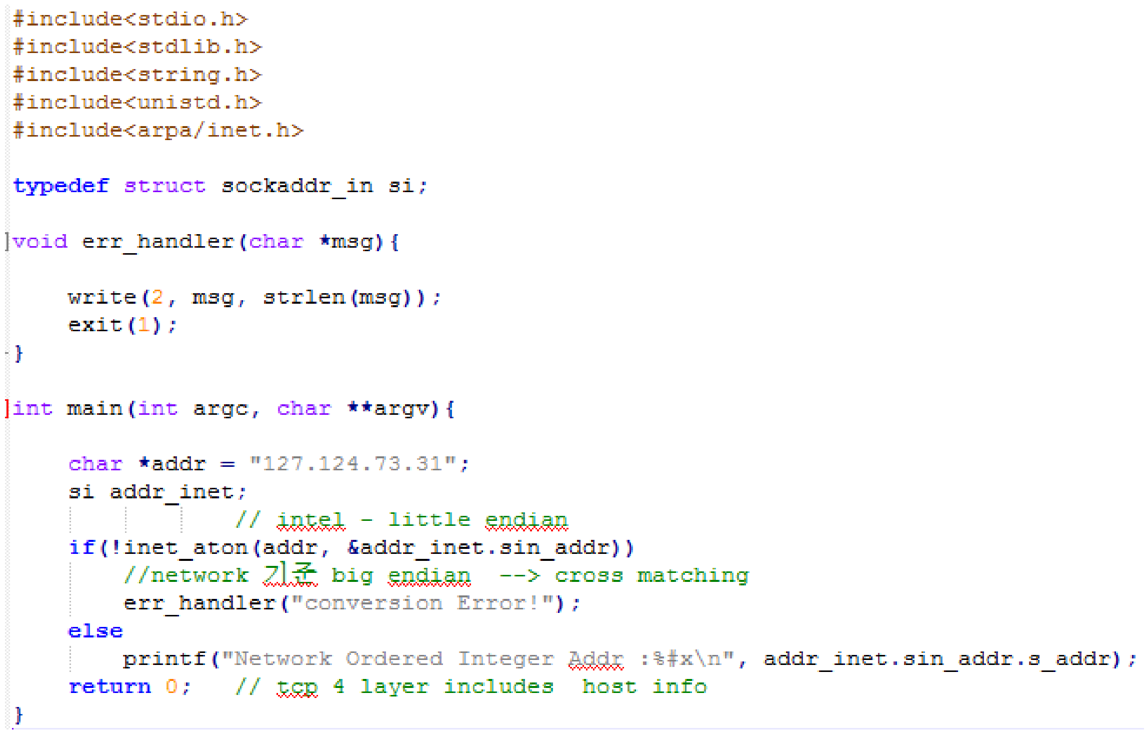
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

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

학생 – GJ (박현우)
uc820@naver.com

2. 네트워크 프로그래밍 – inet_aton



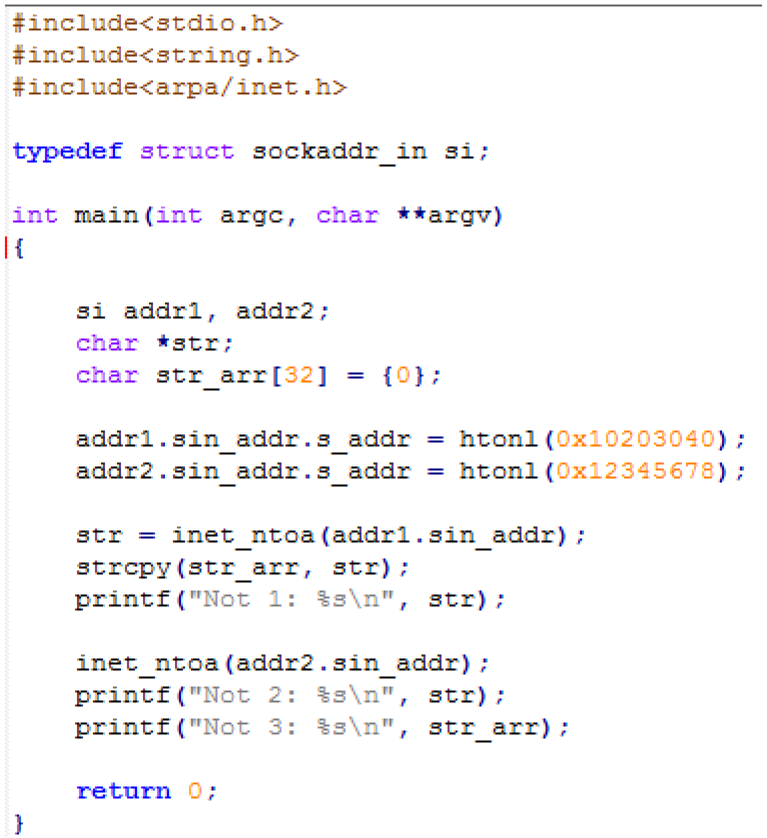
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>

typedef struct sockaddr_in si;

void err_handler(char *msg){

    write(2, msg, strlen(msg));
    exit(1);
}

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

    char *addr = "127.124.73.31";
    si addr_inet;
                // intel - little endian
    if(!inet_aton(addr, &addr_inet.sin_addr))
        //network 기준 big endian  --> cross matching
        err_handler("conversion Error!");
    else
        printf("Network Ordered Integer Addr :%#x\n", addr_inet.sin_addr.s_addr);
    return 0;   // tcp 4 layer includes  host info
}
```

- inet_aton()

Network 기준으로 ip를 변환해준다.

Network의 기준은 big endian이다.

2. 네트워크 프로그래밍 – inet_ntoa



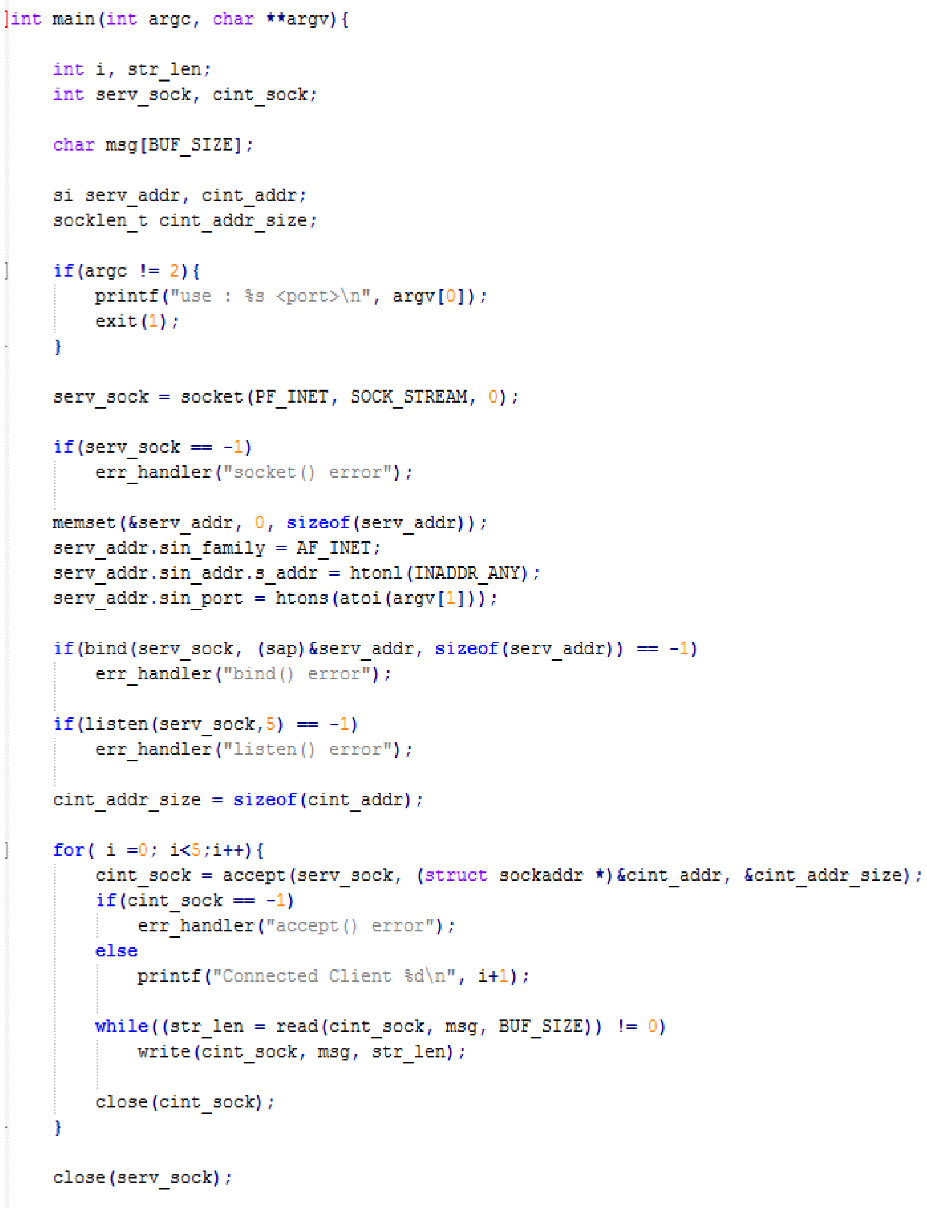
```
#include<stdio.h>
#include<string.h>
#include<arpa/inet.h>

