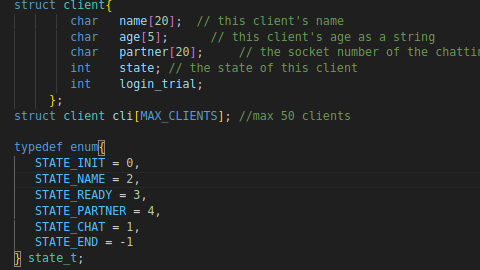
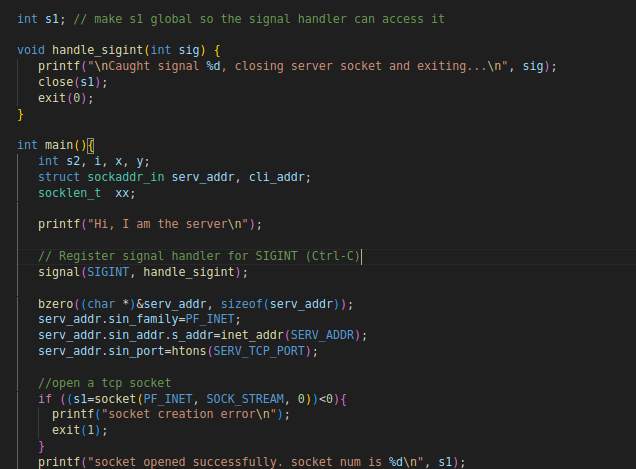
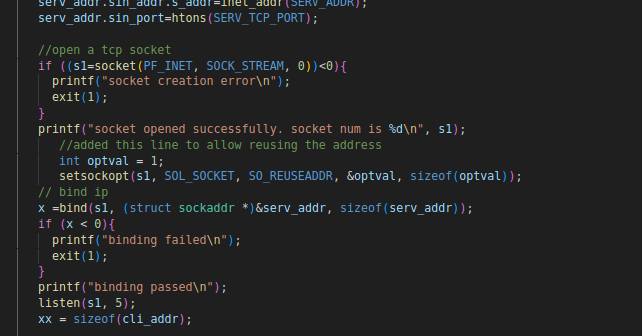
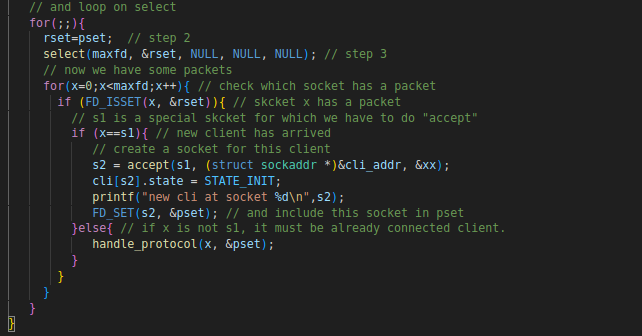
7) Implement a chatting server. The state of the client during the protocol is as follows. At any moment multiple pair of clients should be able to talk at the same time.

서버의 코드부터 확인하자.

  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
코드 작성의 편의를 위해 state를 enum으로 정의해 의미를 가진 단어로 상태를 확인할 수 있게 했다. state를 client 구조체 내부에 저장했다.

  
  
Ctrl-c 시그널을 핸들링 해 프로그램을 안전하게 종료시키기 위해 SIGINT handler 함수를 만들었다. 이를 만들어 포트 자체를 직접 닫아주면 프로그램 종료 후 새로 같은 ip, 같은 port를 여는데 대기시간이 없어질 것이라 생각했지만 관련은 없었다.  
  
인터넷 검색을 통해 setsockopt 함수와 코드에 보이는 것과 같은 파라메터를 넣어주면 소켓 재오픈 대기시간이 없어진다는 것을 알게 되었다.

  
  
소켓을 열고 닫고, 클라이언트에서 오는 정보들을 처리하는 코드는 기존과 크게 다르지 않다.

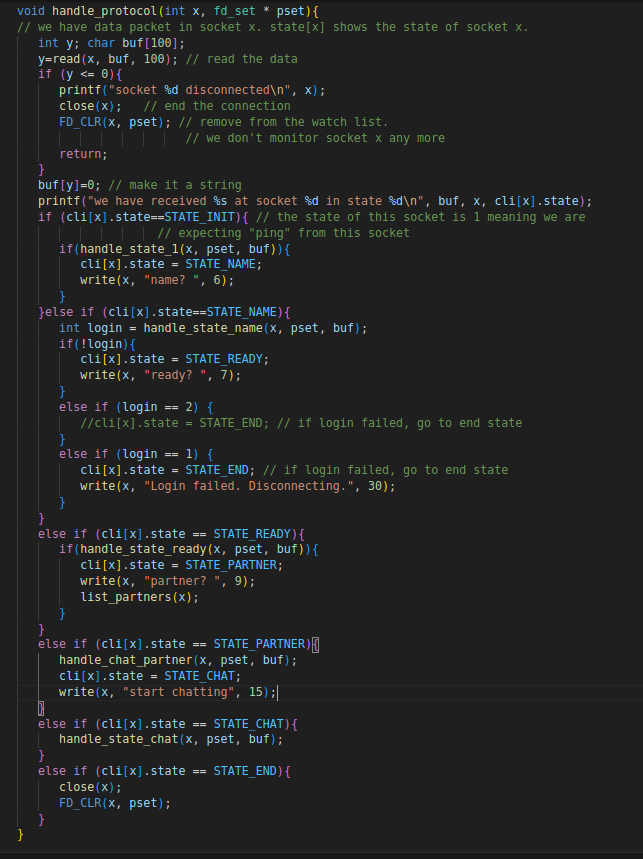
먼저 socket함수를 통해 tcp 채팅 서버를 위한 소켓을 열고, 열리지 않았을 경우 에러를 처리하고 프로그램을 종료했다. 그리고 열린 소켓을 bind함수를 이용해 ip에 바인드해주었다. 이 반환된 x값은 하나의 file discriptor로 작동을 하게 된다. 이 것을 통해 client의 요청이 들어온다.

