

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

2018-03-26 (24 회차)

강사: Innova Lee(이상훈)

gcccompil3r@gmail.com

학생: 정유경

ucong@naver.com

1. signal.c

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

void my_sig(int signo) // 시그널 핸들러의 프로토 타입: void (*)(int)
{
    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);

    old_p = signal(SIGINT, my_sig); // signal() 함수는 기존의 처리함수를 반환(없으므로 0 을 반환)
    pause(); // 시그널을 받을 때까지 프로세스 중지
    old_p2 = signal(SIGINT, my_sig2); // 기존의 처리함수 my_sig 가 old_p2 에 저장
    pause();

    // case 1. old_p2 = signal(SIGINT, old_p); // old_p 즉 null 실행, 이전 처리함수 my_sig2 가 old_p2 에 저장
    // case 2. old_p2 = signal(SIGINT, old_p2); // old_p2 즉, my_sig 실행, 이전 처리함수 my_sig2 가 old_p2 에 저장

    pause();
    for(;;)

        pause(); // case1 의 경우는 null, case2 의 경우는 my_sig 가 계속 실행된다

    /*for 문인데 종료하는 이유: null 일경우 프로세스 종료 , my_sig 에 0 넣으면 프로세스 종료된다*/

    return 0;
}
```

```
yukyoun@yukyoun-Z20NH-A551B1U:~/Workspace/0327$ ./a.out
^Cmy_sig called
^Cmy_sig2 called
^C
yukyoun@yukyoun-Z20NH-A551B1U:~/Workspace/0327$
```

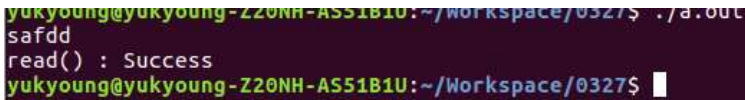

3. goto.c

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

int main()
{
    int ret;

    char buf[1024]="Hello, are you there"; // 문자열 넣어주지 않아도 된다. 표준입력 0 으로 입력받기
    때문
    if((ret=read(0,buf,sizeof(buf)))>0)
        goto err;
    return 0;

err:
    perror("read() "); //괄호안에 들어있는 시스템 콜에 대한 에러 사유 출력
    exit(-1);
}
```



4. goto_func.c : goto 는 스택을 해제할 수 없기 때문에 main()의 err 로 갈 수 없다. Setjmp(), longjmp() 시스템 콜로 해결 가능하다.

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

void func()
{
    goto err;
}
```

```

int main()
{
    func();

    return 0;

err:
    perror("doesn't work!\n");
    exit(-1);
}

```

5. setjmp0.c

```

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

jmp_buf env;

void test(void)
{
    longjmp(env,1);
}

int main(void)
{
    int ret;
    if((ret=setjmp(env))==0)
        test();

    else if( ret > 0)
        printf("error\n");

    return 0;
}

```

6. setjmp2.c

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

jmp_buf env; // 상태를 저장하는 버퍼 구조체

void test(void)
{
    longjmp(env,1); // env 에 저장된 상태를 복구하여 스택 내용 등이 저장된 곳으로 점프
}
// 1 말고 0 을 주면 if 에 걸려서 무한루프를 돌게된다

int main(void)
{
    int ret;

    if( (ret=setjmp(env))==0 ) // 1. setjmp( ) 함수 호출되면 스택의 상태 등은 env 에 저장된다, 처음
    반환값은 0
    {
        // 2. longjmp() 에 의해 두번째 반환되면 반환값은 1
        printf("thisWn");
        test();
    }
    else if( ret > 0 ) // 두번째 1 이 반환되면 else if 로 들어가서 error 출력
        printf("errorWn");

    return 0;
}
```

