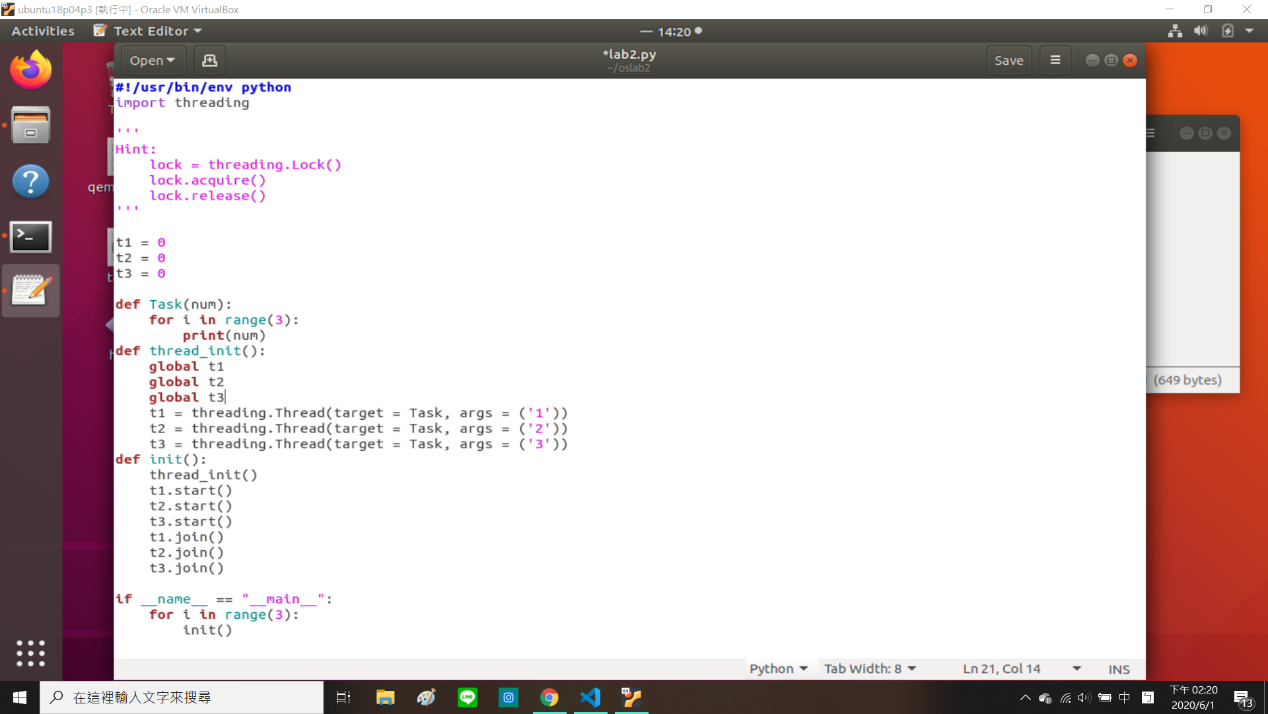
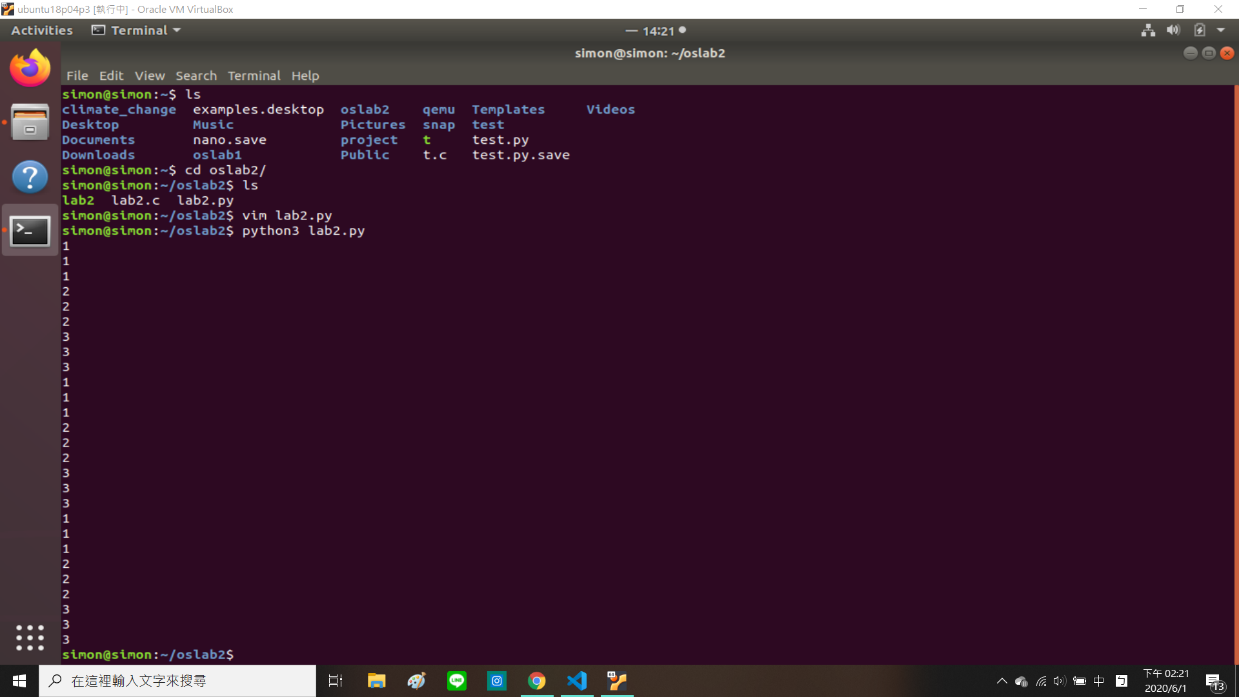
OS-B Lab2 Easy Synchronization

