

# REPORT

## 보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,  
성.군.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 컴퓨터 네트워크 개론  
과 제 명 : HW 1  
학 과 : 컴퓨터공학과  
학 번 : 2013314967  
이 름 : 정기빈

## 1. Development Environment

OS : Windows 7 64 bits

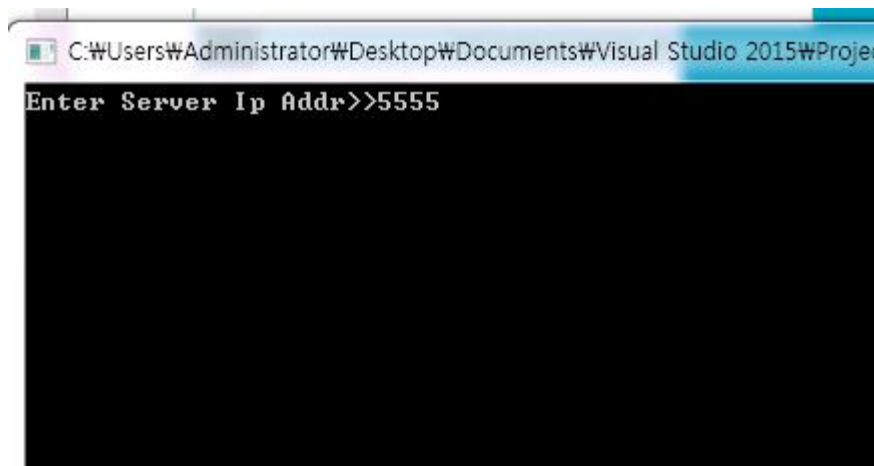
PL : C Language

Compiler Version : 14.0 (Visual 2015)

## 2. Server Description

### 2.1 Get Information

초기화면에서 포트 넘버를 설정 가능하다.



### 2.2 Accept multiple client

다수의 클라이언트를 수용하기 위한 변수이다.

- SOCKET clint\_soc[USER\_NUM]
- int user\_num
- char user\_list[USER\_NUM][50]
- byte user\_flag[USER\_NUM]

```
*#define USER_NUM 5000
```

클라이언트에게 0~4999 중 고유의 번호(index)를 순차적으로 부여하여, 이를 통해 각 클라이언트를 관리한다. 시스템을 간단히 하기 위해, 클라이언트 종료 시 index를 갱신하지 않고, 대신에 user\_flag를 FALSE로 설정함으로써 종료하였음을 알 수 있도록 하였다.

- user\_num은 현재 수용 중인 클라이언트의 수를 나타낸다.
- clint\_soc은 각 클라이언트의 소켓이다.
- user\_list는 각 클라이언트의 id이다.
- user\_flag는 클라이언트의 접속을 나타낸다. 클라이언트 종료 시 FALSE로 설정한다

다수의 클라이언트를 수용하기 위해 쓰레드 API beginthreadex()를 사용하였다. 클라이언트 접속 시 쓰레드를 생성하여 클라이언트의 request를 전달하도록 하고, 주 쓰레드는 다시 클라이언트의 접속 요청에 대기한다. 해당 클라이언트가 접속을 종료하면 소켓을 종료하고, 해당 user\_flag를 FALSE로 설정한다.

```
while (1)
{
    if (listen(serv_soc, USER_NUM) == SOCKET_ERROR)
        error_handle("listen() error!");

    len_clint_addr = sizeof(clint_addr);

    clint_soc[user_num] = accept(serv_soc, (SOCKADDR*)&clint_addr, &len_clint_addr);
    if (clint_soc[user_num] == INVALID_SOCKET)
        error_handle("accept() error!");

    // 쓰레드 생성
    hThread = (HANDLE)_beginthreadex(NULL, 0, thread_func, (void*)user_num, 0, &threadID[user_num]);
    if (0 == hThread)
        error_handle("_beginthreadex() error!");

    user_num = (user_num + 1) % USER_NUM;
}
```

## 2.3 Request Handle

쓰레드 함수에서 서버는 클라이언트로부터 id를 수신하여 위에서 설명한 변수에 클라이언트 정보를 저장한다.

```
// 클라이언트로부터 USER_ID를 받음
msg_len = recv(clint_soc[cur_idx], user_id, sizeof(user_id) - 1, 0);
user_flag[cur_idx] = TRUE;
strcpy(user_list[cur_idx], user_id);
```

그 후 무한루프를 돌면서 클라이언트로부터 수신한 메시지에 strtok 함수를 수행하여 명령어를 판독하고, 해당하는 기능들을 수행한다.

```
while (1)
{
    msg_len = recv(clint_soc[cur_idx], msg, sizeof(msg) - 1, 0);
    if (msg_len > 1)
    {
        char* comm = strtok(msg, " ");

        if (!strcmp(comm, "/who"))
        {
            make_userlist(cur_idx);
        }
    }
}
```

'/q' 메시지 수신 시 반복문을 빠져나오며 user\_flag를 FALSE로 설정하고 해당 소켓을 종료한다.

