

Operating System Homework 4 Report

Student ID:0516072

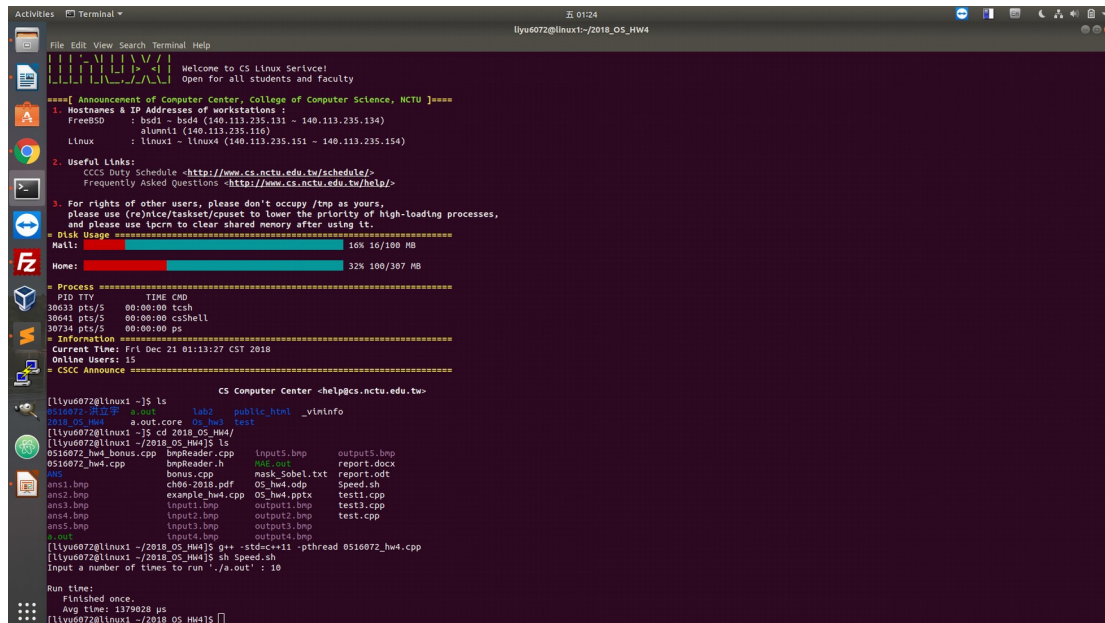
Name:洪立宇

Detailed description of the implementation:

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

在作業中我用了 2 個 thread 來分別做 mean filter 和 sobel filter，首先 sobel filter 必須再 mean filter 的九宮格做完後才有辦法做，所以在要做 sobel filter 必須要 wait 到 mean filter 的九宮格右下角那格做完才能做，然後將兩個 filter 透過 thread 同時來做，此外將所有的 function 和不必要的 for 回圈的東西丟到這兩個 filter function 中來減少的時間再 bonus 中則是將 mean filter 和 sobel filter 分成了上半個回圈 還有下半個回圈來同時做，所以總共用 4 個 thread，此時就要考慮到 mean filter 九宮格的右上和右下那格是否完成，來決定是否能做 sobel，所以一樣要 wait 到右上和右下的部份做完才能做。

