ELEC462003 System Programming, Fall 2019

Term Project Assignment – Final Report

Project Topic: **Multi Client-Server Chatting Program**

Team name: 1조

Collaborator: 2017114553 김경숙, 2018110124 최준영

<https://github.com/cjychoi/SP_2019_2_team1>

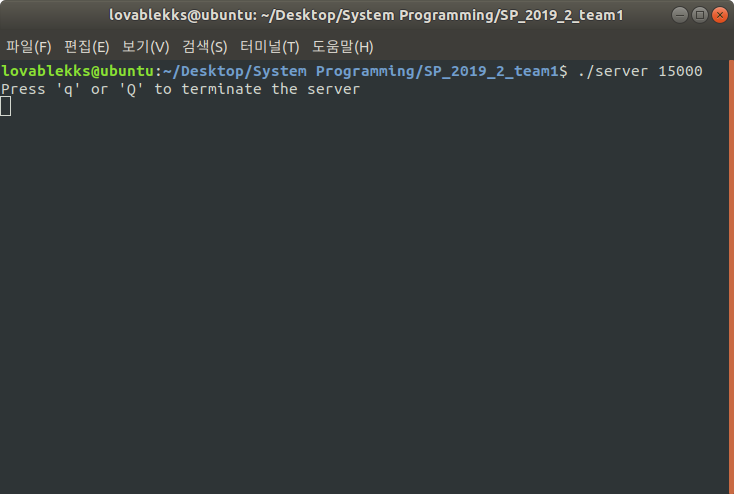
1. Project Topic

* **Multi Client-Server Chatting Program**
* Socket, thread, mutex 및 커널에서 제공하는 다양한 System Call을 사용하여 여러 사용자들이 사용할 수 있도록 만든 다중 채팅 프로그램

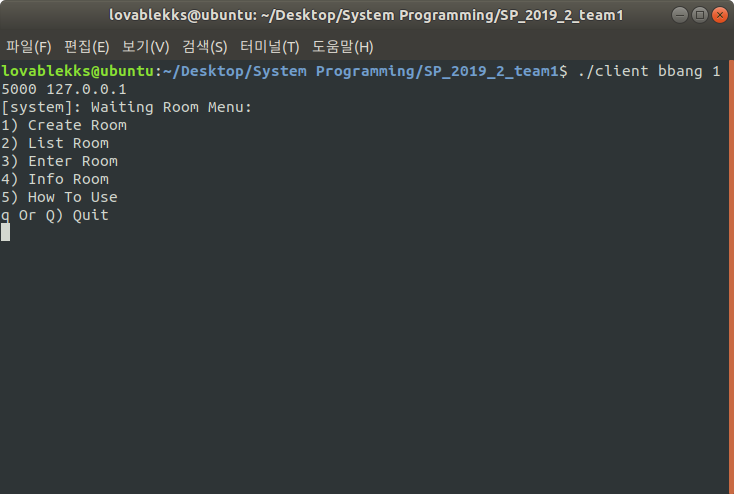
1. Project Summary

* Socket을 사용하여 Server와 Client 간의 통신이 이루어진다.
* Server에서 Client에게 채팅을 위한 다양한 기능을 제공한다.
* Client는 1명 이상의 다른 Client와 채팅 할 수 있다.
* Server에서는 Client가 선택한 메뉴, 입력한 대화 등을 확인할 수 있다.

1. Project Architecture
2. 프로그램은 기본적으로 Client와 Server와 Client로 구성되어 있으며, 이 두 파일을 실행시키면서 채팅 프로그램이 시작된다.
3. Server 파일을 Port Number와 함께, 그리고 Client 파일을 채팅 이름, Port Number, 그리고 IP 주소와 함께 실행시킨다. (이 때, 서로 다른 컴퓨터 간의 연결도 가능하지만, 테스트 영상에서는 하나의 컴퓨터 안에서 실행하기 위해 Loopback IP 주소인 (127.0.0.1)를 사용했다.)



