Xilinx Zynq FPGA, TI DSP,MCU기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - Innov (이상훈)

gcccompil3r@gmail.com

학생 - 이유성

dbtjd1102@naver.com

signal1

```
#include<stdio.h>
#include<signal.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);//signal 행동지침 (등록 ,my_sig실행)
	old_p =  signal(SIGINT,my_sig); //my_sig를 실행하고 ,처음 반환값은 :NULL.. 전 시그널의 실행한 함수의 이름(포인터) 을 반환
	pause();//위의 signal을 일단 점프해서 포즈가 걸리면 ctrl+c를 누른다 그러면 signal실행
	old_p2 = signal(SIGINT, my_sig2);// my_sig2를 실행 ,my_sig 반환 ->old_p2=my_sig
	pause();
	old_p2 = signal(SIGINT,old_p2);//my_sig를 실행하고 ,my_sig2 반환
	pause();
                .
	for(;;){  //메뉴얼은 그대로고 ctrl+c 메뉴얼에 해당하니까 실행된다.
	pause();
	printf("a\n");
	}

	return 0;
}
```

결과

```
^Cmy_sig called
^Cmy_sig2 called
^Cmy_sig called
^Cmy_sig called
a
^Cmy_sig called
a
```

signal2

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

int main(void)
{
        signal(SIGINT,SIG_IGN);//행동지침  -무시
        pause();  // pause걸리면 멈췄다가 ctrl+C 누르면 조건만족 ->SIG_IGNORE실행
        return 0;
}
```

바로 퍼즈
// kill -2 로하면 안죽음, kill -9로 죽임.

goto1

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

void func()
{


        goto err;
}

int main(void) //goto가 함수내에서만 작용함 .. goto는 스택을 해제할 능력이 없다..(다른 함수를 만들면 스택이 만들어짐. )
{
        int ret;
        char buf[1024] ;
        if((ret = read(0,buf,sizeof(buf))) >0 )//0 - scanf랑같다.
        func();

        return 0;

        err:
        perror("read() ");// 이것에 들어있는 시스템콜에대한 메세지.
        exit(-1);
}
```

2. goto -> longjmp(), setjmp()

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



jmp_buf env;//


void test(void)
{
        longjmp(env,1); //longjmp->첫번째 인자 값을 갖고 있는 setjmp로 이동,두번째 인자 값은
반환..
}

int main(void)
{
        int ret;
        if((ret = setjmp(env)) == 0){       //최초 반환값 0            lable(err)=env

        printf("test\n");
        test();
        }

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

        return 0;
}
```

결과
test
error

3 .goto -> longjmp(), setjmp()

```
#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);//goto
}

int main(void)
{
	int ret;
	if((ret = setjmp(env1)) == 0)   //첫번째 리턴값 0
	{
		printf("this\n");
		test1();
	}
	else if(ret == 1)
	{
		printf("1\n");
		test2();
	}
	else if(ret == 2)
		printf("2\n");
	else
	{
		printf("goto letsgo label\n");
		goto letsgo;//**
	}

	if((ret = setjmp(env2)) == 0)  //다시 첫번째 리턴값 0
	{
		printf("second label\n");
		test3();
	}
	else
		longjmp(env1, 3);
letsgo:/**
		goto err;
```

```
        return 0;

err:
        printf("Error!!!\n");
        exit(-1);
}
```

결과

```
this
1
2
second label
goto letsgo label
Error!!!
```

4 goto -> longjmp(), setjmp()

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

jmp_buf env;

void test(void)
{
        int flag = -1;
        if(flag <0)
        longjmp(env,1);   //이 밑의 코드는 의미 없다.
        printf("call test\n");
}

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

결과

error

5 goto -> longjmp(), setjmp()

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

// 코인게임을 만들 수 있다. 행맨 .

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

int main(void)
{
        char buf[1024]; //="aaa"
        int ret;
        signal(SIGALRM,my_sig);
        alarm(3); //signal패쓰. 바로 스타트.  alarm()- 3초되면 SIGALRM을 반환.

       read(0,buf,sizeof(buf)); //write (1,
                //3초내에 입력했다면 그것을 읽고 버퍼에 저장 그리고  alarm(0)실행
        alarm(0);//초기 알람 초기 세팅 자체를 끈다/무시한다.정확하게 끝난다.
        //for(;;); ->
        return 0;
}
```

결과

입력 안했을 때.
You must insert coin

숫자 입력했을 때.
1

2

6 goto -> longjmp(), setjmp()

```
#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);
		read(0, buf, sizeof(buf)); //read는 문자열로 읽으니까 atoi(buf)로 int받는다
		alarm(0);//알람을꺼버림
		cmp = atoi(buf);//문자를 숫자로  비쥬얼계산기cmp

		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;
}
```