系統程式設計 System Programming 2021

電機四 406415089 宋 O 天

Hw11 Signal

<回答問題>

<1>用『kill-l』指令，印出所有 signal 名字，挑五個 signal 解釋該它的的意義

```
nash@SleepyCat:~/Desktop/hw$ 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
```

SIGCHLD 訊號

子程序狀態發生變化(子程序結束)產生該訊號，父程序需要使用 wait 呼叫來等待子程序結束並回收它避免殭屍程序。

SIGPIPE 訊號

嘗試傳送資料到一個已關閉的 socket 上兩次，就會出現此訊號，例如用 TCP 的 socket 程式設計，伺服器是不能知道客戶機什麼時候已經關閉了 socket，導致還在向已關閉的 socket 上傳送。

SIGIO 訊號

異步通知的意思是：一旦設備就緒，則主動通知應用程序，這樣一來，應用程序就不需要查詢設備狀態，非常類似于硬件上"中斷"的概念，比較準確的說法是"信號驅動(SIGIO)的異步 I/O"。

SIGTRAP 訊號

由斷點指令或**陷阱（trap）指令**產生，由 debugger 使用。

SIGBUS 訊號

非法地址、內存地址 alignment 出錯。比如訪問一個 4 bytes 的整數, 但其地址不是 4 的倍數。它和 SIGSEGV 的區別在於後者是對合法儲存地址的非法訪問觸發(訪問不屬於自己的存儲空間)

<2>攔截 SIGINT，並印出『按下 ctr-c，但殺不死我』

如下，我猜老師的意思應該是這樣

```
54      //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
55      do {
56         // printf("告訴作業系統, 使用者按下ctr-c時, 這個 「訊號 (singal) 」 不處理\n");
57         //assert(signal(SIGINT, SIG_IGN)!=SIG_ERR);
58          signal(SIGINT,sighandler);
59      }while(0);
```

<3>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』

10 次

<4>在 SIGINT 的處理函數中，加入 sleep(10)，此時快速按下十次 ctr+c，會出現幾次『按下 ctr-c，但殺不死我』？

```
^C按下ctr-c，但殺不死我
^C^C^C^C^C^C^C^C^C^C
```

1 次，因為 sleep : the fuctions causes the calling process to be suspended　(APUE)

<5>承上，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？

```
nash@SleepyCat: ~/Desktop/hw    nash@SleepyCat: ~/Desktop/hw    nash@SleepyCat: ~/Desktop/hw
-rw-rw-r--  1 nash nash 4.8K  五  22 19:22 export.sh
-rw-rw-r--  1 nash nash 8.7K  五  22 19:22 .flake8
drwxrwxr-x  9 nash nash 4.0K  五  22 19:22 .git
drwxrwxr-x  2 nash nash 4.0K  五  22 19:22 .github
-rw-rw-r--  1 nash nash 1005  五  22 19:22 .gitignore
-rw-rw-r--  1 nash nash 4.1K  五  22 19:22 .gitlab-ci.yml
-rw-rw-r--  1 nash nash 2.3K  五  22 19:22 .gitmodules
-rw-rw-r--  1 nash nash  529  五  22 19:22 install.bat
-rwxrwxr-x  1 nash nash  357  五  22 19:22 install.sh
-rw-rw-r--  1 nash nash 9.9K  五  22 19:22 Kconfig
-rw-rw-r--  1 nash nash  12K  五  22 19:22 LICENSE
drwxrwxr-x  2 nash nash 4.0K  五  22 19:22 make
-rw-rw-r--  1 nash nash 5.7K  五  22 19:22 README.md
-rw-rw-r--  1 nash nash  538  五  22 19:22 .readthedocs.yml
-rw-rw-r--  1 nash nash  462  五  22 19:22 requirements.txt
-rw-rw-r--  1 nash nash 1.8K  五  22 19:22 sdkconfig.rename
-rw-rw-r--  1 nash nash 2.8K  五  22 19:22 SUPPORT_POLICY.md
drwxrwxr-x 19 nash nash 4.0K  五  22 19:22 tools
-rw-rw-r--  1 nash nash  133  五  22 19:22 .travis.yml

/home/nash/Desktop/MicroPython/micropython-master/ports/esp32/esp-idf/components:
^C按下ctr-c，但殺不死我
get a signal named '17', 'Child exited'
```

會，子行程會複製父行程的記憶體，所以處理訊號的方式跟著變動。

<6>請使用範例程式加以修改忽略 SIGINT（SIG_IGN）

打開老師寫的 code

```
//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);
    //signal(SIGINT,sighandler);
}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

不行，ctrl+\才可以，因為 signal SIGINT 已經被忽略了

```
-rw-rw-r--  1 nash nash 2.0K  五  21 19:55 bufhelper.c
-rw-rw-r--  1 nash nash 1.5K  五  21 19:55 bufhelper.h
-rw-rw-r--  1 nash nash  14K  五  21 19:55 can.c
-rw-rw-r--  1 nash nash 3.1K  五  21 19:55 can.h
-rw-rw-r--  1 nash nash  18K  五  21 19:55 dac.c
-rw-rw-r--  1 nash nash 1.4K  五  21 19:55 dac.h
-rw-rw-r--  1 nash nash  44K  五  21 19:55 dma.c
-rw-rw-r--  1 nash nash 4.3K  五  21 19:55 dma.h
-rw-rw-r--  1 nash nash  26K  五  21 19:55 eth.c
-rw-rw-r--  1 nash nash 1.6K  五  21 19:55 eth.h^\get a signal named '3', 'Quit'
get a signal named '17', 'Child exited'
```

對於 signal 的想法

查了一下 non-blocking 和 blocking 的優缺點，
在非阻塞 IO 模型中，行程需要不斷地詢問內核數據是否就緒，也就是說 non-blocking 不會交出 CPU，而會一直占用 CPU。另外，OS 感覺不需要去裡組合鍵，只要知道信號式哪一個就可以了，ctrl+c 感覺是 bash 處理的。Signal 真的好難啊。

<參考資料>

https://www.itread01.com/content/1546180142.html
Linux 程序與訊號——SIGCHLD 訊號、kill 和 raise 函式以及 alarm 函式
https://www.itread01.com/content/1549922071.html
linux socket 程式設計 出現訊號 SIGPIPE，分析及解決
https://blog.csdn.net/xiaopohaibebo/article/details/8088672?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.baidujs&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.baidujs
信号驱动(SIGIO)的异步 I/O
https://blog.csdn.net/u014470361/article/details/83591513
linux——signal 信号
https://b8807053.pixnet.net/blog/post/327113148-linux-%E4%BF%A1%E8%99%9Fsignal%E8%99%95%E7%90%86%E6%A9%9F%E5%88%B6
Linux 信號 signal 處理機制

<致謝>

羅 O 五老師
摯友 博禕