

当 Socket 处于阻塞调用时, 调用 WSACleanup 函数, 此时对于 WINSOCK1.1 版本系统会无法返回有效值, 也就是说无法使用 WSAGetLastError 来确定错误的类型。因此当 Socket 处于阻塞调用时, 应该先调用 WSACancelBlockingCall 函数以取消阻塞调用, 一旦获得控制然后调用 WSACleanup 函数。

### 3. Socket()函数

Socket 函数用于创建一个套接字 Socket, 该套接字被绑定到一个特定的服务提供者。

函数原型:

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

表 13-4 参数说明

| 参 数      | 说 明                                                                        |
|----------|----------------------------------------------------------------------------|
| Af       | 地址族, 其值可取: AF_INET, AF_UNIX 等                                              |
| Type     | 套接字的类型, 其值可取: SOCK_STREAM, SOCK_DGRAM, SOCK_RAW                            |
| protocol | 使用协议, 其值可取: IPPROTO_TCP, IPPROTO_UDP, IPPROTO_ICMP 等                       |
| 输出参数     | 成功调用则返回新套接字的描述符, 否则返回 INVALID_SOCKET, 这时你可以使用 WSAGetLastError 函数来判定发生错误的类型 |

其中套接字类型的应用如表 13-5 所示。

表 13-5 套接字类型的应用

| 类 型         | 应 用                                                                                              |
|-------------|--------------------------------------------------------------------------------------------------|
| SOCK_STREAM | 提供可靠的、有序的、双向的、基于连接的数据流传输机制, 对于 Internet 应用来说, 它主要采用 TCP 协议                                       |
| SOCK_DGRAM  | 提供无连接的、不可靠的数据报传输机制, 对于 Internet 应用来说, 它采用的是 UDP 协议                                               |
| SOCK_RAW    | 原始套接字, 提供对诸如 ICMP, TCP, UDP 等协议数据的传输, 采用这种传输机制, 要求编程者自己填充 ICMP、TCP、UDP 乃至 IP 数据结构中的内容, 主要面向高级程序员 |

应用程序调用 socket() 建立一个新的 Socket 用于通讯。它返回一个新的套接字 Socket 描述字。该套接字用于以后的网络通信。

Socket 函数一旦返回 INVALID\_SOCKET, 可以调用 WSAGetLastError 函

数来确定错误的原因，如下所示。

|                    |                                              |
|--------------------|----------------------------------------------|
| WSANOTINITIALISED  | WSAStartup 没有调用或者没有调用成功                      |
| WSAENETDOWN        | 网络系统或者相关的网络服务提供者出错                           |
| WSAEAFNOSUPPORT    | 不支持指定的地址族                                    |
| WSAEINPROGRESS     | 一个 WINSOCKET1.1 版本的阻塞调用正在处理或者服务提供者正在处理一个回调函数 |
| WSAEMFILE          | 没有可用的套接字描述符                                  |
| WSAENOBUFS         | 没有可用的缓存空间                                    |
| WSAEPROTONOSUPPORT | 不支持指定的协议                                     |
| WSAEPROTOTYPE      | 对于本套接字来说，指定的协议类型错                            |
| WSAESOCKTNOSUPPORT | 地址族不支持指定的套接字类型                               |

4. Connect()函数

调用 connect()函数来完成到指定地址的连接操作。既可应用于面向连接的协议，又可应用于面向无连接的协议。

函数原型：

```
int connect (
    SOCKET s,
    const struct sockaddr FAR* name,
    int namelen
);
```

