**进程和线程的区别：**

进程：操作系统资源分配的基本单位；

线程：调度的基本单位；

进程是资源分配的最小单位，线程是CPU调度的最小单位；进程间隔离，线程共享进程资源

**进程通信的方式：**

信号、信号量、消息队列、无名管道、有名管道、高级管道、共享内存、套接字

**重写、重载和多态：**

重写：子类对父类中已定义的方法进行重新实现，方法名、参数列表和返回类型必须与父类方法完全相同

重载：同一个类中定义多个同名方法，但参数列表（参数类型、数量或顺序）必须不同

多态：指同一操作（方法）作用于不同对象时，表现出不同的行为，核心是“一个接口，多种实现”

**TCP和UDP的区别：**

TCP：面向连接（Connection-Oriented）。传输数据前需通过“三次握手”建立可靠连接，结束时通过“四次挥手”断开连接。

UDP：无连接（Connectionless）。直接发送数据包，无需预先建立连接

TCP：确保数据可靠传输，通过以下机制：

确认应答（ACK）：接收方收到数据后发送确认信号。

重传机制：未收到ACK时重发丢失的数据包。

流量控制：滑动窗口协议调整发送速率，避免接收方过载。

拥塞控制：动态调整发送速率，防止网络拥堵。

UDP：不保证可靠性，无确认、重传或流量控制机制。

TCP慢UDP快



**TCP三次握手：**

目的是为了确认双方的接收能力和发送能力是否正常

第一次：客户端创建传输控制模块，向服务器端发出连接请求

证明客户端的发送能力正常

第二次：服务器收到请求报文后向客户端发送确认报文

证明服务器端的接收能力、发送能力正常

第三次：客户端收到确认后再向服务器发送确认（防止已经失效的连接请求报文突然又传送到了服务器，从而产生错误）

证明客户端的接收能力正常

**四次挥手：**

第一次：关闭方向被动方发送FIN数据报

第二次：被动方收到后向关闭方发送ACK确认帧

第三次：被动方完成数据发送后向关闭方发送FIN包

第四次：关闭方回复ACK确认帧

**软件开发需要的文档：**

可行性分析报告

项目开发计划

软件需求说明书

概要设计说明书

详细设计说明书

用户操作手册

测试计划

测试分析报告

开发进度月报

项目开发总结报告

软件维护手册

软件问题报告

软件修改报告

**软件开发生命周期：**

计划和需求分析 (Planning and Requirement Analysis)

设计项目架构 (Project Architecture)

开发和编程 (Development and Coding)

测试 (Testing)

部署 (Deployment)

**黑白盒：**

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。

白盒测试:又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。

**死锁的四个必要条件：**

互斥条件：资源是独占的且排他使用，进程互斥使用资源

不可剥夺条件：进程所获得的资源在未使用完毕之前，不被其他进程强行剥夺

请求和保持条件：进程每次申请它所需要的一部分资源，在申请新的资源的同时，继续占用已分配到的资源

循环等待条件：在发生死锁时必然存在一个进程等待队列

**HTTP和HTTPS：**

安全：



技术：



性能 成本：



场景：



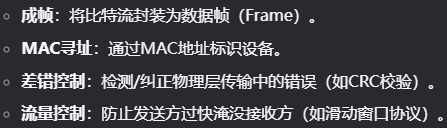
**计算机网络五层结构：**

1. 物理层（Physical Layer）：

负责在物理介质（如电缆、光纤、无线电波）上传输原始比特流

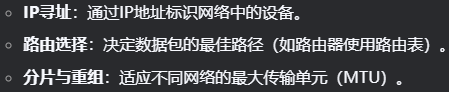
1. 数据链路层（Data Link Layer）：

确保同一局域网（LAN）内相邻节点之间的可靠数据传输



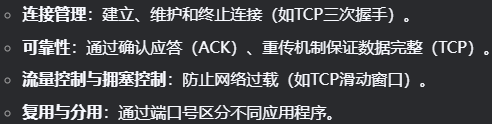
1. 网络层（Network Layer）：

实现不同网络之间的数据路由与转发，确保数据包从源到目的的端到端传输



1. 传输层（Transport Layer）：

提供端到端的通信服务，确保数据可靠传输或高效传输



1. 应用层（Application Layer）：

为应用程序提供网络服务接口，直接面向用户需求

