Socket Programming

2013171045 남정호

자료 구조

구현의 핵심이 되는 데이터는 `rooms` 와 `client_detail` 이 있습니다. 두 데이터 모두 서버에서 관리합니다.

rooms

```
{
    room_name:{
        room_name: room11
        creator: client
        members = [
          client
       ]
    }
}
```

room_name(방 이름)을 key로 가지고 있는 dictionary 입니다. room_name 값이 고유 값인 점과 크기가 커질 경우 리스트로 관리하였을 때 검색에 O(N)의 시간 복잡도가 걸릴 것을 고려하여 dictionary 자료형을 선택하였습니다.

Value 에는 방 이름, 방장의 닉네임과 참여 인원들의 정보가 들어있습니다. room_name 은 key 에도 들어있고 value에도 들어있어 중복이 발생하지만, 데이터 처리의 편의성을 위해 허용하였습니다.

client_detail

```
{
    <client>:{
        "state":"wait",
        "room_name": "",
        "user_name": "",
    }
}
```

위와 동일한 이유로 dictionary 자료형을 선택하였습니다. Key 에는 client 객체를 넣었습니다. Value 에는 해당 클라이언트의 state와 state가 "chat" 일 경우 참여 중인 방 이름, 닉 네임을 포함하였습니다.

구현 아이디어

1. 상태
   클라이언트에서 클라이언트의 상태를 관리하는 것은 보안적인 측면에서 매우 취약하다고 판단하였습니다. 따라서 모든 클라이언트의 상태는 서버에서 관리합니다.

2. 명령어 처리 방식
   서버에서 입력한 명령어는 서버에서 처리합니다. 하지만 클라이언트에서 입력한 명령어들은 대부분 상태 정보가 필요하거나 서버와의 통신이 필수적입니다. 따라서 클라이언트가 서버에게 명령어를 넘기면 서버가 처리를 하여 클라이언트에게 결과를 응답하는 방향으로 구현하였습니다.

주요 함수들

library

send_message(target, msg)
msg를 encode하여 target에게 전송한다.

receive_message(target)
target으로부터 받은 메시지를 decode 하여 반환한다.

Server

operate_server_command(msg, rooms, server_socket, client_details)
서버에서 직접 타이핑한 명령어를 처리한다.

handle_client_message(msg, rooms, r, client_details)
클라이언트에게 받은 명령어를 처리한다.

end_chat_service(server_socket, client_details)
채팅 서비스를 종료한다.

kill_room(msg, rooms, client_details)
채팅 방을 kill 한다.

show_clients(client_details)
접속중인 클라이언트의 리스트와 상태를 출력한다.

join_room(msg, rooms, client, client_details)

/join 명령어를 처리한다.

create_room(msg, rooms, client, client_details)

/create 명령어를 처리한다.

whisper(msg, rooms, client, client_detail)

/whisper 명령어를 처리한다.

run_exit(rooms, client, client_details)

/exit 명령어를 처리한다.

propagate_chat_message(msg, rooms, client, client_detail)

송신한 채팅 메시지를 같은 방에 있는 다른 사용자들에게 전파한다.

Client

handle_server_msg(client_socket, msg)

서버로부터 받은 명령어를 처리한다.

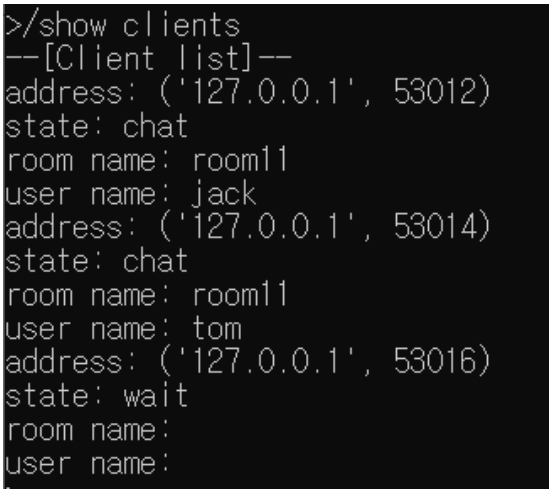
propagate_server_for_client_end (client_socket)

서버에게 해당 클라이언트가 종료되었음을 알린다.

추가 기능들

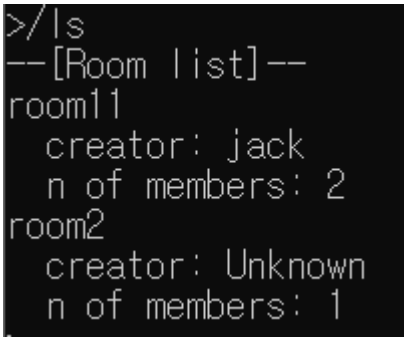
요구 조건 이외에 몇 가지 기능을 추가 & 개선하였습니다.

1. `/show clients`
   서버에서 이 명령어를 입력하면 현재 접속중인 클라이언트들의 목록과 상태 등 정보를 출력합니다.



2. `/ls`

   방 리스트와 상세 정보를 함께 표시합니다.



3. Unknown 사용자

   하나의 채팅 방에 여러 명의 Unknown이 존재할 수 있습니다. 하지만 Unknown에게는 귓속말을 보낼 수 없습니다.

4. 기타

   이외에도 여러 예외 상황에 대한 처리를 추가하였습니다.