表 13-6 参数说明

| 参 数     | 说 明                                                                                                                                                                                                                                                                                                                                                                     |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S       | 已建立的套接字                                                                                                                                                                                                                                                                                                                                                                 |
| Name    | 套接字要连接的服务器地址                                                                                                                                                                                                                                                                                                                                                            |
| Namelen | name 的长度                                                                                                                                                                                                                                                                                                                                                                |
| 输出参数    | 成功调用则返回 0，否则返回 SOCK_ERROR，可以使用 WSAGetLastError 函数来判定发生错误的类型<br>对于一个阻塞的套接字来说，返回企图连接的结果：成功或者失败<br>对于一个非阻塞的套接字来说，连接可能不能立即完成，函数返回 SOCK_ERROR，在这种情况下，调用 WSAGetLastError 函数判定发生错误的类型，若是 WSAEWOULDBLOCK，有三种不同的解决方法：<br>1. 使用 select 函数来确定连接是否完成（通过检查套接字是否可写）<br>2. 使用 WSAAsyncSelect 来确定连接是否完成，一旦连接成功，系统返回 FD_CONNECT 事件<br>3. 使用 WSAAsyncSelect 来确定连接是否完成，一旦连接成功，系统会设置相应的事件对象 |

错误代码:

|                   |                                                                |
|-------------------|----------------------------------------------------------------|
| WSANOTINITIALISED | WSAStartup 没有调用或者没有调用成功                                        |
| WSAENETDOWN       | 网络系统出错                                                         |
| WSAEADDRINUSE     | 套接字的本地地址已经在用, 并且套接字没有被标识为地址重用 (SO_REUSEADDR.)                  |
| WSAEINTR          | 由于调用 WSACancelBlockingCall. 函数导致 (阻塞) Windows Socket 1.1 调用被取消 |
| WSAEINPROGRESS    | 一个 WINSOCKET1.1 版本的阻塞调用正在处理或者服务提供者正在处理一个回调函数                   |
| WSAEALREADY       | 指定的套接字有一个非阻塞的 connect 调用正在进行中                                  |
| WSAEADDRNOTAVAIL  | 无效的远程地址 (如: ADDR_ANY)                                          |
| WSAEAFNOSUPPORT   | 指定的地址不能用于该套接字                                                  |
| WSAECONNREFUSED   | 连接被拒绝                                                          |
| WSAEFAULT         | name 或者 namelen 不对                                             |
| WSAEINVAL         | s 可能是一个监听套接字, 或者指定的地址与该套接字所属的组的地址不一致                           |
| WSAEISCONN        | 该套接字已经连接                                                       |
| WSAENETUNREACH    | 本机此时无法访问网络                                                     |
| WSAENOBUFS        | 没有足够的缓存空间, 导致无法连接                                              |
| WSAENOTSOCK       | 该描述符不是一个套接字                                                    |
| WSAETIMEDOUT      | 连接超时                                                           |
| WSAEWOULDBLOCK    | 该套接字被标识为非阻塞方式, 连接没有立即完成                                        |
| WSAEACCES         | 当没有使用 setsockopt 设置 SO_BROADCAST 选项时, 而试图连接到一个广播地址             |

### 5. Send()和 sendto()函数

send()函数作用是向已建立连接的套接字上发送数据。既可用于向面向连接的套接字发送数据, 又可向面向非连接的套接字发送数据 (要先执行 connect() 函数), sendto()函数向指定的地址发送数据, 仅能用于面向非连接的套接字, 不能用于面向连接的套接字。

Send()函数原型:

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

表 13-7 参数说明

| 参 数   | 说 明                                                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------|
| S     | 套接字                                                                                                                                  |
| Buf   | 准备发送的数据                                                                                                                              |
| Len   | 准备发送的字节数                                                                                                                             |
| Flags | 调用的方式<br>其值可取：<br>MSG_DONTROUTE<br>不经过路由，将数据通过接口直接发送出去。Windows Sockets 服务提供者可选择忽略这个标志<br>MSG_OOB<br>发送带外数据（仅对流式套接字如 SOCK_STREAM 起作用） |
| 输出参数  | 成功调用则返回实际发送的字节数，否则返回 SOCK_ERROR，可以使用 WSAGetLastError 函数来判定发生错误的类型                                                                    |

