

**TI DSP, MCU 및 Xilinx Zynq  
FPGA  
프로그래밍 전문가 과정**

강사 – Innova Lee(이상훈)

[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

학생 – 문한나

[mhn97@naver.com](mailto:mhn97@naver.com)

### 예제 1)

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

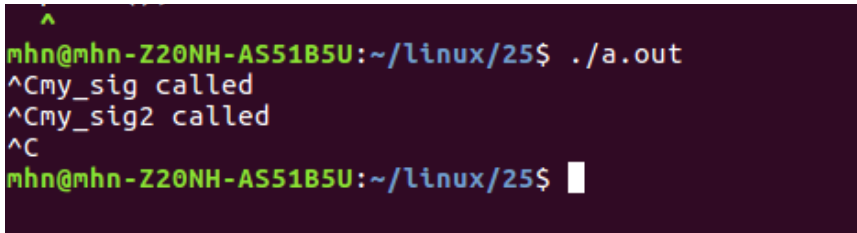
void my_sig(int signo){
    printf("my_sig called\n");
}

void my_sig2(int signo){
    printf("my_sig2 called\n");
}

int main(void){

    void (*old_p)(int); //함수포인터
    void (*old_p2)(int);
    old_p = signal(SIGINT,my_sig); //행동지침등록 my_sig 동작
    pause();
    old_p2 = signal(SIGINT,my_sig2); //my_sig2 동작
    pause();
    old_p2 = signal(SIGINT,old_p); //시그널은 기본적으로 자기자신이 전에 등록해놓은거(한단계전)리턴
    old_p 에는 null, old_p2 는 my_sig
    pause();
    for(;;)
        pause();

    return 0;
}
// SIGINT 는 ctrl+c 이다
```



```
mhn@mhn-Z20NH-AS51B5U:~/linux/25$ ./a.out
^Cmy_sig called
^Cmy_sig2 called
^C
mhn@mhn-Z20NH-AS51B5U:~/linux/25$
```

### 예제 2)

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

int main(void){

    signal(SIGINT,SIG_IGN);
    pause();
    return 0;

}
//만약에 SIGINT 가 나오면 무시하라는 뜻이다. 결국 kill -2 로는 안죽는다.
(SIGINT = SIGNUM2 번)
```

```

mhn@mhn-Z20NH-AS51B5U:~/linux/25$ kill -2 4341

mhn@mhn-Z20NH-AS51B5U:~/linux/25$ ps -ef | grep a.out
mhn      4378  4341  0 10:02 pts/2    00:00:00 ./a.out
mhn      4570  4341  0 10:19 pts/2    00:00:00 grep --color=auto a.out
mhn@mhn-Z20NH-AS51B5U:~/linux/25$ kill -2 4378
mhn@mhn-Z20NH-AS51B5U:~/linux/25$ ps -ef | grep a.out
mhn      4378  4341  0 10:02 pts/2    00:00:00 ./a.out
mhn      4574  4341  0 10:19 pts/2    00:00:00 grep --color=auto a.out
mhn@mhn-Z20NH-AS51B5U:~/linux/25$

```

그래서 kill-9 pid 로 죽였다

```

mhn@mhn-Z20NH-AS51B5U:~/linux/25$ kill -9 4378
mhn@mhn-Z20NH-AS51B5U:~/linux/25$ ps -ef | grep a.out
mhn      4607  4341  0 10:20 pts/2    00:00:00 grep --color=auto a.out
[1]+  Killed                  ./a.out
mhn@mhn-Z20NH-AS51B5U:~/linux/25$

```

### 예제 3)

```

#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <stdio.h>

```

```
void arr(int ret,char *buf){
```

```
if(ret = read(0,buf,sizeof(buf)) > 0)
```

```
goto err;//스택을 해제할 수 있는 능력이 없음 그러니까 메인으로 갈 수 없음
```

```
}
```

```
int main(void){
```

```
int ret;
```

```
char buf[1024] = {"aaa"};
```

```
/*
```

```
if(ret = read(0,buf,sizeof(buf)) > 0)
```

```
*/
```

```
arr(ret,buf);
```

```
return 0;
```

```
err:
```

```
perror("read()");//메시지보여줌
```

```
exit(-1);
```

```
}
```

```

mhn@mhn-Z20NH-AS51B5U:~/linux/25$ gcc test3.c
test3.c: In function 'arr':
test3.c:12:1: error: label 'err' used but not defined
goto err;//스택을 해제할 수 있는 능력이 없음 그러니까 메인으로
^

