

<Code>

```
freopen("/dev/null", "R+", stdin); // 1
//底下這一行是將『標準輸出』改成從 "stdout.log"
freopen("stdout.log", "w+", stdout); // 2
//底下這一行是將『標準錯誤』改成從 "stderr.log"
freopen("stderr.log", "w+", stderr); // 3
//假裝我們的程式需要印出一些東西，然後又讀取一些東西
for (int i=0; i<10; i++) {
    printf("%d: hello\n", i); // 4
    int c = getchar(); // 5
}
//睡10秒鐘
//告訴作業系統，忽略掛斷的信號SIGHUP。signal hang up
signal(SIGHUP, SIG_IGN); // 6
//在20秒內切斷terminal的連線會怎樣？ls還會正常執行嗎？
printf("試著在20秒內切斷連線\n");
for (int i=0; i<20; i++) {
    fprintf(stderr, "*");
    sleep(1);
    sync();
}
execlp("ls", "ls", "/", "-R", NULL); // 7
```

<1>註解掉①

不讓 stdin 導向到/dev/null，且②,③不註解，導向到檔案

```
//freopen("/dev/null", "R+", stdin); // 1
//底下這一行是將『標準輸出』改成從 "stdout.log"
freopen("stdout.log", "w+", stdout); // 2
//底下這一行是將『標準錯誤』改成從 "stderr.log"
freopen("stderr.log", "w+", stderr); // 3
```

```
nash@SleepyCat:~/Desktop/sp_hw/12$ ./lovelyRon
```

打開 stdout 來看，發現只有後來⑦執行 execlp 要印出的東西，④⑤都不見了，沒有 hello

```
1 /:
2 bin
3 boot
4 cdrom
5 dev
6 etc
7 home
8 lib
9 lib32
10 lib64
11 libx32
```

參考 stackoverflow

Why does stdout need explicit flushing when redirected to file?

The behaviour of `printf()` seems to depend on the location of `stdout`.

1. If `stdout` is sent to the console, then `printf()` is line-buffered and is flushed after a newline is printed.
2. If `stdout` is redirected to a file, the buffer is not flushed unless `fflush()` is called.
3. Moreover, if `printf()` is used before `stdout` is redirected to file, subsequent writes (to the file) are line-buffered and are flushed after newline.

我猜是因為 stdout redirect 到其他 stdout.log 後，變成 fully-buffered，要加上 fflush() 才會出現 hello 或其他 printf() 的內容，至於為什麼執行 execlp 要印出的東西會印出，大概是因為執行結束時 buffer 裡的東西都會印出吧。測試如下

加入 fflush(stdout)

```
for (int i=0; i<10; i++) {  
    printf("%d: hello\n", i);           // 4  
    int c = getchar();                 // 5  
    fflush(stdout);  
}
```

檔案裡面確實出現 hello

```
nash@SleepyCat: ~/Desktop/sp_hw/12  
1 0: hello  
2 1: hello  
3 2: hello  
4 3: hello  
5 4: hello  
6 5: hello  
7 6: hello  
8 7: hello  
9 8: hello  
10 9: hello  
11 /:  
12 bin  
13 boot  
14 cdrom  
15 dev  
16 etc
```

<2> 註解掉②

①保留、註解掉②，④正常輸出 stdout，且⑤getchar()完全沒用，stdin 被導向/dev/null

```
//freopen("/dev/null", "R+", stdin);           // 1  
//底下這一行是將『標準輸出』改成從 "stdout.log"  
freopen("stdout.log", "w+", stdout);           // 2  
//底下這一行是將『標準錯誤』改成從 "stderr.log"  
freopen("stderr.log", "w+", stderr);           // 3
```

```
nash@SleepyCat:~/Desktop/sp_hw/12$ ./lovelyRon  
0: hello  
1: hello  
2: hello  
3: hello  
4: hello  
5: hello  
6: hello  
7: hello  
8: hello  
9: hello  
試著在 20秒內切斷連線
```

