Tcp fighter를 스크롤이 되는 필드 처리 한 서버를 만들 것. Select 모델로 검증된 모델을 만들어야한다. 네트워크 파트에 대한 오류는 없는 상태로 해야한다. 네트워크 코드를 검증하고, 복붙하고 다음 서버에서 또 쓸 거다. 거기서 stress tool만들어서 검증하고. 또 게임서버 컨텐츠를 갖다 붙이는 형식으로 진행.

만들 건 기본적인 채팅. 서버 ip입력하고 닉네임 입력하고.

근데 그러려면 윈도우가 필요하다. 콘솔창에서는 못한다. 입력하는중에는 블락걸려서 네트워크 코드가 돌질 못한다. 멀티쓰레드로 해결을 보던지, 윈도우기반의 비동기로 돌아갈 수 있게 async select를 써서. 그리고 키보드 입력은 aditer 를 통해서.

우리는 프로토콜을 맞춰서 서버를 만들어야 한다.

처음 접속해서 닉네임 확인하고 채팅서버에 들어가면 방 목록이 나옴. 이미 만들어져있다면 방 목록이 떠야하고, 누군가는 방을 생성해야하고, 방에 입장해야하고, 채팅할 수 있어야 한다.

기획이 나왔으면 프로토콜 설계.

Packetcode 는 1바이트 0x89 데이터가 깨졌나 안깨졌나를 걱정해야될까? 이미 밑에단계에서 l2 l3 l4에서 송수신받은 데이터를 안전하게 데이터를 보내주는게 걔네 할 일. 우린 그런거 판단하는게 아니라, 누군가 악의적으로 위변조를 해줬나 안해줬나 판단하는거. 또는 아예 쓰레기 데이터. 0x89는 완전 쓰레기 데이터를 체크하고, 체크썸은 위변조를 체크한다. 근데 내 클라가 아닌 다른 클라가 쓰레기를 보내는 확률이 얼마나 될까. 설령 진짜 그렇게 보내더라도 체크썸에서 걸러진다. 여기서 진짜 맞아떨어져서 컨텐츠까지 들어가더라도 거기서 걸러진다.

보통 클라에서 서버로 보내는걸 req라고 하고, 결과를 response로 받을거다. 이게 꼭 요청된 결과는 아니다. 일에 대한 결과만 받는거면 짝이 맞는데, 웹이 그렇다. 근데 stateful 서버라면 서버가 먼저 보내는 경우가 있다. 내가 아무것도 안해도 다른 클라가 채팅을 쓰면 데이터가 온다. 그럼 완전히 짝이 맞는게 아니라, 방향을 결정하는 용도에 그친다.

1. Req: wchar[15] 닉네임(유니코드)
   1. 클라가 서버에 연결하면 로그인. 그럼 클라는 서버에 id를 줘야한다.
   2. 서버에 문자열을 보내는 방법은 2개. 이렇게 유니코드를 보내는 방법. 이건 고정. 가변으로 보내는 채팅은 길이를 먼저 보내고 그만큼 보낸다. 여기서 널문자를 넣어줄 필요가 없다. 널문자는 길이를 모를때만. 고정길이면 널문자가 들어가야한다.
2. Res : 결과(1:ok / 2:중복닉넹미 / 3:사용자 초과 / 4:오류) , 사용자번호
3. Req : 대화방 리스트 요청 (페이로드 사이즈 0 그냥 헤더만 가면 된다.)
4. Res : 대화방 리스트 주기 (방 이름과 고유 id. 모든건 고유번호가 들어간다, 우린 직렬화 버퍼니까 메시지 하나로 끝날거다, 하나의 메시지 안에 몇 개의 방이 들어올 지 모른다. 그래서 방의 개수가 들어가야하고, 방 이름 방번호가 반복적으로 들어가야한다. 방 하나하나마다 길이와 이름을 보낸다. 가변길이로 방 이름을 만들거. )
   1. 문제는 데이터가 너무 크면 send의 링 버퍼에 다 넣지 못해지는 상황이 있을 수 있다. 1만바이트가 나오고, 링버퍼도 여유롭게 했다면. send했을 때 문제점은? Tcp는 애초에 mss 짤라보내니까 mss상황을 논의할건 udp뿐이다. send문제점 결국 없다. 알아서 잘 짜르고 알아서 잘 갈거다.
   2. 방이 정말로 크다면, 한번 갈때마다 페이지 단위로 20개씩 끊어서 간다거나. 클라랑 논의가 되어야 한다. 까보니까 20개가 오고 또 20개가 오면 새로운건지, 추가된건지 구분이 없다. 이건 추가적으로 논의해야한다. 보낼때마다 페이지 번호를 준다던지, 잘 설계해야한다.
   3. 아까 페이로드 2바이트니까 65535. 이것까지는 하나의 패킷 덩어리로 보낼 수 있따. 근데 이게 좋은상황은 아니니까 고민하는거다. 클라 직렬화 버퍼가 512여서 그 이상 보내면 날려버린다.
5. Req : 대화방 생성 (방 이름 사이즈, 방 이름)
   1. 대화방이 생성이 안될 경우는 언제일까? 방 제목이 너무 길거나, 방 개수가 너무 많거나. 방 개수를 제한없이 걸면 안된다.
6. Res : 대화방 생성 결과(수시로 올 수 있는 애) :전체 유저 대상으로 broad casting (성공했을때만 전체에 보내기, 실패했을 땐 유니캐스트.)
   1. 목적 1 . 생성을 요청한 사람한테
   2. 목적 2. 다른 유저들한테 이런 방이 만들어 졌습니다.
7. Req : 방 입장 (방 고유번호) : 로그인을 했어야하고, 로비에 있어야 한다.
8. Res : 방 입장 결과
   1. 방 입장을 요청할 수 있는 앤가 확인. 로그인 한 애가 요청해야함. 원래도 로그인을 했나 안했나 확인하고. 채팅 메시지 보내는것도 로그인을 해야하고 방에 들어가있어야 했다.
9. Req : 채팅 송신 (채팅 사이즈, 채팅내용) : 만약 채팅방 안에 없는데 보내면 접속을 끊어버림.
10. Res : 채팅 수신
11. Req : 방 퇴장 (소켓이 disconnect가 된 것도 퇴장 시켜야 함. 정리 로직에 의해서)
12. Res : 방 퇴장(수시) 사용자 번호
13. Req : 방 삭제 (수시) 방 번호
14. Res : 사용자 채팅방 입장 (사용자 번호, 닉네임)

헤더 프라그마 팩으로 감싸기.

모든 유저를 대상으로 한 프레임send 한 프레임에 recv를 몰아서 해야한다. 보내고 싶을 떄 send하는게 아니라 writeset이 있을때만 send 몰아서하는것.