



세종대학교

Report

네트워크 프로그래밍 숙제 3

성명 : 박상우

학번 : 16013093

클라이언트는 VS2017에서 MFC를 사용해 구현했습니다.

서버의 경우 select방식을 사용해 멀티 채팅을 구현했습니다.

먼저 통신을 위해 정의한 매크로와 구조체들입니다.

```
// 메시지 TYPE
#define FIRSTROOM      11
#define SECONDROOM    12
#define SHOWUSERS     13
#define NAMECHECK     14

// 소켓 정보 저장을 위한 구조체와 변수
struct MSGDATA
{
    int         type;
    int         ID;
    char        name[2][NAMESIZE];
    char        buf[BUFSIZE];
}chatMsg;

typedef struct SOCKETINFO
{
    SOCKET      sock;
    int         recvbytes;
    int         ID;
    char        name[2][NAMESIZE];
}SOCKETINFO;
```

데이터는 MSGDATA 구조체의 형식으로 통신하며 메시지의 종류, 클라이언트의 고유 ID, 각 방의 이름, 메시지 문자열이 포함된 구조입니다.

소켓정보를 저장하는 구조체는 소켓, 받은 데이터의 량, 고유 ID, 각 방의 이름으로 구성했습니다.

메시지 타입에는 이 메시지가 어느 방에서 온 것인지(FIRSTROOM, SECONDROOM), 사용자 리스트를 보여 달라는 요청(SHOWUSER), 닉네임 확인, 변경 시 중복된 닉네임을 체크하기 위한(NAMECHECK)가 있습니다.

서버에서 소켓 접속 시 AddInfo()함수를 호출해 새로 접속한 소켓을 SOCKETINFO 배열에 추가합니다.

클라이언트에서 접속 시 접속 정보를 서버로 송신하는데 이 정보를 이용해 SOCKETINFO 배열에 추가되는 ID, 각 방의 닉네임 정보를 초기화했습니다.

클라이언트에게 데이터 전송이 실패했다면 RemoveInfo()함수를 호출 해 소켓 정보를 삭제했습니다.

메시지를 수신하면 switch - case 문을 사용해 메시지 타입에 따라 처리했습니다.

메시지 타입이 FIRSTROOM, SECONDRoom이라면 반복문을 돌며 접속한 모든 클라이언트에게 수신한 메시지를 다시 전송합니다.

메시지 타입이 SHOWUSER이라면 현재 접속한 사용자의 수를 전송해 총 닉네임이 몇 개 전송되는지 클라이언트에게 알려주고 반복문을 돌며 접속한 닉네임을 전송합니다. 클라이언트에서는 처음 전송된 사용자의 수를 통해 총 몇 개의 닉네임을 수신해야 하는지 알 수 있으며 문자열 함수를 통해 전송된 닉네임을 하나로 합쳐 각 방에 출력합니다.

메시지 타입이 NAMECHECK이라면 NAMECHECK요청을 보낸 클라이언트를 제외하고 다른 사용자 닉네임 중에 같은 닉네임이 있는지 확인합니다. 같은 닉네임이 있다면 “0”을 전송해 클라이언트에서 새로 메시지를 보낼 수 없게 버튼을 비활성화 하게 합니다. 같은 닉네임이 없어 닉네임이 변경 가능하다면 SOCKINFO 배열에서 닉네임을 바꾸고 클라이언트에게 “1”을 전송해 확인했습니다.

클라이언트는 IP, Port, ID(닉네임)를 입력하고 입장 버튼을 누르면 서버에 접속하며 잘못된 정보가 입력될 시 오류 메시지를 출력합니다. (그림1~3)