Send()函数的错误代码：

WSANOTINITIALISED: WSAStartup 没有调用或者没有调用成功。

WSAENETDOWN: 网络系统出错。

WSAEACCES: 套接字的本地地址已经在用，并且套接字没有被标识为地址重用 (SO\_REUSEADDR)。

WSAEINTR: 由于调用 WSACancelBlockingCall 函数导致（阻塞）Windows Socket 1.1 调用被取消。

WSAEINPROGRESS: 一个 WINSOCKET1.1 版本的阻塞调用正在处理或者服务提供者正在处理一个回调函数。

WSAEFAULT: 参数 buf 超出用户地址空间。

WSAENETRESET: 由于在操作正在处理时，“keep-alive”检测到一个错误而导致连接被中断。

WSAENOBUFS: 没有足够缓存。

WSAENOTCONN: 套接字没有连接。

WSAENOTSOCK: 描述符不是套接字。

WSAEOPNOTSUPP: MSG\_OOB 被设定，但是套接字并非流类型（如：SOCK\_STREAM），因而不支持带外数据，或者由于该套接字是单向的且仅仅支持接受数据操作。

WSAESHUTDOWN: 由于执行 shutdown 操作使套接字关闭，因而不能发送数据。

**WSAEWOULDBLOCK:** 该套接字被标识为非阻塞方式, 连接没有立即完成。

**WSAEMSGSIZE:** 该套接字是面向消息的, 并且消息长度大于传输所支持的最大长度

**WSAEHOSTUNREACH:** 此时远端主机不可达

**WSAEINVAL:** 该套接字并未执行 bind 函数, 或者设置无效的标志 (flag), 或有设置了 MSG\_OOB 选项, 但是该套接字同时又设置成 SO\_OOBINLINE 模式

**WSAECONNABORTED:** 由于超时或其他错误导致连接中断。应用程序应及时关闭不使用的套接字和不能使用的套接字

**WSAECONNRESET:** 连接被远端主机重置。对于 UDP 套接字来说, 远端主机不能发送数据, 并接受到一个“端口不可达错误”。

**WSAETIMEDOUT:** 由于网络错误或远程主机关闭导致连接中断

Sendto()的函数原型:

```
int sendto (
    SOCKET s,
    const char FAR * buf,
    int len,
    int flags,
    const struct sockaddr FAR * to,
    int tolen
);
```

表 13-8 参数说明

| 参 数   | 说 明                                                                                                                                     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------|
| S     | 套接字                                                                                                                                     |
| buf   | 准备发送的数据                                                                                                                                 |
| len   | 准备发送的字节数                                                                                                                                |
| flags | 调用的方式<br>其值可取<br>MSG_DONTROUTE:<br>不经过路由, 将数据通过接口直接发送出去。Windows Sockets 服务提供者可选择忽略这个标志<br>MSG_OOB:<br>发送带外数据 (仅对流式套接字如 SOCK_STREAM 起作用) |
| To    | 远程地址                                                                                                                                    |
| tolen | 地址长度                                                                                                                                    |
| 输出参数  | 成功调用则返回实际发送的字节数, 否则返回 SOCK_ERROR, 可以使用 WSAGetLastError 函数来判定发生错误的类型                                                                     |

Sendto()函数的错误代码与 send()的相同。

#### 6. recv()和 recvfrom()函数

recv()函数用于从已连接的套接字上接收数据,recv 可以从面向连接的套接字上接收到来的数据,也可从面向非连接的套接字上接收数据。但是对于面向连接的套接字来说,则一定要使用 recv()函数。对于面向非连接的套接字来说,它还可以使用 recvfrom()函数来接收数据。

recvfrom()函数仅用于从面向非连接的套接字上接收数据。

recv()的函数原型:

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

表 13-9 参数说明

| 参 数   | 说 明                                                                                     |
|-------|-----------------------------------------------------------------------------------------|
| S     | 已建立的套接字                                                                                 |
| Buf   | 接收缓冲区                                                                                   |
| Len   | 缓冲区长度                                                                                   |
| Flags | 标志,表示调用的方式<br>MSG_PEEK:<br>检查到来的数据,但是并不把数据从输入队列里删除,它返回当前准备接收数据的数据<br>MSG_OOB:<br>处理带外数据 |
| 输出参数  | 成功调用则返回读取的字节数,若连接正常关闭,则返回 0,否则返回 SOCK_ERROR<br>以使用 WSAGetLastError 函数来判定发生错误的类型         |

recv()函数的错误代码:

