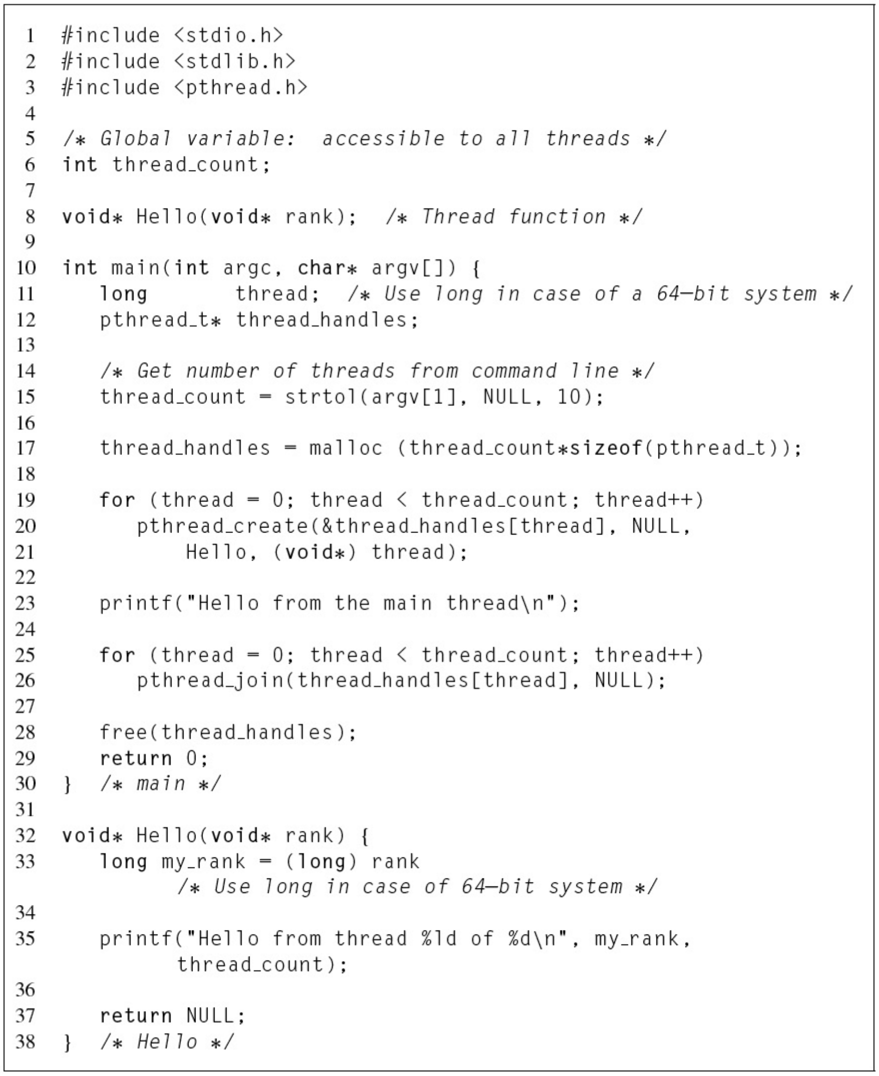
School of Artificial Intelligence and Advanced Computing
Xi'an Jiaotong-Liverpool University

DTS202TC Foundation of Parallel Computing

Lab 3: Profiling, Shared-Memory Programming with Pthreads

Pthreads Hello World

Type the following source code manually (no copy and paste) in a pth_hello.c file.



```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

/* Global variable:  accessible to all threads */
int thread_count;

void* Hello(void* rank);  /* Thread function */

int main(int argc, char* argv[]) {
   long       thread;  /* Use long in case of a 64-bit system */
   pthread_t* thread_handles;

   /* Get number of threads from command line */
   thread_count = strtol(argv[1], NULL, 10);

   thread_handles = malloc (thread_count*sizeof(pthread_t));

   for (thread = 0; thread < thread_count; thread++)
      pthread_create(&thread_handles[thread], NULL,
          Hello, (void*) thread);

   printf("Hello from the main thread\n");

   for (thread = 0; thread < thread_count; thread++)
      pthread_join(thread_handles[thread], NULL);

   free(thread_handles);
   return 0;
}  /* main */

void* Hello(void* rank) {
   long my_rank = (long) rank
         /* Use long in case of 64-bit system */

   printf("Hello from thread %ld of %d\n", my_rank,
         thread_count);

   return NULL;
}  /* Hello */
```

