网络部分面试题

1. 从 URL 输入到页面展现发生了什么？

1. 在浏览器中输入url
2. 应用层DNS解析域名：先本地查找，再查询DNS服务器
3. 应用层客户端发送HTTP请求
4. 传输层TCP传输报文:三次握手
5. 网络层IP协议查询MAC地址
6. 数据到达数据链路层
7. 服务器接收数据
8. 服务器响应请求
9. 服务器返回相应文件
10. 页面渲染。解析HTML以构建DOM树 –> 构建渲染树 –> 布局渲染树 –> 绘制渲染树。

参考： 从输入URL到浏览器显示页面发生了什么

2.cookie和session的异同

1. cookie数据存放在客户的浏览器上，session数据放在服务器上。
2. cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗，考虑到安全应当使用session。
3. session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，考虑到减轻服务器性能方面，应当使用cookie。
4. 单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie
5. 可以考虑将登陆信息等重要信息存放为session，其他信息如果需要保留，可以放在cookie中。

参考： Session和Cookie的区别与联系

3.HTTP和HTTPS的区别

1. HTTP是超文本传输协议，信息是明文传输，HTTPS是具有安全性的SSL加密传输协议。，
2. HTTPS协议需要ca申请证书，一般免费证书少，因而需要一定费用。
3. HTTP和HTTPS使用的是完全不同的连接方式，用的端口也不一样。前者是80，后者是443。
4. HTTP连接是无状态的，HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，安全性高于HTTP协议。

4.ssl加密使用了哪种算法，如何加密

1. 在客户端与服务器间传输的数据是通过使用对称算法（如 DES 或 RC4）进行加密的。
2. 公用密钥算法（通常为 RSA）是用来获得加密密钥交换和数字签名的，此算法使用服务器的SSL数字证书中的公用密钥。

5.TCP三次握手的过程，为什么是三次而不是两次或者四次？

第一次握手：客户端A发送一个syn（同步）包（syn=x）给服务器B，进入SYN_SEND状态，等待服务器确认

第二次握手：服务端B收到客户端A发送的同步包，确认客户端的同步请求（ack=x+1）,同时也发送一个同步包，
也就是一个ACK包+SYN包服务器进入SYN_RECV状态

第三次握手：客户端A收到服务器B的SYN+ACK包，向服务器B发送一个确认包，此包发送完毕，客户端和服务器进入
ESTABLISHED状态，完成三次握手

不是两次是因为服务器B无法知道客户端A是否已经接收到自己的同步信号，如果这个同步信号丢失了，A和B就B的初始序列号将无法达成一致

不是四次的话是因为完全没有必要，三次已经足够了

参考： TCP 为什么是三次握手，而不是两次或四次?

6.TCP的四次挥手

第一次：主动关闭方发送一个FIN包，用来关闭主动关闭方到被动关闭方的数据传送，也就是告诉另一方我不再发送数据了，但此时仍可以接收数据

第二次：被动关闭方收到FIN包之后，发送一个确认（ACK）包给对方

第三次：被动关闭方发送一个FIN包，告诉对方不带发送数据

第四次：主动关闭方收到FIN包之后，发送一个ACK包给对方，至此完成四次挥手

7.HTTP报文的格式，传输中以何种方式传输

HTTP报文分为三个部分，起始行、首部和主体，其中起始行和首部以一个回车和换行符分隔，首部和主体以一个空行分隔，其中起始行是对这次HTTP请求或者响

应的描述，请求报文的起始行包括使用的HTTP方法、请求的url地址、HTTP版本，响应报文的起始行包括HTTP的版本，HTTP状态码，http状态码的描述，首部也就是常说的HTTP头部，如Date、Cookie、Content-Type等，主体是这次请求或响应的数据，传输中以明文传输。

参考： HTTP权威指南

8.常见的HTTP头部

可以将HTTP首部分为通用首部、请求首部、响应首部、实体首部，通用首部表示一些通用信息，如Date表示报文创建时间，请求首部就是请求报文中独有的，如cookie、和缓存相关的If-Modified-Since，响应首部就是响应报文中独有的，如set-cookie和重定向有关的location，实体首部用来描述实体部分，如Allow 用来描述可执行的请求方法，Content-Type描述主体类型，Content-Encoding描述主体的编码方式

