

# 네트워크프로그래밍-9주 화상강의 자료

정인환교수

# 9주 강의

## ▶ 9주

- 중간시험 문제풀이
- 7주 과제 JavaChatServer.java 에 /list, /sleep, /wakeup, /to 추가하기

## ▶ 10주 이후 강의 스케줄, 텀프로젝트 일정

- 10주 : 게임, 텀프로젝트 제안서 제출/리뷰
- 11주 : UDP, Android, 텀프로젝트 설계서 제출/리뷰
- 12주 : Bluetooth, 텀프로젝트 초기화면 제출/리뷰
- 13주 : Android HTTP, 텀프로젝트 중간결과 제출/리뷰
- 14주 : 텀프로젝트 최종 점검
- 15주 : 대면시험 기간, 온라인 수업 휴강, 텀프로젝트 완성 기간
- 16주 : 텀프로젝트 발표 및 평가

# 네트워크프로그래밍 중간시험 문제 풀이

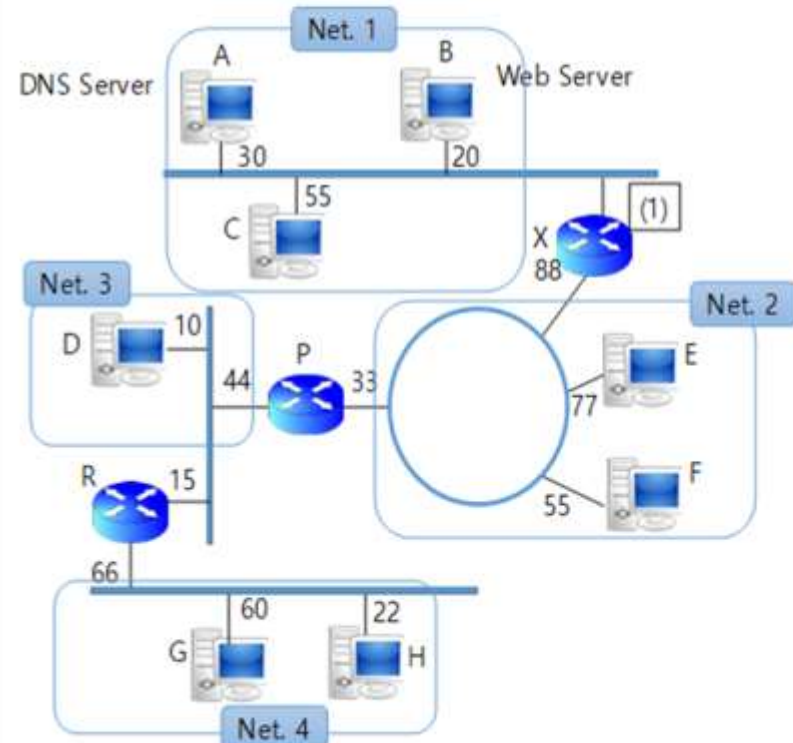
1. OSI 7 계층을 쓰고 각 계층별 대표적인 기능을 쓰시오 (한글 또는 영문 관계 없음)
2. 프로토콜이란?  
신뢰성 있는 통신을 위한 약속, 규약, 절차  
동일한 계층 간 통신을 위한 미리 정의 된 약속, 규정, 규약
3. 프로토콜 헤더의 용도는?  
프로토콜 동작을 위한 중요 정보(주소, 길이, 등등) 포함
4. TCP는 OSI 7 계층 기준 어느 계층인가? - 4
5. IP는 OSI 7 계층 기준 어느 계층인가? - 3
6. HTTP는 OSI 7 계층 기준 어느 계층인가? - 7
7. Socket을 이용한 Client 프로그램이 Server에 연결을 하기 위해 반드시 필요한 2가지 주소는?  
IP Address, Port Number
8. 프로토콜 분석기(Protocol Analyzer)의 용도를 쓰시오.  
Packet을 수집, 분석, 네트워크 프로그램의 디버깅용

▶ 다음 그림을 보고 물음에 답하시오.

단, 알파벳 A, B, C, .. X 은 논리주소(IP주소)이고 10, 20, 30 등 숫자는 물리주소(MAC주소)라고 가정한다.