WSANOTINITIALISED: WSAStartup 没有调用或者没有调用成功。

WSAENETDOWN: 网络系统出错。

WSAEINTR: 由于调用 WSACancelBlockingCall函数导致(阻塞) Windows Socket 1.1 调用被取消。

WSAEINPROGRESS: 一个 WINSOCKET1.1 版本的阻塞调用正在处理或者

服务提供者正在处理一个回调函数。

**WSAEFAULT:** 参数 *buf* 超出用户地址空间。

**WSAENETRESET:** 由于在操作正在处理时,“keep-alive”检测到一个错误而导致连接被中断。

**WSAENOTCONN:** 套接字没有连接。

**WSAENOTSOCK:** 描述符不是套接字。

**WSAEOPNOTSUPP:** MSG\_OOB 被设定,但是套接字并非流类型(如: SOCK\_STREAM),因而不支持带外数据,或者由于该套接字是单向的且仅仅支持接受数据操作。

**WSAESHUTDOWN:** 由于执行 shutdown 操作使套接字关闭,因而不能发送数据。

**WSAEWOULDBLOCK:** 该套接字被标识为非阻塞方式,连接没有立即完成。

**WSAEMSGSIZE:** 该套接字是面向消息的,并且消息长度大于传输所支持的最大长度。

**WSAEINVAL:** 该套接字并未执行 bind 函数,或者设置无效的标志(flag),或设置了 MSG\_OOB 选项,但是该套接字同时又设置成 SO\_OOBINLINE 模式。

**WSAECONNABORTED:** 由于超时或其他错误导致连接中断,应用程序应及时关闭不使用的套接字和不能使用的套接字。

**WSAECONNRESET:** 连接被远端主机重置,对于 UDP 套接字来说,远端主机不能发送数据,并接受到一个“端口不可达错误”。

**WSAETIMEDOUT:** 由于网络错误或远程主机关闭导致连接中断。

**recvfrom()函数原型:**

```
int recvfrom (
    SOCKET s,
    char FAR* buf,
    int len,
    int flags,
    struct sockaddr FAR* from,
    int FAR* fromlen
);
```

表 13-10 参数说明

| 参 数     | 说 明                                                                                          |
|---------|----------------------------------------------------------------------------------------------|
| S       | 已建立的套接字                                                                                      |
| Buf     | 接收缓冲区                                                                                        |
| Len     | 缓冲区长度                                                                                        |
| Flags   | 标志，表示调用的方式<br><br>MSG_PEEK：<br>检查到来的数据，但是并不把数据从输入队列里删除，它返回当前准备接收数据的字节数<br><br>MSG_OOB：处理带外数据 |
| From    | 缓冲区地址，该缓冲区用于填充接收到的数据的原地址                                                                     |
| Fromlen | 表示该缓冲区大小                                                                                     |
| 输出参数    | 成功调用则返回读取的字节数，若连接正常关闭，则返回 0，否则返回 SOCK_ERROR，可以使用 WSAGetLastError 函数来判定发生错误的类型                |

recvfrom()的错误代码与 recv()函数的相同。

7. closesocket()函数

closesocket 用于关闭已存在的套接字，释放系统分配的资源，正在进行的异步操作会被取消执行，同时那些异步操作函数会返回 WSA\_OPERATION\_ABORTED 错误代码。对于每一个 socket()建立的套接字，都应该有一个 closesocket()与之匹配。

函数原型

```
int closesocket (
    SOCKET s
);
```

表 13-11 参数说明

| 参 数  | 说 明                                                       |
|------|-----------------------------------------------------------|
| S    | 已建立的套接字                                                   |
| 输出参数 | 成功则返回 0，否则返回 SOCK_ERROR，可以使用 WSAGetLastError 函数来判定发生错误的类型 |