typedef struct sockaddr_in si;

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

    si addr1, addr2;
    char *str;
    char str_arr[32] = {0};

    addr1.sin_addr.s_addr = htonl(0x10203040);
    addr2.sin_addr.s_addr = htonl(0x12345678);

    str = inet_ntoa(addr1.sin_addr);
    strcpy(str_arr, str);
    printf("Not 1: %s\n", str);

    inet_ntoa(addr2.sin_addr);
    printf("Not 2: %s\n", str);
    printf("Not 3: %s\n", str_arr);

    return 0;
}
```

- inet_ntoa()

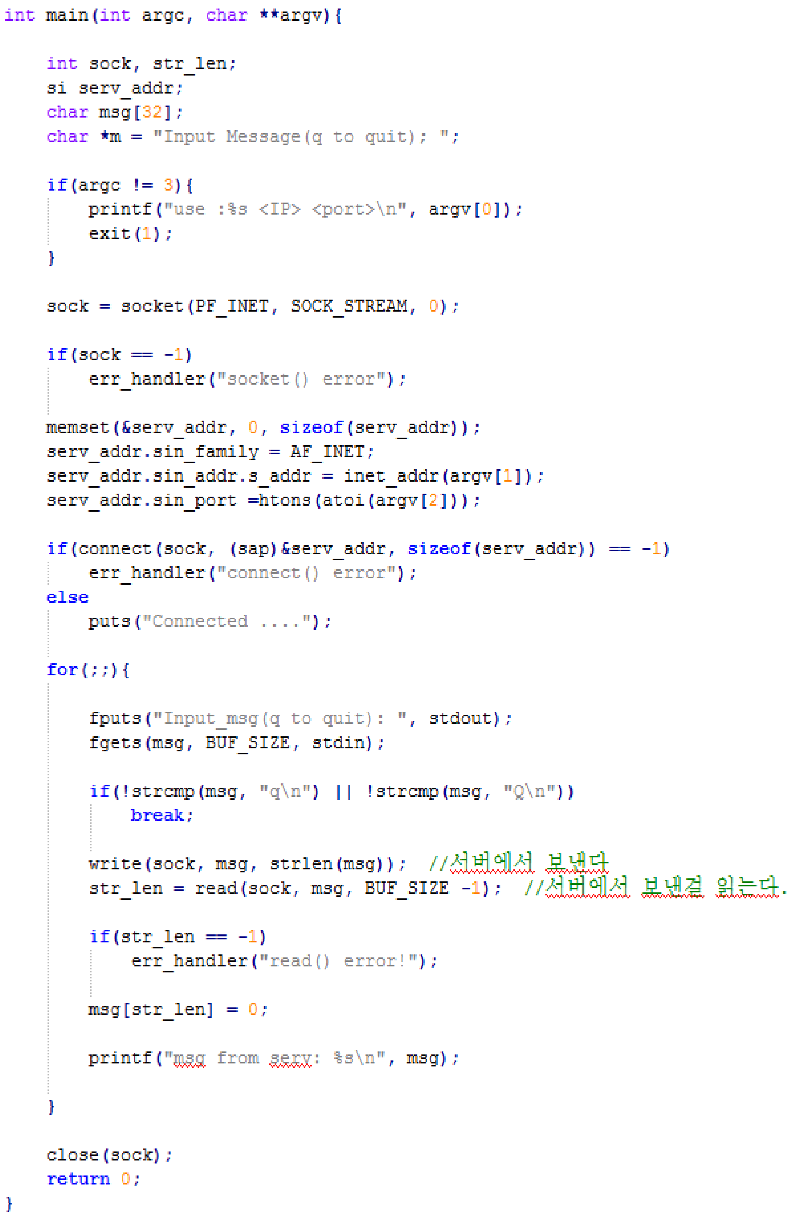
Network 기준으로 ip를 변환된 ip를 각 컴퓨터에 맞게 변환 시킨다.

예를 들면, big endian 에서 big endian
big endian 에서 little endian으로 된다.

2. 네트워크 프로그래밍 – echo 예제(echo_server & client)



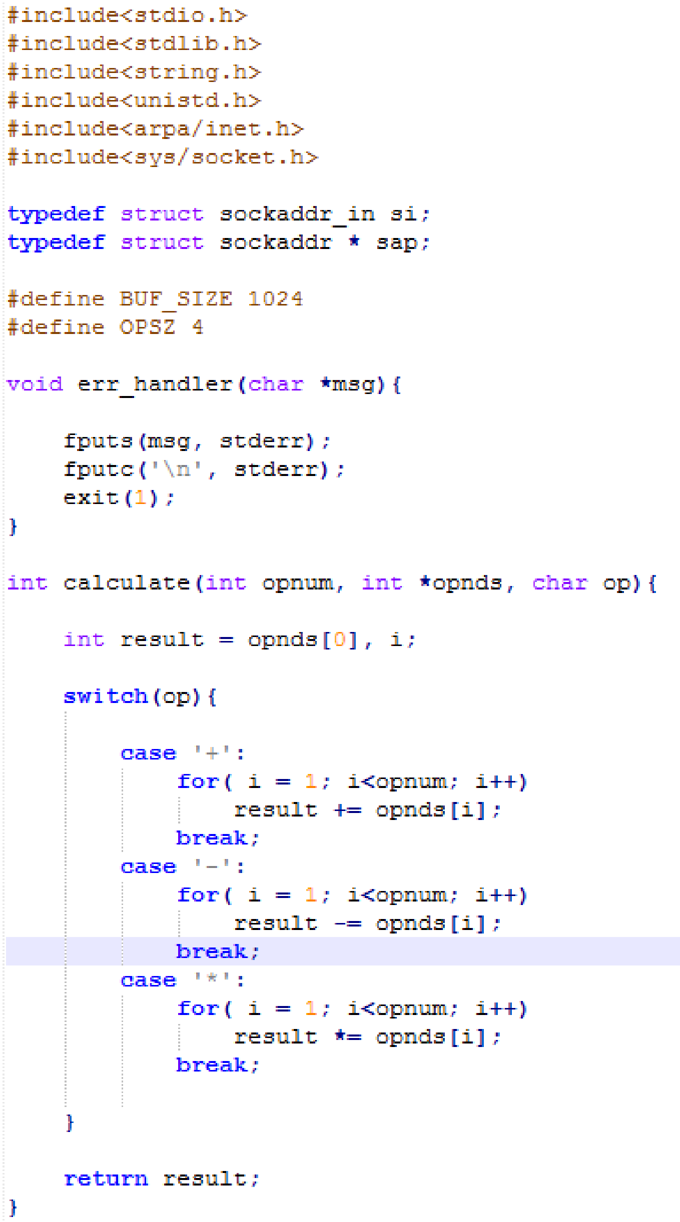
```c
int main(int argc, char **argv){

    int i, str_len;
    int serv_sock, cint_sock;

    char msg[BUF_SIZE];

    si serv_addr, cint_addr;
    socklen_t cint_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");