9.HTTP状态的简要分类

可以按照HTTP状态码的第一个数字分类，1xx表示信息，2xx表示成功，3xx表示重定向，这里需要注意的是304，表示未修改，4xx表示客户端错误，最常见的是404，5xx表示服务端错误。

10.HTTP状态码101、200、301、302、304的具体含义

101：切换协议 200：正常，OK，301：永久重定向，302：临时重定向，304：未修改

14.用户登陆过程的简要说明，如何判断用户是否登录？

用户输入用户名和密码，通过post请求将密码和用户名发送给服务器，服务器比对收到的用户名、密码和数据库中的数据进行比对，不一致则做出响应，反馈信息给客户端，如果比对一致则服务端生成一个session，这个session可以存储在内存、文件、数据库中，同时生成一个与之一一对应的sessionID作为cookie发送给客户端，比对成功之后反馈信息，这时一般会进行一次重定向，重定向至登陆之后的默认页面。判断用户登录则是根据这个sessionID，每次请求会先检查有没有这次类似sessionID的cookie发送过来，没有则认为没有登录，有则是否有相应的session，这个session是否过期等，来判断用户是否登录，登录是否过期。

15.tcp和udp的区别

TCP面向连接的、提供可靠传输的协议，而UDP则是面向非连接。不可靠传输的协议，之所以说TCP是可靠的传输协议是因为TCP协议在传输数据之前有一个确认双方是否连接的过程，而UDP没有，也正是因此在传输速度方面，UDP更快。因此需要可靠传输需要选用TCP，不需要可靠传输情况下选择UDP。

16.udp的阻塞机制，如何处理

17.简要介绍一下socket协议

18. Get和Post的区别

GET - 从指定的资源请求数据。 POST - 向指定的资源提交要被处理的数据。

然而，在以下情况中，请使用 POST 请求：

无法使用缓存文件（更新服务器上的文件或数据库）

向服务器发送大量数据（POST 没有数据量限制）

发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

GET对比POST

19.什么是正向代理？什么是反向代理？

- **正向代理** 就是客户端向代理服务器发送请求，并且指定目标服务器，之后代理向目标服务器转交并且将获得的内容返回给客户端。比如翻墙
- **反向代理** 指代理会判断请求走向何处，并将请求转交给客户端，客户端只会觉得这个代理是一个真正的服务器。如负载均衡。

20.介绍一下HTTPS的连接过程

参考： Https 建立安全连接的过程（SSL原理）

21.介绍一下DNS的查找过程？

递归查询

第一步：在hosts静态文件、DNS解析器缓存中查找某主机的ip地址

第二步：上一步无法找到，去DNS本地服务器（即域服务器）查找，其本质是去区域服务器、服务器缓存中查找

第三步：本地DNS服务器查不到就根据'根提示文件'向负责顶级域'.com'的DNS服务器查询

第四步：‘根DNS服务器’根据查询域名中的‘xyz.com’，再向xyz.com的区域服务器查询

第五步：www.xyz.abc.com的DNS服务器直接解析该域名，将查询到的ip再原路返回给请求查询的主机

迭代查询参考：DNS查询过程

22.http连接性能优化，长连接，keep-alive

HTTP1.1开始,默认采用持久连接,使用了一种叫做keepalive connections 的机制,它可以在传输数据后仍然保持连接,当客户端再次获取数据时,直接使用刚刚空闲下来的连接,而无需再次握手.低线路负载，提高传输速度.

Keep-Alive不会永久保持连接，它有一个保持时间，可以在不同的服务器软件（如Apache）中设定这个时间。实现长连接需要客户端和服务端都支持长连接。

HTTP协议的长连接和短连接，实质上是TCP协议的长连接和短连接。

23. websocket 和http区别

1. websocket是持久连接的协议,而http是非持久连接的协议.
2. websocket是双向通信协议,模拟socket协议,可以双向发送消息,而http是单向的.
3. websocket的服务端可以主动向客服端发送信息,而http的服务端只有在客户端发起请求时才能发送数据,无法主动向客户端发送信息.

参考： http,websocket和socket详解

24.端口号的作用是什么？

作用是区分服务类别和同一时间进行多个会话

参考： 端口号的作用及常见端口号用途说明