

HI.CommunicationEngine
Telnetクライアント
リファレンスマニュアル

株式会社ルネサス北日本セミコンダクタ

ご注意

1. 本製品(ソフトウェア製品及びその関連ソフトウェア製品を含む。以下、同じ。)の使用に際しては、「外国為替及び外国貿易法」等、技術輸出に関する日本及び関連諸国の関係法規の遵守が必要となります。
2. 弊社は、本製品の使用に際しては、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に関し、別途、個別の契約書等(マニュアルの記載を含む。以下、同じ。)にて弊社による明示的な許諾がある場合を除き、その保証または実施権の許諾を行うものではありません。また本製品を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 本製品およびその仕様、またはマニュアルに記載されている事柄については、将来、事前の予告なしに変更することがありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書(マニュアルを含む)をご確認ください。
4. 本製品の使用(マニュアル記載事項に基づくものも含む)により直接または間接に生ずるいかなる損害についても、弊社は一切の責任を負いません。また、本製品の配布に使用される搭載機器や媒体が原因の損害に対しましても、弊社は一切の責任を負いません。
5. 本製品を、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途向けには使用できません。お客様の用途がこれに該当するかどうか疑問のある場合には、事前に弊社営業担当迄ご相談をお願い致します。
6. 本製品を使用してお客様のシステム製品を設計される際には、通常予測される故障発生率、故障モードをご考慮の上、本製品の動作が原因での事故、その他の拡大損害を生じないようにフェールセーフ等の十分なシステム上の対策を講じて頂きますようお願い致します。
7. 本製品およびマニュアルの著作権は弊社が所有しております。お客様は、弊社から提供された本製品を、別途、個別の契約書等にて定める場合を除き、いかなる場合においても全体的または部分的に複写・解析・改変することはできないものとします。
8. お客様は、別途、個別の契約書等にて定める場合を除き、本製品のマニュアルの一部または全部を無断で使用、複製することはできません。
9. 弊社は、本製品を1台のコンピュータで使用する権利をお客様に対してのみ許諾します。よって、本製品を第三者へ譲渡、貸与、賃借することは許諾しないものとします。但し、別途、個別の契約書等にて定められる場合はその条件に従います。
10. 本製品をはじめ弊社製品およびその関連製品についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

μITRON は、Micro Industrial TRON の略称です。TRON は、The Realtime Operating system Nucleus の略称です。

その他、本書で登場するシステム名、製品名は各社の登録商標または商標です。

はじめに

このマニュアルは、HI.CommunicationEngine上で動作するTCP/IPネットワークアプリケーション「HI.CommunicationEngine Telnetクライアント」について説明します。

HI.CommunicationEngine Telnetクライアントは、TCP/IPマネージャを経由し、ネットワーク上のリモート端末として、遠隔操作を支援する機能を提供します。

このリファレンスマニュアルではTelnetクライアントのサービスコールとその使い方および関連事項を説明します。TCP/IPマネージャについては関連マニュアルを参照してください。

