

旁注 套接字接口的起源

套接字接口是加州大学伯克利分校的研究人员在 20 世纪 80 年代早期提出的。因为这个原因，它也经常被称为伯克利套接字。伯克利的研究者使得套接字接口适用于任何底层的协议。第一个实现的就是针对 TCP/IP 协议的，他们把它包括在 Unix 4.2BSD 的内核里，并且分发给许多学校和实验室。这在因特网的历史上是一个重大事件。几乎一夜之间，成千上万的人们接触到了 TCP/IP 和它的源代码。它引起了巨大的轰动，并激发了新的网络和网络互联研究的浪潮。

11.4.1 套接字地址结构

从 Linux 内核的角度来看，一个套接字就是通信的一个端点。从 Linux 程序的角度来看，套接字就是一个有相应描述符的打开文件。

因特网的套接字地址存放在如图 11-13 所示的类型为 `sockaddr_in` 的 16 字节结构中。对于因特网应用，`sin_family` 成员是 `AF_INET`，`sin_port` 成员是一个 16 位的端口号，而 `sin_addr` 成员就是一个 32 位的 IP 地址。IP 地址和端口号总是以网络字节顺序（大端法）存放的。

code/netp/netpfragments.c

```
/* IP socket address structure */
struct sockaddr_in {
    uint16_t    sin_family; /* Protocol family (always AF_INET) */
    uint16_t    sin_port;   /* Port number in network byte order */
    struct in_addr sin_addr; /* IP address in network byte order */
    unsigned char sin_zero[8]; /* Pad to sizeof(struct sockaddr) */
};

/* Generic socket address structure (for connect, bind, and accept) */
struct sockaddr {
    uint16_t    sa_family; /* Protocol family */
    char        sa_data[14]; /* Address data */
};
```

code/netp/netpfragments.c

图 11-13 套接字地址结构

旁注 _in 后缀意味着什么？

`_in` 后缀是互联网络(internet)的缩写，而不是输入(input)的缩写。

`connect`、`bind` 和 `accept` 函数要求一个指向与协议相关的套接字地址结构的指针。套接字接口的设计者面临的问题是，如何定义这些函数，使之能接受各种类型的套接字地址结构。今天我们可以使用通用的 `void*` 指针，但是那时在 C 中并不存在这种类型的指针。解决办法是定义套接字函数要求一个指向通用 `sockaddr` 结构(图 11-13)的指针，然后要求应用程序将与协议特定的结构的指针强制转换成这个通用结构。为了简化代码示例，我们跟随 Steven 的指导，定义下面的类型：

```
typedef struct sockaddr SA;
```

然后无论何时需要将 `sockaddr_in` 结构强制转换成通用 `sockaddr` 结构时，我们都使用这个类型。