

# Operating System Homework 3 Report

Student ID: 0416025

Name: 呂翊愷

## Detailed description of the implementation:

(Number of threads, the purpose of those threads, how do you use mutex lock and semaphore...etc.)

HW3-1 :

總共有八個 thread，各四個分別做 RGB2grey 和 Convolve，即將原本呼叫 RGB2grey 函數把回傳值存到 pic\_grey 這個步驟分成四等份，四個 thread 同時去做計算，讓速度變快，在 Convolve 的部分也是如此。將原本呼叫 GaussianFilter 函數把回傳值存到 pic\_blur 這個步驟分成四等份，四個 thread 同時去做計算，讓速度變快。另外，使用 mutex 來預防 race condition，在 thread 呼叫的函數分別一開始加上 mutex\_lock，讓其他 thread 無法 access 所使用到的資料，在函數結束的時候再加上 mutex\_unlock 解鎖。

Hw3-2 :

總共有八個 thread，各四個分別做 RGB2grey 和 Convolve，即將原本呼叫 RGB2grey 函數把回傳值存到 pic\_grey 這個步驟分成四等份，四個 thread 同時去做計算，讓速度變快，在 Convolve 的部分也是如此。和 HW3-1 不同的是，將 Filter 變成 GxFilter 和 GyFilter 兩個，四個 thread 同時將兩者回傳值做平方根，讓速度變快。另外，使用 semaphore 來預防 race condition，在最後一個 thread 做 RGB2grey 的時候，呼叫 sem\_post，讓其他 thread 知道可以 access 資料了，而在第一個 thread 做 Convolve 的時候再加上 sem\_wait 讓 thread 知道需要等待前面的 thread 完成才能 access 資料。

## Your speed:

```
[> g++ -std=c++11 -pthread 0416025_hw3-1.cpp
[> sh Speed.sh
[Input a number of times to run './a.out' : 10

Run time:
    Finished once.
    Avg time: 903752 μs
[> ./MAE.out ANS/Blur1.bmp Blur1.bmp
MAE = 0
```

HW3-1:  $1544042 / 903752 = 1.708$

```
[> g++ -std=c++11 -pthread 0416025_hw3-2.cpp
[> sh Speed.sh
[Input a number of times to run './a.out' : 10

Run time:
    Finished once.
    Avg time: 840106  $\mu$ s
[> ./MAE.out ANS/Sobel1.bmp Sobel1.bmp
MAE = 0
```

HW3-2:  $1430390 / 840106 = 1.702$

### Problems encountered and solutions:

一開始在想要如何分配 thread 的工作時，苦惱了很久，原本想按照助教提供的方法，將每個步驟的 RGB 分開做，但是試了很久沒有結果，最後決定將 RGB2grey 和 Gaussian Filter 分成不同 thread。雖然沒有快很多，但已有 thread 的效果。此外，在做第二題的時候，不知道為什麼最後出來的答案，MAE 的結果都會是在 0~10 之間，最後發現是因為用來存 GxFilter 和 GyFilter 回傳值的變數的型態錯誤，導致存的結果有所誤差，將型態改為 unsigned char 即可解決。