【関連マニュアル】

- ・ HI.CommunicationEngineTCP/IPマネージャリファレンスマニュアル
- ・ 使用するμITRON仕様OSのユーザーズマニュアル

目次

1. 概要	1
1.1 Telnetの機能概要.....	1
1.2 サービスタスクの実現.....	2
1.3 サービスコールのしくみ.....	3
1.4 サポートする機能.....	4
1.5 Telnetクライアント サービスコールの種類.....	6
2. Telnetサービスコールの使用方法	7
2.1 Telnetクライアントの基本フロー.....	7
2.2 Telnetクライアントの状態遷移図.....	9
2.3 端末エミュレーションの基本処理.....	10
2.3.1 キャラクタ単位モードの場合.....	10
2.3.2 擬似ラインモード、およびリアルラインモードの場合.....	10
3. Telnetサービスコール	11
3.1 初期化サービスコール.....	12
3.1.1 <i>TLN_init</i> Telnetクライアントを初期化する.....	12
3.2 基本サービスコール.....	13
3.2.1 <i>TLN_start</i> Telnetクライアントのサービスを開始する.....	13
3.2.2 <i>TLN_stop</i> Telnetクライアントのサービスを終了する.....	15
3.2.3 <i>TLN_cre_svp</i> Telnetクライアントを生成する.....	16
3.2.4 <i>TLN_del_svp</i> Telnetクライアントを削除する.....	18
3.2.5 <i>TLN_con_srv</i> Telnetサーバに接続する.....	19
3.2.6 <i>TLN_sht_srv</i> Telnetクライアントの接続を切断する.....	21
3.2.7 <i>TLN_put_dat</i> Telnetサーバにデータを送信する.....	22
3.2.8 <i>TLN_get_dat</i> Telnetサーバからデータを受信する.....	23
3.2.9 <i>TLN_set_mod</i> Telnetクライアントの動作モードを設定する.....	24
3.2.10 <i>TLN_get_mod</i> Telnetクライアントの動作モードを取得する.....	29
3.2.11 <i>TLN_snd_cmd</i> Telnetコマンドを送信する.....	32
3.2.12 <i>TLN_isu_opt</i> 任意のTelnetオプションを送信する.....	33
3.2.13 <i>TLN_can_svp</i> ペンディングしている処理をキャンセルする.....	35
4. コールバック	36
4.1 コールバックルーチン.....	37
4.1.1 <i>callback</i> Telnetクライアントイベントを通知する.....	37
(1) <i>TEV_TLN_DISCONN</i> Telnetクライアントの接続切断イベント.....	38
(2) <i>TEV_TLN_RCVDAT</i> データの受信イベント.....	39
(3) <i>TEV_TLN_SNDEMP</i> 送信バッファの空きイベント.....	40
(4) <i>TEV_TLN_BINARY</i> バイナリモード変更イベント.....	41
(5) <i>TEV_TLN_ECHO</i> エコーバック変更イベント.....	42
(6) <i>TEV_TLN_LFLOW</i> リモートフロー制御の変更イベント.....	43
(7) <i>TEV_TLN_LINEMODE</i> ラインモードの変更イベント.....	44
5. サービスコール エラーコード一覧	45

図表目次

図1-1	Telnet構成	1
図1-2	サービスタスク	2
図1-3	各APIとサービスタスクとの通信	3
図2-1	TelnetサービスポイントIDと多重化	8
図2-2	Telnetクライアントの基本フロー	8
図2-3	Telnetクライアント状態遷移図	9
表1-1	Telnetコマンド サポート機能	4
表1-2	Telnetオプション サポート機能	5
表1-3	Telnetサービスコール一覧	6
表5-1	Telnetサービスコール エラーコード一覧	45

1. 概要

1.1 Telnet の機能概要

Telnet はネットワークを介してリモート端末にログインし、アクセスする為のプロトコルです。Telnet プロトコルは Telnet クライアントと Telnet サーバ間の通信を規定したもので、Telnet クライアントから入力されたデータはネットワーク仮想端末 (NVT) のデータフォーマットに変換され、Telnet サーバに送られます。Telnet サーバでは、それをローカル端末のデータフォーマットに変換してアプリケーションに渡します。サーバからクライアントに送られる場合も同様です。

Telnet は TCP プロトコル上で動作し、ログインではユーザ名、パスワードによって認証取得する必要があります。また、サーバ/クライアント間のネゴシエーションにより、ローカルエコーの指定、転送モード、端末タイプの設定等、両者間の通信モードを変更することができます。