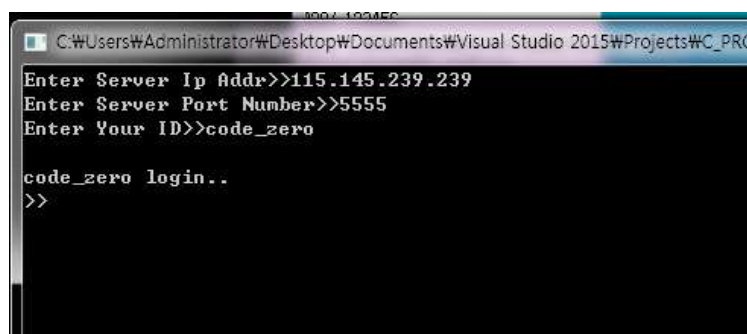
```
    else if (!strcmp(comm, "/q"))
    {
        char buf[BUF_SIZE];
        msg_len = sprintf(buf, "%s logout..", user_id);
        broadcast(buf, msg_len + 1);
        break;
    }
}
user_flag[cur_idx] = FALSE;
closesocket(clint_soc[cur_idx]);

return 0;
}
```

### 3. Client Description

#### 3.1 Get Information

처음 실행 시 사용자로부터 서버 주소, 서버 포트 번호, 사용자 ID를 입력받는다.



```
C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Projects\WC_PRO>
Enter Server Ip Addr>>115.145.239.239
Enter Server Port Number>>5555
Enter Your ID>>code_zero

code_zero login..
>>
```

### 3.2 Receive Process

클라이언트 실행 시 메시지 수신을 위한 쓰레드를 생성한다. 이를 통해 비동기적인 메시지 수신이 가능하다.

- int rec\_on                      // rec\_on =TRUE 시 메시지를 수신한다

// 메인 쓰레드에서 쓰레드 생성

```
// 메시지 수신을 위한 쓰레드 호출
rec_on = TRUE;
hThread = (HANDLE)_beginthreadex(NULL, 0, thread_func, (void*)&rec_on, 0, &threadID);
if (0 == hThread)
    error_handle("_beginthreadex() error!");
```

// 메시지 비동기 수신을 위한 쓰레드 함수

```
UINT WINAPI thread_func(void* para)
{
    int* rec_on = (int*)para;
    char rec_msg[BUF_SIZE];

    while (*rec_on)
    {
        int rec_msg_len = recv(sock, rec_msg, sizeof(rec_msg) - 1, 0);
        if (rec_msg_len != -1)
        {
            puts("");
            puts(rec_msg);
        }
    }
    return 0;
}
```

### 3.3 Request Process

무한루프를 돌며 사용자로부터 엔터전까지 문자열 한 줄을 입력받을 때마다 서버에게 전송한다. '/q' 명령어 입력 시 무한루프를 탈출한다.

```

while (1)
{
    rewind(stdin);
    int idx = 0;
    char c = 0;
    Sleep(500);
    printf(">>");

    while ((c = getchar()) != '\n') msg[idx++] = c;
    msg[idx] = 0;
    if (idx == 0) continue;

    send(sock, msg, sizeof(msg), 0);

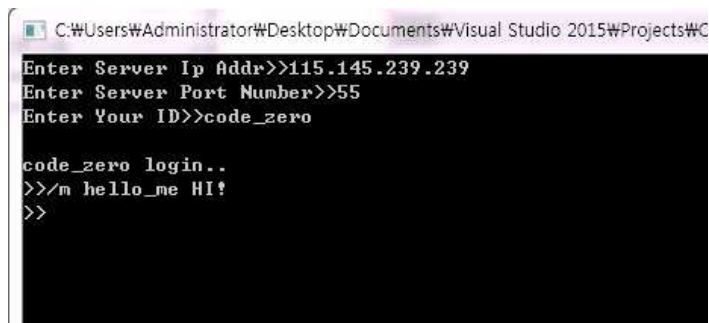
    if (!strcmp(msg, "/q"))
    {
        break;
    }
}
}

```

## 4. Functionality Description

### 4.1 Communicate each other

‘/m’ 명령을 통해 클라이언트 간 대화가 가능하다. 대화는 클라이언트 간 직접하는 것이 아니라 서버를 거친다.

