python面试题搜集（十）：史上最全python面试题详解（四）

python高级进阶-网络编程和并发（?道题详解）

1、简述 OSI 七层协议。

OSI是Open System Interconnection的缩写，意为开放式系统互联。

OSI七层协议模型主要是：应用层（Application）、表示层（Presentation）、会话层（Session）、传输层（Transport）、网络层（Network）、数据链路层（Data Link）、物理层（Physical）。

1、物理层
主要定义物理设备标准，如网线的接口类型、光纤的接口类型、各种传输介质的传输速率等。它的主要作用是传输比特流（就是由1、0转化为电流强弱来进行传输,到达目的地后在转化为1、0，也就是我们常说的数模转换与模数转换）。这一层的数据叫做比特。

2、数据链路层
定义了如何让格式化数据以进行传输，以及如何让控制对物理介质的访问。这一层通常还提供错误检测和纠正，以确保数据的可靠传输。

3、网络层
在位于不同地理位置的网络中的两个主机系统之间提供连接和路径选择。Internet的发展使得从世界各站点访问信息的用户数大大增加，而网络层正是管理这种连接的层。

4、运输层
定义了一些传输数据的协议和端口号（WWW端口80等），如：
TCP（transmission control protocol –传输控制协议，传输效率低，可靠性强，用于传输可靠性要求高，数据量大的数据）
UDP（user datagram protocol–用户数据报协议，与TCP特性恰恰相反，用于传输可靠性要求不高，数据量小的数据，如QQ聊天数据就是通过这种方式传输的）。 主要是将从下层接收的数据进行分段和传输，到达目的地址后再进行重组。常常把这一层数据叫做段。

5、会话层
通过运输层（端口号：传输端口与接收端口）建立数据传输的通路。主要在你的系统之间发起会话或者接受会话请求（设备之间需要互相认识可以是IP也可以是MAC或者是主机名）

6、表示层
可确保一个系统的应用层所发送的信息可以被另一个系统的应用层读取。例如，PC程序与另一台计算机进行通信，其中一台计算机使用扩展二一十进制交换码（EBCDIC），而另一台则使用美国信息交换标准码（ASCII）来表示相同的字符。如有必要，表示层会通过使用一种通格式来实现多种数据格式之间的转换。

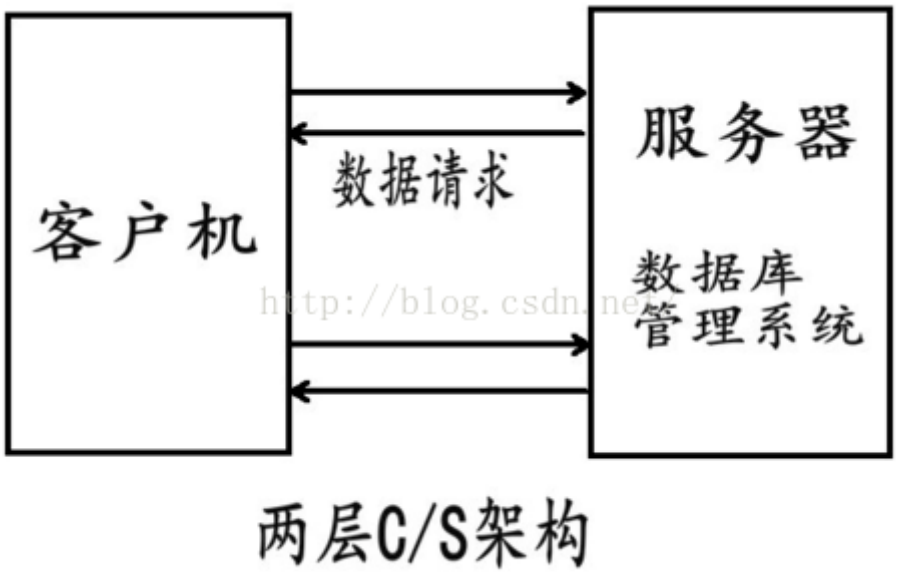
7、应用层
是最靠近用户的OSI层。这一层为用户的应用程序（例如电子邮件、文件传输和终端仿真）提供网络服务。

2、什么是C/S和B/S架构?