    cint_addr_size = sizeof(cint_addr);

    for( i =0; i<5;i++){
        cint_sock = accept(serv_sock, (struct sockaddr *)&cint_addr, &cint_addr_size);
        if(cint_sock == -1)
            err_handler("accept() error");
        else
            printf("Connected Client %d\n", i+1);

        while((str_len = read(cint_sock, msg, BUF_SIZE)) != 0)
            write(cint_sock, msg, str_len);

        close(cint_sock);
    }

    close(serv_sock);
```



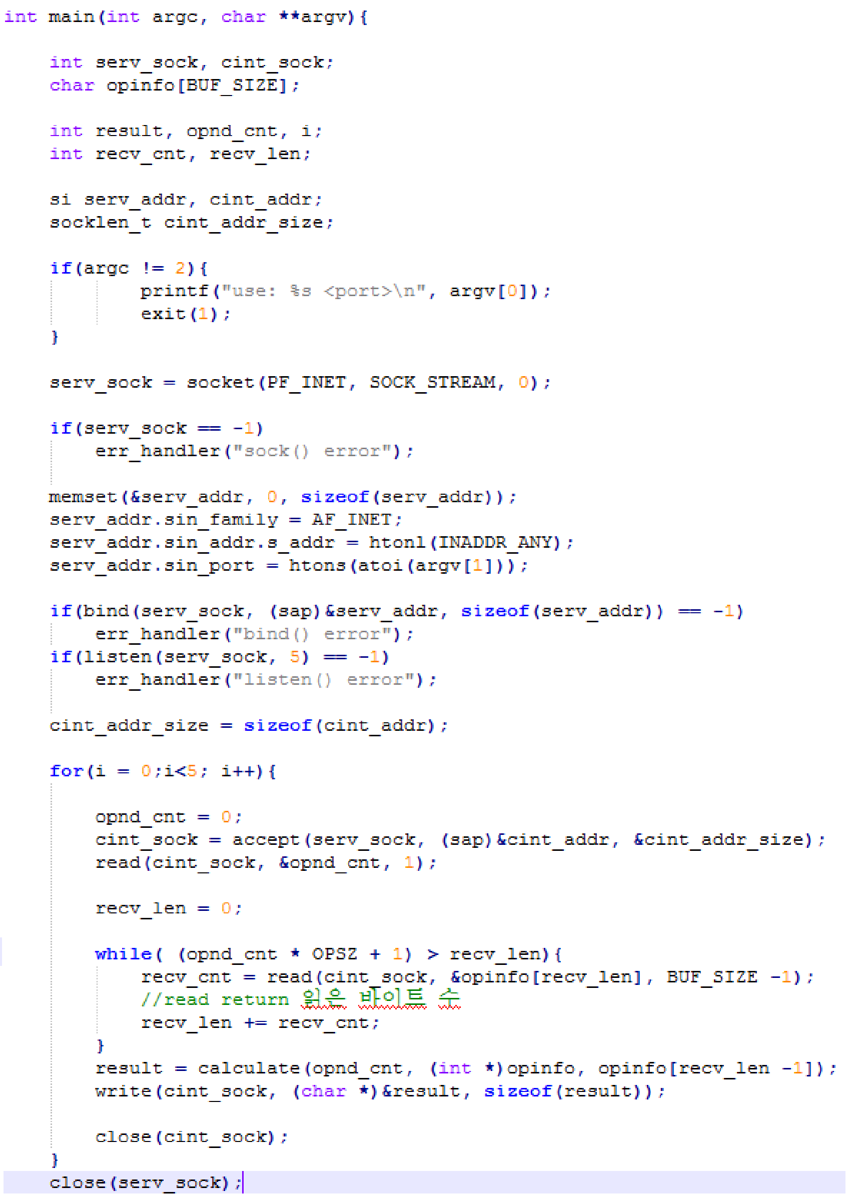
```c
int main(int argc, char **argv){

    int sock, str_len;
    si serv_addr;
    char msg[32];
    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]);
    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 msg(q to quit): ", stdout);
        fgets(msg, BUF_SIZE, stdin);

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

        write(sock, msg, strlen(msg));  //서버에서 보낸다
        str_len = read(sock, msg, BUF_SIZE -1);  //서버에서 보낸걸 읽는다.

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

        msg[str_len] = 0;

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

    }

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

- 단순히 client에서 읽어온 것을 다시 client로 보내는 예제이다.

2. 네트워크 프로그래밍 – 계산기 예제(op_server)



```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
#define OPSZ 4

void err_handler(char *msg){

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

int calculate(int opnum, int *opnds, char op){

    int result = opnds[0], i;

    switch(op){

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

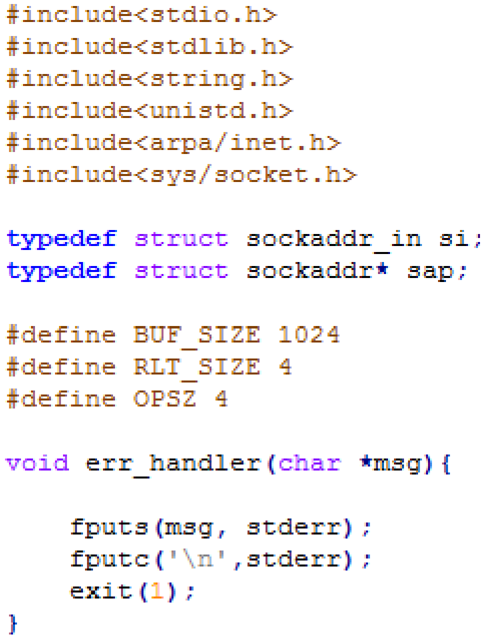


```c
int main(int argc, char **argv){

    int serv_sock, cint_sock;
    char opinfo[BUF_SIZE];

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

    si serv_addr, cint_addr;
    socklen_t cint_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("sock() 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");

    cint_addr_size = sizeof(cint_addr);

    for(i = 0;i<5; i++){

        opnd_cnt = 0;
        cint_sock = accept(serv_sock, (sap)&cint_addr, &cint_addr_size);
        read(cint_sock, &opnd_cnt, 1);

        recv_len = 0;

        while( (opnd_cnt * OPSZ + 1) > recv_len){
            recv_cnt = read(cint_sock, &opinfo[recv_len], BUF_SIZE -1);
            //read return 읽은 바이트 수
            recv_len += recv_cnt;
        }
        result = calculate(opnd_cnt, (int *)opinfo, opinfo[recv_len -1]);
        write(cint_sock, (char *)&result, sizeof(result));