A에서 G에게 Packet을 보내려고 한다. A에서 출발하는 Packet의 형태가 아래와 같다. (순서에 유의)

수신MAC	송신MAC	송신IP	수신IP	DATA
99	30	A	G	DATA



(1) X의 MAC주소는?				99
(2) A에서 G까지 어떤 순서로 전달이 되는가? (예: A->B->C->D->... 이런식으로)				A>X>P>R>G
(3) A가 G에게 보낸 Packet을 G가 수신하는 Packet은?				
수신MAC	송신MAC C	송신IP	수신IP	DATA
60	66	A	G	
(4) C가 D에게 Packet을 보낸 경우 C에서 송신하는 Packet과 D가 수신하는 Packet은?				
수신MAC	송신MAC C	송신IP	수신IP	DATA
99	55	C	D	
10	44	C	D	
(5) H가 E에게 Packet을 보내는 경우 R를 거쳐가게 된다. R이 수신하는 Packet과 송신하는 Packet은?				
수신MAC	송신MAC C	송신IP	수신IP	DATA
66	22	H	E	
44	15	H	E	
(6) G의 전원이 이제 막 On 되었다고 가정하자. G가 Web Server B의 Home Page를 보고자 Web Browser를 실행하고 B의 이름(www.hansung.ac.kr)을 주소창에 입력하였다. 이 경우 G에서 다음과 같은 단계가 선행되어야 한다. 그 이유와 용도를 설명하시오.				
ARP 동작		Router R의 MAC Address를 알기위한 동작 ARP Request / Reply		
A와 통신		www.hansung.ac.kr 을 IP 주소로 변환하기 위한 DNS Query/Response		
TCP 연결 누구와, 왜?		B www.hansung.ac.kr 과 TCP 80 Port 연결 HTTP 통신을 위해		



10. 공유기의 동작원리를 의미하는 용어는?

NAT (Network Address Translation)

11. Computer의 내부 IP 주소와 외부에서 보이는 공인 IP 주소를 확인하는 방법은?

(1) 내부 IP 주소 확인하는 방법 (어떤 명령어를 사용하면 되는지? `ipconfig`, `ip a`)

(2) 공인 IP 주소를 확인하는 방법은? (명령어나 절차를 설명..)

google 검색창 my ip address 나 mylocation.co.kr 접속

12. Windows의 `tracert` 나 Linux의 `traceroute` 의 용도는?

목적지 IP 주소까지 중간에 있는 router들의 IP를 추적할 수 있다.

13. 약자 TCP 와 UDP 의 원어를 쓰고 그 차이점을 설명하시오?

Transmission Control Protocol

User Datagram Protocol

연결형 신뢰성 있는 통신, 비 연결형

14. 다음 Socket 함수들의 함수 원형을 완성하고, 각각의 함수들은 어떤 용도로 사용이 되는지 간략하게 쓰시오.

