

作業十一：

學習目標：

在 UNIX 中，應用程式可以透過 system call 告知作業系統『某些事情』，

例如：『請寫出這個 buffer 裡面的所有資料』。這是由上（APP）往下

（OS）進行的 system call。

作業系統有時候需要通知應用程式發生了『某些事情』。例如『使用者按下

了 ctr-c』，這時候組合鍵『control + c』會被作業系統重新詮釋，變成

SIGINT 傳送給應用程式。

另外一種是當發生錯誤的時候，例如 SIGSEGV，通常代表指標存取錯誤。

也有一些是用來給系統管理人員使用的，例如：刪除程式（SIGKILL）和

暫停程式（SIGSTOP）

題目：

請從 dropbox 下載 happyRon.c，並完成下列事項

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（可以從網路上找到資料，並複製貼上）
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』
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』

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』？
5. 承上題，此時如果使用 `fork+execv` 讓 child 執行『ls』。當『ls』時，按下 `ctr+c` 時，會出現『按下 `ctr-c`，但殺不死我』嗎？
6. 請使用範例程式加以修改，忽略 SIGINT (SIG_IGN)
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？

報告：

1. 請將題目 1~7 寫成報告，所有的題目都要附上一張與該題目有直接關係的截圖，繳交的形式是 pdf
2. 在上述報告的最後面寫上你對 signal 的想法。signal 是一個比較難以理解『系統程式設計』，請你以 50 個字以上的描述，說明一些事項，例如（但不局限於）：

甲、從文字介面的考量，需要 OS 特別詮釋組合鍵嗎？

乙、Signal 的執行流程，與主程式的流程『是彼此獨立的』，有一點像是，『額外的執行緒 (thread)』。但就程式的角度來說並不是獨立的，最後還是要將 signal 的資訊傳遞給 main 程序（或者 main 所呼叫的 function 等等）。這真是很麻煩

丙、從物件導向的觀念，事件驅動可以直接的整合到程式語言中嗎？

丁、從「事件」的概念，可以將所有的東西都變成事件嗎？例如：

system call 是 APP 傳遞給 OS 的事件，OS 收到事件以後，處理完，再回覆給 APP『已完成』。Signal 則是反過來。如果朝著這樣的方式去思考，所有的東西都是 non-blocking。這樣寫程式會比較好寫嗎？會比較有彈性嗎？效能會比較高嗎？

戊、以上的解答方向，只是供大家做為參考，等到修完 OS 後才會比較有感覺。

繳交：

1. 繳交 pdf 檔案。繳交到 ecourse2 上
2. 不能遲交
3. 再次提醒，助教會將所有人的作業於 dropbox 上公開
4. 如果真的不會寫，記得去請教朋友。在你的報告上寫你請教了誰即可。
5. 繳交期限：2021/5/25 早上八點