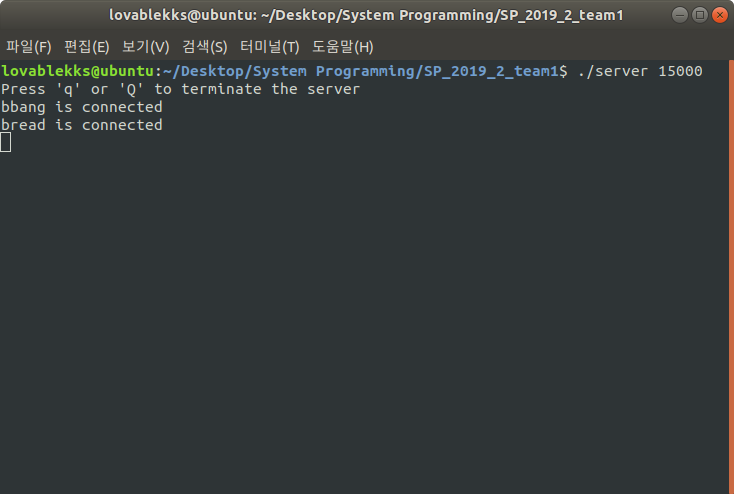
**Server 실행**



**Client 실행**

1. Client가 Server 접속에 성공하면 다음과 같은 기본 메뉴(Waiting Room Menu)가 출력되고, Server에서는 Client의 접속 상태를 알려준다.

**Server에서 Client 접속 상태 확인**

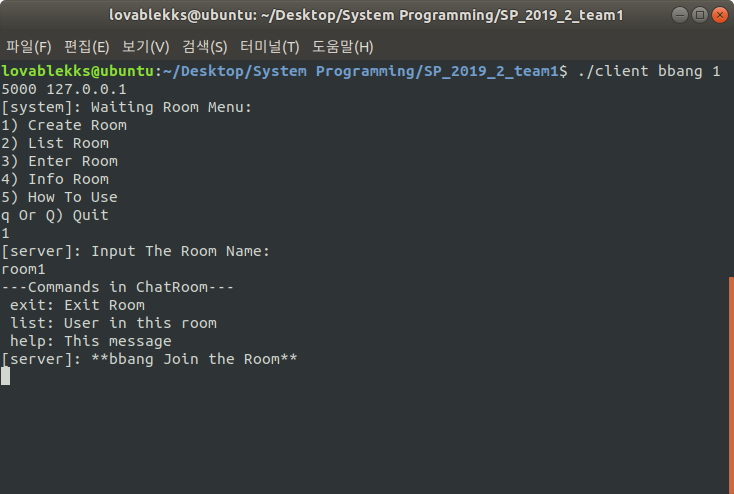


1. Waiting Room Menu

4-1) **Create Room**

- 사용자에게 채팅방의 이름을 입력 받고, 해당 채팅방에 입장한다.

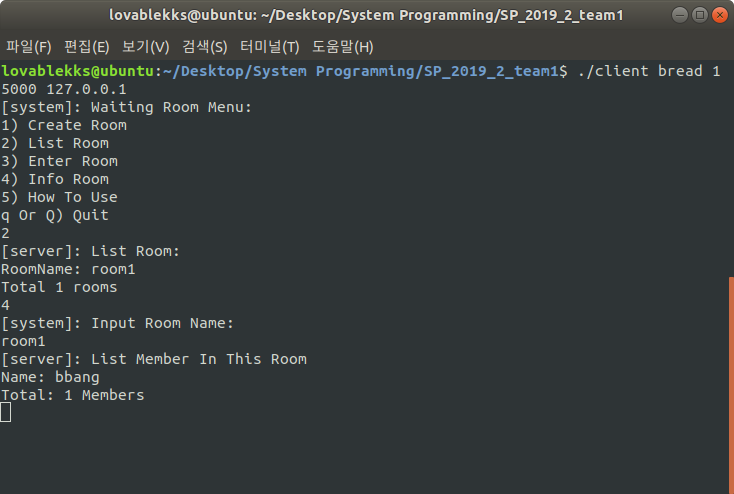
- 채팅방에 입장하면 채팅방에서 사용 가능한 command를 보여준다.



