**NCTU OS HW2 report 2018**

**Name: 林亮穎**

**Student ID: 0516215**

|  |  |
| --- | --- |
| **Question** | **Answer** |
| **Q1.**  **Briefly describe about your design for the problem “Sum Checker” and total num of threads you used in your code.** |  |
| **Q2.**  **Show your thread info screenshots while “Sum Checker” code running.** |  |
| **Q3.**  **Compare the time between Single-thread and Multi-thread.** |  |
| **Q4.**  **What you learned from doing OS hw2 or some improvements you want to say to TAs.** |  |

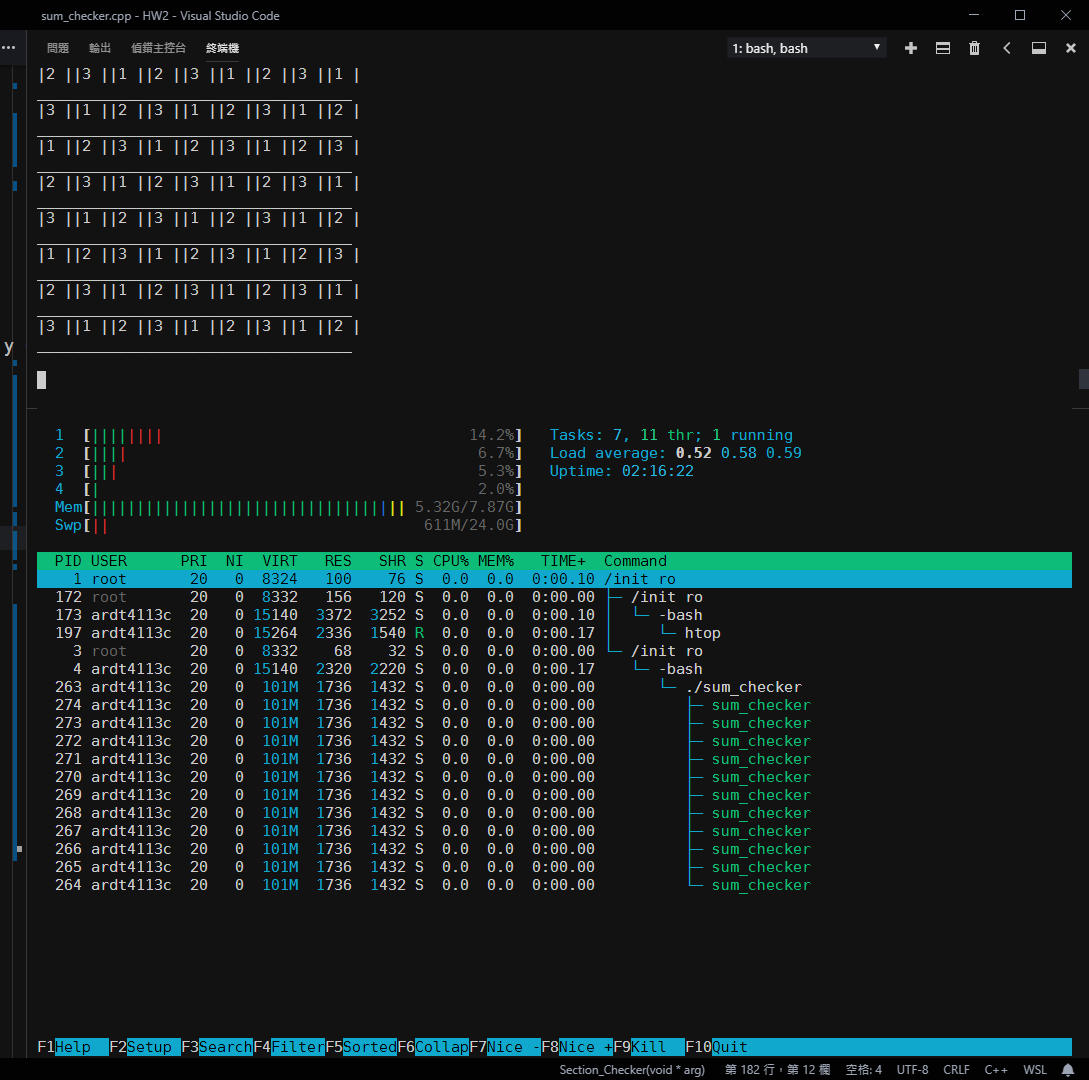
p.s You can reference to homework info pdf and show your answer as the format for Q2 and Q3.

