

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

강사 - Innov (이상훈)

gcccompil3r@gmail.com

학생 - 이유성

dbtjd1102@naver.com

op_server.c

```
#include <stdio.h> //server
#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
#define OPSZ 4 //사이즈 크기

void err_handler(char *msg){

    fputs(msg,stderr);
    fputc('\n',stderr);
    exit(1);

}
//함수추가
int calculate(int opnum,int *opnds,char op){
    // 연산할 수 , 배열,21번째에 연산자가 있다 [크기 22-1]
    int result = opnds[0], i;

    switch(op){

        case '+':
            for(i=1;i<opnum; i++)
                result += opnds[i]; //배열 int형 변환.
            break;

        case '-':
            for(i=1;i<opnum; i++)
                result -= opnds[i];

            break;

        case '*':
            for(i=1;i<opnum; i++)
                result *= opnds[i];

            break;

    }

    return result;
}

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

    int serv_sock, clnt_sock;
    char opinfo[BUF_SIZE];

    int result,opnd_cnt, i;
    int recv_cnt,recv_len;

    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) //소켓파일에 ip주소와
    포트번호를 셋팅.
        err_handler("bind() error");

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

    clnt_addr_size = sizeof(clnt_addr);///

//여기부터 다름.
    for(i=0;i<5;i++){
        opnd_cnt = 0;
        clnt_sock = accept(serv_sock, (sap)&clnt_addr, &clnt_addr_size);//blocking
//전달 받을 파일스크립터 ,연결요청 한 클라이언트의 정보,전달받을 클라이언트의 정보크기
        read(clnt_sock, &opnd_cnt,1);// 한바이트만 읽으니 opnd_cnt = 5만 읽어옴.
        //blocking

        printf("%d\n",i+1);

        recv_len = 0;

        while((opnd_cnt * OPSZ +1) > recv_len){
            recv_cnt = read(clnt_sock, &opinfo[recv_len],BUF_SIZE-1);
//BUF-1 지금은 의미없지만 다른 경우엔 의미 있을 수 있음.3000 recv_cnt몇바이트 읽었는지.
            recv_len += recv_cnt;
        }

        result = calculate(opnd_cnt, (int *)opinfo, opinfo[recv_len-1]);
//((int *)opinfo 인트형 포인터로 변형해서 주소를 반환?
        write(clnt_sock, (char *)&result, sizeof(result));

        close(clnt_sock);

    }
    close(serv_sock);
    return result;
}

```

op_client.c

```
#include <stdio.h> //client
#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
#define RLT_SIZE 4
#define OPSZ 4

void err_handler(char *msg){

    fputs(msg,stderr);
    fputc('\n',stderr);
    exit(1);

}

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

    int i,sock,result,opnd_cnt;
    char opmsg[BUF_SIZE] = {0};
    si serv_addr;

    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)); sss
    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 .....");
    //*****
    fputs("Operation Cnt: ", stdout);//printf기능 "Operation Cnt "2번째인자에 쓴다 ->그냥
    터미널(터미널 자체가 stdout) 에 써짐
    scanf("%d",&opnd_cnt);//여기서 blocking 나한테서 입력을 대기
    // 5썼다 가정.
    opmsg[0] = (char)opnd_cnt;//캐릭터형 배열이니까 문자열 ->저장. 캐릭터형도 1byte짜리
```

정수형이긴 함 ->이거를 아스키코드로 표현할 뿐이지;;;;;->200을 안넘어서 그냥 쓰는거임. 숫자가 문자로 변하지는 않아

```
for(i=0; i<opnd_cnt; i++){

    printf("Operand %d: ", i+1);
    scanf("%d", (int *)&opmsg[i * OPSZ + 1]); //opmsg에 숫자를 저장하기 위해
    인트형으로 받기 위해 형변환

}

fgetc(stdin); //터미널에 내가 쓴다 scanf c한글자
fputs("Operator: ", stdout); //
scanf("%c", &opmsg[opnd_cnt * OPSZ + 1]); //연산자.
write(sock, opmsg, opnd_cnt * OPSZ + 2); // 22크기 서버에 보낸거.
read(sock, &result, RLT_SIZE);

printf("Operation result: %d\n", result);
close(sock);

return 0;

}
```

일단은 op를 잘 이해해보자!

file_server.c

```
//파일 전송하는 프로그램
```

```
#include<fcntl.h>
#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 32
```

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

```
int main(int argc, char **argv)
{
    int serv_sock, clnt_sock, fd;
    char buf[BUF_SIZE] = {0};
    int read_cnt;

    si serv_addr, clnt_addr;
    socklen_t clnt_addr_size;

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

    fd = open("file_server.c", O_RDONLY);
    serv_sock = socket(PF_INET, SOCK_STREAM, 0);

    printf("%d\n", fd);

    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);

        clnt_sock = accept(serv_sock,(sap)&clnt_addr , &clnt_addr_size);
//여기만 바뀜.
        for(;;)
        {
            read_cnt = read(fd,buf,BUF_SIZE); //클라이언트한테 글 받을 때까지 block

            if(read_cnt < BUF_SIZE)
            {
                write(clnt_sock, buf,read_cnt);
                break;
            }
            write(clnt_sock,buf,read_cnt);

        } //다 읽고.
        shutdown(clnt_sock,SHUT_WR); //소켓 종료.write buffer만 차단.
        read(clnt_sock,buf,BUF_SIZE);
        printf("msg from client : %s\n",buf);

        close(fd);
        close(clnt_sock);
        close(serv_sock);

        return 0;
}

```

file_client.c

```

#include<fcntl.h>
#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 32

void err_handler(char *msg)

```

```

{
    fputs(msg,stderr);
    fputc('\n',stderr);
    exit(1);
}

int main(int argc, char **argv)
{
    char buf[BUF_SIZE]={0} ;
    int fd,sock,read_cnt;
    si serv_addr ;

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

    fd = open("receive.txt",O_CREAT |O_WRONLY|O_TRUNC,0644);
    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.....");
//여기만 바뀔.
    puts("Reveived File Data");
    write(sock,"Thanck you",10);
    close(fd);
    close(sock);

    return 0;
}

```


gethostname.c

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

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

int main(int argc, char **argv)
{
    int i;
    struct hostent *host; // 구조체 조사

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

    host = gethostbyname(argv[1]); // host 이름 받는다.

    if(!host)
        err_handler("gethost ...error!");
    printf("Official Name : %s\n",host->h_name);

    for(i=0 ; host->h_aliases[i];i++) //
        printf("Aliases %d: %s\n", i+1,host->h_aliases[i]);

    printf("Address Type : %s\n",host->h_addrtype ==AF_INET ? "AF_INET" : "AF_INET6");
    //h_addrtype ->IPv4 참.

    for( i = 0 ; host->h_addr_list[i]; i++)
        //웹브라우저의 naver.com   wep 번호 80
        printf("IP Addr %d : %s\n",i+1, inet_ntoa(*(struct in_addr *)host->h_addr_list[i]));

    return 0;
}
```