**오픈소스SW활용**

**팀 프로젝트**

**소켓 프로그래밍을 통한 채팅방 구현**

**13조**

**김일규 – 32180955**

**신수환 – 32182369**

**이충훈 – 32183662**

**최민아 – 32187637**

**목차**

1. 프로젝트 목표 ---------------------------------------------------------------- 3

2. 주제 및 기능 ----------------------------------------------------------------- 3

- 주제

- 기능

3. 구현 내용 및 방법 ----------------------------------------------------------- 4

- 소켓 프로그래밍

- 구현 모델 (서버/클라이언트)

- 서버 구현 (스레드)

4. 역할 분담 --------------------------------------------------------------------- 5

5. 개발 일정 --------------------------------------------------------------------- 5

1. 프로젝트 목표

- 소켓 프로그래밍으로 기본적인 서버 프로그램을 만들 수 있다.

- 서버, 클라이언트 개념을 이해하고 프로그램에 적용할 수 있다.

- 스레드를 이용하여 채팅방을 나누고 관리할 수 있다.

2. 주제 및 기능

2.1 주제

이번 오픈소스 프로젝트 과제의 목표는 소켓 프로그래밍을 이용한 채팅방 프로그램이다. 이는 서버의 경우 소켓 프로그래밍과 스레드를 사용하여 사용자가 서버에 접속하고, 원하는 채팅방에 들어가거나 생성하며, 들어간 채팅방을 통해 다른 사용자와 소통할 수 있도록 구현할 것이다. 클라이언트의 경우 서버에 접속하여 채팅방에 입장하고, 같은 채팅방에 속한 다른 사용자로부터 온 메세지를 받고, 자신이 다른 사용자에게 메세지를 전달하도록 구현할 것이다. 추가적으로 채팅방에 들어가 있는 사용자가 서버에 접속해 있는 다른 사용자를 자신이 속한 채팅방에 초대를 하는 기능까지 구현하는 것을 목표로 하고 있다.

2.2 기능

1) 채팅방 입장: 사용자는 원하는 채팅방을 생성하거나, 접속가능

2) 메세지 송수신: 사용자는 자신이 속한 채팅방에 속한 다른 사용자로부터 메세지를 수신가능

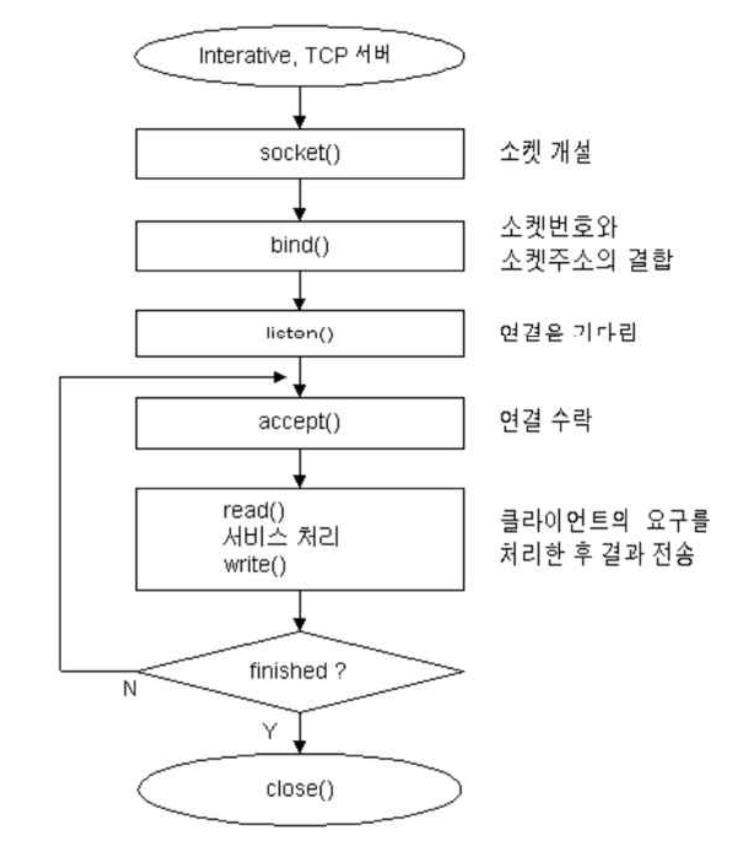
3) 채팅방 초대: 사용자는 서버에 접속해 있는 다른 사용자를 자신이 속한 채팅방에 초대가능

3. 구현 내용 및 방법

3.1 소켓 프로그래밍

소켓 프로그래밍이란 네트워크로 연결되어 있는 서로 다른 두 컴퓨터가 데이터를 주고받을 수 있도록 하는 네트워크 프로그래밍으로 이를 위한 통신을 위한 객체인 소켓과 데이터 송수신 방법은 운영체제가 지원해준다.

