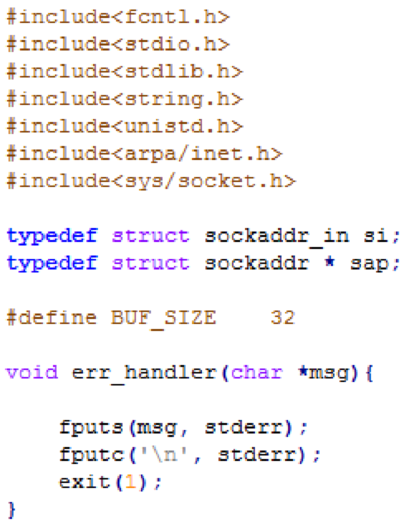
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

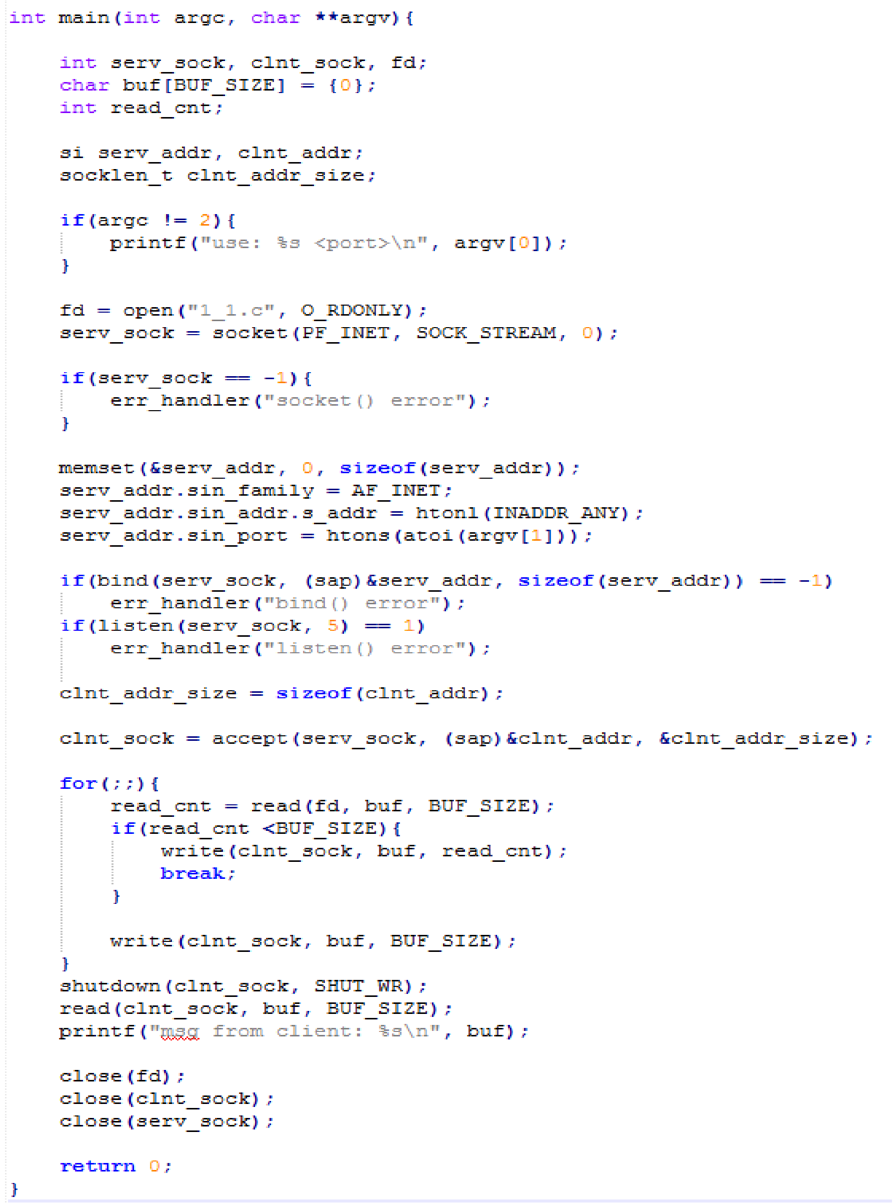
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – GJ (박현우)
uc820@naver.com

2. 네트워크 프로그래밍 – file 전송 (server)



```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);
}
```