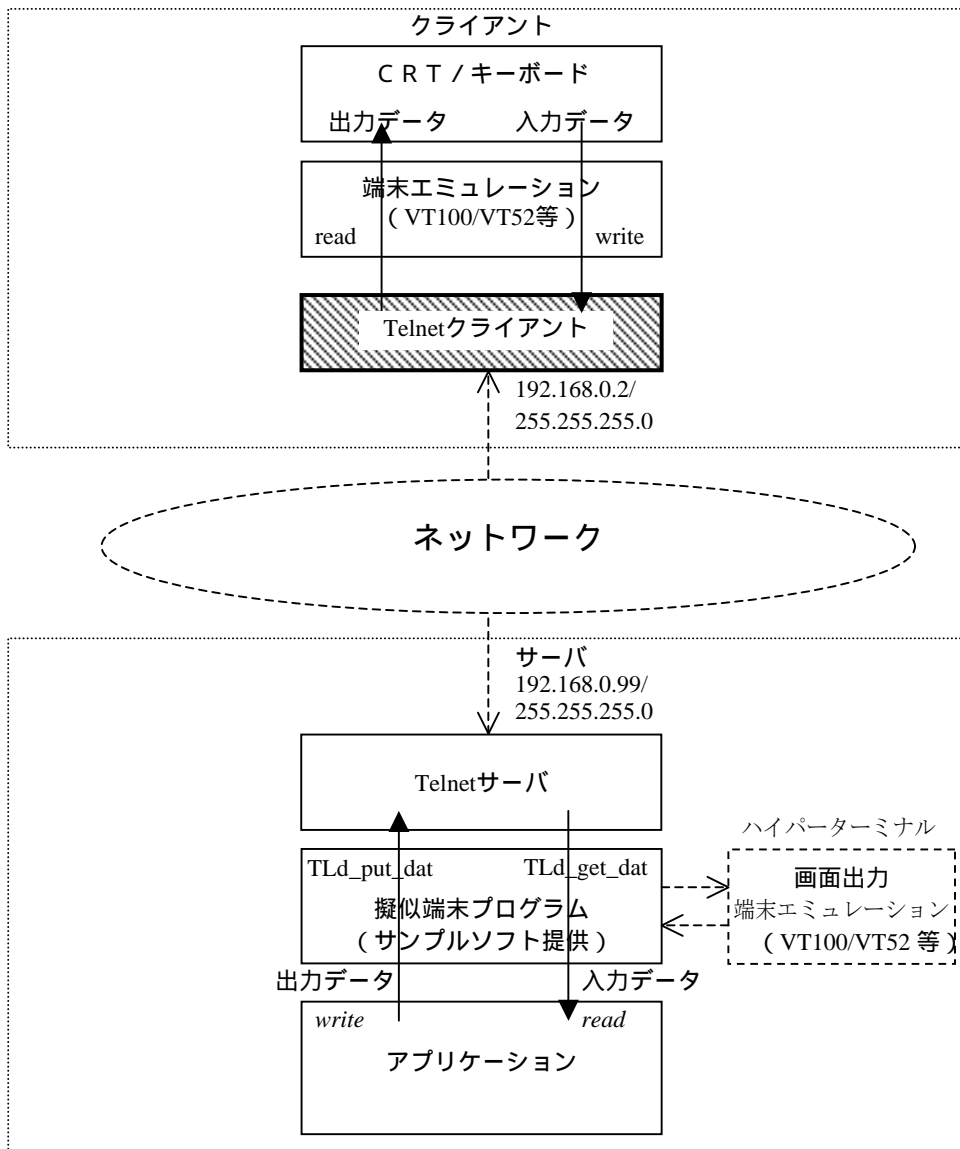


図1-1 Telnet構成

1.2 サービスタスクの実現

Telnet クライアントのサービスは「タスク」として実現します。図 1-2にサービスタスクの概念を示します。

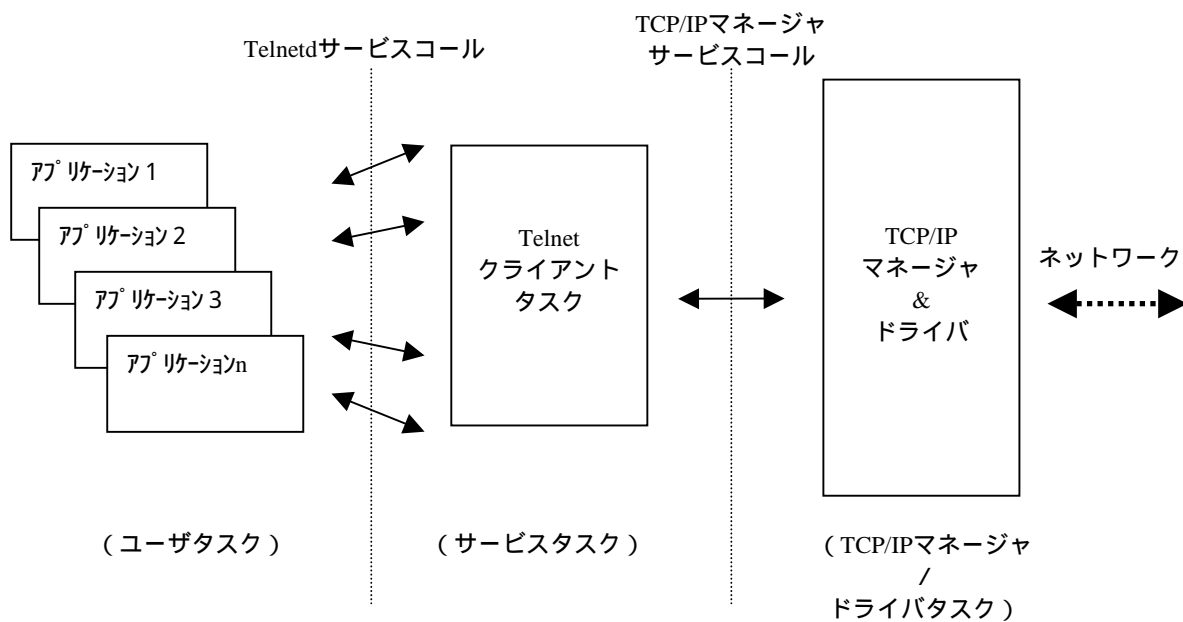


図1-2 サービスタスク

1.3 サービスコールのしくみ

各サービスコールは、拡張サービスコール形式で呼び出されます（TLN_initのみサブルーチン形式）。各サービスコールとサービスタスクとはμITRON仕様OSのメールボックス、およびイベントフラグ機能を利用してデータのやり取りを行います。

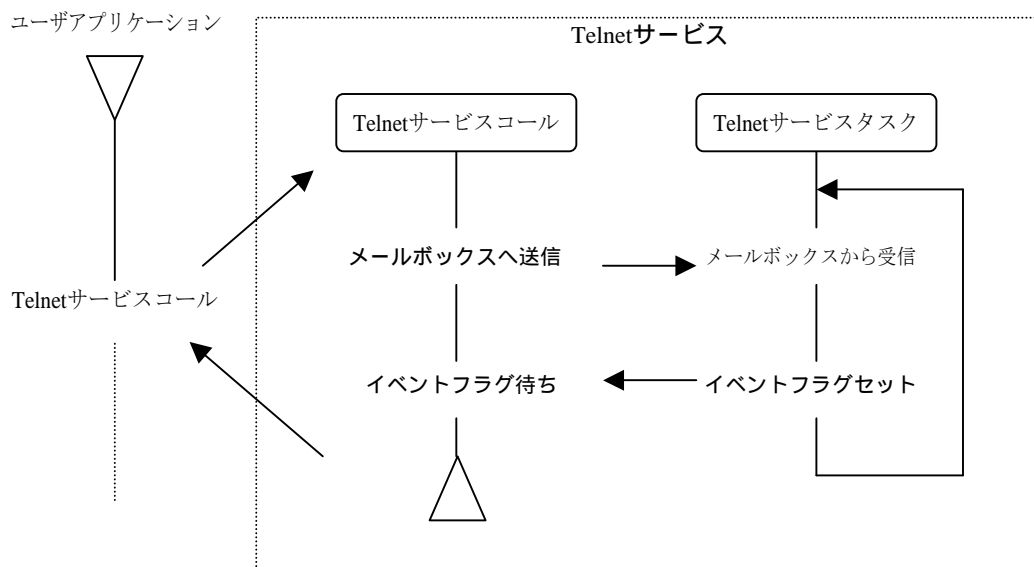


図1-3 各APIとサービスタスクとの通信

本 Telnet クライアントではμITRON仕様OSのリソースのうち、1タスク、1メールボックス、1イベントフラグ、および13個の拡張サービスコールを使用します。

1.4 サポートする機能

本 Telnet クライアントの基本機能は次のとおりです。

- (1) Telnet クライアントはネットワークを経由して Telnet サーバにリモートログインすることができます。
- (2) マルチタスクプログラムで複数の Telnet クライアント(最大 10 個)を同時に制御することができます。また、サービスコールを複数のタスクから同時に発行することができます。
- (3) サーバとの Telnet オプション交渉を自動で行います。また、サポートしている任意の Telnet オプションを送信することができます。
- (4) 以下の Telnet 機能をサポートしています。ただし、接続するサーバが未サポートの機能は使用できません。
 - ・ 表 1-1に示す Telnet コマンドをサポートしています。
 - ・ 表 1-2に示す Telnet オプションをサポートしています。

本バージョンの Telnet クライアントには、次の機能的な制限があります。

- (1) 端末プログラム、および端末エミュレーション機能は実装していません。
- (2) ラインモードでの行データ編集機能はサポートしていません(ユーザが編集する必要があります)。
- (3) ローカルエコーはサポートしていません(ユーザがローカルエコーする必要があります)。