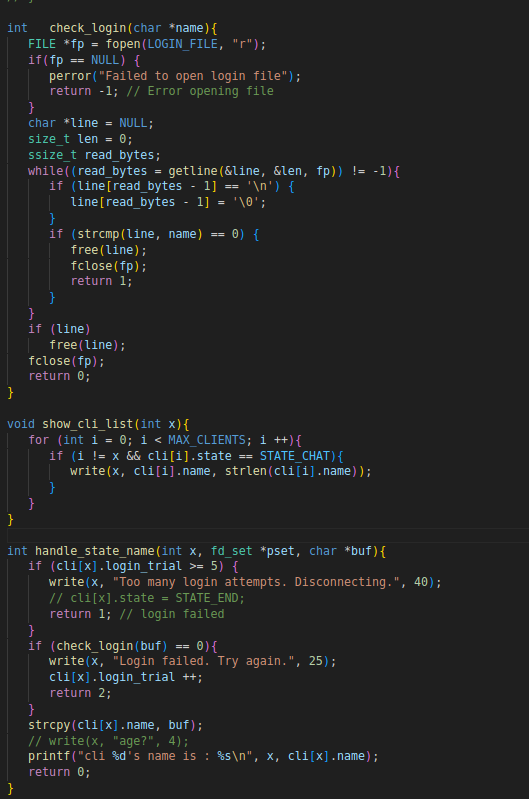
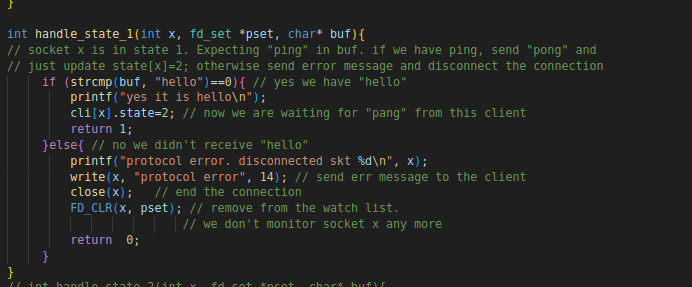
서버 포트를 열고 바인딩 한 이후에 client의 요청을 무기한 기다린다. Select 함수를 이용해 fd\_set으로 1로 바꾸어준 index에 해당하는 소켓에서만 패킷을 기다린다. 초기에는 서버의 소켓인 s1에 대해서만 대기를 하는 셈이다. 이렇게 s1으로 패킷이 들어왔다는 것은 새로운 클라이언트가 접속을 시도한다는 의미이다. 그 경우에는 accept함수를 통해 s2에 해당 클라이언트를 위한 소켓을 새로 만들어 할당한다. pset에 이를 다시 set해준다.(rset은 select하면서 매번 패킷이 들어온 소켓만 남기므로 누적되어 오픈된 소켓을 확인할 수 없다) 그 인덱스에 해당되는 cli 정보들을 담는 구조체 배열 원소의 상태를 STATE\_INIT으로 초기화해준다.

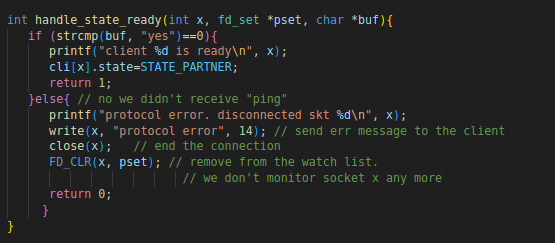
S1으로 패킷이 들어오지 않았을 경우에는 기존에 이미 연결된 클라이언트이므로 프로토콜을 진행한다.

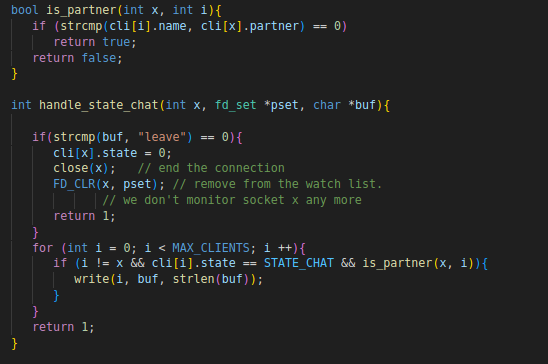
프로토콜 핸들링의 구조는 아래와 같다.