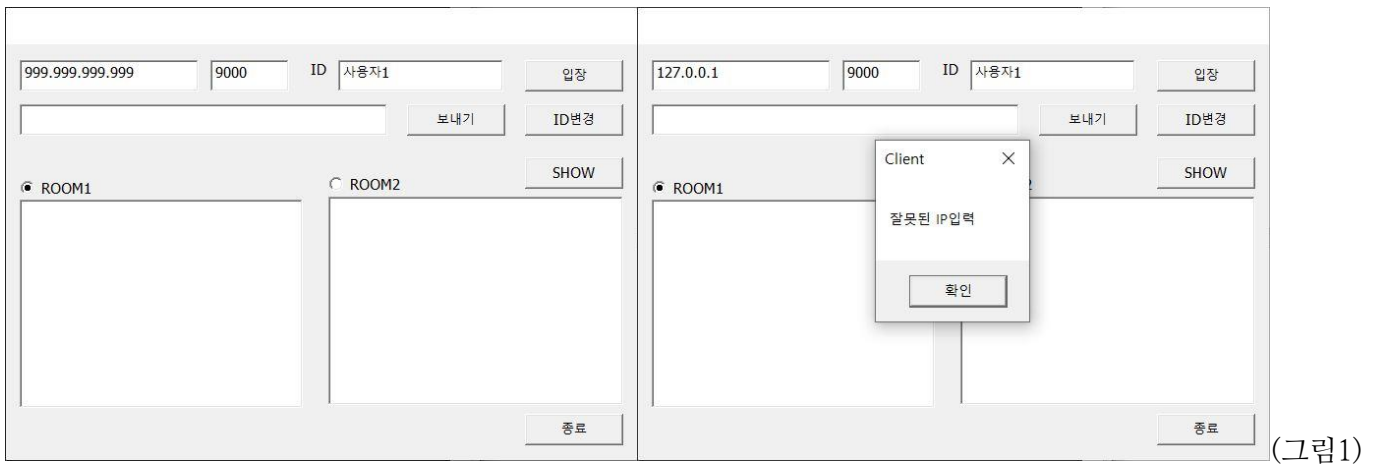
IP의 경우 각 숫자가 0~255 범위인지, ‘.’ 을 제외한 다른 문자가 있는지, ‘.’이 3개 인지 등으로 정상적인 입력인지 확인했습니다. Port의 경우 숫자만 입력 받게 설정했고 0~65535 사이의 값이지만 확인했습니다.

서버 접속에 성공하더라도 기존에 중복된 닉네임이 있다면 경고 메시지를 출력하며 버튼이 비활성화 되 메시지를 보낼 수 없습니다. 또한 채팅 중 닉네임을 변경할 때도 같은 방식으로 체크합니다. (그림7, 8)

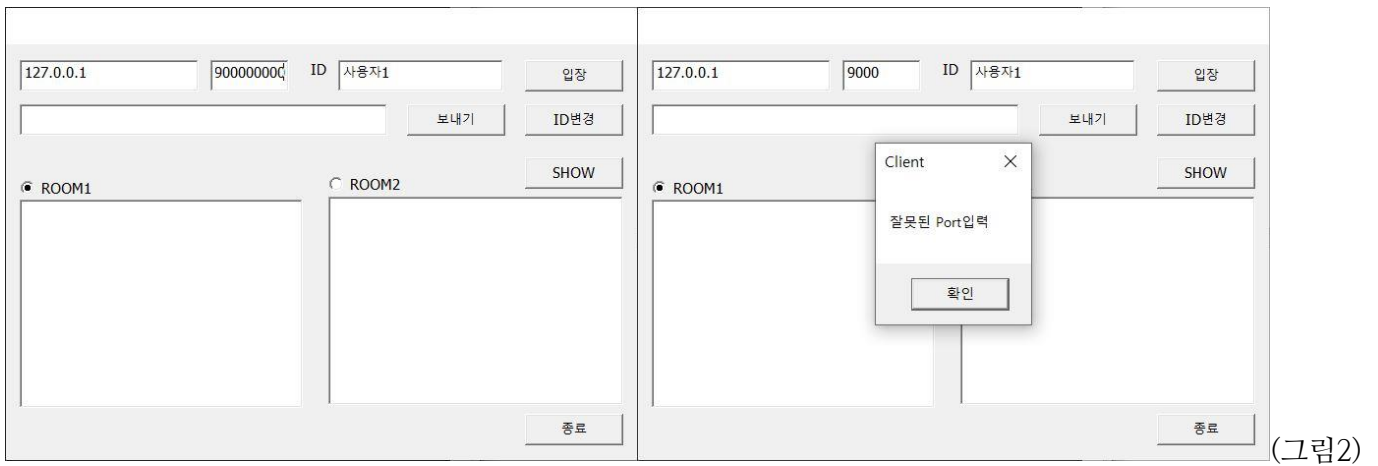
서버 접속 시 최초에는 ROOM1과 ROOM2 모두 처음 입력한 ID로 설정되지만 ID변경 버튼을 누르면 현재 체크되어 있는 방의 닉네임이 변경됩니다. 또한 방 버튼을 눌러 방을 변경할 때 각 방의 닉네임에 맞게 ID 문자열이 변경됩니다. (그림5)

클라이언트에서는 쓰레드를 사용해 메시지를 수신하며 서버에서 사용한 방식과 같이 switch - case 문을 사용해 수신한 메시지의 타입에 맞게 처리했습니다.

