

Xilinx Zynq FPGA, TI DSP MCU 기반의

프로그래밍 및 회로 설계
전문가

강사 이상훈
(Innova Lee)

Gcccompil3r@gmail.com

학생 김민호

minking12@naver.com

```

#include<unistd.h>
#include<stdio.h>
#include<signal.h>
#include<stdlib.h>
#include<fcntl.h>
#include<setjmp.h>
#include<time.h>
#include<stdbool.h>
//signal 1 번, ctrl + c 입력시 출력.
/*
void my_sig(int signo)
{
    printf("my_sig called\n");
}

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

int main()
{
    void (*old_p)(int);//함수 포인터 2 개 선언.
    void (*old_p2)(int);
    old_p=signal(SIGINT,my_sig);//시그널 행동 지침 시그인트 있으면 마이 시그 실행,리턴
    pause();
    old_p2=signal(SIGINT,my_sig2);
    pause();
    //old_p2=signal(SIGINT,old_p);//뭐가 없으니 끝남
    old_p2=signal(SIGINT,old_p2);//안끝남 시그널이 뭔가 리턴하니 끝나지 않는다.//mysig1 이 나옴. 시그
    널은 앞전 1 단계 등록했던것을 리턴해준다.올드 피는 널 올드피 2 는 마이 시그가 들어간다.
    pause();
    for(;;)
        pause();
    return 0;
}
*/

//kill 확인하는 코딩.
/*
int main()
{
    signal(SIGINT,SIG_IGN);
    pause();
    return 0;
}
*/

```

```

//goto
/*
int main()
{
    int ret;
    char buf[1024]="abcdefg";
    if((ret=read(0,buf,sizeof(buf)))>0)//0 은 표준입력
        goto err;

    return 0;

err:
    perror("read() ");//read 에 대한 시스템콜에대한 상태를 띄운다.
    exit(-1);
}
*/
//함수 goto 안되는것. goto 는 스택을 풀어헤치지 못해서. 함수는 스택을 만든다.
/*

void err
{
    goto err;;
}

int main()
{
    int ret;
    char buf[1024]="abcdefg";
    if((ret=read(0,buf,sizeof(buf)))>0)//0 은 표준입력
        goto err();

    return 0;

err:
    perror("read() ");//read 에 대한 시스템콜에대한 상태를 띄운다.
    exit(-1);
}
*/

```

```

//setjmp goto 를 함수역할? 로 이을수있는 문법 .
/*
jmp_buf env;

void test(void)
{
    longjmp(env,1);//env 구간 1 은 setjmp 의 리턴값이 1.
}

int main()
{
    int ret;
    if((ret=setjmp(env))==0)//goto 의 레이블 ... 처음에는 리턴값이 무조건 0
    {
        printf("this\n");
        test();
    }
    else if(ret>0)//리턴값이 0 보다 크니까 출력
        printf("error\n");
    return 0;
}
*/

```

```

/*

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)
    {
        printf("this\n");//this 실행
        test1();//루프점프로 가서 1 리턴
    }
    else if(ret == 1)
    {
        printf("1\n");//1 출력
        test2();//루프점프로 가서 2 리턴
    }
    else if(ret == 2)
        printf("2\n");//2 출력
    else
    {
        printf("goto letsgo label\n");//1 이 없으니 else 문 출력
        goto letsgo;//letsgo 로 이동
    }

    if((ret = setjmp(env2)) == 0)
    {
        printf("second label\n");//출력후 test3 에 루프점프로
        test3();//1 리턴 but 1 이 없으니
    }
    else
        longjmp(env1, 3);//1 을 리턴 했는데 없으니까 else 문 실행 env1 에 3 을 리턴 한것으로 ..
letsgo:
    goto err;//err 로 이동

```

```
return 0;
```

```
err:
```

```
printf("Error!!!\n");//출력
```

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

```
}
```

```
*/
```

```

/*
jmp_buf env;

void test(void)
{
    int flag=-1;
    if(flag<0)
        longjmp(env,1);//1 을 리턴 한다.
    printf("call test\n");//롱점프 밑에는 쳐봐야 의미가 없다.
}

int main()
{
    int ret;
    if((ret=setjmp(env))==0)//test 함수 실행
        test();
    else if(ret>0)
        printf("error\n");//1 리턴 했으니 에러 출력
    return 0;
}
*/
//hang man 기초틀
/*
void my_sig(int signo)
{
    printf("you have to insert the coin\n");
    exit(0);
}

*/

```

```
/*  
int main()  
{  
    char buf[1024];  
    int ret;  
    signal(SIGALRM,my_sig);//3 초후에 마이시그 실행  
    alarm(3);//3 초후에  
    read(0,buf,sizeof(buf));//출력,버프크기만큼 읽어들인다.  
    return 0;  
}  
*/
```


// 랜덤 숫자 생성[정답]

int game_start(int max, int chan)//max 에 9 chan 에 3 을 매개변수로 전달.

```
{
    int cor = 0;
    srand(time(NULL));

    cor = (rand() % max) + 1;//cor 에 1~9 사이 임의의 수를 저장시킨다.

    printf("게임 UP and DOWN! [1] ~ [%d]\n", max);
    printf("기회는 [%d]회 !!\n", chan);

    return cor;//cor 를 리턴한다.
}
```

void my_sig(int signo)

```
{
    printf("\n 시간 초과!\n");
    exit(0);
}
```

// 데이터 입력

void dataRead(int cor, int chan)

```
{
    char buf[1024];
    int data = 0;
    int i = 0;

    while(i < chan)
    {
        signal(SIGALRM, my_sig);//2 초가 지나면 my sig 를 발동시킨다.
        alarm(2);
        read(0, buf, sizeof(buf));//표준입력
        alarm(0); // 알람 끄기
        data = atoi(buf);//표준입력은 문자열로 받으니 상수로 바꿔주는 함수 atoi 를 사용한다.
        if(data == cor)
        {
            printf("\n 승리 !\n");
            break;
        }
        else if(data > cor)
            printf("[%d] 보다 작다\n", data);
        else
            printf("[%d] 보다 크다\n", data);

        i++;
    }
}
```

```
if(i == chan)//chan 이 3 이니 3 번이하로 맞추지 못하면 띄운다.  
    printf("\nGAME OVER\n");  
}
```

```
// 결과 확인 및 처리  
// 처리 1. 8 번 실패시 패배  
// 처리 2. 1 초 입력 안할 시 패배  
// 승리 동일할 경우 승리 처리
```

```
int main(void)  
{  
    int cor;  
    int max = 9;  
    int chan = 3;  
  
    cor = game_start(max, chan);  
    dataRead(cor, chan);  
  
    return 0;  
}
```