系統環境：VMware, Linux, Ubuntu 18.04

- Problem 4.17

  利用一個子程序生成在座標平面上正方形內的隨機點(x, y)，並判斷隨機點與正方形中心點的距離，小於或等於 1/2 正方形邊長則代表該點位於方內圓中，子程序結束後由主程序計算由 Monte Carlo 定理得出的圓周率。

  程式碼：Source code/countPI.c

```
yuwei@ubuntu:~/Desktop$ gcc -Wall -pthread -o countPI countPI.c
yuwei@ubuntu:~/Desktop$ ./countPI
輸入隨機點數量 : 1
pi = 4.000000
總花費時間 : 0.000403 s
yuwei@ubuntu:~/Desktop$ ./countPI
輸入隨機點數量 : 109590004
pi = 3.141592
總花費時間 : 2.299934 s
```

- Problem 4.21

  利用一個子程序生成指定最大項的費式數列，子程序結束後由主程序輸出費式數列。

  程式碼：Source code/fibonacci.c

```
yuwei@ubuntu:~/Desktop$ gcc -Wall -pthread -o fibonacci fibonacci.c
yuwei@ubuntu:~/Desktop$ ./fibonacci
輸入費式數列最大項 : -1
請輸入 >= 0 的項
輸入費式數列最大項 : 0
費式數列 : 0
yuwei@ubuntu:~/Desktop$ ./fibonacci
輸入費式數列最大項 : 20
費式數列 : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987,
1597, 2584, 4181, 6765
```

- Problem 6.33

  與 Problem 4.17 類似，差別在於將次數均分於多個程序跑隨機點，並有一個程序跑無法將次數整除的次數，並且注意共享變數的讀寫控制，所有子程序結束後由主程序輸出圓周率。

  程式碼：Source code/multiCountPI.c

```
yuwei@ubuntu:~/Desktop$ gcc -Wall -pthread -o multiCountPI multiCountPI.c
yuwei@ubuntu:~/Desktop$ ./multiCountPI
輸入隨機點數量 : 109590004
pi = 3.141675
總花費時間 : 45.778034 s
yuwei@ubuntu:~/Desktop$
```