1. `socket()`

2. `bind()`

3. `listen()`

4. `connect()`

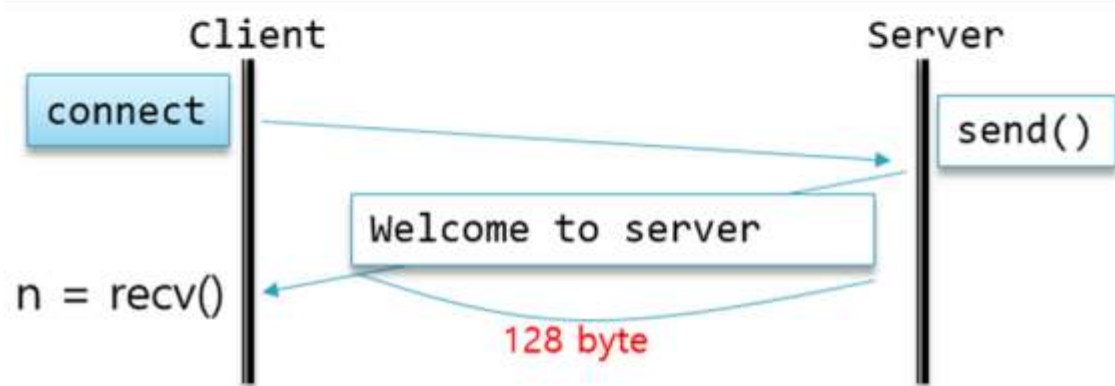
5. `accept()`

6. `send()`

7. `recv()`

15. select() 함수는 어떤 경우에 사용이 되는가?  
여러 개의 Socket의 변화를 감시
16. 프로토콜 용어 중 SAP의 원어를 쓰고 설명하십시오.  
Service Access Point 상위계층과 하위계층의 서비스 요청과 제공
17. 그래프를  $G(V, E)$ 로 표현한다면, 네트워크는 같은 방식으로  $N(N, L)$ 로 표현할 수 있다. N과 L이 의미하는 것은? - Node, Link
18. TCP 연결설정 단계를 무엇이라 부르며 그 절차를 설명하십시오  
3way Handshaking, SYN, ACK+SYN, ACK
19. 다음 각 프로토콜 헤더에는 주소가 있다. 몇 Byte의 어떤 주소를 가지고 있는지 쓰시오.  
(1) Ethernet : MAC Address 6 bytes  
(2) IP : IP Address 4 bytes  
(3) TCP : Port Number 2 bytes
20. 다음 용어들의 원어를 쓰시오. (예: ISO International Standard Organization)  
(1) OSI : Open System Interconnection  
(2) IP : Internet Protocol  
(3) DNS : Domain Name Service  
(4) HTTP : Hyper Text Transfer Protocol  
(5) MAC : Medium Access Control
21. 다음 명령어들의 용도를 쓰시오.  
(1) Windows ipconfig 또는 Linux ip a : 컴퓨터의 IP 주소 확인  
(2) Netstat -na : 컴퓨터의 Network 연결 상태를 모두 표시  
(3) Netstat -nr : 컴퓨터의 Routing Table 을 표시

15. 다음 그림은 client 가 server에 접속하고 Welcome to Server!! 메시지를 받을때 까지 과정을 Wireshark으로 캡쳐한 것이다. 그림을 보고 질문에 답하시오.



	Length	Info
(1)	74	39740 → 30000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_P
(2)	66	30000 → 39740 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1
(3)	60	39740 → 30000 [ACK] Seq=1 Ack=1 Win=64256 Len=0
(4)	182	30000 → 39740 [PSH, ACK] Seq=1 Ack=1 Win=1051136 Len=128
(5)	60	39740 → 30000 [ACK] Seq=1 Ack=129 Win=64128 Len=0

(1) Packet 번호 (1), (2), (3) 은 어떤 과정인지 설명하시오.

- TCP 연결설정, Three Way Handshaking

(2) client 가 server에 접속하면 server는 "Welcome to Server!!" 를 128 byte 길이의 변수(char buf[])에 담아서 "Welcome to Server!!" 를 제외한 뒷부분은 모두 NULL(\0)로 채워서 보낸다. 그 부분을 C 언어로 작성하면 다음과 같다. 코드를 완성하시오.

```
memset(buf, 0, 128);
sprintf(buf, "%s", "Welcome to Server!!");
send(client_fd, buf, 128, 0);
```

(3) "Welcome to Server!!" 전송을 Wireshark에서 잡은 4번째 packet의 전체 길이 Length는 182로 나오고 Data Len=128이다. Data Len 는 128인데 전체 길이가 182로 나오는 이유를 자세히 설명하시오.

