NAME

t_bind – bind an address to a transport endpoint

SYNOPSIS

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

#include <tiuser.h>

int t_bind(int *fildes***, const struct t_bind** ∗*req*, **struct t_bind** ∗*ret***);**

MT-LEVEL

MT-Safe

DESCRIPTION

This function associates a protocol address with the transport endpoint specified by *fildes* and activates that transport endpoint. In connection mode, the transport provider may begin accepting or requesting connections on the transport endpoint. In connectionless mode, the transport user may send or receive data units through the transport endpoint.

The *req* and *ret* arguments point to a **t_bind** structure containing the following members:

```
struct netbuf addr;
unsigned qlen;
```

netbuf is described in **t_connect**(3N). The **addr** field of the **t_bind** structure specifies a protocol address and the **qlen** field is used to indicate the maximum number of outstanding connect indications.

req is used to request that an address, represented by the **netbuf** structure, be bound to the given transport endpoint. **len** (see **netbuf** in **t_connect**(3N); also for **buf** and **maxlen**) specifies the number of bytes in the address and **buf** points to the address buffer. **maxlen** has no meaning for the *req* argument. On return, *ret* contains the address that the transport provider actually bound to the transport endpoint; this may be different from the address specified by the user in *req*. In *ret*, the user specifies **maxlen**, which is the maximum size of the address buffer, and **buf**, which points to the buffer where the address is to be placed. On return, **len** specifies the number of bytes in the bound address and **buf** points to the bound address. If **maxlen** is not large enough to hold the returned address, an error will result.

If the requested address is not available, or if no address is specified in *req* (the **len** field of **addr** in *req* is zero) the transport provider may assign an appropriate address to be bound, and will return that address in the **addr** field of *ret*. The user can compare the addresses in *req* and *ret* to determine whether the transport provider bound the transport endpoint to a different address than that requested.

req may be **NULL** if the user does not wish to specify an address to be bound. Here, the value of **qlen** is assumed to be zero, and the transport provider must assign an address to the transport endpoint. Similarly, *ret* may be **NULL** if the user does not care what address was bound by the provider and is not interested in the negotiated value of **qlen**. It is valid to set *req* and *ret* to **NULL** for the same call, in which case the provider chooses the address to bind to the transport endpoint and does not return that information to the user.

The **qlen** field has meaning only when initializing a connection-mode service. It specifies the number of outstanding connect indications the transport provider should support for the given transport endpoint. An outstanding connect indication is one that has been passed to the transport user by the transport provider. A value of **qlen** greater than zero is only meaningful when issued by a passive transport user that expects other users to call it. The value of **qlen** will be negotiated by the transport provider and may be changed if the transport provider cannot support the specified number of outstanding connect indications. On return, the **qlen** field in *ret* will contain the negotiated value.

This function allows more than one transport endpoint to be bound to the same protocol address (however, the transport provider must support this capability also), but it is not allowable to bind more than one protocol address to the same transport endpoint. If a user binds more than one transport endpoint to the same protocol address, only one endpoint can be used to listen for connect indications associated with that protocol address. In other words, only one **t_bind( )** for a given protocol address may specify a value of **qlen** greater than zero. In this way, the transport provider can identify which transport endpoint should be notified of an incoming connect indication. If a user attempts to bind a protocol address to a second transport endpoint with a value of **qlen** greater than zero, the transport provider will assign another address to be bound to that endpoint. If a user accepts a connection on the transport endpoint that is being used as the listening endpoint, the bound protocol address will be found to be busy for the duration of that connection. No other transport endpoints may be bound for listening while that initial listening endpoint is in the data transfer phase. This will prevent more than one transport endpoint bound to the same protocol address from accepting connect indications.

RETURN VALUES

t_bind( ) returns **0** on success. On failure, **t_bind( )** returns **–1**, **t_errno** is set to indicate the error, and possibly **errno** is set.

ERRORS

On failure, **t_errno** will be set to one of the following:

TACCES	The user does not have permission to use the specified address.
TBADADDR	The specified protocol address was in an incorrect format or contained illegal information.
TBADF	The specified file descriptor does not refer to a transport endpoint.
TBUFOVFLW	The number of bytes allowed for an incoming argument is not sufficient to store the value of that argument. The provider's state will change.
TNOADDR	The transport provider could not allocate an address. **T_IDLE** and the information to be returned in *ret* will be discarded.

TOUTSTATE	The function was issued in the wrong sequence.
TSYSERR	A system error has occurred during execution of this function, **errno** will be set to the specific error.

SEE ALSO

t_connect(3N), **t_open**(3N), **t_optmgmt**(3N), **t_unbind**(3N)

Transport Interfaces Programming Guide

NOTES

This interface is safe in multithreaded applications.