**Client – Create Room**

4-2) **List Room**

- 생성되어 있는 채팅방의 이름을 보여주고, 채팅방의 총 개수를 출력한다.



**Client – List Room**

4-3) **Enter Room**

- 들어가고자 하는 채팅방의 이름을 입력하면 해당 채팅방에 접속하게 된다

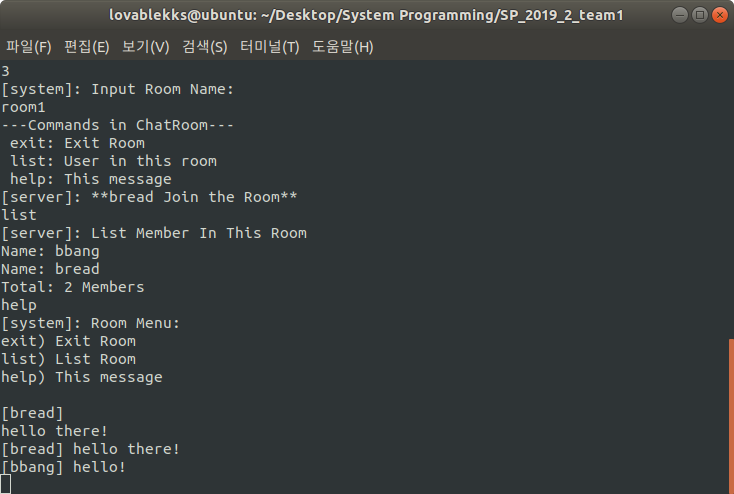
- 채팅방에 입장하면 채팅방에서 사용 가능한 command를 보여준다.

- ‘exit’ 입력 시 채팅 방에서 나온다.

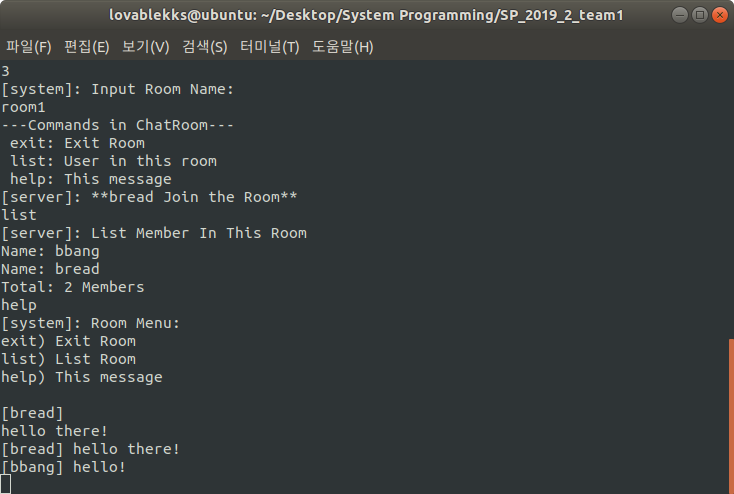
- ‘list’ 입력 시 채팅방에 있는 client의 목록을 보여준다.

- ‘help’ 입력 시 사용 가능한 command를 보여준다.

- 기타 메시지 입력 시 해당 메시지가 server와 다른 client에게 전달된다.



**Client – Enter Room**

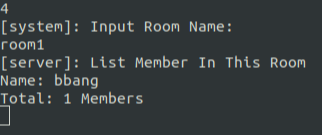


**Client – Enter Room, Chat**

4-4) **Info Room**

- 정보를 조회하고 싶은 채팅방의 이름을 입력 받는다.

- 채팅방에 참여중인 client의 목록과 수를 알려준다.