大氣3A 黃展皇 106601015

**\*以下操作均在ubantu虛擬機上作業\***

**Lab2-1**

將py code鍵入：

可以發現輸出結果是正確的

詢問過後助教表示可能電腦的運作速度太快，還沒有切換到thread就做完了，此部分debug可略。

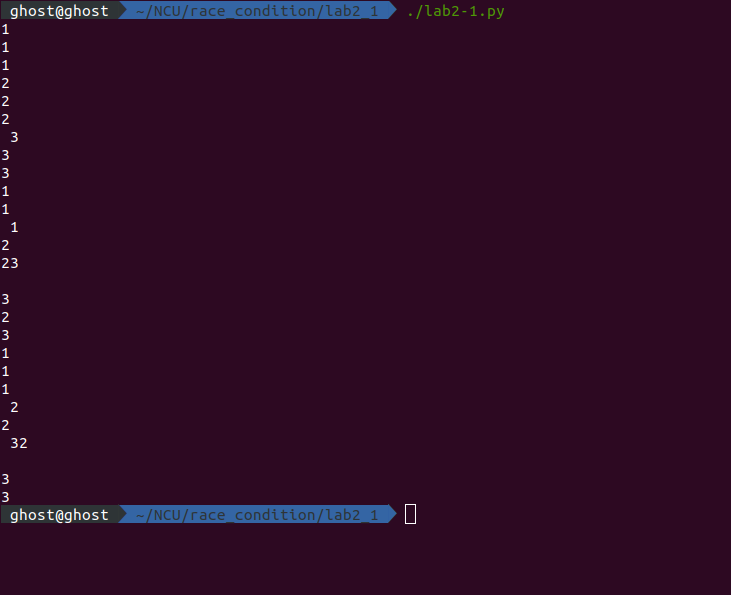
**簡單解釋code原理：**

主程式init()做三次的迴圈，每次均呼叫thread\_init()創立全域變數t1,t2,t3並分別task()，task迴圈三次印出num。  
故若沒有context switch或其他中斷事件，理論上順序應為：