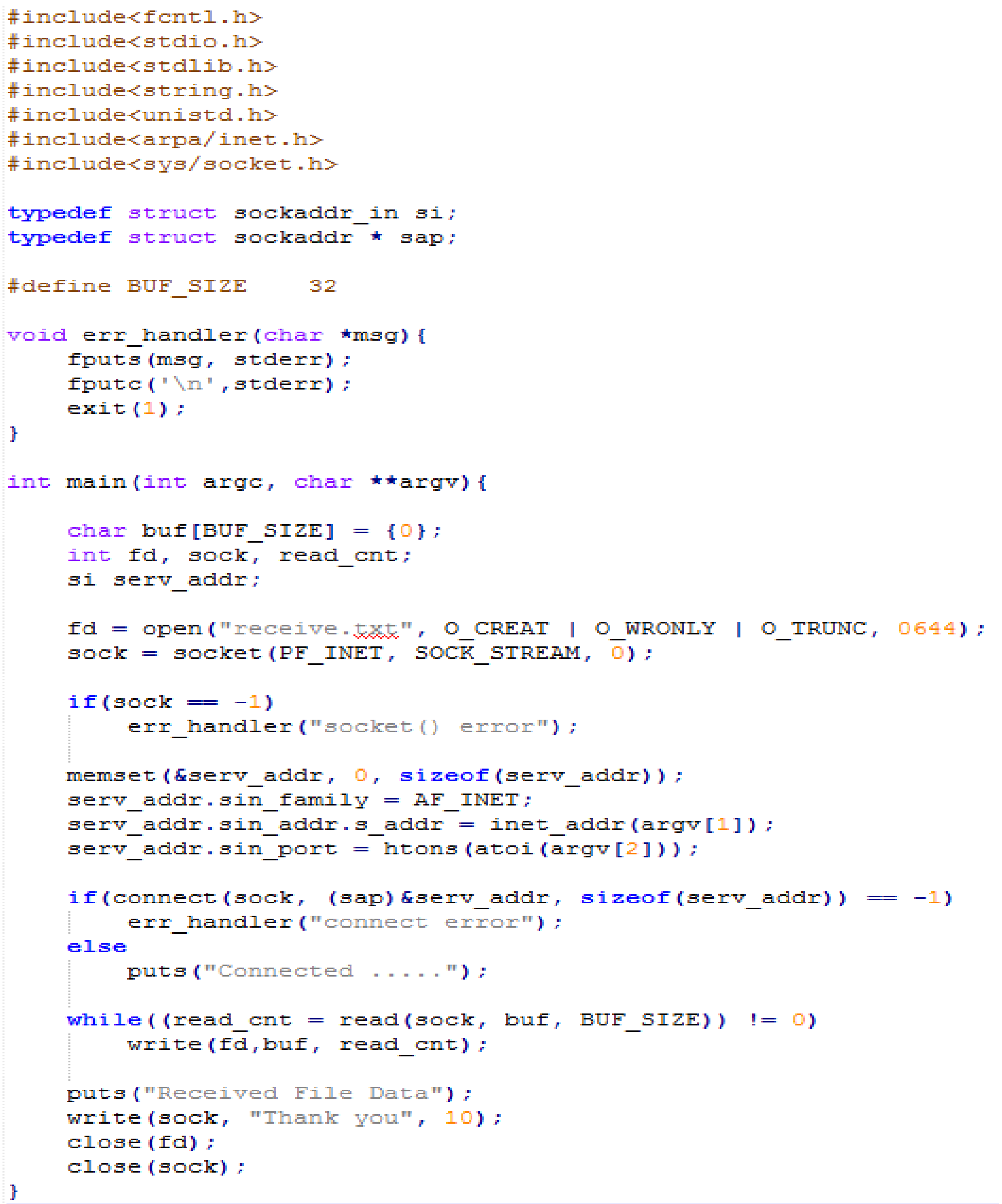


```c
int main(int argc, char **argv){

    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]);
    }

    fd = open("1_1.c", O_RDONLY);
    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);

    clnt_sock = accept(serv_sock, (sap)&clnt_addr, &clnt_addr_size);

    for(;;){
        read_cnt = read(fd, buf, BUF_SIZE);
        if(read_cnt <BUF_SIZE){
            write(clnt_sock, buf, read_cnt);
            break;
        }

        write(clnt_sock, buf, BUF_SIZE);
    }
    shutdown(clnt_sock, SHUT_WR);
    read(clnt_sock, buf, BUF_SIZE);
    printf("msg from client: %s\n", buf);

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

    return 0;
}
```

- open() 으로 1_1.c의 fd 값을 받고 read()로 fd에서 값을 읽어와 buf에 저장한다.