        close(cint_sock);
    }
    close(serv_sock);
```

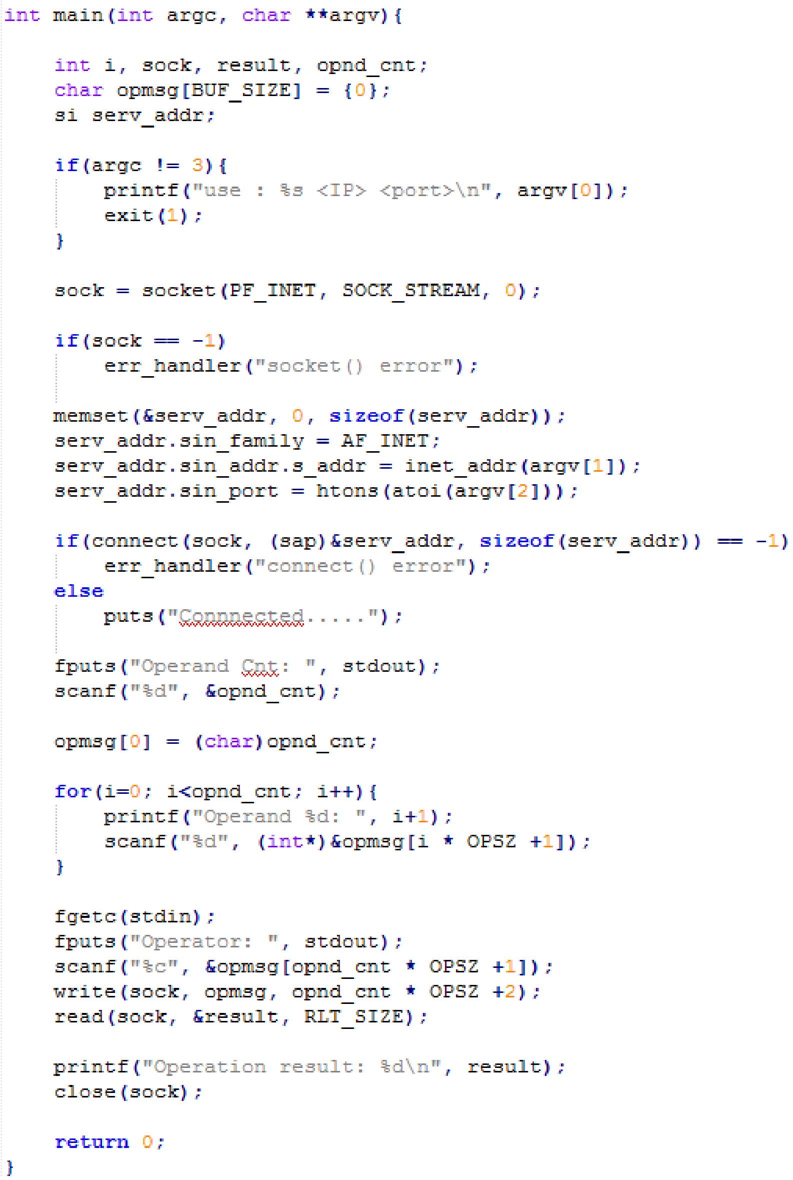
- client로부터 read를 통해 읽어온 숫자와 연산자를 calculate()로 계산한 후 결과값을 client에 write로 보낸다.

- while()이 있는 client로부터의 결과값이 다 전송 받지 못 했을 경우를 위해서 사용된다.

2. 네트워크 프로그래밍 – 계산기 예제(op_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
#define RLT_SIZE 4
#define OPSZ 4

void err_handler(char *msg){

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

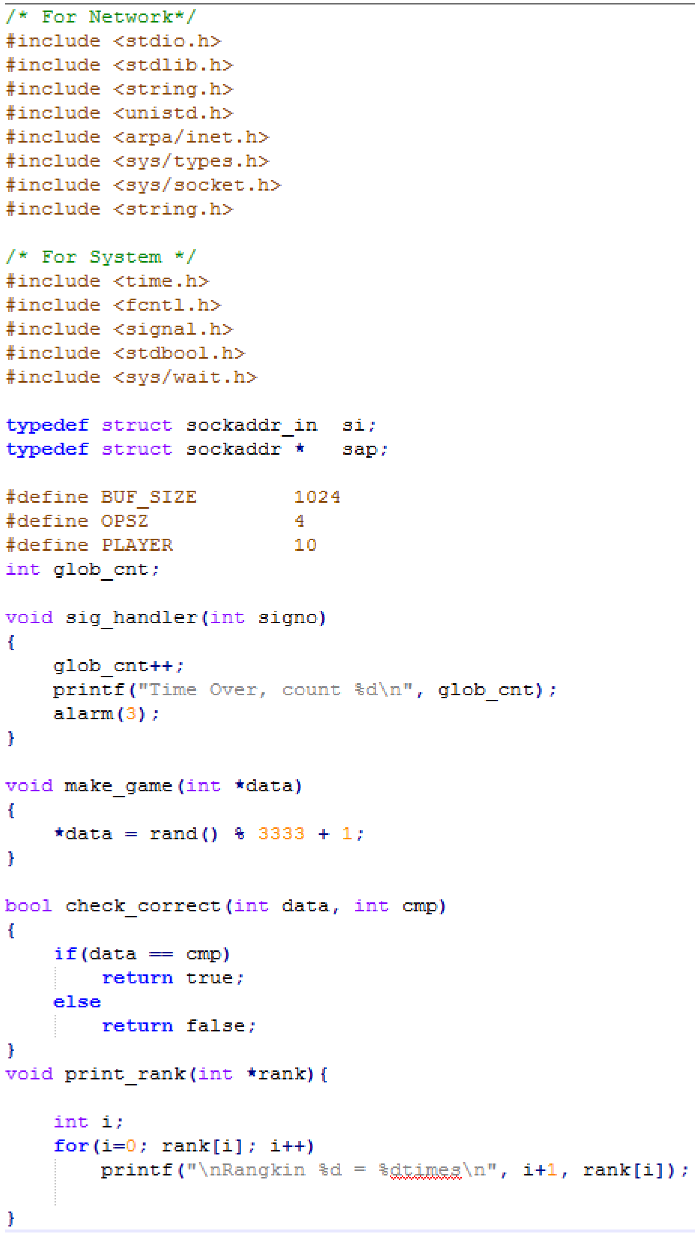


```c
int main(int argc, char **argv){

    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));
    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("Connnected.....");

    fputs("Operand Cnt: ", stdout);
    scanf("%d", &opnd_cnt);

    opmsg[0] = (char)opnd_cnt;

    for(i=0; i<opnd_cnt; i++){
        printf("Operand %d: ", i+1);
        scanf("%d", (int*)&opmsg[i * OPSZ +1]);
    }

    fgetc(stdin);
    fputs("Operator: ", stdout);
    scanf("%c", &opmsg[opnd_cnt * OPSZ +1]);
    write(sock, opmsg, opnd_cnt * OPSZ +2);
    read(sock, &result, RLT_SIZE);

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

    return 0;
}
```

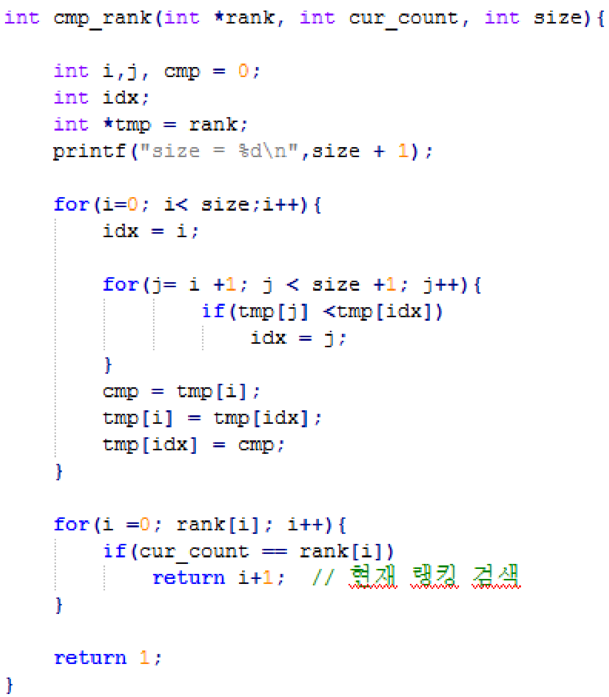
- Opmsg에 숫자와 연산자를 실어 server에 보낸다.

- Server에서 read로 결과값을 읽어와서 result에 입력 후 결과값을 출력한다.

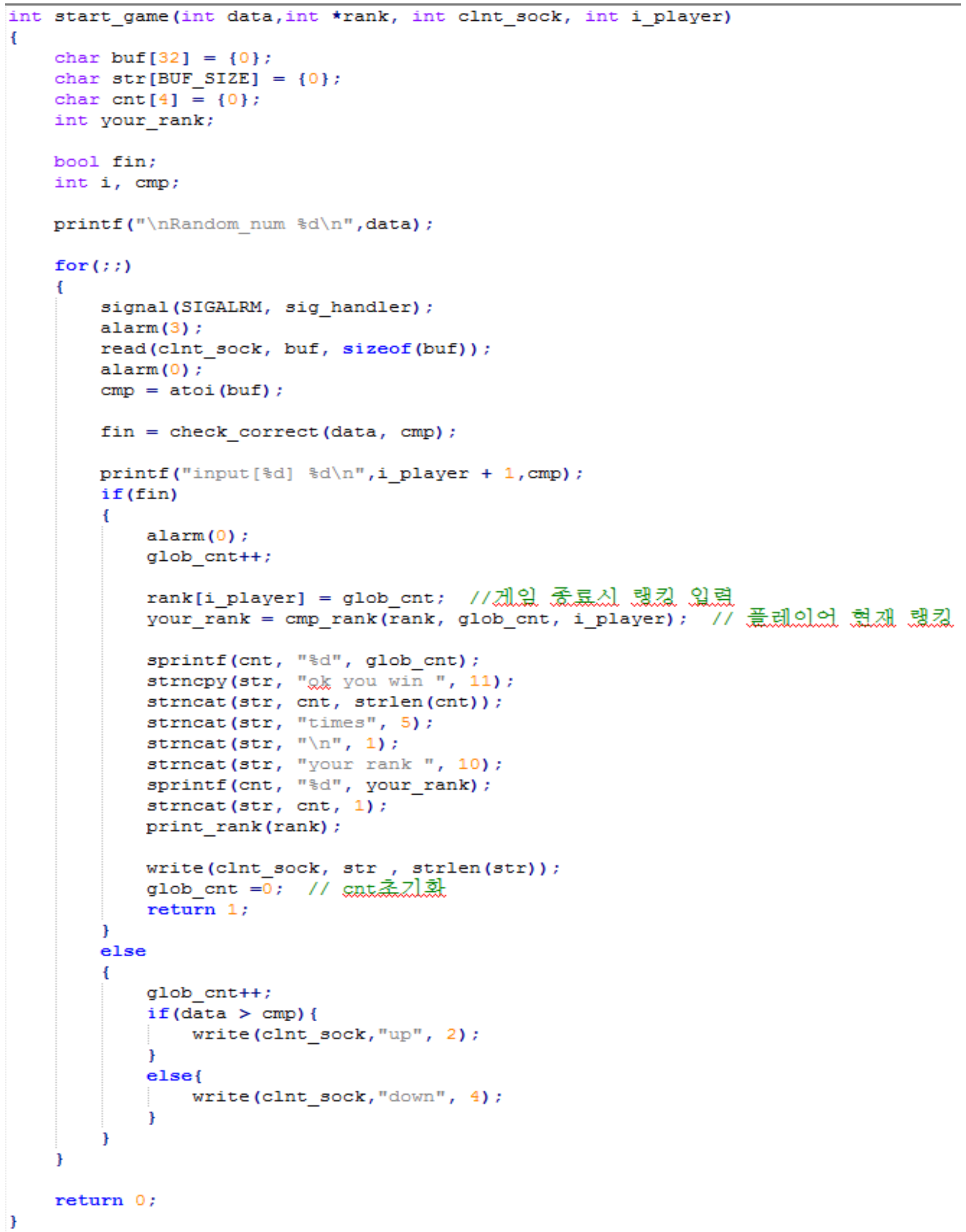
2. 네트워크 프로그래밍 – 숫자 맞추기 게임(server 1)



