**1.什么是socket**

　　所谓socket通常也称作"套接字"，应用程序通常通过"套接字"向网络发出请求或者应答网络请求。以J2SDK-1.3为例，Socket和ServerSocket类库位于java .net包中。ServerSocket用于服务器端，Socket是建立网络连接时使用的。在连接成功时，应用程序两端都会产生一个Socket实例，操作这个实例，完成所需的会话。对于一个网络连接来说，套接字是平等的，并没有差别，不因为在服务器端或在[客户端](http://baike.baidu.com/view/930.htm" \t "_blank)而产生不同级别。不管是Socket还是ServerSocket它们的工作都是通过SocketImpl类及其子类完成的。

**重要的Socket API**

　　重要的Socket API：java .net.Socket继承于java.lang.Object，有八个构造器，其方法并不多，下面介绍使用最频繁的三个方法，其它方法大家可以见JDK-1.3文档。

　　Accept方法用于产生"阻塞"，直到接受到一个连接，并且返回一个客户端的Socket[对象](http://baike.baidu.com/view/2387.htm)实例。"阻塞"是一个术语，它使程序运行暂时"停留"在这个地方，直到一个会话产生，然后程序继续；通常"阻塞"是由循环产生的。

　　getInputStream方法获得网络连接输入，同时返回一个OutputStream对象实例。

　　getOutputStream方法连接的另一端将得到输入，同时返回一个OutputStream对象实例。注意：其中getInputStream和getOutputStream方法均可能会产生一个IOException，它必须被捕获，因为它们返回的流对象，通常都会被另一个流对象使用。

**SOCKET连接过程**

　　根据连接启动的方式以及本地套接字要连接的目标，

      套接字之间的连接过程可以分为三个步骤：服务器监听，客户端请求，连接确认。

　　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。

　　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，然后就向服务器端套接字提出连接请求。

　　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。而服务器端套接字继续处于监听状态，继续接收其他客户端套接字的连接请求。

**如何开发一个Server-Client模型的程序**

　　开发原理：

　　服务器，使用ServerSocket监听指定的端口，端口可以随意指定（由于1024以下的端口通常属于保留端口，在一些操作系统中不可以随意使用，所以建议使用大于1024的端口），等待客户连接请求，客户连接后，会话产生；在完成会话后，关闭连接。

　　客户端，使用Socket对网络上某一个服务器的某一个端口发出连接请求，一旦连接成功，打开会话；会话完成后，关闭Socket。客户端不需要指定打开的端口，通常临时的、动态的分配一个1024以上的端口。

　　Socket接口是TCP/IP网络的API，Socket接口定义了许多函数或例程，[程序员](http://baike.baidu.com/view/39175.htm" \t "_blank)可以用它们来开发TCP/IP网络上的应用程序。要学Internet上的[TCP/IP网络编程](http://baike.baidu.com/view/3058831.htm" \t "_blank)，必须理解Socket接口。Socket接口设计者最先是将接口放在Unix操作系统里面的。如果了解Unix系统的输入和输出的话，就很容易了解Socket了。网络的Socket数据传输是一种特殊的I/O，Socket也是一种[文件描述符](http://baike.baidu.com/view/1303430.htm" \t "_blank)。Socket也具有一个类似于打开文件的函数调用Socket（），该函数返回一个整型的Socket描述符，随后的连接建立、数据传输等操作都是通过该Socket实现的。

**常用的Socket类型**

　　有两种：流式Socket（SOCK\_STREAM）和数据报式Socket（SOCK\_DGRAM）。流式是一种面向连接的Socket，针对于面向连接的TCP服务应用；数据报式Socket是一种无连接的Socket，对应于无连接的UDP服务应用。Socket为了建立Socket，程序可以调用Socket函数，该函数返回一个类似于文件描述符的句柄。socket函数原型为：int socket(int domain,int type,int protocol);domain指明所使用的协议族，通常为PF\_INET，(其与addrinfo 里的 AF\_INET在现在看来是相同的。只是历史上人们曾构想将AF（地址家族address family）与PF(protocol family 协议家族)分开，但实际上这种区分并未真正推广，所以现在AF\_INET和PF\_INET具有相同的意义。其中AF\_INET是基于IPv4而AF\_INET基于IPv6）表示互联网协议族（[TCP/IP协议族](http://baike.baidu.com/view/2221037.htm" \t "_blank)）；type[参数](http://baike.baidu.com/view/327406.htm)指定socket的类型：SOCK\_STREAM 或SOCK\_DGRAM，Socket接口还定义了原始Socket（SOCK\_RAW），允许程序使用低层协议；protocol通常赋值0。Socket（）调用返回一个整型socket描述符，你可以在后面的调用使用它。Socket描述符是一个指向内部数据结构的[指针](http://baike.baidu.com/view/159417.htm)，它指向描述符表入口。调用Socket函数时，socket执行体将建立一个Socket，实际上"建立一个Socket"意味着为一个Socket数据结构分配存储空间。Socket执行体为你管理描述符表。两个网络程序之间的一个网络连接包括五种信息：通信协议、本地[协议地址](http://baike.baidu.com/view/1252611.htm" \t "_blank)、本地主机端口、远端[主机地址](http://baike.baidu.com/view/547482.htm)和远端协议端口。Socket数据结构中包含这五种信息。socket在测量软件中的使用也很广泛

[编辑本段](http://baike.baidu.com/view/13870.htm)**socket函数**

**简介**

　　SOCKET socket( int af,int type,int protocol );

　　应用程序调用socket函数来创建一个能够进行网络通信的套接字。

　　第一个参数指定应用程序使用的通信协议的协议族，对于TCP/IP协议族，该参数置AF\_INET;

　　第二个参数指定要创建的套接字类型，流套接字类型为SOCK\_STREAM、数据报套接字类型为SOCK\_DGRAM、原始套接字SOCK\_RAW(WinSock接口并不适用某种特定的协议去封装它，而是由程序自行处理数据包以及协议首部）；

　　第三个参数指定应用程序所使用的通信协议。

　　该函数如果调用成功就返回新创建的套接字的描述符，如果失败就返回INVALID\_SOCKET。套接字描述符是一个整数类型的值。每个进程的进程空间里都有一个套接字描述符表，该表中存放着套接字描述符和套接字数据结构的对应关系。该表中有一个字段存放新创建的套接字的描述符，另一个字段存放套接字数据结构的地址，因此根据套接字描述符就可以找到其对应的套接字数据结构。每个进程在自己的进程空间里都有一个套接字描述符表但是套接字数据结构都是在操作系统的内核缓冲里。

　　下面是一个创建流套接字的例子：

　　struct protoent \*ppe;

　　ppe=getprotobyname("tcp");

　　SOCKET ListenSocket=socket(PF\_INET,SOCK\_STREAM,ppe->p\_proto);