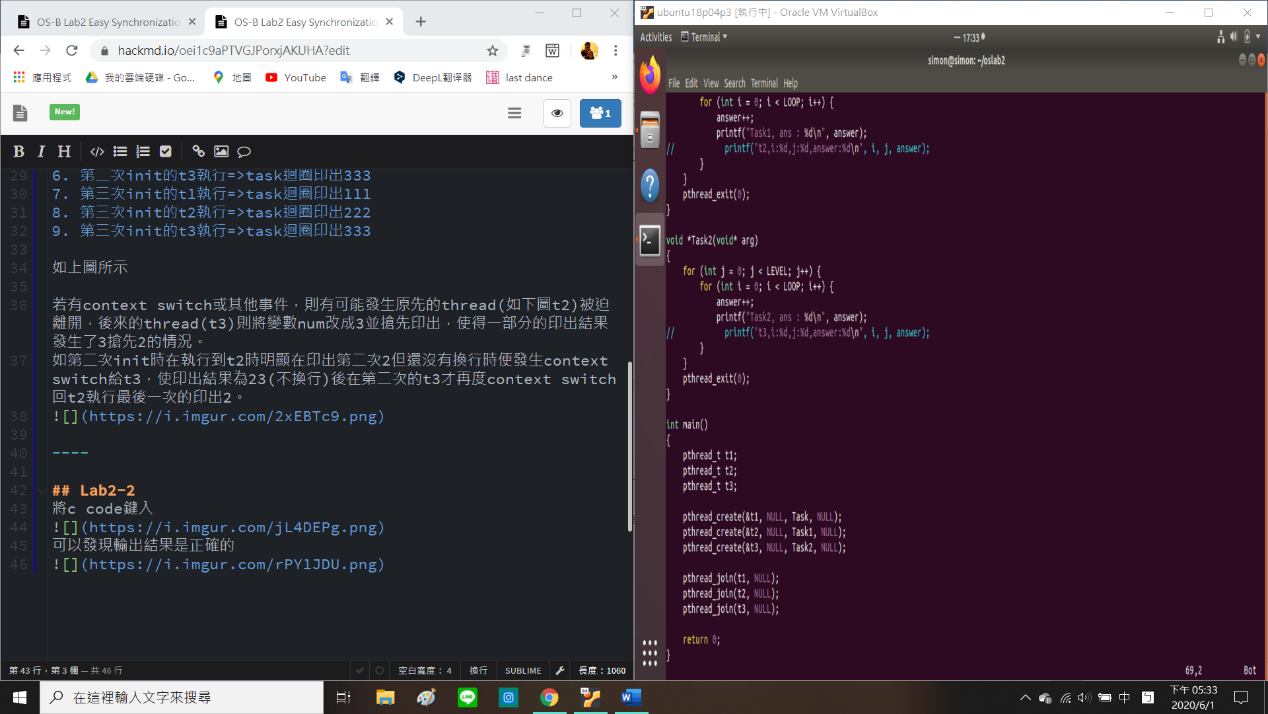
1. 第一次init的t1執行=>task迴圈印出111
2. 第一次init的t2執行=>task迴圈印出222
3. 第一次init的t3執行=>task迴圈印出333
4. 第二次init的t1執行=>task迴圈印出111
5. 第二次init的t2執行=>task迴圈印出222
6. 第二次init的t3執行=>task迴圈印出333
7. 第三次init的t1執行=>task迴圈印出111
8. 第三次init的t2執行=>task迴圈印出222
9. 第三次init的t3執行=>task迴圈印出333

完整順序如上圖所示

若有context switch或其他事件，則有可能發生原先的thread(如下圖t2)被迫離開，後來的thread(t3)則將變數num改成3並搶先印出，使得一部分的印出結果發生了3搶先2的情況。

