

리눅스 시스템 프로그래밍

day7 실습 1



고강태

010-8269-3535


james@thinkbee.kr



<https://www.linkedin.com/in/thinkbeekr/>



<http://www.facebook.com/gangtai.goh>



day7 실습

실습: ex_atexitmsg.c

day7/ex_atexitmsg.c

전역 변수로 임의의 파일을 하나 열어 내용을 출력한다. 그리고 종료 직전 `atexit()` 를 사용해 등록한 핸들러를 이용해 파일을 닫으며 메시지를 출력하도록 해보자.

실습: ex_sig_kill.c

day7/ex_sig_kill.c

자식 프로세스를 생성후 SIGINT 를 발생해 종료후 부모 프로세스도 종료하게 하자.

실습: ex_sigint_msg.c

day7/ex_sigint_msg.c


sigaction 함수를 이용해서 사용자가 인터럽트 키(Ctrl+C)를 입력하면 임의 함수가 호출되고 파일에 종료 메시지를 "ex_sigint_out.txt"에 저장하도록 프로그램을 작성하자.

실습: ex_sigblock_int.c

day7/ex_sigblock_int.c

메인 프로그램이 무한반복으로 실행되고, 1초가 대기후 "메인 프로그램이 실행중입니다..."를 출력한다.

단, 모든 시그널은 블록하고 (sigfillset()), SIGINT 를 수신시 핸들러를 호출해서 수행한다. 이때 Ctrl+W (quit), Ctrl+z(멈춤) 키를 누르면 어떻게 되는지 확인해 보자.



day7 - Sample

실습: ex_atexitmsg.c

day7/ex_atexitmsg.c

SIGINT 를 발생해 종료

```
#include <unistd.h> #include <fcntl.h>

int filedes;
void final(void);

int main() {
    ssize_t nread;
    char buffer[1024];

    atexit(end_msg);

    filedes = open("test.txt", O_RDONLY);
    while((nread = read(filedes, buffer, 1024)) > 0)
    {
        printf("%s", buffer);
    }

    //exit(1);
}
```

```
void end_msg(void)
{
    printf("close filedes\n");
    close(filedes);
}
```


실습: ex_sig_kill.c

day7/ex_sig_kill.c

SIGINT 를 발생해 종료

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

main() {
    pid_t pid;
    int count = 5;

    if((pid = fork()) > 0) {
        printf("[parent] hello\n");
        pause();
        printf("[parent] bye!\n");
    }
    else if(pid == 0) {
        printf("[child] hello\n");
        sleep(1);
        kill(getppid(), SIGINT);
        printf("[child] bye\n");
    }
    else
        printf("fail to fork\n");
}
```

실습: ex_sigint_msg.c

day7/ex_sigint_msg.c

SIGINT 를 잡아내 메시지 저장

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

```
int flag = 1;
void int_handler(int);
```

```
main() {
    static struct sigaction act;

    act.sa_handler = int_handler;
    sigfillset(&(act.sa_mask));
    sigaction(SIGINT, &act, NULL);

    while(flag) {
        printf("in loop...\n");
        sleep(1);
    }
}
```

```
void int_handler(int signum) {
    int fd;
    char *msg = "Interrupted by SIGINT...\n";

    fd = open("ex_sigint_out.txt",
              O_CREAT | O_RDWR, 0644);
    write(fd, msg, strlen(msg));
    close(fd);

    flag = 0;
}
```

실습: ex_sigblock_int.c

SIGINT 를 잡아내 메시지 저장

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

```
int main( void)
{
    struct sigaction act;

    act.sa_handler = sigint_handler;
    sigfillset( &act.sa_mask);

    sigaction( SIGINT, &act, NULL);

    while(1 )
    {
        printf( "메인 프로그램을 실행중입니다...\n");
        sleep( 1);
    }
}
```

```
void sigint_handler( int signo) {
    int    ndx;

    printf( "Ctrl-C 키가 눌렀다.\n");
    printf( "3초간 대기중, Ctrl-\(Quit) 키 눌러

    for ( ndx = 3; 0 < ndx; ndx--){
        printf( "%d 초 남았습니다.\n", ndx);
        sleep( 1);
    }
}
```