一、第一題答案

我用了11條thread來完成Sum Checker，其中一條是用來掃過整個puzzle的九個row，確認每個row的數字總和是不是都相同；另一條是用來確認每個column的數字和是否相同；剩下的九條thread當中的每一條thread各自負責檢查其中一個3\*3方陣的數字和，如此一來puzzle中的九個3\*3方陣都會被檢查到。

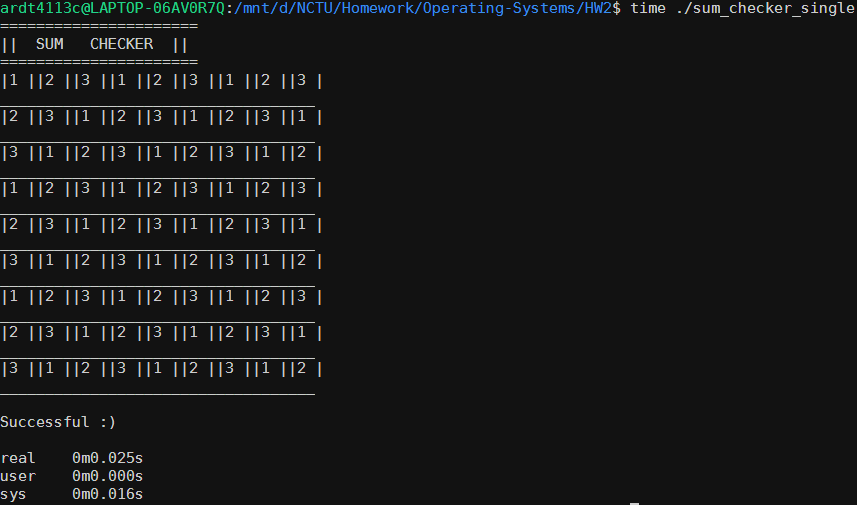
一開始執行程式時，我以第一條row的數字和做為每個區域內的數字和基準。而在每一條thread在檢查各自負責區域內的數字和時，若是數字和與第一條row的數字和相同，該條thread就會回傳0，表示該區域符合puzzle的規則；反之若是不相同，thread便會立即中止並回傳1，表示該區域不符合規則。

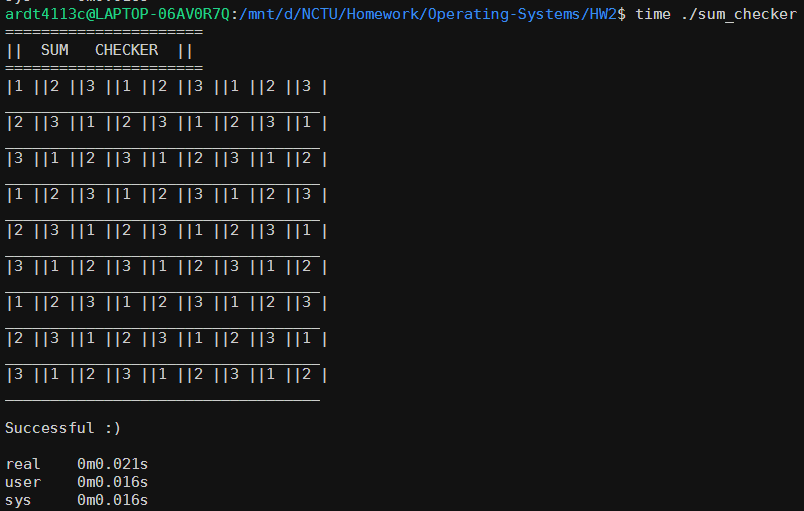
所以當main thread在執行pthread\_join()要合併各條thread的結果時，若是有任何一條thread回傳的結果是1，我就會將rv設為0，代表這個puzzle有部分區域的數字和與其他區域不同，程式就會輸出”Must check again ”的錯誤訊息。而若是每條thread皆回傳0，代表整個puzzle皆符合規則，程式便會輸出”Successful”的成功訊息。

二、第二題答案

由截圖可看出sum\_checker執行時真的創了11條thread。然而由於我寫的sum\_checker程式只要約0.02秒就可以跑完，因為跑得太快了，執行的時候在htop上並不能看到這些創出來的thread。所以為了產生上圖，我在每條thread結束前另外加了sleep(5)，如此一來htop才來得及顯示這些創出來的thread。另外，在下一題以及正式繳交的程式上我並沒有加上sleep(5)。

三、第三題答案





上圖是single thread版的執行結果，下圖是multithread版的執行結果，可以看出multithread的較single thread的快了0.004秒，Speed up= Ts/Tm = 0.025/0.021 =1.1904，multithread版比single thread版的約快上1.19倍。

四、第四題答案

在這次作業中我學到了如何利用pthread的函式庫，創建新的thread以及合併多條thread的執行結果，整個作業並不會太難，也能學到新東西。

另外，給TA的建議是：我覺得作業的puzzle尺寸給得太小了，無論是single thread還是multithread，程式基本上花不到0.05秒就能跑完，這樣子實在是很難看出multithread的所帶來的speedup好處。也因此在寫作業的Q3時，時常會出現single thread比multithread快上0.001秒之類的結果，很容易被這類的小誤差影響實際計算speedup的結果。