Your speed: hw4



```
llyu6072@linux1:~/2018_OS_HW4
Welcome to CS Linux Service!
Open for all students and faculty

=====
1. Hostnames & IP addresses of workstations :
FreeBSD : bsdi ~ bsdi (140.113.235.131 ~ 140.113.235.134)
Linux   : alumn1 (140.113.235.110)
        : llyu1 ~ llyu4 (140.113.235.151 ~ 140.113.235.154)

2. Useful Links:
CCSC Duty Schedule <http://www.cs.nctu.edu.tw/schedule/>
Frequently Asked Questions <http://www.cs.nctu.edu.tw/help/>

3. For rights of other users, please don't occupy /tmp as yours,
please use (re)nice/taskset/cpuset to lower the priority of high-loading processes,
and please use (re)cpio to clear shared memory after using it.

Disk Usage =====
Mail: 10% 16/100 MB
Home: 32% 100/307 MB

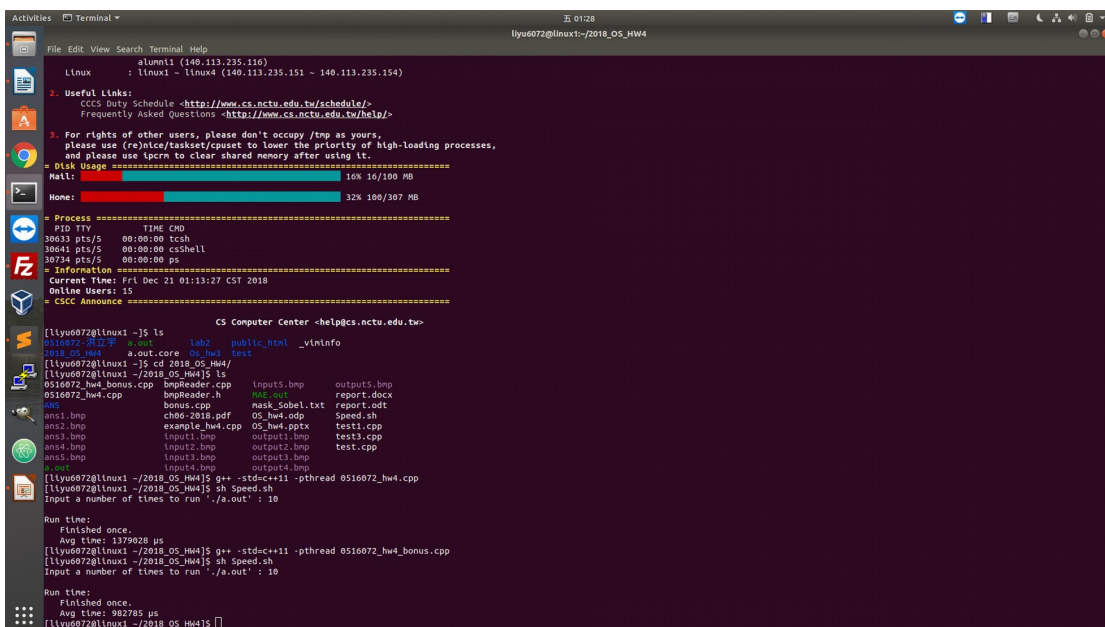
Process =====
PID TTY TIME CMD
30633 pts/5 00:00:00 tcsh
30641 pts/5 00:00:00 cshll
30734 pts/5 00:00:00 ps

Information =====
Current Time: Fri Dec 21 01:13:27 CST 2018
Online Users: 15
CSCC Announce =====
CS Computer Center <help@cs.nctu.edu.tw>

llyu6072@linux1 ~$ ls
a.out core os_hw4 test
llyu6072@linux1 ~$ cd 2018_OS_HW4/
llyu6072@linux1 ~/2018_OS_HW4$ ls
0516072_hw4_bonus.cpp bmpReader.cpp input5.bmp output5.bmp
0516072_hw4.cpp bmpReader.h mask_Sobel.txt report.odt
aks bonus.cpp OS_hw4.odp Speed.sh
ans1.bmp ch06-2018.pdf OS_hw4.odp test1.cpp
ans2.bmp example_hw4.cpp OS_hw4.pptx test3.cpp
ans3.bmp input1.bmp output1.bmp test.cpp
ans4.bmp input2.bmp output2.bmp
ans5.bmp input3.bmp output3.bmp
a.out input4.bmp output4.bmp
llyu6072@linux1 ~/2018_OS_HW4$ g++ -std=c++11 -pthread 0516072_hw4.cpp
llyu6072@linux1 ~/2018_OS_HW4$ sh Speed.sh
Input a number of times to run './a.out' : 10

Run time:
Finished once.
Avg time: 1379028 μs
llyu6072@linux1 ~/2018_OS_HW4$
```

hw4_bonus



```
llyu6072@linux1:~/2018_OS_HW4
Welcome to CS Linux Service!
Open for all students and faculty

=====
1. Hostnames & IP addresses of workstations :
FreeBSD : bsdi ~ bsdi (140.113.235.131 ~ 140.113.235.134)
Linux   : alumn1 (140.113.235.110)
        : llyu1 ~ llyu4 (140.113.235.151 ~ 140.113.235.154)

2. Useful Links:
CCSC Duty Schedule <http://www.cs.nctu.edu.tw/schedule/>
Frequently Asked Questions <http://www.cs.nctu.edu.tw/help/>

3. For rights of other users, please don't occupy /tmp as yours,
please use (re)nice/taskset/cpuset to lower the priority of high-loading processes,
and please use (re)cpio to clear shared memory after using it.

Disk Usage =====
Mail: 10% 16/100 MB
Home: 32% 100/307 MB

Process =====
PID TTY TIME CMD
30633 pts/5 00:00:00 tcsh
30641 pts/5 00:00:00 cshll
30734 pts/5 00:00:00 ps

Information =====
Current Time: Fri Dec 21 01:13:27 CST 2018
Online Users: 15
CSCC Announce =====
CS Computer Center <help@cs.nctu.edu.tw>

llyu6072@linux1 ~$ ls
a.out core os_hw4 test
llyu6072@linux1 ~$ cd 2018_OS_HW4/
llyu6072@linux1 ~/2018_OS_HW4$ ls
0516072_hw4_bonus.cpp bmpReader.cpp input5.bmp output5.bmp
0516072_hw4.cpp bmpReader.h mask_Sobel.txt report.odt
aks bonus.cpp OS_hw4.odp Speed.sh
ans1.bmp ch06-2018.pdf OS_hw4.odp test1.cpp
ans2.bmp example_hw4.cpp OS_hw4.pptx test3.cpp
ans3.bmp input1.bmp output1.bmp test.cpp
ans4.bmp input2.bmp output2.bmp
ans5.bmp input3.bmp output3.bmp
a.out input4.bmp output4.bmp
llyu6072@linux1 ~/2018_OS_HW4$ g++ -std=c++11 -pthread 0516072_hw4_bonus.cpp
llyu6072@linux1 ~/2018_OS_HW4$ sh Speed.sh
Input a number of times to run './a.out' : 10

Run time:
Finished once.
Avg time: 982785 μs
llyu6072@linux1 ~/2018_OS_HW4$
```

Problems encountered and solutions:

本次作業最難的問題就是要想盡辦法讓 code 加速，再思考要如何用 thread，和如何把 code 簡化花了不少時間，最後決定把 function 的內容和再一起 讓 code 更精減，再 bonus 也花時間想了要如何用更多 thread 來加速，才想到了可以將回圈分成上半和下半同時做的方法來實作。