- buf를 client에게 write()한다.

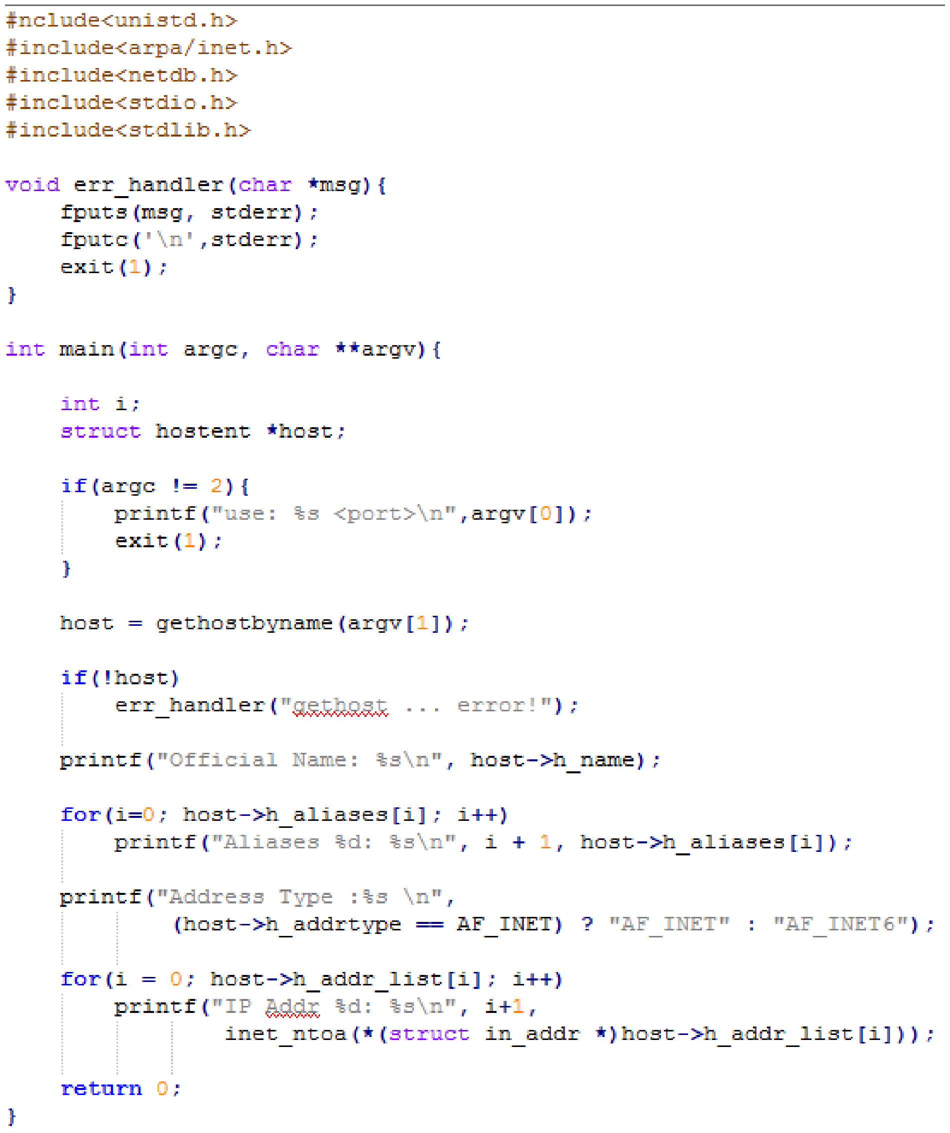
2. 네트워크 프로그래밍 – file 전송 (client)



```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;

    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 .....");

    while((read_cnt = read(sock, buf, BUF_SIZE)) != 0)
        write(fd,buf, read_cnt);

    puts("Received File Data");
    write(sock, "Thank you", 10);
    close(fd);
    close(sock);
}
```

- open() 으로 receive.txt를 만든다.
- Read()로 server로부터 buf를 받아와 fd에 write()로 받은 buf를 저장한다.

