为什么会有socket?

为了能够让server能够同时处理多个client的通信连接。

没 socket 的世界，服务器就只能：

- 在固定端口只能接一个客户端；
- 连接建立后，端口被“占用”；
- 其他客户端连不上，必须等前一个断开连接。

🧠 你的问题核心是：

> socket 是不是为了让一个服务器能同时处理多个客户端连接？
> 否则一个端口被 A 连接后，B 就得等 A 断开后才能连？

答案是：

> ✅ **是的，socket 的机制 + 多线程/多进程/异步 等手段**，正是为了**突破“一个端口只能处理一个连接”的限制，从而让服务器能同时服务多个客户端。**

🔧 原理解析：socket 如何实现“一对多”通信？

🎯 一个端口只能绑定一次，但能接多个连接！

- 服务端通过 `bind(HOST, PORT)` 把一个 socket 绑定到某个端口；
- 客户端通过 `connect(HOST, PORT)` 来发起连接；
- 当服务器调用 `accept()` 成功后，会返回一个新的 socket **专门处理这个连接**，而原始的 socket 仍然留着 **继续监听新的连接**！

Socket

两个主机，要想建立通信，就需要socket。

🔁 Socket 通信过程（以 TCP 为例）：

1. **服务器端（Server）**
   - 创建 socket
   - 绑定 IP 和端口（bind）
   - 监听连接（listen）
   - 等待客户端连接（accept）
2. **客户端（Client）**
   - 创建 socket
   - 发起连接（connect）
3. **双方开始通信（send / recv）**
4. **通信结束后关闭连接（close）**

为什么要socket，知道对方ip地址和port端口不就行了?
因为除了知道对方的ip地址，还需要设定好，你要使用哪一个服务（哪一个端口）

🔍 为什么不能“只靠 IP”通信?

❌ IP 地址 + 端口 ≠ 能完整通信

1. **IP 层是“地址定位”**
   - 它只能定位“目标是谁”
   - 类似你知道一家餐厅的地址，但你得打电话或进店点菜才算“交流”
2. **要进行实际通信，还需要更多层次支持：**
   - **端口号**：告诉系统“我要和哪个服务（Web/SSH）说话”
   - **连接状态、缓冲、流控制、重传机制等**：IP层根本不管这些
   - 这些正是 TCP/UDP + socket 层实现的

如果没有socket，你就自己管理tcp的三次握手状态，连接状态维护等工作。

🧱 Socket = 提供完整通信的“工具包”

Socket 之所以必要，是因为它封装了：

功能	说明
IP + 端口	标识唯一通信端点
协议类型	TCP 还是 UDP?
数据缓冲区	读写数据的内存空间
状态管理	TCP三次握手、断开连接、连接状态维护
接口操作	提供 `send()` / `recv()` / `connect()` 等标准 API 供程序调用

如果没有 socket：

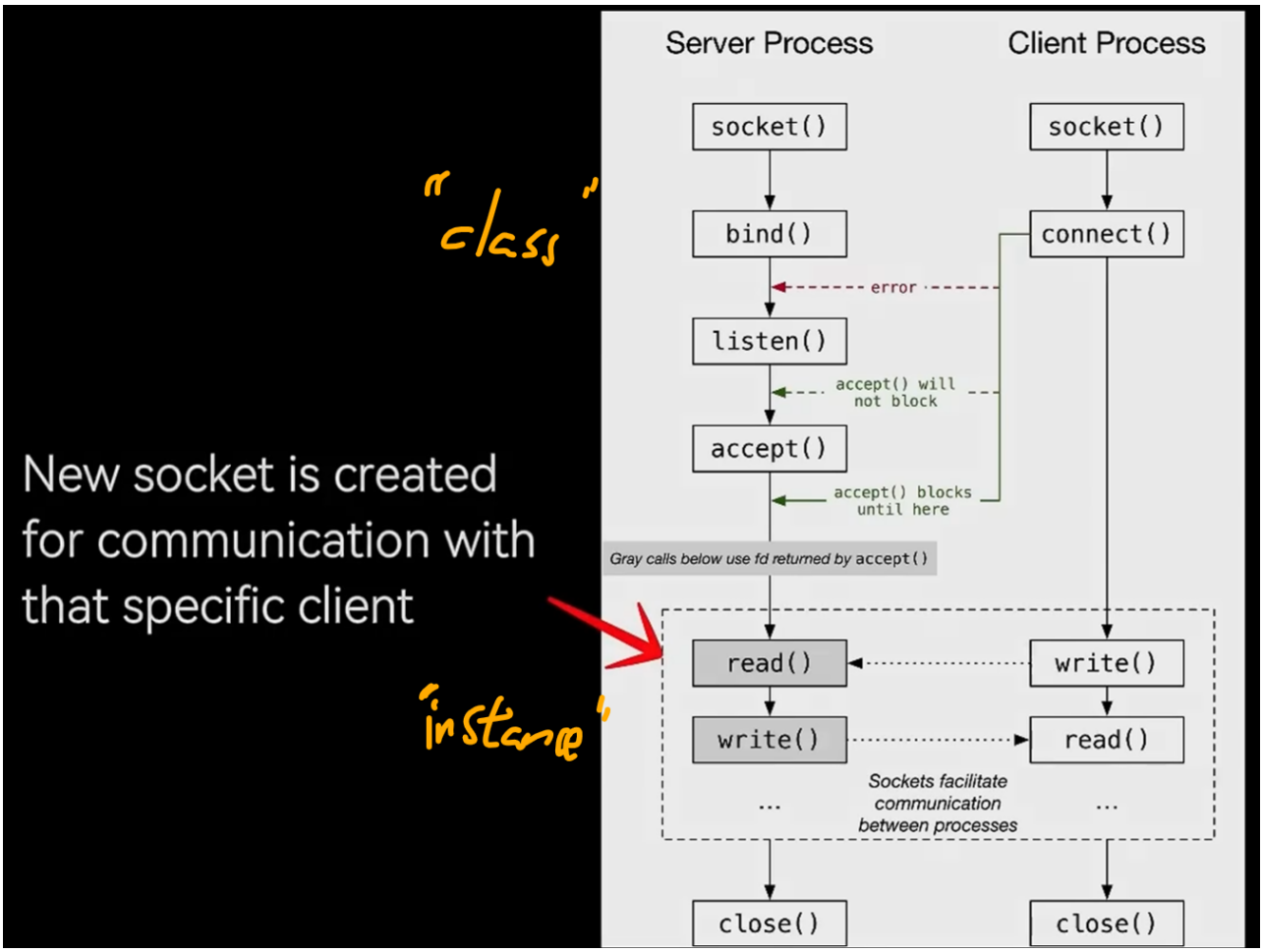
> 你就得自己手动处理 IP 分片、TCP 序列号管理、重传逻辑、流控、超时、应用接口... 太复杂!

