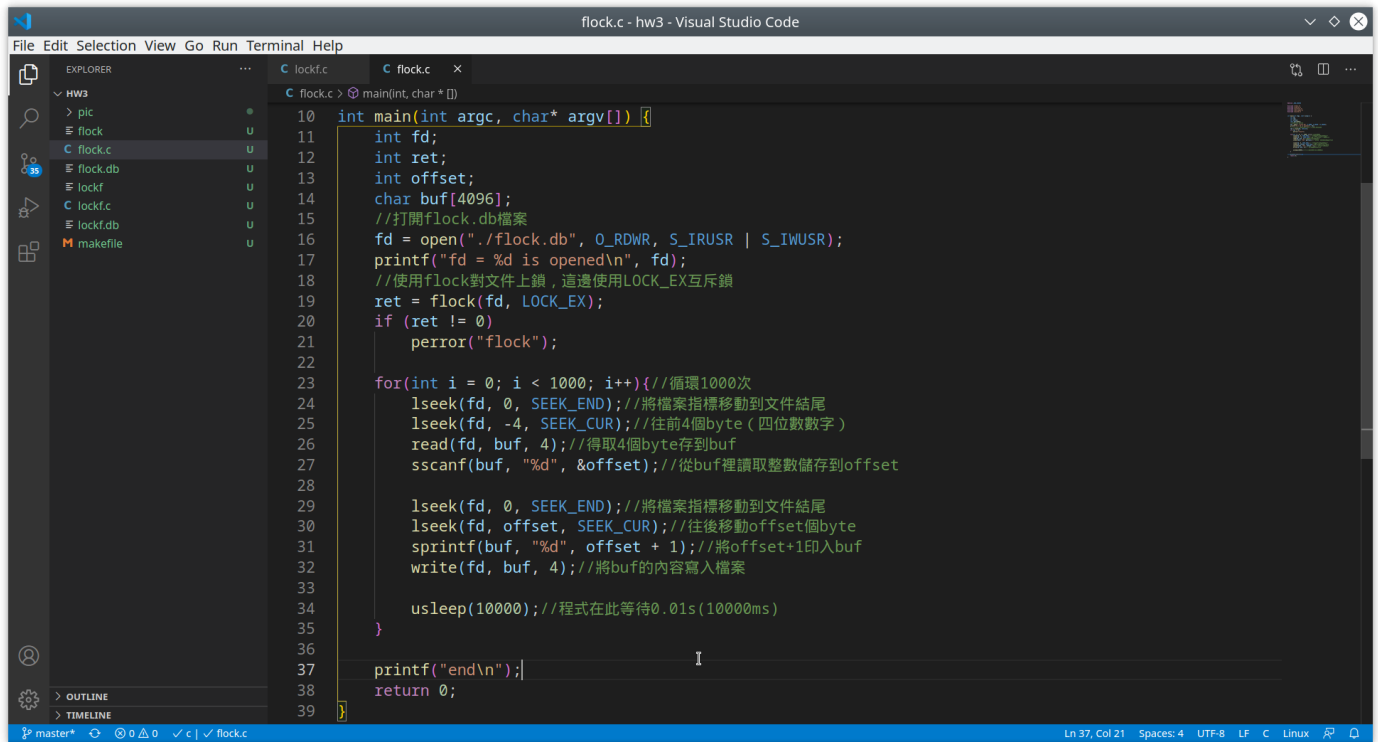


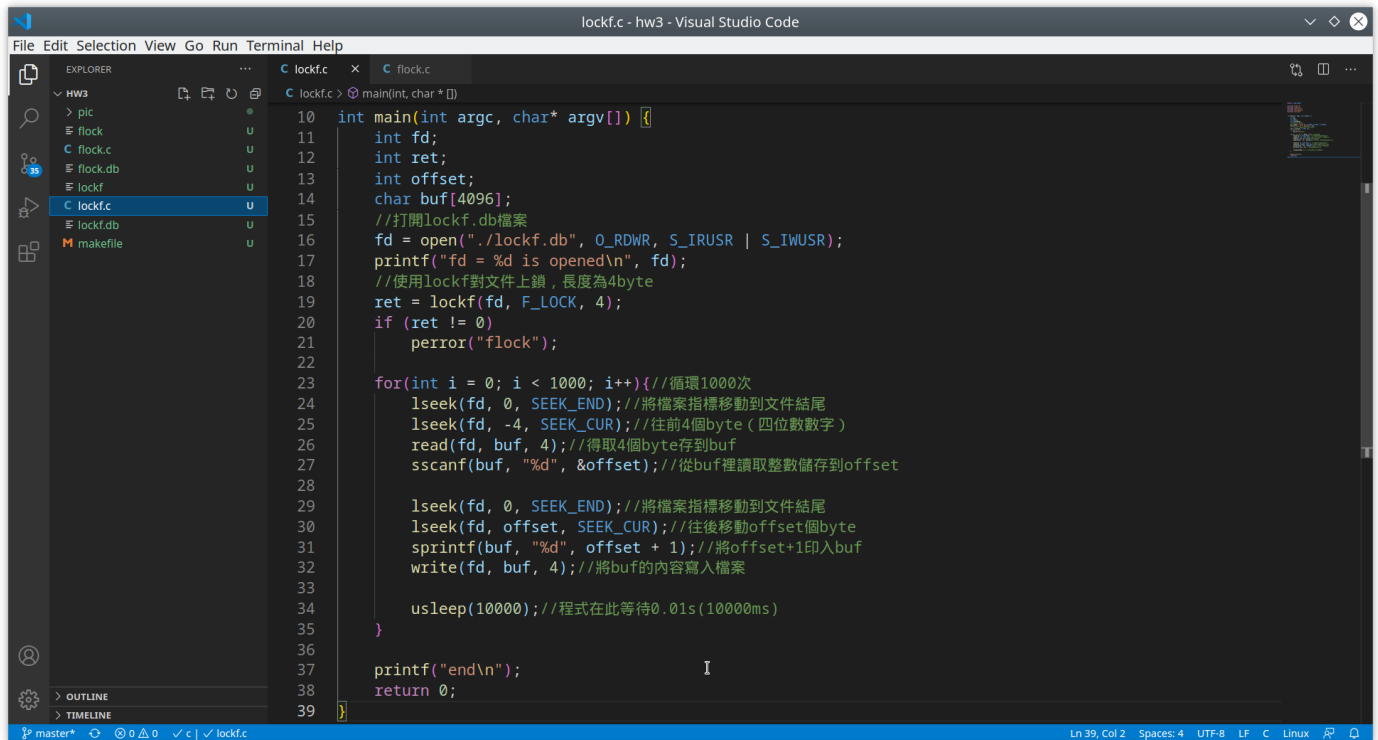
# System Programming Homework #3

王興彥 408410113

0. 按照題目要求撰寫程式，詳細內容寫在註解，圖一為 flock.c，圖二為 lockf.c

A screenshot of the Visual Studio Code editor showing the file flock.c. The Explorer sidebar on the left lists files: pic, flock, flock.db, flock.c, lockf, lockf.c, lockf.db, and makefile. The main editor area displays the C code for flock.c, which includes file opening, flocking, reading, and writing operations with detailed comments in Chinese. The status bar at the bottom indicates the file is at line 37, column 21.