\1. C/S架构及其背景

C/S架构是一种比较早的软件架构，主要应用于局域网内。在这之前经历了集中计算模式，随着计算机网络的进步与发展，尤其是可视化工具的应用，出现过两层C/S和三层C/S架构，不过一直很流行也比较经典的是我们所要研究的两层C/S架构。

C/S架构软件（即客户机/服务器模式）分为客户机和服务器两层：第一层是在客户机系统上结合了表示与业务逻辑，第二层是通过网络结合了数据库服务器。简单的说就是第一层是用户表示层，第二层是数据库层。客户端和服务器直接相连，这两个组成部分都承担着重要的角色。



两层C/S架构

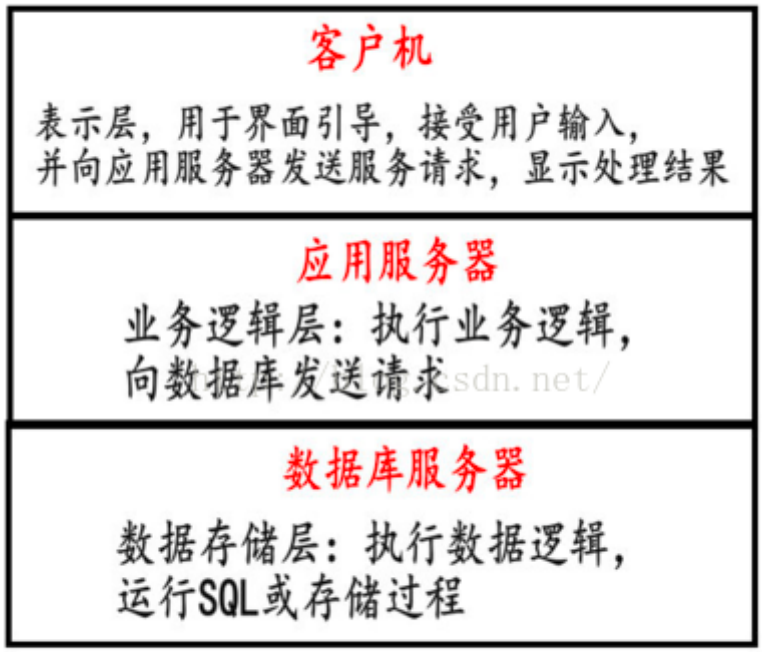
```
2．C/S架构的优点
a．客户端和服务器直接相连。点对点的连接方式更安全，可以直接操作本地文本，比较方便。
b．客户端可以处理一些逻辑事务。可以进行数据处理和数据存储，提供一定的帮助。
c．客户端直接操作界面。

3．C/S架构的缺点
a> C/S架构适用于局域网，对网速的要求比较高。
b> 客户端界面缺乏通用性，且当业务更改时就需要更改界面，重新编写。
c> 随着用户数量的增多，会出现通信拥堵、服务器响应速度慢等情况。
d> 系统的维护也比较麻烦。

4．C/S架构的应用
C/S架构的软件是在是数不胜数，从办公的OFFICE，WPS，WINRAR到杀毒软件如金山，瑞金再到我们的娱乐软件，如播放器，QQ，微信等，无处不见C/S架构。
```

\1. B/S架构及其背景 随着Internet和WWW的流行，以往的主机/终端和C/S都无法满足当前的全球网络开放、互连、信息随处可见和信息共享的新要求，于是就出现了B/S型模式，即浏览器/服务器结构。它是C/S架构的一种改进，可以说属于三层C/S架构。主要是利用了不断成熟的WWW浏览器技术，用通用浏览器就实现了原来需要复杂专用软件才能实现的强大功能，并节约了开发成本，是一种全新的软件系

统构造技术。



三层C/S架构

第一层是浏览器（即客户端）只有简单的输入输出功能，处理极少部分的事务逻辑。由于客户不需要安装客户端，只要有浏览器就能上网浏览，所以它面向的是大范围的用户，所以界面设计得比较简单，通用。 第二层是WEB服务器，扮演着信息传送的角色。当用户想要访问数据库时，就会首先向WEB服务器发送请求，WEB服务器统一请求后会向数据库服务器发送访问数据库的请求，这个请求是以SQL语句实现的。 第三层是数据库服务器，它存放着大量的数据。当数据库服务器收到了WEB服务器的请求后，会对SQL语句进行处理，并将返回的结果发送给WEB服务器，接下来，WEB服务器将收到的数据结果转换为HTML文本形式发送给浏览器。

