

Project #0 : TCP Socket Programming

이름 : 신상윤

학번 : 2018008559

과제 설명 : socket을 사용하여 통신하는 클라이언트/서버를 만드는 방법을 배우고, 직접 구축하여 통신이 어떻게 이루어지는지 확인한다. 또한, socket programming에서 자주 사용하는 문법들을 익힐 수 있도록 한다.

source Files :

Server

```
import java.io.*;
import java.net.*;

public class TCPServer {
    public static void main(String[] args) throws Exception {
        String clientSentence;
        String capitalizedSentence;
        ServerSocket welcomeSocket = new ServerSocket(8888);

        while(true) {
            System.out.println("client waiting...");
            Socket connectionSocket = welcomeSocket.accept();
            System.out.println("client connected!!");

            DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());
            BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

            clientSentence = inFromClient.readLine();

            capitalizedSentence = clientSentence.toUpperCase() + '\n';

            outToClient.writeBytes(capitalizedSentence);
        }
    }
}
```

Client

```
import java.io.*;
import java.net.*;

public class TCPClient {
    public static void main(String[] args) throws Exception{
        String sentence;
        String modifiedSentence;
        String tmp;

        BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));

        Socket clientSocket = new Socket("192.168.56.1", 8888);

        DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

        BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

        sentence = inFromUser.readLine();

        outToServer.writeBytes(sentence + '\n');

        modifiedSentence = inFromServer.readLine();
        System.out.println("FROM SERVER: " + modifiedSentence);

        clientSocket.close();
    }
}
```

1. 서버와 클라이언트가 같은 포트 번호로 socket programming을 해야 한다.
2. host는 cmd 창에서 ipconfig을 입력했을 때 나오는 IPv4 주소들을 사용해야 한다.
3. DataOutputStream, BufferedReader를 새로 만드는 2개의 줄은 말 그대로 새로 stream, Reader 을 만드는 것이기 때문에 순서가 바뀌어도 상관없다.
4. 항상 서버를 먼저 실행시켜야 한다.

Instructions:

cmd 창에서 컴파일러를 추가하고 "javac 이름.java"로 컴파일하여 "java 이름"으로 실행했다.

How the program works:

서버와 클라이언트를 연결하고, 클라이언트에서 서버로 보낸 문장을 서버에서 모두 대문자로 바꾸고 다시 클라이언트로 보내 클라이언트에서 결과가 출력되는 프로그램이다.

먼저 서버를 실행시키면 `ServerSocket welcomeSocket = new ServerSocket(8888);` 줄에서 포트가 8888인 친구들을 가져오는 새로운 서버를 만든다. 이후 `while`문에서 `Socket connectionSocket = welcomeSocket.accept();` 줄과 같이 포트가 8888인 client를 기다린다. 이제 client를 실행시키면 `Socket clientSocket = new Socket("192.168.0.3", 8888);` 줄에서 서버와 연결이된다. 연결됨과 동시에 둘 다 선언된 `Stream`과 `Reader`를 새로 만든다.

이후 서버는 `clientSentence = inFromClient.readLine();`에서 client의 요청을 기다린다. client는 `outToServer.writeBytes(sentence + '\n');`에서 서버로 "sentence\n"를 보내고, 서버는 받아서 모두 대문자로 만든 다음 `outToClient.writeBytes(capitalizedSentence);` 에서 client로 보낸다.

그것을 client는 `modifiedSentence = inFromServer.readLine();`로 받고 client에서 출력한 후 socket을 닫고 client를 종료한다. 서버는 계속 실행되고 있다.

Results:

Server

```
C:\Users\kas12\OneDrive\바탕 화면\2학년 2학기\컴네\과제\project0>javac TCPserver.java
C:\Users\kas12\OneDrive\바탕 화면\2학년 2학기\컴네\과제\project0>java TCPserver
client waiting...
client connected!!
client waiting...
client connected!!
client waiting...
```

Client

```
C:\Users\kas12\OneDrive\바탕 화면\2학년 2학기\컴네\과제\project0>javac TCPClient.java
C:\Users\kas12\OneDrive\바탕 화면\2학년 2학기\컴네\과제\project0>java TCPClient
my name is shin sang yoon
FROM SERVER: MY NAME IS SHIN SANG YOON

C:\Users\kas12\OneDrive\바탕 화면\2학년 2학기\컴네\과제\project0>java TCPClient
i'm fine thank you and you?
FROM SERVER: I'M FINE THANK YOU AND YOU?

C:\Users\kas12\OneDrive\바탕 화면\2학년 2학기\컴네\과제\project0>
```

server와 client를 연결하고, client에서 문장을 입력하면 서버에서 모두 대문자로 만들어준 후 client에 다시 보내 client에서 결과가 출력된다. 예를 들어 위와 같이 i'm fine thank you and you?를 입력해서 서버로 보내주면 서버에서 모두 대문자로 바꾼 후 client로 다시 보내 I'M FINE THANK YOU AND YOU? 를 출력한다.

something Else: 과제의 실행 결과가 있으면 참고하기 좋을 것 같습니다.