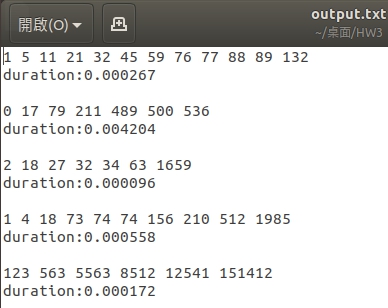
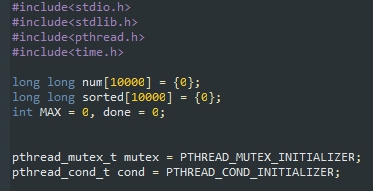
HW3 Multithread Programming

105072123 黃海茵

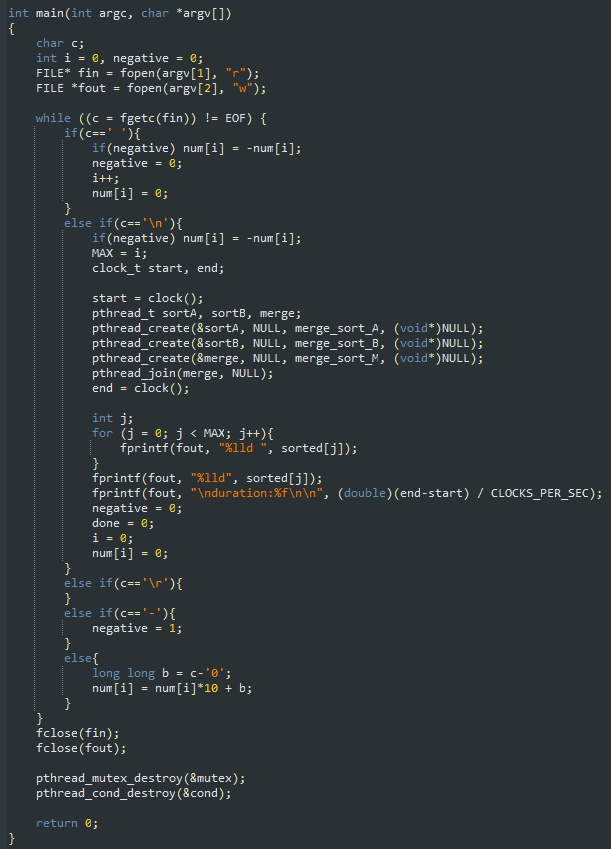
* Output



* Code



首先include所有會用到的header檔，然後建立global variable。num用來存testcase.txt中的數字，sorted用來存merge sort完的數字，MAX用來存該列有幾個數字，done表示有幾個thread已經sort完。接下來兩行則是pthread\_cond\_wait的mutex和condition。



c用來存input的字元，i表示讀到該列第幾幾個數字，negative用來表示是否為負數。argv[1]為testcase.txt，argv[2]為output.txt。

讀到數字時，先把c減掉’0’，從字元c轉為數字b，然後將前一個數字\*10 再加上b存入num[i]。

讀到負號時，把negative設為1，表示為負數。

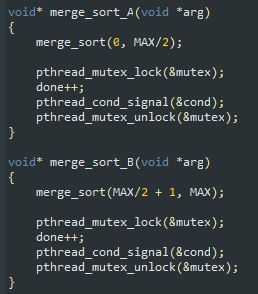
多處理一個讀到’\r’時，是因為在linux環境下讀取testcase.txt時，換行會有’\r’。（但我發現在windows環境下只會有’\n’）

讀到空格時，表示這個數字已結束。若negative為1，就把它加上負號，然後negative就可以歸零。i++到該列下一個數字，並先進行num[i] = 0的初始化。

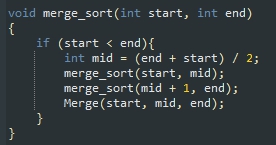
讀到換行時，表示該列最後一個數字已結束。若negative為1，一樣把它加上負號，negative就可以歸零。然後把MAX設為目前的i，就知道該行有幾個數字，便可以開始進行merge sort。

進行merge sort前，先宣告三個pthread變數。sortA用來sort數列的前半段，sortB用來sort數列的後半段，merge則用來做最後前半、後半數列的merge。

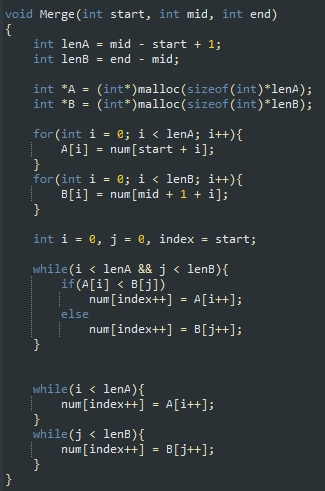
然後用pthread\_create來建立子執行緒，第一個參數為指向執行緒識別符的指標，第二個參數用來設定執行緒屬性，第三個參數是執行緒執行function的起始地址，最後一個參數是執行fucntion的引數。



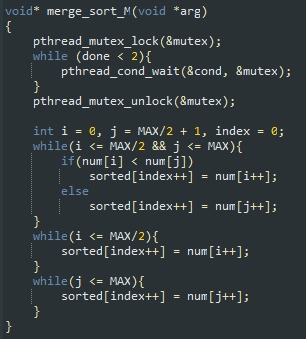
merge\_sort\_A就是pthread\_t sortA用來做merge sort的function。因為它是負責前半段數列的，所以只會做0到MAX/2的merge sort。並用mutex來避免同時修改done的問題，等它做完後，就會發signal給在等待的pthread\_t merge。merge\_sort\_B基本上跟merge\_sort\_A做的事情一模一樣，只是換成數列的後半段，所以是MAX/2+1到MAX。



這就是一個普通的merge sort fuction，把大數列切一半成為兩個小數列，然後把切好的兩個小數列再各自切一半，不斷重複，直到每個小數列都只剩一個數字，再用Merge來合併。

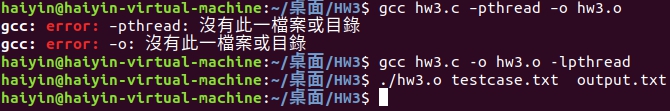


這也是一個普通的Merge fucntion，它所做的事情就是，把兩個小數列合併排序成一個大數列，這邊就不多加說明。



這是最後一個會進到的function，先檢查signal若不是done等於2時（merge\_sort\_A和merge\_sort\_B都完成），就會繼續在while loop中wait。merge\_sort\_M負責把sortA和sortB分別處理完的前半和後半數列，再合併排序成最終的完整數列sorted，做法和Merge fuction一樣。

* Note



我用作業講義中的指令gcc hw3.c –pthread –o hw3.o，會出現error，但改用gcc hw3.c -o hw3.o –lpthread就沒問題了。