Signal

한양대학교 소프트웨어학부

Dept. of Division of Computer Science

Hanyang University

Signal – sigprocmask

사용법

```
#include <signal.h>

int sigprocmask(int how, const sigset_t *set, sigset_t *oset);
```

- 현재 블록 된 시그널들을 변경 시키기 위해서 사용한다
- 호출의 행위는 how 값들에 대해서 의존적이 된다
  - SIG_BLOCK
    - signal set에 설정된 시그널을 블록 시그널셋에 추가 시킨다
  - SIG_UNBLOCK
    - signal set의 시그널을 현재의 블럭된 시그널에서 삭제
  - SIG_SETMASK
    - signal set의 시그널을 블럭화된 시그널로 지정

- oset이 null이 아니면, 시그널 마스크의 이전 값은 oset에 저장된다

Example – sigprocmask 1

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <sys/types.h>
#include <string.h>
#include <errno.h>

int count;

void catch_sigint(int signum)
{
        printf("\n(count = %d) CTRL-C pressed!\n", count);
        return;
}

int main(int argc, char* argv[])
{
        sigset_t masksets;
        sigfillset(&masksets);
        signal(SIGINT, catch_sigint);

        for (count = 0; count < 30; count++) {
                if (count < 10) // 모든 Signal 봉쇄
                        sigprocmask(SIG_SETMASK, &masksets, NULL);
                else if (count >= 10 && count < 20) // 모든 Signal 개방
                        sigprocmask(SIG_UNBLOCK, &masksets, NULL);
                else if (count >= 20) {
                        sigemptyset(&masksets);
                        sigaddset(&masksets, SIGINT);
                        sigprocmask(SIG_BLOCK, &masksets, NULL);
                }

                printf("test: %d\n", count);
                sleep(1);
        }
        return 0;
}
```

Example – sigprocmask 2

```
[TA3@localhost lab9]$ ./sigprocmask
test: 0
test: 1
^Ctest: 2
test: 3
test: 4
test: 5
test: 6
^Ctest: 7
test: 8
test: 9

(count = 10) CTRL-C pressed!
test: 10
test: 11
test: 12
test: 13
^C
(count = 13) CTRL-C pressed!
test: 14
test: 15
test: 16
test: 17
test: 18
^C
(count = 18) CTRL-C pressed!
test: 19
test: 20
test: 21
test: 22
test: 23
^Ctest: 24
test: 25
^Ctest: 26
test: 27
^Ctest: 28
test: 29
[TA3@localhost lab9]$
```

Signal – kill

사용법

```
#include <sys/types.h>
#include <signal.h>

int kill (pid_t pid, int sig);
```

- pid> 0 : pid를 가진 프로세스에게 시그널을 보낸다
- pid== 0 : 시그널은 보내는 프로세스와 같은 프로세스 그룹에 속하는 모든 프로세스에 보내짐
- pid== -1 & not superuser: 프로세스의 유효사용자 식별번호와 같은 모든 프로세스에 보내짐
- pid== -1 & superuser: 특수한 시스템 프로세스를 제외한 모든 프로세스에 보내짐
- pid< -1 : 프로세스의 그룹 식별번호가 pid의 절대값과 같은 모든프로세스에보내짐

Signal – pause

사용법

```
#include <unistd.h>

int pause (void);
```

- 신호를 받을 때까지 호출 프로세스를 중지시킨다

```
 1 #include <stdio.h>
 2 #include <unistd.h>
 3 #include <signal.h>
 4
 5 void sig_handler(int signum)
 6 {
 7         printf("\nSIGINT occur \n");
 8 }
 9
10 int main(void)
11 {
12         printf("hello world\n");
13         signal(SIGINT, sig_handler);
14         pause();
15         printf("Interupt\n");
16 }
```

```
[TA3@localhost lab9]$ ./pause
hello world
^C
SIGINT occur
Interupt
[TA3@localhost lab9]$
```

Example – kill 1

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <signal.h>

int ntimes = 0;
void p_action(int signum)
{
        printf("Parent caught signal #%d\n", ++ntimes);
}

void c_action(int signum)
{
        printf("Child caught signal #%d\n", ++ntimes);
}


int main()
{
        pid_t pid, ppid;
        void p_action(int), c_action(int);
        static struct sigaction pact, cact;

        /* 부모를 위해 SIGUSR1 행동을 지정 */
        pact.sa_handler = p_action;
        sigaction(SIGUSR1, &pact, NULL);
```

Example – kill 2

```
29          switch(pid = fork()) {
30                  case -1:
31                          perror("fork error");
32                          exit(1);
33
34                  case 0:
35                          /* 자식을 위해 SIGUSR1 행동을 지정 */
36                          cact.sa_handler = c_action;
37                          sigaction(SIGUSR1, &cact, NULL);
38
39                          ppid = getppid();
40
41                          for (;;) {
42                                  sleep(1);
43                                  kill(ppid, SIGUSR1);
44                                  pause();
45                          }
46                          /* 퇴장(exit)하지 않음 */
47
48                  default:
49                          for (;;) {
50                                  pause();
51                                  sleep(1);
52                                  kill(pid, SIGUSR1);
53                          }
54                          /* 퇴장(exit)하지 않음 */
55          }
56  }
```

```
[TA3@localhost lab9]$ ./ex1
Parent caught signal #1
Child caught signal #1
Parent caught signal #2
Child caught signal #2
Parent caught signal #3
Child caught signal #3
Parent caught signal #4
Child caught signal #4
^C
[TA3@localhost lab9]$
```

Signal – raise/alarm

사용법 - raise

```
#include <signal.h>
int raise (int sig);
```

현재프로세스에게 sig번호를 가지는 시그널을 전달한다

사용법 - alarm

```
#include <signal.h>
unsigned int alarm (unsigned int secs);
```

alarm은 secs초 후에 프로세스에 SIGALRM을 전달한다
alarm(0) 을 호출하면 alarm이 꺼진다
이전에 설정된 알람이 시그널을 전달할 때까지 남은시간을
초 단위 숫자로 반환하거나, 이전에 설정된 알람이 없을 경우
0을 되돌려준다

Example - alarm

```c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

void myalarm()
{
        printf("ding dong dang\n");
}

int main(void)
{
        int i = 0;

        printf("alarm setting\n");
        signal(SIGALRM, myalarm);

        alarm(1);
        while (i < 5) {
                printf("ok\n");
                pause();
                alarm(2);
                i++;
        }
}
```

```
[TA3@localhost lab9]$ ./ex2
alarm setting
ok
ding dong dang
ok
ding dong dang
ok
ding dong dang
ok
ding dong dang
ok
ding dong dang
[TA3@localhost lab9]$
```

실습 과제 newalarm

- alarm 예제에서 alarm함수를 대신 kill, raise 등을 이용하여 void newalarm (int secs)을 작성하시오!

  (기존 코드 중 main함수 부분은 alarm(n) → newalarm(n) 부분을 제외하고 절대 건드리지 않도록 함)

```
void myalarm();                 // signal handler
void newalarm(int secs);        // secs초 후 SIGALRM발생

int main() { // alarm 예제에서
    alarm(n) → newalarm(n) 으로 교체
}
```

Q & A

- Thank you :)