

Xilinx Zynq FPGA,TI DSP, MCU 기반의 프로그래밍 전문가 과정

날 짜 : 2018 . 3. 22

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – 정한별
hanbulkr@gmail.com

<signal.c>

```
#include<stdio.h>
#include<signal.h>
#include<unistd.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);
    void (*old_p2)(int);
    // 먼저 pause 에 걸림, 행동지침을 실행하고(my_sig), null 값을 반환한다. 그리고 저장을 한다.
    old_p = signal(SIGINT, my_sig);
    pause();
    // my_sig2 를 실행하고, 이전 것을 반환한다, 그리고 현재 my_sig2 를 저장한다.
    old_p2 = signal(SIGINT, my_sig2);
    pause();
    // old_p2 가 이전 값을 가지고 있어서 my_sig 를 실행한다.
    old_p2 = signal(SIGINT, old_p2);
    //old_p2 가 null 값을 가지고 있어서
    //signal 은 자기 자신의 한단계 전걸 리턴해 준다.
    pause();
    for(;;)
    // 하지만 old_p2 는 그담에 쓰지 않는다 그냥 저장하고 끝난 거니까 시그널은 그대로 이전거만 동작
    // signal(SIGINT, old_p2); 이 시그널이 동작을 하는 것이다.
        pause();
    return 0;
}
```

<signal5.c>

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

int main(void)
{
    // pause 에 걸려서 기다리다가 시그널이 날라오면 동작함, ctrl + c 이면 무시한다는 뜻이다. 그래서 무시함
    signal(SIGINT, SIG_IGN);
    pause();
    return 0;
}
```

<goto.c>

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

void run(void)
{
    // goto 는 stack 을 해제를 할 능력이 없다.
    // 따라서 컴파일이 되지 않는다.
    goto err;
}

int main(void)
{
    int ret;
    char buf[1024] = "aaaabbbbcccc";
    if ((ret = read(0, buf, sizeof(buf)))>0)
        run();
    return 0;

err:
    perror("read()");
    exit(-1);
}
```

<goto2.c>

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

//
jmp_buf env;

void test(void)
{
    // 1 은 리턴값을 말하는 것이다.
    longjmp(env, 1);
}

int main(void)
{
    int ret;

    //setjmp 를 아무것도 안했을 때는 무조건 0 이다.
    // env 가 레이블의 위치를 가지고 있다.
    //if((ret = setjmp(env)) == 0)
    if((ret = setjmp(env))==0){
        printf("this\n");
        test();
    }
    else if(ret>0)
        printf("error\n");
    return 0;
}
```

<goto2_1.c>

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

//
jmp_buf env;

int test(void)
{
    // 1 은 리턴값을 말하는 것이다.
    longjmp(env,1);
    return 1;
}

void test1(void)
{
    longjmp(env,2);
}

int main(void)
{
    int ret;
    int res;
    //setjmp 를 아무것도 안했을 때는 무조건 0 이다.
    // env 가 레이블의 위치를 가지고 있다.
    // setjmp 를 하면 레이블이 생긴다.
    //if((ret = setjmp(env)) == 0)
    if((ret = setjmp(env))==0){
        printf("this\n");
        test();
    }
    else if(ret == 1){
        printf("second\n");
        test1();
    }
    else if(ret == 2){
        printf("goto label\n");
        goto letme;
    }
    else
        printf("error\n");
    // letme 가 레이블이다.
letme:
    // see 가 longjmp
    goto see;

    return 0;
see:
    printf("error!!\n");
}
```

<setjmp2.c>

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

jmp_buf env;

void test(void)
{
    int flag= -1;
    if(flag <0)
        // jmp 를 하면 돌아 오지않는다. 밑에 함수들은 실행되지 않는다.
        longjmp(env,1);
    printf("call test\n");
}

int main(void)
{
    int ret;
    if((ret = setjmp(env)) == 0)
        test();
    else if(ret >0)
        printf("error\n");
    return 0;
}
```

<setjmp3.c>

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

void my_sig(int signo)
{
    printf("You must insert coin\n");
    exit(0);
}

int main(void)
{
    char buf[1024];
    int ret;
    // 시그널 함수를 통해서 알람시그널이 오면 my_sig 함수를 실행한다.
    signal(SIGALRM, my_sig);
    // 3 초뒤에 울려라. 라는 알람 시그널
    Alarm(3);
    // 입력까지 대기, 입력을 읽어서 buf 에 저장
    read(0,buf, sizeof(buf));
    alarm(0);
    return 0;
}
```

<game.c>

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

void my_sig(int signo)
{
    printf("입력 시간이 늦었습니다 . 술마시세요!!!\n");
    exit(0);
}

void game(int random)
{
    char buf[1024];
    int ret;
    int i=0;
    int setnum;
    signal(SIGALRM, my_sig);
    printf("술게임을 시작합니다\n");
    printf("내가 내는 번호를 맞춰 보세요!\n");
    while(i<5)
    {
        // 알람 시그널을 보낼 시간 설정. 2 초.
        alarm(2);
        //read 함수로 대기함. 글자가 들어오는 것을 대기 하고 들어오면 buf 에 저장한다.
        read(0,buf,sizeof(buf));
        //글자형으로 받았으니 숫자로 바꾼다.
        setnum = atoi(buf);
        if(setnum == random){
            printf("맞추셨습니다! 술 드세요!!!\n");
            exit(0);
        }
        else if(setnum > random){
            printf("너 높다.\n");
            alarm(2);
        }
        else if(setnum < random){
            printf("너 낮다.\n");
            alarm(2);
        }
        i++;
    }
}

int main(void)
{
    int random;

    srand(time(NULL));
    random = rand()% 26 +1;

    //    longjmp(env, 1);
    game(random);
}
```