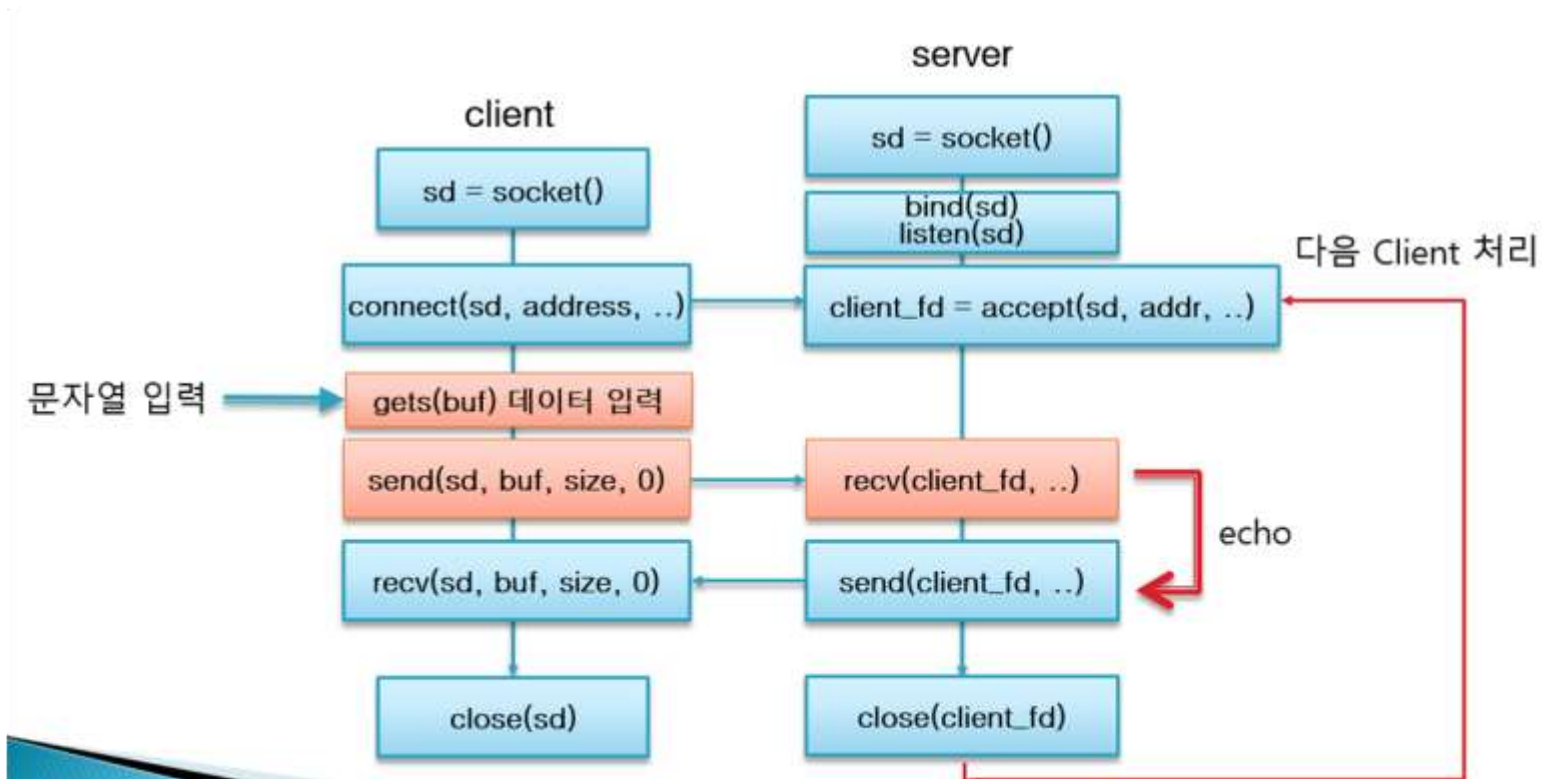
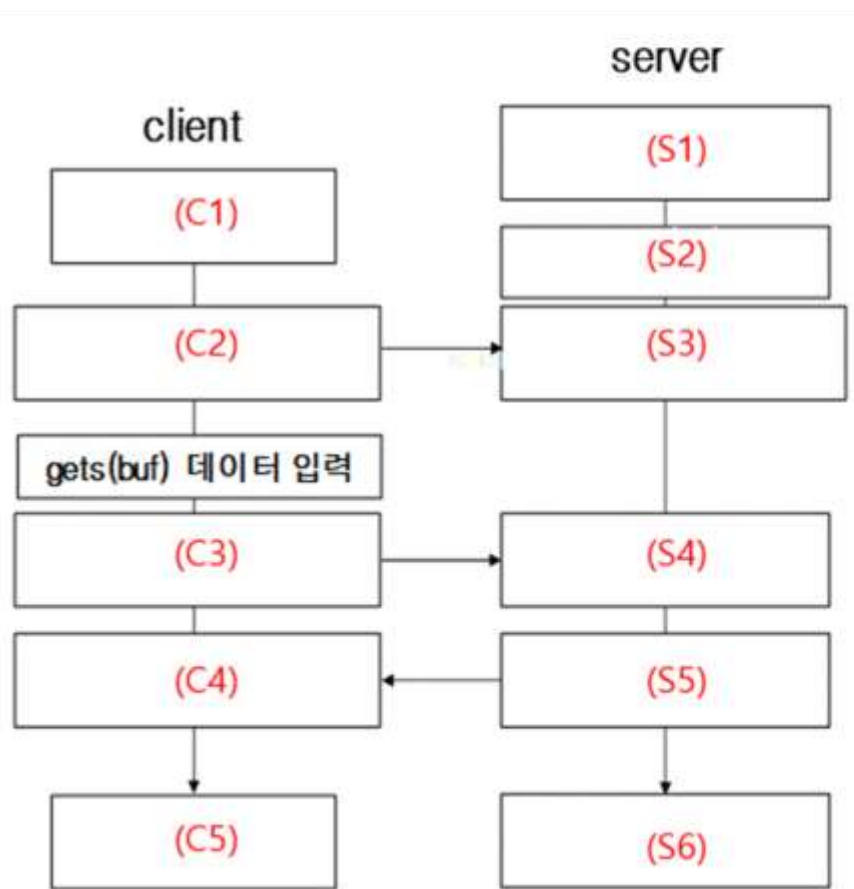
Data len 128 + Ethernet Header 14 + IP Header 20 + TCP Header 20