①②都註解掉，④正常輸出 stdout，且⑤getchar()會有作用

```
//freopen("/dev/null", "R+", stdin); // 1
//底下這一行是將『標準輸出』改成從 "stdout.log"
//freopen("stdout.log", "w+", stdout); // 2
//底下這一行是將『標準錯誤』改成從 "stderr.log"
freopen("stderr.log", "w+", stderr); // 3
```

```
nash@SleepyCat:~/Desktop/sp_hw/12$ ./lovelyRon
0: hello
1: hello
2: hello
```

<3> 註解掉③

```
freopen("/dev/null", "R+", stdin); // 1
//底下這一行是將『標準輸出』改成從 "stdout.log"
freopen("stdout.log", "w+", stdout); // 2
//底下這一行是將『標準錯誤』改成從 "stderr.log"
//freopen("stderr.log", "w+", stderr); // 3
```

註解掉③，不讓stderr導向到檔案。因為stderr是unbuffered，*會直接印到螢幕上，印完20個之後就會開始印⑦執行execvp發生permission denied的那些檔案

```
nash@SleepyCat:~/Desktop/sp_hw/12$ ./lovelyRon
***
ls: cannot open directory '/var/tmp/systemd-private-4f4eb4eeca32402aaff226a79533bc8a-systemd-timesyncd.service-6DoHdi': Permission denied
ls: cannot open directory '/var/tmp/systemd-private-4f4eb4eeca32402aaff226a79533bc8a-upower.service-ovA4vj': Permission denied
nash@SleepyCat:~/Desktop/sp_hw/12$
```

如果不註解掉③，檔案中並不會出現*，因為 buffer 的內容會被⑦執行 execvp 蓋掉。

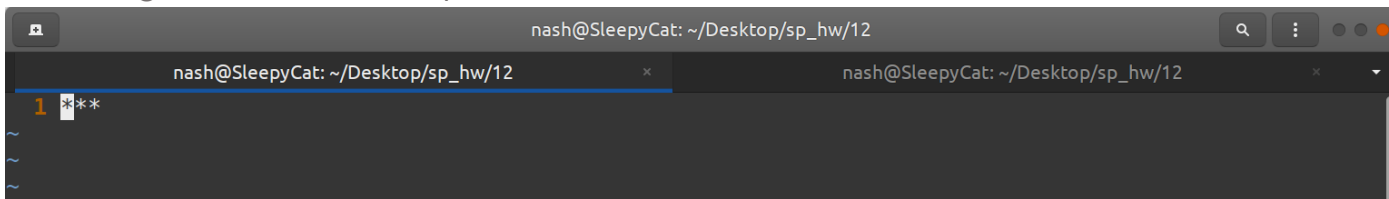
所以如果在 for 迴圈內加上 fflush(stderr)，stderr.log 檔案內就會出現 20 個*和⑦執行 execvp 列出的 permission denied 資訊

```
for (int i=0; i<20; i++) {
    fprintf(stderr, "*");
    sleep(1);
    sync();
    fflush(stderr);
}
```

