面经预热-Day6（计网专题）

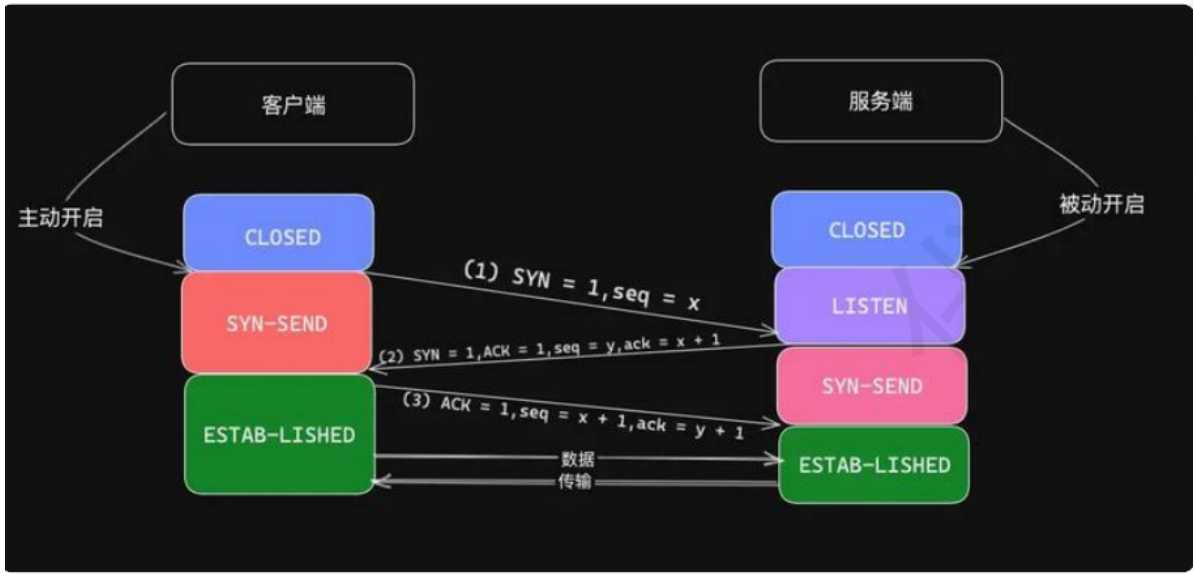
1、从输入URL到页面展示的过程中发生了什么?

1. 浏览器接收到用户请求，先检查浏览器缓存里是否有缓存该资源，如果有直接返回；如果没有进入下一 步网络请求。
2. 网络请求前，进行 DNS解析 ，以获取请求域名的 IP地址 。如果请求协议是 HTTPS ，那么还需要 建立TLS连接 。DNS解析时会按本地浏览器缓存->本地 Host 文件->路由器缓存-> DNS 服务器-> 根 DNS 服务器的顺序查询域名对应 IP ，直到找到为止。
3. 浏览器与服务器IP建立TCP连接。连接建立后，浏览器端会构建请求行、请求头等信息，并把和该域名 相关的 Cookie 等数据附加到请求头中，向服务器构建请求信息。
4. 服务器接收到请求信息，根据请求生成响应数据
5. 浏览器解析响应头。若响应头状态码为 301、302 ，会重定向到新地址；若响应数据类型是字节流类 型，一般会将请求提交给下载管理器；若是HTML类型，会进入下一步渲染流程。
6. 浏览器解析 HTML 文件，创建 DOM 树，解析 CSS 进行样式计算，然后将CSS和DOM合并，构建渲 染树；最后布局和绘制渲染树，完成页面展示。

2、三次握手的过程

三次握手的过程如下：

1. 客户端向服务器发送 SYN 报文、初始化序列号 ISN（seq=x） ，然后客户端进 入 SYN_SEND 状态，等待服务器确认。
2. 服务端发送 ACK 确认服务端的 SYN 报文 (ack=x+1） ，同时发出一个 SYN 报文，带上自 己的初始化序列号 （seq=y ），然后服务端进入 SYN_RECV 状态。
3. 客户端接收到服务端的 SYN、ACK 报文，ACK确认服务端的 SYNC 报文 （ACK=y+1） ，然 后客户端和服务器端都进入 ESTABLISHED 状态，完成 TCP 三次握手。



3、为什么是三次握手，而不是四次、两次

因为三次握手才能保证双方具有接收和发送的能力。 两次握手可能导致资源的浪费，由于没有第三次 握手，服务端就无法确认客户端是否收到了自己的回复，所以每收到一个 SYN ， 服务器都会主动去 建立一个连接，而四次握手可以优化为三次。

4、四次挥手的过程

四次挥手的过程：

1. . 客户端发送一个 FIN 报文给服务端，表示自己要断开数据传送，报文中会指定一个序列号 (s eq=x) 。然后,客户端进入 FIN-WAIT-1 状态。
2. 服务端收到 FIN 报文后，回复 ACK 报文给客户端，且把客户端的序列号值 +1 ，作为ACK + 1 报文的序列号 (seq=x+1) 。然后，服务端进入 CLOSE-WAIT (seq=x+1) 状态，客户端 进入 FIN-WAIT-2 状态。
3. 服务端也要断开连接时，发送 FIN 报文给客户端，且指定一个序列号 (seq=y+1) ， 随后 服务端进入 LAST-ACK 状态。
4. 客户端收到 FIN 报文后，发出 ACK 报文进行应答，并把服务端的序列号值 +1 作为 ACK 报 文序列号 (seq=y+2) 。此时客户端进入 TIME-WAIT 状态。服务端在收到客户端的 ACK 报文后进入 CLOSE 状态。如果客户端等待 2MSL 没有收到回复，才关闭连接

5、为什么是四次挥手?

