

携带。在 IP 地址结构中存放的地址总是以(大端法)网络字节顺序存放的,即使主机字节顺序(host byte order)是小端法。Unix 提供了下面这样的函数在网络和主机字节顺序间实现转换。

```
#include <arpa/inet.h>

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

uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);
```

返回: 按照网络字节顺序的值。

返回: 按照主机字节顺序的值。

htonl 函数将 32 位整数由主机字节顺序转换为网络字节顺序。ntohl 函数将 32 位整数从网络字节顺序转换为主机字节。htons 和 ntohs 函数为 16 位无符号整数执行相应的转换。注意,没有对应的处理 64 位值的函数。

IP 地址通常是以一种称为点分十进制表示法来表示的,这里,每个字节由它的十进制值表示,并且用句点和其他字节间分开。例如,128.2.194.242 就是地址 0x8002c2f2 的点分十进制表示。在 Linux 系统上,你能够使用 HOSTNAME 命令来确定你自己主机的点分十进制地址:

```
linux> hostname -i
128.2.210.175
```

应用程序使用 inet\_pton 和 inet\_ntop 函数来实现 IP 地址和点分十进制串之间的转换。

```
#include <arpa/inet.h>

int inet_pton(AF_INET, const char *src, void *dst);

const char *inet_ntop(AF_INET, const void *src, char *dst,
                      socklen_t size);
```

返回: 若成功则为 1,若 src 为非合法点分十进制地址则为 0,若出错则为 -1。

返回: 若成功则指向点分十进制字符串的指针,若出错则为 NULL。

在这些函数名中,“n”代表网络,“p”代表表示。它们可以处理 32 位 IPv4 地址(AF\_INET)(就像这里展示的那样),或者 128 位 IPv6 地址(AF\_INET6)(这部分我们不讲)。

inet\_pton 函数将一个点分十进制串(src)转换为一个二进制的网络字节顺序的 IP 地址(dst)。如果 src 没有指向一个合法的点分十进制字符串,那么该函数就返回 0。任何其他错误会返回 -1,并设置 errno。相似地,inet\_ntop 函数将一个二进制的网络字节顺序的 IP 地址(src)转换为它所对应的点分十进制表示,并把得到的以 null 结尾的字符串的最多 size 个字节复制到 dst。

#### 练习题 11.1 完成下表:

十六进制地址	点分十进制地址
0x0	
0xffffffff	
0x7f000001	
	205.188.160.121
	64.12.149.13
	205.188.146.23