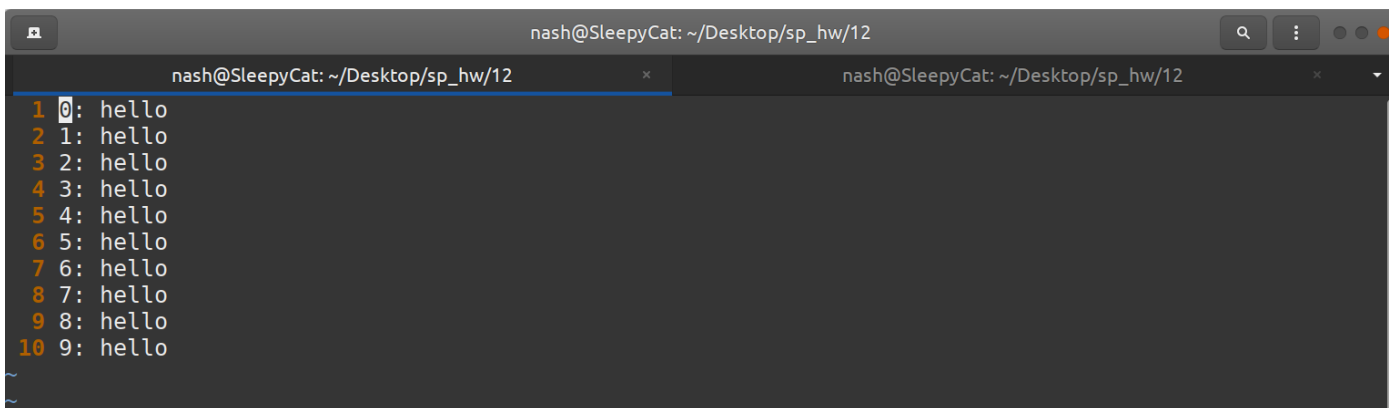
```
nash@SleepyCat: ~/Desktop/sp_hw/12
nash@SleepyCat: ~/Desktop/sp_hw/12
1 *****ls: cannot open directory '/boot/efi': Permission denied
2 ls: cannot open directory '/etc/cups/ssl': Permission denied
3 ls: cannot open directory '/etc/polkit-1/localauthority': Permission denied
4 ls: cannot open directory '/etc/ssl/private': Permission denied
5 ls: cannot open directory '/lost+found': Permission denied
6 ls: cannot open directory '/proc/1/fd': Permission denied
7 ls: cannot open directory '/proc/1/fdinfo': Permission denied
8 ls: cannot open directory '/proc/1/map_files': Permission denied
9 ls: cannot open directory '/proc/1/ns': Permission denied
10 ls: cannot open directory '/proc/1/task/1/fd': Permission denied
11 ls: cannot open directory '/proc/1/task/1/fdinfo': Permission denied
12 ls: cannot open directory '/proc/1/task/1/ns': Permission denied
13 ls: cannot open directory '/proc/10/fd': Permission denied
14 ls: cannot open directory '/proc/10/fdinfo': Permission denied
```

<4> 註解掉⑥

SIGHUP 預設的行為是終止程式，我執行到一半切斷 terminal，*將不會全部寫入 stderr.log
stdout.log 也沒有⑦執行 execlp 列出的資訊，只有 hello，我有在迴圈內加上 fflush()。

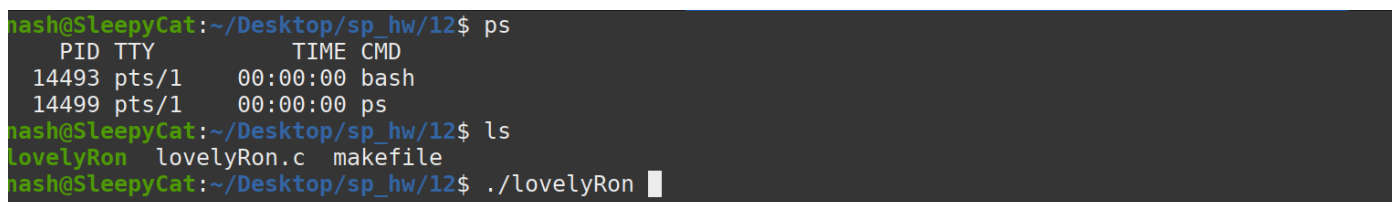


```
nash@SleepyCat: ~/Desktop/sp_hw/12
1 0: hello
```



