



# Computer Networks: Fall 2022 Assignment #2

## Computer Networks: Fall 2022 Programming Assignment 1

2021045796 김도겸

**TCP 프로토콜을 이용한 클라이언트-서버 방식의 오픈 채팅  
프로그램을 구현한다**

### 1. 프로그램 실행 방법

```
PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> javac Server.java -encoding UTF8
PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> javac FileTrans.java -encoding UTF8
PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> javac Client.java -encoding UTF8
```



```
javac Server.java -encoding UTF8
javac FileTrancs.java -encoding UTF8
javac Client.java -encoding UTF8
```

순서대로 Class file을 만들어줍니다.

### 2. 채팅 프로그램 실행

#### 1) Server



```
java Server 2020 2021
```

java Server 포트번호1 포트번호2  
를 입력하여 Server를 실행한다.

```
PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Server 2020 2021
##서버가 시작되었습니다.
█
```

## 2) Client

Server가 실행된 이후



java Client 127.0.0.1 2020 2021

java Client 서버IP 포트번호1 포트번호2  
를 입력하여 Client를 실행한다.

```
PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
█
```

## 3. 채팅방 생성 및 참여

Client를 실행한 이후부터 채팅방 내에서 #으로 시작하는 문장은 아래에서 설명하는 동작을 수행하기 위해서만 사용됩니다.

Client는 새로운 채팅을 시작하기 위한 #CREATE와 기존의 채팅에 참여하기 위한 #JOIN 명령어를 실행할 수 있습니다.

```
PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
#CREATE KAKAO KIM
>현재 대화에 참여하고 있는 유저수 :0명
[##] 채팅방 (KAKAO) 에 입장하였습니다.
```

```

PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
#JOIN KAKAO PARK
현재 대화에 참여하고 있는 유저 수 :1명

[##] 채팅방 (KAKAO) 에 입장하였습니다.

```

Client는 chatState와 roomInfo라는 변수를 이용하여 Client의 상태와 방의 상태를 관리합니다. 둘 다 초기값은 0으로 설정되어있으며, CREATE를 입력받은 경우에는 roomInfo를 1로 설정하며 HashMap으로 채팅방의 이름을 전송합니다. JOIN을 입력받은 경우에는 roomInfo를 2로 설정하여 채팅방이 존재하는지를 확인합니다.

```

if(str[0].equals("#CREATE")){

    if(!str[2].equals("")){
        out.writeUTF("req_logon|"+str[2]);
        name = str[2];
    }else{
        System.out.println("[##] 공백을 입력할수없습니다.\r\n" +
            "▶다시 입력해 주세요:");
    }
    if(!str[1].equals("")){
        out.writeUTF("req_makeRoom|"+name+"|"+str[1]);
        Client.roomInfo= 4;
    }else{
        System.out.println("[##] 공백을 입력할수없습니다.\r\n" +
            "▶다시 입력해 주세요:");
    }
}

```

이름의 중복 여부를 확인할 때는 req\_logon을 붙여서 송신하여 중복 여부를 수신받습니다. 채팅방의 개설이 가능한지 확인하기 위해서 req\_makeRoom을 붙여서 송신하며 채팅방의 개설이 가능할 경우 채팅방을 개설합니다.

```

else if(str[0].equals("#JOIN")){
    if(!str[2].equals("")){
        out.writeUTF("req_logon|"+str[2]);
        name = str[2];
    }else{
        System.out.println("[##] 공백을 입력할수없습니다.\r\n" +
            "▶다시 입력해 주세요:");
    }
    if(!str[1].equals("")){
        out.writeUTF("req_enterRoom|"+name+"|"+str[1]);
        Client.roomInfo= 4;
    }else{

```

```

        System.out.println("[##] 공백을 입력할수없습니다.\r\n" +
            "▶다시 입력해 주세요:");
    }
}

```

JOIN의 경우에도 같은 방식으로 진행합니다.

로그인이 완료되면 chatState를 2로 설정하여 채팅방 내에서 대화가 가능한 상태로 전환합니다.

사용자 이름과 채팅방 이름에 공백이 입력된 경우에는 전송하지 않으며, 다시 입력하라는 메시지를 출력합니다.

Server는 CREATE 명령을 통해 채팅방 이름을 입력받았을 때 HashMap이랑 연결하여 유저를 저장합니다.

