你好，我叫李莉莉，上一家公司是上海速道科技有限公司，担任的是测试工程师职位，我主要负责的两个项目一个是开造项目，主要进行功能逻辑测试，其次是奥特莱斯商城，主要负责接口测试，其中使用的主要工具有postman，linux，mysql，jmeter，这就是我的一个简单的自我介绍

编写测试用例的方法

等价类，边界值，场景设计法

文件上传功能测试点

上传文件的最大值，小于等于最大值，上传成功，大于最大值，上传失败

上传文件格式是否全部支持

上传文件是否支持中文名称

文件名称是否支持特殊字符

上传过程断网，有提示长传是否成功

上传十网速很慢，超过一定时间，提示

界面测试 提示性文字是否正确

选择好，但是未上传的文件是否可以取消选择

是否同时可以上传多个文件

上传过程是否有取消正在上传文件的功能

上传文件过程中，是否可以进行其他操作

Cd 进入某个目录

Pwd查看当前目录

Ls 查看当前目录都有哪些内容 ls -all 当前目录的所有内容 ls -l 文件的详细信息都显示 等于ll

创建文件夹 mkdir 文件名

Mkdir 文件名 -p 如果上级目录不存在，则可以新建目录

Mkdir 文件名 文件名 连续创建两个文件夹

Touch 创建一个空文件 a.txt 编辑文件vim i wq

Gedit test.txt 可以写东西

软连接

Ln -s 1.txt 1\_soft\_link 一种快捷方式

查看文件内容 cat 文件名

Find . -name 2.txt文件名

Grep 查找文件内容

**=========================================**

**Get 和post 的区别**

功能不同 get通常是从服务器上获取数据。 2、post通常是向服务器传送数据。比如提交表单，上传文件

Get方法，请求数据会直接展现在地址栏中 ？以后的为请求数据，各参数之间以&相连

Post方法：会把数据放到报文体，请求数据中，即form data中，地址栏不附带参数

Get比post更不安全，因为参数直接暴露在url中，所以不能用来传递敏感信息

get传送的数据量较小，不能大于2KB。post传送的数据量较大，

Get请求参数会被浏览器完整的记录下来保留在历史记录里，而post参数不会被保留，get请求会被浏览器缓存，而post不会

**OSI七层网络模型** 物理层、数据链路层、网络层、传输层、会话层、表示层、应用层

协议：双方约定俗成的

Http协议

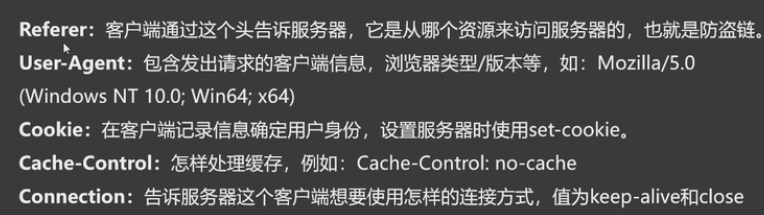
超文本传输协议，属于网络模型中的应用层的协议

Http 请求报文

1：请求行 请求方法 url 协议版本

2：请求头部



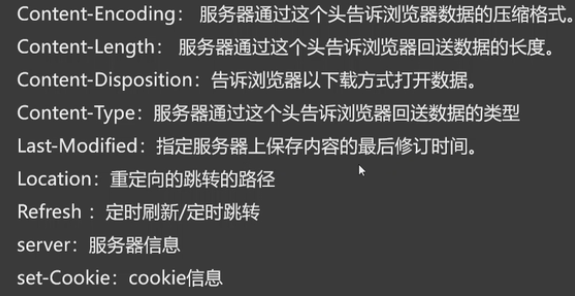


3：请求数据

Http响应报文