```c
/* For Network*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <string.h>

/* For System */
#include <time.h>
#include <fcntl.h>
#include <signal.h>
#include <stdbool.h>
#include <sys/wait.h>

typedef struct sockaddr_in  si;
typedef struct sockaddr *   sap;

#define BUF_SIZE        1024
#define OPSZ            4
#define PLAYER          10
int glob_cnt;

void sig_handler(int signo)
{
    glob_cnt++;
    printf("Time Over, count %d\n", glob_cnt);
    alarm(3);
}

void make_game(int *data)
{
    *data = rand() % 3333 + 1;
}

bool check_correct(int data, int cmp)
{
    if(data == cmp)
        return true;
    else
        return false;
}
void print_rank(int *rank){

    int i;
    for(i=0; rank[i]; i++)
        printf("\nRangkin %d = %dtimes\n", i+1, rank[i]);

}
```

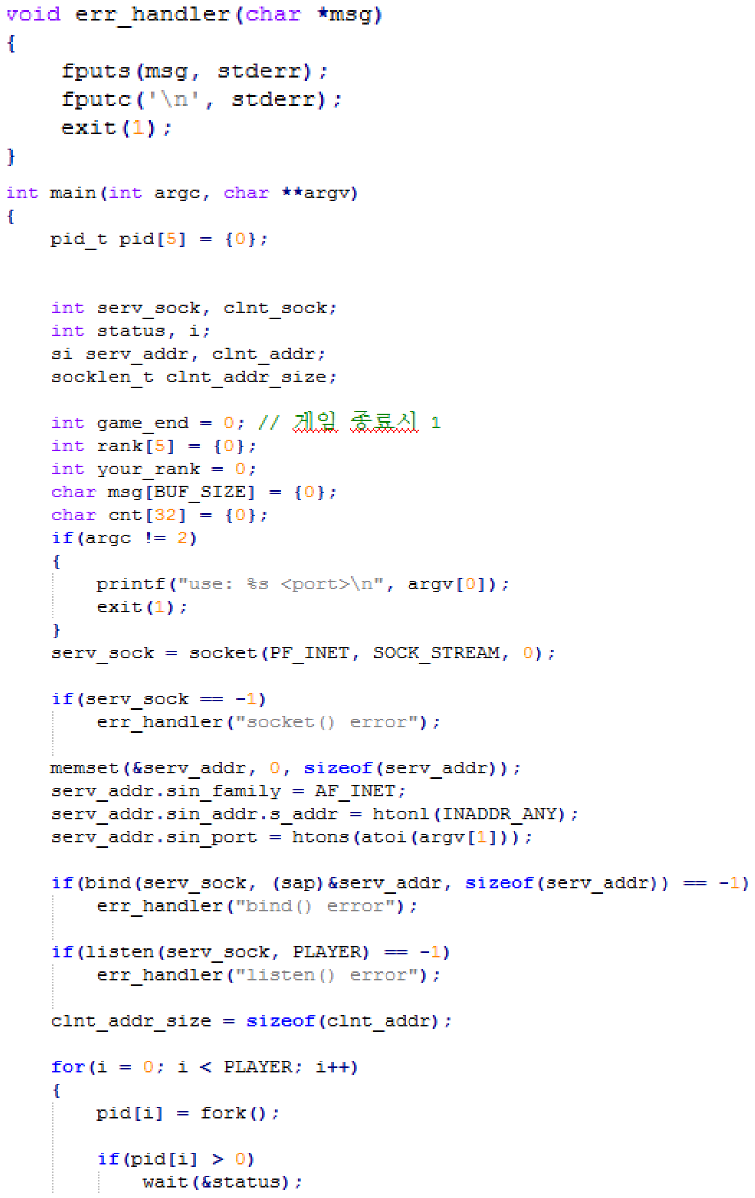


```c
int cmp_rank(int *rank, int cur_count, int size){

    int i,j, cmp = 0;
    int idx;
    int *tmp = rank;
    printf("size = %d\n",size + 1);

    for(i=0; i< size;i++){
        idx = i;

        for(j= i +1; j < size +1; j++){
            if(tmp[j] <tmp[idx])
                idx = j;
        }
        cmp = tmp[i];
        tmp[i] = tmp[idx];
        tmp[idx] = cmp;
    }

    for(i =0; rank[i]; i++){
        if(cur_count == rank[i])
            return i+1;  // 현재 랭킹 검색
    }

    return 1;
}
```

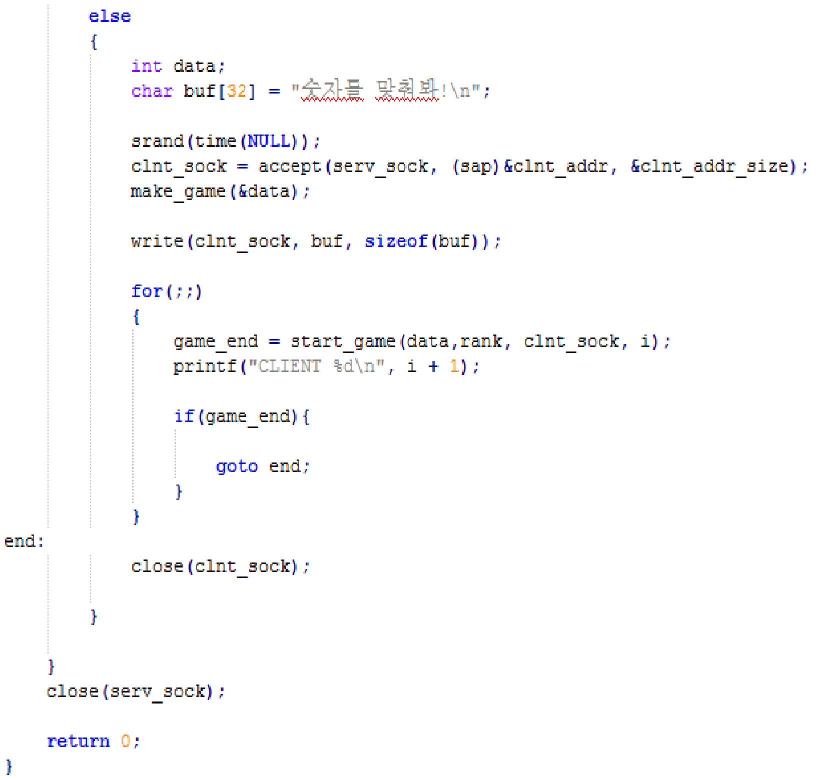


```c
int start_game(int data,int *rank, int clnt_sock, int i_player)
{
    char buf[32] = {0};
    char str[BUF_SIZE] = {0};
    char cnt[4] = {0};
    int your_rank;

