TI DSP, MCU, Xilinx Zynq FPGA Based Programming Expert Program

Instructor – Innova Lee(Sanghoon Lee)
gcccompil3r@gmail.com
Student – Hyungju Kim
mihaelkel@naver.com

the demerits of “go to”

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

//it dosen't work because "goto" only works in the same function.
//goto doesn't have an ability that release the "stack"
void goto_func(void){
    goto err;
}
int main(void){
    int ret;
    char buf[1024];
    if((ret = read(0,buf,sizeof(buf))) > 0)
        goto_func();

    return 0;
err:
    perror("read()");
    exit(-1);
}

```


above program doesn't work, because of compile error.
“goto” can’t release the stack. “go to” can only move to somewhere in the same function ,that is, in the local area. If you want to move another funcion, there is “setjmp” and “longjmp”.

setjmp & longjmp

```c
#include <fcntl.h>
#include <stdlib.h>
#include <setjmp.h>
#include <stdio.h>
#define LABEL1 1
#define LABEL2 2
jmp_buf env;

void test(void){
    //LABEL1 is return value.
    longjmp(env, LABEL1);
}
void label(void){
    longjmp(env, LABEL2);
}

int main(void){
    int ret;
    //setjmp env-> create lable
    if((ret = setjmp(env)) == 0){
        printf("thus\n");
        test();
    }
    else if(ret == LABEL1){
        printf("1st Label \n");
        label();
```

```
27      }
28      else if(ret == LABEL2)
29          printf("2nd Label \n");
30
31      return 0;
32  }
33
```

implement timer

alarm function returns SIGALRM when the time(s) in the parameter flew.

```
1   #include <stdio.h>
2   #include <signal.h>
3   #include <fcntl.h>
4   #include <stdlib.h>
5   #include <unistd.h>
6
7   void my_sig(int signo){
8       printf("You must insert coin\n");
9       exit(0);
10  }
11  int main(void){
12      char buf[1024];
13      int ret;
14      signal(SIGALRM,my_sig);
15      alarm(3);
16      read(0,buf,sizeof(buf));
17      alarm(0);
18      return 0;
19  }
20
```


implement up&down game

```
1   #include <stdio.h>
2   #include <time.h>
3   #include <signal.h>
4   #include <fcntl.h>
5   #include <stdlib.h>
6   #include <unistd.h>
7   #include <setjmp.h>
8   #include <string.h>
9   jmp_buf env;
10
11  int init_game_num(void);
12  void time_over(int signo);
13  int main(void){
14      int num;
15      num = init_game_num();
16      int cnt = 1;
17      char buf[100];
18      int input;
19      int gamechk = 0;
20      int ret;
21      char* s_str = "input your number :";
22      if((ret = setjmp(env)) == 0){
23          while(cnt <= 5){
24              signal(SIGALRM,time_over);
25              alarm(1);
```

```
26                  write(1,s_str,strlen(s_str));
27                  read(0,buf,sizeof(buf));
28                  input = atoi(buf);
29                  if(input == num){
30                      gamechk = 1;
31                      break;
32                  }
33                  else if(input > num)
34                      printf("lower\n");
35                  else
36                      printf("upper\n");
37                  cnt++;
38              }
39              if(gamechk == 0)
40                  printf("Lose, num : %d\n",num);
41              else
42                  printf("Win!, count : %d, num : %d\n",cnt,num);
43
44      }
45      else if(ret > 0)
46          printf("\ntime over, num : %d\n",num);
47
48      return 0;
49  }
50  int init_game_num(void){
51      srand(time(NULL));
52      int num;
53      num = rand()%25 + 1;
54      return num;
55  }
56  void time_over(int signo){
57      int flag = -1;
58      if(flag < 0)
59          longjmp(env,1);
60  }
61
```