시스템 프로그래밍 보고서

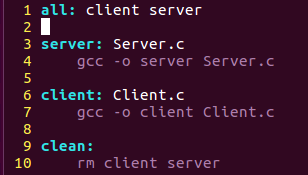
**Chat Program (TCP)**

2012018109

컴퓨터공학부 컴퓨터전공

권현우

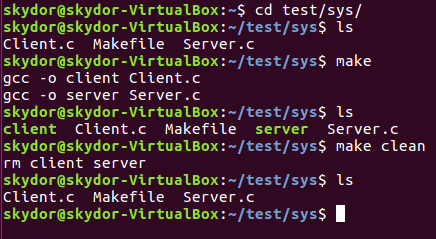
**1. Makefile 사용법**



make를 통해서 Server.c와 Client.c에 대한 컴파일 결과를 각각 server/client라는 이름으로 생성.

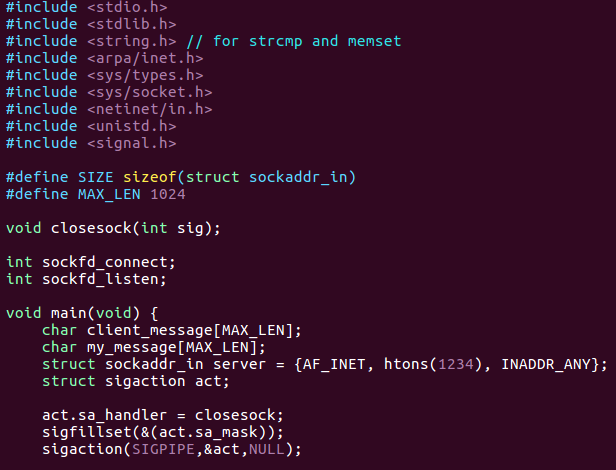
또한, make clean을 이용해서 생성한 client, server 파일을 삭제할 수 있다.

실행 예시)



**주의 : <정상적인 프로그램의 작동을 위해서는 Server가 Client보다 먼저 실행되어야 한다.>**

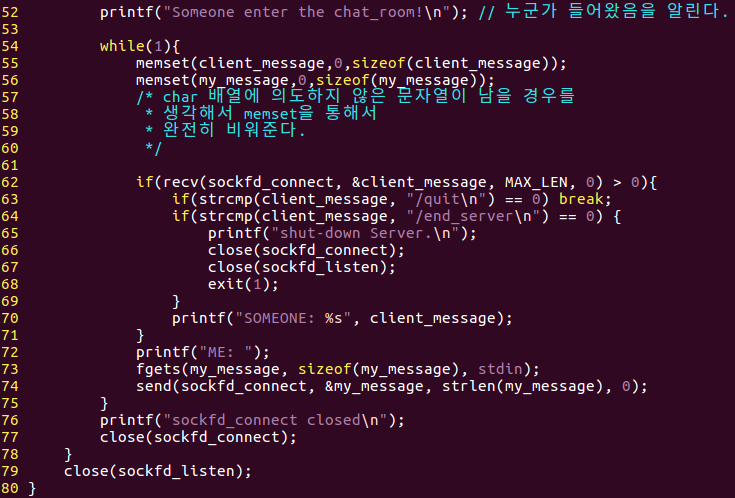
**2. “Server.c” Source-Code**

****

Client로부터 받은 메시지를 저장할 client\_message 와 server에서 client로 보낼 메시지를 저장할 my\_message가 있다. 해당 배열의 크기는 MAX\_LEN (1024)로 지정해두었다.



Server를 열기 위해서 socket -> bind -> listen의 과정을 거쳐 client로부터 connect를 accept할 준비를 마친다. 또한, while문을 돌면서 Client가 접속하는 것을 기다린다.



만약 client가 들어온다면 printf를 통해 누군가가 들어왔음을 알린다.

While문 내의 첫 2줄은 주석과 같이 client\_message와 my\_\_message가 의도하지 않은 문자열이 남아있을 상황을 예방하기 위해 memset을 통해 전부 0으로 초기화한다.