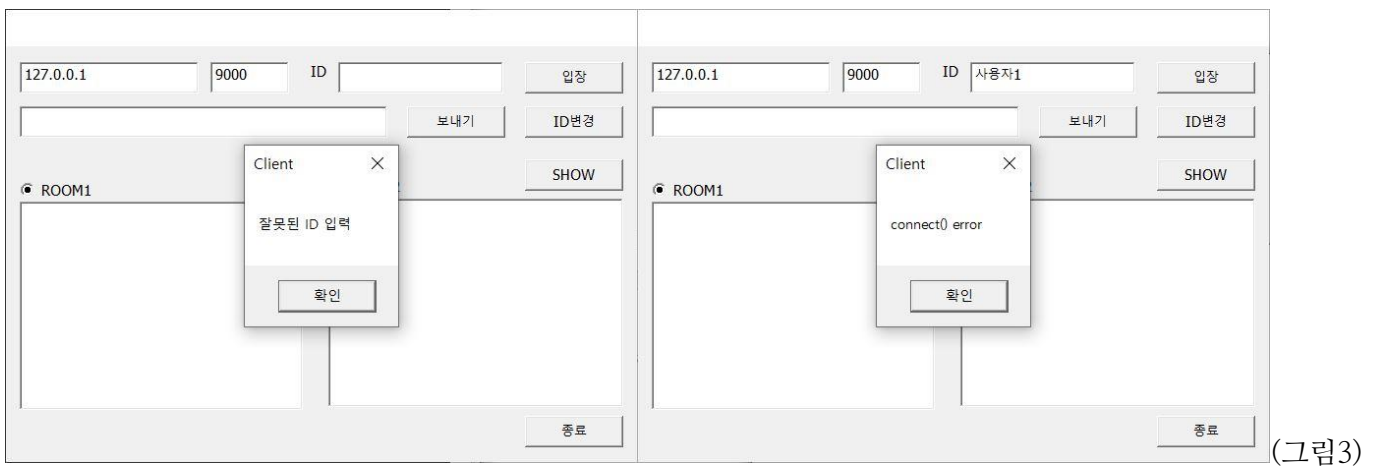
메시지 타입이 FIRSTROOM, SECONDRoom이라면 각 방에 맞는 ListBox에 수신한 메시지를 추가하며 SHOWUSER의 경우에도 각 방의 유저별로 출력합니다. (그림4, 6)



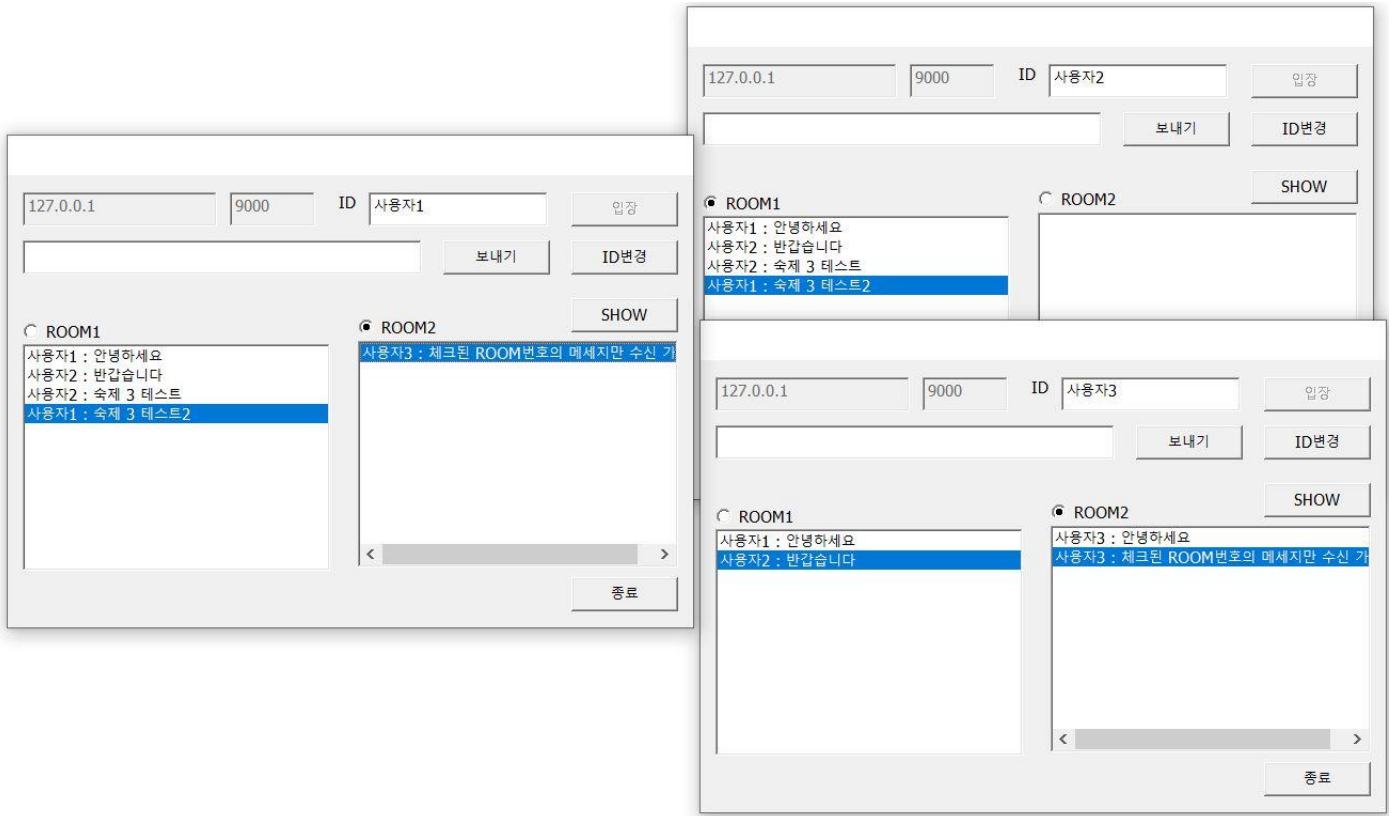
잘못된 ID입력 시 오류입니다.