TCP 是全双工通信，可以双向传输数据。任何一方都可以在数据传送结束后发出连接释放的通知， 待对方确认后进入半关闭状态。 当另一方也没有数据再发送的时候，则发出连接释放通知，对方确认 后才会完全关闭了 TCP 连接。 总结：两次握手可以释放一端到另一端的 TCP 连接，完全释放 连接一共需要四次握手

6、TCP与UDP的概念、特点、区别和对应的使用场景

TCP与UDP的概念

- TCP （传输控制协议）是一种面向连接的、可靠的、基于字节流的传输层通信协议。
- UDP （用户数据报协议）为应用程序提供了一种无需建立连接就可以发送封装的IP数据包的方 法。

特点

1. TCP： 面向连接，传输可靠，传输形式为字节流，传输效率慢，所需资源多。
2. UDP： 无连接、传输不可靠、传输形式为数据报文段，传输效率快，所需资源少。

区别

- 是否面向连接: TCP 是面向连接的传输，UDP 是无连接的传输。
- 是否是可靠传输：TCP是可靠的传输服务，在传递数据之前，会有三次握手来建立连接；在数据 传递时，有确认、窗口、重传、拥塞控制机制。 UDP是不可靠传输，数据传递不需要给出任何确 认，且不保证数据不丢失及到达顺序。
- 是否有状态：TCP 传输是有状态的，它会去记录自己发送消息的状态比如消息是否发送了、是否 被接收了等等，而 UDP 是无状态的。
- 传输形式: TCP 是面向字节流的，UDP 是面向报文的。
- 传输效率:由于TCP 传输的时候多了连接、确认重传等机制，所以 TCP 的传输效率要比UDP 低。

- 首部开销 :TCP 首部开销 (20 ~ 60字节)比UDP 首部开销 (8字节)要大。
- 是否提供广播或多播服务: TCP 只支持点对点通信UDP 支持一对一、一对多、多对一、多对多。

对应的使用场景

- TCP常用于要求通信数据可靠场景（如网页浏览、文件传输、邮件传输、远程登录、数据库操作 等）。
- UDP常用于要求通信速度高场景（如域名转换、视频直播、实时游戏等）。

	TCP	UDP
可靠性	可靠	不可靠
连接性	面向连接	无连接
报文	面向字节流	面向报文
效率	传输效率低	传输效率高
双工性	全双工	一对一、一对多、多对一、多对多
流量控制	滑动窗口	无
拥塞控制	慢开始、拥塞避免、快重传、快恢复	无
传输速度	慢	快
应用场景	对效率要求低，对准确性要求高或者要求有连接的场景	对效率要求高，对准确性要求低

7、HTTP请求常见的状态码和字段

字段

- Host字段：客户端发送请求时，用来指定服务器的域名
- Content-length字段：服务器返回数据时，带有该字段，表示回应的数据长度
- Connection字段：常用于客户端要求服务器使用HTTP长连接时使用
- Content-Type字段：服务器回应时，告诉客户端，本次数据的格式
- Content-Encoding字段：服务器返回的数据使用了什么压缩格式

状态码

2xx：请求成功

1. 200（成功）服务器已成功处理了请求，通常，这表示服务器提供了请求的网页
2. 201（已创建）请求成功并且服务器创建了新的资源
3. 202（已接受）服务器已接受请求，但尚未处理
4. 203（非授权信息）服务器已成功处理了请求，但返回的信息可能来自另一来源
5. 204（无内容）服务器成功处理了请求，但没有返回任何内容
6. 206（部分内容）服务器成功处理了部分 GET 请求，服务器返回的body数据是资源的一部分

3xx：请求重定向

1. 301(永久重定向) 请求的网页已永久移动到新位置
2. 302(临时重定向)，说明请求的资源还在，但暂时需要用另一个URL来访问
3. 304(未修改)自从上次请求后，请求的网页未修改过。服务器返回此响应时，不会返回网页内容，重定向到已存在的缓存文件。

4xx：请求错误

1. 401(未授权) 请求要求身份验证。 对于需要登录的网页，服务器可能返回此响应
2. 403（禁止）服务器拒绝请求

3. 404(未找到) 服务器找不到请求的网页

5xx：服务器错误

1. 500(服务器内部错误) 服务器遇到错误，无法完成请求
2. 501(尚未实施) 服务器不具备完成请求的功能。
3. 502(错误网关)服务器作为网关或代理，从上游服务器到无效响应
4. 503(服务不可用) 服务器目前无法使用(由于超载或停机维护)。 通常，这只是暂时状态。
5. 504(网关超时) 服务器作为网关或代理，但是没有及时从上游服务器收到请求