

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

#27

강사:Innova Lee(이 상훈)
학생: 김시윤

과제 술게임

*멀티 클라이언트 시도 실패했습니다. 서버와 클라이언트 1:1 통신밖에 되지 않았습니다.

Server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <time.h>
#include <signal.h>
#include <fcntl.h>
```

```
typedef struct sockaddr_in si;
typedef struct sockaddr * sap;
int cnt;
#define BUF_SIZE 1024
```

```
void sig_handler(int signo)
{
    cnt++;

    alarm(3);
}
```

```
void err_handler(char *msg)
{
    fputs(msg, stderr);
    fputc('\n', stderr);
    exit(1);
}
```

```
}

void my_sig(int signo)
{
    cnt++;
    printf("cnt = %d\n", cnt);
}

void randomnum(int *rnd)
{
    *rnd = rand() % 3333 + 1;
}

void rndgame(int clnt_sock, char *msg, int rnd)
{
    int i;

    for(i=0; msg[i]; i++)
    {
        if(msg[i] == '\n')
            msg[i] = '\0';
    }
    int num = atoi(msg);

    char test[1024];
    char turn[1024] = "turn = ";
    char win[32] = "YOU WIN";
    char no1[64] = "숫자를 더 작게 해주세요\n";
    char no2[64] = "숫자를 더 크게 해주세요\n";
    char cntt[64];

    sprintf(test, "%d", num);
    signal(SIGALRM, sig_handler);

    if(num == rnd)
    {
        write(clnt_sock, win, strlen(win));
    }
}
```

```

        //write(clnt_sock, test,strlen(test));
        write(clnt_sock, turn ,strlen(turn));
        write(clnt_sock, cntt, strlen(cntt));
        close(clnt_sock);
        exit(0);
    }else
    {
        if(num > rnd)
        {
            cnt++;
            printf("%d \n",cnt);
            write(clnt_sock,no1,strlen(no1));
            sprintf(cntt,"%d",cnt);
            write(clnt_sock,turn,strlen(turn));
            write(clnt_sock,cntt,strlen(cntt));
            //write(clnt_sock,test,strlen(test));

        }
        else if(num < rnd)
        {
            cnt++;
            printf("%d \n",cnt);
            write(clnt_sock,no2,strlen(no2));
            sprintf(cntt,"%d",cnt);
            write(clnt_sock,turn,strlen(turn));
            write(clnt_sock,cntt,strlen(cntt));
            //write(clnt_sock,test,strlen(test));

        }
    }
    msg="0";
}

int main(int argc, char **argv)
{
    int i, str_len;
    int serv_sock, clnt_sock;
    int rnd;

```

```

char cnt_num[BUF_SIZE];
char msg[BUF_SIZE];

```

```

srand(time(NULL));
//rnd= rand() % 3333+1;
randomnum(&rnd);

```

```

printf("%d",rnd);

```

```

si serv_addr, clnt_addr;
socklen_t clnt_addr_size;

```

```

if(argc !=2)
{
    printf("use: %s <port>\n",argv[0]);
    exit(1);
}

```

```

serv_sock = socket(PF_INET, SOCK_STREAM, 0);

```

```

if(serv_sock == -1)
    err_handler("socket() error");

```

```

memset(&serv_addr , 0 , sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
serv_addr.sin_port = htons(atoi(argv[1]));

```

```

if(bind(serv_sock , (sap)&serv_addr , sizeof(serv_addr)) == -1)
    err_handler("bind() error");

```

```

if(listen(serv_sock , 5) == -1)
    err_handler("listen() error");

```

```

clnt_addr_size = sizeof(clnt_addr);

```

```

        for(i=0; i<5; i++)
        {
            clnt_sock = accept(serv_sock, (struct sockaddr *)&clnt_addr,
&clnt_addr_size);

            if(clnt_sock == -1)
                err_handler("accept() error");
            else
                printf("Connected Client %d\n", i + 1);

            for(;;)
            {
                read(clnt_sock, msg, BUF_SIZE);

                rndgame(clnt_sock,msg,rnd);

                signal(SIGALRM,sig_handler);
                alarm(3);
            }
            // write(clnt_sock, msg, str_len);

            close(clnt_sock);
        }
        close(serv_sock);

        return 0;
}

```

Client.c

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