错误代码：

WSANOTINITIALISED：WSAStartup 没有调用或者没有调用成功。

WSAENETDOWN：网络系统出错。

**WSAENOTSOCK:** 描述符不是套接字。

**WSAEINPROGRESS:** 一个 WINSOCKET1.1 版本的阻塞调用正在处理或者服务提供者正在处理一个回调函数。

**WSAEINTR:** 由于调用 WSACancelBlockingCall 函数导致（阻塞）Windows Socket 1.1 调用被取消。

**WSAEWOULDBLOCK:** 该套接字被标识为非阻塞方式，连接没有立即完成。  
注意：套接字的设置选项 SO\_LINGER 和 SO\_DONTLINGER（缺省设置）会影响套接字的关闭方式。见表 13-12。

表 13-12 套接字的设置选项

| 选 项           | 间 隔 | 关闭方式    | 等 待 |
|---------------|-----|---------|-----|
| SO_DONTLINGER |     | 正常      | 不   |
| SO_LINGER     | 0   | 直接（硬关闭） | 不   |
| SO_LINGER     | 非 0 | 正常      | 是   |

## 8. Bind()函数

一旦为某种特定的协议创建了套接字，就必须将套接字绑定到一个已知地址。Bind 函数可将指定的套接字同一个已知的地址绑定在一起。本函数既可用于面向连接的套接字，又可用于面向无连接的套接字。

函数原型：

```
int bind (
    SOCKET s,
    const struct sockaddr FAR* name,
    int namelen
);
```

表 13-13 参数说明

| 参 数     | 说 明                                                       |
|---------|-----------------------------------------------------------|
| S       | 已建立的套接字                                                   |
| Name    | 要绑定的地址                                                    |
| Namelen | 地址长度                                                      |
| 输出参数    | 成功则返回 0，否则返回 SOCK_ERROR，可以使用 WSAGetLastError 函数来判定发生错误的类型 |

错误代码：

WSANOTINITIALISED: WSAStartup 没有调用或者没有调用成功。

WSAENETDOWN: 网络系统出错。

WSAEADDRINUSE : 地址已经在用，并且其套接字并没有设置地址重用选项 SO\_REUSEADDR。

WSAEADDRNOTAVAIL: 指定的地址不是本机合法地址。

WSAEFAULT: 参数 Name 和 namelen 不是有效的用户地址空间；namelen 太小；name 包含不正确的地址格式或者 name 的前两个字节包含的内容与该套接字相应的地址族不匹配。

WSAEINPROGRESS: 一个 WINSOCKET1.1 版本的阻塞调用正在处理或者服务提供者正在处理一个回调函数。

WSAEINVAL: 该套接字已经绑定到一个地址。

WSAENOBUFS: 没有足够的缓冲，连接太多。

WSAENOTSOCK: 该描述符不是一个套接字。

9. Listen()函数

Listen 函数时将套接字置入监听模式。本函数一般应用于服务进程，要求用 socket()先建立套接字，然后用 Bind 函数的作用指示将建立的套接字和本机的地址关联起来，其后调用 listen()函数等待客户进程的连接请求。服务器一般容许处理多个进入的连接请求，同时处理的连接请求数由 backlog 的值来确定，当客户的连接请求到来时，该请求队列已满，此时客户端的连接请求会返回一个 WSAECONNREFUSED 错误。

函数原型：

```
int listen (
    SOCKET s,
    int backlog
);
```

表 13-14 参数说明

| 参 数     | 说 明                                                                                       |
|---------|-------------------------------------------------------------------------------------------|
| S       | 已建立的套接字                                                                                   |
| Backlog | 等待处理连接队列的长度。如果设置此值为 SOMAXCONN，那么由相应的服务提供者负责将 backlog 设置为一个合理的值。并且目前还没有手段来确定实际的 backlog 的值 |
| 输出参数    | 成功则返回 0，否则返回 SOCK_ERROR，可以使用 WSAGetLastError 函数来判定发生错误的类型                                 |