

408410007鄭雅云

此學習重點: **signal(SIGHUP, SIG_IGN);** //告訴作業系統, 忽略『掛斷』的信號
有此行, 將終端機關掉, 程式仍繼續跑完; 註解掉, 將終端機關掉, 程式終止

註解1『freopen("/dev/null", "R+", stdin);』:

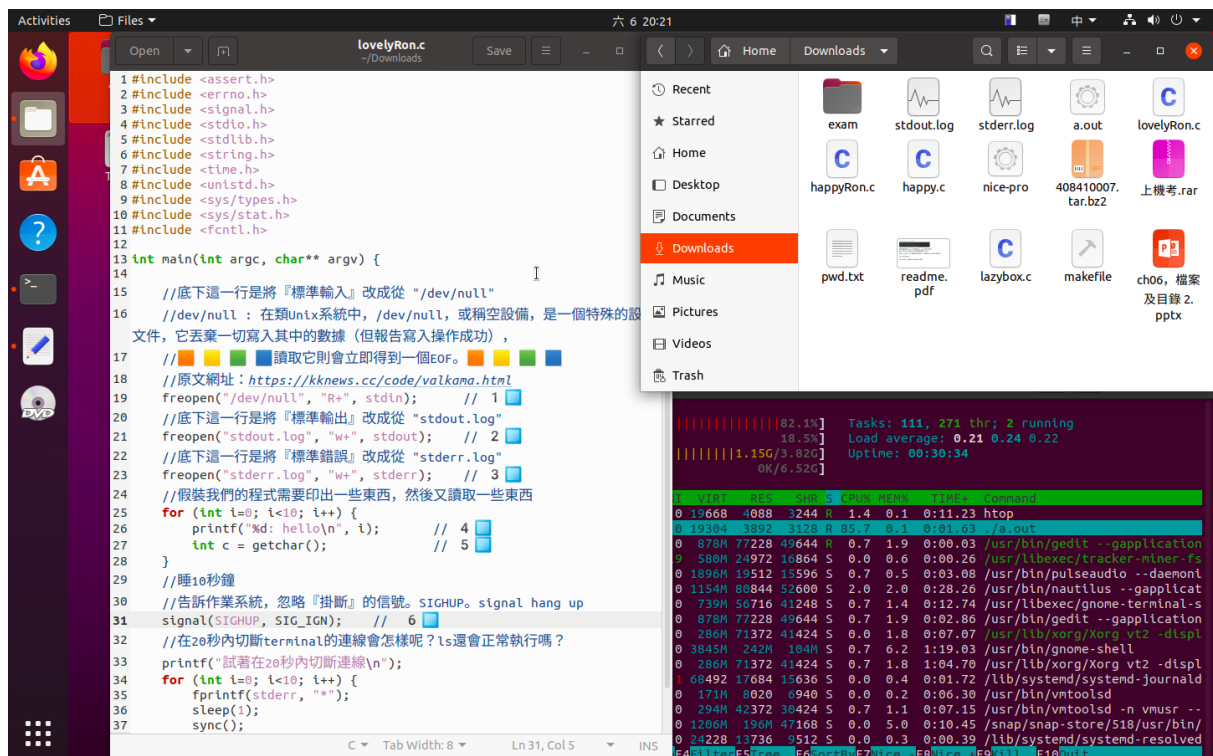
```
11 12 13 14 15 16
34
ariel@ariel-vm: ~/Downloads$ 11 12 13 14 15 16
3411: command not found
ariel@ariel-vm: ~/Downloads$ 34
34: command not found
ariel@ariel-vm: ~/Downloads$ 34
34: command not found
ariel@ariel-vm: ~/Downloads$ ./a.out
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15
ariel@ariel-vm: ~/Downloads$ 11 12 13 14 15
11: command not found
ariel@ariel-vm: ~/Downloads$ ./a.out
1 2 3 4 5 6 7 8 9 10
ariel@ariel-vm: ~/Downloads$ gcc lovelyRon.c
ariel@ariel-vm: ~/Downloads$ ./a.out
1 2 3 4 5 6 7 8 9 10
ariel@ariel-vm: ~/Downloads$ ./a.out
1 2 3 4 5 6 7 8 9 10
ariel@ariel-vm: ~/Downloads$ ./a.out
1 2 3 4 5 6 7 8 9 10
ariel@ariel-vm: ~/Downloads$ [ ]
24 //假裝我們的程式需要印出一些東西, 然後又讀取一些東西
25 for (int i=0; i<10; i++) {
26     printf("%d: hello\n", i); // 4
27     int c = getchar(); // 5
28 }
29 //睡10秒鐘
30 signal(SIGHUP, SIG_IGN); // 6
31 //在20秒內切斷terminal的連線會怎樣呢? ls還會正常執行
32 printf("試著在20秒內切斷連線\n");
33 for (int i=0; i<10; i++) {
34     fprintf(stderr, "%d\n", i);
35     sleep(1);
36     sync();
37 }
38 execlp("ls", "ls", "/", "-R", NULL); // 7
39
40
```