```

#### 예제 4)

```
#include <fcntl.h>
#include <stdlib.h>
#include <setjmp.h>
#include <stdio.h>

jmp_buf env1;
jmp_buf env2;
jmp_buf env3;

void test(void){

    longjmp(env1,1); //env1 을 타고 호출했던 곳으로 감

}

void test2(void){

    longjmp(env1,2);

}

void test3(void){

    longjmp(env2,3);
}

void test4(void){

    printf("test4\n");
    longjmp(env3,1);

}

int main(void){

    int ret;
    if((ret = setjmp(env1)) == 0){ //setjmp 는 처음 호출되면 무조건 0 이다
        test();
    }else if(ret == 1){
        printf("ret = 1\n");
        test2();
    }else if(ret == 2){
        printf("ret = 2\n");
    }//else
    //      goto go;

    if((ret = setjmp(env2))==0){
        test3();
    }else if(ret == 3){
        printf("ret = 3, error\n");
    }else
        printf("else\n");
}
```

```

        //      longjmp(env1,7);

        if((ret = setjmp(env3))==0){
        test4();
        }

        return 0;
/*go:
        printf("goto\n");
        exit(-1);*/
}

```

//setjmp는 최초 호출 시 0 반환

//setjmp는 env를 세팅해준 위치는 env가 가지고 있음

int setjmp(jmp\_buf env)

- 함수가 호출되는 순간 스택값들은 env에 저장됩니다.
- setjmp()호출, longjmp 할 곳을 지정합니다.

void longjmp(jmp\_buf env, int val)

- longjmp를 호출하면 setjmp()를 한 곳으로 돌아갑니다.

```

mhn@mhn-Z20NH-AS51B5U:~/linux/25$ ./a.out
ret = 1
ret = 2
ret = 3, error
test4
mhn@mhn-Z20NH-AS51B5U:~/linux/25$

```

## 예제 5)

```

#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <stdlib.h>

```

```

void my_sig(int signo){

```

```

    printf("You must insert coin\n");
    exit(0);

```

```

}

```

```

int main(void){

```

```

    char buf[1024];
    int ret;
    signal(SIGALRM, my_sig);
    alarm(3);
    read(0, buf, sizeof(buf));

```

```

        alarm(0);

        return 0;
}
//SIGALRM 을 준다. 3 초 안으로 안쓰면 my_sig 가 호출된다
//알람은 alarm(0)으로 꺼줘야 한다

```

```

mhn@mhn-Z20NH-AS51B5U:~/linux/25$ ./a.out
You must insert coin
mhn@mhn-Z20NH-AS51B5U:~/linux/25$ █

```

그래서 이것을 활용하여 게임을 만들어보았다.

```

#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <stdlib.h>
#include <time.h>

void my_sig(int signo){
//void my_sig(){

        printf("You lose\n");
        exit(0);

}

int main(void){

        int randd;
        int input;
        int i;
        int count=0;
        char buf[1024] = {0};
        srand(time(NULL));
        randd = 1+(rand()%100);
        printf("\n 정답은 %d 임!!!\n\n",randd); //변수가 잘 들어갔는지 확인하려고 넣었다

        signal(SIGALRM,my_sig);
        printf("게임을 시작합니다. 입력은 1 초 이내입니다\n");

        re:

        if(count == 5){ //5 번안에 못하면 실패다
        printf("기회는 5 번이었습니다ㅜ\n");

//        alarm(1);
//        signal(SIGALRM,my_sig);

```

```

my_sig(14);
exit(0);
}

printf("숫자를 입력하세요!!!\n");
alarm(1); //만약에 1 초 안에 입력을 하지 않으면 signal 이 울려서 함수를 호출한다
read(0,buf,sizeof(buf));
input = atoi(buf); //read 는 문자열로 받으니까 그것을 int 로 바꿔줬다
signal(SIGALRM,my_sig);
alarm(1);
if(input > randd){
    printf("입력한 숫자가 더 큼니다\n");
    count++; //기회는 5 번이므로 카운트를 센다
    goto re; //re 로 돌아간다
}else if(input < randd){
    printf("입력한 숫자가 더 작습니다\n");
    count++;
    goto re;
}else if(input == randd){
    printf("정답입니다!!!!\n");
    alarm(0);
    exit(0);
}else
    printf("잘못된 입력입니다.\n");
    exit(0);

return 0;
}

```

```

mhn@mhn-Z20NH-AS51BSU:~/linux/25$ ./a.out
정답은 28임!!!!
게임을 시작합니다. 입력은 1초 이내입니다
숫자를 입력하세요!!!
32You lose
mhn@mhn-Z20NH-AS51BSU:~/linux/25$ 32
32: command not found
mhn@mhn-Z20NH-AS51BSU:~/linux/25$ ./a.out
정답은 62임!!!!
게임을 시작합니다. 입력은 1초 이내입니다
숫자를 입력하세요!!!
54You lose
mhn@mhn-Z20NH-AS51BSU:~/linux/25$ 54
54: command not found
mhn@mhn-Z20NH-AS51BSU:~/linux/25$ ./a.out
정답은 47임!!!!
게임을 시작합니다. 입력은 1초 이내입니다
숫자를 입력하세요!!!
43
입력한 숫자가 더 작습니다
숫자를 입력하세요!!!
23
입력한 숫자가 더 작습니다
숫자를 입력하세요!!!
12
입력한 숫자가 더 작습니다
숫자를 입력하세요!!!
32
입력한 숫자가 더 작습니다
숫자를 입력하세요!!!
43
입력한 숫자가 더 작습니다
기회는 5번이였습니다
You lose

```