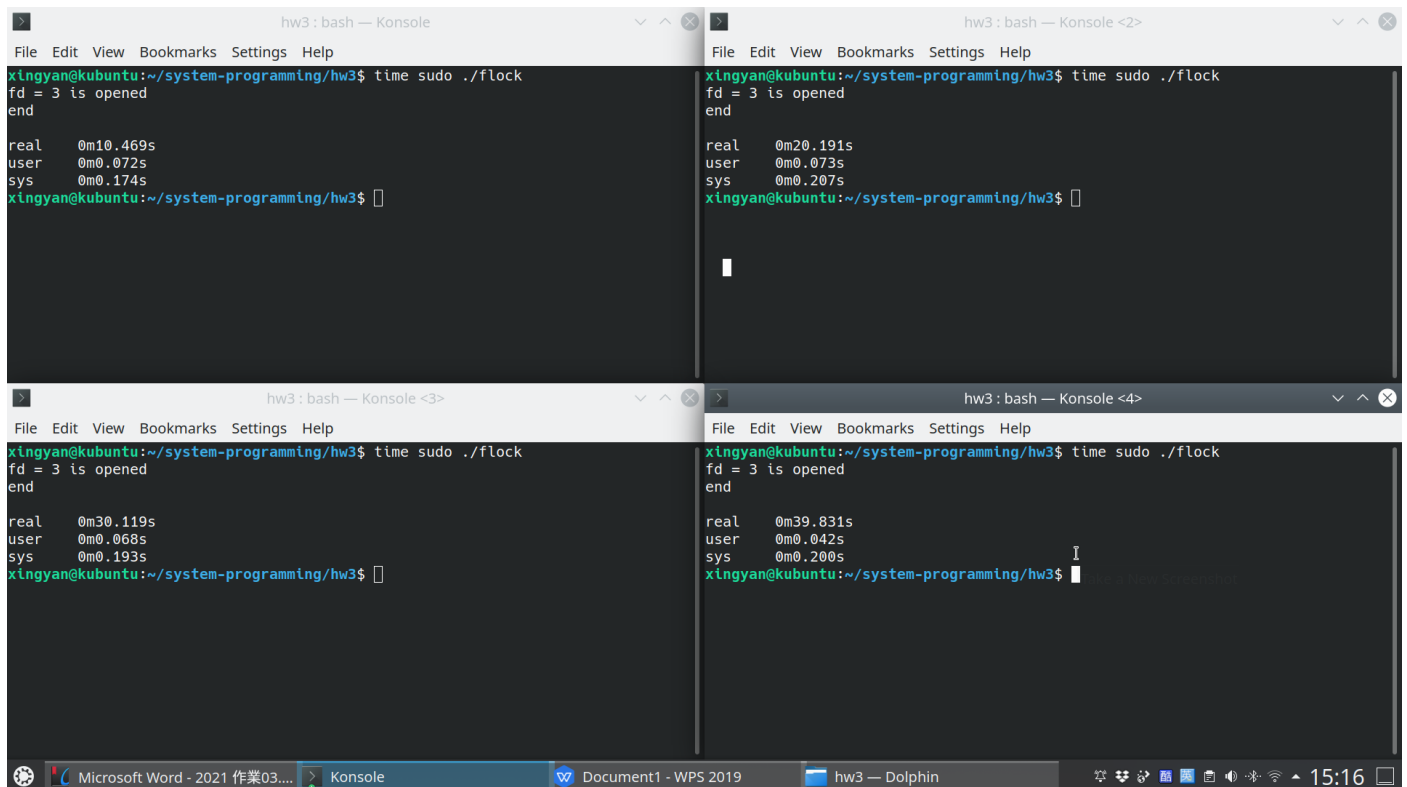
```
10 int main(int argc, char* argv[]) {
11     int fd;
12     int ret;
13     int offset;
14     char buf[4096];
15     //打開flock.db檔案
16     fd = open("./flock.db", O_RDWR, S_IRUSR | S_IWUSR);
17     printf("fd = %d is opened\n", fd);
18     //使用flock對文件上鎖，這邊使用LOCK_EX互斥鎖
19     ret = flock(fd, LOCK_EX);
20     if (ret != 0)
21         perror("flock");
22
23     for(int i = 0; i < 1000; i++){//循環1000次
24         lseek(fd, 0, SEEK_END);//將檔案指標移動到文件結尾
25         lseek(fd, -4, SEEK_CUR);//往前4個byte (四位數數字)
26         read(fd, buf, 4);//得取4個byte存到buf
27         sscanf(buf, "%d", &offset);//從buf裡讀取整數儲存到offset
28
29         lseek(fd, 0, SEEK_END);//將檔案指標移動到文件結尾
30         lseek(fd, offset, SEEK_CUR);//往後移動offset個byte
31         sprintf(buf, "%d", offset + 1);//將offset+1印入buf
32         write(fd, buf, 4);//將buf的內容寫入檔案
33
34         usleep(10000);//程式在此等待0.01s(10000ms)
35     }
36
37     printf("end\n");
38     return 0;
39 }
```

A screenshot of the Visual Studio Code editor showing the file lockf.c. The Explorer sidebar on the left lists files: pic, flock, flock.db, flock.c, lockf, lockf.c, lockf.db, and makefile. The main editor area displays the C code for lockf.c, which is similar to flock.c but uses lockf instead of flock for file locking. The status bar at the bottom indicates the file is at line 39, column 2.

```
10 int main(int argc, char* argv[]) {
11     int fd;
12     int ret;
13     int offset;
14     char buf[4096];
15     //打開lockf.db檔案
16     fd = open("./lockf.db", O_RDWR, S_IRUSR | S_IWUSR);
17     printf("fd = %d is opened\n", fd);
18     //使用lockf對文件上鎖，長度為4byte
19     ret = lockf(fd, F_LOCK, 4);
20     if (ret != 0)
21         perror("flock");
22
23     for(int i = 0; i < 1000; i++){//循環1000次
24         lseek(fd, 0, SEEK_END);//將檔案指標移動到文件結尾
25         lseek(fd, -4, SEEK_CUR);//往前4個byte (四位數數字)
26         read(fd, buf, 4);//得取4個byte存到buf
27         sscanf(buf, "%d", &offset);//從buf裡讀取整數儲存到offset
28
29         lseek(fd, 0, SEEK_END);//將檔案指標移動到文件結尾
30         lseek(fd, offset, SEEK_CUR);//往後移動offset個byte
31         sprintf(buf, "%d", offset + 1);//將offset+1印入buf
32         write(fd, buf, 4);//將buf的內容寫入檔案
33
34         usleep(10000);//程式在此等待0.01s(10000ms)
35     }
36
37     printf("end\n");
38     return 0;
39 }
```

