



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'inet\_addr.3'***

***\$ man inet\_addr.3***

INET(3)           Linux Programmer's Manual           INET(3)

NAME

inet\_aton, inet\_addr, inet\_network, inet\_ntoa, inet\_makeaddr,  
inet\_lnaof, inet\_netof - Internet address manipulation routines

SYNOPSIS

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

int inet_aton(const char *cp, struct in_addr *inp);
in_addr_t inet_addr(const char *cp);
in_addr_t inet_network(const char *cp);
char *inet_ntoa(struct in_addr in);
struct in_addr inet_makeaddr(in_addr_t net, in_addr_t host);
in_addr_t inet_lnaof(struct in_addr in);
in_addr_t inet_netof(struct in_addr in);
```

Feature Test Macro Requirements for glibc (see feature\_test\_macros(7)):

inet\_aton(), inet\_ntoa():

Since glibc 2.19:

`_DEFAULT_SOURCE`

In glibc up to and including 2.19:

`_BSD_SOURCE || _BSD_SOURCE`

## DESCRIPTION

`inet_aton()` converts the Internet host address `cp` from the IPv4 numbers-and-dots notation into binary form (in network byte order) and stores it in the structure that `inp` points to. `inet_aton()` returns nonzero if the address is valid, zero if not. The address supplied in `cp` can have one of the following forms:

`a.b.c.d` Each of the four numeric parts specifies a byte of the address; the bytes are assigned in left-to-right order to produce the binary address.

`a.b.c` Parts `a` and `b` specify the first two bytes of the binary address. Part `c` is interpreted as a 16-bit value that defines the rightmost two bytes of the binary address. This notation is suitable for specifying (outmoded) Class B network addresses.

`a.b` Part `a` specifies the first byte of the binary address. Part `b` is interpreted as a 24-bit value that defines the rightmost three bytes of the binary address. This notation is suitable for specifying (outmoded) Class A network addresses.

`a` The value `a` is interpreted as a 32-bit value that is stored directly into the binary address without any byte rearrangement.

In all of the above forms, components of the dotted address can be specified in decimal, octal (with a leading 0), or hexadecimal, with a leading 0X). Addresses in any of these forms are collectively termed IPv4 numbers-and-dots notation. The form that uses exactly four decimal numbers is referred to as IPv4 dotted-decimal notation (or sometimes: IPv4 dotted-quad notation).

`inet_aton()` returns 1 if the supplied string was successfully interpreted, or 0 if the string is invalid (`errno` is not set on error).

The `inet_addr()` function converts the Internet host address `cp` from





network and host components at arbitrary bit (rather than byte) bound?

aries.

## EXAMPLES

An example of the use of `inet_aton()` and `inet_ntoa()` is shown below.

Here are some example runs:

```
$ ./a.out 226.000.000.037 # Last byte is in octal
226.0.0.31
```

```
$ ./a.out 0x7f.1 # First byte is in hex
127.0.0.1
```

## Program source

```
#define _BSD_SOURCE
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>

int
main(int argc, char *argv[])
{
    struct in_addr addr;

    if (argc != 2) {
        fprintf(stderr, "%s <dotted-address>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    if (inet_aton(argv[1], &addr) == 0) {
        fprintf(stderr, "Invalid address\n");
        exit(EXIT_FAILURE);
    }

    printf("%s\n", inet_ntoa(addr));
    exit(EXIT_SUCCESS);
}
```

## SEE ALSO

`byteorder(3)`, `getaddrinfo(3)`, `gethostbyname(3)`, `getnameinfo(3)`, `getnetent(3)`, `inet_net_pton(3)`, `inet_ntop(3)`, `inet_pton(3)`, `hosts(5)`, `net?works(5)`

## COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.

GNU

2020-12-21

INET(3)