一開始產生stdout.log及stderr.log檔 (無內容)。在切斷連線前, 5 getchar()輸入10個數字後, 兩個檔案裡皆跑出7的ls(此時cpu使用多)皆有資料

4: 沒有印出hello

註解2『freopen("stdout.log", "w+", stdout);』:

註解6『signal(SIGHUP, SIG_IGN); //告訴作業系統, 忽略『掛斷』的信號』:



The screenshot shows a Linux desktop with a file manager and a terminal. The file manager displays a C program named 'lovelyRon.c' in the editor. The program includes headers for assert, errno, signal, stdio, stdlib, string, time, unistd, sys/types, sys/stat, andfcntl. The main function sets up file descriptors for stdin, stdout, and stderr, then enters a loop where it reads characters from stdin and prints them to stdout and stderr. It also includes a signal handler for SIGHUP that ignores the signal. The terminal window shows the output of the program, which is a series of characters (0-9) and a newline character, indicating that the program is running and processing input.

```
1 #include <assert.h>
2 #include <errno.h>
3 #include <signal.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7 #include <time.h>
8 #include <unistd.h>
9 #include <sys/types.h>
10 #include <sys/stat.h>
11 #include <fcntl.h>
12
13 int main(int argc, char** argv) {
14
15     //底下這一行是將『標準輸入』改成從 "/dev/null"
16     //dev/null : 在類Unix系統中, /dev/null, 或稱空設備, 是一個特殊的設備
17     //文件, 它丟棄一切寫入其中的數據 (但報告寫入操作成功),
18     //讀取它則會立即得到一個eof.
19     //原文網址: https://kknews.cc/code/valkana.html
20     freopen("/dev/null", "r+", stdin); // 1
21
22     //底下這一行是將『標準輸出』改成從 "stdout.log"
23     freopen("stdout.log", "w+", stdout); // 2
24
25     //底下這一行是將『標準錯誤』改成從 "stderr.log"
26     freopen("stderr.log", "w+", stderr); // 3
27
28     //假裝我們的程式需要印出一些東西, 然後又讀取一些東西
29     for (int i=0; i<10; i++) {
30         printf("%d: hello\n", i); // 4
31         int c = getchar(); // 5
32     }
33
34     //睡10秒鐘
35     //告訴作業系統, 忽略『掛斷』的信號. SIGHUP. signal hang up
36     signal(SIGHUP, SIG_IGN); // 6
37
38     //在20秒內切斷terminal的連線會怎樣呢? ls還會正常執行嗎?
39     printf("試著在20秒內切斷連線\n");
40     for (int i=0; i<10; i++) {
41         sleep(1);
42         fprintf(stderr, "%d\n", i);
43     }
44     sync();
45 }
```

一開始產生stdout.log及stderr.log檔 (無內容), 在5 getchar()輸入10個數字此時把終端機關掉, 產生的兩個檔案裡皆不會跑出7的ls (一樣保持無內容, 因為沒有忽略掛掉信號)、4:沒有印出hello