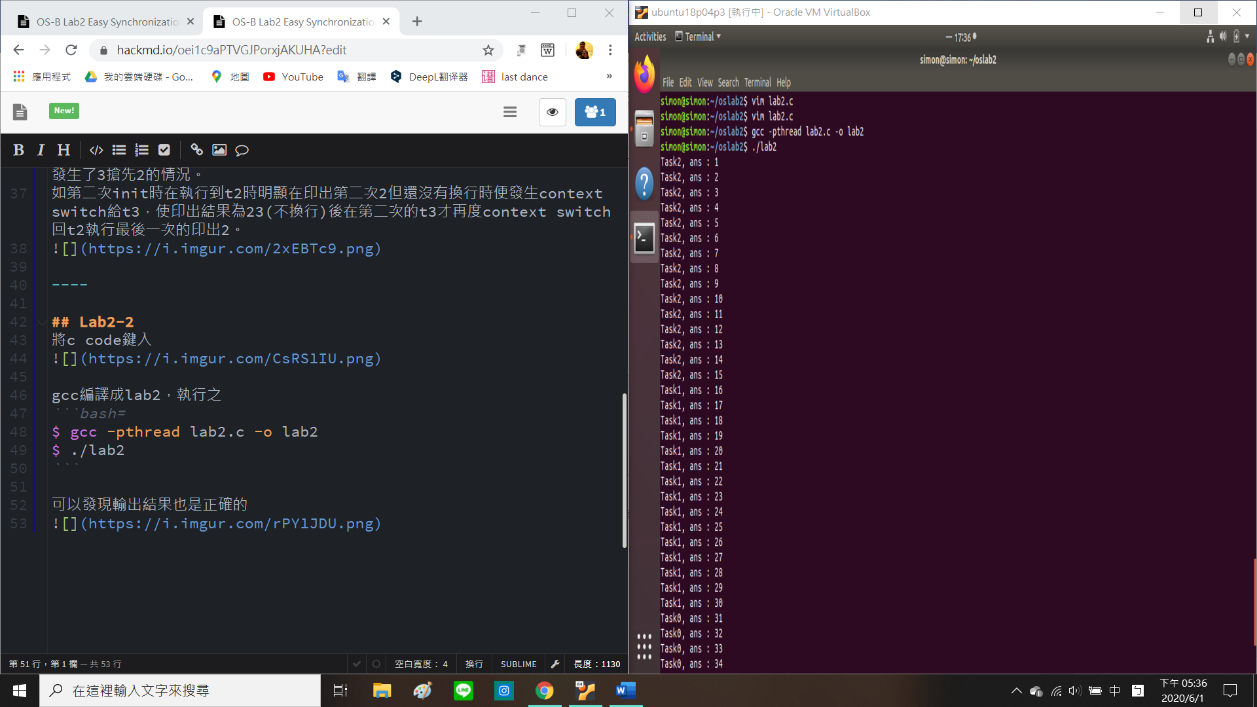
如第二次init時在執行到t2時明顯在印出第二次2但還沒有換行時便發生context switch給t3，使印出結果為23(不換行)後在第二次的t3才再度context switch回t2執行最後一次的印出2。

**Lab2-2-1**

將c code鍵入

gcc編譯成lab2，執行之

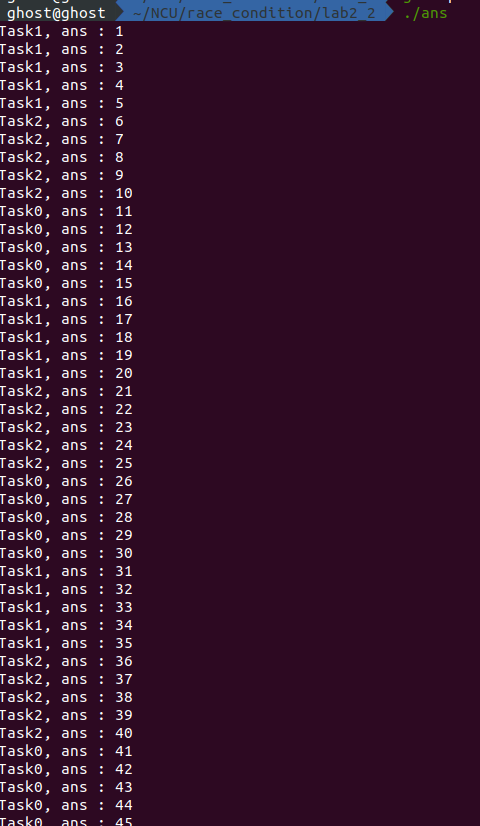
* *$ gcc -pthread lab2.c -o lab2*
* *$ ./lab2*

可以發現輸出結果也是正確的

同樣詢問過後助教表示可能電腦的運作速度太快，還沒有切換到thread就做完了，此部分debug可略。同時可以發現本電腦執行thread的順序為t3->t2->t1，task的順序為2->1->0且ans嚴格遞增。