2．B/S架构的优点
a> 浏览器和数据库服务器采用多对多的方式连接。因此适合在广域网里实现巨大的互联网，甚至是全球网，有着很强大的信息共享性。
b> 浏览器只处理一些简单的逻辑事务，负担小。
c> 数据都集中存放在数据库服务器，所以不存在数据不一致现象。
d> 随着服务器负载的增加，可以平滑地增加服务器的个数并建立集群服务器系统，然后在各个服务器之间做负载均衡。
e> B/S建立在广域网上，所以需要的网速要求不高。
f> 不需要安装客户端，只要能连上网，就能随时随地的浏览页面。
g> 能有效地保护数据平台和管理访问权限，确保服务器数据库的数据安全。

3．B/S架构的缺点
a> 服务器承担着重要的责任，数据负荷较重。一旦发生服务器“崩溃”等问题，后果不堪设想。
b> 页面需要不断地动态刷新，当用户增多时，网速会变慢。

4．B/S架构的应用
比如WEBQQ，从WEBQQ名称中的WEB就不难看出它属于B/S架构，是一种浏览器服务器结构。事实上也是如此，因为WEBQQ根本不需要安装客户端，只需要有浏览器就可以进行聊天交互了。

3、简述 三次握手、四次挥手的流程。

三次握手：

初始状态：客户端A和服务器B均处于CLOSED状态，然后服务器B创建socket，调用监听接口使得服务器处于LISTEN状态，等待客户端连接。（后续内容用A，B简称代替）

1、A首先向B发起连接，这时TCP头部中的SYN标识位值为1，然后选定一个初始序号seq=x（一般是随机的），消息发送后，A进入SYN_SENT状态，SYN=1的报文段不能携带数据，但要消耗一个序号。

2、B收到A的连接请求后，同意建立连接，向A发送确认数据，这时TCP头部中的SYN和ACK标识位值均为1，确认序号为ack=x+1，然后选定自己的初始序号seq=y（一般是随机的），确认消息发送后，B进入SYN_RCVD状态，与连接消息一样，这条消息也不能携带数据，同时消耗一个序号。

3、A收到B的确认消息后，需要给B回复确认数据，这时TCP头部中的ACK标识位值为1，确认序号是ack=y+1，自己的序号在连接请求的序号上加1，也就是seq=x+1，此时A进入ESTABLISHED状态，当B收到A的确认回复后，B也进入ESTABLISHED状态，至此TCP成功建立连接，A和B之间就可以通过这个连接互相发送数据了。

四次挥手：

初始状态：客户端A和服务器B之间已经建立了TCP连接，并且数据发送完成，打算断开连接，此时客户端A和服务器B是等价的，双方都可以发送断开请求，下面以客户端A主动发起断开请求为例。（后续内容用A，B简称代替）

1、A首先向B发送断开连接消息，这时TCP头部中的FIN标识位值为1，序号是seq=m，m为A前面正常发送数据最后一个字节序号加1得到的，消息发送后A进入FNI_WAIT_1状态，FIN=1的报文段不能携带数据，但要消耗一个序号。

2、B收到A的断开连接请求需要发出确认消息，这时TCP头部中的ACK标识位值为1，确认号为ack=m+1，而自己的序号为seq=n,n为B前面正常发送数据最后一个字节序号加1得到的，然后B进入CLOSE_WAIT状态，此时就关闭了A到B的连接，A无法再给B发数据，但是B仍然可以给A发数据（此处存疑），同时B端通知上方应用层，处理完成后被动关闭连接。然后A收到B的确认信息后，就进入了FIN_WAIT_2状态。

3、B端应用层处理完数据后，通知关闭连接，B向A发送关闭连接的消息，这时TCP头部中的FIN和ACK标识位值均为1，确认号ack=m+1，自己的序号为seq=k，（B发出确认消息后有发送了一段数据，此处存疑），消息发送后B进入LACK_ACK状态。

4、A收到B的断开连接的消息后，需要发送确认消息，这是这时TCP头部中的ACK标识位值为1，确认号ack=k+1，序号为m+1（因为A向B发送断开连接的消息时消耗了一个消息号），然后A进入TIME_WAIT状态，若等待时间经过2MSL后，没有收到B的重传请求，则表明B收到了自己的确认，A进入CLOSED状态，B收到A的确认消息后则直接进入CLOSED状态。至此TCP成功断开连接。

4、什么是arp协议?