(4) client에서 `n = recv(s, buf, 256, 0);` 라고 가정하면 n 값은 얼마로 return 되는가?

128 (server가 128만 보냈으므로)



15. 다음 Flow Chart는 Socket API를 사용한 Echo Client / Server 프로그램 흐름이다. client 는 한번 문장 입력을 받고 Server에 전송하고, Server는 받은 문장을 그대로 Echo 해준다. Socket 함수를 이용해 빈칸을 채우시오.



24. 공유기의 Port Forwarding 은 어떤 용도로 사용이 되는지 설명하시오.

특정 Port 로 들어오는 것을 내부 특정 IP로 전달하는 용도

25. VMware 네트워크 설정에서 NAT 모드와 Bridged 모드를 설명하시오.

NAT 모드 : Window가 공유기 역할을 하며 Vmware는 Windows로 부터 IP 할당

Bridged 모드 : Vmware <> Windows <> 공유기 환경, Vmware는 공유기로 부터 IP 할당.

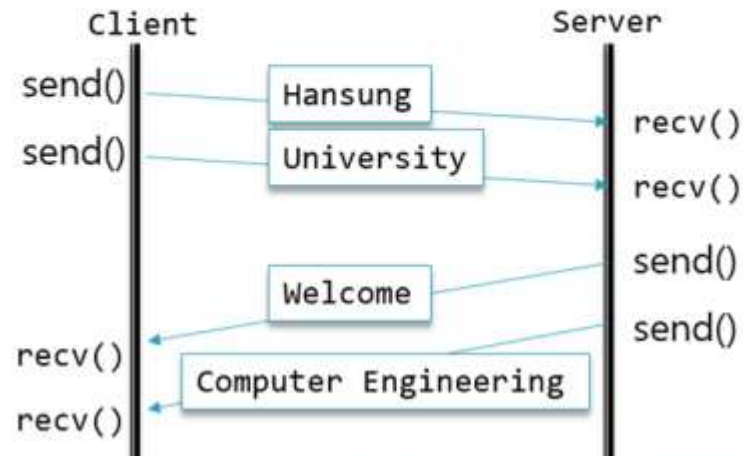
26. Wireshark 에서 HTTP 데이터만 분석하려면 어떤 필터를 써야 하는가?

tcp and port 80

27. (문제27) Wireshark 에서 DNS 데이터만 분석하려면 어떤 필터를 써야 하는가?

udp and port 53

15. 아래 화면은 client/server 흐름과 결과 화면이다. 그리고 주어진 코드는 미완성 프로그램으로 그대로 실행하면 화면과 같이 나오지 않는다.



