

Operating System Homework 3 Report

Student ID: 0410137

Name: 劉家麟

Detailed description of the implementation:

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

Hw3-1: Blur #threads = 8

Lock0[3]
Lock1[3]
Lock2[3]
Lock3[3]

在第一題，我用的是 **mutex**。

我把圖片拆成 4 塊，並用鎖來控制 RGB2Grey 跟 Gaussian Filter 的先後順序。

一開始直接先宣告 4 個 thread 來做 RGB2Grey 的部分。

後來再宣告另外 4 個 thread 來做 Convolution，但因為 Convolution 用的 Filter 會影響到前後的區塊，所以我用鎖來控制 convolution 的開始。

每個區塊的開始須先滿足「前面那塊做完 RGB、自己做完 RGB、後面那塊開始做了部分 RGB」才能做 Convolution，lockx[0]&lockx[1]代表做完了，lock[2]則代表已經做了部分。

Hw3-2: Sobel #threads = 8

Lock0[2]
Lock1[2]
Lock2[2]
Lock3[2]

在第一題，我用的是 **semaphore**。

第二題其實是直接延續第一題的概念，一樣把圖片拆成 4 塊分別做 RGB2Grey 以及 Convolution(共 8 個 thread)，再用鎖來控制 RGB2Grey 跟 Gaussian Filter 的先後順序。

但因為第二題用 semaphore，所以就沒有宣告那麼多鎖，做完的部分一次可以允許 2 個 thread 進入，所以在一開始鎖 2 次後，做完直接一次釋放 2 個鎖，讓 convolution 在要求時，可以兩個 thread 都要求同一個做完的 lock。

Your speed:

每次執行結果都不太一樣，兩個 Speedup 大概都 1.5~2.0 之間。

```
> g++ -std=c++11 -pthread blur.cpp
> ./a.out
> ./MAE.out Blur1.bmp ../ANS/Blur1.bmp
MAE = 0
> ./MAE.out Blur2.bmp ../ANS/Blur2.bmp
MAE = 0
> ./MAE.out Blur3.bmp ../ANS/Blur3.bmp
MAE = 0
> ./MAE.out Blur4.bmp ../ANS/Blur4.bmp
MAE = 0
> ./MAE.out Blur5.bmp ../ANS/Blur5.bmp
MAE = 0
> sh Speed.sh
Input a number of times to run './a.out' : 10

Run time:
  Finished once.
  Avg time: 968798 μs
> g++ -std=c++11 -pthread sobel.cpp
> ./a.out
> ./MAE.out Sobel1.bmp ../ANS/Sobel1.bmp
MAE = 0
> ./MAE.out Sobel2.bmp ../ANS/Sobel2.bmp
MAE = 0
> ./MAE.out Sobel3.bmp ../ANS/Sobel3.bmp
MAE = 0
> ./MAE.out Sobel4.bmp ../ANS/Sobel4.bmp
MAE = 0
> ./MAE.out Sobel5.bmp ../ANS/Sobel5.bmp
MAE = 0
> sh Speed.sh
Input a number of times to run './a.out' : 10

Run time:
  Finished once.
  Avg time: 855185 μs
```

Problems encountered and solutions:

1. 一開始做 sync 的部分一直有問題，常常圖片時好時壞，本來一直以為是硬體 thread 不夠的問題，後來想一想，可能是要求鎖的部分沒有鎖好。
本來鎖在進 RGB2Grey 各自的 thread 之後才鎖，後來把鎖在宣告 thread 前就先鎖好，進入 thread 做完 RGB2Grey 才釋放後，output 就沒有問題了。
2. 一開始是先做好部分的 RGB2Grey 在開始做 convolution，結果加速超慢，只有 1.03 倍，怎麼樣都無法提升。後來就決定拆圖片再加上同時做，不過同時做還是會有 sync 的問題，有時候有的區塊沒做好。最後把鎖放對位置整體加速終於提升到 1.5 倍以上。