    bool fin;
    int i, cmp;

    printf("\nRandom_num %d\n",data);

    for(;;)
    {
        signal(SIGALRM, sig_handler);
        alarm(3);
        read(clnt_sock, buf, sizeof(buf));
        alarm(0);
        cmp = atoi(buf);

        fin = check_correct(data, cmp);

        printf("input[%d] %d\n",i_player + 1,cmp);
        if(fin)
        {
            alarm(0);
            glob_cnt++;

            rank[i_player] = glob_cnt;  //게임 종료시 랭킹 입력
            your_rank = cmp_rank(rank, glob_cnt, i_player);  // 플레이어 현재 랭킹

            sprintf(cnt, "%d", glob_cnt);
            strncpy(str, "ok you win ", 11);
            strncat(str, cnt, strlen(cnt));
            strncat(str, "times", 5);
            strncat(str, "\n", 1);
            strncat(str, "your rank ", 10);
            sprintf(cnt, "%d", your_rank);
            strncat(str, cnt, 1);
            print_rank(rank);

            write(clnt_sock, str , strlen(str));
            glob_cnt =0;  // cnt초기화
            return 1;
        }
        else
        {
            glob_cnt++;
            if(data > cmp){
                write(clnt_sock,"up", 2);
            }
            else{
                write(clnt_sock,"down", 4);
            }
        }
    }

    return 0;
}
```

2. 네트워크 프로그래밍 – 숫자 맞추기 게임(server 2)

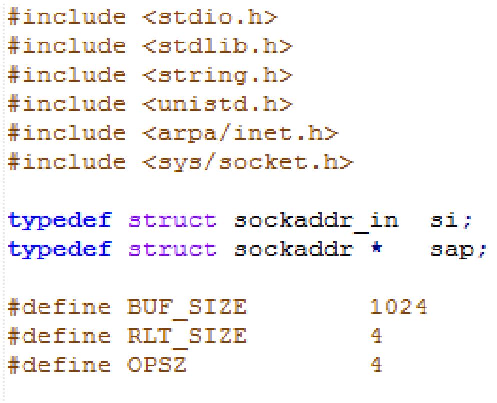


```c
void err_handler(char *msg)
{
    fputs(msg, stderr);
    fputc('\n', stderr);
    exit(1);
}
int main(int argc, char **argv)
{
    pid_t pid[5] = {0};


    int serv_sock, clnt_sock;
    int status, i;
    si serv_addr, clnt_addr;
    socklen_t clnt_addr_size;

    int game_end = 0; // 게임 종료시 1
    int rank[5] = {0};
    int your_rank = 0;
    char msg[BUF_SIZE] = {0};
    char cnt[32] = {0};
    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, PLAYER) == -1)
        err_handler("listen() error");

    clnt_addr_size = sizeof(clnt_addr);

    for(i = 0; i < PLAYER; i++)
    {
        pid[i] = fork();

        if(pid[i] > 0)
            wait(&status);
```

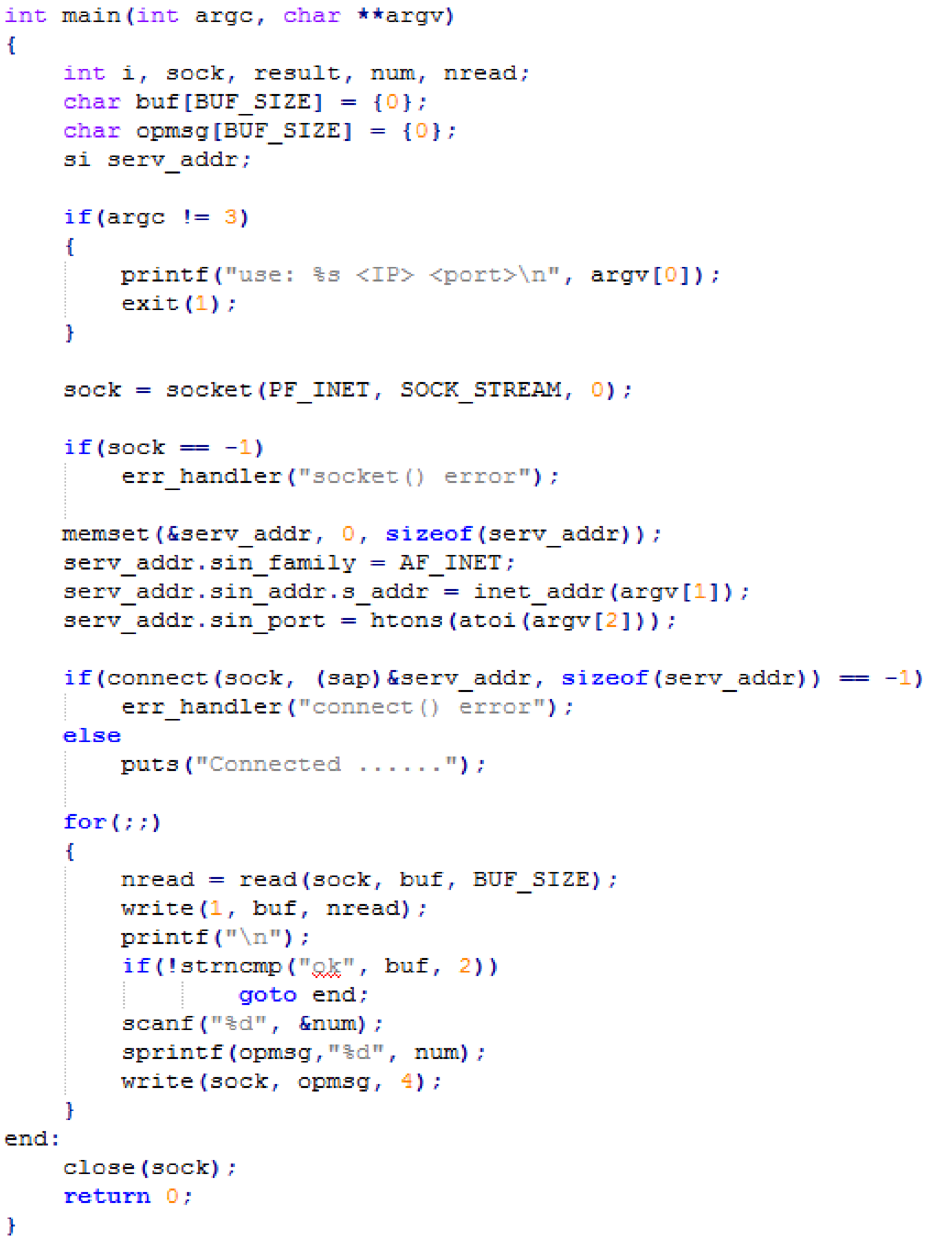


```c
        else
        {
            int data;
            char buf[32] = "숫자를 맞춰봐!\n";

            srand(time(NULL));
            clnt_sock = accept(serv_sock, (sap)&clnt_addr, &clnt_addr_size);
            make_game(&data);

            write(clnt_sock, buf, sizeof(buf));

            for(;;)
            {
                game_end = start_game(data,rank, clnt_sock, i);
                printf("CLIENT %d\n", i + 1);

                if(game_end){

                    goto end;
                }
            }
end:
            close(clnt_sock);

