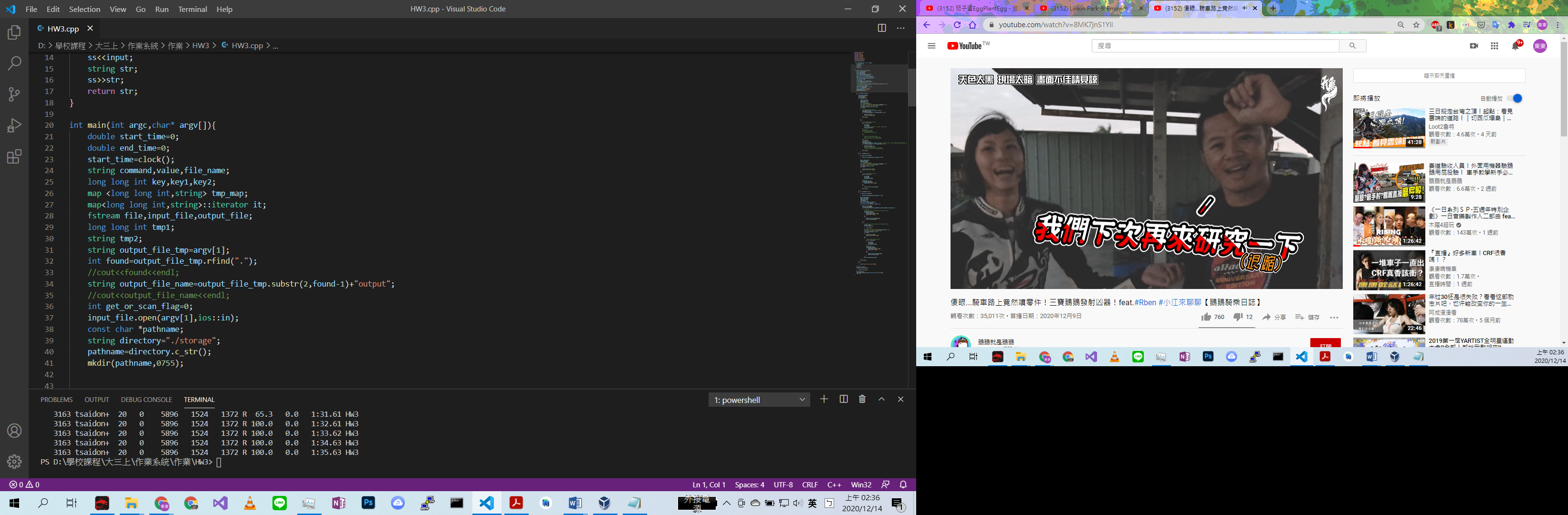
基本資訊

* 姓名:蔡東霖
* 系級:資訊111

開發環境：

* OS:ubuntu-20.04-desktop-amd64
* CPU: Intel® Core™i7-8750H CPU @ 2.20GHz × 3
* Ｍemory: 5GB
* Programming Language(version): gcc version 9.3.0

