

作業 3

408210005 謝宗哲

1. 4 個 flock 同時執行:

```
aqua@aquu-ubuntu:~/system-programming/hw$ time ./flock & time ./flock & time ./flock & time ./flock
[1] 13351
[2] 13352
[3] 13353
1
1
1
2
2
2
3
3
3
```

我在 flock 的程式碼裡面加了在上鎖期間印出一個 count。

所以根據上圖結果來看，當其中一個 flock 執行到 sleep 的時候，會換另外一個 flock 執行上鎖並更新 flock.db 的內容。

```
real    1m41.582s
user    0m0.001s
sys     0m0.231s
1000

real    1m41.681s
user    0m0.006s
sys     0m0.096s
[2]- 已完成                time ./flock

real    1m41.687s
user    0m0.011s
sys     0m0.464s
[1]- 已完成                time ./flock
aqua@aquu-ubuntu:~/system-programming/hw$
real    1m41.688s
user    0m0.001s
sys     0m0.130s
^C
[3]+ 已完成                time ./flock
```

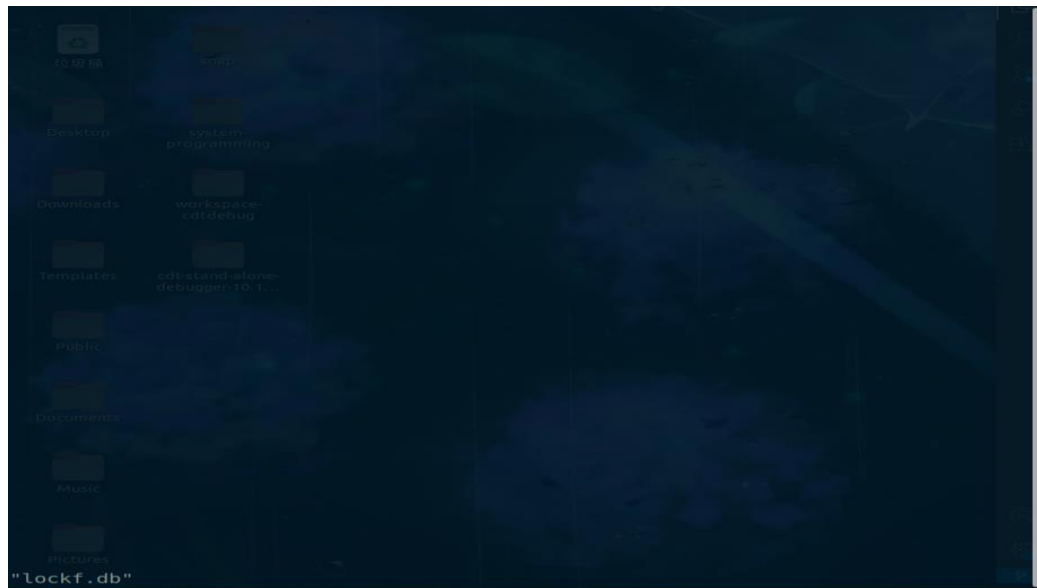
由於 sleep 的時間互相重疊，4 個 flock 幾乎同時完成，全部花的時間大約是 100 秒。

而在 `lockf` 的測試中，我在迴圈內上鎖期間印出當前是第幾個 `process` 在執行，測試出來卻不是同步執行，而是等到上一個執行結束才換下一個。
但是我沒有找到是什麼原因造成的

```
aqua@aqua-ubuntu:~/system-programming/hw$ time ./lockf 1 & time ./lockf 2 & time .  
./lockf 3 & time ./lockf 4  
[1] 8326  
[2] 8327  
[3] 8329  
process 1  
1  
process 1  
2  
process 1  
3  
process 1  
4  
process 1  
1000  
loop end  
process 3  
1  
real 1m40.847s  
user 0m0.008s  
sys 0m0.234s  
process 3  
2  
process 3  
3  
process 3  
4  
process 3  
1000  
loop end  
process 2  
1  
real 3m21.698s  
user 0m0.001s  
sys 0m0.288s  
process 2  
2  
process 2  
3  
process 2  
4  
process 2  
1000  
loop end  
process 4  
1  
real 5m2.622s  
user 0m0.001s  
sys 0m0.315s  
process 4  
2  
process 4  
3  
process 4  
4  
process 4  
1000  
loop end  
[1] 已完成  
[2] 已完成  
[3] 已完成  
time ./lockf 1  
time ./lockf 2  
time ./lockf 3  
real 6m43.399s  
user 0m0.030s  
sys 0m1.092s
```

全部執行時間約 400 秒。

2.多個 lockf 在更新 lockf.db 時, 使用 vim 打開 lockf.db:



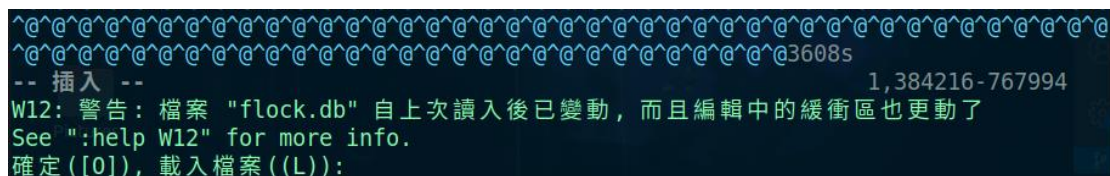
無法顯示 lockf.db 的內容

換成多個 flock 在更新 flock.db 時, 用 vim 打開 flock.db:



可以成功打開。

接著在 3608 後面加上一個 s 字元並直接儲存



按確定([O])之後
等待 flock 跑完
再 cat flock.db :

```
aqua@aqua-ubuntu:~/system-programming/hw$ cat flock.db
3500350135023503350435053506350735083509351035113512351335143515351635173518351935203521352235233524352535263
5273528352935303531353235333534353535363537353835393540354135423543354435453546354735483549355035513552355335
5435553556355735583559356035613562356335643565356635673568356935703571357235733574357535763577357835793580358
1358235833584358535863587358835893590359135923593359435953596359735983599360036013602360336043605360636073608
s
```

發現 3608 後面有 s 字元, 修改成功
但是在 s 字元之後沒有看到任何 flock 對 flock.db 修改的內容。

3. 將更新到最後一個數字是 7500 之後, lockf.db 的大小如下圖所示:

```
aqua@aqua-ubuntu:~/system-programming/hw$ ls -alhs lockf.db
16M -rw-rwSr-- 1 aqua aqua 21M 3月 28 19:57 lockf.db
```

所佔空間: 21M

實際大小: 16M

邏輯大小: $(7500-3500)*4 = 16000$ 約 16K

用 `stat -f` 查詢到 block size 是 4096 = 4K

由於大約有 4000 數字, 如果把每個數字間隔都當作大於 block size, 可以推算出檔案實際大小大概是 $4000*(\text{block size})=16\text{M}$, 跟實際情況差不多。