蓝鸥郑州技术分享：你不知道的Socket网络通讯都在这里

原创 2016-03-04 神英强 [蓝鸥](javascript:void(0);)

Socket方面的技术更偏技术的中高阶段了，很多已经毕业的学员，特别是工作一年以上的同学都觉得Socket是非常有用和值得学习的。

正好周三的时候，有一个同学提出想看Socket方面的技术文章，一拍即合，这篇文章应运而生。

1

套接字（Socket）概念

socket起源于UNIX，在Unix一切皆文件哲学的思想下，socket是一种"打开—读/写—关闭"模式的实现，服务器和客户端各自维护一个"文件"，在建立连接打开后，可以向自己文件写入内容供对方读取或者读取对方内容，通讯结束时关闭文件。

我们经常把socket翻译为套接字，socket是在应用层和传输层之间的一个抽象层，它把TCP/IP层复杂的操作抽象为几个简单的接口供应用层调用已实现进程在网络中通信。

套接字（socket）是通信的基石，是支持TCP/IP协议的网络通信的基本操作单元。

它是网络通信过程中端点的抽象表示，包含进行网络通信必须的五种信息：连接使用的协议，本地主机的IP地址，本地进程的协议端口，远地主机的IP地址，远地进程的协议端口。

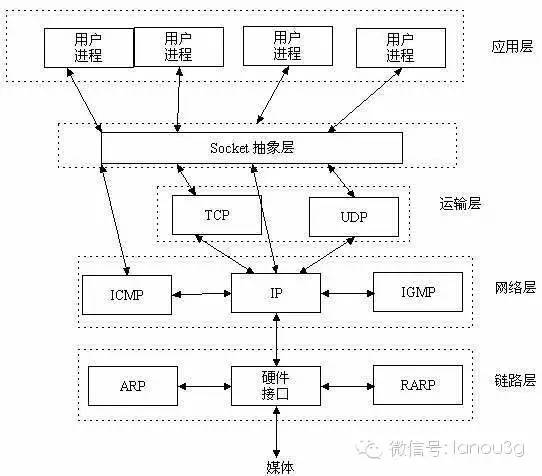
应 用层通过传输层进行数据通信时，TCP会遇到同时为多个应用程序进程提供并发服务的问题。

多个TCP连接或多个应用程序进程可能需要通过同一个 TCP协议端口传输数据。

为了区别不同的应用程序进程和连接，许多计算机操作系统为应用程序与TCP／IP协议交互提供了套接字(Socket)接口。

应用层可以和传输层通过Socket接口，区分来自不同应用程序进程或网络连接的通信，实现数据传输的并发服务。

**正如下图TCP/IP网络模型所示：**



 socket是应用层和传输层之间的桥梁

socket 通信流程

socket是"打开—读/写—关闭"模式的实现，建立Socket连接至少需要一对套接字，其中一个运行于客户端，称为ClientSocket，另一个运行于服务器端，称为ServerSocket。

套接字之间的连接过程分为三个步骤：服务器监听，客户端请求，连接确认。

服务器监听：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，等待客户端的连接请求。

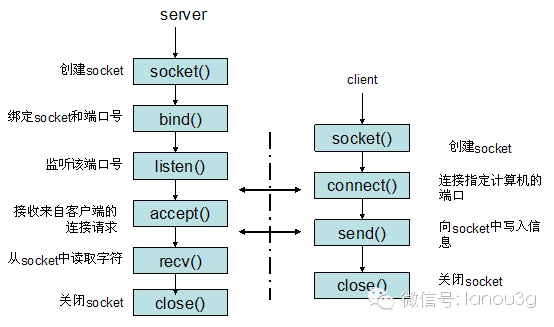
客户端请求：指客户端的套接字提出连接请求，要连接的目标是服务器端的套接字。

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：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时，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，双方就正式建立连接。

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。

以使用TCP协议通讯的socket为例，其交互流程大概是这样子的



 socket互流程(TCP)

socket连接与http连接

由于通常情况下Socket连接就是TCP连接，因此Socket连接一旦建立，通信双方即可开始相互发送数据内容，直到双方连接断开。

但在实际网 络应用中，客户端到服务器之间的通信往往需要穿越多个中间节点，例如路由器、网关、防火墙等，大部分防火墙默认会关闭长时间处于非活跃状态的连接而导致 Socket 连接断连，因此需要通过轮询告诉网络，该连接处于活跃状态。

而HTTP连接使用的是“请求—响应”的方式，不仅在请求时需要先建立连接，而且需要客户端向服务器发出请求后，服务器端才能回复数据。

Socket连接与TCP连接

创建Socket连接时，可以指定使用的传输层协议，Socket可以支持不同的传输层协议（TCP或UDP），当使用TCP协议进行连接时，该Socket连接就是一个TCP连接。

TCP与udp的区别

TCP：面向连接、传输可靠(保证数据正确性,保证数据顺序)、用于传输大量数据(流模式)、速度慢，建立连接需要开销较多(时间，系统资源)。

UDP：面向非连接、传输不可靠、用于传输少量数据(数据包模式)、速度快。

关于TCP是一种流模式的协议，UDP是一种数据报模式的协议，这里要说明一下，TCP是面向连接的，也就是说，在连接持续的过程中，socket中收到的数据都是由同一台主机发出的（劫持什么的不考虑）。

