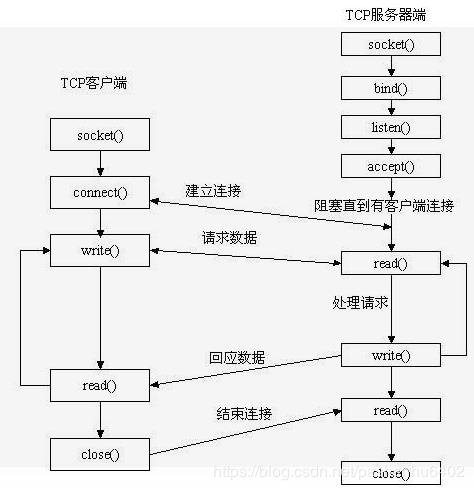
**socket套接字**

# 定义

Socket是应用层与TCP/IP协议族通信的中间软件抽象层，它是一组接口。在设计模式中，Socket其实就是一个门面模式，它把复杂的TCP/IP协议族隐藏在Socket接口后面，对用户来说，一组简单的接口就是全部，让Socket去组织数据，以符合指定的协议。

上socket经典图：



从这图可以很明确的看出socket套接字的工作原理：先从服务器端说起。服务器端先初始化Socket，然后与端口绑定(bind)，对端口进行监听(listen)，调用accept阻塞，等待客户端连接。在这时如果有个客户端初始化一个Socket，然后连接服务器(connect)，如果连接成功，这时客户端与服务器端的连接就建立了。客户端发送数据请求，服务器端接收请求并处理请求，然后把回应数据发送给客户端，客户端读取数据，最后关闭连接，一次交互结束。

# API 简单介绍

## 创建socket描述符

**int socket(int domain, int type, int protocol);**

**domain**：即协议域，又称为协议族（family）。

常用的地址族有：

* AF\_INET
* AF\_INET6
* AF\_LOCAL（AF\_UNIX，本地通信用）
* AF\_ROUTE

**type**：信息传送方式。

* SOCK\_STREAM
* SOCK\_DGRAM
* SOCK\_RAW
* SOCK\_PACKET
* SOCK\_SEQPACKET

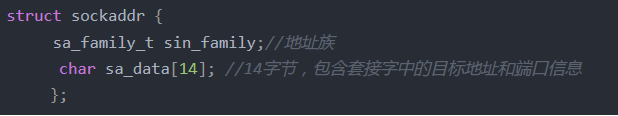
**protocol**：对应协议。

* IPPROTO\_TCP TCP传输协议
* IPPROTO\_UDP UDP传输协议
* IPPROTO\_SCTP STCP传输协议
* IPPROTO\_TIPCTIPC传输协议  
  **通常设置为0,让其自动匹配。**

## 绑定实际地址

**int bind(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);**

**sockfd** 一般服务端才需要绑定，客户端由系统内核解决  
**addr** 所有协议都有一个公共的结构叫做const struct sockaddr,不同协议对应不同的具体结



## 主机监听

**int listen(int sockfd, int backlog);**

listen函数的第一个参数即为要监听的socket描述字，第二个参数为相应socket可以排队的最大连接个数。socket()函数创建的socket默认是一个主动类型的，listen函数将socket变为被动类型的，等待客户的连接请求。

## 从机链接

**int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);**

connect函数的第一个参数即为客户端的socket描述字，第二参数为服务器的socket地址，第三个参数为socket地址的长度。客户端通过调用connect函数来建立与TCP服务器的连接。

## **建立链接**

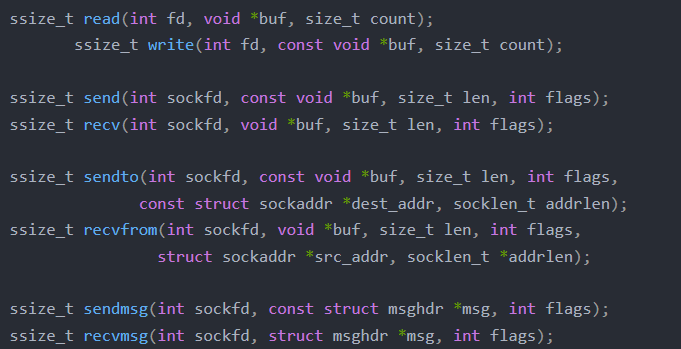
**int accept(int sockfd, struct sockaddr \*addr, socklen\_t \*addrlen);**

第一个参数为服务器的socket描述字，

第二个参数为指向struct sockaddr \*的指针，用于返回客户端的协议地址

第三个参数为协议地址的长度。

## read和write函数，读写



read函数是负责从fd中读取内容.当读成功时，read返回实际所读的字节数，如果返回的值是0表示已经读到文件的结束了，小于0表示出现了错误。如果错误为EINTR说明读是由中断引起的，如果是ECONNREST表示网络连接出了问题。

write函数将buf中的nbytes字节内容写入文件描述符fd.成功时返回写的字节 数。失败时返回-1，并设置errno变量。在网络程序中，当我们向套接字文件描述符写时有俩种可能。1)write的返回值大于0，表示写了部分或者是 全部的数据。2)返回的值小于0，此时出现了错误。我们要根据错误类型来处理。如果错误为EINTR表示在写的时候出现了中断错误。如果为EPIPE表示 网络连接出现了问题(对方已经关闭了连接)。

## 关闭服务

**int close(int fd);**

close一个TCP socket的缺省行为时把该socket标记为以关闭，然后立即返回到调用进程。该描述字不能再由调用进程使用，也就是说不能再作为read或write的第一个参数。