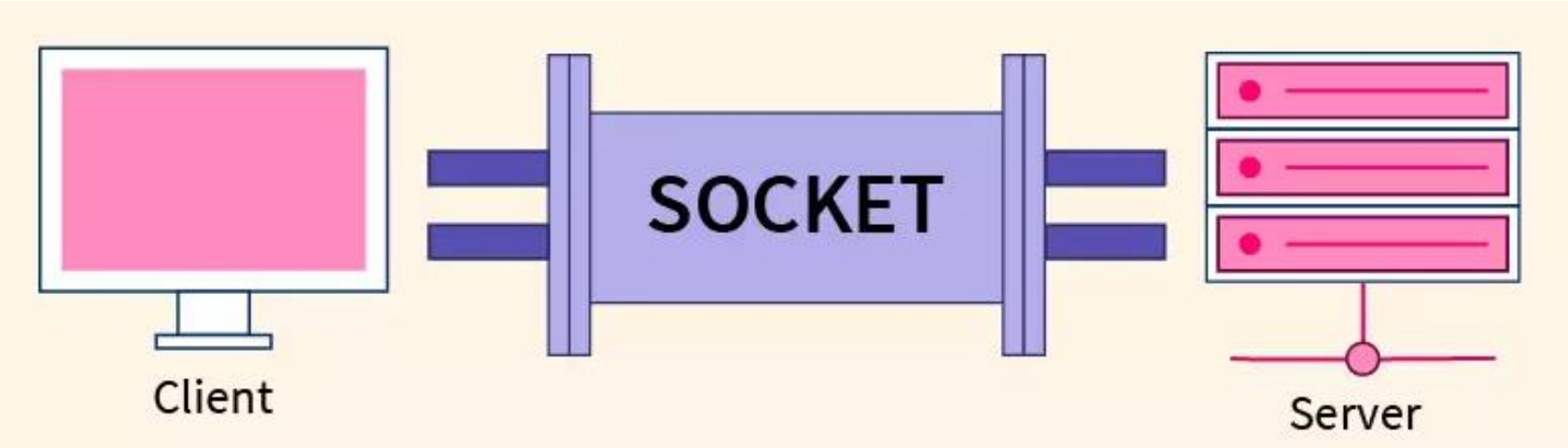
야매로 서버 개발자 되는 법 2강

소켓 프로그래밍

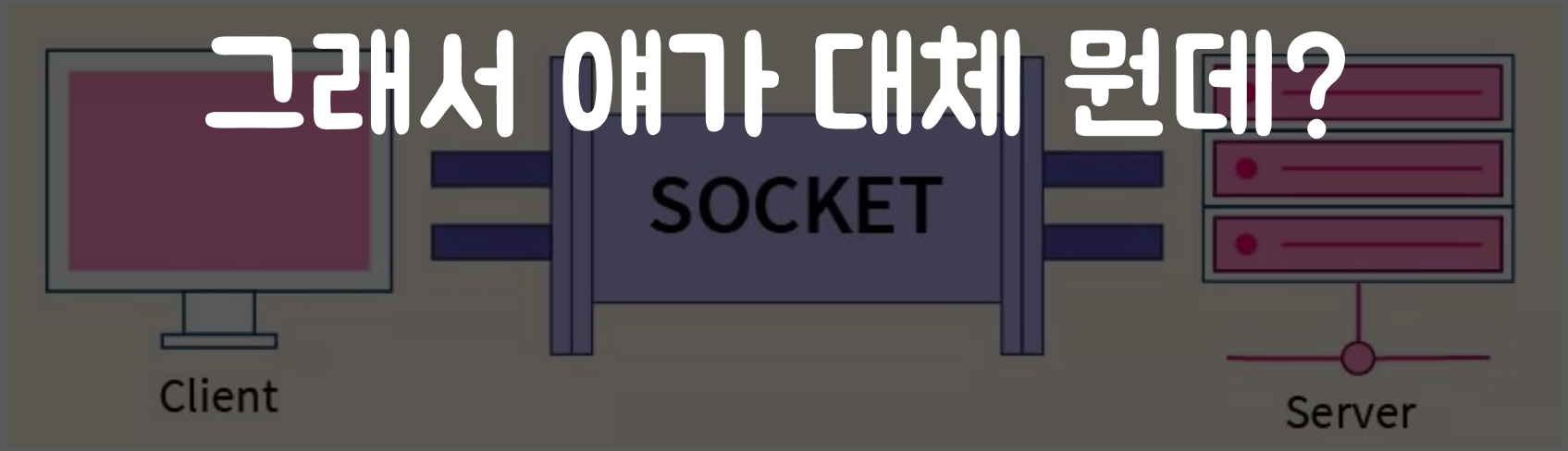
소켓이 뭐에요?