CREATE 명령을 받았을 때 이미 존재하는 채팅방은 이미 존재하는 이름이라는 메시지를 전송합니다.

```

PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
#CREATE KAKAO MIM
○현재 대화에 참여하고 있는 유저수 :2명
[##] 중복된 이름이 존재합니다

```

JOIN 명령을 받았을 때 존재하지 않는 채팅방의 경우에는 존재하지 않는 채팅방이라는 메시지를 전송합니다.

```

PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
#JOIN PALA PAP
○현재 대화에 참여하고 있는 유저수 :2명
[##] 입력하신 [PALA]는 존재하지 않는 채팅방입니다.
▶채팅방을 다시 입력해 주세요:

```

채팅방을 생성하거나 입장할 때 사용자의 이름이 중복된 경우에 이미 존재하는 이름이라는 메시지를 전송합니다.

```
PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
#CREATE MALO KIM
[##] 중복된 이름이 존재합니다
[##] 중복된 이름이 존재합니다
█
```

```
if(msg.startsWith("logon#yes")){ //로그온 시도 (대화명)
    Client.chatState= 1; //채팅 상태를 로그인 된 상태로 변경.
    //logon#yes|그룹리스트
    System.out.println(msgArr[0]);
    //System.out.println("▶방 생성하기/방 입장하기:");

}
else if(msg.startsWith("logon#no")){ //로그온 실패 (대화명)

    Client.chatState= 0;
    System.out.println("[##] 중복된 이름이 존재합니다");
    //1. 이름이 중복될경우(서버전체 or 그룹) logon#no 패킷이 서버로부터 전달됨.
    //2. 이름이 중복될경우 서버에서 자체적으로 name(1), name(2) 이런식으로 중복되지 않게
    변경하는 방법.

}
else if(msg.startsWith("enterRoom#yes")){ //그룹입장

    //enterRoom#yes|지역
    System.out.println("[##] 채팅방 (" +msgArr[0]+") 에 입장하였습니다.");
    Client.chatState= 2; //채팅 상태 변경 ( 채팅방입장 완료로 대화가능상태)

}
else if(msg.startsWith("enterRoom#no")){
    //enterRoom#no|지역
    System.out.println("[##] 입력하신 [" +msgArr[0]+ "]는 존재하지않는 채팅방입니
다.");
    System.out.println("▶채팅방을 다시 입력해 주세요:");
    Client.chatState= 0;
}
}
```

#### 4. 채팅

Client는 송신자의 채팅방에 전달하고자 하는 메시지를 서버로 전송합니다.

```

[##] 채팅방 (KAKAO) 에 입장 하였습니다 .
[##] PARK님이 입장 하셨습니다 .
[##] PARK님이 입장 하셨습니다 .
[##] PARK님이 입장 하셨습니다 .
Hello Park
[KIM] Hello Park
[PARK] I'm Paka

```

```

[##] 채팅방 (KAKAO) 에 입장 하였습니다 .
[KIM] Hello Park
I'm Paka
[PARK] I'm Paka

```

Server는 송신자의 메시지를 수신하여 같은 채팅방에 있는 모든 수신자에게 메시지를 전송 합니다.

```

msgParse():msg?    KIM|Hello Park
msgParse():msg?    PARK|I'm Paka

```

채팅방이 여러 개가 존재하여도 복수의 채팅방의 요청을 Thread를 이용하여 전달 할 수 있습니다.

<pre> 작 하세요! https://aka.ms/PSWindows PS C:\Users\admin\Language\Java\Assignment2_202104579 6.김도겸&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... CREATE map1 c11 &gt;현재 대화에 참여하고있는 유저수 :0명  [##] 채팅방 (map1) 에 입장 하였습니다 . [##] c12님이 입장 하셨습니다 . [##] c13님이 입장 하셨습니다 . hello [c11] hello im c1c1 [c11] im c1c1 [c12] hi c1c1c1c1c1 </pre>	<pre> 새로운 기능 및 개선 사항에 대 한 최신 PowerShell을 보 a.ms/PSWindows PS C:\Users\admin\Language\Java\Assignment2_202104579 6.김도겸&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... #30IN map1 c12 &gt;현재 대화에 참여하고있는 유저수 :1명  [##] 채팅방 (map1) 에 입장 하였습니다 . [##] c13님이 입장 하셨습니다 . [c11] hello [c11] im c1c1 hi c1c1c1c1c1 [c12] hi c1c1c1c1c1 </pre>	<pre> 작 하세요! https://aka.ms/PSWindows PS C:\Users\admin\Language\Java\Assignment2_2021045796 _김도겸&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... #30IN map1 c13 [##] 완전한 입력입니다.. &gt;방 이름을 다시 지정해주세요 : #30IN map1 c13 &gt;현재 대화에 참여하고있는 유저수 :2명  [##] 채팅방 (map1) 에 입장 하였습니다 . [c11] hello [c11] im c1c1 [c12] hi c1c1c1c1c1 </pre>	<pre> Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.  새로운 기능 및 개선 사항에 대 한 최신 PowerShell을 설치 하세요! dows PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... CREATE map2 c14 &gt;현재 대화에 참여하고있는 유저수 :3명  [##] 채팅방 (map2) 에 입장 하였습니다 . I'm alone [c14] I'm alone </pre>
--	---	---	---