C:\Windows\system32\cmd.exe

```

Connecting 127.0.0.1 30000
Welcome
Computer Engineering
계속하려면 아무 키나 누르십시오

```

C:\Windows\system32\cmd.exe

```

echo_server1 waiting connection..
server_fd = 208
Server : waiting connection request.
Client connected from 127.0.0.1:11189
client_fd = 216
Hansung
University
계속하려면 아무 키나 누르십시오 . . .

```

1) 화면과 같이 안 나오고 어떤 문제가 발생할 수 있는지 설명하시오.  
(필요하면 위의 코드를 client/server에서 실행하고 결과를 보고 설명하여도 됨)

HansungUniversity 가 불어날 수 있다.

str2 보낼때 문제가 발생할 수 있다.

welcome 보내면 client가 welcome 뒤에 있는 데이터까지 받을 수 있다.

(2) 화면과 같이 나오도록 문제점을 해결한 코드로 수정하시오. (변수 추가하거나 함수 사용하여도 됨)

client	server
<pre> char buf[129]; char* str1 = "Hansung"; char* str2 = "University"; memset(buf, 0, 128); sprintf(buf, str1); send(s, buf, 128, 0); memset(buf, 0, 128); sprintf(buf, str2); send(s, buf, 128, 0); recv(s, buf, 128, 0); printf("%s\n", buf); recv(s, buf, 128, 0); printf("%s\n", buf); </pre>	<pre> char buf[129]; char* welcome = "Welcome"; char* cs = "Computer Engineering"; recv(client_fd, buf, 128, 0); printf("%s\n", buf); recv(client_fd, buf, 128, 0); printf("%s\n", buf); memset(buf, 0, 128); sprintf(buf, welcome); send(client_fd, buf, 128, 0); send(client_fd, cs, strlen(cs), 0); </pre>

client	server
<pre> char buf[129]; char* str1 = "Hansung"; char* str2 = "University"; send(s, str1, strlen(str1), 0); send(s, str2, 128, 0); recv(s, buf, 128, 0); printf("%s\n", buf); recv(s, buf, 128, 0); printf("%s\n", buf); </pre>	<pre> char buf[129]; char* welcome = "Welcome"; char* cs = "Computer Engineering"; recv(client_fd, buf, 128, 0); printf("%s\n", buf); recv(client_fd, buf, 128, 0); printf("%s\n", buf); send(client_fd, welcome, strlen(welcome), 0); send(client_fd, cs, strlen(cs), 0); </pre>

## 7주 과제2 : Win chat\_server4 기능 JavaChatServer에 구현하기

- ▶ Client는 추가 기능 없음
- ▶ Windows chat\_server4
  - 입장/퇴장 알리기
  - /exit, /list, /to 귓속말, /sleep, /wakeup
- ▶ 입장/퇴장 알리기
  - 입장 : AcceptServer Accept하고, UserService() 생성되면 생성자에서 처리
    - UserName = msg[1].trim() → UserName을 이용 WriteAll() 하면 됨
  - 퇴장 : UserVec.removeElement(this); 하기 전에 퇴장 알림 처리.
- ▶ /exit, /list, /to
  - dis.read() 다음 처리, UserVec.RemoveElement(this); 로 제거
  - JavaChatServer의 class UserService {}
  - private → public String UserName 변수를 이용
    - /list 목록을 보여주고
    - /to 상대를 찾는다.
- ▶ /sleep, /wakeup
  - JavaChatServer의 class UserService {}
  - public String UserStatus 변수 추가



