

OS HW4 report

Name: 吳原博

Student ID: 0816004

Q1:

Compare results between hw4_1_1 with/without synchronization.

```
$ ./1-1 < 12000-1.txt
0: 4044
1: 3973
2: 3983
```

```
$ ./1-1x < 12000-1.txt
0: 3592
1: 3554
2: 3559
```

左圖是加上 mutex 後的結果，和預期中的一樣算出每個數字的數量。右圖則為拿掉 mutex 的結果，每次執行都不太一樣，可以確定的是結果並不正確。

Q2:

Compare results between hw4_1_2 with/without synchronization.

```
$ ./1-2 < 12000-2.txt
0: 4044
1: 3973
2: 3983
```

```
$ ./1-2x < 12000-2.txt
0: 4044
2: 3983
1: 3973
```

這題我有用兩組 mutex，一組包住更新全域變數，另一組則確保輸出正確。左圖有加上 mutex，結果正常。由於我是在每個 thread 裡用一個無窮迴圈 busy waiting 直到 0,1,2 的數量加起來和測資字串長度一樣才 break 出來，因此從上題的結果會發現拿掉所有 mutex 程式很容易在這裡卡住。如果留下第一組 mutex 但是輸出不加以保護，結果可能會像右圖一樣，雖然數字正確但是輸出的順序混在一起，仍然變得不符合預期。

Q3:

Compare results between hw4_2 with/without synchronization.

<pre>\$./4-2 4 1000000 3.14052</pre>	<pre>\$./4-2x 4 1000000 3.11236</pre>
---------------------------------------	--

左圖有加上 `mutex`，得到的圓周率不能說有多準確，但至少是眾所皆知的 3.14。拿掉 `mutex` 後，我試過好幾次結果普遍都變小了，從第一題和這題的結果來看，似乎少了 `mutex` 保護會有導致 `critical section` 沒有執行到的趨勢。

Q4:

Some problems you meet and how to resolve.
or some Reflections.

我覺得主要是在 1-2 在 `thread` 裡做輸出遇到比較多問題。為了要保證程式跑完整個字串後才開始輸出，我用了一個無窮迴圈卡著直到 `cnt` 被更新完。但是這樣仍無法確保輸出的順序，所以除了加上 `mutex` 以外，在取得 `mutex` 的部份我也用一個迴圈包起來，確保三個 `thread` 會按照 0->1->2 的順序輸出。我覺得自己的作法大概不是最好的，但我想應該可以滿足題目要求。