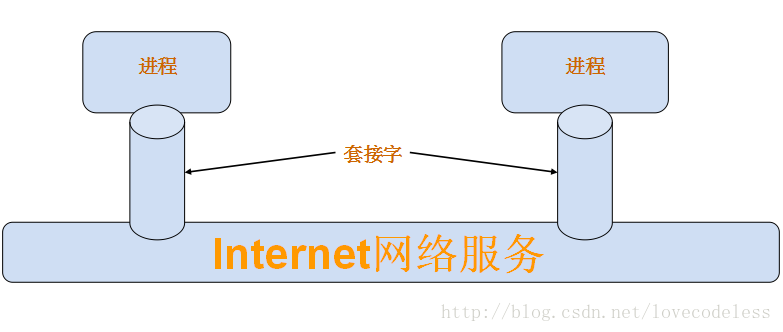
**Windows Socket+TCP服务器客户端**

**Winsock是 Windows下套接字标准**。

**Socket套接字基于计算机网络，提供同一系统上不同进程或由局域网连接在一起的不同机器上的进程间通讯功能。如下图：**

****

**套接字通过IP地址，Port端口号标识，通过这个标识可以在整个局域网定位一个套接字，通过套接字进程便可以相互传输数据。如：进程A与进程B之间欲通过套接字通信，首先进程A创建一个有IP地址，端口号唯一标识的套接字，进程B同样创建一个有IP地址，端口号唯一标识的套接字，进程A，B便可以通过对方套接字发送与接收信息。**