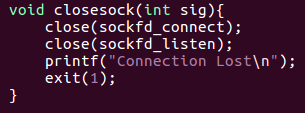
이후 recv를 통해 client로부터 메시지를 받는데 만약 /quit 혹은 /end\_server인 경우 각각에 알맞은 기능이 작동한다.

/quit : 현재 client와의 연결이 종료된다.

/end\_server : 현재 client와의 연결이 종료됨과 동시에 서버 역시 닫힌다.

Client의 메시지를 받은 후 server에서도 답을 할 수 있다.

**(무조건 Client에서 처음 메시지를 보내야 하며 이 후에는 Server-> Client -> Server.. 순으로 순서대로 메시지를 보내야 한다.)**

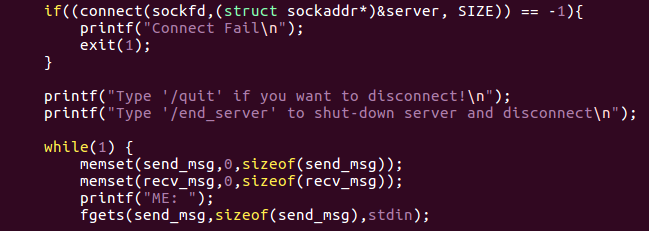


Client와의 연결이 예상치 못한 경우로 끊겼을 경우 서버가 닫히도록 하는 함수이다.

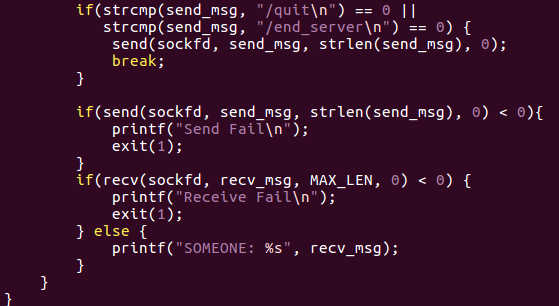
**3. “Client.c” Source-Code**



Server와 연결을 하기 위해 Client에서 socket을 연다.



연결이 된 경우, /quit 와 /end\_server 명령에 대한 설명을 출력함 과 동시에 입력을 받는다.

While문 내의 memset은 Server.c에서와 마찬가지로 의도하지 않은 문자가 입력된 것을 비우기 위한 명령이다.

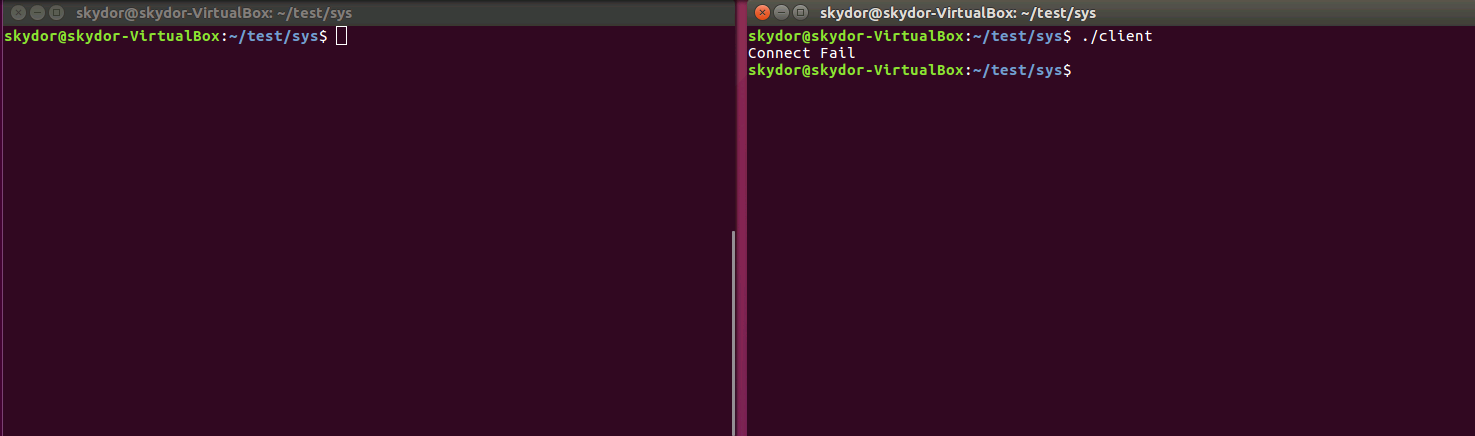
입력받은 문자가 /quit 혹은 /end\_server인지 체크하고 맞다면 이 명령을 server에 보내준 뒤 break를 통해 while문을 빠져나오게 되며, 이는 client의 종료를 의미한다.