ARP（Address Resolution Protocol）即地址解析协议， 用于实现从 IP 地址到 MAC 地址的映射，即询问目标IP对应的MAC地址。

详细见 https://www.cnblogs.com/csguo/p/7527303.html

5、TCP和UDP的区别?

```
TCP:
TCP编程的服务器端一般步骤是:
    1、创建一个socket，用函数socket();
    2、设置socket属性，用函数setsockopt(); * 可选
    3、绑定IP地址、端口等信息到socket上，用函数bind();
    4、开启监听，用函数listen();
    5、接收客户端上来的连接，用函数accept();
    6、收发数据，用函数send()和recv()，或者read()和write();
    7、关闭网络连接;
    8、关闭监听;
```

```
UDP:
与之对应的UDP编程步骤要简单许多，分别如下:
    UDP编程的服务器端一般步骤是:
    1、创建一个socket，用函数socket();
    2、设置socket属性，用函数setsockopt();* 可选
    3、绑定IP地址、端口等信息到socket上，用函数bind();
    4、循环接收数据，用函数recvfrom();
    5、关闭网络连接;
```

TCP与UDP基本区别

1. 基于连接与无连接
2. TCP要求系统资源较多，UDP较少
3. UDP程序结构较简单
4. 字节流模式（TCP）与数据报模式(UDP);
5. TCP保证数据正确性，UDP可能丢包
6. TCP保证数据顺序，UDP不保证

具体编程时的区别 1.socket()的参数不同　　2.UDP Server不需要调用listen和accept　　3.UDP收发数据用sendto/recvfrom函数　　4.TCP：地址信息在connect/accept时确定　　5.UDP：在sendto/recvfrom函数中每次均 需指定地址信息　　6.UDP：shutdown函数无效

6、什么是局域网和广域网?

局域网：（Local Area Network，LAN），局域网是一个局部范围的计算计组，比如家庭网络就是一个小型的局域网，里面包含电脑、手机和平板等，他们共同连接到你家的路由器上。又比如学校的机房就是一个局域网，里面有几百几千台电脑，当机房无法上外网时，但是电脑之间仍可以通信，你们可以通过这个局域网来打CS 、玩红警。理论上，局域网是封闭的，并不可以上外网，可以只有两台电脑，也可以有上万台。

广域网：（WAN，Wide Area Network），广域网的范围就比较大了，可以把你家和别人家、各个省、各个国家连接起来相互通信。广域网和局域网都是从范围的角度来划分的，广域网也可以看成是很多个局域网通过路由器等相互连接起来。

7、为何基于tcp协议的通信比基于udp协议的通信更可靠?

TCP的可靠保证，是它的三次握手双向机制，这一机制保证校验了数据，保证了他的可靠性。而UDP就没有了，udp信息发出后,不验证是否到达对方,所以不可靠。不过UDP的发送速度是TCP比不了的，而且UDP的反应速度更快。

8、什么是socket？简述基于tcp协议的套接字通信流程。

套接字，也称为BSD套接字，是支持TCP/IP的网络通信的基本操作单元，可以看做是不同主机之间的进程进行双向通信的端点，简单的说就是通信的两方的一种约定，用套接字中的相关函数来完成通信过程。应用层通过传输层进行数据通信时，TCP和UDP会遇到同时为多个应用程序进程提供并发服务的问题。

1.服务器先用 socket 函数来建立一个套接字,用这个套接字完成通信的监听。 2.用 bind 函数来绑定一个端口号和 IP 地址。因为本地计算机可能有多个网址和 IP,每一个 IP 和端口有多个端口。需要指定一个 IP 和端口进行监听。 3.服务器调用 listen 函数,使服务器的这个端口和 IP 处于监听状态,等待客户机的连接。 4.客户机用 socket 函数建立一个套接字,设定远程 IP 和端口。 5.客户机调用 connect 函数连接远程计算机指定的端口。 6.服务器用 accept 函数来接受远程计算机的连接,建立起与客户机之间的通信。 7.建立连接以后,客户机用 write 函数向 socket 中写入数据。也可以用 read 函数读取服务器发送来的数据。 8.服务器用 read 函数读取客户机发送来的数据,也可以用 write 函数来发送数据。 9.完成通信以后,用 close 函数关闭 socket 连接。

