

컴퓨터 네트워크 Assignment #1

디자인 및 구현 문서

2017033654 산업공학과

류동원

구현한 파일은 UDPMulticast.java, ListeningThread.java, WritingThread.java로 총 3가지 입니다.

1. UDPMulticast.java

각각의 피어 프로그램을 수행하기 위한 파일입니다.

```
public static void main(String args[]) throws IOException {
    int port;
    String address;
    String userName;
    Scanner sc = new Scanner(System.in);
    System.out.print("Java Peer ");
    port = sc.nextInt();

    System.out.println("참여하고자 하는 채팅방 이름과 별명을 설정해주세요.");
    System.out.println("예시 : #JOIN (채팅방 이름) (사용자 이름)");

    String command = sc.next();
    if (command.equals("#JOIN")) {
        address = sc.next();
        userName = sc.next();
        address = addrHashing(address);

        new UDPMulticast(port, address, userName);
    } else {
        System.out.println("잘못 입력하셨습니다.");
        System.out.println("프로그램을 종료합니다.");
    }
}
```

main method는 multicast port number를 입력받습니다. (ex : Java Peer 999)

- 각각의 피어 프로그램이 동일한 multicast message를 수신하기 위해서는 동일한 port number를 입력해야 합니다. (peer1 : Java Peer 999 & peer2 : Java Peer 999)

이후 참여하고자 하는 채팅방 이름과 사용자 이름을 입력합니다.

(ex : #JOIN cnet peer1)

- 입력받은 첫번째 element가 #JOIN이 아니면 프로그램을 종료합니다.
- 입력받은 첫번째 element가 #JOIN이면 두번째, 세번째 element를 각각 address, userName 변수에 할당을 하고, 입력받은 address를 addrHashing 함수를 이용하여 Multicast address로 변환합니다.
- 각각의 피어 프로그램을 수행할 때 입력한 두번째 element인 address가 앞

선 피어 프로그램의 문자열과 동일하다면 동일한 채팅방에 참여하게 되고, 새로운 문자열이 입력되면 새로운 채팅방에 개설됩니다.

```
public static String addrHashing(String address) {
    try {
        MessageDigest mdSHA256 = MessageDigest.getInstance("SHA-256");
        mdSHA256.update(address.getBytes("UTF-8"));
        byte[] sha256Hash = mdSHA256.digest();
        address = "225";
        for(int i=1; i<=3; i++) {
            address = address + "." + (sha256Hash[sha256Hash.length - i] + 128);
        }
    } catch(NoSuchAlgorithmException e) {
        e.printStackTrace();
    } catch(UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    return address;
}
```

그 이후 입력받은 multicast port number, 변환된 address와 userName을 인자로 하여 UDPMulticast 함수를 호출합니다.

```
public UDPMulticast(int port, String address, String userName) {
    this.port = port;
    try {
        socket = new MulticastSocket(port);
        group = InetAddress.getByName(address);
        socket.joinGroup(group);

        ListeningThread t1 = new ListeningThread(socket);
        WritingThread t2 = new WritingThread(socket, group, port, userName);
        t1.start();
        t2.start();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

UDPMulticast 함수에서는 multicast port number를 이용하여 MulticastSocket을 생성하고 MulticastSocket의 ListeningThread와 WritingThread를 각각 실행합니다.

- 각각의 peer program은 두 가지 thread를 수행함으로써 채팅 메시지를 전송하는 동작과 채팅 메시지를 읽는 동작을 동시에 수행할 수 있습니다.

2. ListeningThread.java

```
MulticastSocket clientSocket;
public ListeningThread(MulticastSocket clientSocket) {
    this.clientSocket = clientSocket;
}
```

UDPMulticast.java 에서 ListeningThread가 호출될 때 MulticastSocket 인스턴스 전달인자로 받습니다.

```

public void run() {
    try {
        while (true) {
            byte[] receiveData = new byte[512];
            DatagramPacket receivePacket =
                new DatagramPacket(receiveData, receiveData.length);
            clientSocket.receive(receivePacket);
            String sentence = new String(receivePacket.getData());
            System.out.println(sentence);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

run 메소드를 수행하면서 clientSocket.receive(receivePacket) 함수를 통하여 소켓에 multicast message가 도착하기를 기다리고 message가 도착하면 화면에 출력합니다.

