# socket简介

https://blog.csdn.net/xiaoquantouer/article/details/58001960

即套接字，，用于描述地址和端口，是一个通信链的句柄。应用程序通过socket向网络发出请求或者回应。

sockets(套接字)编程有三种，流式套接字（SOCK\_STREAM），数据报套接字（SOCK\_DGRAM），原始套接字（SOCK\_RAW）；前两种较常用。基于TCP的socket编程是采用的流式套接字。

（1）SOCK\_STREAM表示面向连接的数据传输方式。数据可以准确无误地到达另一台计算机，如果损坏或丢失，可以重新发送，但效率相对较慢。常用的HTTP协议就使用SOCK\_STREAM传输数据，因为要确保数据的正确性，否则网页不能正常解析。

（2）SOCK\_DGRAM表示无连接的数据传输方式。计算机只管传输数据，不作数据校验，如果数据在传输中损坏，或者没有到达另一台计算机，是没有办法补救的。也就是说，数据错了就错了，无法重传。因为SOCK\_DGRAM所做的校验工作少，所以效率比SOCK\_STREAM高。

QQ视频聊天和语音聊天就使用SOCK\_DGRAM传输数据，因为首先要保证通信的效率，尽量减小延迟，而数据的正确性是次要的，即使丢失很小的一部分数据，视频和音频也可以正常解析，最多出现噪点或杂音，不会对通信质量有实质的影响。

注意：SOCK\_DGRAM没有想象中的糟糕，不会频繁的丢失数据，数据错读只是小概率事件。

有可能多种协议使用同一种数据传输方式，所以在socket编程中，需要同时指明数据传输方式和协议。

二、客户端/服务端模式：

在TCP/IP网络应用中，通信的两个进程相互作用的主要模式是客户/服务器模式，即客户端向服务器发出请求，服务器接收请求后，提供相应的服务。客户/服务器模式的建立基于以下两点：

（1）建立网络的起因是网络中软硬件资源、运算能力和信息不均等，需要共享，从而就让拥有众多资源的主机提供服务，资源较少的客户请求服务这一非对等作用。

（2）网间进程通信完全是异步的，相互通信的进程间既不存在父子关系，又不共享内存缓冲区。

因此需要一种机制为希望通信的进程间建立联系，为二者的数据交换提供同步，这就是基于客户/服务端模式的TCP/IP。

服务端：建立socket，声明自身的端口号和地址并绑定到socket，使用listen打开监听，然后不断用accept去查看是否有连接，如果有，捕获socket，并通过recv获取消息的内容，通信完成后调用closeSocket关闭这个对应accept到的socket，如果不再需要等待任何客户端连接，那么用closeSocket关闭掉自身的socket。

客户端：建立socket，通过端口号和地址确定目标服务器，使用Connect连接到服务器，send发送消息，等待处理，通信完成后调用closeSocket关闭socket。

三、编程步骤

（1）服务端

1、加载套接字库，创建套接字（WSAStartup()/socket()）;

2、绑定套接字到一个IP地址和一个端口上（bind()）;

3、将套接字设置为监听模式等待连接请求（listen()）；

4、请求到来后，接受连接请求，返回一个新的对应于此次连接的套接字（accept()）；

5、用返回的套接字和客户端进行通信（send()/recv()）；

6、返回，等待另一个连接请求；

7、关闭套接字，关闭加载的套接字库（closesocket()/WSACleanup()）；

（2）客户端

1、加载套接字库，创建套接字（WSAStartup()/socket()）；

2、向服务器发出连接请求（connect()）；

3、和服务器进行通信（send()/recv()）；

4、关闭套接字，关闭加载的套接字库（closesocket()/WSACleanup()）；

————————————————

版权声明：本文为CSDN博主「小拳头」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/xiaoquantouer/article/details/58001960