        }

    }
    close(serv_sock);

    return 0;
}
```

2. 네트워크 프로그래밍 – 숫자 맞추기 게임(client 1)

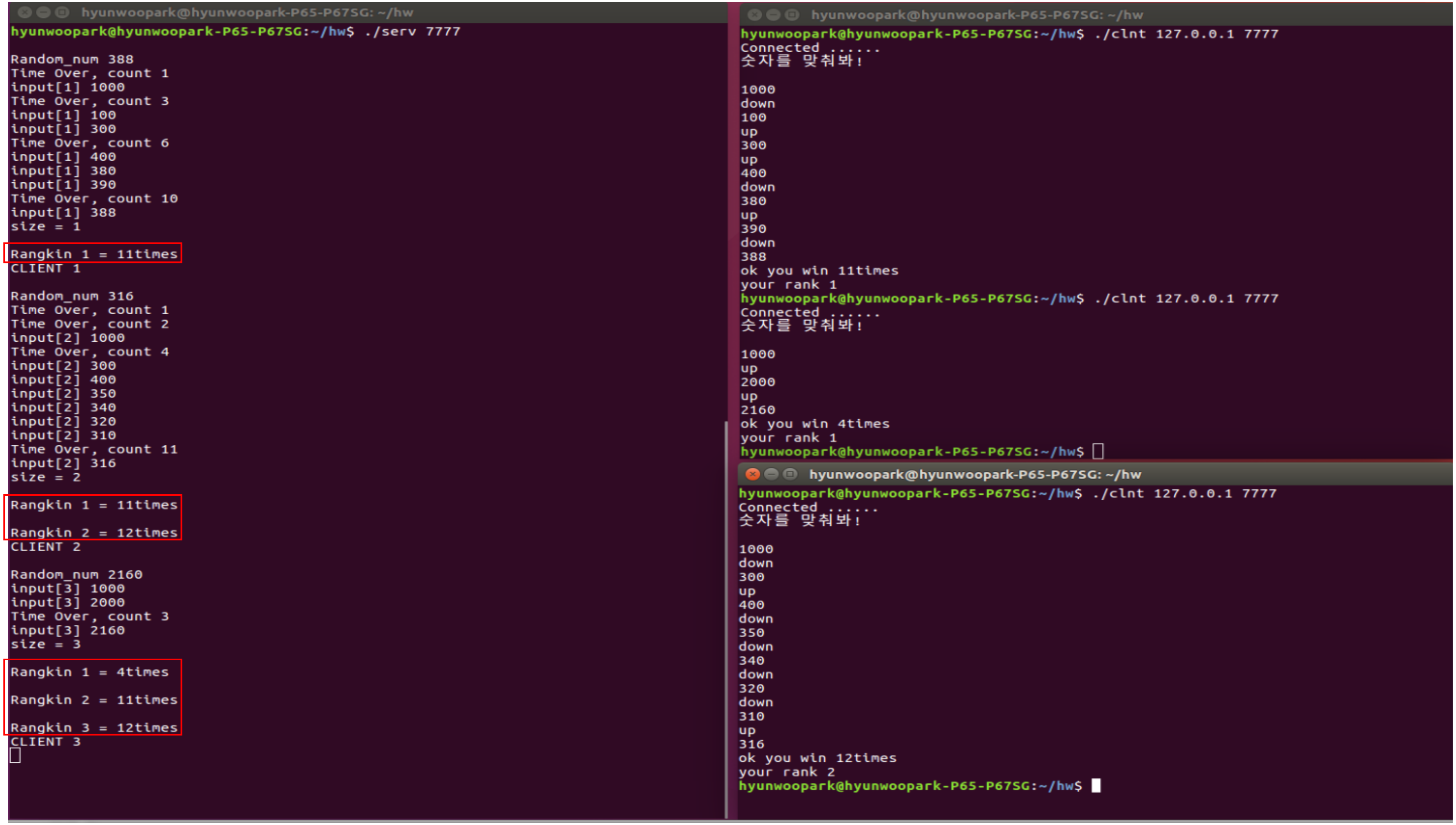


```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>

typedef struct sockaddr_in  si;
typedef struct sockaddr *   sap;

#define BUF_SIZE        1024
#define RLT_SIZE        4
#define OPSZ            4
```



```c
int main(int argc, char **argv)
{
    int i, sock, result, num, nread;
    char buf[BUF_SIZE] = {0};
    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));
    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(;;)
    {
        nread = read(sock, buf, BUF_SIZE);
        write(1, buf, nread);
        printf("\n");
        if(!strncmp("ok", buf, 2))
                goto end;
        scanf("%d", &num);
        sprintf(opmsg,"%d", num);
        write(sock, opmsg, 4);
    }
end:
    close(sock);
    return 0;
}
```

2. 네트워크 프로그래밍 – 숫자 맞추기 게임(Result 1)

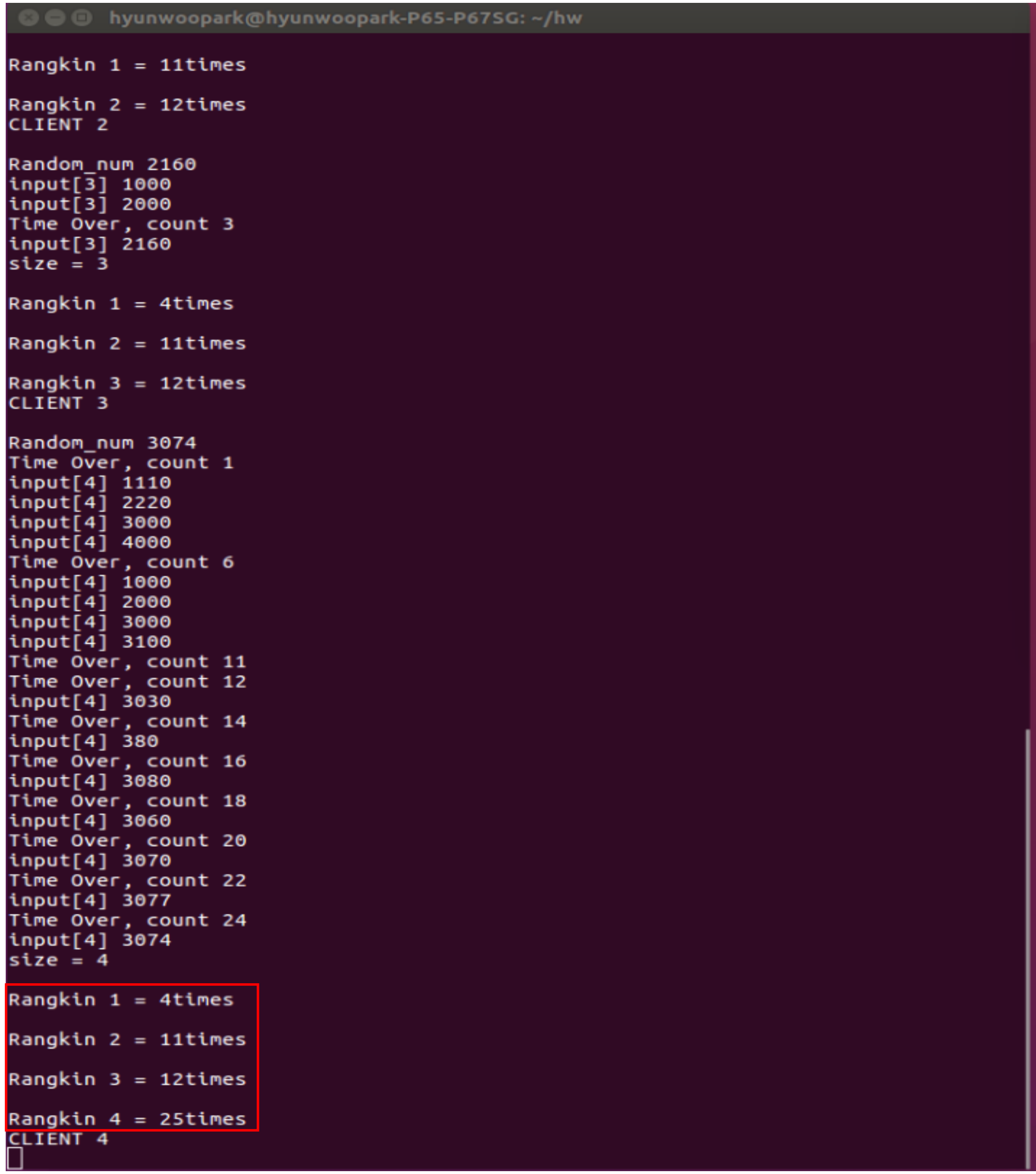

hyunwoopark@hyunwoopark-P65-P67SG: ~/hw
hyunwoopark@hyunwoopark-P65-P67SG: ~/hw
hyunwoopark@hyunwoopark-P65-P67SG: ~/hw