3. WritingThread.java

```

MulticastSocket clientSocket;
InetAddress IPAddress;
int port;
String userName;

public WritingThread(MulticastSocket clientSocket, InetAddress IPAddress, int
port, String userName) {
    this.clientSocket = clientSocket;
    this.IPAddress = IPAddress;
    this.port = port;
    this.userName = userName;
}

```

UDPMulticast.java에서 WritingThread가 호출될 때 MulticastSocket 인스턴스, muticast address, multicast port number와 해당 peer의 user name을 전달인자로 받습니다.

```

public void run() {
    try {
        while(true) {
            BufferedReader inFromUser = new BufferedReader(new
InputStreamReader(System.in));
            String sentence = inFromUser.readLine();
            if (sentence.equals("#EXIT")) {
                clientSocket.leaveGroup(IPAddress);
                clientSocket.close();
            } else if ((sentence.substring(0,1)).equals("#")) {
                continue;
            }
            sentence = userName + " : " + sentence;
            byte[] sendData = new byte[512];
            sendData = sentence.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, port);
            clientSocket.send(sendPacket);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

run 메소드를 수행하면서 peer가 입력한 문자를 DatagramPacket으로 변환시킨 후

multicast Socket의 IP address와 port number를 입력하여 전송합니다.

이때 #EXIT를 입력하면 multicast group에서 탈퇴한 후 socket을 close 합니다.

#EXIT가 아닌 #으로 시작하는 문자열이 입력되는 메시지가 전달되지 않습니다.

다른 peer program들이 message를 수신할 때 송신자를 알 수 있도록 sendPacket에 user name을 포함하여 전송합니다.

<실행화면>

<pre>Terminal: Local x Local (2) x Local (3) x Local (4) x + v dongwon-ryu@DonDon-Mac src % java UDPMulticast Java Peer 9999 참여하고자 하는 채팅방 이름과 별명을 설정해주세요. 예시 : #JOIN (채팅방 이름) (사용자 이름) #JOIN cnet1 peer2 peer1 : hi hello, nice to meet you peer2 : hello, nice to meet you</pre>	<pre>Terminal: Local x Local (2) x Local (3) x Local (4) x + v dongwon-ryu@DonDon-Mac src % java UDPMulticast Java Peer 9999 참여하고자 하는 채팅방 이름과 별명을 설정해주세요. 예시 : #JOIN (채팅방 이름) (사용자 이름) #JOIN cnet1 peer1 hi peer1 : hi peer2 : hello, nice to meet you</pre>
---	--

- multicast port number 9999로 cnet1 채팅룸에서 peer1과 peer2과 통신

```
Terminal: Local x Local (2) x Local (3) x Local (4) x +
dongwon-ryu@DonDon-Mac src % java UDPMulticast
Java Peer 9999
참여하고자 하는 채팅방 이름과 별명을 설정해주세요.
예시 : #JOIN (채팅방 이름) (사용자 이름)
#JOIN cnet2 peer1
Hello, peer2 in cnet2!!
peer1 : Hello, peer2 in cnet2!!
peer2 : Hi, peer1 in cnet2!!
#
#EXIT
java.net.SocketException: Socket closed
    at java.base/sun.nio.ch.DatagramSocketAdaptor.send(DatagramSocketAdaptor.java:100)
    at java.base/java.net.DatagramSocket.send(DatagramSocket.java:540)
    at WritingThread.run(WritingThread.java:32)
Caused by: java.nio.channels.ClosedChannelException
    at java.base/sun.nio.ch.DatagramChannelImpl.ensureOpen(DatagramChannelImpl.java:100)
    at java.base/sun.nio.ch.DatagramChannelImpl.blockForSend(DatagramChannelImpl.java:100)
    at java.base/sun.nio.ch.DatagramSocketAdaptor.send(DatagramSocketAdaptor.java:100)
    ... 2 more
java.net.SocketException: Socket closed
    at java.base/sun.nio.ch.DatagramSocketAdaptor.receive(DatagramSocketAdaptor.java:100)
    at java.base/java.net.DatagramSocket.receive(DatagramSocket.java:540)
    at ListeningThread.run(ListeningThread.java:16)
Caused by: java.nio.channels.AsynchronousCloseException
    at java.base/sun.nio.ch.DatagramChannelImpl.endOfChannel(DatagramChannelImpl.java:100)
    at java.base/sun.nio.ch.DatagramChannelImpl.blockForReceive(DatagramChannelImpl.java:100)
    at java.base/sun.nio.ch.DatagramSocketAdaptor.receive(DatagramSocketAdaptor.java:100)
    ... 2 more
dongwon-ryu@DonDon-Mac src %
```

```
Terminal: Local x Local (2) x Local (3) x Local (4) x + v
dongwon-ryu@DonDon-Mac src % java UDPMulticast
Java Peer 9999
참여하고자 하는 채팅방 이름과 별명을 설정해주세요.
예시 : #JOIN (채팅방 이름) (사용자 이름)
#JOIN cnet2 peer2
peer1 : Hello, peer2 in cnet2!!
Hi, peer1 in cnet2!!
peer2 : Hi, peer1 in cnet2!!
```

- multicast port number은 9999로 동일하지만 새로운 채팅룸인 cnet2를 생성하고 message를 송/수신하는 모습
- #으로 시작하는 단어를 입력하면 전송되지 않는다.
- #EXIT를 입력하면 multicast group에서 leave되고 socket이 close됨으로써 peer program이 종료 된다.