잘못된 포트 번호 입력 시 오류입니다.



ID를 입력하지 않았을 때와 서버에 연결되지 못했을 때 오류 메시지입니다.



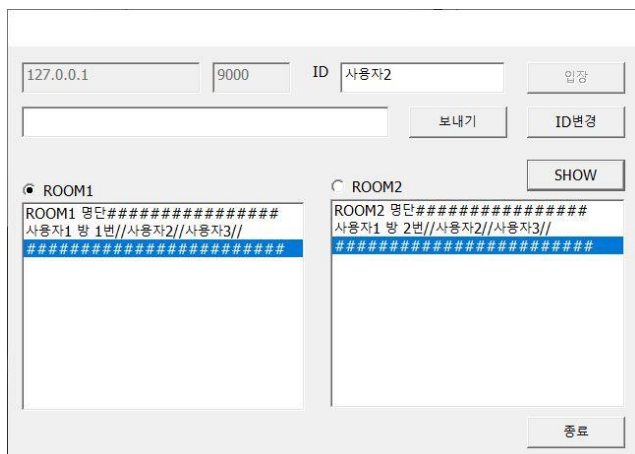
(그림4)

대화 테스트입니다. 체크된 방 번호의 메시지만 화면의 출력됩니다.



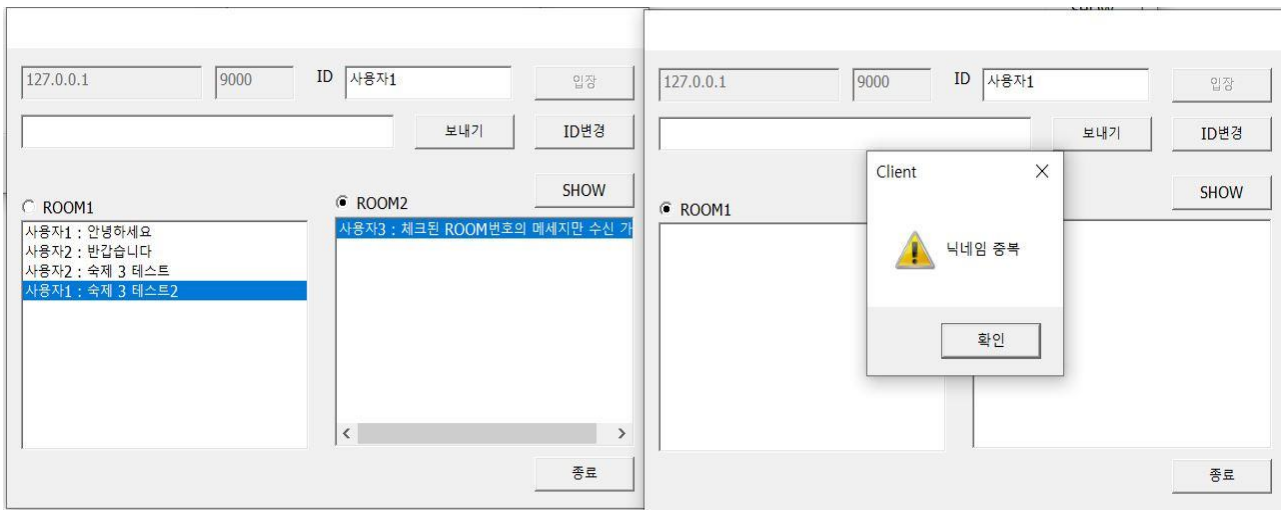
(그림5)

최초 입장 시에는 방1과 방2의 아이디가 같게 설정되지만 ID변경 버튼을 누르면 현재 체크된 방의 ID를 따로 설정할 수 있습니다.