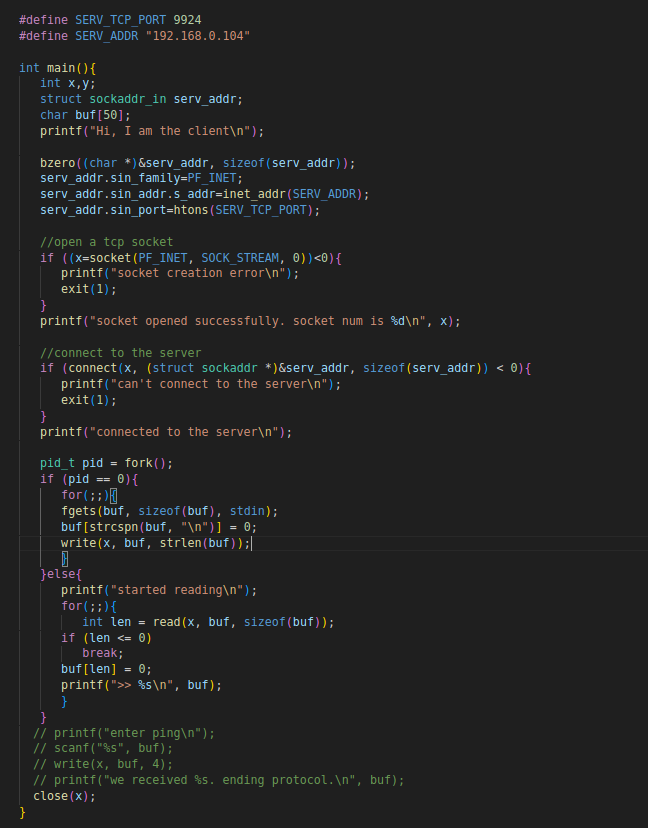
각 state를 확인하고 가독성을 위해 스테잍트에 대한 핸들링 함수를 다시 만들어 연결했다. 핸들링 함수 종료 이후 다음 state에 대한 출력물까지 한번에 write를 해 주었는데, 이는 다음 state에서 넘어가고 handling 함수가 호출되는 시점은 포트로 패킷이 들어오는 시점인데, 그렇게 된다면 입력을 받은 이후에 필요한 정보를 주게 된다.



그렇게 해당 상태에 대한 함수들을 수행한 후에 그 입력받은 포트 번호에 해당하는 cli배열의 state를 업데이트 시켜준다.   
  
  
Init 스테이트에 대한 행동 함수이다. Hello를 입력해야만 계속 진행되고 아닐 경우에는 강제종료

login에 대한 함수들이다. Check login 함수는 입력 이름을 받아와 그 이름이 파일에 한 줄로 존재하는 문자열인지 확인한다.getline함수를 사용하기 위해서 로그인파일을 FILE 포인터로 선언했고, readline함수를 이용해 한 줄씩 읽어나갔다. 이 읽은 결과물이 입력받은 이름과 같으면 true이고 이것을 로그인파일이 끝날 때 까지 반복한다. 그럼에도 불구하고 존재하지 않을 경우에는 false를 반환한다. 그렇게 handle\_state\_name함수에서 입력받은 이름을 login이 되는 지 확인을 한 후에 login이 안 될 경우에 해당 클라이언트의 login trial을 1 증가시킨다. 이것이 5번 이상이 된다면 서버에서 더이상 진행시키지 않는다. 올바른 이름이 들어온 경우에는 문제 없이 0을 반환하고 다음 스탭으로 진행된다.  
  
  
  
State ready에 해당하는 함수는 그냥 yes를 입력받았는지를 확인하고 받은경우 0을 반환, 틀린 경우1을 반환시켜 호출 된 곳에서 에러처리를 한다.  
  
List partners함수는 해당 연결된 클라이언트 제외, state가 chat인 client를 모두 확인(전역변수 선언되어있음), 이에 대한 이름들을 출력한다. 이 행동은 ready에 대한 프로토콜 확인이 참인 경우에 호출된다.  
  
  
그리고 partner스테이트에서 들어온 입력을 해당 cli의 partner에 저장한다.

채팅을 할 때에는 client로부터 입력받은 내용을 x의 chat partner에 해당하는 client의 소켓에만 write를 한다.

  
클라이언트 코드는 비교적 간단하다. 서버와 동일한 포트를 열고 해당 포트를 이용해서버에 connect 함수를 이용해 연결을 시도한다. x가 서버와 연결된 소켓이다. 이를 이용해 코드를 fork를 이용해 client로서 stdin을 사용자로부터 받아들이는 부분과 서버로부터 받은 패킷을 stdout으로 출력하는 두가지의 부분으로 분리했다.

