

컴퓨터 네트워크 과제2 – 디자인 및 구현 문서

작성자: 2021038131 장준혁

1. 프로그램 설명

클라이언트는 비즈니스 로직 처리 등의 복잡한 작업을 하지 않는다. 클라이언트는 단순히 명령어를 포함한 데이터를 서버에 보내고 서버에서 받은 데이터를 그대로 화면에 출력해주는 정도의 프로그램이다. 대부분의 처리는 서버에서 이루어진다.

클라이언트에서 서버로 데이터를 보낼 때는 사용자가 입력한 텍스트 앞에 "#WRITE @"를 붙여서 보낸다. 즉 "#WRITE @사용자가 입력한 텍스트"의 형태로 보낸다. 이렇게 하는 이유는 서버에서 데이터를 받고 @를 기준으로 텍스트를 분리하기 위해서이다. 다른 방법으로 해도 되지만 이렇게 하는게 서버에서 받은 데이터 중 메시지만을 분리하는 코드가 간결해진다.

클라이언트는 자신이 현재 채팅방에서 몇 번째 텍스트까지 읽었는 지 정보를 저장해서 서버에 채팅 기록 동기화 요청을 보낼 때 해당 인덱스를 같이 넣어서 "#READ 인덱스번호"의 형식으로 메시지를 보낸다. 이는 5초 간격으로 자동으로 이루어진다. 서버는 #READ 요청을 받으면 해당 인덱스 이후에 온 채팅들을 보낸다. 이때 메시지들은 @기호로 구분해서 보낸다. #READ 요청을 보냈을 때 새로운 채팅이 없으면 서버는 " . "로 응답한다. 채팅 기록 동기화 코드는 유저 IO에 의해 실행이 막히지 않도록 서브 스레드에서 실행된다.

2. 핵심 클래스 설명

ClientInfo: 서버에서 새로운 클라이언트의 TCP 연결이 들어오면 해당 클래스의 인스턴스를 생성한다. 이 클래스는 서브 스레드에서 실행된다. 서버는 해당 클래스의 통신을 기다려야 하는데, 이것이 메인 스레드의 실행을 방해하면 안 되기 때문이다. 해당 클래스는 사용자 이름, 사용자가 들어간 채팅방, 소켓 정보 등을 저장하고 있다.

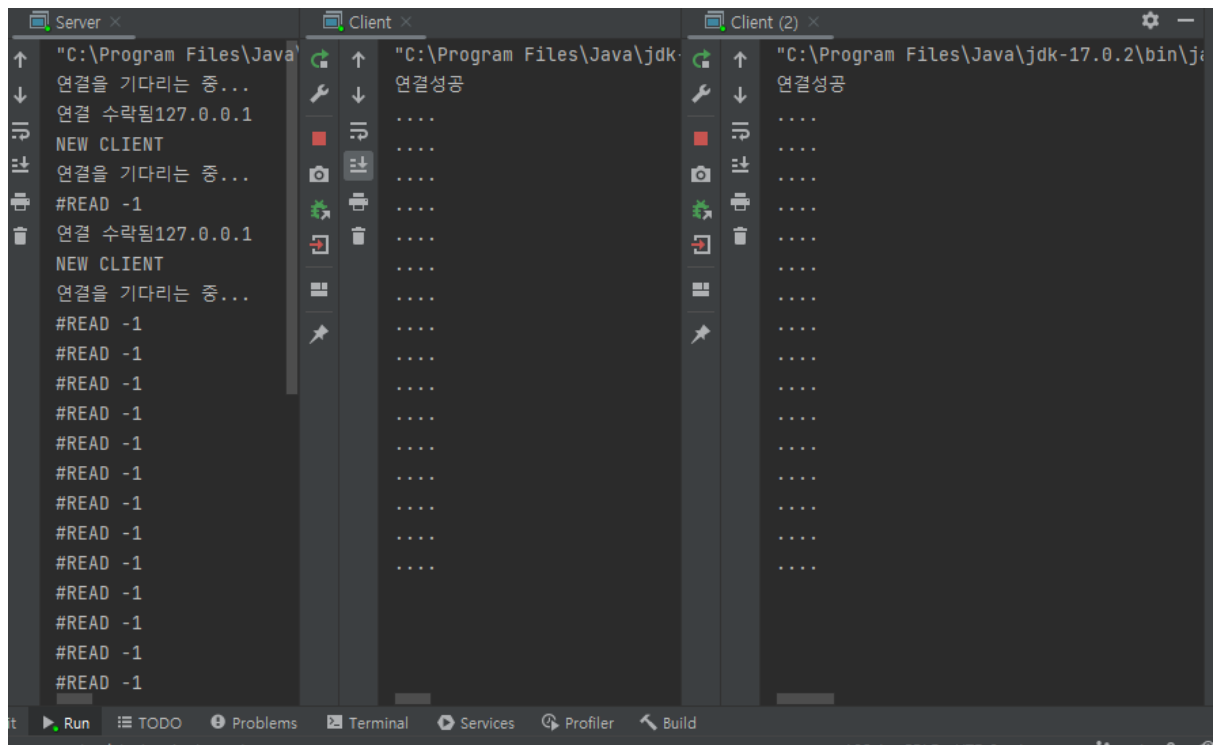
ChatRoomInfo: 채팅방의 정보를 저장하는 클래스이다. 사용자가 #CREATE 명령어를 서버에 전송했을 때 새로운 채팅방 인스턴스가 생성된다. 해당 클래스에는 채팅방의 이름, 연결된 클라이언트 목록, 채팅 기록이 저장된다.

ClientChatReciver: 서브스레드에서 실행되며 주기적으로 서버에 채팅기록 동기화 요청을 보낸다.

