

# OS HW4 report

Name: 王偉誠

Student ID: 0711506

Q1: Compare results between hw4\_1\_1 with/without synchronization.

沒用 semaphore :

```
mnt > ... > OS_HW4 10:04 0.007s ./a.out < 12000-1.txt
0: 3417
1: 3361
2: 3332
```

有用 semaphore :

```
mnt > ... > OS_HW4 10:10 0.725s ./hw4_1_1 < 12000-1.txt
0: 4044
1: 3973
2: 3983
```

沒用 semaphore 可能會出現 data race，導致結果出錯。但因為測資數目較小，所以需要多跑幾次才會觀察到出錯的結果。

Q2: Compare results between hw4\_1\_2 with/without synchronization.

沒用 semaphore :

```
mnt > ... > OS_HW4 10:05 0.008s ./a.out < 12000-2.txt
0: 3924
1: 3873
2: 3883
```

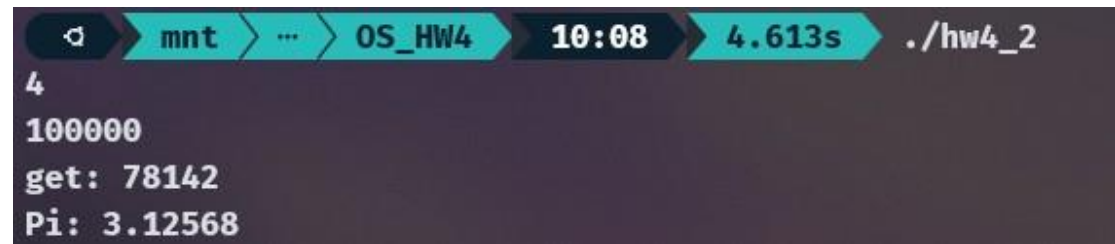
有用 semaphore :

```
mnt > ... > OS_HW4 10:09 0.735s ./hw4_1_2 < 12000-2.txt
0: 4044
1: 3973
2: 3983
```

沒用 semaphore 可能會出現 data race，導致結果出錯。但因為測資數目較小，所以需要多跑幾次才會觀察到出錯的結果。

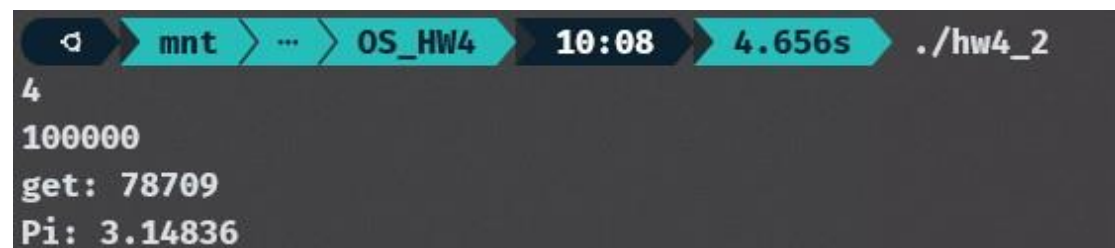
Q3: Compare results between hw4\_2 with/without synchronization.

沒用 semaphore：



```
mnt > ... > OS_HW4 10:08 4.613s ./hw4_2
4
100000
get: 78142
Pi: 3.12568
```

有用 semaphore：



```
mnt > ... > OS_HW4 10:08 4.656s ./hw4_2
4
100000
get: 78709
Pi: 3.14836
```

沒用 semaphore 的狀況會出現比較大的誤差，結果大概都是 3.11~3.13 左右；相較之下有使用 semaphore 的跑出來的結果幾乎都是 3.14 左右，答案相對準確。

Q4: Some problems you meet and how to resolve. or some Reflections.

HW4\_1\_2 需要用不同 thread 輸出答案，但是一開始沒有處理好，一直出現 deadlock，後來乾脆直接用 global variable 跟 while loop 來處理，最後就成功得到正確答案。

遇到的另一個問題就是寫 report 的時候，一開始把 semaphore 的部分註解掉

去跑，結果出來的答案卻沒有改變，去看討論區才知道要多跑幾次才比較容易觀察到 data race。