<pre> 지 하세요! https://aka.ms/PSWindows PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도원&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... #CREATE map1 c11 o현재 대화에 참여하고있는 유저수 :0명  [##] 채팅방 (map1) 에 입장하였습니다. [##] c12님이 입장하였습니다. [##] c13님이 입장하였습니다. hello [c11] hello Is c1cl [c11] Im c1cl [c12] hi c1clc1cl [##] c13님이 입장하였습니다. [c13] wawa [c12] hawawa </pre>	<pre> Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.  새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치 하 요! https://aka.ms/PSWindows PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도원&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... #JOIN map1 c12 o현재 대화에 참여하고있는 유저수 :4명  [##] 채팅방 (map1) 에 입장하였습니다. [##] c13님이 입장하였습니다. wawa [c13] wawa [c12] hawawa </pre>	<pre> 지 하세요! https://aka.ms/PSWindows PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도원&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... #JOIN map1 c13 #방 이름을 다시 지정해주세요: #JOIN map1 c13 o현재 대화에 참여하고있는 유저수 :2명 [##] 채팅방 (map1) 에 입장하였습니다. [c11] hello [c11] Im c1cl [c12] hi c1clc1cl #EXIT [##] 클라이언트를 종료합니다. PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도원&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... #JOIN map2 c15 o현재 대화에 참여하고있는 유저수 :3명  [##] 채팅방 (map2) 에 입장하였습니다. holololollo [c15] holololollo </pre>	<pre> Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.  새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치 하 PSWindows PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도원&gt; java Client 127.0.0.1 2020 2021 [##] 서버와 연결이 되었습니다..... #CREATE map2 c14 o현재 대화에 참여하고있는 유저수 :3명 [##] 채팅방 (map2) 에 입장하였습니다. I'm alone [c14] I'm alone [##] c15님이 입장하였습니다. [c15] holololollo </pre>
---	---	---	---

```

else if(msg.startsWith("show")){ //서버에서전달하고자하는 메시지

        //show|메시지내용
        System.out.println(msgArr[0]);

    }else if(msg.startsWith("say")){ //대화내용
        //say|아이디|대화내용
        System.out.println("["+msgArr[0]+"] "+msgArr[1] );

    }

```

show의 경우에는 #STATUS와 같이 서버가 개인에게 보내는 메시지를 출력할 때 사용되며, say의 경우에는 client가 server에 보낸 메시지를 출력할 때 사용합니다.

## 5. 파일 송수신

FileTrans.java를 이용하여 FileSender과 FileReceiver를 구현하였습니다. #PUT (파일 주소)를 입력하여 파일을 송신하고, #GET 명령어를 이용하여 수신할 수 있도록 하려고했으나, 다음 명령을 수신하지 못하고 채팅을 입력받지 못하는 상태에 들어가게되고, 이러한 문제를 해결하지 못했습니다.

파일을 수신하는 경우에는 chatState를 5로 변경하여 포트번호2번과 연결하여 파일을 전송 받으려고 했었습니다.

```

}else if(Client.chatState== 5) { //5 : 상대방이 파일전송을 시도하여 사용자의 수락요청을 기다림.
        //fileSend|result)

```

```

        if(msg.trim().equalsIgnoreCase("#GET")){
            out.writeUTF("fileSend|yes|" + Client.inputIP);
        }
    else if(msg.trim().equalsIgnoreCase("y")){
        out.writeUTF("fileSend|yes");
    }else if(msg.trim().equalsIgnoreCase("#NO")){
        out.writeUTF("fileSend|no");
    }else{
        System.out.println("입력한 값이 올바르지 않습니다.");
        out.writeUTF("fileSend|no");
    }

    Client.chatState=2; //파일전송수락요청에대한 응답완료 상태
}

```

```

else if(msg.startsWith("fileSender")){ //파일을 보내기위해 파일서버 준비

        //fileSender|filepath;
        System.out.println("fileSender:"+InetAddress.getLocalHost().getHostAddress());

        System.out.println("fileSender:"+msgArr[0]);
        //String ip=InetAddress.getLocalHost().getHostAddress();
        //TODO ip, port number, file name을 어떻게 해결할 지 고민해야함,
//TODO이 경우에 과연 전체 참가자에게 파일이 전송될지 궁금함.
        try {
            new FileSender(msgArr[0]).start(); //쓰레드 실행.
            //new FileSender(msgArr[0], InetAddress.getLocalHost().getHostAddress(), 2000, msgArr[0]).start(); //쓰레드 실행.
        } catch (Exception e) {
            System.out.println("FileSender 쓰레드 오류:");
            e.printStackTrace();
        }

        }else if(msg.startsWith("fileReceiver")){ //파일받기
            //fileReceiver|ip|fileName;

            System.out.println("fileReceiver:"+InetAddress.getLocalHost().getHostAddress());
            System.out.println("fileReceiver:"+msgArr[0]+"-"+msgArr[1]);

            String ip = Client.inputIP; //서버의 아이피를 전달 받음
            String fileName = msgArr[1]; //서버에서 전송할 파일이름.

            try {
                new FileReceiver(ip,fileName).start(); //쓰레드 실행.
            } catch (Exception e) {
                System.out.println("FileSender 쓰레드 오류:");
                e.printStackTrace();
            }
        }
}