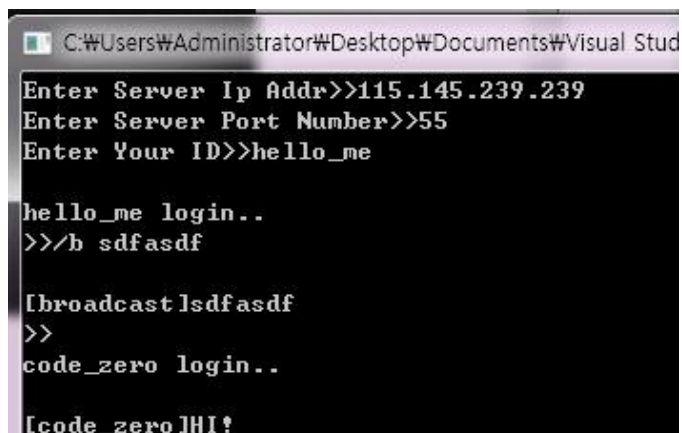


```

C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Projects\WC
Enter Server Ip Addr>>115.145.239.239
Enter Server Port Number>>55
Enter Your ID>>code_zero

code_zero login..
>>/m hello_me HI!
>>

```



```

C:\Users\Administrator\Desktop\Documents\Visual Stud
Enter Server Ip Addr>>115.145.239.239
Enter Server Port Number>>55
Enter Your ID>>hello_me

hello_me login..
>>/b sdfasdf

[broadcast]sdfasdf
>>
code_zero login..

[code_zero]HI!

```

## 4.2 /m command

/m 명령어를 통해 서로 대화가 가능하다.

```
C:\Users\Administrator\Desktop\Documents\Visual Stud
Enter Server Ip Addr>>115.145.239.239
Enter Server Port Number>>55
Enter Your ID>>hello_me

hello_me login..
>>/b sdfasdf

[broadcast]sdfasdf
>>
code_zero login..

[code_zero]HI!
```

만약에 해당 사용자가 네트워크에 없을 때는 다음과 같은 메시지를 출력한다.

```
/m sdfaera anybody?

sdfaera dose not exist.
>>
```

작동방식은 서버에서 '/m' 명령어 수신 시 dest\_id(메시지를 수신하는 유저의 ID)에 해당하는 유저 번호(idx)를 check\_userid() 함수를 통해 구한다. 그 후 해당 idx의 소켓을 통해 메시지를 전송한다. 만약 해당하는 클라이언트가 존재하지 않을 경우 메시지를 전송한 클라이언트에게 이를 알리는 메시지를 재전송한다.

```
else if (!strcmp(comm, "/m"))
{
    int len_tmsg = 0;
    char* dest_id = strtok(NULL, " ");
    char tmsg[BUF_SIZE], *temp;
    sprintf(tmsg, "[%s]", user_id);
    while (temp = strtok(NULL, " ")) strcat(tmsg, temp);
    while (tmsg[len_tmsg++] != 0);
    int dest_idx = check_userid(dest_id);
    if (dest_idx != -1)
        send(clint_soc[dest_idx], tmsg, len_tmsg, 0);
    else
    {
        msg_len = sprintf(msg, "%s dose not exist.", dest_id);
        send(clint_soc[cur_idx], msg, msg_len+1, 0);
    }
}
```

아래는 해당 함수이다.

```
int check_userid(char* user_id)
{
    for (int i = 0; i < user_num; i++)
        if ((user_flag[i] == TRUE) && !strcmp(user_list[i], user_id))
            return i;
    return -1;
}
```

#### 4.3 Notify connection of new client

아래의 클라이언트 id는 'hello\_me'이고 마지막 메시지는 'code\_zero'의 접속 메시지이다.

```
C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Projects\C_PRGRAMM
Enter Server Ip Addr>>115.145.239.239
Enter Server Port Number>>55
Enter Your ID>>hello_me

hello_me login..
>>/b sdfasdf

[broadcast]sdfasdf
>>
code_zero login..
```

