 and 1;
    y = random double between -1 and 1;
    distance_squared = x*x + y*y;
    if (distance_squared <= 1)
        number_in_circle++;
}
pi_estimate = 4*number_in_circle/((double) number_of_tosses);
```

This is called a “Monte Carlo” method, since it uses randomness (the dart tosses). Write a Pthreads program that uses a Monte Carlo method to estimate π . The main thread should read in the total number of tosses and print the estimate. You may want to use `long long ints` for the number of hits in the circle and the number of tosses, since both may have to be very large to get a reasonable estimate of π .

2 Requirement

- Your submitted solution contains only one source file, named `pi.c` (in C) or `pi.cpp` (in C++).
- Your program takes two command-line arguments, which indicate the number of CPU cores and the number of tosses.
- Your program should be scalable.

3 Grading Policies

- Correctness (70%): Your parallelized program should output an acceptable π value (e.g., 3.14XXXX with at least 10 million tosses). In addition, your program should run faster than the original (serial) program.
- Scalability (15%): We will test your program on 2 or 4 (or higher) CPU cores with corresponding environments. Your program is expected to be scalarable.
- Performance (15%): You will be ranked with other classmates by the program execution time. The score will be based on the rank; that is, the best time performance will get all 15%.

4 Evaluation Platform

Your program should be able to run on UNIX-like OS platforms. We will test your program on the workstations (dedicated for this course) of CSCC, NCTU. You can access these workstations by ssh'ing into `pp1.cs.nctu.edu.tw`–`pp4.cs.nctu.edu.tw` with your CS accounts.

5 Submission

Be sure to upload your zipped source codes, which includes no folder, to e-Campus system by the due date and name your file as “**HW1_XXXXXXX.zip**”, where **XXXXXXX** is your student ID.

Due Date: 23:59, November 3, Friday, 2017

6 References

- <https://computing.llnl.gov/tutorials/pthreads/#PthreadsAPI>
- <http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>