SOCKET

Ø伯克利套接字应用程序接口（Berkeley Sockets API）,最初是作为BSD4.2的一部分发布的，提供了进程与TCP/IP模型各个层之间通信的标准方法。发布以来，这个API已经被移植到每一个主要的操作系统和流行的编程语言，所以它是网络编程中名副其实的标准。

Socket初始化

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

af协议族

宏	含义
AF_UNSPEC	未指定
AF_INET	网际协议第四版（IPv4）
AF_IPX	网间分组交换：流行于Novell和MS-DOS系统的早期网络层协议
AF_APPLETALK	Appletalk协议：流行于苹果系统的早期网络协议系列
AF_INET6	网际协议第六版（IPv6）

type类型

宏	含义
SOCK_STREAM	数据包代表有序的、可靠的数据流分段
SOCK_DGRAM	数据包代表离散的报文
SOCK_RAW	数据包头部可以由应用层自定义
SOCK_SEQPACKET	与SOCK_STREAM类似，但是数据包接收时需要整体读取

protocol协议

宏	需要的类型	含义
IPPROTO_UDP	SOCK_DGRAM	数据包封装UDP数据报
IPPROTO_TCP	SOCK_STREAM	数据包封装TCP报文段
IPPROTO_IP/0	Any	为给定的类型使用默认协议

Socket关闭

```
int closesocket(SOCKET sock);
```

停止传输

```
int shutdown(SOCKET sock, int how)

// SD_SEND 停止发送
// SD_RECV 停止接收
// SD_BOTH 停止发送和接收
```

Socket地址

```
struct sockaddr {
    uinit16_t sa_family;
    char sa_data[14];
}

struct sockaddr_in {
    short sa_family;
    uint16_t sin_port;
    struct in_addr sin_addr;
    char sin_zero[8];
}

```

主机字节转换函数

```
uint16_t htons(uint16_t hostshort);
uint32_t htonl(uint32_t hostlong);
```

使用字符串表示地址

```
unsigned long inet_addr( const char *cp );
char * inet_ntoa( in_addr in );
```

使用字符串初始化SOCKADDR

```
int inet_pton(int af, const char*src, void* dest);
int InetPton(int af, const char*src, void* dest);
```

通过域名获取IP地址

```
int getaddrinfo(
    const char*pNodeName,
    const char* pServiceName,
    const addrinfo*pHints,
    addrinfo**ppResult
); //阻塞线程

void freeaddrinfo( addrinfo* pAddrInfo );

INT WSAAPI GetAddrInfoEx(
    PCSTR pName,
    PCSTR pServiceName,
    DWORD dwNameSpace,
    LPGUID lpNspId,
    const ADDRINFOEXA *hints,
    PADDRINFOEXA *ppResult,
    timeval *timeout,
    LPOVERLAPPED lpOverlapped,
    LPLOOKUPSERVICE_COMPLETION_ROUTINE lpCompletionRoutine,
    LPHANDLE lpNameHandle
);
```

绑定SOCKET

```
int bind( SOCKET s, const sockaddr *addr, int namelen );
```

发送数据报

```
// 错误返回-1，否者表示数据成功进入队列
int sendto(SOCKET s, const char *buf, int len, int flags, const sockaddr *to,
int tolen);
```

接收数据报

```
// 默认线程进入阻塞状态
int recvfrom( SOCKET s, char *buf, int len, int flags, sockaddr *from, int
*fromlen );
```