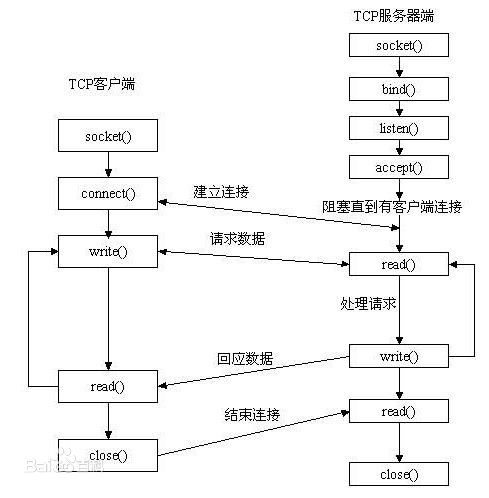
<https://blog.csdn.net/min996358312/article/details/62231033>

Socket解释：

网络上的两个程序通过一个双向的通信连接实现数据的交换，这个连接的一端称为一个socket。

Socket的英文原义是“孔”或“插座”。作为BSD UNIX的进程通信机制，取后一种意思。通常也称作"套接字"，用于描述IP地址和端口，是一个通信链的句柄，可以用来实现不同虚拟机或不同计算机之间的通信。在Internet上的主机一般运行了多个服务软件，同时提供几种服务。每种服务都打开一个Socket，并绑定到一个端口上，不同的端口对应于不同的服务。Socket正如其英文原意那样，像一个多孔插座。一台主机犹如布满各种插座的房间，每个插座有一个编号，有的插座提供220伏交流电， 有的提供110伏交流电，有的则提供有线电视节目。 客户软件将插头插到不同编号的插座，就可以得到不同的服务。



socket交互原理图解

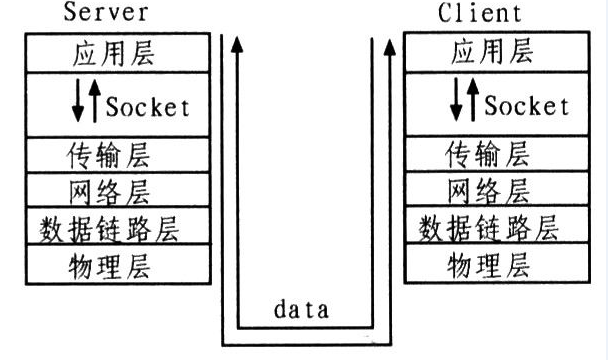
连接过程：

根据连接启动的方式以及本地[套接字](http://baike.baidu.com/view/538713.htm)要连接的目标，套接字之间的连接过程可以分为三个步骤：[服务器](http://baike.baidu.com/view/899.htm)监听，客户端请求，连接确认。

（1）服务器监听：是服务器端套接字并不定位具体的客户端套接字，而是处于等待连接的状态，实时监控网络状态。

（2）客户端请求：是指由客户端的套接字提出连接请求，要连接的目标是服务器端的套接字。为此，客户端的套接字必须首先描述它要连接的服务器的套接字，指出服务器端套接字的地址和[端口号](http://baike.baidu.com/view/642103.htm)，然后就向服务器端套接字提出连接请求。

（3）连接确认：是指当服务器端套接字监听到或者说接收到客户端套接字的连接请求，它就响应客户端套接字的请求，建立一个新的线程，把[服务器](http://baike.baidu.com/view/899.htm)端套接字的描述发给客户端，一旦客户端确认了此描述，连接就建立好了。而服务器端[套接字](http://baike.baidu.com/view/538713.htm)继续处于[监听状态](http://baike.baidu.com/view/1624776.htm)，继续接收其他客户端套接字的连接请求。



socket连接过程图解

下面是我用socket实现简单的服务器客户端交互demo：

Server端:

复制代码

1 　　　　　　try {

2 //创建绑定到特定端口的服务器套接字

3 ServerSocket serverSocket=new ServerSocket(9999);

4 System.out.println("\*\*\*服务器即将启动，等待客户端链接\*\*\*");

5 //侦听并接受到此套接字的连接

6 Socket socket=serverSocket.accept();

7 //获取输入流

8 InputStream is=socket.getInputStream();

9 //将字节输入流转为字符输入流

10 InputStreamReader isr=new InputStreamReader(is);

11 BufferedReader br=new BufferedReader(isr);

12 String str=null;

13 while((str=br.readLine())!=null){

14 System.out.println("服务器说：客户端发送了："+str);

15 br.readLine();

16 }

17 //此套接字的输入流置于"流的末尾"

18 socket.shutdownInput();

19 br.close();

20 isr.close();

21 is.close();

22 } catch (IOException e) {

23 e.printStackTrace();

24 }

复制代码

Client端:

复制代码

1 　　　　　try {

2 Socket socket=new Socket("127.0.0.1",9999);

3 OutputStream os=socket.getOutputStream();

4 PrintWriter print=new PrintWriter(os);

5 print.write("你好啊！服务器！");

6 print.close();

7 os.close();

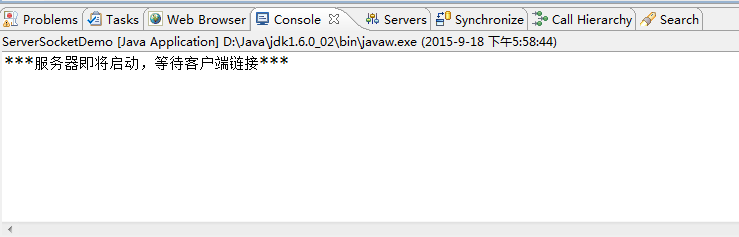
8 } catch (Exception e) {

9 e.printStackTrace();

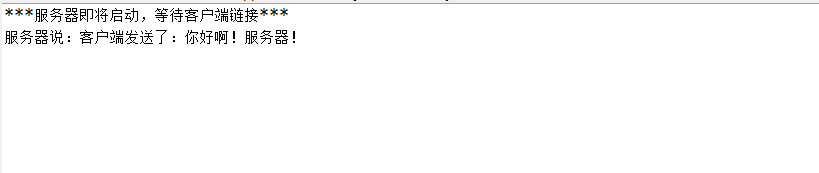
10 }

复制代码

首先需要启动服务器端，创建ServerSocket等待客户端连接：



然后，启动客户端：



这样，一个简单的Socket通信就实现了！

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/min996358312/article/details/62231033