

System programming homework #11

王興彥 408410113

1. 使用『kill -l』指令，印出所有的 signal 的名字，針對你有興趣的五個 signal，解釋該 signal 的意義。

```
xingyan@kubuntu:~$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS     8) SIGFPE     9) SIGKILL    10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH   29) SIGIO      30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

1. SIGBUS

匯流排發生錯誤時，行程接收到一個 SIGBUS 訊號。舉例來說，記憶體存取對齊或者不存在對應的實體位址都會產生 SIGBUS 訊號。

2. SIGCHLD

當子行程終止、被中斷或被中斷後恢復時，SIGCHLD 訊號被傳送到行程。該訊號的一個常見用法是指示作業系統在子行程終止後清理其使用的資源，而不顯式呼叫等待系統呼叫。

3. SIGCONT

SIGCONT 訊號指示作業系統繼續(重新啟動)先前由 SIGSTOP 或 SIGTSTP 訊號暫停的行程。Unix 殼的作業控制是該訊號的一個重要應用。

4. SIGFPE

當行程執行了一個錯誤的算術運算時，例如被零除，訊號被傳送到一個行程。這可能包括整數被零除，以及整數在除結果中溢位(在 C 中只有 INT_MIN/-1、INT64_MIN/-1 和%-1 會觸發該行為)。注意該訊號與浮點數溢位無關。

5. SIGHUP

檢測到控制中斷掛起或者控制行程死亡時，行程會收到 SIGHUP。現在作業系統，該訊號通常意味著使用的虛擬終端已經被關閉。許多守護行程在接收到該訊號時，會多載他們的設定和重新打開紀錄檔檔案(logfiles)，而不是去結束程式。nohup 命令用於無視該訊號。

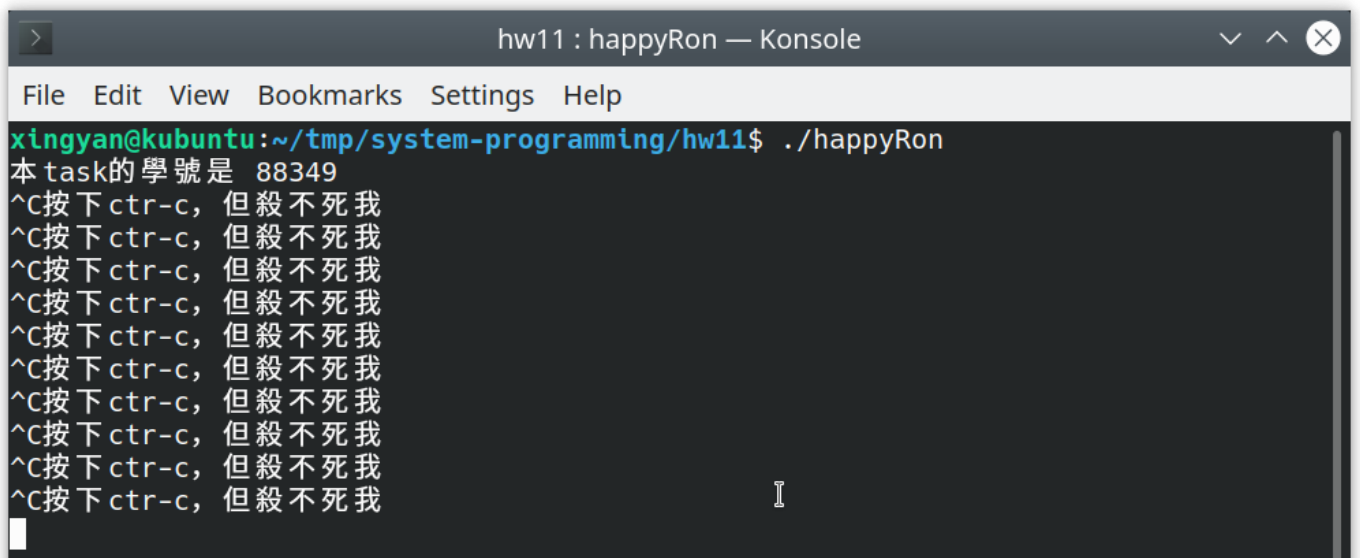
資料來源: <https://zh.wikipedia.org/wiki/Unix%E4%BF%A1%E5%8F%B7>

2. 請使用範例程式加以修改，攔截 SIGINT，並印出『按下 ctr-c，但殺不死我』。