**TCP提供是可靠的数据传输服务，通过TCP套接口函数使用。**

**1.库支持：**

**Winsock API 函数由WS2\_32.DLL支持，可通过WS2\_32.LIB访问。Windows socket编程前需要初始化WS2\_32.DLL，通过han函数WSAStartup完成初始化。**

**#include <winsock2.h>**

**#pragma comment(lib,"ws2\_32.lib")  //静态加载ws2\_32.lib**

**WS2\_32.DLL 初始化：**

**int  WSAStartup(WORD wVersionRequested,LPWSADATA lpWSAData);**

**该函数第一个参数为WS2\_32.DLL版本，通常设为MAKEWORD(2,0)，第二个参数为一个指向WSADATA的指针，用于返回WS2\_32.DLL的配置信息。Socket编程前需要调用该函数。**

**WS2\_32.DLL 释放：**

**int  WSACleanup( );**

**该函数用于释放WS2\_32.DLL，不需要Winsock功能时调用该函数。**

**2.TCP基本函数：**

**创建socket：**

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

**第一个参数af:  表示族地址，网络编程一般使用AF\_INET宏。**

**第二个参数type:  表示连接类型，TCP选面向连接SOCK\_STREAM，UDP选数据报SOCK\_DGRAM。**

**第三个参数protocol:  表示协议，使用AF\_INET族地址TCP连接时，设为IPPROTO\_TCP。**

**绑定socket：**

**int  bind(SOCKET s, const struct sockaddr \*saddr,int namelen);**

**第一个参数:  需要绑定的socket。**

**第二个参数:  对应AF\_INET，使用struct sockaddr\_in包含协议，IP，端口等信息。**

**第三个参数:  对应AF\_INET，使用struct sockaddr\_in结构大小。**

**该函数将socket与协议，IP，端口号绑定起来，相对于给socket“命名”唯一的标识，这样其他的进程就可以通过这个标识找到这个socket。**

**监听socket：**

**int  listen(SOCKET s,  int nQueueSize);**

**第一个参数:  监听的socket。**

**第二个参数:  套接字监听队列最大连接请求数。**

**该函数将监听对socket的连接请求。**

**请求连接：**

**int  connect(SOCKET s,const struct sockaddr \*saddr,int namelen) ;**

**第一个参数:  socket本地进程的socket。**

**第二个参数:  对应AF\_INET，对方IP，端口等socket地址标识sockaddr\_in。**

**第三个参数:  对应AF\_INET，使用sockaddr\_in结构大小。**

**接受连接请求：**

**SOCKET  accept(SOCKET s,  struct sockaddr  \*addr,  int  \*addrlen);**

**第一个参数:  socket为被监听的socket。**

**第二个参数:  对应AF\_INET，一个sockaddr指针，将写入发送请求方的sockaddr\_in信息。**

**第三个参数:  对应AF\_INET，sockaddr\_in结构体的大小。**

**该函数用于接受一个socket连接请求，返回一个新的连接socket（实质是请求方的socket），发送与接收数据通过这个连接套socket。**

**发送数据：**

**int  send(SOCKET s, const char \*buf,  int len, int flags );**

**第一个参数:  socket为对方的socket。**

**第二个参数:  发送数据的缓冲区。**

**第三个参数:  数据缓冲区大小。**

**第四个参数:  紧急状态，一般这为0。**

**该函数用于向对方socket发送数据，成功返回发送数据的大小数。**

**发送数据：**

**int  recv(SOCKET s, char \*buf,  int len,  int flags);**

**第一个参数:  socket为对方的socket。**

**第二个参数:  接收数据的缓冲区。**

**第三个参数:  缓冲区大小。**

**第四个参数:  紧急状态，一般这为0。**

**该函数用于接收对方发送的数据，成功返回发送数据的大小数。**

**关闭套接字：**

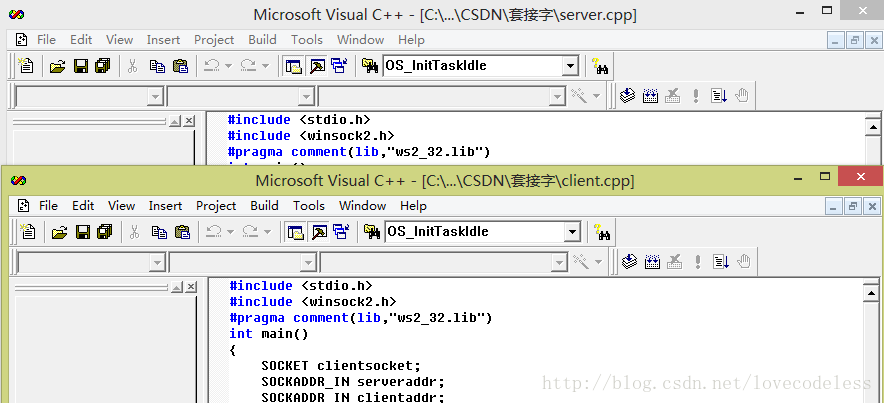
**int  closesocket(SOCKET s);**

**参数为socket。**

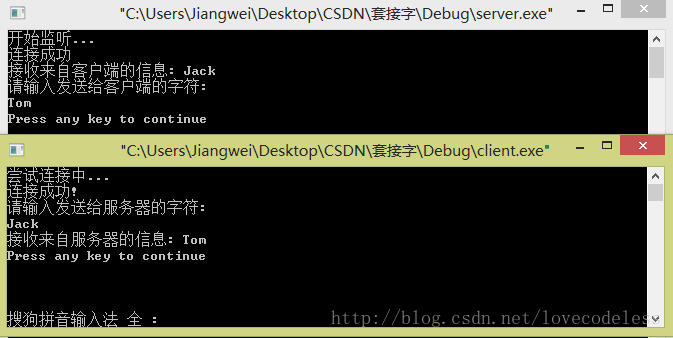
**该函数用于关闭套接字。**

**3.牛刀小试：**

**VC6.0中，先运行TCP服务器程序，再打开一个VC6.0，运行TCP客户端程序。**

****

**运行效果：**

****

**服务器端程序：**

|  |  |
| --- | --- |
| [1](http://blog.csdn.net/lovecodeless/article/details/25061141#L1)  [2](http://blog.csdn.net/lovecodeless/article/details/25061141#L2)  [3](http://blog.csdn.net/lovecodeless/article/details/25061141#L3)  [4](http://blog.csdn.net/lovecodeless/article/details/25061141#L4)  [5](http://blog.csdn.net/lovecodeless/article/details/25061141#L5)  [6](http://blog.csdn.net/lovecodeless/article/details/25061141#L6)  [7](http://blog.csdn.net/lovecodeless/article/details/25061141#L7)  [8](http://blog.csdn.net/lovecodeless/article/details/25061141#L8)  [9](http://blog.csdn.net/lovecodeless/article/details/25061141#L9)  [10](http://blog.csdn.net/lovecodeless/article/details/25061141#L10)  [11](http://blog.csdn.net/lovecodeless/article/details/25061141#L11)  [12](http://blog.csdn.net/lovecodeless/article/details/25061141#L12)  [13](http://blog.csdn.net/lovecodeless/article/details/25061141#L13)  [14](http://blog.csdn.net/lovecodeless/article/details/25061141#L14)  [15](http://blog.csdn.net/lovecodeless/article/details/25061141#L15)  [16](http://blog.csdn.net/lovecodeless/article/details/25061141#L16)  [17](http://blog.csdn.net/lovecodeless/article/details/25061141#L17)  [18](http://blog.csdn.net/lovecodeless/article/details/25061141#L18)  [19](http://blog.csdn.net/lovecodeless/article/details/25061141#L19)  [20](http://blog.csdn.net/lovecodeless/article/details/25061141#L20)  [21](http://blog.csdn.net/lovecodeless/article/details/25061141#L21)  [22](http://blog.csdn.net/lovecodeless/article/details/25061141#L22)  [23](http://blog.csdn.net/lovecodeless/article/details/25061141#L23)  [24](http://blog.csdn.net/lovecodeless/article/details/25061141#L24)  [25](http://blog.csdn.net/lovecodeless/article/details/25061141#L25)  [26](http://blog.csdn.net/lovecodeless/article/details/25061141#L26)  [27](http://blog.csdn.net/lovecodeless/article/details/25061141#L27)  [28](http://blog.csdn.net/lovecodeless/article/details/25061141#L28)  [29](http://blog.csdn.net/lovecodeless/article/details/25061141#L29)  [30](http://blog.csdn.net/lovecodeless/article/details/25061141#L30)  [31](http://blog.csdn.net/lovecodeless/article/details/25061141#L31)  [32](http://blog.csdn.net/lovecodeless/article/details/25061141#L32)  [33](http://blog.csdn.net/lovecodeless/article/details/25061141#L33)  [34](http://blog.csdn.net/lovecodeless/article/details/25061141#L34)  [35](http://blog.csdn.net/lovecodeless/article/details/25061141#L35)  [36](http://blog.csdn.net/lovecodeless/article/details/25061141#L36)  [37](http://blog.csdn.net/lovecodeless/article/details/25061141#L37)  [38](http://blog.csdn.net/lovecodeless/article/details/25061141#L38)  [39](http://blog.csdn.net/lovecodeless/article/details/25061141#L39)  [40](http://blog.csdn.net/lovecodeless/article/details/25061141#L40)  [41](http://blog.csdn.net/lovecodeless/article/details/25061141#L41)  [42](http://blog.csdn.net/lovecodeless/article/details/25061141#L42)  [43](http://blog.csdn.net/lovecodeless/article/details/25061141#L43)  [44](http://blog.csdn.net/lovecodeless/article/details/25061141#L44)  [45](http://blog.csdn.net/lovecodeless/article/details/25061141#L45)  [46](http://blog.csdn.net/lovecodeless/article/details/25061141#L46)  [47](http://blog.csdn.net/lovecodeless/article/details/25061141#L47)  [48](http://blog.csdn.net/lovecodeless/article/details/25061141#L48)  [49](http://blog.csdn.net/lovecodeless/article/details/25061141#L49)  [50](http://blog.csdn.net/lovecodeless/article/details/25061141#L50)  [51](http://blog.csdn.net/lovecodeless/article/details/25061141#L51)  [52](http://blog.csdn.net/lovecodeless/article/details/25061141#L52)  [53](http://blog.csdn.net/lovecodeless/article/details/25061141#L53)  [54](http://blog.csdn.net/lovecodeless/article/details/25061141#L54)  [55](http://blog.csdn.net/lovecodeless/article/details/25061141#L55)  [56](http://blog.csdn.net/lovecodeless/article/details/25061141#L56)  [57](http://blog.csdn.net/lovecodeless/article/details/25061141#L57)  [58](http://blog.csdn.net/lovecodeless/article/details/25061141#L58)  [59](http://blog.csdn.net/lovecodeless/article/details/25061141#L59)  [60](http://blog.csdn.net/lovecodeless/article/details/25061141#L60)  [61](http://blog.csdn.net/lovecodeless/article/details/25061141#L61)  [62](http://blog.csdn.net/lovecodeless/article/details/25061141#L62)  [63](http://blog.csdn.net/lovecodeless/article/details/25061141#L63)  [64](http://blog.csdn.net/lovecodeless/article/details/25061141#L64)  [65](http://blog.csdn.net/lovecodeless/article/details/25061141#L65)  [66](http://blog.csdn.net/lovecodeless/article/details/25061141#L66)  [67](http://blog.csdn.net/lovecodeless/article/details/25061141#L67)  [68](http://blog.csdn.net/lovecodeless/article/details/25061141#L68)  [69](http://blog.csdn.net/lovecodeless/article/details/25061141#L69)  [70](http://blog.csdn.net/lovecodeless/article/details/25061141#L70)  [71](http://blog.csdn.net/lovecodeless/article/details/25061141#L71)  [72](http://blog.csdn.net/lovecodeless/article/details/25061141#L72)  [73](http://blog.csdn.net/lovecodeless/article/details/25061141#L73)  [74](http://blog.csdn.net/lovecodeless/article/details/25061141#L74)  [75](http://blog.csdn.net/lovecodeless/article/details/25061141#L75) | **#include <stdio.h>**  **#include <winsock2.h>**  **#pragma comment(lib,"ws2\_32.lib")**  **int** **main**()  {  SOCKET serversoc;  SOCKET clientsoc;  SOCKADDR\_IN serveraddr;  SOCKADDR\_IN clientaddr;  **char** buf[1024];  **int** len;  WSADATA wsa;  WSAStartup(MAKEWORD(2,0),&wsa); //初始化WS2\_32.DLL  //创建套接字  **if**((serversoc = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP)) <= 0)  {  printf("套接字socket创建失败!\n");  **return** -1;  }    //命名协议，IP，端口  serveraddr.sin\_family = AF\_INET;  serveraddr.sin\_port = htons(9102);  serveraddr.sin\_addr.S\_un.S\_addr = htonl(INADDR\_ANY);    //绑定套接字  **if**(bind(serversoc, (SOCKADDR \*)&serveraddr, **sizeof**(serveraddr)) != 0)  {  printf("套接字绑定失败!\n");  **return** -1;  }    printf("开始监听...\n");  //监听请求  **if**(listen(serversoc, 1) != 0)  {  printf("监听失败!\n");  **return** -1;  }  len = **sizeof**(SOCKADDR\_IN);    //接收请求  **if**((clientsoc = accept(serversoc, (SOCKADDR \*)&clientaddr, &len))<=0)  {  printf("接受连接失败!\n");  **return** -1;  }  printf("连接成功\n");  //接收数据  **while**(1)  {  **if**(recv(clientsoc, buf, 1024, 0) <= 0)  {  printf("关闭连接!\n");  closesocket(clientsoc);  }  printf("接收来自客户端的信息: %s\n",buf);  **break**;  }  //发送数据  printf("请输入发送给客户端的字符:\n");  scanf("%s", buf);  //send to client  **if**(send(clientsoc, buf, strlen(buf)+1, 0)<=0)  {  printf("发送错误!\n");  }  WSACleanup(); //释放WS2\_32.DLL  **return** 0;  } |