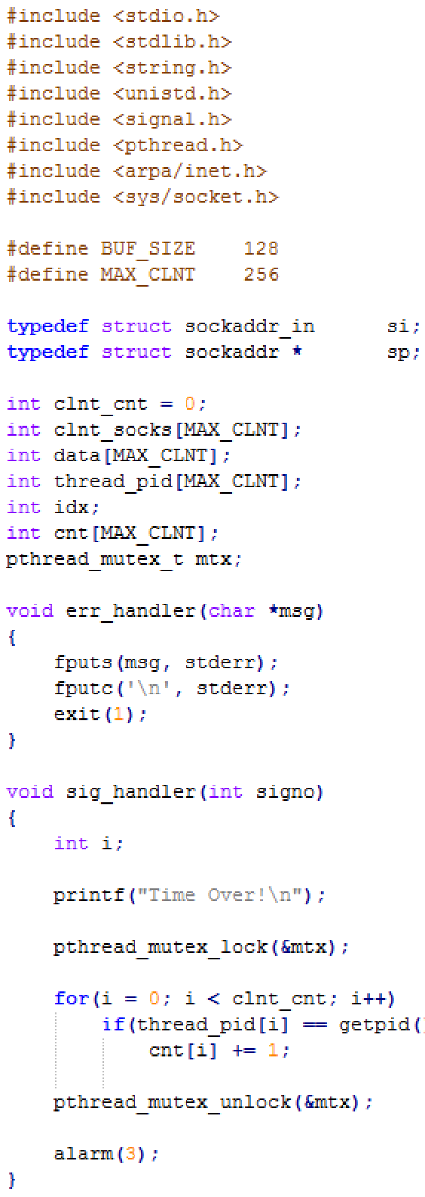
2. 네트워크 프로그래밍 – get_host_by_name



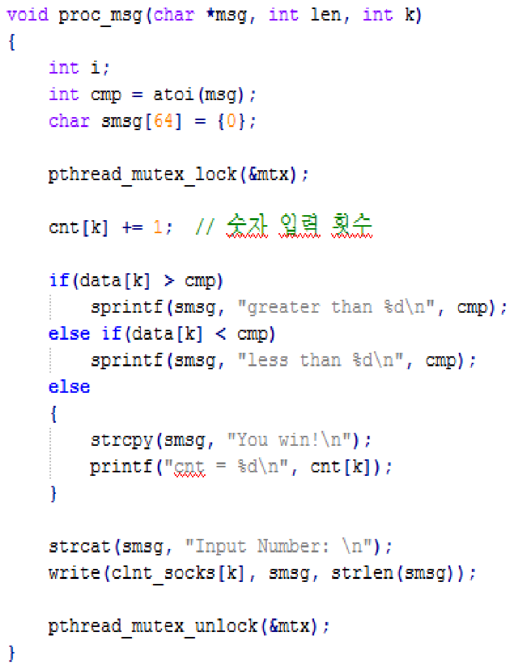
```c
#nclude<unistd.h>
#include<arpa/inet.h>
#include<netdb.h>
#include<stdio.h>
#include<stdlib.h>

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

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

    int i;
    struct hostent *host;

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

    host = gethostbyname(argv[1]);

    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");

    for(i = 0; host->h_addr_list[i]; i++)
        printf("IP Addr %d: %s\n", i+1,
                inet_ntoa(*(struct in_addr *)host->h_addr_list[i]));

    return 0;
}
```

- gethostbyname()으로 host의 이름, host의 별명, host의 주소 타입, host의 ip address를 얻어 온다.