```



## 6. 채팅방 나가기

Client는 #EXIT 명령어를 이용하여 채팅방을 종료할 수 있습니다.

```
[##] 채팅방 (KAKAO) 에 입장 하였습니다 .  
[##] PARK님이 입장 하셨습니다 .  
[##] PARK님이 입장 하셨습니다 .  
[##] PARK님이 입장 하셨습니다 .  
Hello Park  
[KIM] Hello Park  
[PARK] I'm Paka  
#EXIT  
[##] 클라이언트를 종료합니다 .
```

Server는 채팅방을 떠난 Client를 HashMap에서 제거합니다.

클라이언트가 #EXIT 한 경우 HashMap에서 삭제되어 그 이름을 다시 사용할 수 있습니다.

```
else if(msg.trim().startsWith("#") && !msg.trim().equals("#GET")){  
    //명령어 기능 추가. ( /접속자 , 전달할메시지... 등 )  
    //클라이언트단에서 체크  
    if(msg.equalsIgnoreCase("#EXIT")){  
        System.out.println("[##] 클라이언트를 종료합니다.");  
        System.exit(0);  
        break;  
    }else{  
        out.writeUTF("req_cmdMsg|" + name + "|" + msg);  
        //req_cmdMsg|대화명|/접속자  
    }  
}
```

명령어가 아닌 #의 경우에는 입력받지 않습니다. #EXIT를 입력받으면 클라이언트를 종료합니다 메시지를 출력합니다.

## 7. 채팅방 상태 확인

Client는 #STATUS 명령어를 이용하여 채팅방에 접속 중인 채팅방의 이름과 구성원의 정보를 확인 할 수 있습니다.

```

PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
#JOIN KAKAO KIM
○현재 대화에 참여하고 있는 유저수 :1명

[##] 채팅방 (KAKAO) 에 입장하였습니다.
#STATUS
==채팅방 상태==
채팅방 이름:KAKAO
참여자 이름:
KIM (*)
PARK
==2명 접속중==

```

Client가 채팅방에서 떠나면 Server는 채팅방을 떠난 Client를 HashMap에서 제거합니다.

```

PS C:\Users\admin\Language\Java\Assignment2_2021045796_김도겸> java Client 127.0.0.1 2020 2021
[##] 서버와 연결이 되었습니다.....
#JOIN KAKAO PARK
○현재 대화에 참여하고 있는 유저수 :1명

[##] 채팅방 (KAKAO) 에 입장하였습니다.
[KIM] Hello Park
I'm Paka
[PARK] I'm Paka
[##] KIM님이 입장하셨습니다.
[##] KIM님이 입장하셨습니다.
#EXIT
[##] 클라이언트를 종료합니다.

```

이후에 다시 #STATUS를 통해 Client 목록을 확인하면 #EXIT를 통해 채팅방을 나간 사람의 이름이 사라진 것을 확인할 수 있습니다.

```
[##] 채팅방 (KAKAO) 에 입장 하였습니다 .
```

```
#STATUS
```

```
==채팅방 상태==
```

```
채팅방 이름 :KAKAO
```

```
참여자 이름 :
```

```
KIM (*)
```

```
PARK
```

```
==2명 접속 중==
```

```
#STATUS
```

```
==채팅방 상태==
```

```
채팅방 이름 :KAKAO
```

```
참여자 이름 :
```

```
KIM (*)
```

```
==1명 접속 중==
```

```
□
```

```
else if(msg.startsWith("req_cmdMsg")){ //명령어 전송
    //req_cmdMsg|대화명|접속자
    if(msgArr[1].trim().equals("#STATUS")){
        out.writeUTF("show|"+showUserList()); //접속자 출력
    }
}
```

명령어로 #STATUS를 입력받았을 때 userList를 출력합니다.

## 8. 참고

[https://m.blog.naver.com/PostView.naver?  
isHttpsRedirect=true&blogId=javaking75&logNo=140189868916](https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=javaking75&logNo=140189868916)

<https://javakorea.tistory.com/218>

<https://godsu94.tistory.com/4>