Operating System Term Project

Multithreaded Chat Server

2013171045 남정호

1. 구현 방법

* Server  
  a. Socket 생성: socket 함수를 이용하여 listenfd 변수에 반환 값을 저장하고 -1이 반환될 경우, 에러 메시지가 나오도록 구현했습니다.

b. Port data 전달: scanf 함수를 통해 int로 입력을 받고 atoi로 1차 변환, htons로 2차 변환 시켜 sin\_port에 입력하였습니다.

c. Server Address 설정: 참조 자료를 따라 해결했습니다.

d. Bind, Listen, Accept 설정: c와 동일합니다.

e. Client들의 번호 배정: 번호는 0부터 9까지 총 10개가 있으며, 인덱스를 설정하여 0부터 순서대로 배정받도록 하였습니다. 또한, 인덱스가 10에 도달하면, 0~9 사이에서 연결이 끊어진 번호를 찾아 해당 번호로 배정 받도록 만들었습니다.

f. Connection Notice: 연결이 되었을 경우, 우선 printf 함수를 통해 server 창에 해당 client가 연결 되었음을 알립니다. 그 다음, for문과 if 문을 이용하여, 현재 연결 중인 다른 Client들에게 현 Client가 연결 되었음을 알립니다.

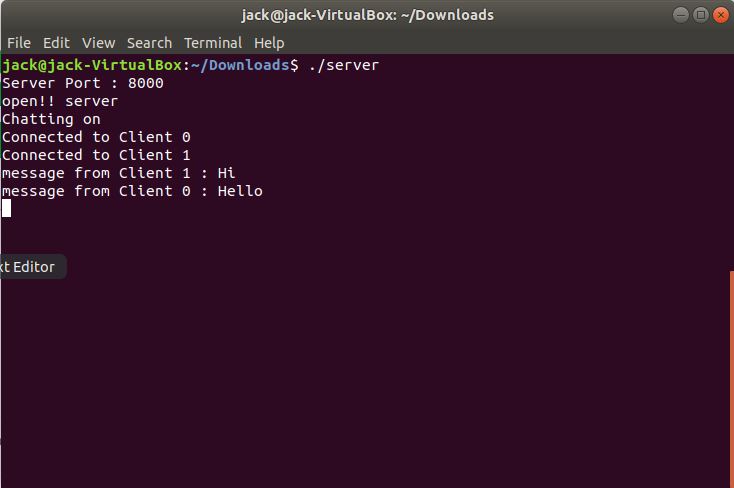
g. thread 함수 구현: pthread\_create에 들어갈 함수 server\_handler를 구상합니다. 우선 server\_handler에 전달될 값으로 hin이라는 struct를 만들었습니다. 이 struct에는 현재 client의 인덱스 값과 Accept 함수의 반환값이 들어 있습니다.  
whille구문 안에 recv 함수를 넣음으로써, client로부터 데이터를 받을 수 있게끔 합니다. 전달 받은 데이터가 “Q”인 경우, 해당 client와의 연결이 끊겼음을 표시합니다.  
strcat을 통해 client로부터 전달 받은 메시지와 “Client x :” 구문을 합칩니다. 이 후 for문과 if구문을 통해 연결 중인, 메시지를 전달한 client를 제외한 다른 client들에게 메시지를 전달시킵니다.

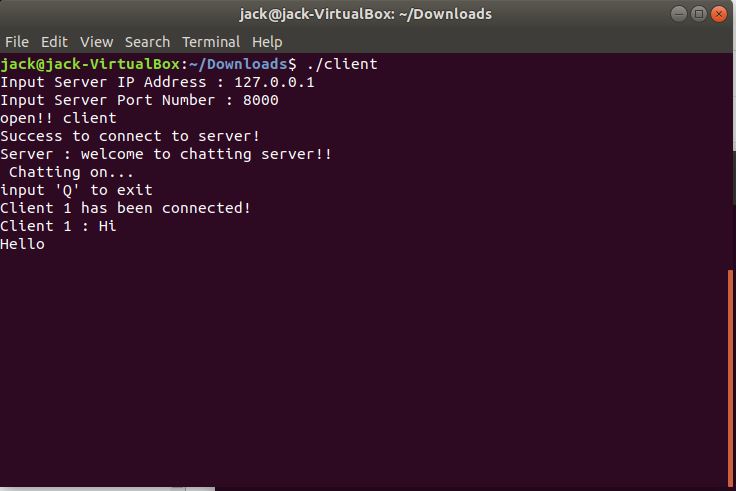
* Client  
  a. Socket 생성부터 connect까지의 과정은 Server 와 크게 다르지 않기 때문에 IP 주소를 받는 부분만 서술하겠습니다.  
  scanf 함수를 통해 IP 주소를 입력 받습니다. 만약 입력 값이 127.0.0.1이 아닌 경우 에러 창이 나타나도록 설정했습니다. 이후 inet\_addr()을 이용하여 IP값을 변환 시켜 sin\_addr.s\_addr에 들어가도록 만들었습니다.

