

HW3

7.15*:

這個修改使父執行緒可以在子執行緒計算斐波那契數列的同時就開始存取它們，而不需要等到整個計算完成

```
int main() {  
    printf("Enter the number of Fibonacci numbers to generate: ");  
    scanf("%d", &n);  
  
    pthread_t tid;  
    pthread_create(&tid, NULL, calculate_fibonacci, &n);  
  
    pthread_join(tid, NULL); // 等待子线程结束  
  
    // 父线程读取斐波那契数列  
    pthread_mutex_lock(&mutex); // 获取锁  
    printf("Fibonacci numbers:\n");  
    for (int i = 0; i < count; i++) {  
        printf("%d ", fib_numbers[i]);  
    }  
    printf("\n");  
    pthread_mutex_unlock(&mutex); // 释放锁  
  
    return 0;  
}
```

執行看到輸出結果

```
yu1un@DESKTOP-1TCLE4T:/mnt/c/Users/user/Desktop/作業系統/HW3$ ./fibonacci  
Enter the number of Fibonacci numbers to generate: 10  
Fibonacci numbers:  
0 1 1 2 3 5 8 13 21 34
```

7.17**:

修改後的程式創建了多個執行緒，每個執行緒生成隨機點計算在圓內的點的數量

```
int main() {
    pthread_t threads[NUM_THREADS];

    srand((unsigned int)time(NULL));

    for (int i = 0; i < NUM_THREADS; ++i) {
        pthread_create(&threads[i], NULL, generate_points, NULL);
    }

    for (int i = 0; i < NUM_THREADS; ++i) {
        pthread_join(threads[i], NULL);
    }

    double pi_estimate = 4.0 * ((double)points_in_circle / TOTAL_POINTS);
    printf("Estimate of Pi: %lf\n", pi_estimate);

    return 0;
}
```

執行看到輸出結果

```
yulun@DESKTOP-1TCLE4T:/mnt/c/Users/user/Desktop/作業系統/HW3$ ./monte_carlo_pi
Estimate of Pi: 3.140940
yulun@DESKTOP-1TCLE4T:/mnt/c/Users/user/Desktop/作業系統/HW3$ ./monte_carlo_pi
Estimate of Pi: 3.141324
```

8.32*:

執行看到輸出結果

```
yulun@DESKTOP-1TCLE4T:/mnt/c/Users/user/Desktop/作業系統/HW3$ ./farmers_on_bridge
前往北行的農民正在穿過橋樑。
前往北行的農民正在穿過橋樑。
前往南行的農民正在穿過橋樑。
前往北行的農民正在穿過橋樑。
前往南行的農民正在穿過橋樑。
前往北行的農民已經穿過橋樑。
前往北行的農民已經穿過橋樑。
前往南行的農民已經穿過橋樑。
前往南行的農民已經穿過橋樑。
前往北行的農民已經穿過橋樑。
█
```

9.28*:

執行看到輸出結果

```
yulun@DESKTOP-1TCLE4T:/mnt/c/Users/user/Desktop/作業系統/HW3$ gcc -o addresses addresses.c
yulun@DESKTOP-1TCLE4T:/mnt/c/Users/user/Desktop/作業系統/HW3$ ./addresses 19986
The address 19986 contains:
page number=4
offset=3602
```