

[ 3 ]

( 3a )

當 num\_thr = 3->2  
noncritical\_section\_size = 10000 -> 200  
critical\_section\_size = 1  
num\_iter = 1000->20000

Spinlock 的效能會比較好 2 秒。

( 3b )

< 1 >

由於改變 num\_thr critical\_section\_size 會讓 thread 等待 lock 的時間比較明顯，  
當 num\_thr 還有 critical\_section\_size 兩者的值”比較小”的時候，對於 thread 來說只需要在迴圈”等待一下”，原因是”排隊”的 thread 很少，所以 thread 在外面等待的時候，很快就可以拿到 lock，這樣的機制對於 spinlock 來說會比 mutex 來的好，因為 mutex 的機制是會 context-switch，不會一直進行等待，context-switch 會有額外的時間成本，導致 spinlock 比 mutex 快。

< 2 >

那什麼樣的數值是”比較小”呢？

這個答案會因為軟硬體的环境而有所差異。

像是我是用 vmware 的环境下去測試，

只有 2 顆 cpu。

如果是用工作站有多核心的情形之下，

num\_thr critical\_section\_size 可能”很小的數值”跑出來的結果是 mutex > spinlock

反而對於我的环境來說是 spinlock > mutex。

(3c) 截圖

```
shiwulo@vm:~/aosft$ time ./spin_lock 3 1000 1 1000
real    0m10.424s
user    0m19.885s
sys     0m0.750s
shiwulo@vm:~/aosft$ time ./mutex_lock 3 1000 1 1000
real    0m1.813s
user    0m0.000s
sys     0m0.188s
```

調整後：

```
shiwulo@vm:~/aosft$ time ./mutex_lock 2 200 1 20000
waiting for child threads
all threads finish their work
real    0m20.498s
user    0m0.063s
sys     0m1.721s
shiwulo@vm:~/aosft$ time ./spin_lock 2 200 1 20000
waiting for child threads
waiting for child threads
all threads finish their work
```

```
real    0m17.243s
user    0m0.087s
sys     0m1.213s
```

---

[ 4 ]

( 4a )

當 num\_thr = 2 -> 3

noncritical\_section\_size = 200-> 1000

critical\_section\_size = 1

num\_iter = 1000 -> 1000

mutex 的效能會比較好(8 秒 )。

( 4b )

< 1 >

透過改變 num\_thr critical\_section\_size 會讓 thread 等待 lock 的時間有明顯變化，當 num\_thr 還有 critical\_section\_size 兩者的值比較大的時候，對於 spinlock 來說，如果不 context-switch，一直在迴圈外面一直等待。反而會佔據 cpu 的使用。當很多個 thread 不願意放棄 cpu 的使用權，導致拿到 lock 的 thread 的效能上會被這些不願意放棄 cpu 使用權的 thread 給拖累。因此，在這種情況當中，mutex 明顯會比 spinlock 好，因為 thread 會願意放棄 cpu 的使用權，去 context-switch。

< 2 >

那什麼樣的數值是”比較大”呢？

這個答案會因為軟硬體的环境而有所差異。

我是用 vmware 的環境下去測試，

只有 2 顆 cpu。

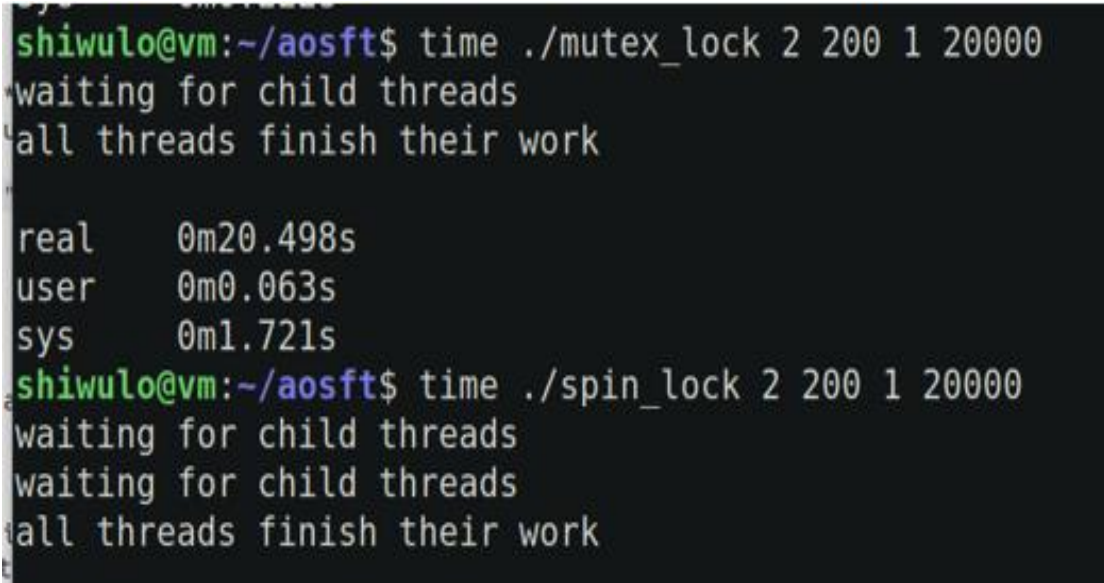
如果是用我的環境，核心數較少的情形之下，

num\_thr critical\_section\_size 可能”很大的數值”跑出來的結果是 mutex > spinlock

反而對於工作站的環境來說

結果是 spinlock > mutex

( 4c ) 截圖



```
shiwulo@vm:~/aosft$ time ./mutex_lock 2 200 1 20000
waiting for child threads
all threads finish their work

real    0m20.498s
user    0m0.063s
sys     0m1.721s
shiwulo@vm:~/aosft$ time ./spin_lock 2 200 1 20000
waiting for child threads
waiting for child threads
all threads finish their work
```

```
real    0m17.243s
user    0m0.087s
sys     0m1.213s
```

---

--- 調整後

```
shiwulo@vm:~/aosft$ time ./spin_lock 3 1000 1 1000
real    0m10.424s
user    0m19.885s
sys     0m0.750s
shiwulo@vm:~/aosft$ time ./mutex_lock 3 1000 1 1000
real    0m1.813s
user    0m0.000s
sys     0m0.188s
```