

Hw3

使用 flock 和 lockf 分別設計「只允許一個人寫入」的小型資料庫

系級：資工二 學號：409410114 姓名：周述君

一、如果同時執行 4 個 lockf (/flock)，這些程式可以同時執行，並更新 lockf.db (flock.db) 嗎？請問總共花地多少時間？

先 make，接著 sudo mount -oremount,mand /

```
chmod g+s ./lockf.db
```

```
chmod g-x ./lockf.db
```

預備強制鎖

- flock:

```
sujean@sujean:~/sp/sphw3$ sudo mount -oremount,mand /
[sudo] password for sujean:
mount: /: permission denied.
sujean@sujean:~/sp/sphw3$ chmod g+s ./lockf.db
sujean@sujean:~/sp/sphw3$ chmod g-x ./lockf.db
sujean@sujean:~/sp/sphw3$ time ./flock & time ./flock & time ./flock & time ./flock
[1] 18615
[2] 18616
[3] 18618

real    1m40.813s
user    0m0.001s
sys     0m0.168s

real    1m40.813s
user    0m0.001s
sys     0m0.166s
[1] Done
time ./flock
sujean@sujean:~/sp/sphw3$
real    1m40.820s
user    0m0.035s
sys     0m0.138s

real    1m40.831s
user    0m0.000s
sys     0m0.169s
sujean@sujean:~/sp/sphw3$
```

flock 鎖定檔案是可以同時執行並更新 flock.db 的，總共花費約 6 分 40 秒。

- lockf:

```
sujean@sujean:~/sp/sphw3$ time ./lockf & time ./lockf & time ./lockf & time ./lockf
[4] 18719
[5] 18720
[6] 18722

real    1m40.938s
user    0m0.036s
sys     0m0.213s

real    3m21.844s
user    0m0.161s
sys     0m0.114s

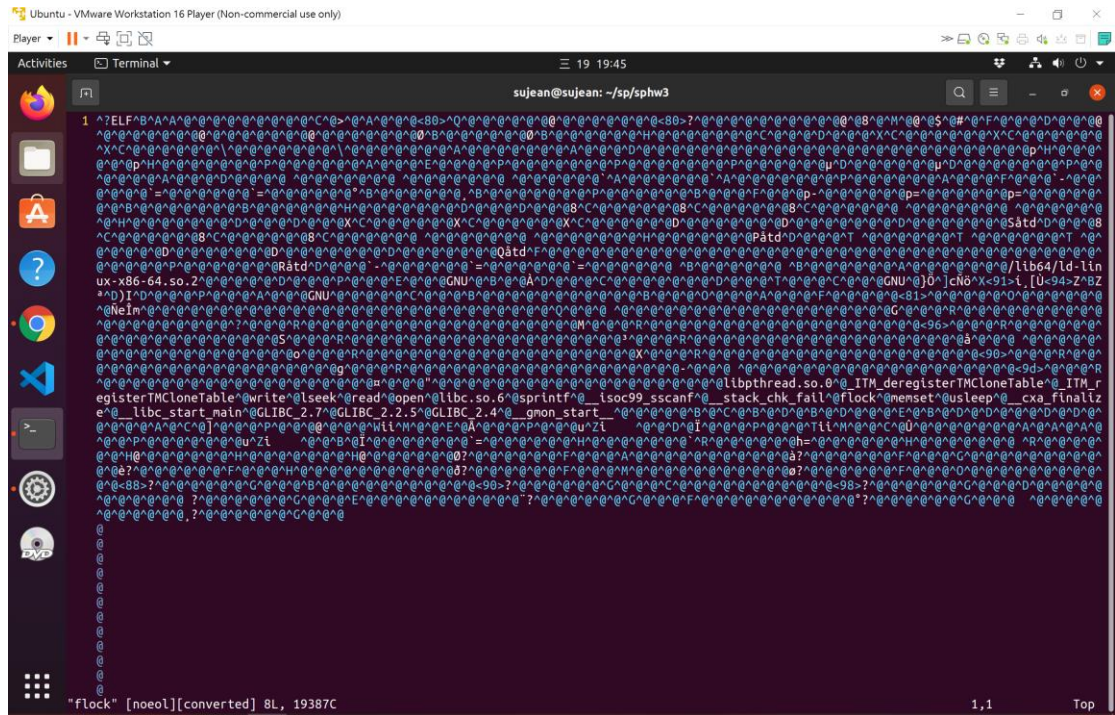
real    5m2.781s
user    0m0.095s
sys     0m0.191s
[2] Done
[3] Done
[4] Done
[5] Done
[6]+ Done
time ./flock
time ./flock
time ./lockf
time ./lockf
time ./lockf

real    6m43.656s
user    0m0.389s
sys     0m0.710s
```