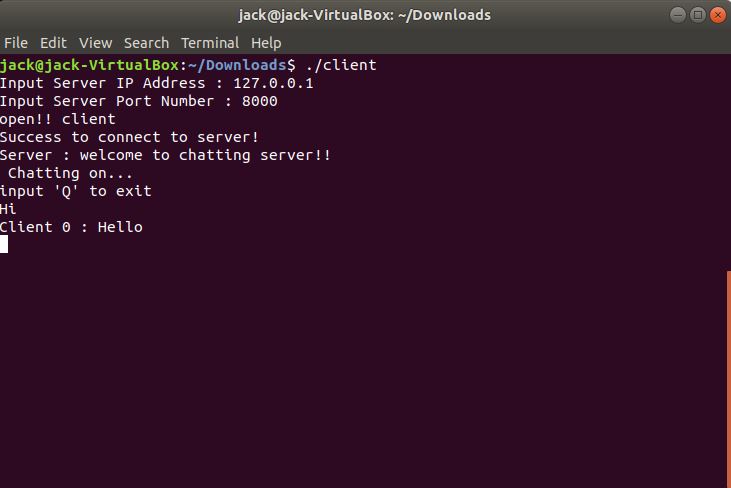
b. 데이터의 수신 : Server 에서는 데이터의 수신과 전송을 둘다 tread를 이용하여 구현한 반면, Client에서는 수신 부분만 thread로 구현했습니다.  
while(recv~) 구문을 통해 서버로부터 받은 데이터를 출력하도록 하였습니다. 데이터 출력이 끝나면 memset을 이용하여 버퍼를 비워줬습니다.

e. 데이터의 송신 : while(1){ ~ fgets ~ if(send~) } 구문을 통해 구현하였습니다.  
fgets를 통해 stdin으로부터 데이터를 입력 받고, send 함수를 통해 전송하였습니다. send 함수가 음의 값을 반환하는 경우, 에러 메시지를 띄우도록 만들었습니다.