```
signal(SIGINT, sighandler);
void sighandler(int signo) {
    if (signo == SIGINT) {
        printf("按下ctr-c，但殺不死我\n");
        //sleep(10); //假裝在ctr-c裡面運算了10秒鐘
    }
}
```

收到 SIGINT 的 signal 會呼叫 sighandler 印出訊息。

3. 承上題，如果快速的按下十次 ctr+c，會出現多少次『按下 ctr-c，但殺不死我』？



```
hw11 : happyRon — Konsole
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/tmp/system-programming/hw11$ ./happyRon
本task的學號是 88349
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
^C按下ctr-c，但殺不死我
```

會出現 10 次，因為印出訊息後 sighandler 結束，馬上就能繼續處理 SIGINT。

4. 承上題，在 SIGINT 的處理函數中，加入 sleep(10)，此時如果快速按下十次 ctr+c，會出現幾次『按下 ctr-c，但殺不死我』？

```
void sighandler(int signo) {
    if (signo == SIGINT) {
        printf("按下ctr-c，但殺不死我\n");
        sleep(10); //假裝在ctr-c裡面運算了10秒鐘
    }
}
```

```
hw11 : happyRon — Konsole
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/tmp/system-programming/hw11$ ./happyRon
本task的學號是 88250
^C按下ctr-c，但殺不死我
^C^C^C^C^C^C^C^C^C按下ctr-c，但殺不死我
```

總共會出現兩次，第一次是一開始攔截到的 SIGINT，然後第二個是當第一個 sighandler 還在處理時，所攔截的第二個 SIGINT，狀態是 pending，當第一個 sighandler 處理完成，會處理 pending 的 signal，而剩下的 signal 則會被遮罩掉。

5. 承上題，此時如果使用 fork+execv 讓 child 執行『ls -alh / -R』。當『ls -alh / -R』時，按下 ctr+c 時，會出現『按下 ctr-c，但殺不死我』嗎？

```
hw11 : happyRon — Konsole
File Edit View Bookmarks Settings Help
-rw----- 1 xingyan xingyan 93K 五 19 22:13 532ba7ee1af807ab_0
-rw----- 1 xingyan xingyan 61K 五 19 22:13 6444a3bb38259569_0
-rw----- 1 xingyan xingyan 16K 五 19 22:13 66942585d05c3680_0
-rw----- 1 xingyan xingyan 116K 五 19 22:13 8785296ec2722bd9_0
-rw----- 1 xingyan xingyan 61K 五 19 22:13 9f9a8aa789cb3c59_0
-rw----- 1 xingyan xingyan 93K 五 19 22:13 b411221d3369f37f_0
-rw----- 1 xingyan xingyan 62K 五 19 22:13 f71418cfce284879_0
-rw----- 1 xingyan xingyan 60K 五 19 22:13 fd429bc9190a98e7_0
-rw----- 1 xingyan xingyan 24 五 19 22:13 index
drwx----- 2 xingyan xingyan 4.0K 五 19 22:14 index-dir

'/home/xingyan/.config/google-chrome/Default/Service Worker/CacheStorage/8cbb992
fe0cd9ef960e69a214646bd270516a23e/bee90674-5487-49b0-a57a-c9c3fef54b23/index-dir
':
total 12K
drwx----- 2 xingyan xingyan 4.0K 五 19 22:14 .
drwx----- 3 xingyan xingyan 4.0K 五 19 22:13 ..
-rw----- 1 xingyan xingyan 312 五 19 22:14 the-real-index