測試：

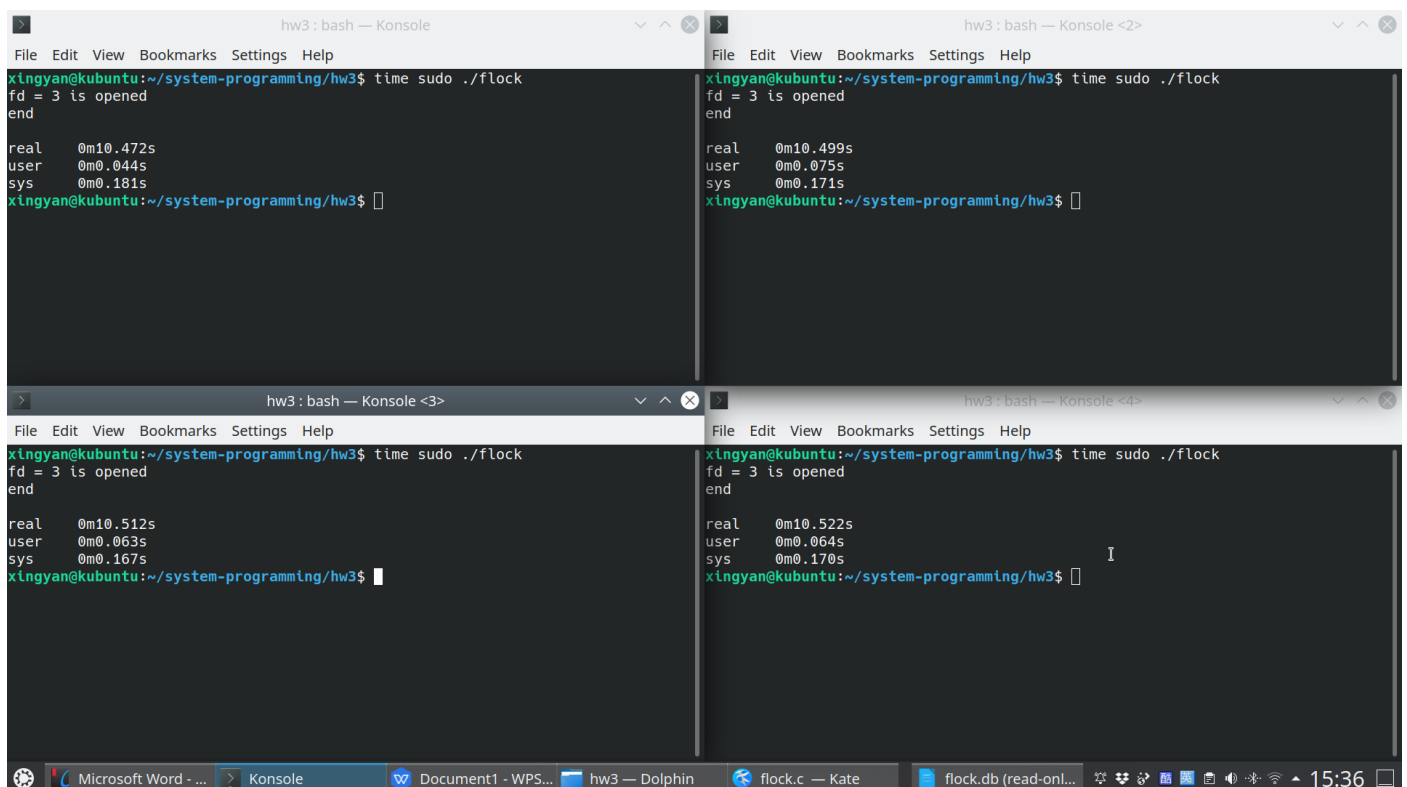
1-1. 4 個 flock 一起執行，因為都使用互斥鎖，所以會等到前一個 flock 完成後才會接著執行下一個，可從執行時間發現此結果。每個 flock 約執行 10s 因此最後總花費約 40s，每個程式執行 1000 次迴圈，最後也成功到 7500 ( 3500 + 1000 \* 4 )。下圖



The screenshot shows four terminal windows, each running the command `time sudo ./flock`. The output for each window is as follows:

Window	real	user	sys
1	0m10.469s	0m0.072s	0m0.174s
2	0m20.191s	0m0.073s	0m0.207s
3	0m30.119s	0m0.068s	0m0.193s
4	0m39.831s	0m0.042s	0m0.200s