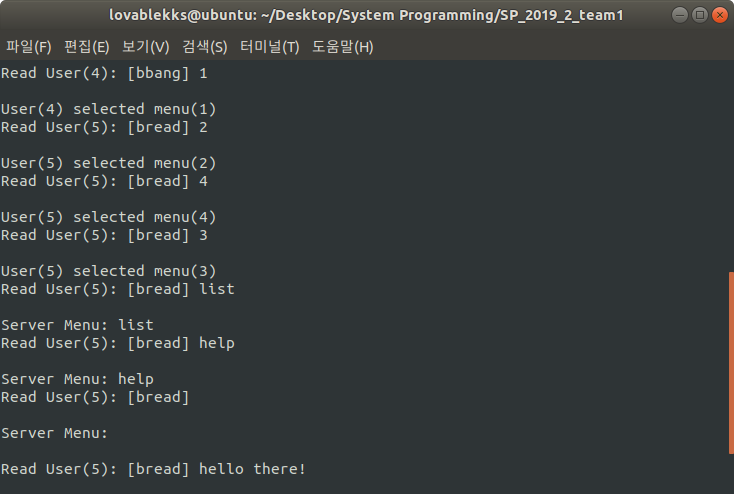


**Client – Info Room**

4-5) **How To Use**

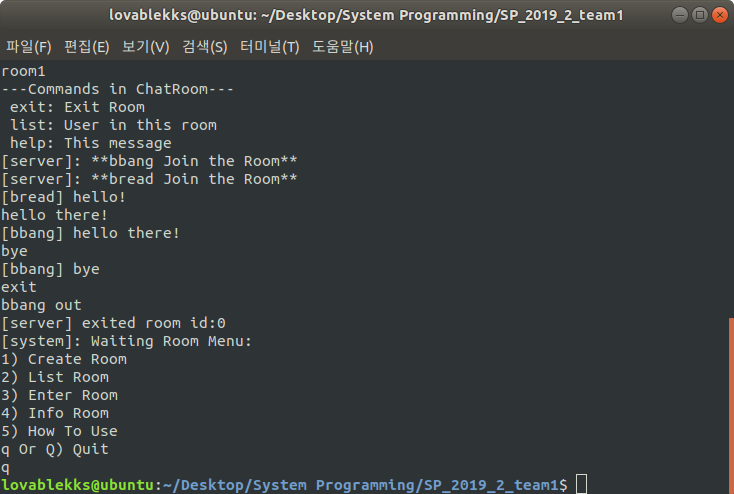
* + - Waiting Room과 Chat Room에서 사용할 수 있는 Command들의 사용법을 보여준다.

1. Server는 Client가 입력한 명령어들을 전달받아 화면에서 확인할 수 있다.



**Server**

1. 채팅방에서는 ‘exit’ 명령어로, Main Menu에서는 ‘q, Q’로 프로그램을 종료할 수 있다.



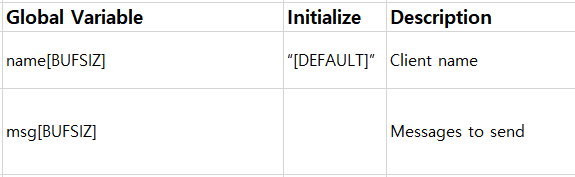
**Exit, q**

1. Project Description

‘client.c’와 ‘server.c’ 코드의 세부 내용을 분석해보자.

1. **client.c**





* **main() 함수**



* 사용자에게 입력 받은 기준으로 socket을 생성한다.
* Server socket address structure의 값들에 사용자에게 입력 받은 port number, IP주소를 저장한다.
* send가 실행 중일 때, receive 요청이 들어오는 경우 간섭 없이 두 요청을 각각 처리 하기 위해 send와 receive에 대한 thread를 따로 생성한다. 이후 pthread\_join 함수를 통해 thread의 종료를 기다린다.
* **send\_msg() 함수**