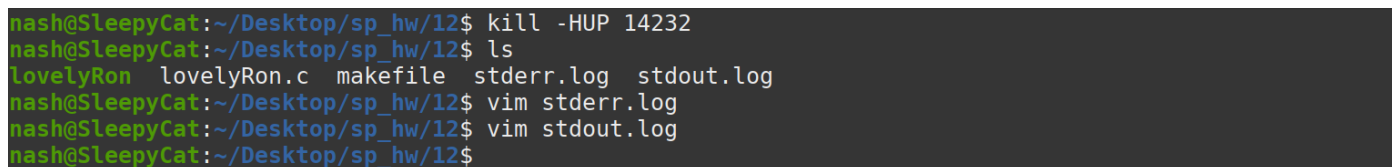
```
nash@SleepyCat: ~/Desktop/sp_hw/12
1 0: hello
2 1: hello
3 2: hello
4 3: hello
5 4: hello
6 5: hello
7 6: hello
8 7: hello
9 8: hello
10 9: hello
```

如果不註解掉⑥，我開兩個 terminal，其中一個用 kill 殺掉另一個 `kill -HUP pid`
切斷 terminal 後等個 20 秒看看



```
nash@SleepyCat:~/Desktop/sp_hw/12$ ps
  PID TTY          TIME CMD
 14493 pts/1    00:00:00 bash
 14499 pts/1    00:00:00 ps
nash@SleepyCat:~/Desktop/sp_hw/12$ ls
lovelyRon  lovelyRon.c  makefile
nash@SleepyCat:~/Desktop/sp_hw/12$ ./lovelyRon
```

14232 是上一次的，這裡節錯圖，因為想睡覺了



```
nash@SleepyCat:~/Desktop/sp_hw/12$ kill -HUP 14232
nash@SleepyCat:~/Desktop/sp_hw/12$ ls
lovelyRon  lovelyRon.c  makefile  stderr.log  stdout.log
nash@SleepyCat:~/Desktop/sp_hw/12$ vim stderr.log
nash@SleepyCat:~/Desktop/sp_hw/12$ vim stdout.log
nash@SleepyCat:~/Desktop/sp_hw/12$
```

stdout.log



```
1 0: hello
2 1: hello
3 2: hello
4 3: hello
5 4: hello
6 5: hello
7 6: hello
8 7: hello
9 8: hello
10 9: hello
11 /:
12 bin
13 boot
14 cdrom
15 dev
16 etc
```

stderr.log

```
1 *****ls: cannot open directory '/boot/efi': Permission denied
2 ls: cannot open directory '/etc/cups/ssl': Permission denied
3 ls: cannot open directory '/etc/polkit-1/localauthority': Permission denied
4 ls: cannot open directory '/etc/ssl/private': Permission denied
5 ls: cannot open directory '/lost+found': Permission denied
6 ls: cannot open directory '/proc/1/fd': Permission denied
7 ls: cannot open directory '/proc/1/fdinfo': Permission denied
8 ls: cannot open directory '/proc/1/map_files': Permission denied
9 ls: cannot open directory '/proc/1/ns': Permission denied
10 ls: cannot open directory '/proc/1/task/1/fd': Permission denied
11 ls: cannot open directory '/proc/1/task/1/fdinfo': Permission denied
12 ls: cannot open directory '/proc/1/task/1/ns': Permission denied
13 ls: cannot open directory '/proc/10/fd': Permission denied
14 ls: cannot open directory '/proc/10/fdinfo': Permission denied
15 ls: cannot open directory '/proc/10/map_files': Permission denied
16 ls: cannot open directory '/proc/10/ns': Permission denied
17 ls: cannot open directory '/proc/10/task/10/fd': Permission denied
```

<參考資料>

<https://www.796t.com/post/OHdxYzl=.html>

【LINUX】什麼是“輸出到 stdout”

<https://stackoverflow.com/questions/13932932/why-does-stdout-need-explicit-flushing-when-redirected-to-file>

Why does stdout need explicit flushing when redirected to file?

<https://blog.gtwang.org/linux/linux-nohup-command-tutorial/>

Linux 的 nohup 指令使用教學與範例，登出不中斷程式執行

<致謝>

摯友 博禕