响应行 协议版本 状态码 原因短语（ok）

响应头部



响应体

Http 状态码

200 请求成功，服务器成功返回内容

400 请求语法错误或者参数错误

403 服务器拒绝执行请求

404 服务器找不到请求资源

500 服务器故障无法提供服务

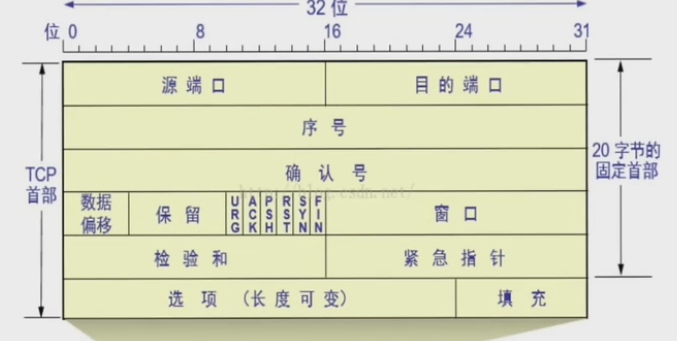
503 服务器超负载或停机维护，一段时间后能提供服务

TCP 和 UDP属于传输层的协议

Udp只关注发送数据，而不关心收发状态，可能出现丢包状况，发送的文本顺序有可能也错误

Tcp 会保证数据的正确性以及数据的顺序，保证可靠性

Tcp的报文头

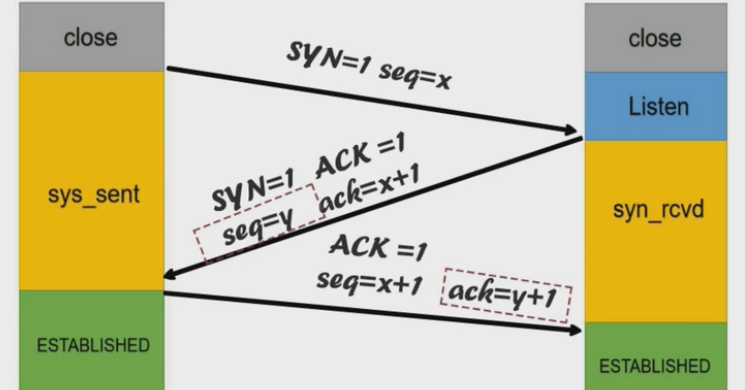


序号Seq 序列号

确认号编号ack 即接收到的上一次远端主机传来的seq然后+1，再发送给远端主机。提示远端主机已经成功接收上一次所有数据。

ACK确认字符，是连接建立状态的标志，为1时表示确认号有效

SYN 同步序列编号,当本字段为1时，表示这是一个连接请求或者连接接受报文



第一次握手客户端将SYN=1产生一个随机序号x ，并将该数据包发送给服务端，此时客户端进入sent状态。

第二次握手，当服务端收到数据包后，由标志位SYN等于1，知道客户端想要建立连接，服务端将标志位SYN和ACK都置为1，确认号ack为x+1 表示对客户端x的确认，同样服务端也会产生一个随机值seq=y，并且将数据包发送给客户端以确认连接请求

第三次握手

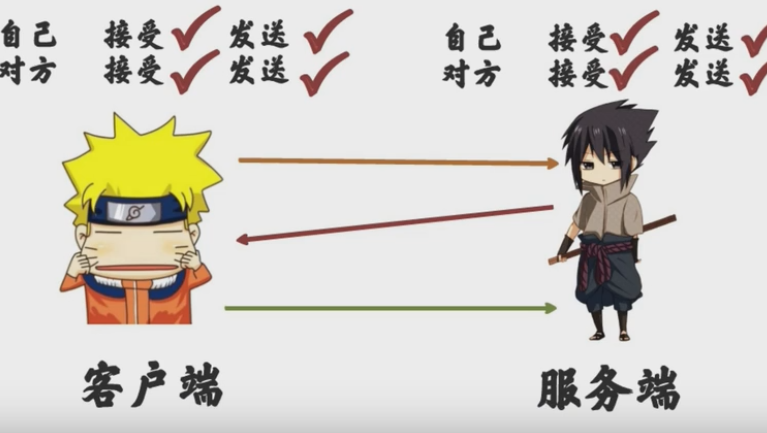
客户端再进行一次确认，将ACK=1，序号为x+1，ack为y+1以表示对服务端y的确认

此时客户端便进入了一个established的状态

服务端收到报文后也会进入established状态。经过三次握手，客户端和服务器能通信

为什么是三次握手，而不是两次握手？

首先必须保证通信双方都有能力发送信息并且接受响应，第一次客户端发送给服务器，服务器知道了客户端的发送是正常的，服务器的接受也是正常的，第二次当服务器发送给服务端，说明客户端知道了自己的发送和接受都是正常的，也说明客户端知道了服务器的发送和接受都是正常的，第三次当客户端在发送给服务器确认报文的时候，服务器就知道了自己的发送是正常的，同时客户端的接受也是正常的



软件测试的定义、

**生命周期：** 需求分析 测试计划 测试用例设计与开发 测试执行

测试用例常见的设计方法：1）等价类划分2）边界值分析法

3）因果图方法4）正交表分析法5）场景分析方法

6）状态图法7）错误猜测法

bug的生命周期：新建、指派、已解决、待验、关闭。  
测试计划、测试报告的编写；

SVN、Git版本控制工具的使用；

sql 这一块，增删改查，条件查询，分组查询 group by，关联查询（内关联，左关联，右关联）一些聚合函数，求和，求个数，最大值，最小值。