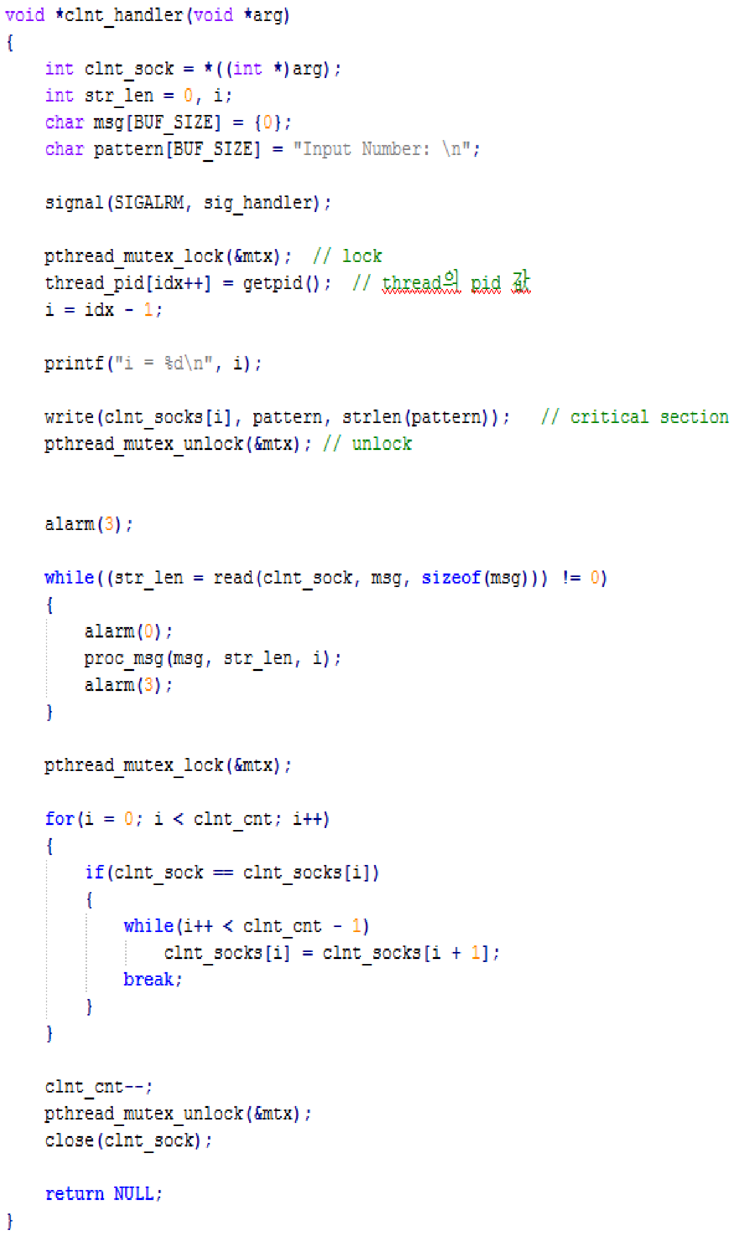
2. 네트워크 프로그래밍 – up & down game (server)



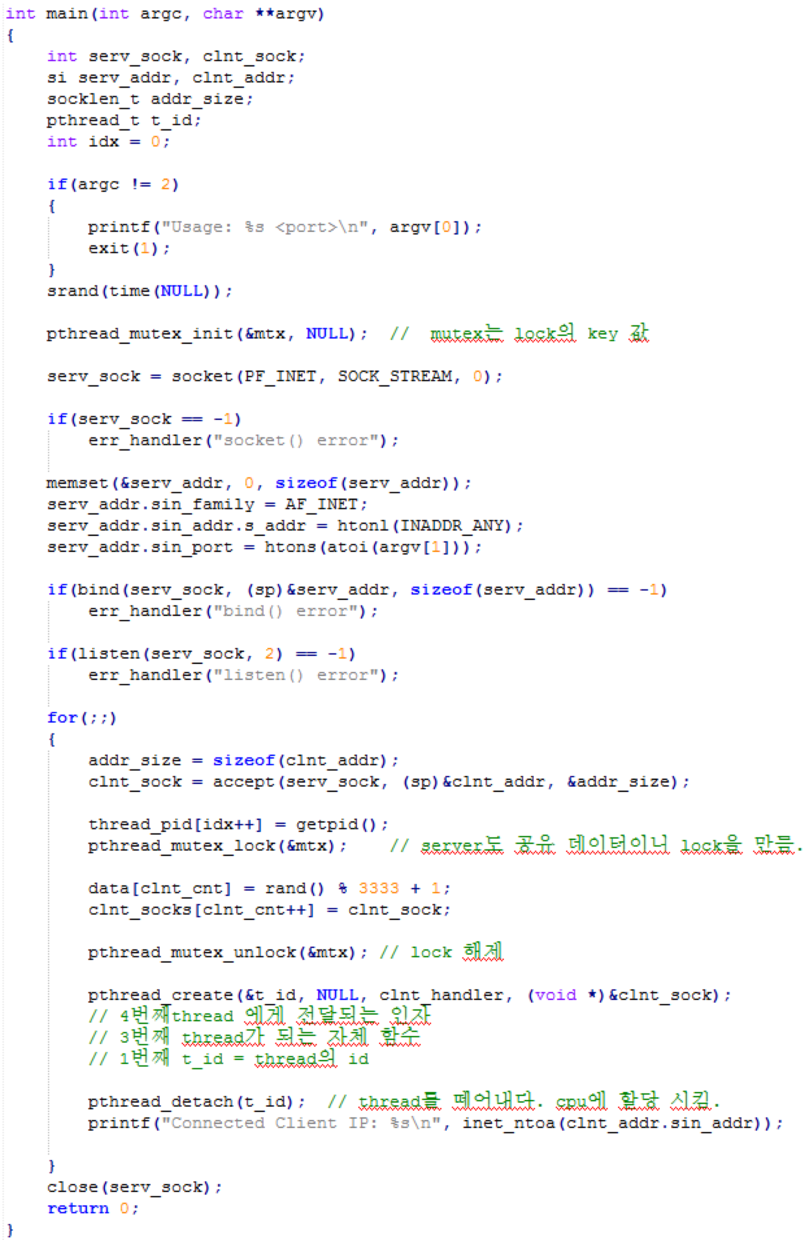
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <pthread.h>
#include <arpa/inet.h>
#include <sys/socket.h>

