CS4540 – Operating Systems

Assignment 4. Process Synchronization

Due November 20

An interesting way of calculating π is to use a technique known as Monte Carlo simulation.

This algorithm works as follows:

a) Suppose you have a circle inscribed within a square, as shown in Figure 1. The radius of this circle is 1.

b) Generate a series of random points as simple (x, y) coordinates. These points must fall within the Cartesian coordinates that bound the square. Of the total number of random points that are generated, some will occur within the circle.

c) Estimate π by performing the following calculation:

$$\pi = 4 \times (\text{number of points in circle}) / / (\text{total number of points})$$

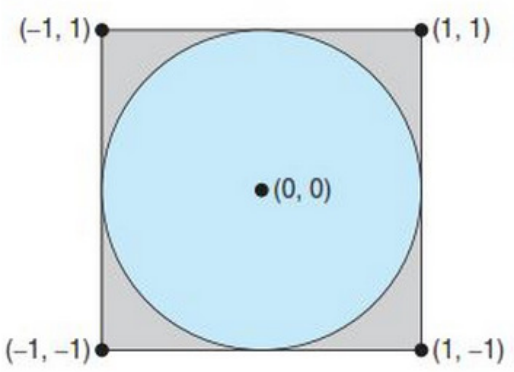

Figure 1. Calculating π with the Monte Carlo technique.

Write a multithreaded version of this algorithm that creates five threads, each of which: (i) generates a predefined number of random points, (ii) determines if the points fall within the circle, and (iii) counts the number of points that fall within the circle.

Each thread will have to update the shared global count *circleCount* of all points that fall within the circle. Protect against race conditions on updates to *circleCount* by using mutex locks.

When all thread have exited, the parent thread will calculate and output the estimated value of π.

Extra Credit (10%)

Print (with proper labels) the estimated value of π for *each* thread before it exits. This will allow us to see different estimates of π: (i) as calculated individually by each thread, and (ii) as estimated by all threads together.

Requirements and Hints

1) Use the constant NR_PTS for the total number of points; set it to 10,000,000.

2) Use the constant NR_THREADS determining the number of threads used to generate random points and count the number of points that occur within the circle.

3) Note that mutex locks are available in Pthreads.

4) You are required to use makefiles (starting with Assignment 3).
Include in makefile an entry for cleaning up the directory.
A brief makefile tutorial is at: http://mrbook.org/tutorials/make/ (for C programs on Ubuntu, replace "`.cpp`" with "`.c`" and replace "`g++`" with "`gcc`").

Background

1) *Monte Carlo methods* (or *Monte Carlo experiments* [or *Monte Carlo simulations*]) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results; typically one runs simulations many times over in order to obtain the distribution of an unknown probabilistic entity.

   [… A] *simulation* is a fictitious representation of reality, a *Monte Carlo method* is a technique that can be used to solve a mathematical or statistical problem, and a *Monte Carlo simulation* uses repeated sampling to determine the properties of some phenomenon (or behavior). […D]istinctions [between these terms] are not always easy to maintain.

   [http://en.wikipedia.org/wiki/Monte_Carlo_method]

2) The formula used in (c) is derived from the following facts:

   i) (circle_area) / (square_area) $=_{approx.}$ (number of points in circle) / (number of points in square)

   ii) For r = 1, (circle_area) / (square_area) = $\pi r^2 / (2r)^2 = \pi / 4$. Hence: $\pi = 4 *$ (circle_area) / (square_area)