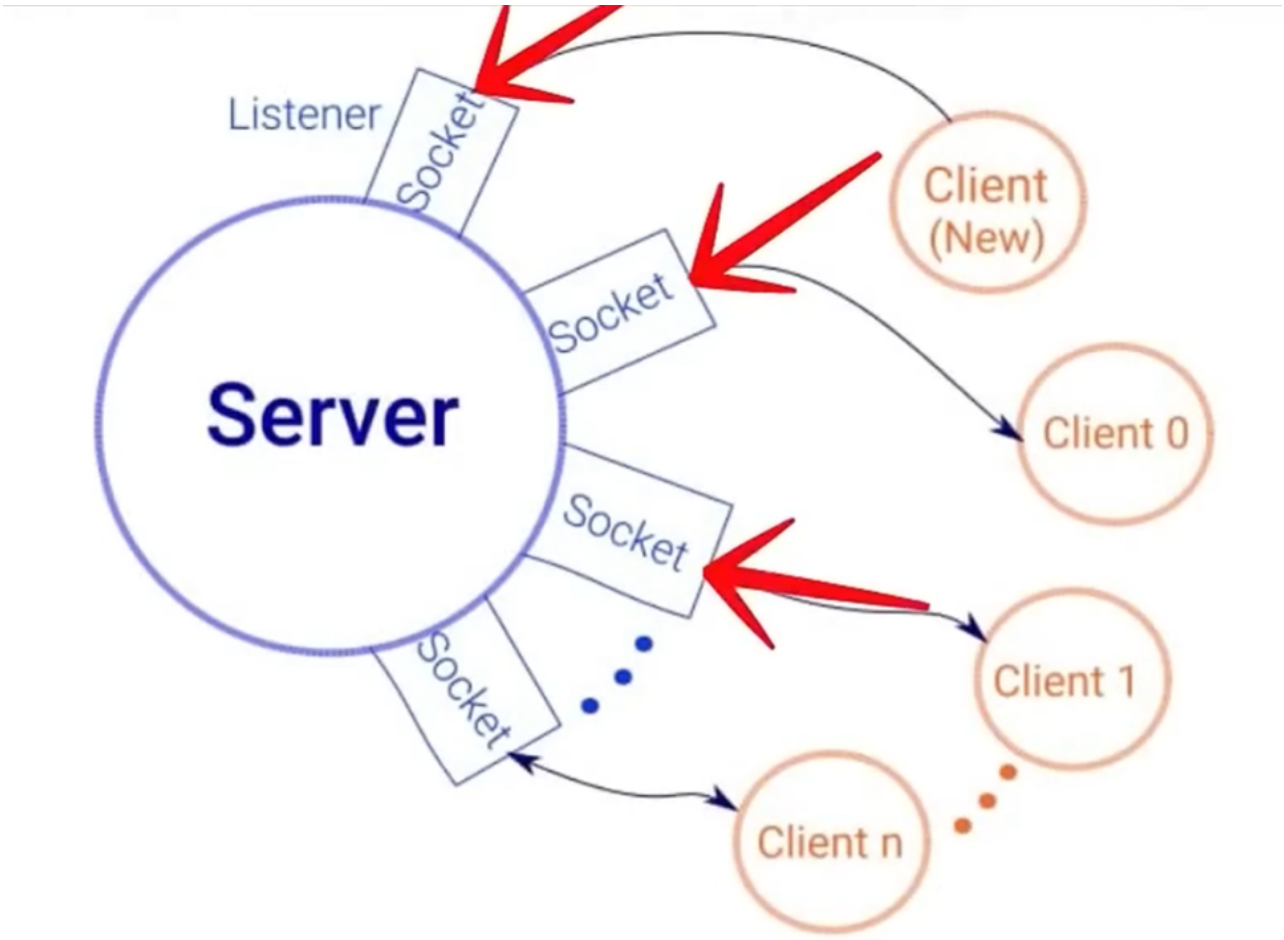
https://www.bilibili.com/video/BV1P9jRzXEbZ/?spm_id_from=333.1007.top_right_bar_window_history.content.click&vd_source=1dea49639bb13f3bfc9b196ac437ce8e —— 详细

当server-client要建立连接的时候，它们各自创造一个socket，从而开始进行沟通。

server socket会绑定自己的ip和端口，并开始listen是否有其他主机的request。若有，则建立新的socket，相当于instance，来专门与client A进行沟通。

否则一个client一个port，根本就不够用。





Server and Clients