그 외의 경우에는 일반적인 메시지이므로 server에 보내준 뒤 server로부터의 응답을 기다린다.

Server로부터 응답이 오면 SOMEONE: 이라는 이름으로 출력을 한다.

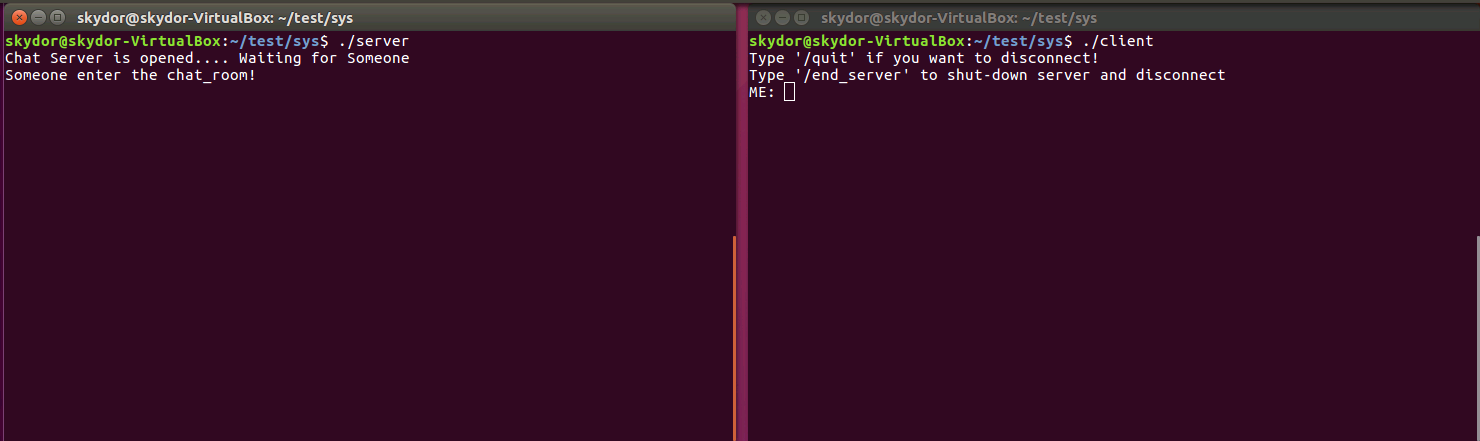
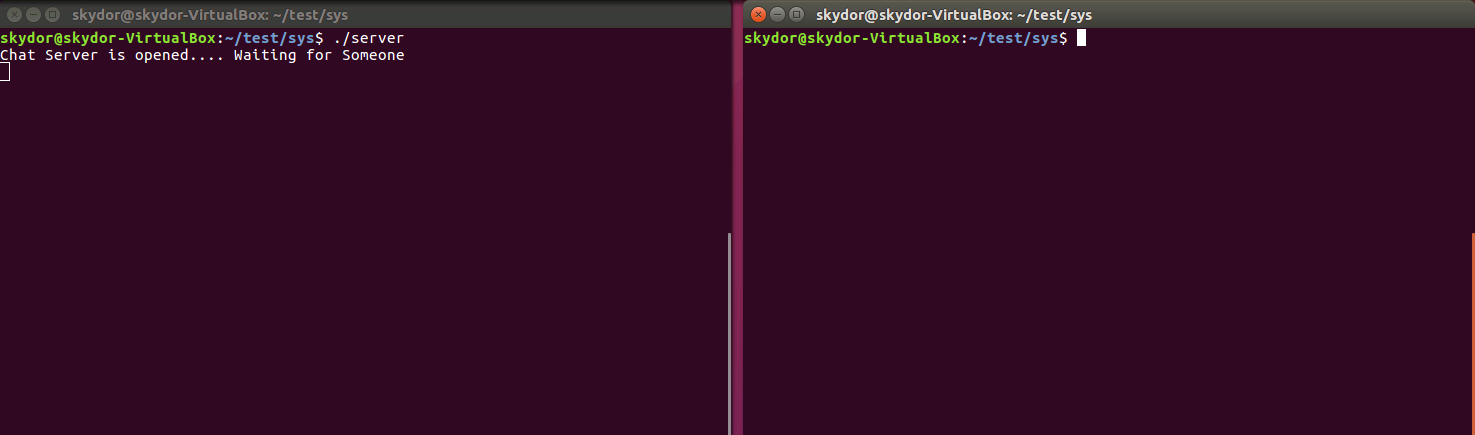
**4. 실행 예시**