```
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$ ./serv 7777

Random_num 388
Time Over, count 1
input[1] 1000
Time Over, count 3
input[1] 100
input[1] 300
Time Over, count 6
input[1] 400
input[1] 380
input[1] 390
Time Over, count 10
input[1] 388
size = 1

Rangkin 1 = 11times
CLIENT 1

Random_num 316
Time Over, count 1
Time Over, count 2
input[2] 1000
Time Over, count 4
input[2] 300
input[2] 400
input[2] 350
input[2] 340
input[2] 320
input[2] 310
Time Over, count 11
input[2] 316
size = 2

Rangkin 1 = 11times

Rangkin 2 = 12times
CLIENT 2

Random_num 2160
input[3] 1000
input[3] 2000
Time Over, count 3
input[3] 2160
size = 3

Rangkin 1 = 4times

Rangkin 2 = 11times

Rangkin 3 = 12times
CLIENT 3
```

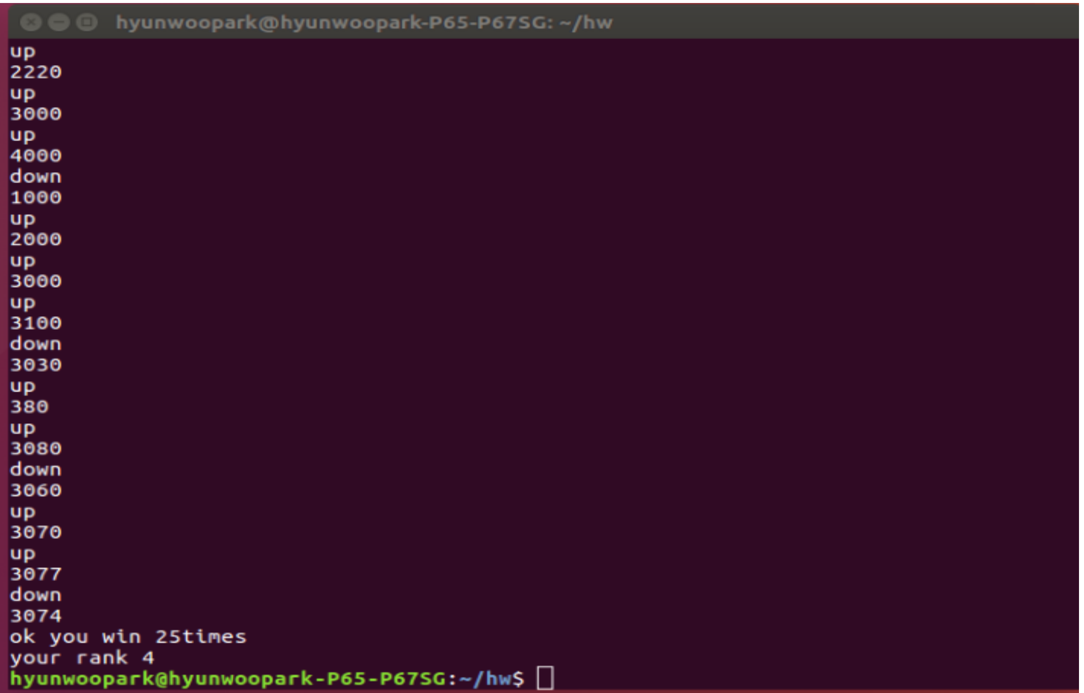
```
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$ ./clnt 127.0.0.1 7777
Connected ......
숫자를 맞춰봐!

1000
down
100
up
300
up
400
down
380
up
390
down
388
ok you win 11times
your rank 1
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$ ./clnt 127.0.0.1 7777
Connected ......
숫자를 맞춰봐!

1000
up
2000
up
2160
ok you win 4times
your rank 1
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$
```

```
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$ ./clnt 127.0.0.1 7777
Connected ......
숫자를 맞춰봐!

1000
down
300
up
400
down
350
down
340
down
320
down
310
up
316
ok you win 12times
your rank 2
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$
```

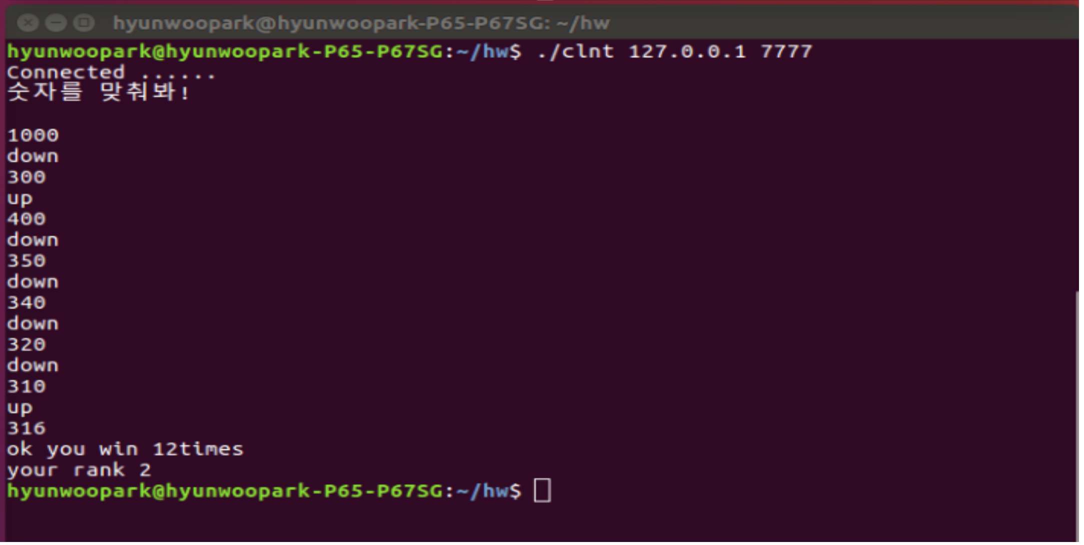
2. 네트워크 프로그래밍 – 숫자 맞추기 게임(Result 2)



```

Rangkin 1 = 11times

Rangkin 2 = 12times
CLIENT 2

Random_num 2160
input[3] 1000
input[3] 2000
Time Over, count 3
input[3] 2160
size = 3

Rangkin 1 = 4times

Rangkin 2 = 11times

Rangkin 3 = 12times
CLIENT 3

Random_num 3074
Time Over, count 1
input[4] 1110
input[4] 2220
input[4] 3000
input[4] 4000
Time Over, count 6
input[4] 1000
input[4] 2000
input[4] 3000
input[4] 3100
Time Over, count 11
Time Over, count 12
input[4] 3030
Time Over, count 14
input[4] 380
Time Over, count 16
input[4] 3080
Time Over, count 18
input[4] 3060
Time Over, count 20
input[4] 3070
Time Over, count 22
input[4] 3077
Time Over, count 24
input[4] 3074
size = 4

Rangkin 1 = 4times

Rangkin 2 = 11times

Rangkin 3 = 12times

Rangkin 4 = 25times
CLIENT 4
```



```
up
2220
up
3000
up
4000
down
1000
up
2000
up
3000
up
3100
down
3030
up
380
up
3080
down
3060
up
3070
up
3077
down
3074
ok you win 25times
your rank 4
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$
```



```
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$ ./clnt 127.0.0.1 7777
Connected ......
숫자를 맞춰봐!

1000
down
300
up
400
down
350
down
340
down
320
down
310
up
316
ok you win 12times
your rank 2
hyunwoopark@hyunwoopark-P65-P67SG:~/hw$
```