NAME

t_snd – send data or expedited data over a connection

SYNOPSIS

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

#include <tiuser.h>

int t_snd(int *fildes*, **char** **buf*, **unsigned** *nbytes*, **int** *flags***);**

MT-LEVEL

MT-Safe

DESCRIPTION

This function is used to send either normal or expedited data. *fildes* identifies the local transport endpoint over which data should be sent, *buf* points to the user data, *nbytes* specifies the number of bytes of user data to be sent, and *flags* specifies any optional flags described below.

By default, **t_snd( )** operates in synchronous mode and may wait if flow control restrictions prevent the data from being accepted by the local transport provider at the time the call is made. However, if **O_NDELAY** or **O_NONBLOCK** is set (using **t_open**(3N) or **fcntl**(2)), **t_snd( )** will execute in asynchronous mode, and will fail immediately if there are flow control restrictions.

Even when there are no flow control restrictions, **t_snd( )** will wait if **STREAMS** internal resources are not available, regardless of the state of **O_NDELAY** or **O_NONBLOCK**.

On successful completion, **t_snd( )** returns the number of bytes accepted by the transport provider. Normally this will equal the number of bytes specified in *nbytes*. However, if **O_NDELAY** or **O_NONBLOCK** is set, it is possible that only part of the data will be accepted by the transport provider. In this case, **t_snd( )** will set **T_MORE** for the data that was sent (see below) and will return a value less than *nbytes*. If *nbytes* is zero and sending of zero bytes is not supported by the underlying transport provider, **t_snd( )** will return −1 with **t_errno** set to **TBADDATA**. A return value of zero indicates that the request to send a zero-length data message was sent to the provider.

If **T_EXPEDITED** is set in *flags*, the data will be sent as expedited data, and will be subject to the interpretations of the transport provider.

If **T_MORE** is set in *flags*, or is set as described above, an indication is sent to the transport provider that the transport service data unit (**TSDU**) or expedited transport service data unit (**ETSDU**) is being sent through multiple **t_snd( )** calls. Each **t_snd( )** with the **T_MORE** flag set indicates that another **t_snd( )** will follow with more data for the current **TSDU**. The end of the **TSDU** (or **ETSDU**) is identified by a **t_snd( )** call with the **T_MORE** flag not set. Use of **T_MORE** enables a user to break up large logical data units without losing the boundaries of those units at the other end of the connection. The flag implies nothing about how the data is packaged for transfer below the transport interface. If the transport provider does not support the concept of a **TSDU** as indicated in the *info* argument on return from **t_open**(3N) or **t_getinfo**(3N), the **T_MORE** flag is not meaningful and should be ignored.

The size of each **TSDU** or **ETSDU** must not exceed the limits of the transport provider as returned by **t_open**(3N) or **t_getinfo**(3N). If the size is exceeded, a **TSYSERR** with system error **EPROTO** will occur. However, the **t_snd( )** may not fail because **EPROTO** errors may not be reported immediately. In this case, a subsequent call that accesses the transport endpoint will fail with the associated **TSYSERR**.

If **t_snd( )** is issued from the **T_IDLE** state, the provider may silently discard the data. If **t_snd( )** is issued from any state other than **T_DATAXFER**, **T_INREL**, or **T_IDLE**, the provider will generate a **TSYSERR** with system error **EPROTO** (which may be reported in the manner described above).

RETURN VALUES

On successful completion, **t_snd( )** returns the number of bytes accepted by the transport provider. On failure **t_snd( )** 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:

TBADDATA — *nbytes* is zero and sending zero bytes is not supported by the transport provider.

TBADF — The specified file descriptor does not refer to a transport endpoint.

TFLOW — **O_NDELAY** or **O_NONBLOCK** was set, but the flow control mechanism prevented the transport provider from accepting data at this time.

TNOTSUPPORT — This function is not supported by the underlying transport provider.

TSYSERR — A system error (see **intro**(2)) has been detected during execution of this function.

SEE ALSO

fcntl(2), **t_getinfo**(3N), **t_open**(3N), **t_rcv**(3N)

Transport Interfaces Programming Guide

NOTES

This interface is safe in multithreaded applications.