
Signals

System Programming

gcc 사용하기

□ 책자(이론) 예제 코드 컴파일 및 실행 해보기

- Prog. 10. 2
- Prog. 1. 4 을 아래와 같이 수정해보기
 - 이 프로그램을 먼저 수행해서 Ctrl-C로 인터럽트 신호 보내서 종료되는 것 확인
 - 이 프로그램에 SIGINT에 대한 함수 sig_int를 등록하고(Prog.1.10 참고) Ctrl-C를 입력하면, 종료되지 않고 Interrupt만 화면에 출력 후 계속 수행되도록 하기.
 - ✓ 이론 수업의 Interrupted system call 내용을 참고하기
 - 이후에 종료를 원하면 Ctrl-D를 입력하기
- Prog. 10. 8 & Prog. 10.9
 - 컴파일할 때, 두개의 c 소스코드를 같이 컴파일 하기 (예) `gcc -o test prog10_8.c prog10_9.c`
 - 실행파일을 실행하고 2-3초 정도 지난후에 Ctrl-C 입력하기

Process Control

□ Prog. 10.2 코드

```
#include "apue.h"

static void sig_usr(int); /* one handler for both signals */

int
main(void)
{
    if (signal(SIGUSR1, sig_usr) == SIG_ERR)
        err_sys("can't catch SIGUSR1");
    if (signal(SIGUSR2, sig_usr) == SIG_ERR)
        err_sys("can't catch SIGUSR2");
    for ( ; ; )
        pause();
}

static void
sig_usr(int signo) /* argument is signal number */
{
    if (signo == SIGUSR1)
        printf("received SIGUSR1\n");
    else if (signo == SIGUSR2)
        printf("received SIGUSR2\n");
    else
        err_dump("received signal %d\n", signo);
}
```

Figure 10.2 Simple program to catch SIGUSR1 and SIGUSR2

Process Control

□ Prog. 1.4 코드

```
#include "apue.h"
#define BUFSIZE 4096
int main(void)
{
    int  n;
    char buf[BUFSIZE];
    while ((n = read(STDIN_FILENO, buf, BUFSIZE)) > 0)
        if (write(STDOUT_FILENO, buf, n) != n)
            err_sys("write error");

    if (n < 0)
        err_sys("read error");

    exit(0);
}
```

Process Control

□ Prog. 10.8 코드

```
#include <setjmp.h>
#include <signal.h>
#include <unistd.h>

static jmp_buf env_alm;

static void
sig_alm(int signo)
{
    longjmp(env_alm, 1);
}

unsigned int
sleep2(unsigned int seconds)
{
    if (signal(SIGALRM, sig_alm) == SIG_ERR)
        return(seconds);
    if (setjmp(env_alm) == 0) {
        alarm(seconds);      /* start the timer */
        pause();             /* next caught signal wakes us up */
    }
    return(alarm(0));        /* turn off timer, return unslept time */
}
```

Figure 10.8 Another (imperfect) implementation of sleep

Process Control

□ Prog. 10.9 코드

```
#include "apue.h"

unsigned int    sleep2(unsigned int);
static void     sig_int(int);

int
main(void)
{
    unsigned int    unslept;

    if (signal(SIGINT, sig_int) == SIG_ERR)
        err_sys("signal(SIGINT) error");
    unslept = sleep2(5);
    printf("sleep2 returned: %u\n", unslept);
    exit(0);
}

static void
sig_int(int signo)
{
    int            i, j;
    volatile int    k;

    /*
     * Tune these loops to run for more than 5 seconds
     * on whatever system this test program is run.
     */
    printf("\nsig_int starting\n");
    for (i = 0; i < 300000; i++)
        for (j = 0; j < 4000; j++)
            k += i * j;
    printf("sig_int finished\n");
}
```

Figure 10.9 Calling sleep2 from a program that catches other signals

Process Control

□ Prog.10.2 실행

```
$ ./a.out &  
[1]          7216  
$ kill -USR1 7216  
received SIGUSR1  
$ kill -USR2 7216  
received SIGUSR2  
$ kill 7216  
[1]+  Terminated
```

./a.out

start process in background

job-control shell prints job number and process ID

send it SIGUSR1

send it SIGUSR2

now send it SIGTERM

Process Control

□ Prog.1. 4 수정 프로그램의 실행

```
sjhong@ubuntu: ~/sysprog
sjhong@ubuntu:~/sysprog$ ./mycat
sdfc
sdfc
ee
ee
wegesfr
wegesfr
sdf
sdf
^CInterrupt!!
sdfe
sdfe
svdvse
svdvse
^CInterrupt!!
dsf
dsf
read error: Interrupted system call
sjhong@ubuntu:~/sysprog$
```


Process Control

❑ Prog.10.8 & 9 실행

```
$ ./a.out  
^C  
sig_int starting  
sleep2 returned: 0
```

we type the interrupt character

Thank you for your attention !!

Q and A