소켓

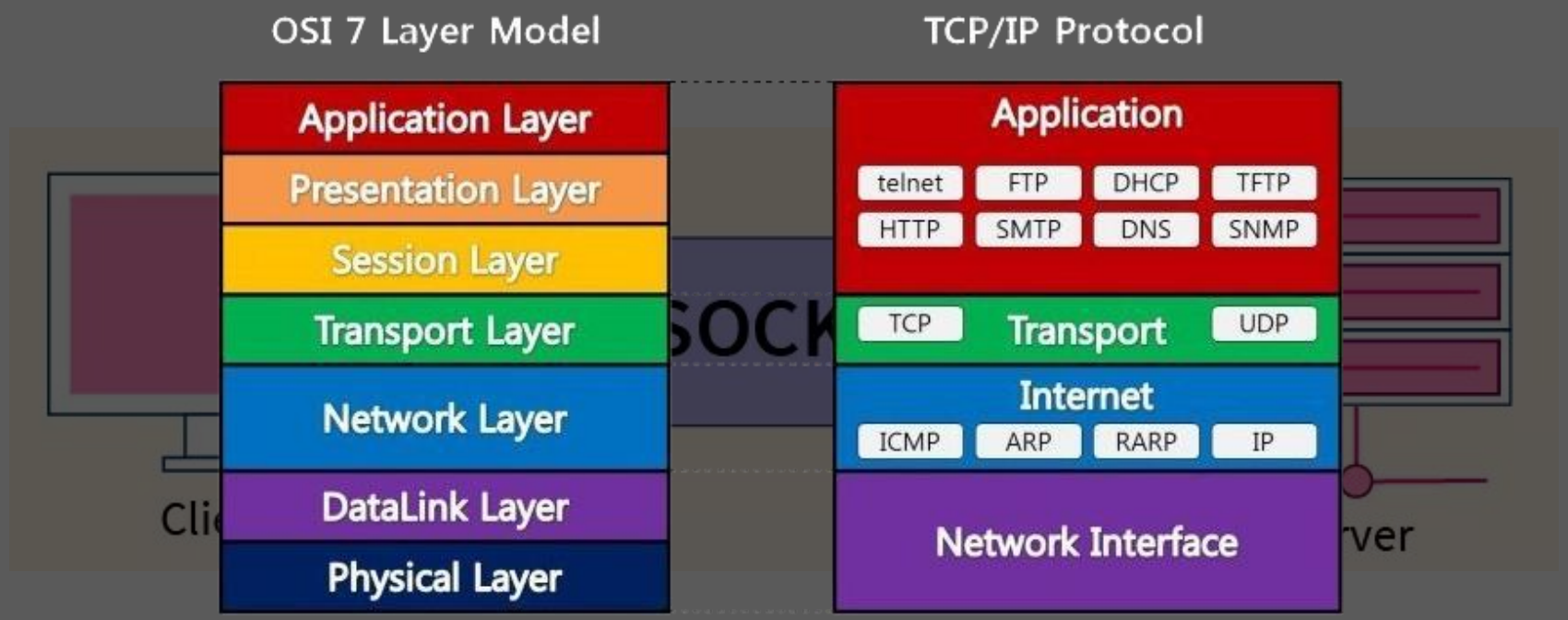


소켓

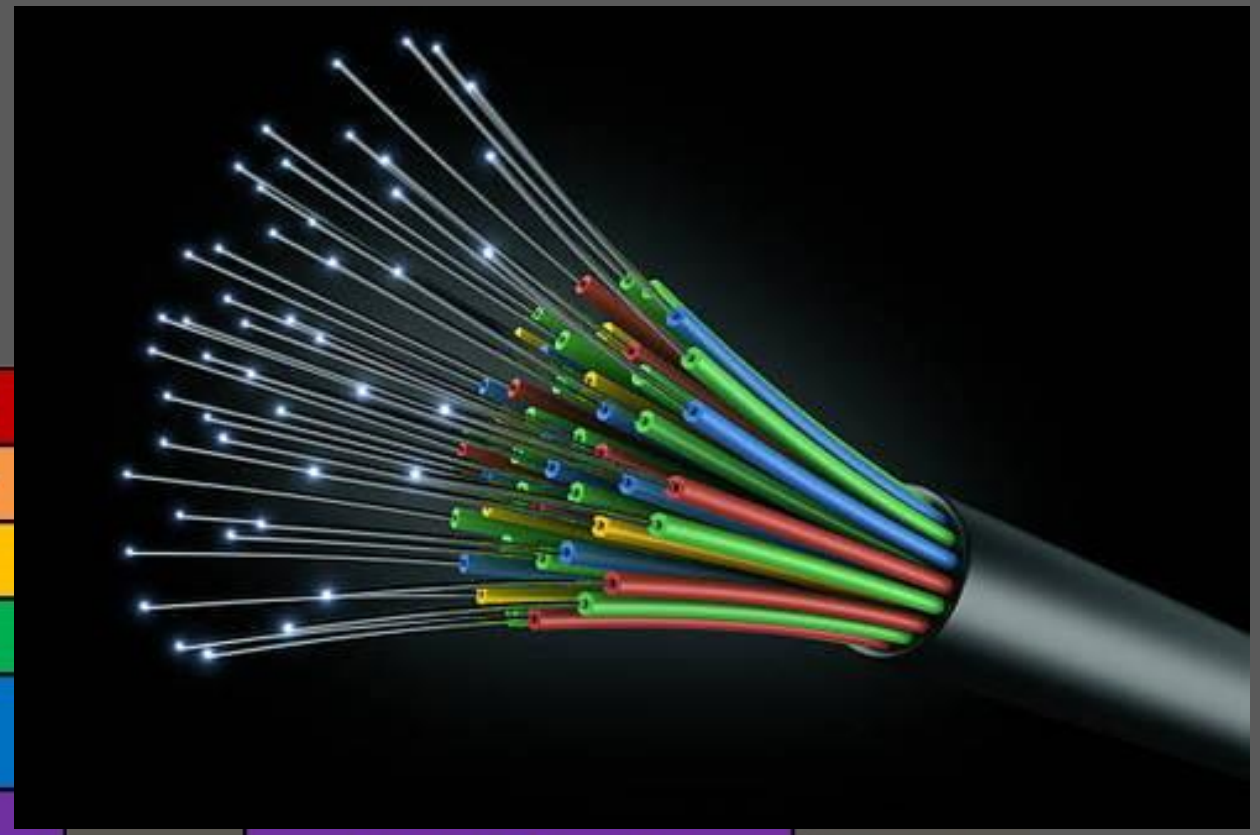
그래서 얘가 대체 뭔데?



소켓


OSI 7 Layer Model
TCP/IP Protocol
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
DataLink Layer
Physical Layer
Application
telnet
FTP
DHCP
TFTP
HTTP
SMTP
DNS
SNMP
TCP
Transport
UDP
Internet
ICMP
ARP
RARP
IP
Network Interface

소켓



OSI 7 Layer Model

Application Layer

Presentation Layer

Session Layer

Transport Layer

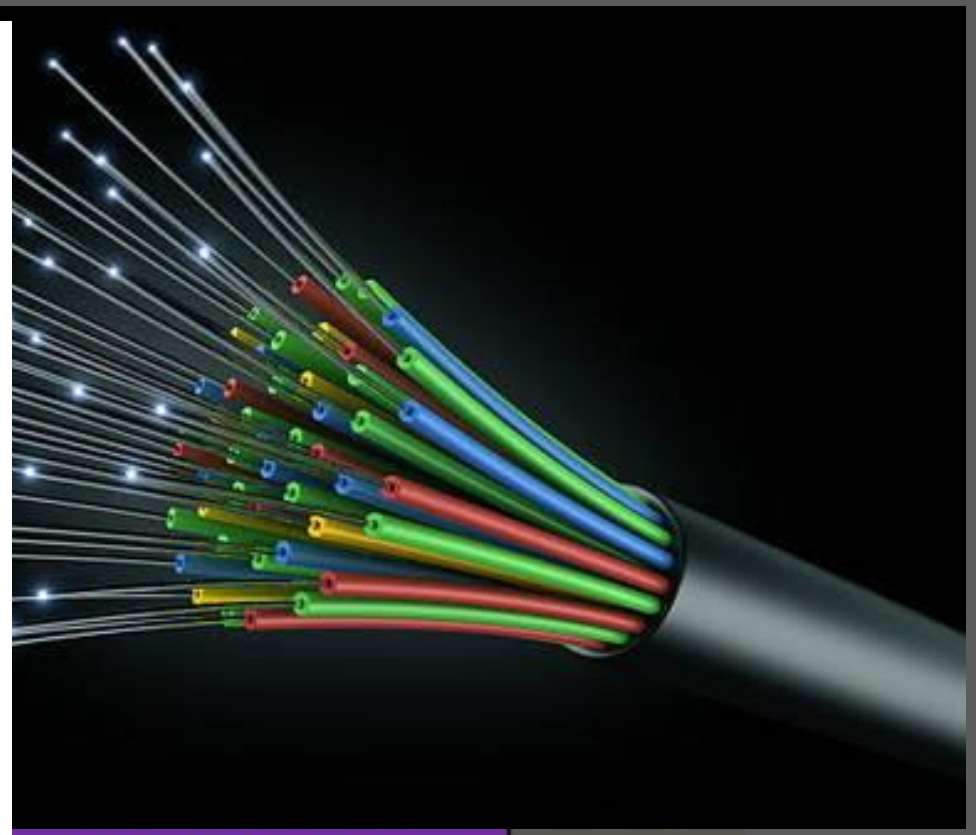
Network Layer

DataLink Layer

Physical Layer

Network Interface

소켓



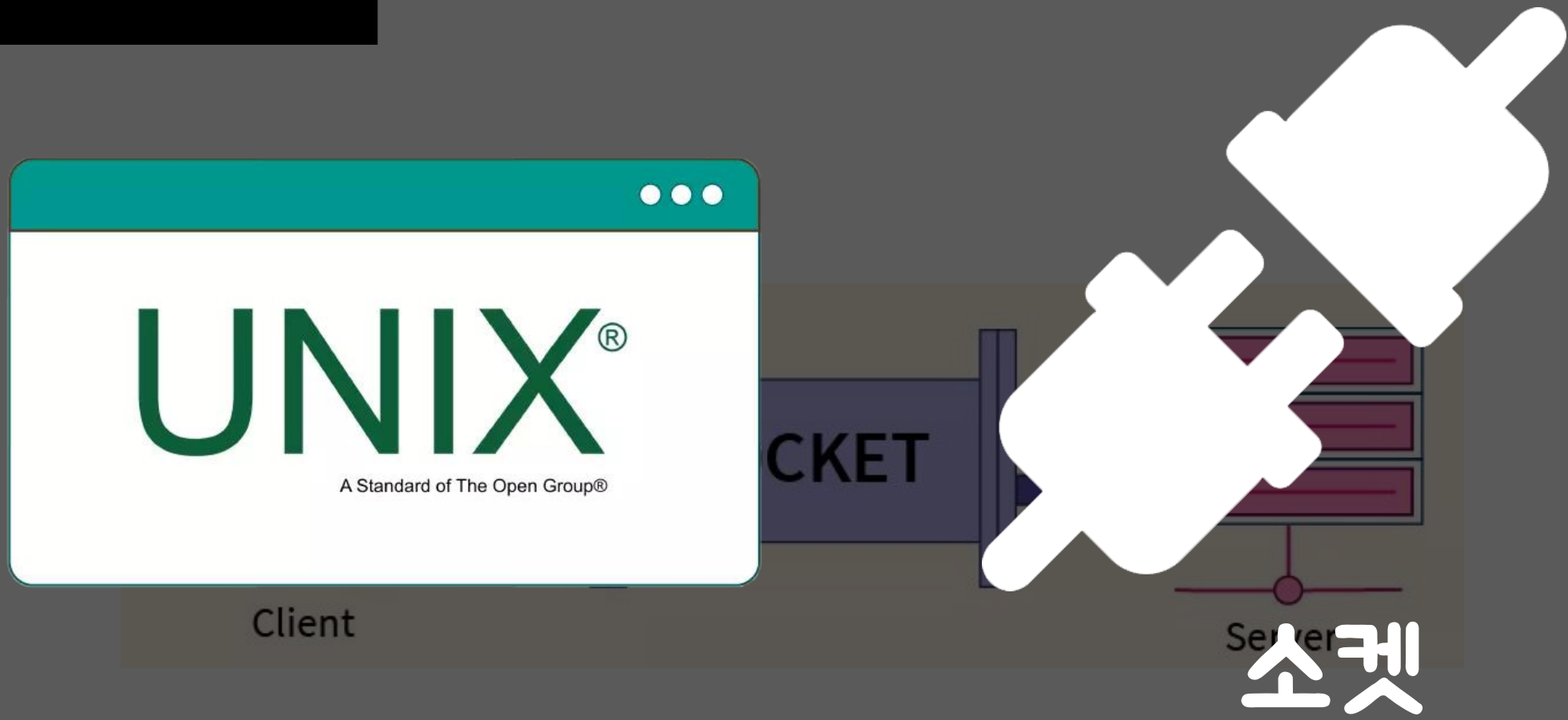
Network Interface

rver

소켓


symbian OS
iOS
AMIGA OS
OS/2 WARP
BlackBerry
Windows 10
android
MS DOS
Mac OS
Linux
solaris
AIX
ubuntu
UNIX
TIZEN