SHOW버튼을 누르면 각 방의 접속한 사용자 명단이 출력됩니다.

그림(6)



(그림7)



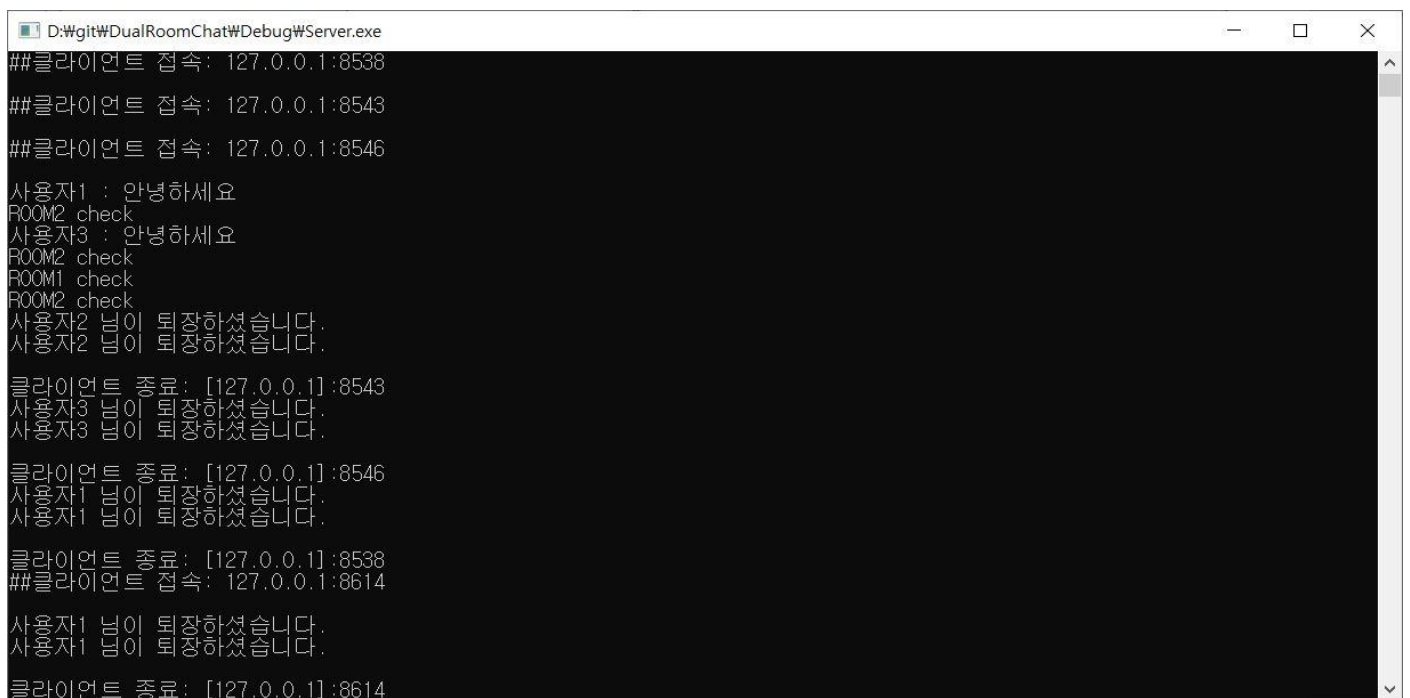
입장 시 이미 존재하는 닉네임으로 입장한다면 ID변경과 종료 버튼을 제외한 나머지 버튼이 비활성화 되며 메시지를 보낼 수 없습니다.

ID를 변경해야 새로 메시지를 보낼 수 있습니다.

또한 ID변경 시에도 중복되는 닉네임이라면 버튼이 활성화되지 않습니다.

(그림8)

방 변경 시 기존 방에서 나가고 새로운 방으로 입장하는 것이기 때문에 원래 있던 방으로 돌아가도 다시 닉네임을 체크하며 다른 방에 있던 중 기존 방에서 같은 닉네임으로 변경한 사용자가 있다면 새로 닉네임을 변경해야 합니다.



서버의 모습입니다.