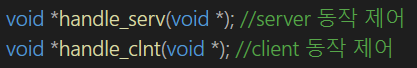


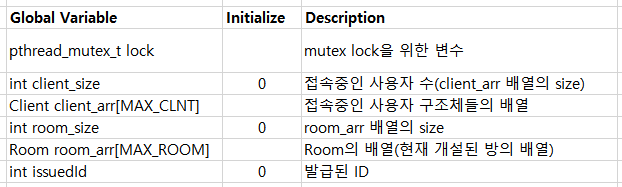
* terminal에서 전달받은 message를 write() system call을 통해 server에 전송한다.
* ‘q’나 ‘Q’를 입력 받은 경우 socket을 종료한다.
* **recv\_msg() 함수**

* 
* server에서 전달받은 메시지를 받는다.
* read() system call을 통해 메시지를 받고, ‘q’나 ‘Q’를 입력받은 경우 소켓을 종료한다.
* fputs() 함수를 통해 client에게 메시지를 출력한다.

1. **server.c**







* **매크로와 전역변수**



* **main() 함수**



* Terminal에서 server 파일을 실행할 때, argv[1]에 Port 번호를 입력 받는다.
* socket() system call을 통해 serv\_sock이라는 소켓을 생성한다.
* serv\_addr이라는 socket 주소 구조체를 생성하여 IP주소와 Port 번호를 저장한다.
* bind() system call을 통해 주소를 할당해준다.
* listen() system call을 통해 socket에서 incoming call을 허용해준다.
* while 반복문을 통해 각 client마다 accept()를 실행한다.
  + accept() system call을 통해 client와 sever를 연결 및 동기화하고, return 값을 clnt\_sock에 저장하여 이 소켓을 통해 정보를 주고받는다.
  + client에게 채팅방에서 사용할 닉네임인 ‘nick’을 입력받고, 해당 정보로 client 구조체를 생성하여 addClient() 함수를 통해 client 배열에 저장한다.
  + handle\_clnt()을 실행하기 위한 thread를 생성하고, 그 이후에 handle\_serv()를 실행하기 위한 thread를 생성한다.
  + thread 생성 이후에 연결된 client의 이름을 출력한다
  + 생성한 두 개의 thread를 소멸시킨다.
* 반복문이 종료되면 server socket을 close 하고 프로그램을 종료한다.
* **handle\_serv() 함수**



* + server에서 전달받은 메시지를 처리하는 함수이다.
  + fgets() 함수를 통해 메시지를 받고, 해당 메시지가 ‘q’ 또는 ‘Q’인 경우에 sendMessageUser() 함수로 client에게 메시지를 전달하고, 모든 client socket을 종료한다.
* **handle\_clnt() 함수**



* + client를 처리하기 위한 함수이다
  + 우선, printWaitingRoomMenu() 함수를 통해 대기화면의 메뉴를 출력한다.
  + client에 대한 socket과 client가 속해 있는 채팅방의 room ID를 불러온다.
  + read() system call을 통해 client socket에서 message를 읽어온다.
  + client가 채팅방 안에 있는 경우
    - getSelectedRoomMenu() 함수를 통해 client가 입력한 menu를 받아온다.
    - serveRoomMenu() 함수를 통해 client에게 서비스를 제공한다.
  + client가 Waiting Room에 있는 경우
    - getSelectedWaitingRoomMenu() 함수를 통해 client가 입력한 menu를 받아온다.
    - serveWaitingRoomMenu() 함수를 통해 client에게 서비스를 제공한다.
  + 더 이상 읽을 message가 없어서 while문을 나오게 되면 removeClient() 함수를 통해 client가 disconnect 되었음을 알린다.
* **addClient() 함수**



* + main 함수에서 연결된 client를 client 배열에 저장하기 위해 실행되는 함수이다.
  + 함수 기능 실행 전에 mutex lock 을 걸고 전역변수인 client\_arr에 접근한다.
  + client 구조체에 필요한 정보를 저장한 후에 mutex unlock으로 돌려놓는다.
  + 저장된 client 구조체를 반환한다.
* **removeClient() 함수**



