

Operating System Homework 3 Report

Student ID: 0416308

Name: 林正偉

Detailed description of the implementation:

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

1) Number of threads:

第1題我用12個threads，因為工作站cpu是六核心，12執行緒

第2題我用12個threads

(其實我把threads的個數define成macro，所以可以從第19行直接改，然後還是可以跑出正確結果)

2) The purpose of those threads:

這兩題各有三大部份：轉成灰階 -> 濾波 & 展開3倍

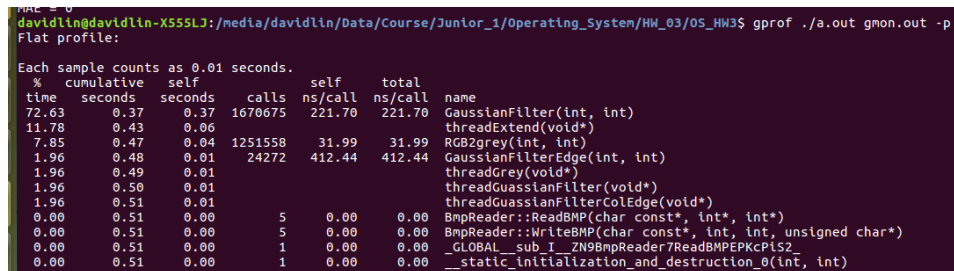
我把每一個部份切成threads個數個data，然後用threads平行運算，算完一個部份再換下一個部份

3) How do I use mutex lock and semaphore:

我用mutex lock and semaphore來等轉成灰階這個部份做完，然後才做濾波

4) How do I speed up my code:

用threads做加速大概快1.多倍（約670000）就極限了，所以我用gprof分析程式各個function總共用的時間，被呼叫的次數和每次呼叫所花的時間，發現濾波花的時間最久，而且每次呼叫花的時間很多（下圖）



Each sample counts as 0.01 seconds.					
%	cumulative	self	calls	self	total
time	seconds	seconds		ns/call	ns/call
72.63	0.37	0.37	1670675	221.70	221.70
11.78	0.43	0.06			
7.85	0.47	0.04	1251558	31.99	31.99
1.96	0.48	0.01	24272	412.44	412.44
1.96	0.49	0.01			
1.96	0.50	0.01			
1.96	0.51	0.01			
0.00	0.51	0.00	5	0.00	0.00
0.00	0.51	0.00	5	0.00	0.00
0.00	0.51	0.00	1	0.00	0.00
0.00	0.51	0.00	1	0.00	0.00

所以我把濾波的函式做unrolling，然後就快到3.多倍了（約355000）—原本是要用unrolling來加速，但後來得知filter的size會改變，所以就不能用了QQ 我現在做的只是減少濾波函式裡加減乘除的次數。

Your speed:

第1題：359504

第2題：325117

run1.sh和run2.sh是用來測正確性的東東（裡面放的是5個./MAE.out）

```

> g++ -std=c++11 -pthread 0416308_hw3-1.cpp
> ./Speed.sh
Input a number of times to run './a.out' : 10

Run time:
  Finished once.
  Avg time: 359504 µs
> ./run1.sh
MAE = 0
MAE = 0
MAE = 0
MAE = 0
MAE = 0
> g++ -std=c++11 -pthread 0416308_hw3-2.cpp
> ./Speed.sh
Input a number of times to run './a.out' : 10

Run time:
  Finished once.
  Avg time: 325117 µs
> ./run2.sh
MAE = 0
MAE = 0
MAE = 0
MAE = 0
MAE = 0

```

Problems encountered and solutions:

我已經放棄了XD 想不到要怎麼用threads加速了QQ 有試過跑完一些轉grey scale就開始濾波，但效果不是很好，而且版面看起來不好看，所以就沒用了。

工作站是六核十二執行緒，但我測的結果（下圖），我們好像只被分到兩核.....所以我一直在想我們要怎麼用threads快3倍以上，也就是原本的4倍速（如果被分到四核的話還可以理解啦）

```

> lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 2
On-line CPU(s) list:   0,1
Thread(s) per core:    1
Core(s) per socket:    2
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  26
Model name:             Intel(R) Xeon(R) CPU           X5675   @ 3.07GHz
Stepping:               4
CPU MHz:                3066.558
BogoMIPS:               6136.11
Hypervisor vendor:     VMware
Virtualization type:    full
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
L3 cache:               12288K
NUMA node0 CPU(s):     0,1
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse

```

還有第2題我覺得spec應該要附上image x和image y數值的限制（也就是當數值小於0時，其值為0，但大於255的還是維持原本的值），這個部份看了超久，因為跟第1題高斯濾波的限制條件不一樣.....，我是看結果的圖片只有右邊和下方的邊界有顯示，而推測出image x和image y數值的限制，覺得這個部份應該要在spec講一下