#define BUF_SIZE    128
#define MAX_CLNT    256

typedef struct sockaddr_in      si;
typedef struct sockaddr *       sp;

int clnt_cnt = 0;
int clnt_socks[MAX_CLNT];
int data[MAX_CLNT];
int thread_pid[MAX_CLNT];
int idx;
int cnt[MAX_CLNT];
pthread_mutex_t mtx;

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

void sig_handler(int signo)
{
    int i;

    printf("Time Over!\n");

    pthread_mutex_lock(&mtx);

    for(i = 0; i < clnt_cnt; i++)
        if(thread_pid[i] == getpid())
            cnt[i] += 1;

    pthread_mutex_unlock(&mtx);

    alarm(3);
}
```



```c
void proc_msg(char *msg, int len, int k)
{
    int i;
    int cmp = atoi(msg);
    char smsg[64] = {0};

    pthread_mutex_lock(&mtx);

    cnt[k] += 1;  // 숫자 입력 횟수

    if(data[k] > cmp)
        sprintf(smsg, "greater than %d\n", cmp);
    else if(data[k] < cmp)
        sprintf(smsg, "less than %d\n", cmp);
    else
    {
        strcpy(smsg, "You win!\n");
        printf("cnt = %d\n", cnt[k]);
    }

    strcat(smsg, "Input Number: \n");
    write(clnt_socks[k], smsg, strlen(smsg));