因此，知道保证数据是有序的到达就行了，至于每次读取多少数据自己看着办。

而UDP是无连接的协议，也就是说，只要知道接收端的IP和端口，且网络是可达的，任何主机都可以向接收端发送数据。

这时候，如果一次能读取超过一个报文的数据，则会乱套。

比如，主机A向发送了报文P1，主机B发送了报文P2，如果能够读取超过一个报文的数据，那么就会将P1和P2的数据合并在了一起，这样的数据是没有意义的。

通俗解释socket

socket是网络编程的基础，本文用打电话来类比socket通信中建立TCP连接的过程。

socket函数，表示你买了或者借了一部手机。

bind函数，告诉别人你的手机号码，让他们给你打电话。

listen函数，打开手机的铃声，而不是静音，这样有电话时可以立马反应。

listen函数的第二个参数，最大连接数，表示最多有几个人可以同时拨打你的号码。

不过我们的手机，最多只能有一个人打进来，要不然就提示占线。

connect函数，你的朋友知道了你的号码，通过这个号码来联系你。

在他等待你回应的时候，不能做其他事情，所以connect函数是阻塞的。

accept函数，你听到了电话铃声，接电话，accept it！然后“喂”一声，你的朋友听到你的回应，知道电话已经打进去了。

至此，一个TCP连接建立了。

read/write函数，连接建立后，TCP的两端可以互相收发消息，这时候的连接是全双工的。

对应打电话中的电话煲。

close函数，通话完毕，一方说“我挂了”，另一方回应"你挂吧"，然后将连接终止。

实际的close(sockfd)有些不同，它不止是终止连接，还把手机也归还，不在占有这部手机，就当是公用电话吧。

注意到，上述连接是阻塞的，你一次只能响应一个用户的连接请求，但在实际网络编程中，一个服务器服务于多个客户，上述方案也就行不通了，怎么办？

想一想1860，移动的声讯服务台，也是只有一个号码，它怎么能同时服务那么多人呢？

可以这样理解，在你打电话到1860时，总服务台会让一个接线员来为你服务，而它自己却继续监听有没有新的电话接入。

在网络编程中，这个过程类似于fork一个子进程，建立实际的通信连接，而主进程继续监听。

1860的接线员是有限的，所以当连接的人数达到上线时，它会放首歌给你听，忙等待，直到有新的空闲接线员为止。

下面我们基于第三方开元框架CocoaAsyncSocket基于TCP协议模拟搭建服务器socket和客户端socket实现建立连接发送接收消息功能。

（CocoaAsyncSokcet框架下载链接：https://github.com/robbiehanson/CocoaAsyncSocket）

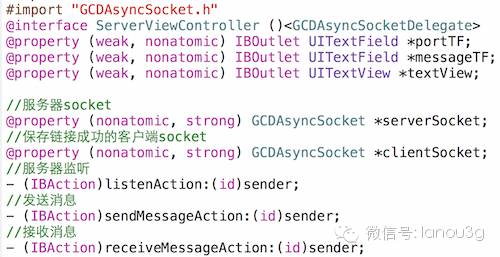
首先搭建服务器代码：

服务器界面如图所示：



服务器界面

服务器端代码：

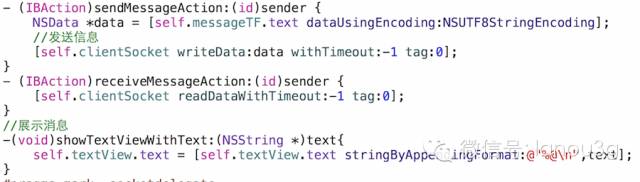


点击服务器监听后创建服务器socket代码：创建服务器socket监听的时候必须指定开放的端口号，客户端socket才能通过服务器地址（我们默认127.0.0.1）和端口号查找连接。



发送接收消息方法：注意此时write跟read必须是服务器为连接过来的客户端生产的socket，指定给哪个客户端socket发送或者读消息。

**注意：客户端socket是服务器socket代理方法中监听到客户端连接为客户端生产的**



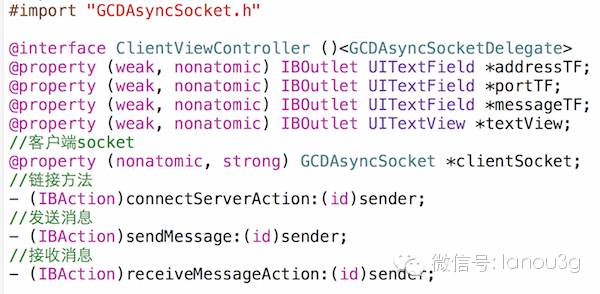
服务器socket代理方法：



客户端界面如图所示：



客户端代码：



点击客户端连接方法后创建客户端socket代码：创建客户端socket连接的时候必须指定服务器地址和服务的端口号，才能连接成功服务器端监听的socket。



客户端调用自己的socket发送读取数据，客户端socket代理方法中获取连接是否成功以及接收发送消息的成功回调。



参考代码下载链接：https://github.com/syq1122/AsynSocketDemoSYQ.git

如果问题，请加QQ3328469990，备注写解决问题，和我单独沟通哦。

原文链接地址：http://www.jianshu.com/p/13e6a943b3f5