6. <퀴즈> setjmp.c 에 새로운 레이블 추가해보기

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

jmp_buf env1;
jmp_buf env2;

void test1(void)
{
    longjmp(env1, 1);
```

```

}
void test2(void)
{
    longjmp(env1, 2);
}
void test3(void)
{
    longjmp(env2, 1);
}

int main(void)
{
    int ret;
    if( (ret = setjmp(env1)) == 0 ) // ret 가 0 인지 검사(setjmp(env1)의 리턴값은 0)
    {
        printf("this\n");
        test1(); // setjmp(env1)의 리턴값은 1
    }
    else if(ret == 1)
    {
        printf("1\n");
        test2(); // // setjmp(env1)의 리턴값은 2
    }
    else if(ret == 2)
        printf("2\n");
    else // setjmp(env1)의 리턴값은 3 이므로 else 진입한다
    {
        printf("goto letsgo label\n");
        goto letsgo; // letsgo 레이블로 이동
    }
    if( (ret = setjmp(env2)) == 0 ) // setjmp(env2)의 리턴값은 0
    {
        printf("second label\n");
        test3(); // setjmp(env2)의 리턴값은 1
    }
    else
        longjmp(env1, 3); // setjmp(env1)의 리턴값은 3
}

```

letsgo:

```
        goto err; // goto 레이블로 이동
    return 0;
err:
    printf("Error!!!\n");
    exit(-1); // 에러이므로 -1 을 반환하고 프로세스 종료
}
```

7. setjump22.c

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

jmp_buf env;

void test()
{
    int flag = -1;
    if(flag < 0)
        longjmp(env, 1); // setjmp(env)의 리턴 값은 1
    printf("call test\n"); // 출력되지 않는다 (longjmp 하고 다시 돌아오지 않는다 )
}

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


8. 행맨, updown 게임 만들기 위한 전초작업

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

void my_sig(int signo) // SIGALRM 의 시그널 핸들러
{
    printf("you must insert coin\n");
    exit(0); // 프로세스 종료
}

int main(void)
{
    char buf[1024];
    int ret;
    signal(SIGALRM, my_sig);
    alarm(3); // 3 초후에 SIGALRM 시그널이 발생
    read(0,buf,sizeof(buf));
    alarm(0); // 알람 끄기
    return 0;
}
```

9. <퀴즈> updown 게임을 만들어보자 – updown.c

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

jmp_buf env;
void sigalrm(int signo)
{
    printf("\n1 초내에 입력해야 합니다.\n");
    exit(0);
}
```

```

int main()
{
    int buf;
    srand(time(NULL));
    int num = (rand()%25)+1;
    int ret;

    ret = setjmp(env);
    if(ret==1)
    {
        printf("\n1~25 사이의 숫자를 입력하세요\n");
        scanf("%d",&buf);

        if(buf > num && 25 > buf)
            printf("down!\n");
        else if(buf < num && 1<buf)
            printf("up!\n");
        else if(buf==num)
        {
            printf("Correct!\n");
            return 0;
        }
    }
    /*1 초내로 입력하지 않으면 프로세스 종료*/
    signal(SIGALRM, sigalrm);
    alarm(3);
    longjmp(env,1);
    return 0;
}

```

[정답]

```

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

```

```

#include <stdlib.h>
#include <stdbool.h>

void sig_handler(int signo)
{
    printf("You lose! Input should be within 1 second!\n");
    exit(0);
}

void make_game(int *data)
{
    *data = rand() % 100 + 1;
}

bool check_correct(int data, int cmp)
{
    if(data == cmp)
        return true;
    else
        return false;
}

void start_game(int data)
{
    char buf[32] = {0};
    bool fin;
    int i, cmp;

    for(i = 0; i < 10; i++)
    {
        signal(SIGALRM, sig_handler);
        printf("숫자를 맞춰봐!\n");
        alarm(1); // 1 초후에 SIGALRM 발생
        read(0, buf, sizeof(buf));
        alarm(0);
        cmp = atoi(buf); // read 로 들어온건 문자열로 인식!! 숫자로 바꿔주어야 한다

        fin = check_correct(data, cmp);

        if(fin)
        {

```

```
        printf("You Win!!!\n");
        exit(0);
    }
    else
    {
        if(data > cmp)
            printf("%d 보다 크다\n", cmp);
        else
            printf("%d 보다 작다\n", cmp);
    }
}

// printf("You Lose!!! You Babo\n");
}

int main(void)
{
    int data;

    srand(time(NULL));
    make_game(&data);
    start_game(data);
    return 0;
}
```