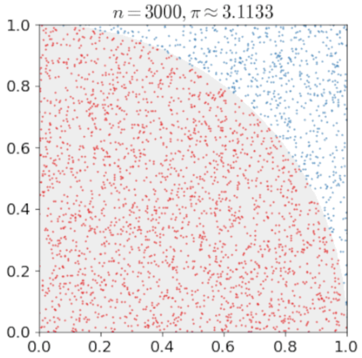
Compile the code like an ordinary C program, with linking in the Pthreads library:

```
gcc -g -Wall -o pth_hello pth_hello.c -lpthread
```

Run the program:

```
./pth_hello <number of threads>
```

Monte-Carlo Estimation with Pthreads



Suppose we toss darts randomly at a square dartboard, whose bullseye is at the origin, and whose sides are 2 feet in length. Suppose also that there's a circle inscribed in the square dartboard. The radius of the circle is 1 foot, and it's 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

$$\frac{\text{number in circle}}{\text{total number of tosses}} = \frac{\pi}{4} \tag{1}$$

This is called a "Monte Carlo" method, since it uses randomness (the dart tosses), more about Monte-Carlo method can be found at https://en.wikipedia.org/wiki/Monte_Carlo_method. Instead of estimating π, write a Pthreads program that uses a Monte Carlo method to estimate the shaded area in Figure 1. 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 and the number of tosses, since both may have to be very large to get a reasonable estimation.

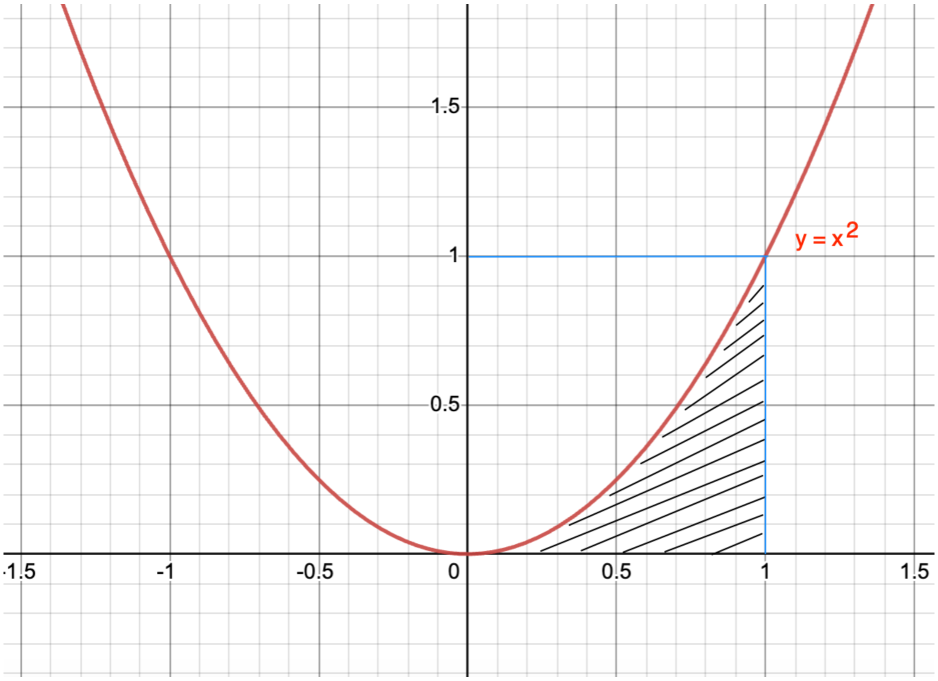


Figure 1: Estimate the shaded area

Reflection

What challenge did you face and how did you overcome it?