

系統程式 HW11

410410060 資工二 林柔均

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

```
A ~ > kill -l
HUP INT QUIT ILL TRAP IOT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM TERM STKFLT CHLD CONT STOP TSTP
TTIN TTOU URG XCPU XFSZ VTALRM PROF WINCH POLL PWR SYS
```

- `Int(SIGINT)`: 使用者中斷 process(ctrl + c)
- `KILL(SIGKILL)`: 終止 process
- `ILL(SIGILL)`: 偵測到 process 執行格式錯誤獲非法的指令
- `STP(SISTP)`: 使用者暫停 process (ctrl +z)
- `Int(SIGINT)`: 使用者退出 process

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』

```
73      signal(SIGINT, sighandler);
```

```
29  void sighandler(int signo) {
30      if (signo == SIGINT) {
31          printf("按下ctr-c, 但殺不死我\n");
32          //sleep(10); //假裝在ctr-c裡面運算了10秒鐘
33      }
34      if (signo == SIGTSTP)
35          printf("按下ctr-z, 但殺不死我\n");
36      if (signo == SIGKILL)
37          printf("按下ctr-z, 但殺不死我\n");
38  }
```


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』？

```
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  
}
```

```
~/De/h/Sy/hw11 > ./happyRon  
本task的學號是 10324  
SIGKILL是直接殺掉一個task  
無法改變SIGKILL的行為  
SIGKILL的問題，具體來說是： : Invalid argument  
無法改變SIGSTOP的行為  
SIGSTOP的問題，具體來說是： : Invalid argument  
child的pid是10325  
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  
^C按下ctr-c，但殺不死我  
^C^C^C^C^C^C^C^C^C
```

ANS: 1次

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？

```
//測試在fork的環境下是child是否繼承parent的signal的設定  
int child_pid = fork();  
if (child_pid == 0) { //child  
    printf("child: child準備執行execv('ls'), 等一下試著按下ctr-c\n");  
    printf("child: 如果按下ctr-c無法終止，試著按下ctr-\\n");  
    printf("child: 按下輸入鍵以後開始\n");  
    char c = getchar();  
    execlp("ls", "ls", "/", "-alhR", NULL);  
}
```

```
drwx----- 2 rogewood users 4.0K Oct 17 2022 index-dir'  
'/home/rogewood/.config/google-chrome/Default/Service Worker/CacheStorage/2b7e01c68377c54595891d9  
b971/d1f7e403-470b-4fd1-810f-dd70cb9d637f/index-dir':  
total 12K  
drwx----- 2 rogewood users 4.0K Oct 17 2022 .  
drwx-----^C按下ctr-c，但殺不死我
```

ANS: 會出現『按下ctr-c，但殺不死我』

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);
```

```
~/De/h/Sy/hw11 > ./happyRon  
本task的學號是 9626  
告訴作業系統，使用者按下ctr-c時，這個「訊號(singal)」不處理  
SIGKILL是直接殺掉一個task  
無法改變SIGKILL的行為  
SIGKILL的問題，具體來說是: : Invalid argument  
無法改變SIGSTOP的行為  
SIGSTOP的問題，具體來說是: : Invalid argument  
child的pid是9627  
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  
^C^C^C^C^C^C
```

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？

```

-rw-r--r-- 1 rogewood users 6.2K Oct 19 2022 b398361f08a0e8acbe148bba6702350c
-rw-r--r-- 1 rogewood users 4.6K Oct 19 2022 b3b9624940ae7a43ab2aedb086e1e20a
-rw-r--r-- 1 rogewood users 4.7K Oct 19 2022 b3d03dd747528c09cbd538597970dda8
-rw-r--r-- 1 rogewood users 4.4K Oct 19 2022 b3d119e6483a19c4e2b83b48fdf936b7
-rw-r--r-- 1 rogewood users 4.6K Oct 19 2022 b3d1cef0a151d56780df9cb7491b4e87
-rw-r--r-- 1 rogewood users 8.1K Oct 19 2022 b3d31c3fce7a01f5839e28b6fff0aa77
-rw-r--r-- 1 rogewood users 8.2K Oct 19 2022 b3d3f1e71e73a02c7e2c654718a50463
-rw-r--r-- 1 rogewood users 53K Oct 19 2022 b3dc651168caee6177944edbbd7bdf75
-rw-r--r-- 1 rogewood users 5.5K Oct 19 2022 b3e734b276b3607d92579d96798d5c34
-rw-r--r-- 1 rogewood users 8.0K Oct 19 2022 b3fbdd7ce7b43c2cad2602745a0c7de9

/home/rogewood/Desktop/project/Unity/Client-side-prediction/Library/Artifacts/b4:
total 500K
drwxr-xr-x  2 rogewood users 4.0K Oct 19 2022 .
drwxr-xr-x 258 rogewood users 4.0K Oct 19 2022 ..
-rw-r--r--  1 rogewood users 22K Oct 19 2022 b419b5dde648a3912a0a65acfacc1275
-rw-r--r--  1 rogewood users 21K Oct 19 2022 b42f41f8dffefcf106465b6cadb4688d
-rw-r--r--  1 rogewood users 84K Oct 19 2022 b42fa530d60362ca4307aae68fac1b01
-rw-r--r--  1 rogewood users 6.8K Oct 19 2022 b431f0e1770202cf2225083f7bdf7e2b

```

ANS: 不行

8.對signal的想法

signal 和一般的程序不同，寫成是的感覺不像程序本身，而是被覆加上去，已OS的角度去看 process 的狀況。而非同步的性質當在處理一個信號時，會收到另一個信號，不像以前的順序下來的寫法。

註: 有參考學長姐的作業