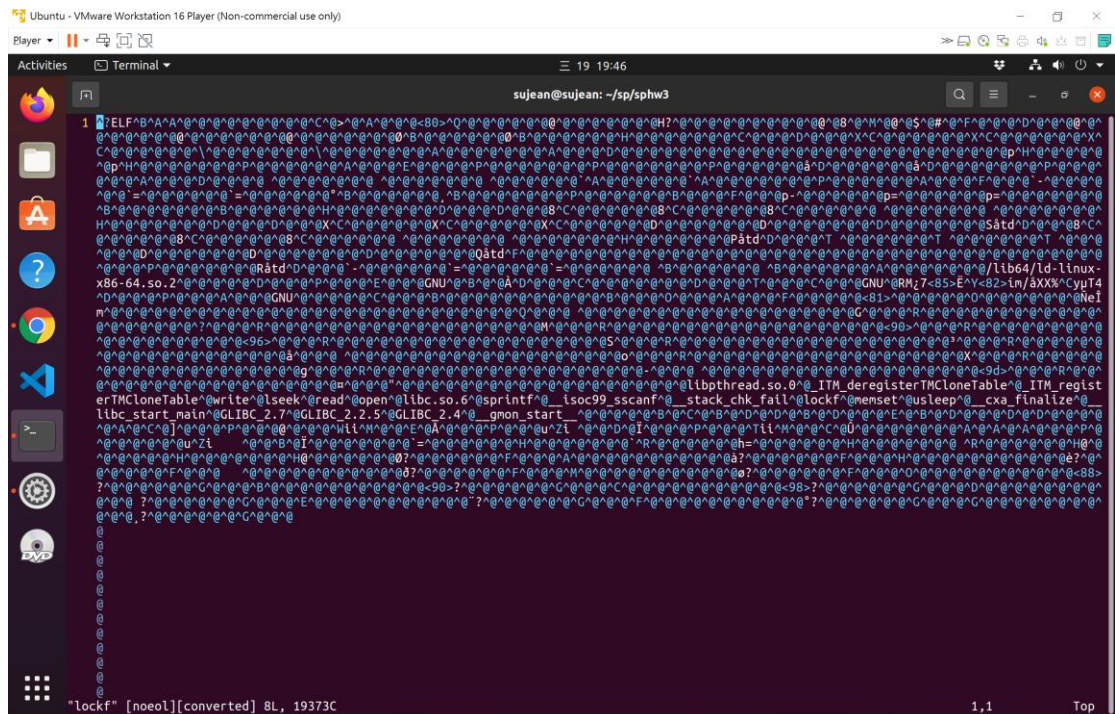
但 lockf 鎖定檔案因為要等前一個解鎖才會鎖定下一個，所以無法同時執行並更新 lockf.db，總共花費約 16 分 46 秒。