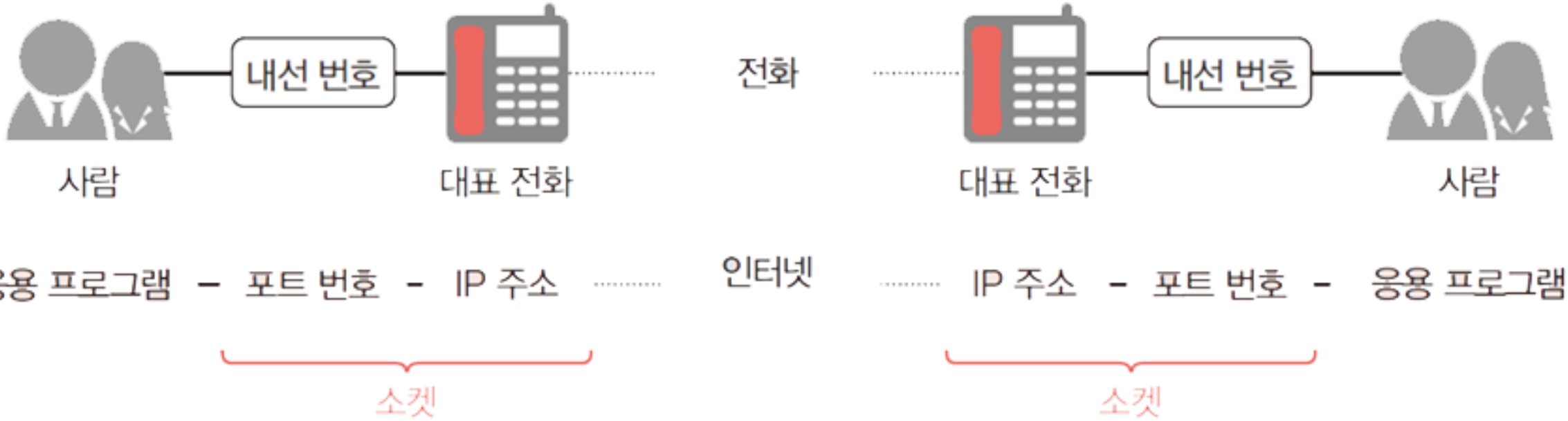
소켓



소켓을 어떻게 써요?

소켓

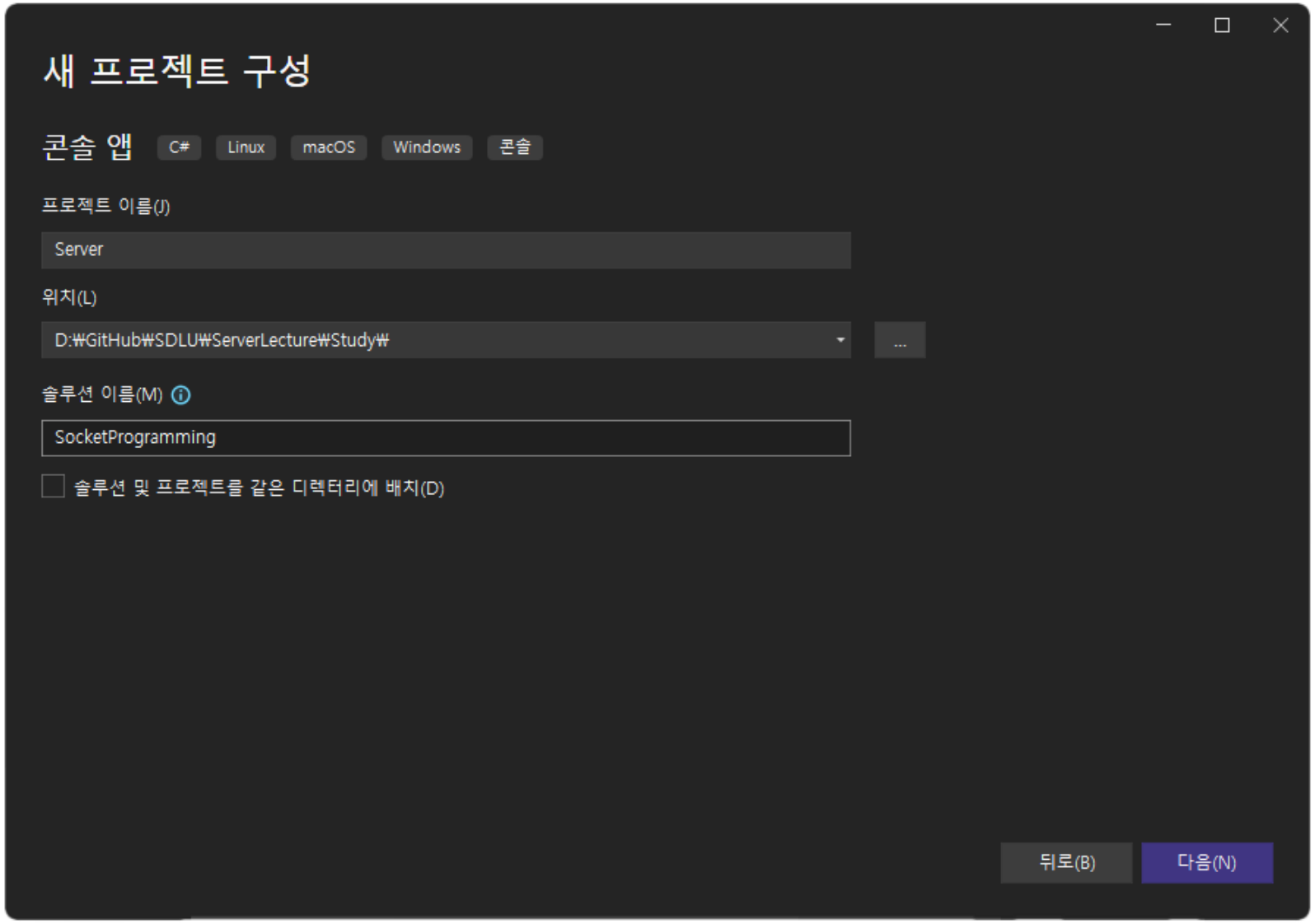
소켓


내선 번호
전화
내선 번호
사람
대표 전화
대표 전화
사람
응용 프로그램 - 포트 번호 - IP 주소
인터넷
IP 주소 - 포트 번호 - 응용 프로그램
소켓
소켓