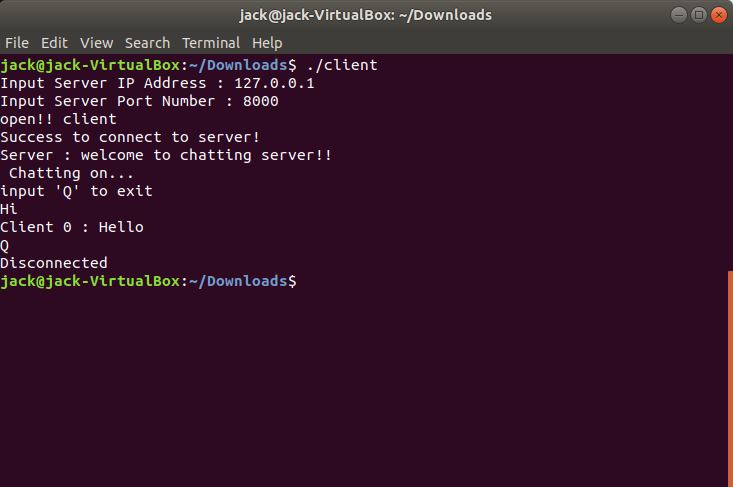
1. 실행 화면

**기본 실행 화면**

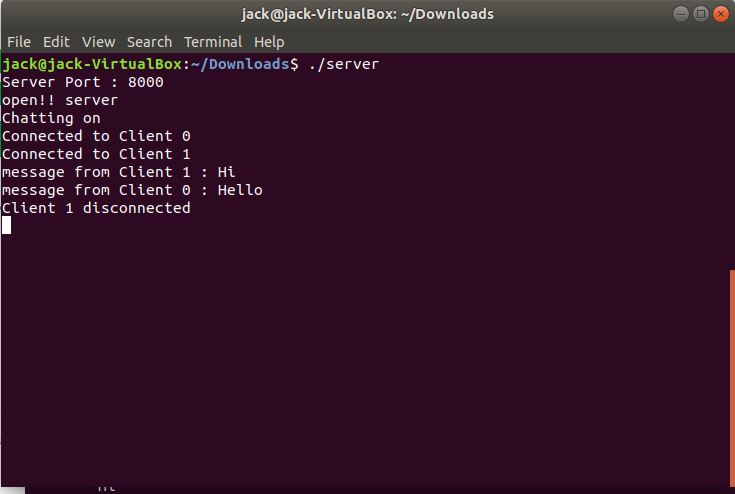
Server

Client 0

Client 1

**Q 입력시**

Client

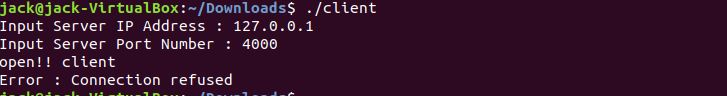


Server

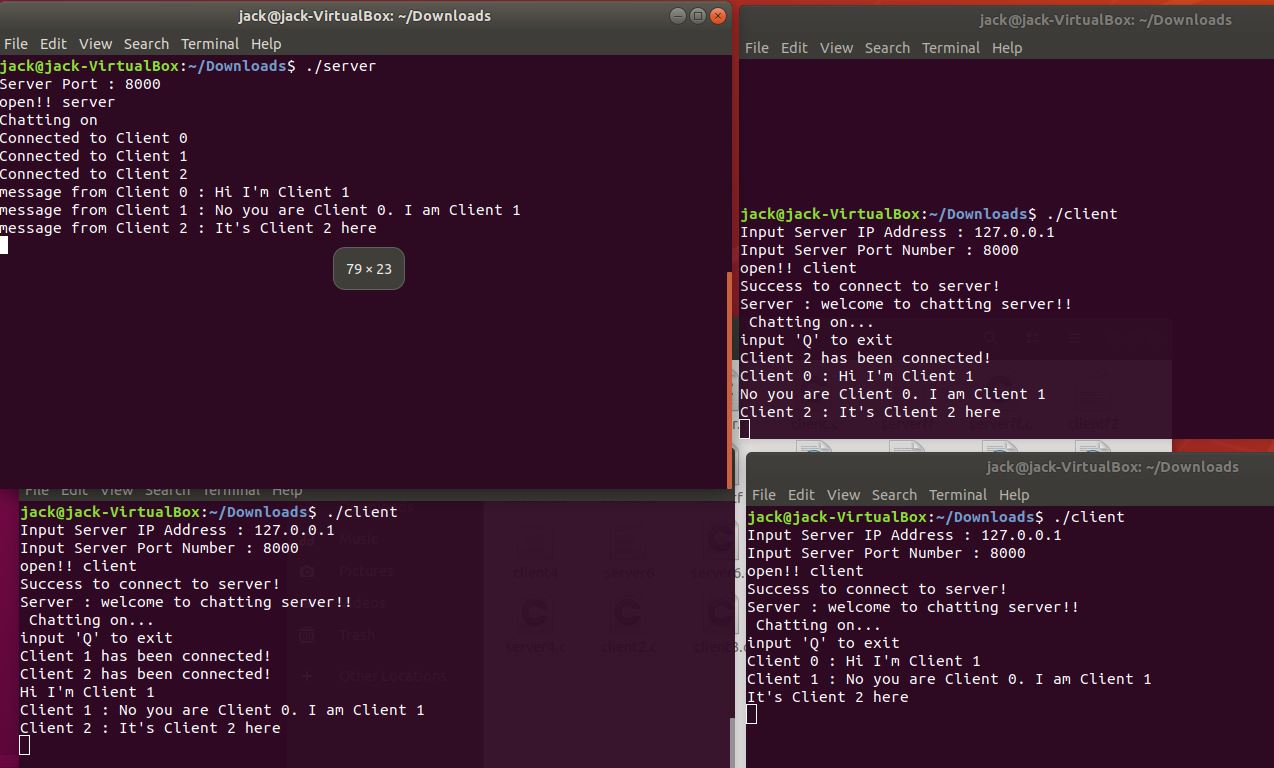
**Client 연결 시 IP ADD 혹은 Port를 잘못 입력한 경우**



ADD 잘못 입력(127.0.0.1을 제외하고는 다 잘못된 입력으로 처리했습니다.)



Port 번호 잘못 입력(해당 예시에서 Server의 Port Number은 8000이었습니다.)

**다중 대화(3인 대화)**

Server Client1

Client0 Client2 순서

1. 한계& 개선점  
   a. PPT에 나와 있던 pthread\_join 과 pthread\_detach 의 이용법을 파악하는데 실패하여 적용 하지 못하였습니다.  
   b. Clinet 의 경우 송&발신 모두 포함하는 thread 함수를 구축하지 못하고 수신의 기능만 있는 thread 함수를 구현했습니다.  
   c. 수용 인원이 10명으로 한정 되어있습니다.  
   d. 특정 Client의 연결이 해제되었을 때, 해당 정보를 Server에만 표출하고 다른 Client에게 전달하는 기능을 구현하지 않았습니다.
2. 참고 자료

<http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html>

<https://codereview.stackexchange.com/questions/154969/c-multithreaded-chat-client-and-server>

<https://www.youtube.com/watch?v=jCcenSa-8zw>

첫번째 자료에서 C를 이용한 linux 서버 구축에 대한 기본을  
두번째 자료에서 127.0.0.1을 입력 받아서 변환하는 방법과 thread 구상 방법을  
세번째 자료에서 제 프로젝트의 베이스 틀을 얻었습니다.