**Lab2-2-2**

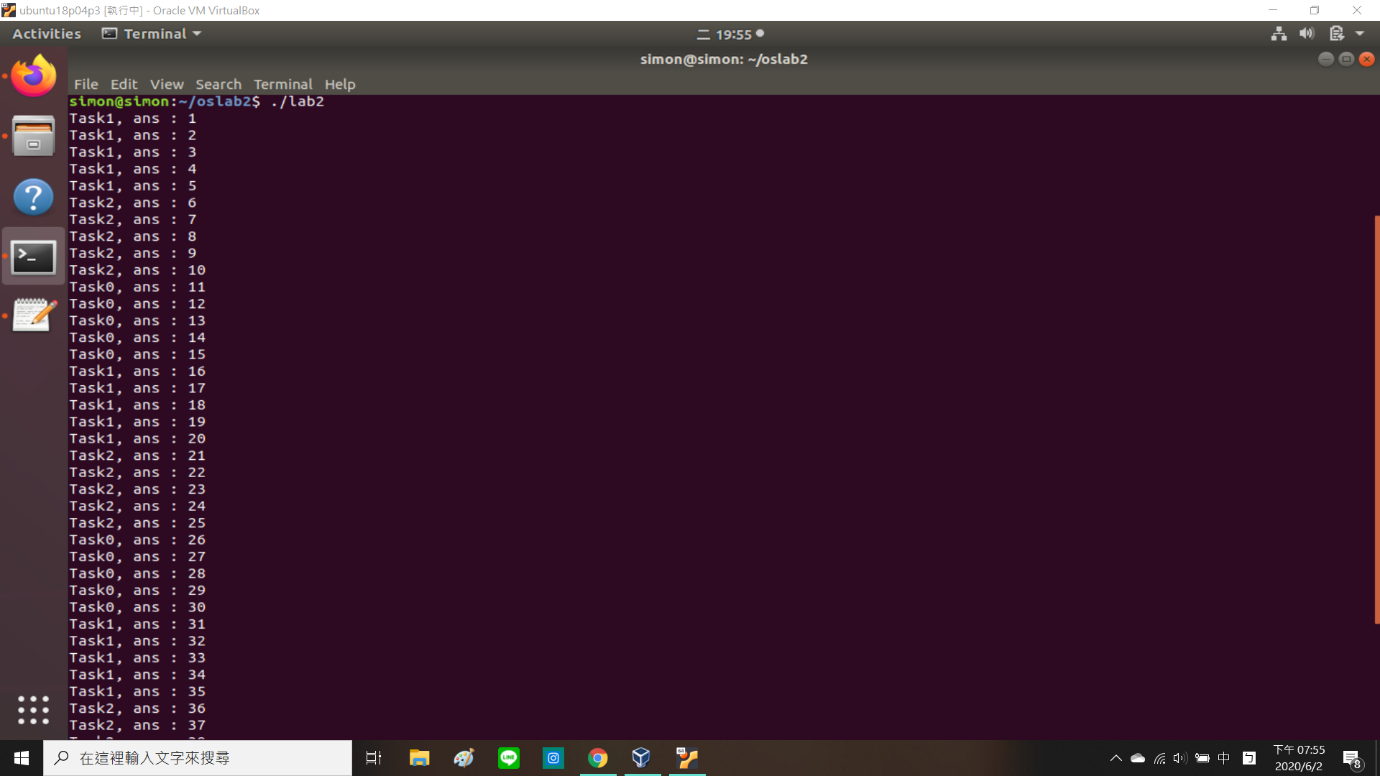
題目要求：Task是1,2,0,1,2,0的順序跟ans是1,2,3…,45執行，如下範例圖。