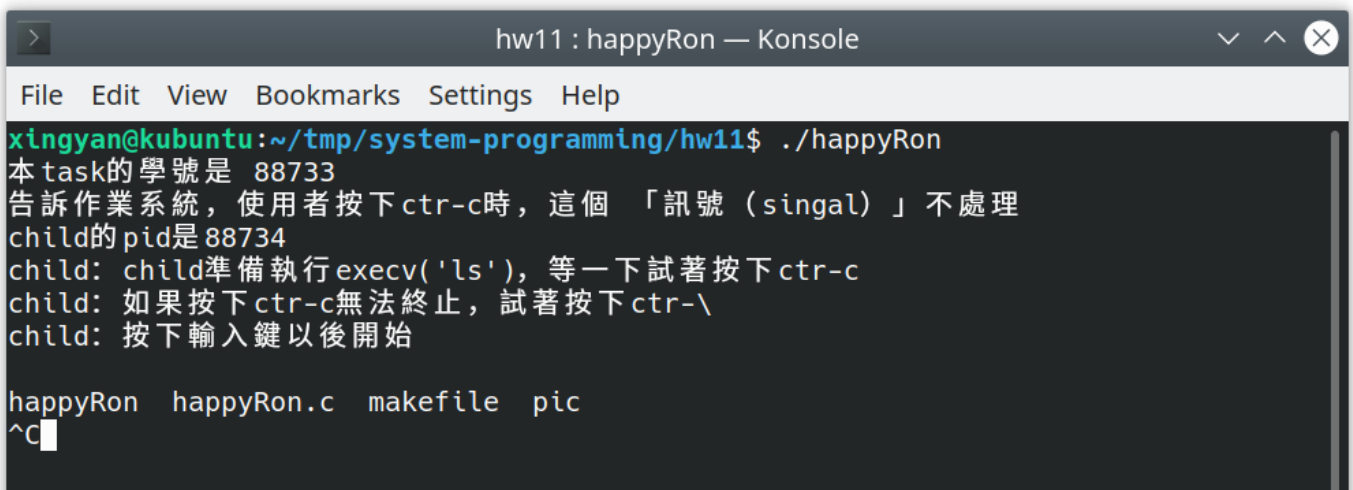
'/home/xingyan/.config/google-chrome/Default/Service Worker/CacheStorage/8cbb992
fe0cd9ef960e69a214646bd270516a23e/e2090eb3-d6ac-4cc9-9d83-3892896f2d6d':
total 57M
drwx----- 3 xingyan xingyan 4.0K 五 18 09:05 ^C按下ctr-c，但殺不死我
```

會。

6. 請使用範例程式加以修改，忽略 SIGINT (SIG_IGN) 。

```
//使用者按下ctr-c，OS直接忽略ctr-c，然後這個signal『『不會』』送給這個task
do {
    printf("告訴作業系統，使用者按下ctr-c時，這個「訊號(singal)」不處理\n");
    assert(signal(SIGINT, SIG_IGN)!=SIG_ERR);
}while(0);
```

7. 承上題，此時如果使用 fork+execv 讓 child 執行『ls』。當『ls』時，按下 ctr+c 時，可以終止『ls』的執行嗎？



```
hw11 : happyRon — Konsole
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/tmp/system-programming/hw11$ ./happyRon
本task的學號是 88733
告訴作業系統，使用者按下ctr-c時，這個「訊號(singal)」不處理
child的pid是88734
child: child準備執行execv('ls'), 等一下試著按下ctr-c
child: 如果按下ctr-c無法終止，試著按下ctr-\
child: 按下輸入鍵以後開始

happyRon happyRon.c makefile pic
^C
```

此時 SIGINT 已被忽略，所以不會終止執行。

Opinion:

學到 signal 的第一個感覺是很有趣，讓我們寫程式有更多技巧可以使用，但雖然它有些時候方便，卻也存在很多風險，主因是因為它是 asynchronous(非同步)的，這導致一個不小心容易造成 race conditions，沒有做好預防的話很可能會造成無法預期的結果，尤其是如果 handler 有 malloc 和 printf 這類的函式會更危險。