

HW3 Report

B10505047 邱郁喆

Q6.14 :

- a. 全域變數 `number_of_processes` 會在兩個函式 `allocate_process()` 跟 `release_process()` 中被修改，如果兩者同時執行即會造成 `race condition`。
- b. 在函式 `allocate_process()` 中，應該在判斷 `number_of_processes` 是否等於 `MAX_PROCESSES` 之前加上 `acquire(mutex);`，並且在 `if` 條件式中的 `return -1;` 這行之前以及 `else` 條件式中的 `++number_of_processes;` 這行之後都加上 `release(mutex);`。
在函式 `release_process()` 中，應該在 `--number_of_processes;` 這行之前加上 `acquire(mutex);`，並在這行之後加上 `release(mutex);`。

Q6.33 :

改寫 `hw2_Q1.c`，一開始使用 `pthread_mutex_init()` 初始化 `mutex`，讓後續各 `thread` 對共享變數進行上鎖保護，避免 `race condition`。接著透過 `pthread_create()` 建立 5 個 `threads`，每個 `thread` 會執行函數 `generatePoints()`，產生 1000 個 `random points` 並判斷每個點是否落在單位圓內。若在圓內，使用 `mutex` 加鎖後更新全域變數 `circlePoints`。所有 `threads` 執行完後，`main thread` 再計算 `pi` 值。

Compile :

在 linux 環境下執行

```
gcc hw3.c
```

```
./a.out
```