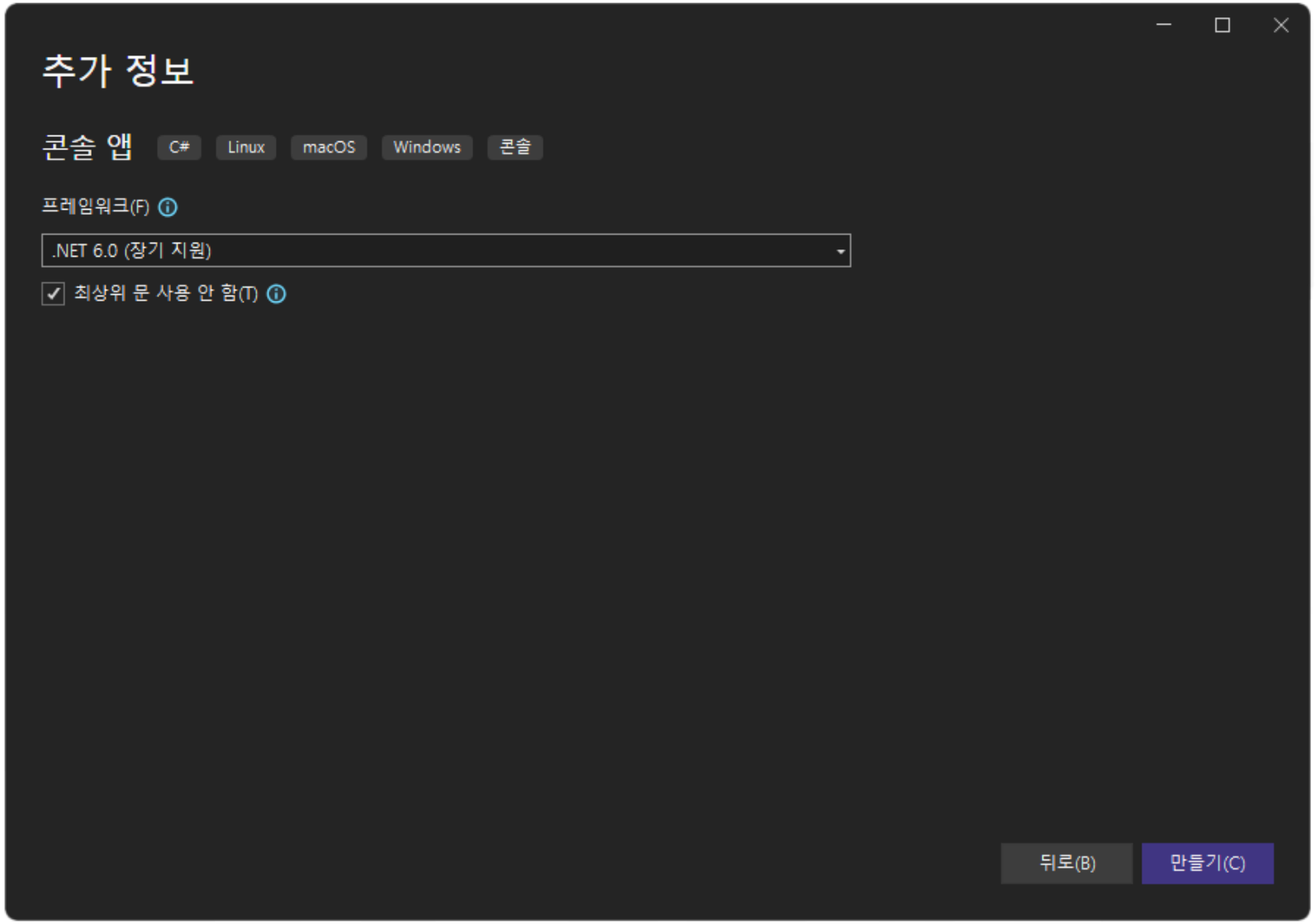
소켓 실습

프로젝트 생성

프로젝트 생성

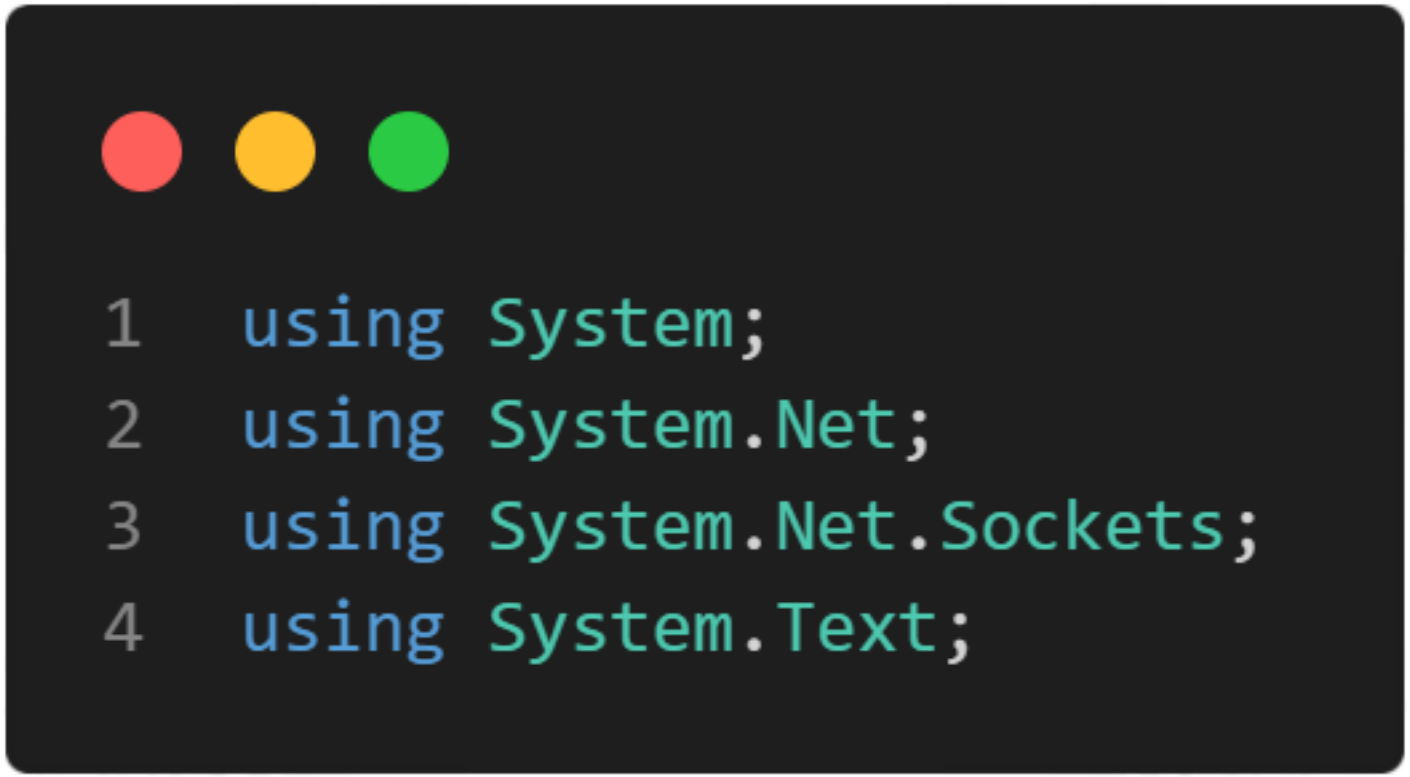

새 프로젝트 구성
콘솔 앱
C#
Linux
macOS
Windows
콘솔
프로젝트 이름(J)
Server
위치(L)
D:₩GitHub₩SDLU₩ServerLecture₩Study₩
...
솔루션 이름(M)
SocketProgramming
솔루션 및 프로젝트를 같은 디렉터리에 배치(D)
뒤로(B)
다음(N)

프로젝트 생성


추가 정보
콘솔 앱
C#
Linux
macOS
Windows
콘솔
프레임워크(F)
.NET 6.0 (장기 지원)
최상위 문 사용 안 함(T)
뒤로(B)
만들기(C)

소켓 서버

라이브러리



```
1   using System;
2   using System.Net;
3   using System.Net.Sockets;
4   using System.Text;
```

리슨 소켓

```
1  private static Socket CreateListenSocket()
2  {
3      string host = Dns.GetHostName();
4      IPHostEntry ipHost = Dns.GetHostEntry(host);
5      IPAddress ipAddress = ipHost.AddressList[1];
6      IPEndPoint endPoint = new IPEndPoint(ipAddress, 8081);
7
8      Socket socket = new Socket(endPoint.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
9      socket.Bind(endPoint);
10     socket.Listen(1);
11
12     Console.WriteLine($"Server opened on port : {endPoint.Port}");
13
14     return socket;
15 }
```

소켓 연결

```
1  static void Main(string[] args)
2  {
3      Socket listenSocket = CreateListenSocket();
4      Socket clientSocket = listenSocket.Accept();
5
6      Console.WriteLine("Client joined the server");
7
8      // 통신하기
9
10     listenSocket.Close();
11     Console.WriteLine("Server closed");
12 }
```

통신

```
1  private static bool Communication(Socket clientSocket)
2  {
3      try
4      {
5          byte[] buffer = new byte[1024];
6          int receivedSize = clientSocket.Receive(buffer);
7
8          string receivedMessage = Encoding.UTF8.GetString(buffer, 0, receivedSize);
9          IPEndPoint clientEndPoint = (clientSocket.RemoteEndPoint as IPEndPoint);
10         Console.WriteLine($"MESSAGE FROM {clientEndPoint.Address} : {receivedMessage}");
11
12         if (receivedMessage.IndexOf("exit") > -1)
13             return false;
14
15         string echoMessage = $"SERVER MESSAGE : {receivedMessage}";
16         byte[] echoBytes = Encoding.UTF8.GetBytes(echoMessage);
17
18         clientSocket.Send(echoBytes);
19
20         return true;
21     }
22     catch (Exception err)
23     {
24         Console.WriteLine(err.Message);
25         return false;
26     }
27 }
```

통신

```
1  static void Main(string[] args)
2  {
3      Socket listenSocket = CreateListenSocket();
4      Socket clientSocket = listenSocket.Accept();
5
6      Console.WriteLine("Client joined the server");
7
8      while (true)
9      {
10         bool isSuccess = Communication(clientSocket);
11
12         if (isSuccess == false)
13         {
14             clientSocket.Shutdown(SocketShutdown.Both);
15             clientSocket.Close();
16             Console.WriteLine("Disconnected with client");
17
18             break;
19         }
20     }
21
22     listenSocket.Close();
23     Console.WriteLine("Server closed");
24 }
```

소켓 클라이언트

서버 소켓

```
1  private static Socket CreateServerSocket(out IPEndPoint endPoint)
2  {
3      string host = Dns.GetHostName();
4      IPHostEntry ipHost = Dns.GetHostEntry(host);
5      IPAddress ipAddress = ipHost.AddressList[1];
6      endPoint = new IPEndPoint(ipAddress, 8081);
7
8      Socket socket = new Socket(endPoint.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
9
10     return socket;
11 }
```

서버 연결

```
1  static void Main(string[] args)
2  {
3      Socket serverSocket = CreateServerSocket(out IPEndPoint endPoint);
4      serverSocket.Connect(endPoint);
5
6      Console.WriteLine("Success to join server");
7
8      // 통신하기
9  }
```