程式執⾏時間：

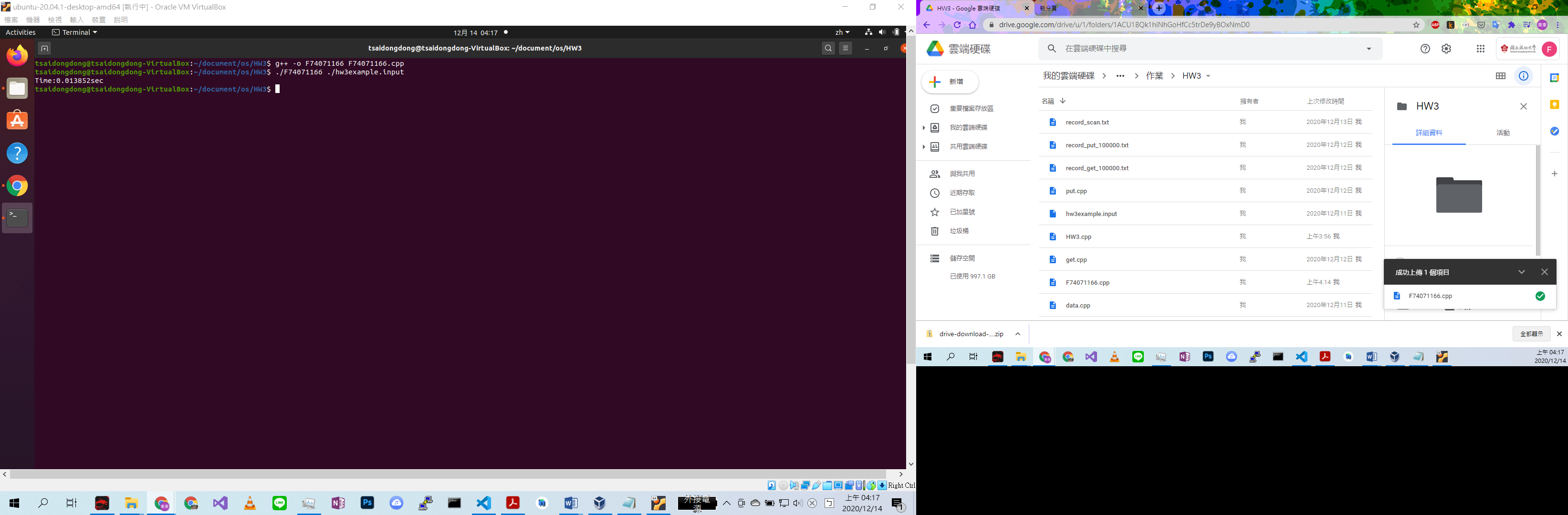


* 在code裡面有放start\_time=clock()、end\_time=clock()及cout<<(end\_time-start\_time)/CLOCKS\_PER\_SEC<<"sec"<<endl; 把時間印出來
* 在put 100000筆資料的時候平均花了85sec的時間，get 100000筆資料時，平均花了59sec，scan 10646565 65462312花了96.22sec

程式開發與使⽤說明：

* 首先讀取輸入的指令類別，再分為PUT、GET、SCAN的部分
* PUT的部分，以10000000存一個tmp檔案，先看是否已經有檔案，如果沒有的話，就自己開一個並且寫入數值，如果有的話，就將檔案先全部讀入tmp\_map再PUT進去，最後再將全部tmp\_map寫入檔案。
* GET的部分，先判斷get\_or\_scan\_flag是否等於0，若等於0就開一個output檔存資料，先判斷要求的key是否有這個tmp，沒有的話，就直接輸出EMPTY，若有的話將ˇ案內的資料全部存到tmp\_map，再尋找有沒有，有的話就輸出值，沒有的話，就輸出EMPTY。
* SCAN的部分，幾乎同上面的GET，但有多判斷下一個key是否在同一個tmp中，若在同一個的話，就不要關檔，繼續搜尋，反之就關檔在開檔搜尋。

使⽤說明：



分析報告：

* 整理資料的方法是每10000000存取一個tmp檔案，因此輸入一個key選擇存入的tmp是將key除以10000000，再尋找相符的檔案名存入，利用這種算法，可以使的存取的key數量超過memory的大小，除此之外，因為每次要寫入tmp檔案前，都會先將這個tmp裡面的資料存到map裡面，因為map內部會有紅黑樹，因此會自動排序資料，對搜尋跟插入友善，增加讀取、存取的速度，另一方面，因為SCAN相當於連續的GET，因此多設置判斷下一個key是否在同一個tmp中，若在同一個的話，就不要關檔，繼續搜尋，反之就關檔在開檔搜尋，可以避免開關檔的時間。
* PUT

從PUT的圖表中，我們可以知道CPU並非每分每秒都是100%狀態，然而，因為PUT是要將資料放入資料中，因此基本上所有的運算都是靠CPU去完成的，唯有在寫入硬碟時，CPU的使用率才會降低。

在MEMORY的部分，可以清出的看到，前期的時候，MEMORY的使用率都是0，直到後才多一點使用率，但那應該是代表多使用資料了，MEMORY的使用率才會提高。

* GET

在GET的圖表中，可以知道CPU的使用率是有週期的忽大忽小，可知是在使用時，計算完要讀取硬碟裡面的資料時，CPU的使用率便會下降，如此反覆週期。

倒是MEMORY的使用率都是0真的不太能理解。

在SCAN的方面，因為有設定若連續值的話，就不要開閉檔，因此很明顯的，CPU的使用率都處在滿點，雖然功能跟GET差不多，但因為沒有頻繁的開閉檔，浪費計算的時間。

如同GET的模式，在SCAN時，MEMORY的使用率為0