**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.)

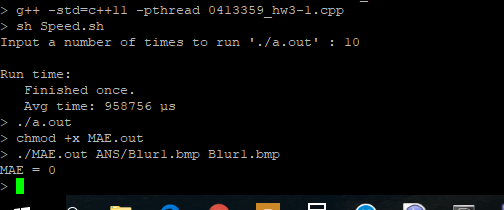
**在HW 3-1中 我實作mutex\_lock 利用pthread\_mutex\_t宣告了兩個mutexlock 也分別宣告兩個thread陣列 thread number我測了滿多組數字的發現 24是個比較快的速度 (//20 = 1323709, 25 = 1166220, 24 = 1003134||975223, 16 = 1109368, 23 = 1887611) 兩個thread陣列分別做1.convert RGB image to grey image ---> 在Grey function內做 Grey function內放置了mutexlock來取得使用權**

**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**

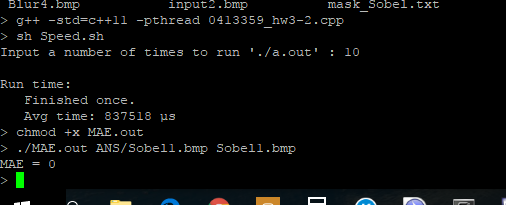


**Baseline = 1544042**

**My time = 958756**

**Speed = 1.61046397**

**HW 3-1 在linux2測試**



**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Gx \* Gy sqrt那邊的數字轉換才調整好**