통신

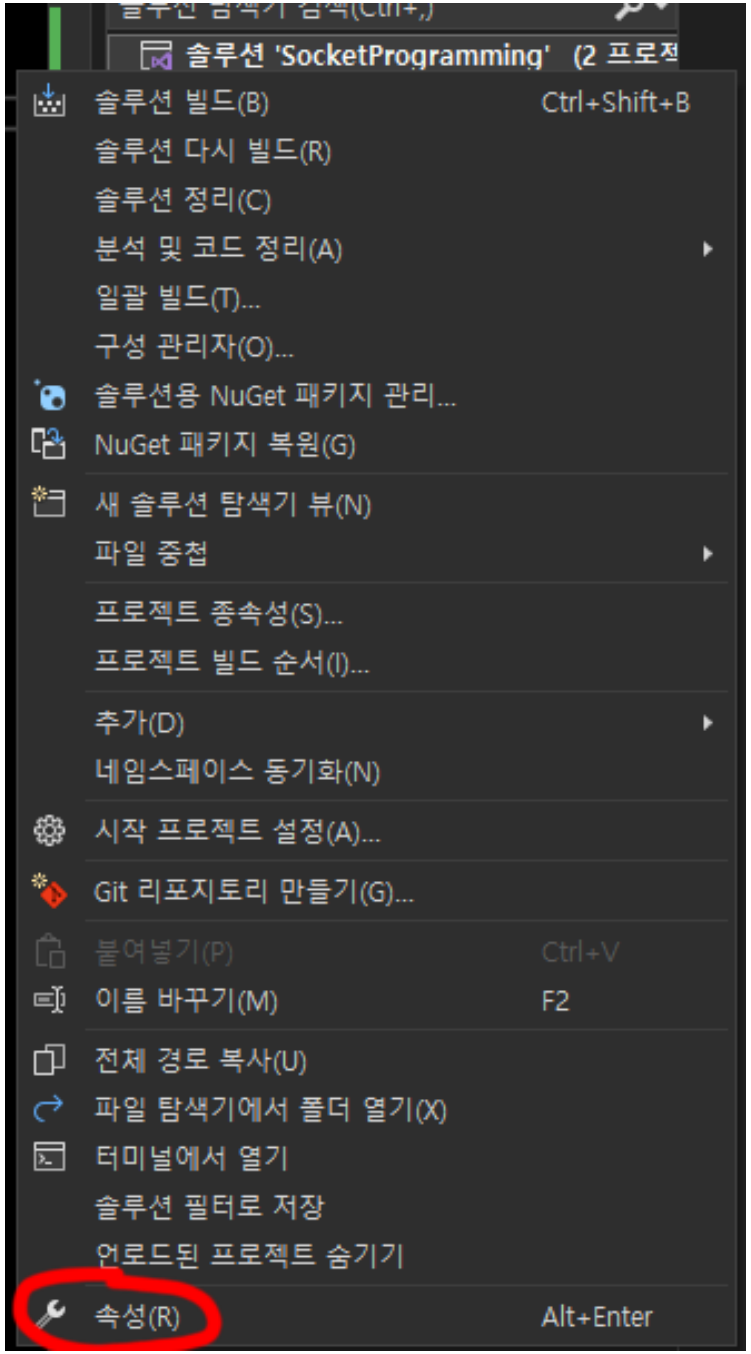
```
1  private static bool Communication(Socket serverSocket)
2  {
3      try
4      {
5          string message = Console.ReadLine();
6          byte[] messageBytes = Encoding.UTF8.GetBytes(message);
7
8          serverSocket.Send(messageBytes);
9          Console.WriteLine($"SENT MESSAGE : {message}");
10
11         if (message.IndexOf("exit") > -1)
12             return false;
13
14         byte[] buffer = new byte[1024];
15         int receivedSize = serverSocket.Receive(buffer);
16
17         string receivedMessage = Encoding.UTF8.GetString(buffer, 0, receivedSize);
18         Console.WriteLine(receivedMessage);
19
20         return true;
21     }
22     catch (Exception err)
23     {
24         Console.WriteLine(err.Message);
25         return false;
26     }
27 }
```

통신

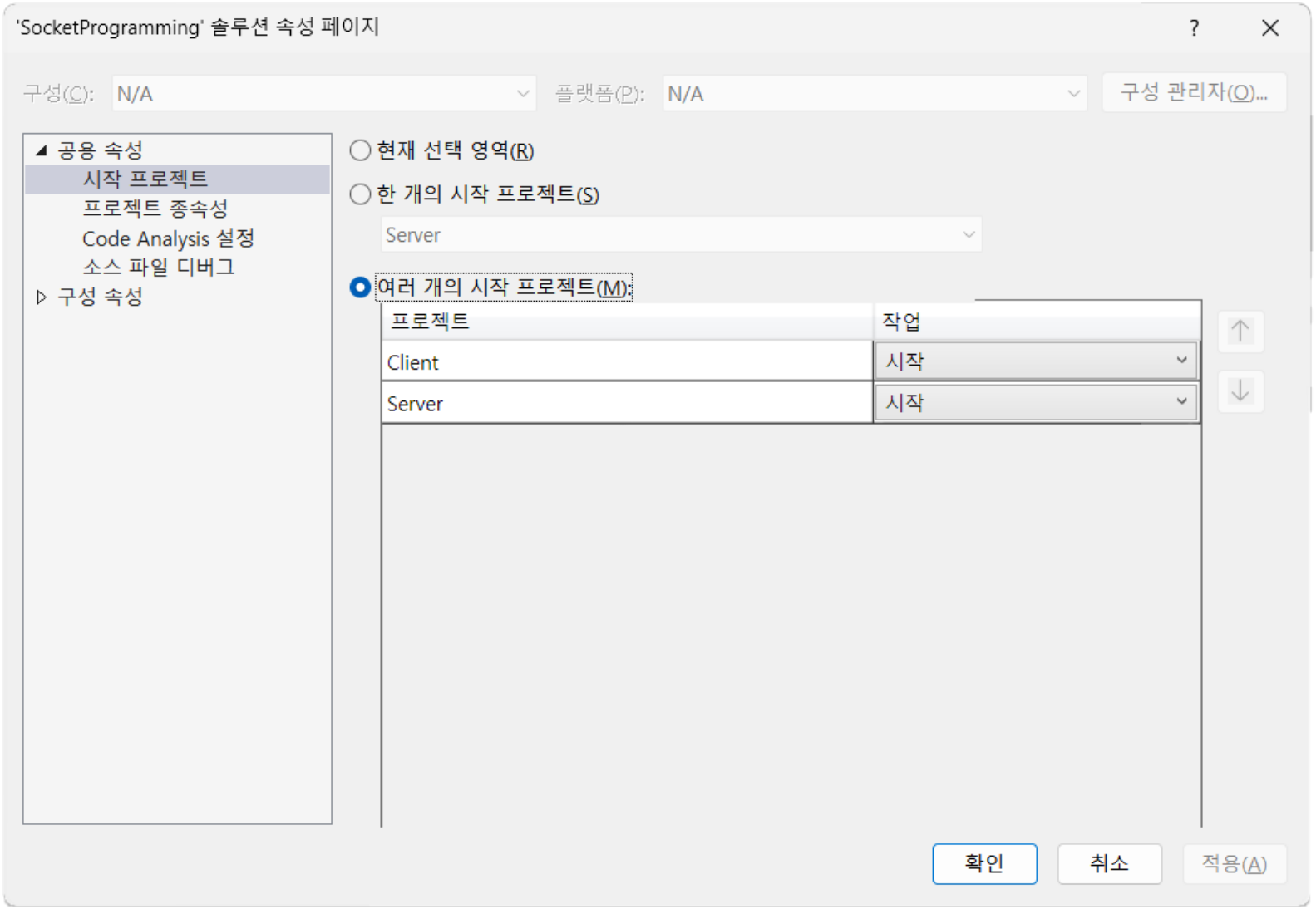
```
1  static void Main(string[] args)
2  {
3      Socket serverSocket = CreateServerSocket(out IPEndPoint endPoint);
4      serverSocket.Connect(endPoint);
5
6      Console.WriteLine("Success to join server");
7
8      while (true)
9      {
10         bool isSuccess = Communication(serverSocket);
11
12         if (isSuccess == false)
13         {
14             serverSocket.Shutdown(SocketShutdown.Both);
15             serverSocket.Close();
16             Console.WriteLine("Disconnected with server");
17
18             break;
19         }
20     }
21 }
```

