NAME

tliaddr – print out the Transport Layer Interface address

SYNOPSIS

tliaddr *host* [*port*] [**-i** *interface[:address_family]*] [**-f** *address_family*] [**-p** *port*] [**-D**] [**-V**]

DESCRIPTION

Introduction

The *tliaddr* program is most commonly used on a UNIX system to create the Transport Layer Interface (TLI) address in hexadecimal for a TCP/IP network host. This is needed to setup the Network Listener Service (NLS) to listen on the desired network address for incoming connections. Remote File Sharing is often setup to accept incoming network connections from the NLS (as well as from other possible sources). Note that the output of this program can be used any place that a hexadecimal TLI address for the TCP/IP transport provider might be needed. Another common place that these types of addresses are needed is in the administration of UUCP TLI addresses for calling out.

This program is a little bit more general than the R&D UNIX program *hostrfs* that is specific to TCP/IP TLI addresses. This program can be used to create a hexadecimal TLI address for several interface address types (in addition to the socket API) and can also process more address families within the socket address type than just TCP.

By default, the socket address type interface is assumed and the address family of ’INET’ is assumed. These are most commonly used and with these as default. This program behaves very similarly to the *hostrfs* program (although that program can only accept numeric port specifications).

Arguments

An address argument (usually the name of a remote host machine) is always required. When the `socket` interface address type is being used (the default), additional possible host names consisting of:

```
anyhost
```

can also be used. In this case, *anyhost* is a built-in name for the all-host or any-host address. A host name of consisting of a single dash character (’-’) is also taken to be the any-host address. Any-host addresses are commonly used for listeners since they often want to listen for incoming requests on all addresses that the node has. Note that RFS, REXEC, UUCP usually do **not** want to listen on the local-host address (hostname ’localhost’) since they usually are not setup to accept incoming connections from the same machine that originated the connection. In these sorts of cases, use actual interface host names. For machines that have more than one interface host name, setup additional listeners for those other addresses.

An optional TCP port number argument can be specified as the second positional argument on invocation. If no TCP port number is specified, the default port of 2766 (corresponding to the `listen` Internet TCP service) is used.

Options

The options provide a way to set a non-default interface address type (other than `socket`) or an address family (with the `socket` interface address type) other than ’INET’.

Output

The output of the program will be a character string that represents the Transport Layer Interface (TLI) address needed by TLI based facilities. The string is the hexadecimal representation of the address used by the programs that interface to the TLI API.

ENVIRONMENT VARIABLES

There are no environment variables used by this program.

EXAMPLES

☞ get the TLI address for a TCP host named `bob` on the default port of 2766:

```
tliaddr bob
```

- ☞ print out the TLI address for a TCP host named
  `fred`
  using a port of 1025:

  ```
  tliaddr fred 1025
  ```

SEE ALSO

hostrfs(1m), nlsadmin(1m), sacadm(1m), pmadm(1m), rfsaddr(1)

PATH TO

This program is currently located in `/usr/add-on/local/bin` or possibly where ever "LOCAL" programs are stored on your system. This is often at `${LOCAL}/bin` on some systems.

ACKNOWLEDGMENTS

This program was inspired by the `hostrfs` program previously written by the R&D UNIX folks out in Illinois.

AUTHOR

David A.D. Morano