```

```

#include <sys/socket.h>

```

```

typedef struct sockaddr_in si;
typedef struct sockaddr * sap;

```

```

#define BUF_SIZE 1024

```

```

void err_handler(char *msg)
{
    fputs(msg, stderr);
    fputc('\n',stderr);
    exit(1);
}

```

```

int main(int argc, char **argv)
{

```

```

    int sock, str_len;
    si serv_addr;
    char msg[32];
    char *m = "Input number(q to quit): ";

```

```

    if(argc !=3)
    {
        printf("use: %s <IP> <port>\n", argv[0]);
        exit(1);
    }

```

```

    sock = socket(PF_INET , SOCK_STREAM, 0);

```

```

    if(sock == -1)
        err_handler("socket() error");

```

```

    memset(&serv_addr, 0, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = inet_addr(argv[1]);
    serv_addr.sin_port = htons(atoi(argv[2]));

```

```

if(connect(sock, (sap)&serv_addr, sizeof(serv_addr)) == -1)
    err_handler("connect() error");
else
    puts("Connected .....");

for(;;)
{
    fputs("Input number(q to quit): ", stdout); //write
    fgets(msg, BUF_SIZE, stdin); //read

    if(!strcmp(msg, "q\n") || !strcmp(msg, "Q\n"))
        break;

    write(sock, msg, strlen(msg)); //fgets
    str_len = read(sock, msg, BUF_SIZE - 1);

    if(str_len == -1)
        err_handler("read() error!");

    msg[str_len] = 0;

    printf("msg from serv: %s\n", msg);
}
close(sock);

return 0;
}

```

* 3 초가 지날때마다 카운트를 올리는건 성공했는데,
3 초가 지날때마다 카운트를 write 하는것을 실패했습니다.
sig_handler 에 client sock 을 보내면 가능할거같은데 , 포인터로 보내야할지 아니면 인자를
추가 해야할지 고민입니다.

```

11
^C
siyun@siyun-CR62-6M:~/my_proj/27$ ./a.out 7777
1457Connected Client 1
1
3
4
5
6
9
11
12
13
14
15
^C
siyun@siyun-CR62-6M:~/my_proj/27$ █

```

```

siyun@siyun-CR62-6M: ~/my_proj/27
siyun@siyun-CR62-6M:~/my_proj/27$ ./clnt 127.0.0.1 7777
Connected .....
Input number(q to quit): 3333
msg from serv: 숫자를 더 작게 해주세요
turn = 1
Input number(q to quit): 2222
msg from serv: 숫자를 더 작게 해주세요
turn = 3
Input number(q to quit): 1111
msg from serv: 숫자를 더 크게 해주세요
turn = 4
Input number(q to quit): 1
msg from serv: 숫자를 더 크게 해주세요
turn = 5
Input number(q to quit): 4444
msg from serv: 숫자를 더 작게 해주세요
turn = 6
Input number(q to quit): 3
msg from serv: 숫자를 더 크게 해주세요
turn = 9
Input number(q to quit): 3
msg from serv: 숫자를 더 크게 해주세요
turn = 11
Input number(q to quit): 333
msg from serv: 숫자를 더 크게 해주세요
turn = 12
Input number(q to quit): 3333
msg from serv: 숫자를 더 작게 해주세요
turn = 13
Input number(q to quit): 3333
msg from serv: 숫자를 더 작게 해주세요
turn = 14
Input number(q to quit): ^C
siyun@siyun-CR62-6M:~/my_proj/27$ █

```