디버깅

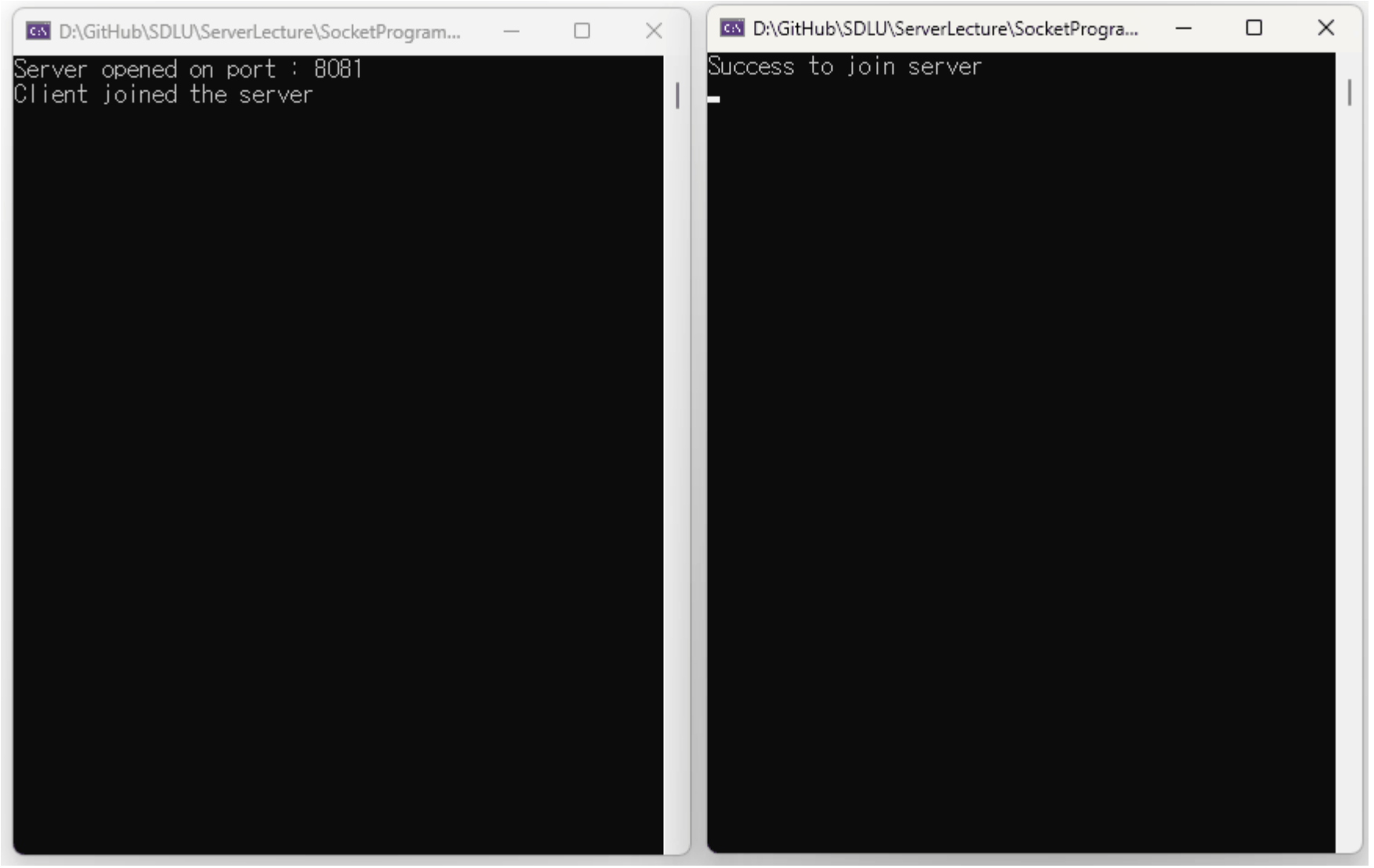
프로젝트 실행


솔루션 'SocketProgramming' (2 프로젝
솔루션 빌드(B) Ctrl+Shift+B
솔루션 다시 빌드(R)
솔루션 정리(C)
분석 및 코드 정리(A)
일괄 빌드(T)...
구성 관리자(O)...
솔루션용 NuGet 패키지 관리...
NuGet 패키지 복원(G)
새 솔루션 탐색기 뷰(N)
파일 중첩
프로젝트 종속성(S)...
프로젝트 빌드 순서(I)...
추가(D)
네임스페이스 동기화(N)
시작 프로젝트 설정(A)...
Git 리포지토리 만들기(G)...
붙여넣기(P) Ctrl+V
이름 바꾸기(M) F2
전체 경로 복사(U)
파일 탐색기에서 폴더 열기(X)
터미널에서 열기
솔루션 필터로 저장
언로드된 프로젝트 숨기기
속성(R) Alt+Enter

프로젝트 실행


'SocketProgramming' 솔루션 속성 페이지
구성(C): N/A
플랫폼(P): N/A
구성 관리자(O)...
공용 속성
시작 프로젝트
프로젝트 종속성
Code Analysis 설정
소스 파일 디버그
구성 속성
현재 선택 영역(R)
한 개의 시작 프로젝트(S)
Server
여러 개의 시작 프로젝트(M):
프로젝트
작업
Client
시작
Server
시작
확인
취소
적용(A)

프로젝트 실행


D:\GitHub\SDLU\ServerLecture\SocketProgram...
Server opened on port : 8081
Client joined the server
D:\GitHub\SDLU\ServerLecture\SocketProgra...
Success to join server

나도 이제 서버개발자?!

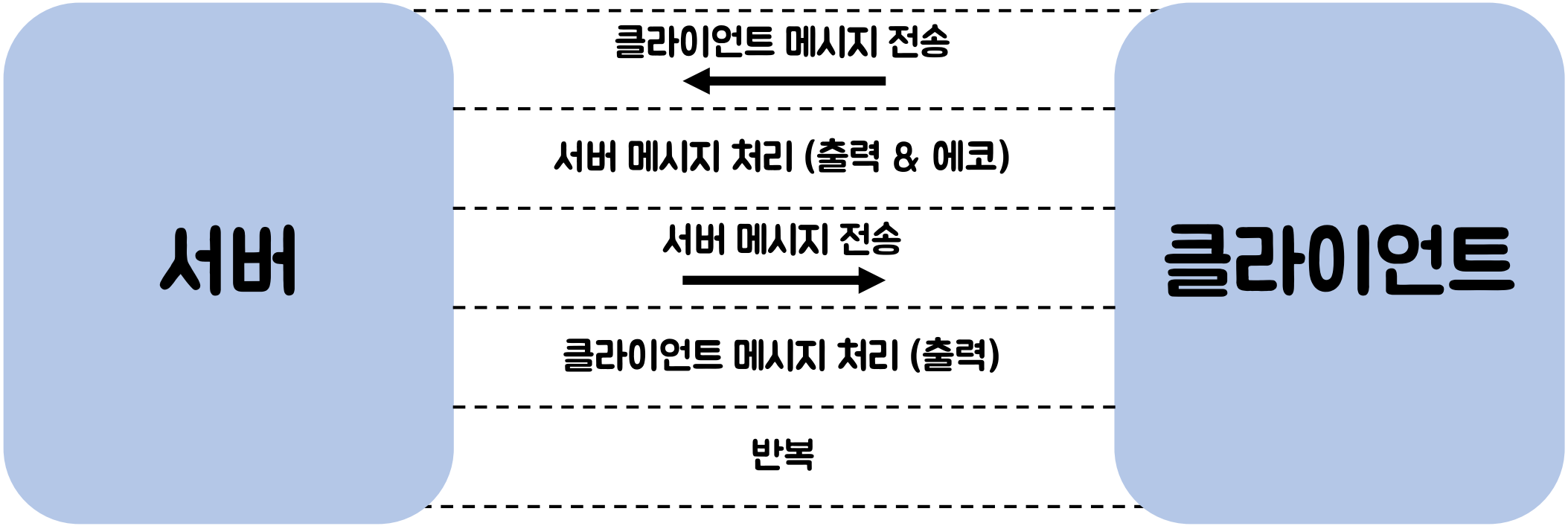
문제점

문제점 찾기

개쩌는 서버도 만들었는데 왜?