    pthread_mutex_unlock(&mtx);
}
```

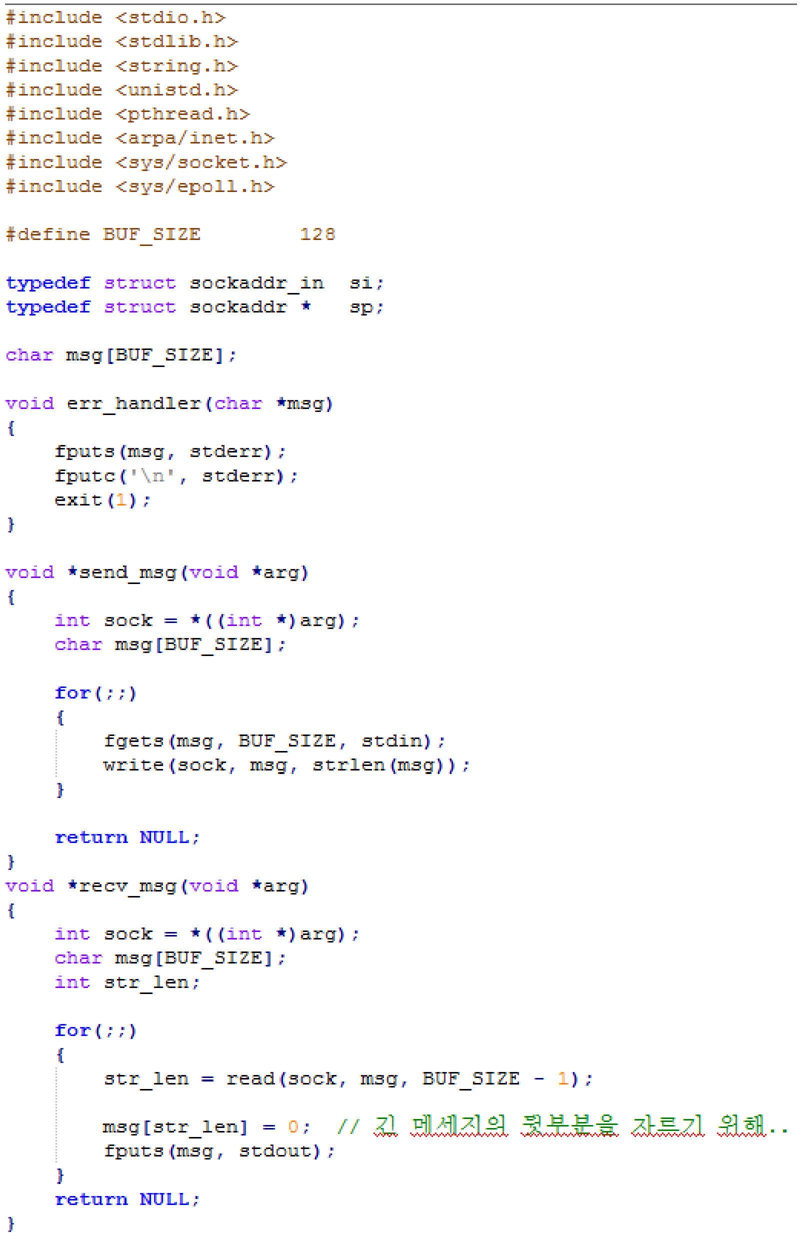


```c
void *clnt_handler(void *arg)
{
    int clnt_sock = *((int *)arg);
    int str_len = 0, i;
    char msg[BUF_SIZE] = {0};
    char pattern[BUF_SIZE] = "Input Number: \n";

    signal(SIGALRM, sig_handler);

    pthread_mutex_lock(&mtx);  // lock
    thread_pid[idx++] = getpid();  // thread의 pid 값
    i = idx - 1;

    printf("i = %d\n", i);

    write(clnt_socks[i], pattern, strlen(pattern));   // critical section
    pthread_mutex_unlock(&mtx); // unlock


    alarm(3);

    while((str_len = read(clnt_sock, msg, sizeof(msg))) != 0)
    {
        alarm(0);
        proc_msg(msg, str_len, i);
        alarm(3);
    }

    pthread_mutex_lock(&mtx);

    for(i = 0; i < clnt_cnt; i++)
    {
        if(clnt_sock == clnt_socks[i])
        {
            while(i++ < clnt_cnt - 1)
                clnt_socks[i] = clnt_socks[i + 1];
            break;
        }
    }

    clnt_cnt--;
    pthread_mutex_unlock(&mtx);
    close(clnt_sock);