# JavaChatServer.java

```
public void Login() {
    AppendText("새로운 참가자 " + UserName + " 입장.");
    WriteOne("Welcome to Java chat server\n");
    WriteOne(UserName + "님 환영합니다.\n"); // 연결된 사용자에게 정상접속을 알림
    String msg = "[" + UserName + "]"님이 입장 하였습니다.\n";
    WriteAll(msg); // 아직 user_vc에 새로 입장한 user는 포함되지 않았다.
}

public void Logout() {
    String msg = "[" + UserName + "]"님이 퇴장 하였습니다.\n";
    UserVec.removeElement(this); // Logout한 현재 객체를 벡터에서 지운다
    WriteAll(msg); // 나를 제외한 다른 User들에게 전송
    AppendText("사용자 " + "[" + UserName + "]" 퇴장. 현재 참가자 수 " + UserVec.size());
}

// 모든 User들에게 방송. 각각의 UserService Thread의 WriteOne() 을 호출한다.
public void WriteAll(String str) {
    for (int i = 0; i < user_vc.size(); i++) {
        UserService user = (UserService) user_vc.elementAt(i);
        if (user.UserStatus == "0")
            user.WriteOne(str);
    }
}
```

```
// User 당 생성되는 Thread
// Read One 에서 대기 -> Write All
class UserService extends Thread {
    private InputStream is;
    private OutputStream os;
    private DataInputStream dis;
    private DataOutputStream dos;
    private Socket client_socket;
    private Vector user_vc;
    public String UserName = "";
    public String UserStatus;

    public UserService(Socket client_socket) {
        // TODO Auto-generated constructor stub
        // 매개변수로 넘어온 자료 저장
        this.client_socket = client_socket;
        this.user_vc = UserVec;
        try {
            is = client_socket.getInputStream();
            dis = new DataInputStream(is);
            os = client_socket.getOutputStream();
            dos = new DataOutputStream(os);
            // line1 = dis.readUTF();
            // /login user1 ==> msg[0] msg[1]
            byte[] b = new byte[BUF_LEN];
            dis.read(b);
            String line1 = new String(b);
            String[] msg = line1.split(" ");
            UserName = msg[1].trim();
            UserStatus = "0"; // Online 상태
            Login();
        } catch (Exception e) {
            AppendText("userService error");
        }
    }
}
```

# JavaChatServer.java

```
public void run() {
    while (true) { // 사용자 접속을 계속해서 받기 위해 while문
        try {
            // String msg = dis.readUTF();
            byte[] b = new byte[BUF_LEN];
            int ret;
            ret = dis.read(b);
            if (ret < 0) {
                AppendText("dis.read() < 0 error");
                try {
                    dos.close();
                    dis.close();
                    client_socket.close();
                    Logout();
                    break;
                } catch (Exception ee) {
                    break;
                } // catch문 끝
            }
            String msg = new String(b, "euc-kr");
            msg = msg.trim(); // 앞뒤 blank NULL, \n 모두 제거
            AppendText(msg); // server 화면에 출력
        }
    }
}
```

```
String[] args = msg.split(" "); // 단어들을 분리한다.
if (args.length == 1) { // Enter key 만 들어온 경우 Wakeup 처리만 한다.
    UserStatus = "0";
} else if (args[1].matches("/exit")) {
    Logout();
    break;
} else if (args[1].matches("/list")) {
    WriteOne("User list\n");
    WriteOne("Name\tStatus\n");
    WriteOne("-----\n");
    for (int i = 0; i < user_vc.size(); i++) {
        UserService user = (UserService) user_vc.elementAt(i);
        WriteOne(user.UserName + "\t" + user.UserStatus + "\n");
    }
    WriteOne("-----\n");
} else if (args[1].matches("/sleep")) {
    UserStatus = "S";
} else if (args[1].matches("/wakeup")) {
    UserStatus = "0";
} else if (args[1].matches("/to")) { // 귓속말
    for (int i = 0; i < user_vc.size(); i++) {
        UserService user = (UserService) user_vc.elementAt(i);
        if (user.UserName.matches(args[2]) && user.UserStatus.matches("0")) {
            String msg2 = "";
            for (int j = 3; j < args.length; j++) { // 실제 message 부분
                msg2 += args[j];
                if (j < args.length - 1)
                    msg2 += " ";
            }
            user.WriteOne("[귓속말] " + args[0] + " " + msg2 + "\n");
            break;
        }
    }
} else { // 일반 채팅 메시지
    UserStatus = "0";
    WriteAll(msg + "\n"); // Write All
}
```