

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

강사 - Innov (이상훈)

gcccompil3r@gmail.com

학생 - 이유성

dbtjd1102@naver.com

```

#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 i,str_len;
    int serv_sock,clnt_sock;

    char msg[BUF_SIZE];

    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);//서버 소켓 생성.

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

    memset(&serv_addr,0,sizeof(serv_addr));//메모리 초기화 패턴 (값을 채울 메모리의 시작
주소,채울 값(int형으로 전달 받는 것은 unsig char,채울 값으로 채울 바이트 수)
    serv_addr.sin_family = AF_INET; //ipv4주소체계
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);//주소 설정해줌(내 시스템의 ip) 32bit
IPV4 주소
    serv_addr.sin_port = htons(atoi(argv[1]));

    if(bind(serv_sock, (sap)&serv_addr, sizeof(serv_addr)) == -1)// ip주소랑 port번호 셋팅
// 주소를 받을 fd,bind()함수를 이용하여 socket에 serv_sock에 필요한 정보를 할당하고
커널에 등록
// 이 socket에 주소를 할당하고 port번호를 할당해서 커널에 등록->커널에 등록->시스템
통신 가능.

        err_handler("bind() error");

```

```

if(listen(serv_sock,5)==-1) //접속할 수 있는 클라이언트 수
    err_handler("listen() error");

clnt_addr_size = sizeof(clnt_addr);

for(i=0;i<5;i++){ //들어온 클라이언트 승인 클라이언트 전용 소켓 생성됨
//*****
    clnt_sock = accept(serv_sock,(struct sockaddr *)&clnt_addr, &clnt_addr_size);
    //서버 소켓이 클라이언트의 접속을 허용 ,리턴값 -1 or 소켓디스크립터., clnt_sock
= 4
    // 인자 :(전달받을 fd , 연결요청한 client의 정보 , 전달 받을 client 정보 크기)
    //accept()로 접속 허락하면 client와 통신하기 위해 커널이 자동으로 소켓 생성(clnt_sock)
    //이제 read write가능.

    if(clnt_sock == -1)
        err_handler("accept() error");

    else
        printf("Conneted Client: %d\n",i+1);
        while((str_len = read(clnt_sock,msg,BUF_SIZE)) != 0) //read는 블록이라서 처리가
while이 끝나면 한번에 처리가 됨 해결하려면 논블로킹으로 바꿔주자
        // write(clnt_sock,msg,str_len);
        write(clnt_sock,"hello",6);
        close(clnt_sock);
    }
    close(serv_sock);

    return 0;
}

```

```

#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); //파일로부터 문자열 printf ? 저장 ?
    fputc('\n',stderr); //문자 하나
    exit(1);

}

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

    int sock,str_len;
    si serv_addr;
    char msg[1024];
    char *m = "Input Message(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]); //포트하고 ip주소가 있어야 연결 가능.
    serv_addr.sin_port = htons(atoi(argv[2]));

    if(connect(sock, (sap)&serv_addr,sizeof(serv_addr)) == -1) //소켓 디스크립터 ,서버 주소
정보에 대한 포인터 , 포인터가 가르키는 구조체의 크기
        err_handler("connect() error");
    else
        puts("Connected .....");

    for(;;){

        //read write 로 (할경우 aaaaa입력 시 q\n했으면 q\naaa가 돼서
        //안나가짐.
        fputs("Input msg(q to quit): ",stdout); //(write) 모니터(터미널에)
        fgets(msg,BUF_SIZE,stdin); //(read) 키보드(입력) 스트링 읽기-1.배열의 주소값
    }
}

```

2. 최대 몇개 읽을 건지 3. 어느 버퍼에서 읽을 건지 (gets는 무조건 stdin에서 읽어올 수 있다.

```
    if(!strcmp(msg, "q\n") || !strcmp(msg, "Q\n")) // 세미콜론 있으면 여기서 바로 끝.
        break; // strcmp 스트링 비교 원<오 리턴 값 0보다 작음
                // 원 = 오 0 원>오 0보다 큼 --> q나 Q면 break
    write(sock, msg, strlen(msg));
    str_len = read(sock, msg, BUF_SIZE-1);

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

    msg[str_len] = 0;

    printf("msg form serv: %s\n", msg);

}

close(sock);
return 0;
}
```

//strcmp 함수가 문자열이 같으면 0을 반환.(~ , ~)

//strncmp은 문자열 중 입력한 길이만큼만 비교 (a, b , 길이)

// gets() scanf에서 문자열을 입력받을 때 제약이 있어서 공백이 있는 문자열을 입력받기 위해

// gets(str)함수는 주소값(어디서부터 저장할건지)만 넘겨주면 공백이 있는 문자열 입력이 가능

// 내부적으로 getchar()함수를 반복하기 때문에 공백문자까지 입력할 수 있다.

// gets() 제약 char arr[10]; get(arr); 11개 입력시 애러.

// get 같은 경우는 무조건 표준입출력버퍼(stdin)에서 읽어올 수 있지만

// fgets는 표준입출력버퍼든지 , 파일, 네트워크이든 지정 가능. 또한 gets는 엔터값을 저장하지 않음