

NAME

link_addr, **link_ntoa** - elementary address specification routines for link level access

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
#include <net/if_dl.h>
```

void

```
link_addr(const char *addr, struct sockaddr_dl *sdl);
```

*char **

```
link_ntoa(const struct sockaddr_dl *sdl);
```

DESCRIPTION

The routine **link_addr**() interprets character strings representing link-level addresses, returning binary information suitable for use in system calls. The routine **link_ntoa**() takes a link-level address and returns an ASCII string representing some of the information present, including the link level address itself, and the interface name or number, if present. This facility is experimental and is still subject to change.

For **link_addr**(), the string *addr* may contain an optional network interface identifier of the form "name unit-number", suitable for the first argument to `ifconfig(8)`, followed in all cases by a colon and an interface address in the form of groups of hexadecimal digits separated by periods. Each group represents a byte of address; address bytes are filled left to right from low order bytes through high order bytes.

Thus `le0:8.0.9.13.d.30` represents an ethernet address to be transmitted on the first Lance ethernet interface.

RETURN VALUES

The **link_ntoa**() function always returns a null terminated string. The **link_addr**() function has no return value. (See *BUGS*.)

SEE ALSO

`getnameinfo(3)`

HISTORY

The **link_addr()** and **link_ntoa()** functions appeared in 4.3BSD-Reno.

BUGS

The returned values for **link_ntoa** reside in a static memory area.

The function **link_addr()** should diagnose improperly formed input, and there should be an unambiguous way to recognize this.

If the *sdl_len* field of the link socket address *sdl* is 0, **link_ntoa()** will not insert a colon before the interface address bytes. If this translated address is given to **link_addr()** without inserting an initial colon, the latter will not interpret it correctly.