    return NULL;
}
```

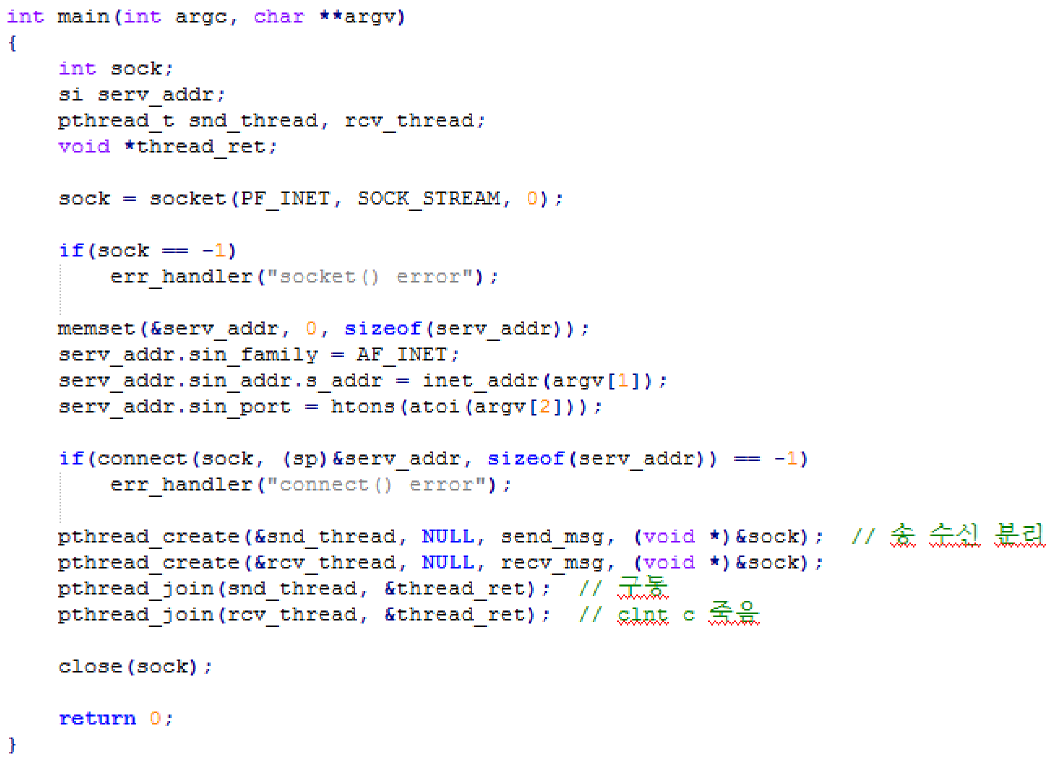


```c
int main(int argc, char **argv)
{
    int serv_sock, clnt_sock;
    si serv_addr, clnt_addr;
    socklen_t addr_size;
    pthread_t t_id;
    int idx = 0;

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

    pthread_mutex_init(&mtx, NULL);  //  mutex는 lock의 key 값

    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, (sp)&serv_addr, sizeof(serv_addr)) == -1)
        err_handler("bind() error");

    if(listen(serv_sock, 2) == -1)
        err_handler("listen() error");

    for(;;)
    {
        addr_size = sizeof(clnt_addr);
        clnt_sock = accept(serv_sock, (sp)&clnt_addr, &addr_size);

        thread_pid[idx++] = getpid();
        pthread_mutex_lock(&mtx);    // server도 공유 데이터이니 lock을 만듬.

        data[clnt_cnt] = rand() % 3333 + 1;
        clnt_socks[clnt_cnt++] = clnt_sock;

        pthread_mutex_unlock(&mtx); // lock 해제

        pthread_create(&t_id, NULL, clnt_handler, (void *)&clnt_sock);
        // 4번째thread 에게 전달되는 인자
        // 3번째 thread가 되는 자체 함수
        // 1번째 t_id = thread의 id

        pthread_detach(t_id);  // thread를 떼어내다. cpu에 할당 시킴.
        printf("Connected Client IP: %s\n", inet_ntoa(clnt_addr.sin_addr));

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

2. 네트워크 프로그래밍 – up & down game (client)



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/epoll.h>

#define BUF_SIZE        128

typedef struct sockaddr_in  si;
typedef struct sockaddr *   sp;

char msg[BUF_SIZE];

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

void *send_msg(void *arg)
{
    int sock = *((int *)arg);
    char msg[BUF_SIZE];

    for(;;)
    {
        fgets(msg, BUF_SIZE, stdin);
        write(sock, msg, strlen(msg));
    }

    return NULL;
}
void *recv_msg(void *arg)
{
    int sock = *((int *)arg);
    char msg[BUF_SIZE];
    int str_len;

    for(;;)
    {
        str_len = read(sock, msg, BUF_SIZE - 1);

        msg[str_len] = 0;  // 긴 메세지의 뒷부분을 자르기 위해..
        fputs(msg, stdout);
    }
    return NULL;
}
```



```c
int main(int argc, char **argv)
{
    int sock;
    si serv_addr;
    pthread_t snd_thread, rcv_thread;
    void *thread_ret;

    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, (sp)&serv_addr, sizeof(serv_addr)) == -1)
        err_handler("connect() error");

    pthread_create(&snd_thread, NULL, send_msg, (void *)&sock);  // 송 수신 분리
    pthread_create(&rcv_thread, NULL, recv_msg, (void *)&sock);
    pthread_join(snd_thread, &thread_ret);  // 구동
    pthread_join(rcv_thread, &thread_ret);  // clnt c 죽음

    close(sock);

    return 0;
}
```