* + client가 server에서 disconnect 할 때 client 배열에서 삭제하기 위한 함수이다.
  + 반복문을 통해 저장된 client의 정보를 삭제한다.
* **sendMessageUser(), sendMessageRoom() 함수**



* + 특정한 사용자, 또는 채팅방에 있는 모든 사용자에게 메시지를 보낸다.
  + sendMessageUser() 함수는 client에게 메시지를 출력하기 위한 모든 기능에 사용된다.
* **isInARoom() 함수**
  + handle\_clnt() 함수에서 client가 채팅방에 있는 지의 여부를 검사한다.
* **getIndexSapce() 함수**
  + 입력받은 message에서 공백문자의 위치를 찾아준다.
* **getSelectedWaitingRoomMenu() 함수**
  + handle\_clnt() 함수에서 client가 waiting room에 있는 경우 client가 선택한 메뉴를 int형으로 받아 오기 위한 함수이다.
* **getSelectedRoomMenu() 함수**
  + handle\_clnt() 함수에서 message에서 사용자가 원하는 menu를 받기 위해 사용된다.
  + 공백문자를 기준으로 복사된 문자열을 menu 배열에 저장한다.
* **addRoom() 함수**
  + createRoom() 함수에서 client에게 받은 이름으로 채팅방을 만든다.
* **removeRoom() 함수**
  + exitRoom() 함수에서 client가 채팅방에서 나올 때, 방이 비었으면 삭제한다.
* **isExistRoom() 함수**
  + 채팅방 ID를 받아서 해당 채팅방을 발견하면 TRUE, 발견하지 못하면 FALSE를 반환한다
* **enterRoom() 함수**



* + client가 특정 채팅방 ID를 가진 방에 입장을 시도한다.
  + 해당 채팅방이 존재하지 않는 경우 에러 메시지를 출력하고,
  + 해당 채팅방이 존재하는 경우 확인 메세지를 출력한다.
* **createRoom() 함수**



* + main menu에서 사용자가 채팅방을 만들기 위한 함수이다.
  + 사용자가 입력한 이름의 채팅방이 존재하는 경우 해당 채팅방에 enterRoom()하게 되고,
  + 기존에 없는 경우에는 addRoom()으로 새로운 채팅방을 만든다.
* **listRoom() 함수**
  + serveWaitingRoomMenu() 함수에서 client가 2번 메뉴인 list room을 선택하는 경우 현재 생성되어 있는 모든 채팅방의 목록을 출력한다.
* **listMember() 함수**
  + serveWaitingRoomMenu() 함수와 serveRoomMenu() 함수에서 해당 채팅방에 접속되어 있는 client들의 목록을 출력한다.
* **getRoomId() 함수**



* + serveWaitingRoomMenu() 함수에서 client가 들어가고 싶은 채팅방의 이름을 입력 받아 해당 채팅방의 ID를 반환해준다.
* **serveWaitingRoomMenu() 함수**
  + Switch 문을 통해 Main menu에서 1) Create Room, 2) List Room, 3) Enter Room, 4) Info Room, 5) How to Use, q or Q) Quit 등의 기능을 수행한다.
* **exitRoom() 함수**
  + serveRoomMenu() 함수에서 client가 ‘exit’ 명령어를 입력했을 때 채팅방을 빠져 나오고, 방이 비는 경우 삭제해준다.
* **serveRoomMenu() 함수**
  + 채팅방에 들어왔을 때, client가 입력하는 메시지에 따라 exit, list, help, chatting의 기능을 수행한다.
* **isEmptyRoom() 함수**
  + 채팅방에 참여하는 client가 있는지 확인해주고, TRUE 또는 FALSE를 반환한다.

1. **사용된 System Call**

* close(), read(), write(), exit()
* socket(), bind(), listen(), accept(),
* pthread\_mutex\_init()
* pthread\_create(), pthread\_dtach()
* pthread\_mutex\_lock(), pthread\_mutex\_unlock()