<https://blog.csdn.net/lindyl/article/details/10427925>

**inet\_pton 和 inet\_ntop**

Linux下这2个IP地址转换函数，可以在将IP地址在“点分十进制”和“整数”之间转换而且，inet\_pton和inet\_ntop这2个函数能够处理ipv4和ipv6。算是比较新的函数了。

**inet\_pton**

函数原型如下[将"点分十进制" -> "整数"]

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

int inet\_pton(int af, const char \*src, void \*dst);

//这个函数转换字符串到网络地址，第一个参数af是地址族，转换后存在dst中

inet\_pton是inet\_addr的扩展，支持的多地址族有下列：  
af = AF\_INET  
src为指向字符型的地址，即ASCII的地址的首地址（ddd.ddd.ddd.ddd格式的），函数将该地址转换为in\_addr的结构体，并复制在\*dst中  
af = AF\_INET6  
src为指向IPV6的地址，函数将该地址转换为in6\_addr的结构体，并复制在\*dst中。如果函数出错将返回一个负值，并将errno设置为EAFNOSUPPORT，如果参数af指定的地址族和src格式不对，函数将返回0。

**inet\_ntop**

函数原型如下[将"点分十进制" -> "整数"]

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

const char \*inet\_ntop(int af, const void \*src, char \*dst, socklen\_t cnt);

//这个函数转换网络二进制结构到ASCII类型的地址，参数的作用和上面相同，只是多了一个参数socklen\_t cnt,

//他是所指向缓存区dst的大小，避免溢出，如果缓存区太小无法存储地址的值，则返回一个空指针，并将errno置为ENOSPC

## 编程实例：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <netinet/in.h>

int main (void)

{

char IPdotdec[20]; // 存放点分十进制IP地址

struct in\_addr s; // IPv4地址结构体

// 输入IP地址

printf("Please input IP address: ");

scanf("%s", &IPdotdec);

// 转换

inet\_pton(AF\_INET, IPdotdec, (void \*)&s);

printf("inet\_pton: 0x%x\n", s.s\_addr); // 注意得到的字节序

// 反转换

inet\_ntop(AF\_INET, (void \*)&s, IPdotdec, 16);

printf("inet\_ntop: %s\n", IPdotdec);

}