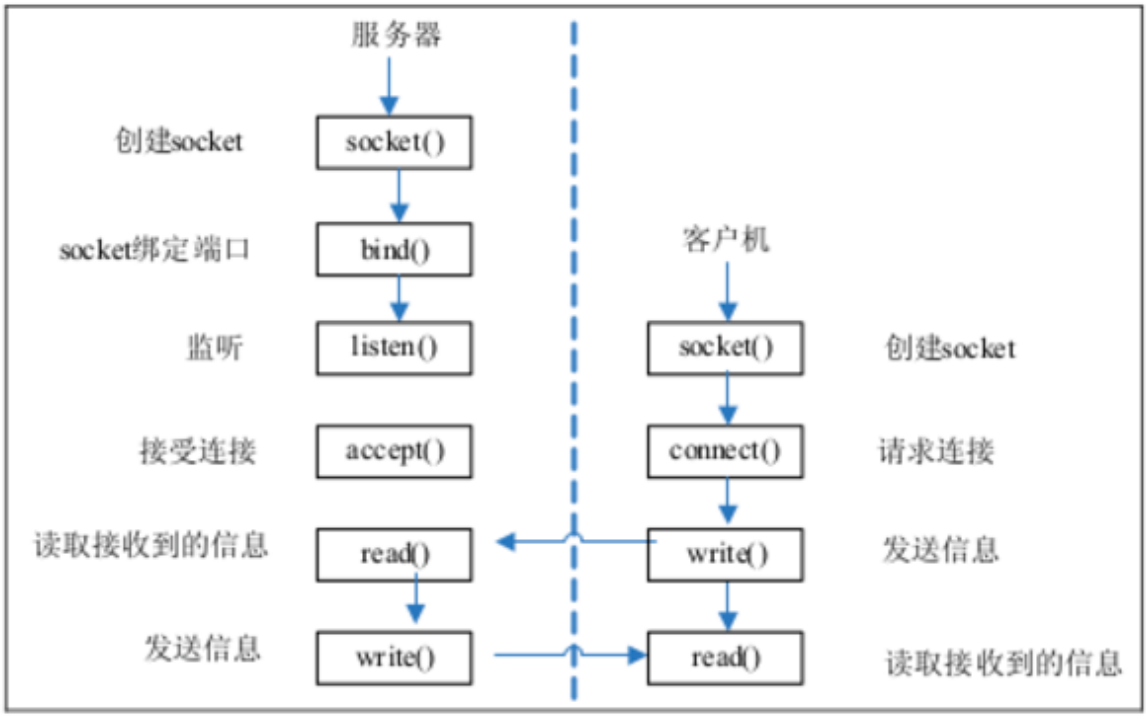
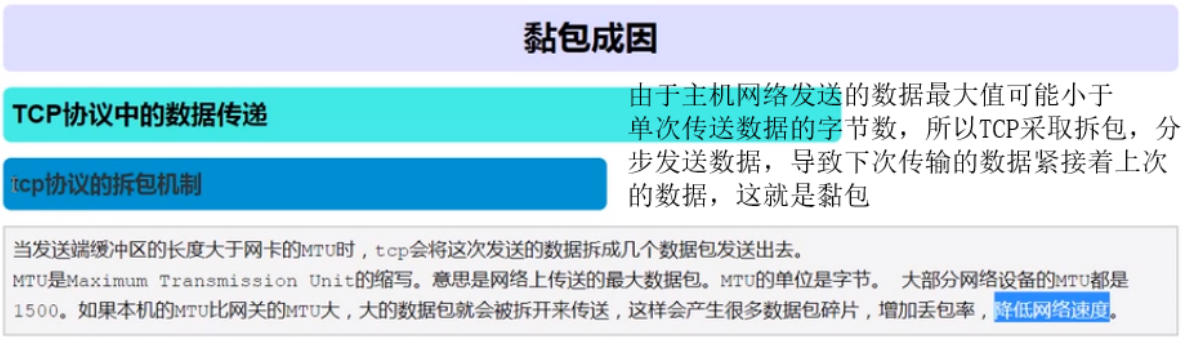


图 13-1　面向连接的套接字通信工作流程

9、什么是粘包？ socket 中造成粘包的原因是什么？ 哪些情况会发生粘包现象？



10、IO多路复用的作用?

一、基本概念

IO多路复用是指内核一旦发现进程指定的一个或者多个IO条件准备读取，它就通知该进程。IO多路复用适用如下场合：

（1）当客户处理多个描述字时（一般是交互式输入和网络套接口），必须使用I/O复用。

（2）当一个客户同时处理多个套接口时，而这种情况是可能的，但很少出现。

（3）如果一个TCP服务器既要处理监听套接口，又要处理已连接套接口，一般也要用到I/O复用。

（4）如果一个服务器即要处理TCP，又要处理UDP，一般要使用I/O复用。

（5）如果一个服务器要处理多个服务或多个协议，一般要使用I/O复用。

与多进程和多线程技术相比，I/O多路复用技术的最大优势是系统开销小，系统不必创建进程/线程，也不必维护这些进程/线程，从而大大减小了系统的开销。

请关注，未完待续!