문제점 찾기

현재 서버 구조

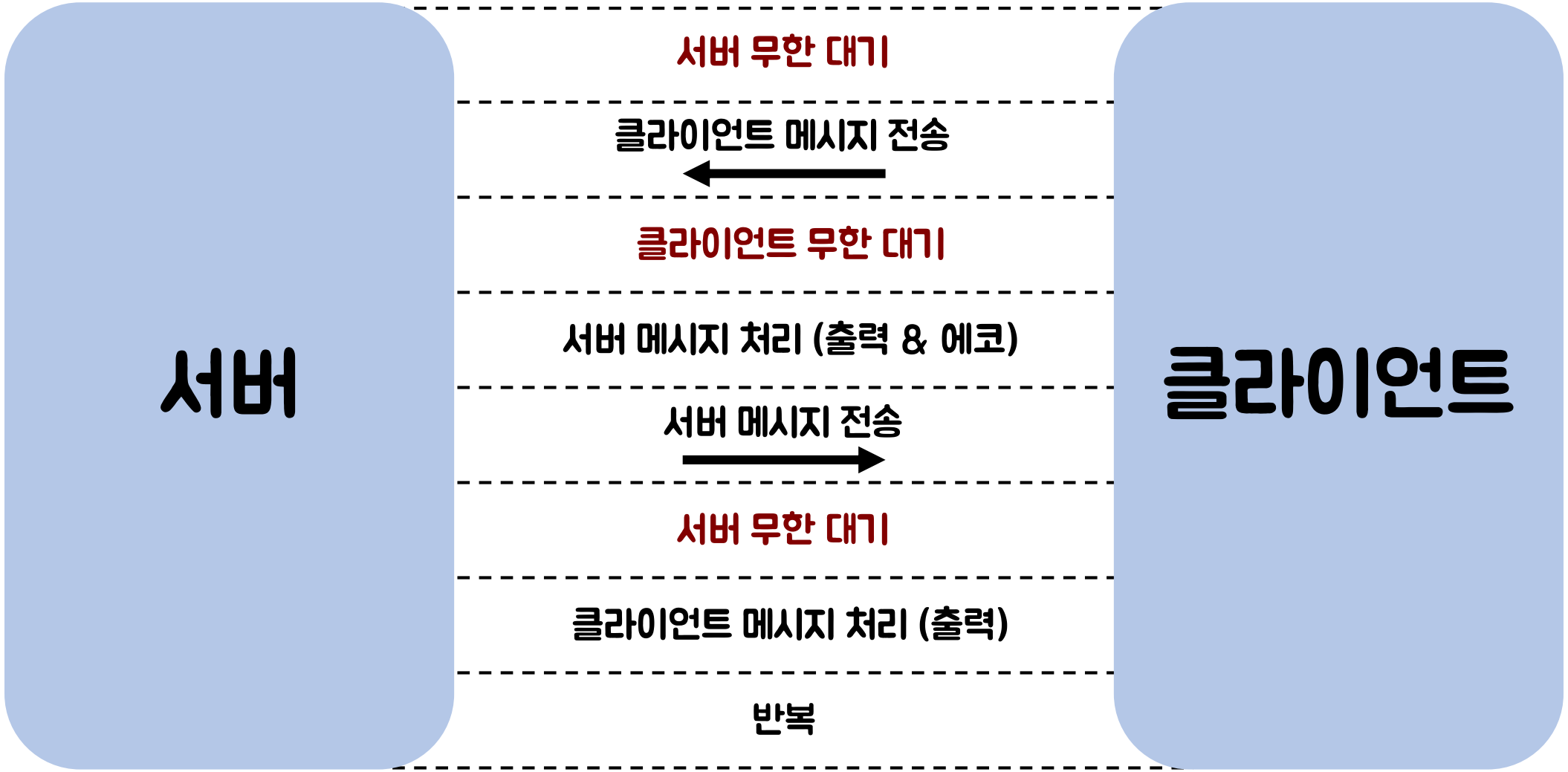


문제점 찾기

아무리 봐도 개쩌는데?

문제점 찾기

현재 서버 구조



문제점 찾기

현재 서버 구조

서버

서버

클라이언트

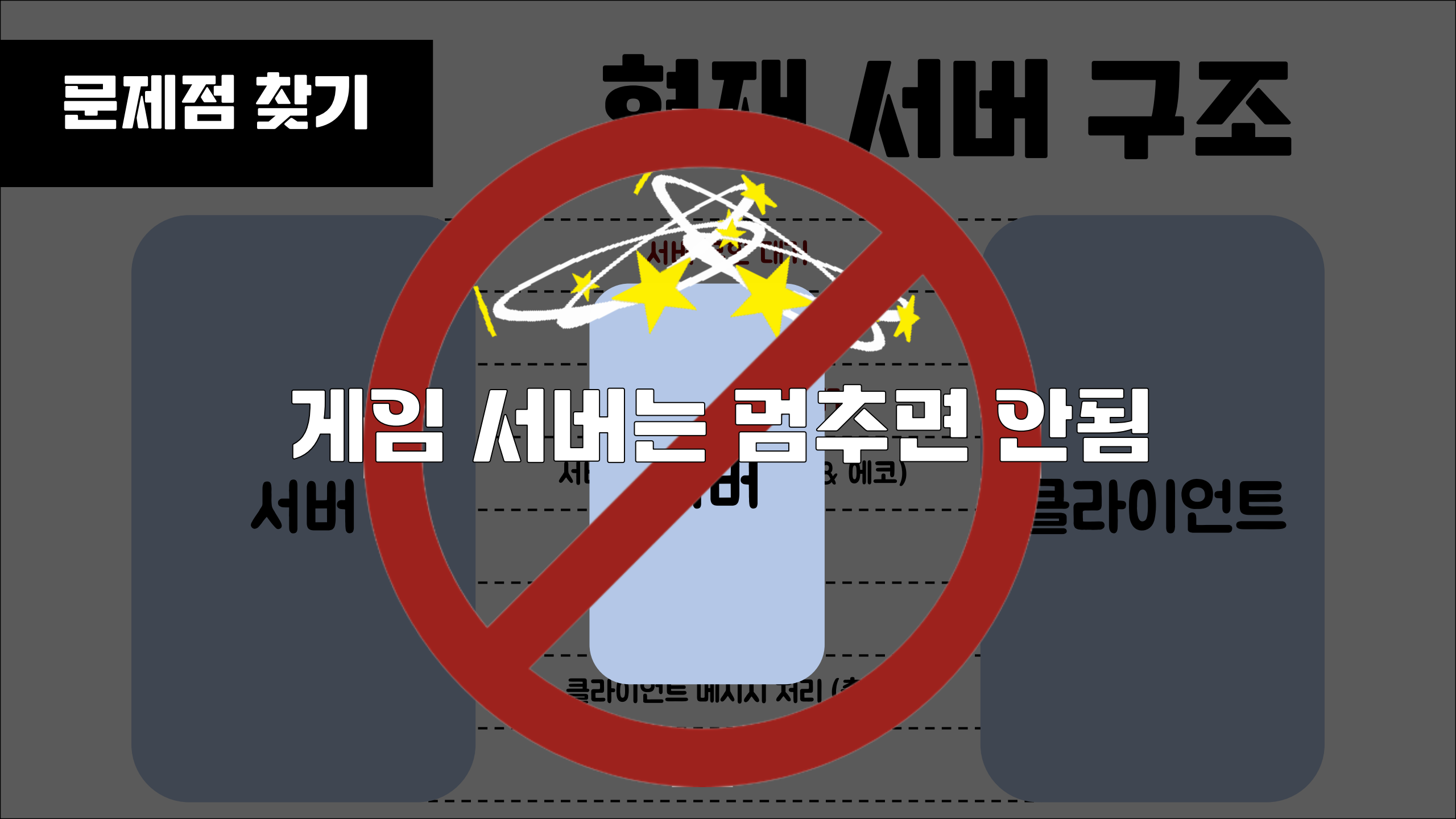
& 에코)

클라이언트 메시지 처리 (출력)

