NAME

inet, inet_addr, inet_network, inet_makeaddr, inet_lnaof, inet_netof, inet_ntoa – Internet address manipulation

SYNOPSIS

cc [*flag* ...] *file* ... –**lnsl** [*library* ...]

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

unsigned long inet_addr(const char **cp***);**

unsigned long inet_network(const char **cp***);**

struct in_addr inet_makeaddr(const int *net***, const int** *lna***);**

int inet_lnaof(const struct in_addr *in***);**

int inet_netof(const struct in_addr *in***);**

char *inet_ntoa(const struct in_addr *in***);**

MT-LEVEL

Safe

DESCRIPTION

The **inet_addr( )** and **inet_network( )** routines interpret character strings representing numbers expressed in the Internet standard '**.**' notation, returning numbers suitable for use as Internet addresses and Internet network numbers, respectively. The routine **inet_makeaddr( )** takes an Internet network number and a local network address and constructs an Internet address from it. The routines **inet_netof( )** and **inet_lnaof( )** break apart Internet host addresses, returning the network number and local network address part, respectively.

The routine **inet_ntoa( )** returns a pointer to a string in the base 256 notation "d.d.d.d" described below.

All Internet addresses are returned in network order (bytes ordered from left to right). All network numbers and local address parts are returned as machine format integer values.

INTERNET ADDRESSES

Values specified using the '**.**' notation take one of the following forms:

a.b.c.d
a.b.c
a.b
a

When four parts are specified, each is interpreted as a byte of data and assigned, from left to right, to the four bytes of an Internet address.

When a three part address is specified, the last part is interpreted as a 16-bit quantity and placed in the right most two bytes of the network address. This makes the three part address format convenient for specifying Class B network addresses as "128.net.host".

When a two part address is supplied, the last part is interpreted as a 24-bit quantity and placed in the right most three bytes of the network address. This makes the two part address format convenient for specifying Class A network addresses as “net.host”.

When only one part is given, the value is stored directly in the network address without any byte rearrangement.

All numbers supplied as “parts” in a ‘**.**’ notation may be decimal, octal, or hexadecimal, as specified in the C language (that is, a leading 0x or 0X implies hexadecimal; otherwise, a leading 0 implies octal; otherwise, the number is interpreted as decimal).

RETURN VALUES

The value −**1** is returned by **inet_addr( )** and **inet_network( )** for malformed requests.

The routines **inet_netof( )** and **inet_lnaof( )** break apart Internet host addresses, returning the network number and local network address part, respectively.

The routine **inet_ntoa( )** returns a pointer to a string in the base 256 notation “d.d.d.d” described below.

SEE ALSO

gethostbyname(3N), **getnetbyname**(3N), **hosts**(4), **networks**(4)

NOTES

The return value from **inet_ntoa( )** points to a buffer which is overwritten on each call. This buffer is implemented as thread-specific data in multithreaded applications.

BUGS

The problem of host byte ordering versus network byte ordering is confusing. A simple way to specify Class C network addresses in a manner similar to that for Class B and Class A is needed.