(1) Server를 열지 않은 채 Client를 여는 경우

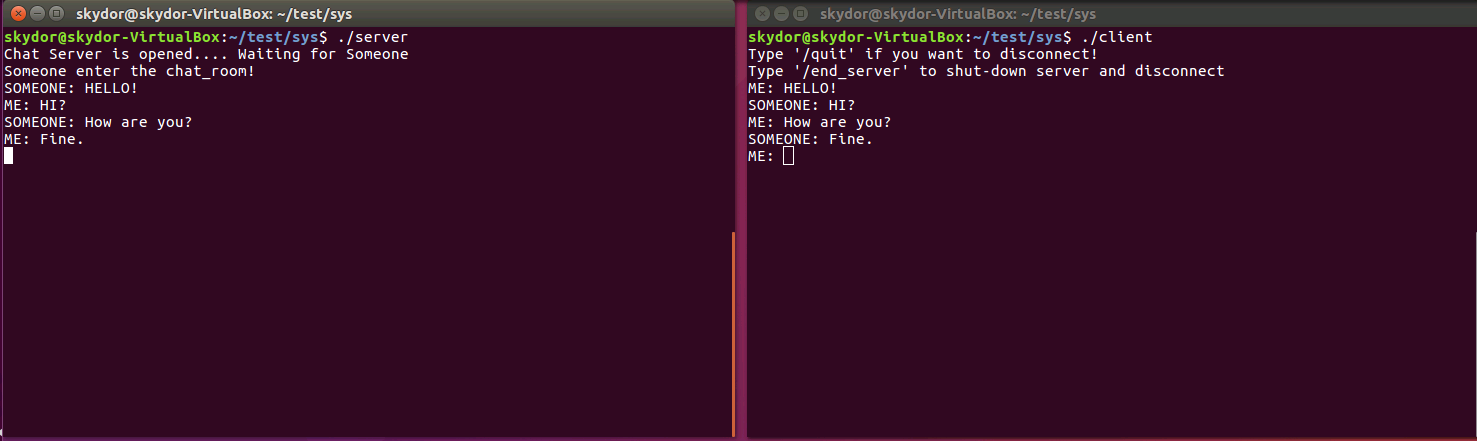


- connect fail이 발생

(2)Server를 연 뒤 Client를 여는 경우

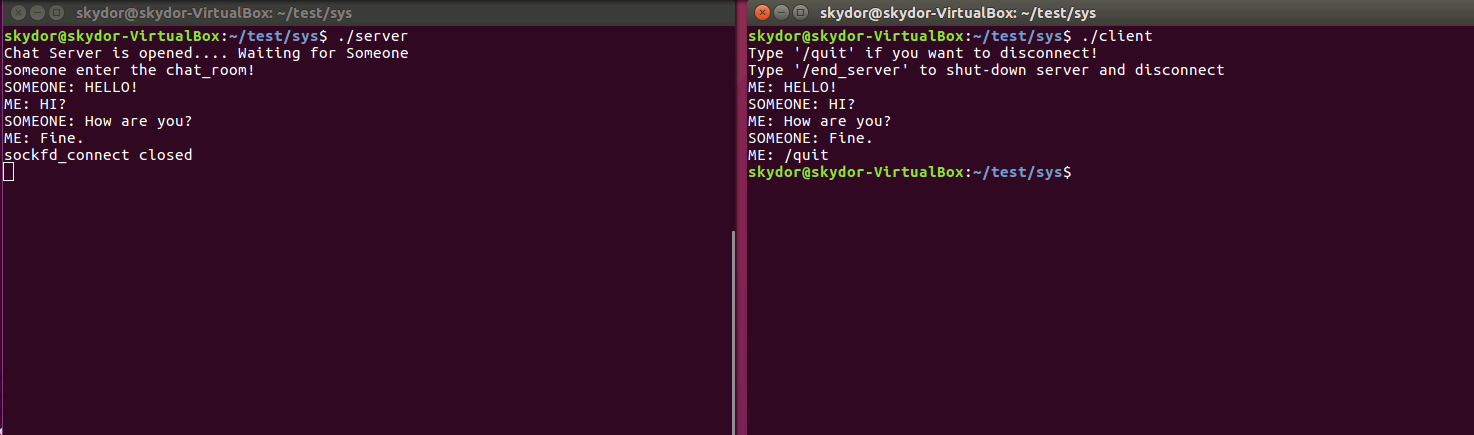


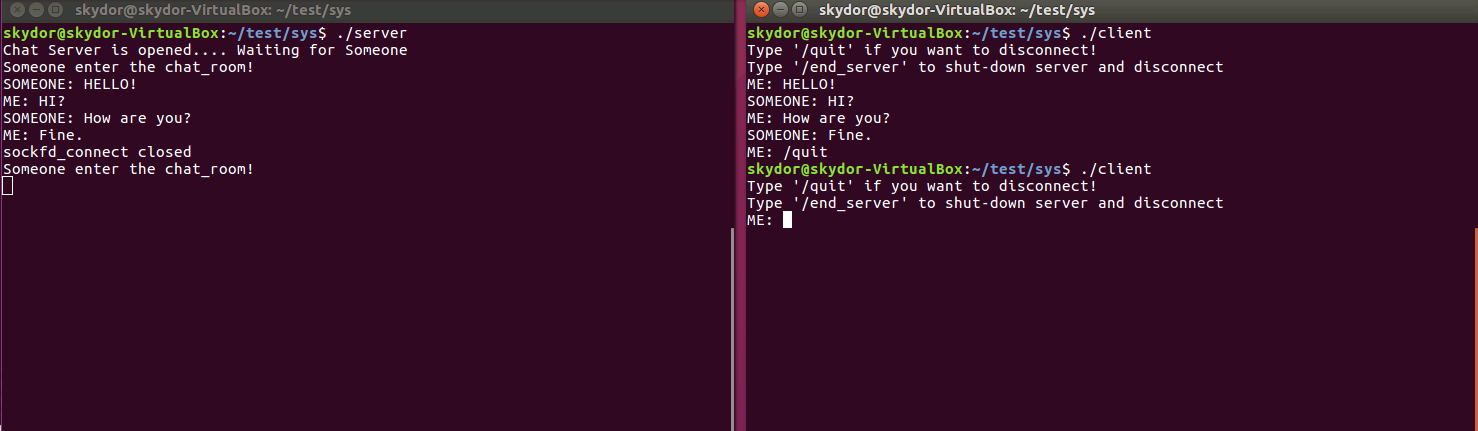
- 정상적으로 프로그램이 작동하면서 Client로부터 입력을 받을 준비를 함.

(3) 1:1 채팅 예시

자신이 메시지를 입력한 경우 “ME: “ 와 함께 표시가 되며,

상대방으로부터 받은 메시지를 출력하는 경우 “SOMEONE: “ 과 함께 표시가 되어 구분 가능.

(4) 명령어 입력 예시

Client에서 /quit을 입력한 경우 client는 즉시 종료되며 server 측에서 sockfd\_connect가 닫히는 것을 확인할 수 있다. 이 후 즉시 client를 다시 실행한 경우 아래 이미지과 같이 server와 다시 연결이 된다.

아예 /end\_server를 통해 서버와 클라이언트를 동시에 닫아버리는 경우 아래와 같이 서버도 닫혀서 즉시 Client를 실행시키는 경우 Connect Fail이 발생함을 알 수 있다.

