

# HW9

408410094 資工三 葉X勛

---

硬體：

**CPU** : intel i5-8265U 1.6GHz(8CPUs), ~1.8GHz

**microarchitecture** : Whiskey Lake

此CPU的核心與核心間是用一個ring(環狀網路)連接的。

實驗數據:

(1) 完全沒對程式碼進行修改

```
gcc -g -O3 -pthread fair.c -o fair
> ./fair
0, 5199701
1, 4339871
2, 4119209
3, 4643010
4, 4314672
5, 5197823
6, 5033694
7, 5200500
```

:不夠公平，高低落差很大。

(2)將最多等待次數從73改為100

```
> make
gcc -g -O3 -pthread fair.c -o fair
> ./fair
0, 5367107
1, 4082365
2, 3989209
3, 3966171
4, 3928022
5, 5549355
6, 5611795
7, 5715571
```

:整體看起來跟(1)並沒有明顯差異。

(3)將最多等待次數從73改為1000

```
> make
gcc -g -O3 -pthread fair.c -o fair
> ./fair
0, 4254306
1, 4793913
2, 4792765
3, 4823937
4, 4737735
5, 4157535
6, 4149453
7, 4088868
```

:高低落差有變小，但進去critical section次數也下降了，而且這樣還是不公平。

(4)將最多等待次數從73改為100000

```
> make
gcc -g -O3 -pthread fair.c -o fair
> ./fair
0, 5349527
1, 4145701
2, 4200810
3, 4169176
4, 4192261
5, 5074986
6, 5206291
7, 5482557
```

:高低落差反而變大了，發現等越久並不代表會越公平。

(5)不使用O3進行優化，且最多等待次數設成100

```
> make
gcc -g -pthread fair.c -o fair
> ./fair
0, 3448685
1, 3343089
2, 3366486
3, 3362735
4, 3291598
5, 3377341
6, 3392145
7, 3440661
```

:發現高低落差有明顯改善，但是進入critical section的次數明顯變少。

(6)不使用O3進行優化，且最多等待次數設成1000

```
gcc -g -pthread fair.c -o fair
> ./fair
0, 3345003
1, 3347412
2, 3350192
3, 3351212
4, 3339658
5, 3338937
6, 3349933
7, 3343254
```

:發現高低落差改善了很多，且進入critical section的次數並無明顯減少，可以說有達到每個thread進入critical section時是公平的。

(7)不使用O3進行優化，且最多等待次數設成100000

```
gcc -g -pthread fair.c -o fair
> ./fair
0, 36621
1, 37511
2, 37585
3, 37581
4, 37538
5, 36590
6, 36560
7, 36623
```

:發現高低落差雖然有減少，但進入critical section的次數大幅減少，雖然說有達到公平性但同時造成了程式效能低落。

## 解釋實驗數據:

1.在一開始有用O3對程式進行優化的情況下，將最多等待次數慢慢往上並沒有有效的改善thread間進入critical section的公平性，加到後面反而造成高低落差有加大一點點。我認為這個現象的原因是在fair.c內，我們是用一個空的for迴圈去讓thread做類似sleep的功能，而O3在進行優化的時候發現這個迴圈什麼事都沒做就將它拆解掉了，導致thread並沒有真正的依我們意思去sleep，才會造成無論最高等待次數加到多大都不會明顯的改善公平性或是有進入critical section次數降低的現象發生。

2.在之後我將O3的優化拿掉的情況下，在(5)的時候發現thread間的公平性明顯的改善，且當我將最多等待次數越設越大時，公平性也越來越明顯，進入critical section的次數越來越接近，但是測到(7)時我發現雖然最多等待次數越設越大能越公平，但會造成每個thread在for迴圈裡面等太久，進而降低了效能。從這裡可以得知的確是O3的優化將for拆掉造成不論最多等待次數設多大都無法改善公平性的情形。我認為我的程式跑出來的結果能看起來公平的原因有一個很大的因素是我的電腦只有8顆CPU，核心與核心間的距離很近，廣播時其他CPU馬上就可以收到最新訊息了，才會讓進入critical section的次數幾乎一樣。