**面试**

# 1 Http

## 1.1http协议介绍

http是一个基于请求与响应模式的、无状态、应用层的协议，基于TCP的连接方式。特点：无连接，限制每次连接只处理一个请求，节省传输时间；无状态，无状态是指协议对于事务处理没有记忆能力，即如果后续处理需要前面的信息，必须重传。

* *我们在传输数据时，可以只使用(传输层)TCP/IP协议，但是那样的话，如果没有应用层，便无法识别数据内容。如果想要使传输的数据有意义，则必须使用到应用层协议。应用层协议有很多，比如HTTP、FTP、TELNET等，也可以自己定义应用层协议。WEB使用HTTP协议作应用层协议，以封装HTTP文本信息，然后使用TCP/IP做传输层协议将它发到网络上*
* *socket是对TCP/IP协议的封装，Socket本身并不是协议，而是一个调用接口(API)；TCP/IP只是一个协议栈，就像操作系统的运行机制一样，必须要具体实现，同时还要提供对外的操作接口。*

*这个就像操作系统会提供标准的编程接口，比如win32编程接口一样，TCP/IP也要提供可供程序员做网络开发所用的接口，这就是Socket编程接口。*

* *传输层的TCP是基于网络层的IP协议的，而应用层的HTTP协议又是基于传输层的TCP协议的，而Socket本身不算是协议，它只是提供了一个针对TCP或者UDP编程的接口。*
* *TCP的三次握手：****第一次握手****：客户端发送syn包(syn=j)到服务器，并进入SYN\_SEND状态，等待服务器确认;****第二次握手****：服务器收到syn包，必须确认客户的SYN(ack=j+1)，同时自己也发送一个SYN包(syn=k)，即SYN+ACK包，此时服务器进入SYN\_RECV状态;****第三次握手****：客户端收到服务器的SYN+ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。*

## 1.2 http请求

* http请求由三部分组成：请求行、消息报头、请求正文
* 请求方法：
  + get：请求获取request-URL所标识的资源
  + post：在request-URL所标识的资源后附加新的数据
  + head：获取资源响应消息报头，方法常用于测试超链接的有效性，是否可以访问，以及最近是否更新。
  + put：请求服务器存储一个资源，并用request-URL作为其标识
  + delete：请求服务器删除request-URL所标识的资源
  + trace：请求服务器回送收到的请求信息，主要用于测试或诊断
  + options：请求查询服务器的性能，或者查询与资源相关的选项和需求，返回服务器支持的请求方法
* header内容：
  + Accept:image/gif/text/html
  + Content-length:22
  + Connection:keep-alive
  + Cache-Control:no-cache

## 1.3 http响应

* 响应状态码：
  + 1xx : 指示信息，表示请求已接收，继续处理
  + 2xx： 成功，表示请求已被成功接收、理解、接受
  + 3xx： 重定向，要完成请求必须进行进一步的操作；301-moved permanently；302-
  + 4xx：客户端错误，请求语法错误或请求无法实现；400-bad request；401-Unauthorized；403-Forbidden；404-Not Found；
  + 5xx：服务器错误，服务器未能实现合法的请求；500-Internal Server Error；502-bad gateway；504-gateway timeout

## 1.4其他（网关代理通道）

* **代理(Proxy)**：一个中间程序，它可以充当一个服务器，也可以充当一个客户机，为其它客户机建立请求。请求是通过可能的翻译在内部或经过传递到其它的 服务器中。一个代理在发送请求信息之前，必须解释并且如果可能重写它。代理经常作为通过防火墙的客户机端的门户，代理还可以作为一个帮助应用来通过协议处 理没有被用户代理完成的请求。
* **网关(Gateway)**：一个作为其它服务器中间媒介的服务器。与代理不同的是，网关接受请求就好象对被请求的资源来说它就是源服务器；发出请求的客户机并没有意识到它在同网关打交道。网关经常作为通过防火墙的服务器端的门户，网关还可以作为一个协议翻译器以便存取那些存储在非HTTP系统中的资源。
* **通道(Tunnel)**：是作为两个连接中继的中介程序。一旦激活，通道便被认为不属于HTTP通讯，尽管通道可能是被一个HTTP请求初始化的。当被中继 的连接两端关闭时，通道便消失。当一个门户(Portal)必须存在或中介(Intermediary)不能解释中继的通讯时通道被经常使用。

## 1.5 缓存

## 1.6 cookie与session区别

<https://www.zhihu.com/question/19786827>

sessionID是服务器发送给客户端的用于标识同一用户，比如购物车、结算类似需要验证是哪一个用户对服务器的请求，sessionID在服务器可存入内存、数据库、文件等等，客户端则会存入cookie中，在每次请求中都会把sessionID发送到服务器，若客户端禁用了cookie功能，则会通过一种重写URL的技术来识别用户（在每次请求的URL后追加sid=xxx的参数）。Cookie可以记录用户的信息（如登录账号密码，在下次访问同一网页时自动填入账号密码）

* session在服务器，cookie在客户端
* session默认被存在服务器的一个文件里（不是内存）
* session的运行依赖sessionID 而sessionID是存在cookie中的，也就是说，如果浏览器禁用了cookie 同时session也会失效（但可通过在URL中传递sessionID的方式实现）
* session可放在文件、数据库、或内存中
* 用户验证这种场合一般会用session

# linux命令

将当前目录下的文件a变成隐藏文件：

mv a .a