HW2: Pthreads 1

Rahul Kumar

October 18, 2025

1 Introduction

This document explains the implementation of a multithreaded C program that generates random numbers and searches for prime numbers using POSIX threads. Each thread performs a naive primality test and synchronizes with other threads using a custom barrier.

2 Source Code

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

// --------------------------------
// Configurable Parameters
// --------------------------------
#define NUM_THREADS 100
#define MAX_TRIES 10
#define MAXVAL 99999999999999999UL

// --------------------------------
// Barrier Structure and Functions
// --------------------------------
struct barrier {
    pthread_mutex_t lock;
    pthread_cond_t cond;
    int num_threads;
    int count;
};

void barrier_init(struct barrier* b, int nt) {
    pthread_mutex_init(&b->lock, NULL);
    pthread_cond_init(&b->cond, NULL);
```

```
27      b->count = 0;
28      b->num_threads = nt;
29  }
30
31  void barrier_destroy(struct barrier* b) {
32      pthread_mutex_destroy(&b->lock);
33      pthread_cond_destroy(&b->cond);
34  }
35
36  void barrier_wait(struct barrier* b) {
37      pthread_mutex_lock(&b->lock);
38      b->count++;
39
40      if (b->count == b->num_threads) {
41          pthread_cond_broadcast(&b->cond);
42      } else {
43          while (b->count < b->num_threads) {
44              pthread_cond_wait(&b->cond, &b->lock);
45          }
46      }
47      pthread_mutex_unlock(&b->lock);
48  }
49
50  // --------------------------------
51  // Thread Argument Structure
52  // --------------------------------
53  struct thread_args {
54      int rank;
55      struct barrier* barrier;
56  };
57
58  // --------------------------------
59  // Primality Test (Brute Force)
60  // --------------------------------
61  int is_prime(unsigned long int n) {
62      if (n < 2) return 0;
63      if (n % 2 == 0 && n != 2) return 0;
64
65      unsigned long int limit = (unsigned long int)sqrt((long double)n
        );
66      for (unsigned long int i = 3; i <= limit; i += 2) {
67          if (n % i == 0) return 0;
68      }
69      return 1;
70  }
71
72  // --------------------------------
```

```
// Thread Function
// -------------------------------
void* start(void* x) {
    struct thread_args* args = (struct thread_args*) x;
    unsigned long int to_test = 0;
    int found = 0;

    for (int i = 0; i < MAX_TRIES; i++) {
        // Generate a random unsigned long int
        to_test = ((unsigned long int)rand() << 32) | rand();
        to_test = to_test % MAXVAL;

        if (is_prime(to_test)) {
            found = 1;
            break;
        }
    }

    // Barrier synchronization before reporting results
    barrier_wait(args->barrier);

    if (found) {
        printf("thread %d reports that %lu is prime\n", args->rank,
           to_test);
        pthread_exit((void*)to_test);
    } else {
        printf("thread %d reports no prime\n", args->rank);
        pthread_exit((void*)0);
    }
}

// -------------------------------
// Main Function
// -------------------------------
int main(int argc, char *argv[]) {
    pthread_t threads[NUM_THREADS];
    struct thread_args args[NUM_THREADS];
    struct barrier sb;

    // Seed the random number generator
    srand(time(NULL));

    barrier_init(&sb, NUM_THREADS);

    for (int i = 0; i < NUM_THREADS; i++) {
        args[i].rank = i;
        args[i].barrier = &sb;
```

```
        pthread_create(&threads[i], NULL, start, &args[i]);
    }

    // Collect thread return values
    for (int i = 0; i < NUM_THREADS; i++) {
        void* retval;
        pthread_join(threads[i], &retval);
        // Optionally store retval for later
    }

    barrier_destroy(&sb);
    return 0;
}
```

3 Compilation and Results

The program was compiled using the `gcc` compiler with optimization and warning flags enabled. The `-Wall` flag enables all common compiler warnings, while `-O3` applies a high level of optimization for performance. The `-lpthread` and `-lm` flags link the POSIX threads and math libraries, respectively.

```
gcc -Wall -O3 pthread_1.c -o pthread_1.ex -lpthread -lm.
.\pthread_1.ex
```

Log Contents

```
thread 6 reports that 93403885782262703 is prime
thread 81 reports no prime
thread 4 reports no prime
thread 54 reports no prime
thread 8 reports no prime
thread 19 reports no prime
thread 0 reports that 60536618493578089 is prime
thread 86 reports that 79634737370222449 is prime
thread 22 reports no prime
thread 3 reports no prime
thread 65 reports no prime
thread 99 reports no prime
thread 18 reports no prime
thread 93 reports that 74829274336832471 is prime
thread 2 reports no prime
thread 76 reports no prime
thread 13 reports no prime
thread 33 reports no prime
thread 30 reports no prime
thread 88 reports no prime
```

```
thread 70 reports no prime
thread 94 reports no prime
thread 44 reports no prime
thread 32 reports no prime
thread 55 reports no prime
thread 7 reports no prime
thread 15 reports no prime
thread 60 reports that 32646353652468889 is prime
thread 17 reports that 5831460856650941 is prime
thread 66 reports that 64877380611958697 is prime
thread 97 reports no prime
thread 46 reports no prime
thread 83 reports no prime
thread 72 reports no prime
thread 23 reports no prime
thread 96 reports no prime
thread 56 reports that 1209450816962249 is prime
thread 25 reports that 80476545731465621 is prime
thread 92 reports no prime
thread 37 reports that 87520811199548899 is prime
thread 35 reports no prime
thread 20 reports that 63054376804144529 is prime
thread 16 reports no prime
thread 38 reports no prime
thread 90 reports no prime
thread 74 reports no prime
thread 89 reports that 20150120763659543 is prime
thread 98 reports no prime
thread 78 reports no prime
thread 61 reports no prime
thread 69 reports no prime
thread 49 reports no prime
thread 75 reports no prime
thread 64 reports no prime
thread 95 reports no prime
thread 53 reports no prime
thread 10 reports no prime
thread 62 reports no prime
thread 91 reports no prime
thread 39 reports no prime
thread 48 reports no prime
thread 57 reports that 74685894321656741 is prime
thread 84 reports no prime
thread 31 reports no prime
thread 26 reports no prime
thread 21 reports no prime
thread 28 reports no prime
```

```
thread 58 reports no prime
thread 5 reports no prime
thread 82 reports no prime
thread 47 reports no prime
thread 40 reports no prime
thread 85 reports no prime
thread 14 reports no prime
thread 68 reports no prime
thread 27 reports no prime
thread 42 reports that 38479059003928843 is prime
thread 24 reports no prime
thread 34 reports that 25593585467846651 is prime
thread 29 reports that 44541169914754451 is prime
thread 45 reports no prime
thread 1 reports no prime
thread 52 reports no prime
thread 36 reports that 1523468514736127 is prime
thread 41 reports no prime
thread 73 reports no prime
thread 9 reports no prime
thread 11 reports no prime
thread 12 reports no prime
thread 59 reports no prime
thread 87 reports that 2739679154009683 is prime
thread 77 reports no prime
thread 50 reports no prime
thread 79 reports that 12922952785916981 is prime
thread 51 reports no prime
thread 63 reports no prime
thread 67 reports no prime
thread 80 reports that 55026843631095493 is prime
thread 43 reports that 72164393696765723 is prime
thread 71 reports no prime
```