서버에서는 클라이언트 접속시 user\_id를 수신하여 전체 네트워크에 브로드캐스트한다. 브로드캐스트에 대한 기능 설명은 4.5절에서 설명한다.

```
// 클라이언트로부터 USER_ID를 받음
msg_len = recv(clint_soc[cur_idx], user_id, sizeof(user_id) - 1, 0);
user_flag[cur_idx] = TRUE;
strcpy(user_list[cur_idx], user_id);

// 클라이언트의 접속을 알림

msg_len = sprintf(msg, "%s login..", user_id);
broadcast(msg, msg_len + 1);
```

#### 4.4 /who command

'/who' 명령어 입력 시 네트워크에 접속 중인 클라이언트의 리스트를 볼 수



있다.

```
>>/who

User List ::
hello_me
code_zero
```

서버는 '/who' 명령어 수신 시 make\_userlist() 함수를 호출한다.

```
if (!strcmp(comm, "/who"))
{
    make_userlist(cur_idx);
}
```

Make\_userlist()함수에서는 user\_flag를 검색해 TRUE일 시 list에 포함시킨다. 그 후 메시지를 전송한다.

```
void make_userlist(int user_idx)
{
    char list[BUF_SIZE] = "\nUser List ::";

    for (int i = 0; i < user_num; i++)
    {
        if (user_flag[i] == TRUE)
        {
            strcat(list, "\n ");
            strcat(list, user_list[i]);
        }
    }

    send(clint_soc[user_idx], list, sizeof(list), 0);
}
```

#### 4.5 /b command

아래의 그림은 브로드캐스트 실행시 모습이다. 모든 클라이언트가 동일한 메시지를 받은 것을 알 수 있다.

```
C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Projects\WC_PROGRAMMEN_1>
hello_me login..
>>/b sdfasdf
[broadcast]sdfasdf
>>
code_zero login..
[code_zero]HI!
/n sdfaera anybody?
sdfaera duse not exist.
>>
nser_me login..
[broadcast]HAHAHAHA!!!!
```

```
C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Projects\WC_PROGRAMMEN_1>
Enter Server Ip Addr>>115.145.239.239
Enter Server Port Number>>55
Enter Your ID>>nser_me
nser_me login..
>>/b HAHAHAHA!!!!
[broadcast]HAHAHAHA!!!!
>>
```

```
C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Projects\WC_PROGRAMMEN_1>
Enter Your ID>>code_zero
code_zero login..
>>/n hello_me HI!
>>/who
User List ::
hello_me
code_zero
>>
nser_me login..
[broadcast]HAHAHAHA!!!!
```

서버는 다음과 같이 브로드캐스트 메시지를 처리한다. Broadcast() 함수에서는 user\_flag == TRUE인 클라이언트에게 메시지를 전송한다.

```
else if (!strcmp(comm, "/b"))
{
    int len_tmsg = 0;
    char tmsg[BUF_SIZE] = "[broadcast]", *temp;
    while (temp = strtok(NULL, " ")) strcat(tmsg, temp);
    while (tmsg[len_tmsg++] != 0);
    broadcast(tmsg, len_tmsg);
}
```

```
void broadcast(char* msg, int msg_len)
{
    for (int i = 0; i < user_num; i++)
        if (user_flag[i] == TRUE)
            send(clint_soc[i], msg, msg_len, 0);
}
```

## 4.5 /q command

특정 클라이언트가 종료할 경우 다른 모든 클라이언트에게 이를 알린다.

```
C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Projects\C_PRGRA...
code_zero login..
>>/m hello_me HI!
>>/who

User List ::
hello_me
code_zero

>>
nser_me login..

[broadcast]HAHAHAHA!!!!

hello_me logout..
```

```
C:\Users\Administrator\Desktop\Documents\Visual Studio 2015\Proje...
Enter Server Ip Addr>>115.145.239.239
Enter Server Port Number>>55
Enter Your ID>>nser_me

nser_me login..
>>/b HAAAAAAAAA!!!!

[broadcast]HAHAHAHA!!!!

>>/
hello_me logout..
```

서버에서는 이를 broadcast()함수를 통해 처리한다.

```
else if (!strcmp(comm, "/q"))
{
    char buf[BUF_SIZE];
    msg_len = sprintf(buf, "%s logout..", user_id);
    broadcast(buf, msg_len + 1);
    break;
}
```