\* 소켓 프로그래밍을 통한 서버 생성 절차

1) socket() 생성

- 다른 노드(컴퓨터)들과 통신하기 위한 소켓 생성

2) bind()

- 서버의 ip주소와 포트(번호)를 소켓에 할당

3) listen()

- 응답을 받을 수 있도록 대기상태로 유지

4) accept()

- 클라이언트로부터 연결 요청이 오면 수락

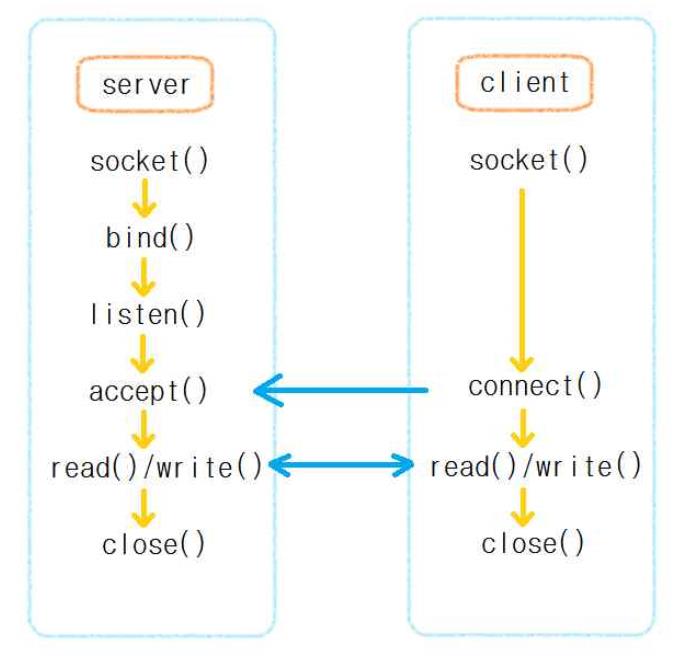
5) read(), write()

- 데이터 송수신

6) close()

- 연결 종료

3.2 구현 모델

서버-클라이언트 모델을 사용한다. 서버 클라이언트 모델이란 서비스 요청자인 클라이언트와 서비스 자원의 제공자인 서버 간에 작업을 분리해주는 구조를 의미한다. 따라서 서버는 사용자의 접속을 받아 채팅방을 제공하고, 사용자들 간의 소통을 관리하며, 클라이언트는 서버에 접속하여 채팅방을 통해 다른 사용자와 통신하도록 설계할 것이다.

3.3 서버 구현

서버는 사용자의 접속과 채팅방 관리를 동시에 수행해야 한다. 또한 채팅방을 관리하기 위해서는 많은 데이터들을 서로 공유해야 한다. 따라서 이를 구현하기 위해 다중 프로세스를 운영하는 것보다 각각의 일들을 스레드 단위로 처리하는 것이 효율적이다. 즉, 각 스레드들이 사용자의 접속, 채팅방 생성/삭제, 사용자의 채팅방 입장, 채팅방 내의 통신을 관리하도록 설계할 것이다.

4. 역할 분담

|  |  |
| --- | --- |
| 김일규 | 서버 구조 설계, 사용자 채팅방 초대 기능 구현, 테스트 서버 구축 |
| 신수환 | 채팅방 생성/삭제, 사용자의 채팅방 입장 구현 |
| 이충훈 | 클라이언트 구현 |
| 최민아 | 하나의 채팅방 내에서 사용자 간의 통신 구현 |

5. 개발 일정

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1주차 | 2주차 | 3주차 | | | 4주차 | 5주차 | |
| 서버 구조 설계 |  |  | | | | | | |
| 채팅방 생성/삭제, 입장 |  |  | | |  | | | |
| 채팅방 내의 통신 |  |  | | | | | |  |
| 클라이언트 |  |  | | | | | |  |
| 채팅방 초대 |  | | |  | | | | |

서버를 구현하기 위해서는 서버의 구조와 공유데이터를 먼저 설계하는 것이 필요하다. 따라서 이를 1주차에 설계를 마치고 2주차부터 구조에 맞춰 세부적인 구현을 진행한다. 이후 각 기능이 테스트 가능해질 때마다 가상머신을 통해 이를 테스트한다.

6. GitHub Link

https://github.com/Student5421/2022\_OpenSourceSW\_Project\_Socket\_Programming