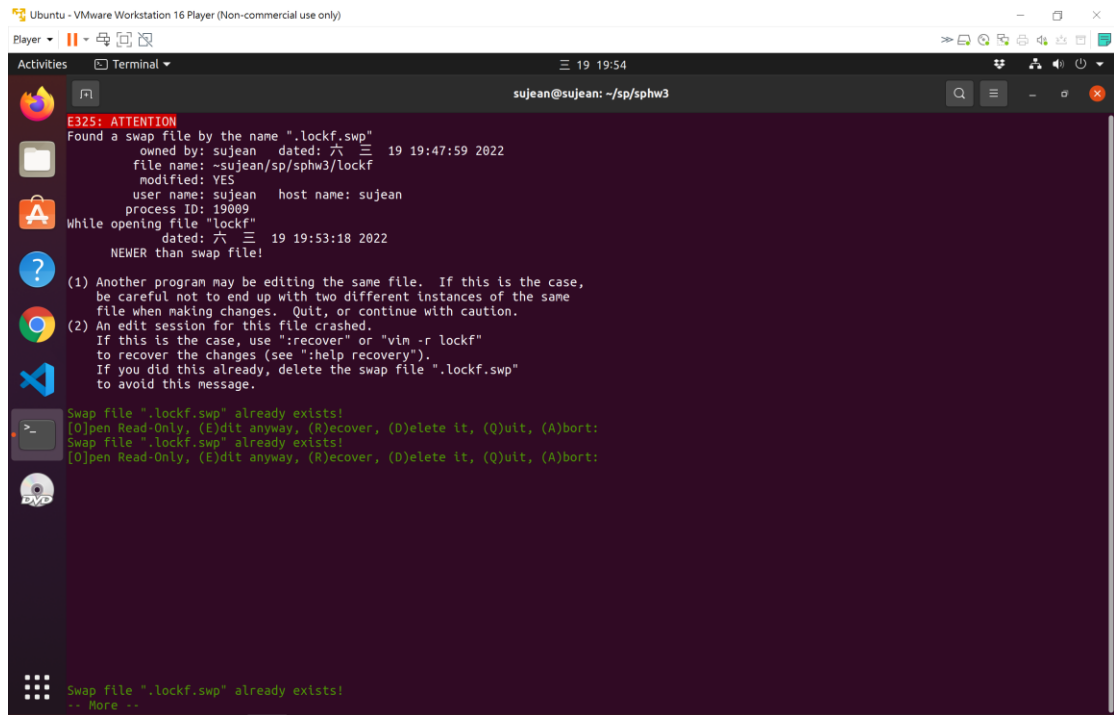
二、如果多個 `lockf`（`/flock`）在更新 `lockf.db`（`/flock.db`）的時候，你使用 `vim` 或其他編輯器打開 `lockf.db`（`/flock.db`），請問此時會發生什麼情況？如果儲存的話，會發生什麼樣的情況？

- flock: 可以儲存



- lockf: 出現警告





lockf 檔案讀取時已被修改

三、請問你的檔案的實際大小與邏輯大小？你有辦法『大致上』推算出檔案大小。

- flock:

執行一次的實際大小為 3.5M，邏輯大小為 4M (4*1000)

```
sujean@sujean:~/sp/sphw3$ ./flock
sujean@sujean:~/sp/sphw3$ ls -alhs flock.db
3.5M -rw-rwSr-- 1 sujean sujean 3.5M  三  21 02:01 flock.db
```

執行四次的實際大小為 12M，邏輯大小為 16M (4M*4)

```
sujean@sujean:~/sp/sphw3$ ls -alhs flock.db
12M -rw-rwSr-- 1 sujean sujean 12M  三  21 02:18 flock.db
```

- lockf:

執行一次的實際大小為 2.9M，邏輯大小為 4M (4*1000)

```
sujean@sujean:~/sp/sphw3$ ./lockf
sujean@sujean:~/sp/sphw3$ ls -alhs lockf.db
2.9M -rw-rwSr-- 1 sujean sujean 2.9M  三  21 02:11 lockf.db
```

執行四次的實際大小為 12M，邏輯大小為 16M (4M*4)

```
sujean@sujean:~/sp/sphw3$ ls -alhs lockf.db
18M -rw-rwSr-- 1 sujean sujean 21M  三  21 02:22 lockf.db
```

嘗試了非常多次後檔案大小還是有些怪異，參考學長姊作業有發現一樣的問題，可能是 ENOLCK/EDADLK/EINTR 的關係，3 threads 以上鎖太多區間而導致錯誤。

flock:

- 函式原型: `int flock(int fd, int operation);`
- LOCK_SH: 共享鎖
- LOCK_EX: 互斥鎖
- LOCK_UN: 解鎖
- LOCK_NB: 可與上述三種一起使用

lockf:

- 函式原型: `int lockf(int fd, int cmd, off_t len);`
- F_LOCK: 給檔案互斥加鎖，若檔案已被加鎖則返回 0，反之返回-1
- F_TLOCK: 同 F_LOCK，但若已被加鎖，不會阻塞，而返回錯誤
- F_ULOCK: 解鎖
- F_TEST: 測試檔案是否被上鎖，若檔案沒被上鎖返回 0，反之返回-1
- len: 從檔案當前位置要鎖住的長度