1-2. 但若將 flock 改成 share ( LOCK\_SH )，則可同時執行，因此時間約為本身的 10s，但會造成文檔讀寫內容不同步，所以最後寫出的結果並不為 7500，無法預期結果。下圖



The screenshot shows four terminal windows, each running the command `time sudo ./flock`. The output for each window is as follows:

Window	real	user	sys
1	0m10.472s	0m0.044s	0m0.181s
2	0m10.499s	0m0.075s	0m0.171s
3	0m10.512s	0m0.063s	0m0.167s
4	0m10.522s	0m0.064s	0m0.170s

1-3. lockf 執行結果則和 flock 互斥鎖相似。

```
hw3: bash — Konsole
File Edit View Bookmarks Settings Help
real    0m10.557s
user    0m0.057s
sys     0m0.215s
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./flock
fd = 3 is opened
end

real    0m10.562s
user    0m0.065s
sys     0m0.209s
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./lockf
fd = 3 is opened
end

real    0m10.475s
user    0m0.060s
sys     0m0.170s
xingyan@kubuntu:~/system-programming/hw3$

hw3: bash — Konsole <2>
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./flock
fd = 3 is opened
end

real    0m20.191s
user    0m0.073s
sys     0m0.207s
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./lockf
fd = 3 is opened
end

real    0m20.520s
user    0m0.058s
sys     0m0.180s
xingyan@kubuntu:~/system-programming/hw3$

hw3: bash — Konsole <3>
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./flock
fd = 3 is opened
end

real    0m30.119s
user    0m0.068s
sys     0m0.193s
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./lockf
fd = 3 is opened
end

real    0m30.002s
user    0m0.061s
sys     0m0.158s
xingyan@kubuntu:~/system-programming/hw3$

hw3: bash — Konsole <4>
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./flock
fd = 3 is opened
end

real    0m39.831s
user    0m0.042s
sys     0m0.200s
xingyan@kubuntu:~/system-programming/hw3$ time sudo ./lockf
fd = 3 is opened
end

real    0m39.697s
user    0m0.070s
sys     0m0.172s
xingyan@kubuntu:~/system-programming/hw3$
```

2-1. 當執行 flock.c 時，使用 vim 進入 flock.db 可以讀寫，因為此時 vim 並沒有使用 flock，所以無須遵守 flock 規則，儲存也能夠成功。

2-2. 當執行 lockf.c 時，使用 vim 進入 lockf.db 無法進入，因為是使用強制鎖，因此也無法儲存。

3. 寫了一個程式來計算理論大小，跟實際大小一樣 3.8 MiB (4,003,504)

```
test.c - hw3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
  HW3
    > pic
    flock
    C flock.c
    flock.db
    C lockf.c
    lockf.db
    makefile
    test
    C test.c

C test.c > main()
1 #include<stdio.h>
2
3 int main ()
4 {
5     int num = 3500;
6     int sum = 4;
7     for(int i = 0; i < 1000; i++){
8         sum += num;
9         sum += 4;
10        num += 1;
11    }
12    printf("%f\n", sum /1048576.); //1Mib = 1048576byte
13    return 0;
14
15 }

TERMINAL
1: bash
xingyan@kubuntu:~/system-programming/hw3$ gcc test.c -o test
xingyan@kubuntu:~/system-programming/hw3$ ./test
3.818039
xingyan@kubuntu:~/system-programming/hw3$
```

```
hw3 : bash — Konsole
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/system-programming/hw3$ ls -lsh
total 7.6M
 20K -rwxr-xr-x 1 root    root      20K  19 23:10 flock
 4.0K -rw-rw-r-- 1 xingyan xingyan 1.2K  19 15:42 flock.c
 3.8M -rw-r--r-- 1 root    root     3.9M  19 23:12 flock.db
 20K -rwxr-xr-x 1 root    root      20K  19 23:10 lockf
 4.0K -rw-rw-r-- 1 xingyan xingyan 1.2K  19 14:56 lockf.c
 3.8M -rw-r-Sr-- 1 root    root     3.9M  19 23:20 lockf.db
 4.0K -rw-rw-r-- 1 xingyan xingyan 416   19 15:13 makefile
 4.0K drwxrwxr-x 2 xingyan xingyan 4.0K  19 23:14 pic
 20K -rwxrwxr-x 1 xingyan xingyan 17K   19 23:17 test
 4.0K -rw-rw-r-- 1 xingyan xingyan 238   19 23:17 test.c
xingyan@kubuntu:~/system-programming/hw3$
```