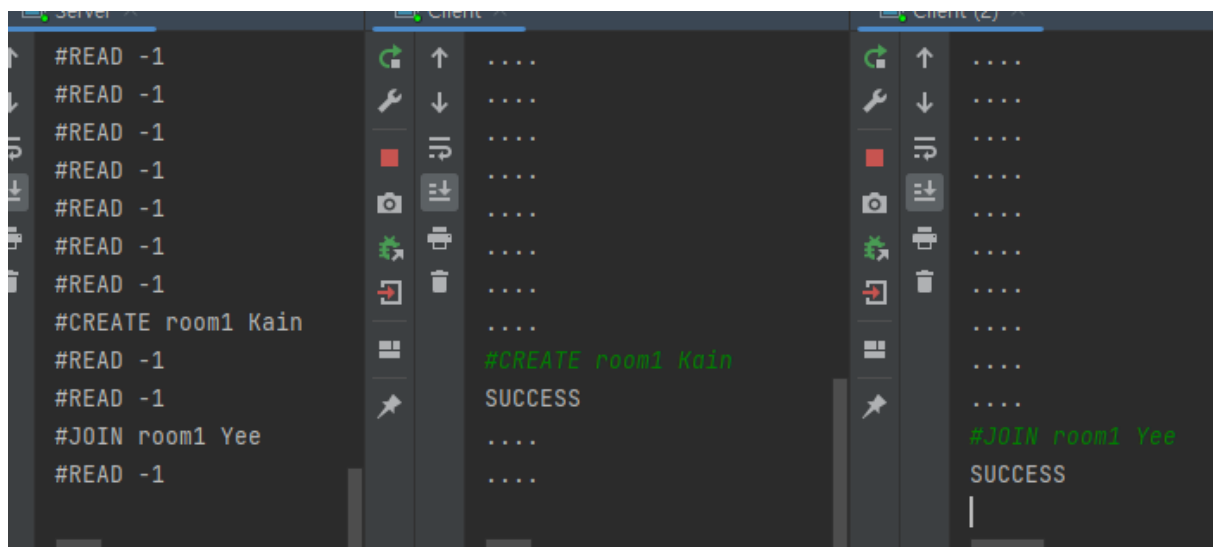
3. 실행방법 및 실행 결과

젯브레인사의 인텔리제라는 IDE에서 코드를 작성했다. UTF-8로 인코딩되어 있으므로 이클립스처럼 EUC-KR을 기본 인코딩으로 사용하는 IDE에서는 인코딩이 깨질 경우 인코딩 방식을 수정해서 다시 불러와야 한다. 물론 코드에는 한글을 사용하지 않아서 큰 문제는 없을 것으로 예상된다.

과제 가이드라인에 기술된 바와 같이 Server와 Client 파일을 실행하고 실행 시 포트 번호를 인자로 넘겨주면 된다. 인자가 없을 경우 기본 포트 번호인 2020, 2021로 설정된다.



위와 같이 서버 한 개와 클라이언트 2개를 실행하자.



클라이언트1에는 **#CREATE room1 Kain** 을 입력하고, 클라이언트2에는 **#JOIN room1 Yee** 를 입력하자.

```
Run: Server x Client x Client (2) x
#READ -1
#READ -1
#READ 0
#READ 0
#READ 0
#READ 0
#WRITE @Hello! I'm Yee
Yee: Hello! I'm Yee
#READ 0
#READ 0
#READ 1
....
....
....
....
Hi! I'm Kain
Chat Sent!
Kain: Hi! I'm Kain
....
Yee: Hello! I'm Yee
....
....
Kain: Hi! I'm Kain
....
Hello! I'm Yee
Chat Sent!
Yee: Hello! I'm Yee
|
```

채팅을 보내면 채팅기록이 서로 동기화가 잘 되는 것을 볼 수 있다.

```
Server (3) x Client x Client (2) x
"C:\Program Files\Java\j
연결을 기다리는 중...
연결 수락됨127.0.0.1
NEW CLIENT
연결을 기다리는 중...
#READ -1
연결 수락됨127.0.0.1
NEW CLIENT
연결을 기다리는 중...
#READ -1
#READ -1
#READ -1
#CREATE room1 Kain
#READ -1
#JOIN room1 Yee
#READ -1
#READ -1
#STATUS
#READ -1
#READ -1
#STATUS
#READ -1
#READ -1
#READ -1
"C:\Program Files\Java\jdk
연결성공
....
....
#CREATE room1 Kain
SUCCESS
....
....
#STATUS
RoomName: room1
[Members]
Kain
Yee
....
....
"C:\Program Files\Java\jdk-17.0.2\bin\j
연결성공
....
....
#JOIN room1 Yee
SUCCESS
....
#STATUS....|
#STATUS
RoomName: room1
[Members]
Kain
Yee
....
....
```

#STATUS를 입력하면 방제목과 참여자 목록을 확인할 수 있다.

```
Run: Server (3) x Client x Client (2) x
#READ -1
#STATUS
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#READ -1
#EXIT'
#EXIT
#READ -1
#READ -1
#EXIT
#READ -1
#READ -1

#CREATE room1 Kain
SUCCESS
....
....
#STATUS
RoomName: room1
[Members]
Kain
Yee
....
....
....
....
....
....
....
....
....
#EXIT'
#EXIT'
#EXIT
SUCCESS
....
....

#JOIN room1 Yee
SUCCESS
....
....
#STATUS....
#STATUS
RoomName: room1
[Members]
Kain
Yee
....
....
....
....
....
....
....
....
#EXIT....
#EXIT
SUCCESS
....
|
```

#EXIT을 입력하면 정상적으로 방에서 나가진다.