**struct sockaddr\_in {**

**short sin\_family; /\* Address family \*/**

**unsigned short sin\_port; /\* Port number \*/**

**struct in\_addr sin\_addr; /\* Internet address \*/**

**unsigned char sin\_zero[8]; /\* Same size as struct sockaddr \*/**

**};**

sin\_family指代协议族，在socket编程中只能是AF\_INET

sin\_port存储[端口号](http://baike.haosou.com/doc/1540075.html" \t "_blank)（**使用网络[字节顺序](http://baike.haosou.com/doc/6261441.html" \t "_blank)**），在linux下，端口号的范围0~65535,同时0~1024范围的端口号已经被系统使用或保留。

**例：mysock.sin\_port=htons(1234); //1234是[端口号](http://baike.haosou.com/doc/1540075.html" \t "_blank)**

sin\_addr存储IP地址，使用in\_addr这个[数据结构](http://baike.haosou.com/doc/5398760.html" \t "_blank)

**例：mysock.sin\_addr.s\_addr=inet\_addr("192.168.0.1");**

sin\_zero是为了让sockaddr与sockaddr\_in两个[数据结构](http://baike.haosou.com/doc/5398760.html" \t "_blank)保持大小相同而保留的空字节。

**在windows下：**

**s\_addr按照网络字节顺序存储IP地址**

**例：mysock.sin\_addr.s\_addr=inet\_addr("192.168.0.1");**

**typedef struct in\_addr{**

**union{**

**struct{unsigned char s\_b1,s\_b2,s\_b3,s\_b4;} S\_un\_b;**

**struct{unsigned short s\_w1,s\_w2;} S\_un\_w;**

**unsigned long S\_addr;**

**} S\_un;**

**} IN\_ADDR;**

例：sockaddr\_in name;

name.sin\_family = AF\_INET;

name.sin\_addr.S\_un.S\_addr = inet\_addr(ADDR\_ANY);

name.sin\_port = 9876;