[来自CODE的代码片](https://code.csdn.net/snippets/327943" \t "_blank)

server.c

**客户端程序：**

|  |  |
| --- | --- |
| [1](http://blog.csdn.net/lovecodeless/article/details/25061141#L1)  [2](http://blog.csdn.net/lovecodeless/article/details/25061141#L2)  [3](http://blog.csdn.net/lovecodeless/article/details/25061141#L3)  [4](http://blog.csdn.net/lovecodeless/article/details/25061141#L4)  [5](http://blog.csdn.net/lovecodeless/article/details/25061141#L5)  [6](http://blog.csdn.net/lovecodeless/article/details/25061141#L6)  [7](http://blog.csdn.net/lovecodeless/article/details/25061141#L7)  [8](http://blog.csdn.net/lovecodeless/article/details/25061141#L8)  [9](http://blog.csdn.net/lovecodeless/article/details/25061141#L9)  [10](http://blog.csdn.net/lovecodeless/article/details/25061141#L10)  [11](http://blog.csdn.net/lovecodeless/article/details/25061141#L11)  [12](http://blog.csdn.net/lovecodeless/article/details/25061141#L12)  [13](http://blog.csdn.net/lovecodeless/article/details/25061141#L13)  [14](http://blog.csdn.net/lovecodeless/article/details/25061141#L14)  [15](http://blog.csdn.net/lovecodeless/article/details/25061141#L15)  [16](http://blog.csdn.net/lovecodeless/article/details/25061141#L16)  [17](http://blog.csdn.net/lovecodeless/article/details/25061141#L17)  [18](http://blog.csdn.net/lovecodeless/article/details/25061141#L18)  [19](http://blog.csdn.net/lovecodeless/article/details/25061141#L19)  [20](http://blog.csdn.net/lovecodeless/article/details/25061141#L20)  [21](http://blog.csdn.net/lovecodeless/article/details/25061141#L21)  [22](http://blog.csdn.net/lovecodeless/article/details/25061141#L22)  [23](http://blog.csdn.net/lovecodeless/article/details/25061141#L23)  [24](http://blog.csdn.net/lovecodeless/article/details/25061141#L24)  [25](http://blog.csdn.net/lovecodeless/article/details/25061141#L25)  [26](http://blog.csdn.net/lovecodeless/article/details/25061141#L26)  [27](http://blog.csdn.net/lovecodeless/article/details/25061141#L27)  [28](http://blog.csdn.net/lovecodeless/article/details/25061141#L28)  [29](http://blog.csdn.net/lovecodeless/article/details/25061141#L29)  [30](http://blog.csdn.net/lovecodeless/article/details/25061141#L30)  [31](http://blog.csdn.net/lovecodeless/article/details/25061141#L31)  [32](http://blog.csdn.net/lovecodeless/article/details/25061141#L32)  [33](http://blog.csdn.net/lovecodeless/article/details/25061141#L33)  [34](http://blog.csdn.net/lovecodeless/article/details/25061141#L34)  [35](http://blog.csdn.net/lovecodeless/article/details/25061141#L35)  [36](http://blog.csdn.net/lovecodeless/article/details/25061141#L36)  [37](http://blog.csdn.net/lovecodeless/article/details/25061141#L37)  [38](http://blog.csdn.net/lovecodeless/article/details/25061141#L38)  [39](http://blog.csdn.net/lovecodeless/article/details/25061141#L39)  [40](http://blog.csdn.net/lovecodeless/article/details/25061141#L40)  [41](http://blog.csdn.net/lovecodeless/article/details/25061141#L41)  [42](http://blog.csdn.net/lovecodeless/article/details/25061141#L42)  [43](http://blog.csdn.net/lovecodeless/article/details/25061141#L43)  [44](http://blog.csdn.net/lovecodeless/article/details/25061141#L44)  [45](http://blog.csdn.net/lovecodeless/article/details/25061141#L45)  [46](http://blog.csdn.net/lovecodeless/article/details/25061141#L46)  [47](http://blog.csdn.net/lovecodeless/article/details/25061141#L47)  [48](http://blog.csdn.net/lovecodeless/article/details/25061141#L48)  [49](http://blog.csdn.net/lovecodeless/article/details/25061141#L49)  [50](http://blog.csdn.net/lovecodeless/article/details/25061141#L50)  [51](http://blog.csdn.net/lovecodeless/article/details/25061141#L51)  [52](http://blog.csdn.net/lovecodeless/article/details/25061141#L52)  [53](http://blog.csdn.net/lovecodeless/article/details/25061141#L53)  [54](http://blog.csdn.net/lovecodeless/article/details/25061141#L54)  [55](http://blog.csdn.net/lovecodeless/article/details/25061141#L55)  [56](http://blog.csdn.net/lovecodeless/article/details/25061141#L56) | **#include <stdio.h>**  **#include <winsock2.h>**  **#pragma comment(lib,"ws2\_32.lib")**  **int** **main**()  {  SOCKET clientsocket;  SOCKADDR\_IN serveraddr;  SOCKADDR\_IN clientaddr;  **char** buf[1024];  WSADATA wsa;  WSAStartup(MAKEWORD(2,0),&wsa); //初始化WS2\_32.DLL    //创建套接字  **if**((clientsocket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP)) <= 0)  {  printf("套接字socket创建失败!\n");  **return** -1;  }  serveraddr.sin\_family = AF\_INET;  serveraddr.sin\_port = htons(9102);  serveraddr.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1");    //请求连接  printf("尝试连接中...\n");  **if**(connect(clientsocket, (SOCKADDR \*)&serveraddr, **sizeof**(serveraddr)) != 0)  {  printf("连接失败!\n");  **return** -1;  }  printf("连接成功!\n");    //发送数据  printf("请输入发送给服务器的字符:\n");  scanf("%s", buf);  **if**(send(clientsocket, buf, strlen(buf)+1, 0)<=0)  {  printf("发送错误!\n");  }    //接收数据  **while**(1){  **if**(recv(clientsocket, buf, 1024, 0) <= 0)  {  printf("关闭连接!\n");  closesocket(clientsocket);  }  printf("接收来自服务器的信息: %s\n",buf);  **break**;  }  //关闭套接字  closesocket(clientsocket);  WSACleanup(); //释放WS2\_32.DLL  **return** 0;  } |

[来自CODE的代码片](https://code.csdn.net/snippets/327957" \t "_blank)

client.c

**参考资料**：http://blog.csdn.net/lovecodeless/article/details/25061141