반복

문제점 찾기

현재 서버 구조



게임 서버는 멈추면 안됨

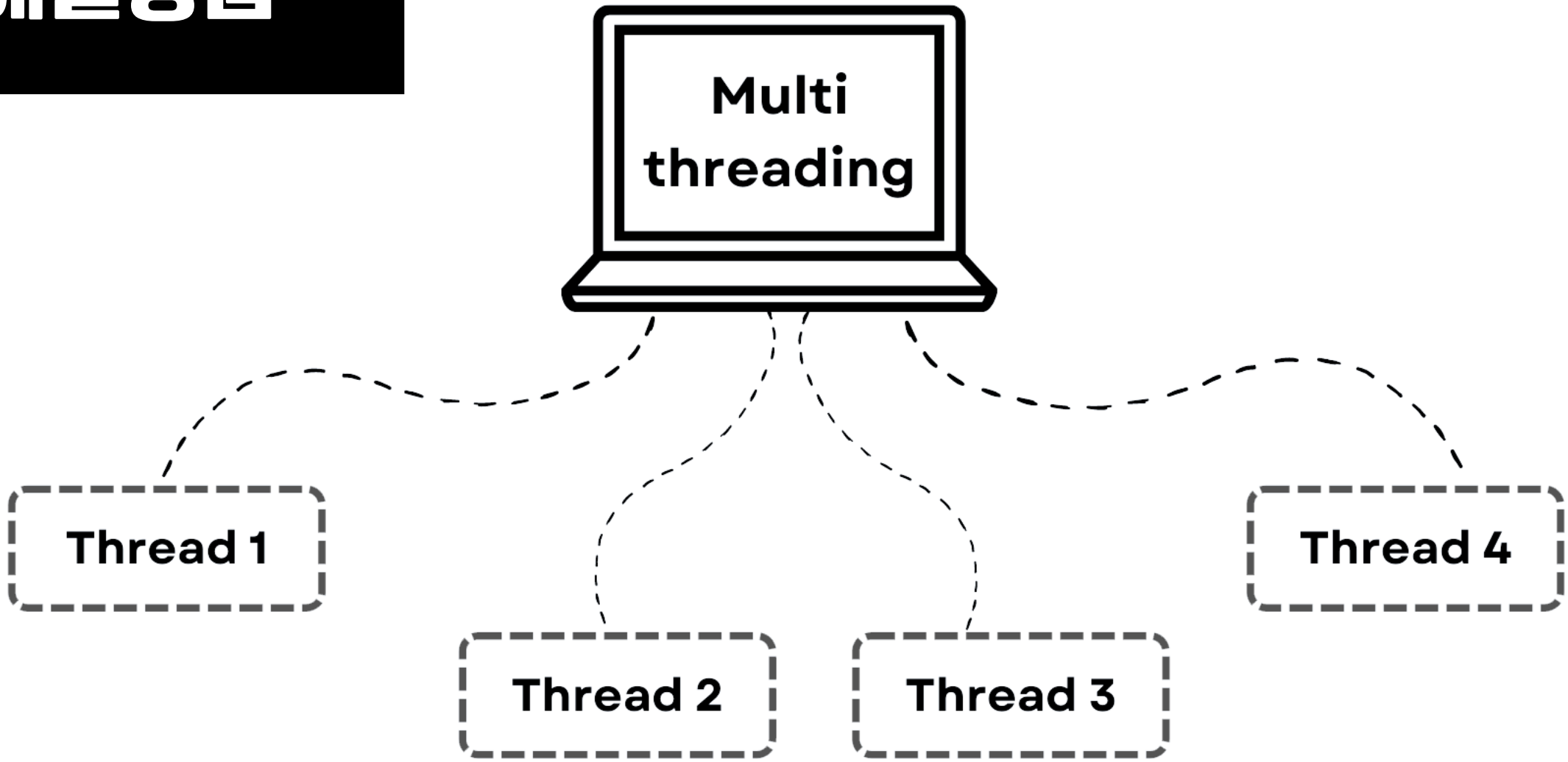
문제점 찾기

현재 서버 구조

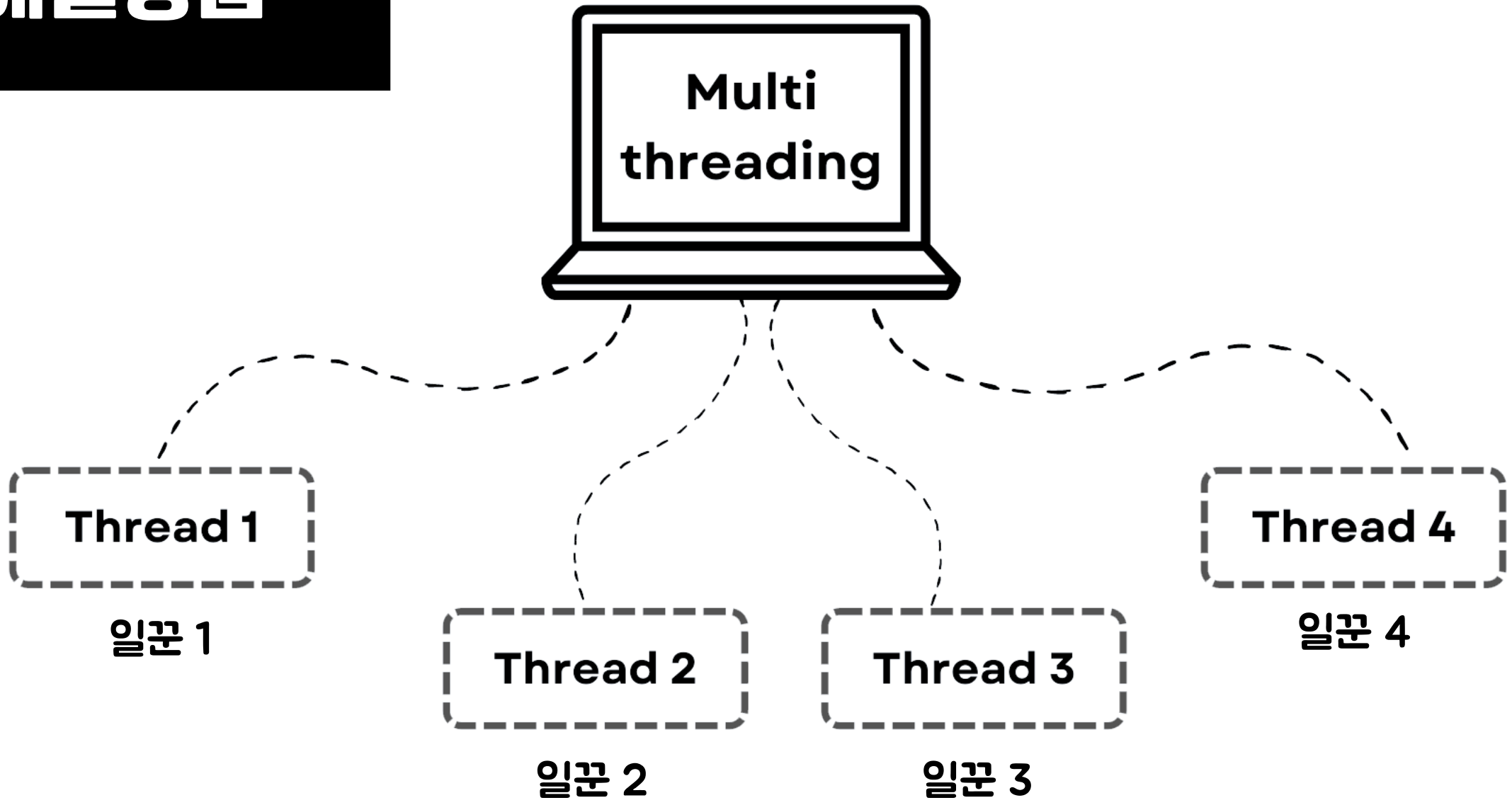
그럼 어떻게 하는데?

해결 방법

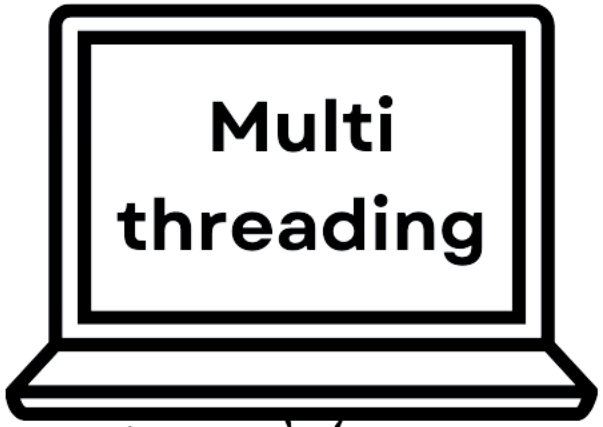
해결방법



해결방법


Multi
threading
Thread 1
일꾼 1
Thread 2
일꾼 2
Thread 3
일꾼 3
Thread 4
일꾼 4

해결방법



Thread 1

클라이언트
연결 & 연결 해제

Thread 2

메시지 수신

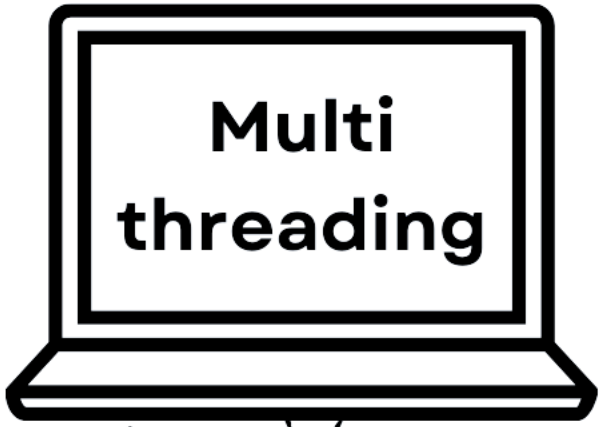
Thread 3

메시지 송신

Thread 4

메시지 처리

해결방법



Thread 1

클라이언트
연결 & 연결 해제

Thread 2

메시지 수신

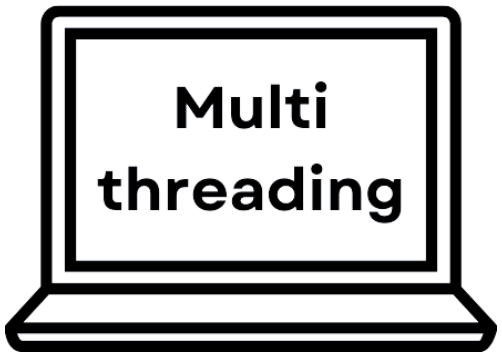
Thread 3

메시지 송신

Thread 4

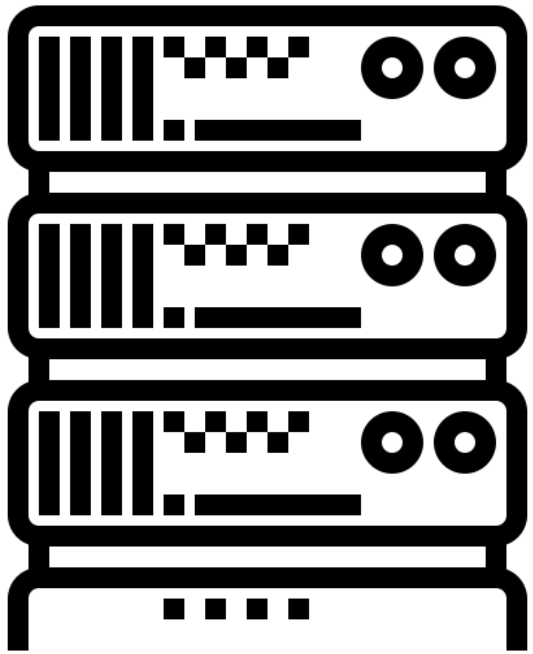
메시지 처리

결론



+

서버



=

개쩌는
서버 개발자