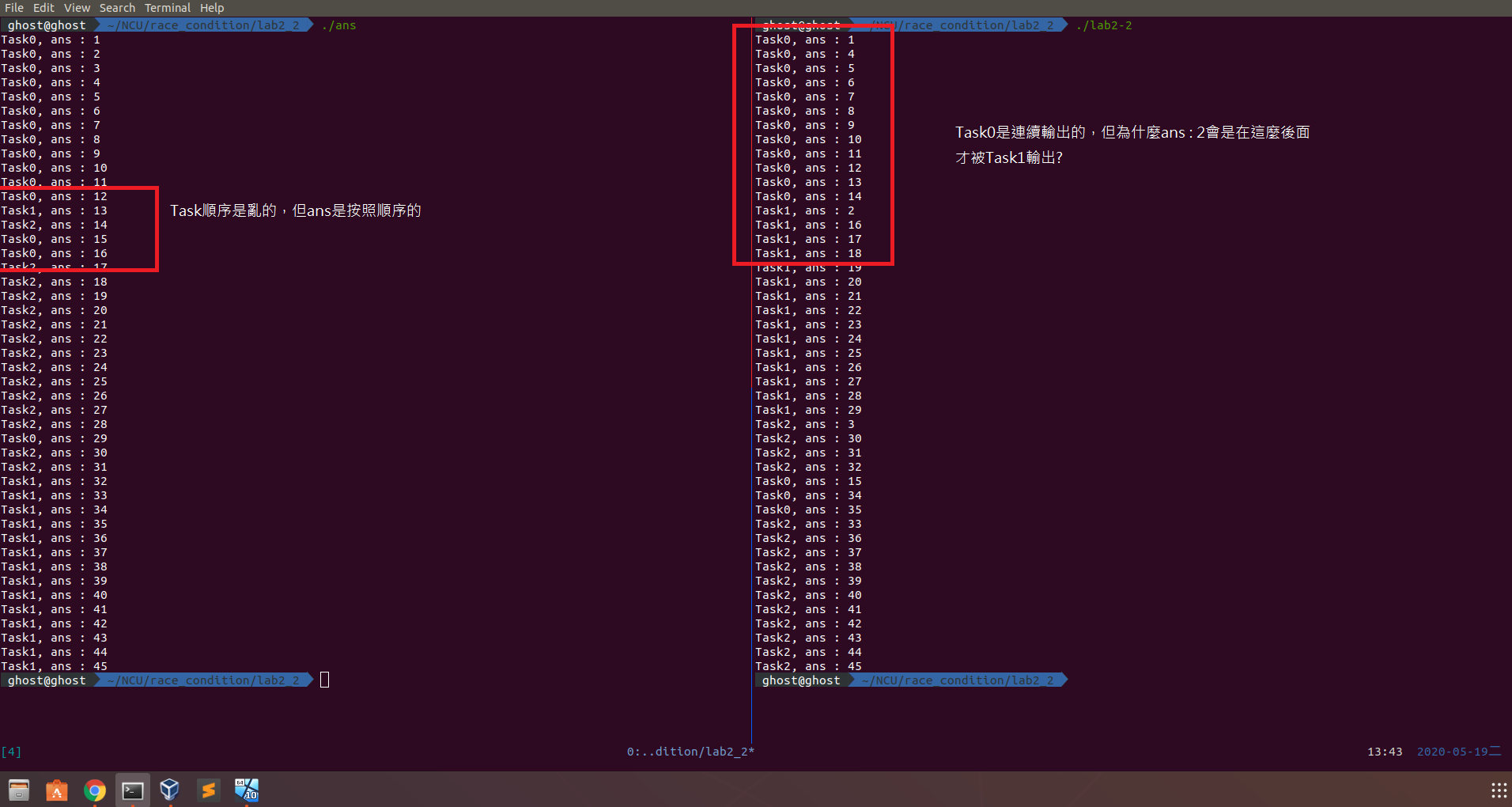
解決辦法：採用semaphore.h中的sem\_t、sem\_init()、sem\_wait()、sem\_post() function，以及sem\_t bs1, bs2, bs3分別控制pthread t1、t2、t3的順序

==> 每做一次小迴圈後換thread(t2->t3->t1)重複直至大迴圈結束。

***{詳細之code請洽lab2.c}***

執行結果如下圖：

成功:)

**Lab2-2-3**

問題：

1.“Lab2-2-1”(左)：task亂的但answer照順序；

2.“錯誤順序”(右)：task0持續輸出但answer=2很晚才被印出。

解答：

1. 主要是因為answer是公用變數而且t1、t2、t3對於answer的操作同樣都是+=1，在2.2.1修正I/O印出的順序問題後，無論是否有其他thread搶先在原thread還沒完成小迴圈時就context switch，皆不影響answer+=1的操作以及總共執行的次數(45次)，故answer照順序，但由於thread仍有搶先context switch的情況，故task仍然是亂的。

2. 以上圖為例，在I/O輸出沒問題的情況下明顯是task0在第二次存取answer時answer就已經是4了，表示answer先被其他的task操作過=>thread有搶先的情況。情況應該是：task0操作完一次後先後被task1和task2操作一次但暫時沒有印出，此時answer為3，接著task0第二次操作並印出此時為4的answer並以此類推至task1，印出本來該印的answer=2後繼續task1的部分，以此類推task2。

故仍有搶先context switch的情況，並沒有解決實際上的同步問題。

參考文獻與網址

https://www.itread01.com/content/1550033487.html

https://www.itread01.com/content/1541317627.html

http://www.syscom.com.tw/ePaper\_Content\_EPArticledetail.aspx?id=213&EPID=176&j=5&HeaderName=NonStop專欄

https://shouzo.github.io/2017/03/31/linux\_system/#一作業系統

詳情可洽hackmd：https://hackmd.io/@p7jEhLpkSaOxZJKlDSI4\_Q/SyRshzGnU