

Operating System Homework 3 Report

Student ID: 0413359

Name: 童柏勛

Detailed description of the implementation:

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

在 HW 3-1 中 我實作 mutex_lock 利用 pthread_mutex_t 宣告了兩個 mutexlock 也分別宣告兩個 thread 陣列 thread number 我測了滿多組數字的發現 24 是個比較快的速度 ($//20 = 1323709$, $25 = 1166220$, $24 = 1003134||975223$, $16 = 1109368$, $23 = 1887611$) 兩個 thread 陣列分別做 1.convert RGB image to grey image ---> 在 Grey function 內做 Grey function 內放置了 mutexlock 來取得使用權
另一個 thread 陣列放在 2.apply the Gaussian filter to the image ---> G_filter function 中 function 內一樣用 mutexlock 來控制

在 HW 3-2 中 我實作 semaphore 一開始在讀檔時 Gx Gy 分開讀 在實作時也是分開寫的 這裡 thread 的 semaphore size 我選的是 6 ($//5 = 979729$, $6 = 876140$, $8 = 1410740$, $4 = 871489$, $3 = 1328378$, $2 = 1630426$) 分別宣告兩個 sem_t sem_x 跟 sem_y 當 grey 完成後 Gx Gy 收到訊號後可以開始進行接下來的動作 當 grey function 做時 sem_post sem_x 和 sem_y 這樣我的

Gx 和 Gy sem_wait 就會接著進行動作

Your speed:

HW 3-1 在 linux2 測試

```
> g++ -std=c++11 -pthread 0413359_hw3-1.cpp
> sh Speed.sh
Input a number of times to run './a.out' : 10

Run time:
    Finished once.
    Avg time: 958756 µs
> ./a.out
> chmod +x MAE.out
> ./MAE.out ANS/Blur1.bmp Blur1.bmp
MAE = 0
> █
```

Baseline = 1544042

My time = 958756

Speed = 1.61046397

HW 3-1 在 linux2 測試

```
Blur4.bmp      input2.bmp      mask_Sobel.txt
> g++ -std=c++11 -pthread 0413359_hw3-2.cpp
> sh Speed.sh
Input a number of times to run './a.out' : 10

Run time:
    Finished once.
    Avg time: 837518 µs
> chmod +x MAE.out
> ./MAE.out ANS/Sobell.bmp Sobell.bmp
MAE = 0
> █
```

Baseline = 1430390

My time = 837518

Speed = 1.70789

Problems encountered and solutions:

比較常遇到的是有時候明明開的 thread 數量一樣 但卻會越跑越慢

我會重開工作站

再來是找 thread number 和 semaphore size 要試試滿多次的

第二題原本一開始 mask 一直是 7 左右 後來發現是 unsigned char

*pic_blur_x....等幾個宣告忘記 free 了 還有更改 $G_x * G_y \sqrt{}$ 那邊的

數字轉換才調整好