表1-1 Telnetコマンド サポート機能

項目	説明	サポート有無		
		送信	受信	
Telnet コマンド	EOF	ファイル終端		×
	SUSP	プロセス一時停止		×
	ABORT	プロセス中断		×
	EOR	レコード終端		×
	SE	サブオプション終了		
	NOP	何もしない		×
	DM	データマーク		
	BRK	ブレーク		×
	IP	プロセス割り込み		×
	AO	出力停止		×
	AYT	相手存在確認		×
	EC	文字削除		×
	EL	行削除		×
	GA	進行抑止		×
	SB	サブオプション開始		
	DO	オプション交渉		
	DONT	オプション交渉		
	WILL	オプション交渉		
	WONT	オプション交渉		

注：サポート有無の欄に 印があるものがサポートする機能で、×印は未サポートの機能です。

表1-2 Telnetオプション サポート機能

項目	説明	サポート有無		
		送信	受信	
Telnet オプション	TRANSMIT-BINARY	バイナリ転送		
	ECHO	エコーバック		×
	RCP	再接続オプション	×	×
	SUPPRESS-GA	進行抑止		
	NAMS	メッセージ長の調停	×	×
	STATUS	Telnet オプション状態通知		
	TIMING-MARK	タイミングマーク		
	RCTE	送信&エコーのリモート制御	×	×
	NAOL	行幅出力の調停	×	×
	NAOP	ページサイズ出力の調停	×	×
	NAOCRD	キャリッジリターン出力の調停	×	×
	NAOHTS	水平タブストップ出力の調停	×	×
	NAOHTD	水平タブキャラクタ処理の調停	×	×
	NAOFFD	フォームフィード出力処理の調停	×	×
	NAOVTS	垂直タブストップ出力の調停	×	×
	NAOVTD	垂直タブ出力処理の調停	×	×
	NAOLFD	ラインフィード出力の調停	×	×
	EXTEND-ASCII	拡張 ASCII	×	×
	LOGOUT	強制ログアウト		×
	BM	バイトマクロ	×	×
	DET	データエントリ端末	×	×
	SUPDUP	SUPDUP ディスプレイ制御	×	×
	SUPDUP-OUTPUT	SUPDUP 出力	×	×
	SEND-LOCATION	ロケーション ID 送信	×	×
	TERMINAL-TYPE	ターミナルタイプ		×
	END-OF-RECORD	レコード終端を示す文字の選択	×	×
	TUID	TACACS ユーザ識別	×	×
	OUTMRK	マーク出力	×	×
	TTYLOC	端末ロケーション番号	×	×
	3270-REGIME	TELNET3270 様式	×	×
	X3-PAD	X3PAD	×	×
	NAWS	ウィンドウサイズの調停		×
	TERMINAL-SPEED	端末スピード		×
TOGGLE-FLOW-CONTROL	リモートフロー制御			
LINEMODE	ラインモード			
XDISPLOC	X ディスプレイロケーション	×	×	
ENVIRON	環境変数	×	×	
AUTHENTICATION	認証	×	×	
ENCRYPTION	暗号	×	×	
NEW-ENVIRON	新環境変数	×	×	
TN3270E	TELNET3270 拡張	×	×	
EXOPL	拡張オプション	×	×	

注：サポート有無の欄に 印があるものがサポートする機能で、×印は未サポートの機能です。

1.5 Telnet クライアント サービスコールの種類

以下に本 Telnet クライアントのサービスコールの一覧を示します。

表1-3 Telnetサービスコール一覧

区分	サービスコール名称	サービスコールの機能
初期化サービスコール	TLN_init	Telnetクライアントのサービスを初期化する
基本サービスコール	TLN_start	Telnetクライアントのサービスを開始する
	TLN_stop	Telnetクライアントのサービスを終了する
	TLN_cre_svp	Telnetクライアントのサービスポイントを生成する
	TLN_del_svp	Telnetクライアントのサービスポイントを削除する
	TLN_con_srv	Telnetサーバに接続する
	TLN_sht_srv	Telnetサーバとの接続を切断する
	TLN_put_dat	Telnetサーバにデータを送信する
	TLN_get_dat	Telnetサーバからデータを受信する
	TLN_set_mod	Telnetクライアントの動作モードを設定する
	TLN_get_mod	Telnetクライアントの動作モードを取得する
	TLN_isu_opt	Telnetオプションを送信する
	TLN_snd_cmd	Telnetコマンドを送信する
	TLN_can_svp	Telnetクライアントのペンディング処理のキャンセル
コールバックルーチン	TLN_callback (仮称)	Telnetクライアントイベントの通知 (ユーザ実装ルーチン)

2. Telnet サービスコールの使用方法

Telnet サービスコールを使用するには HI.CommunicationEngine TCP/IP マネージャおよびドライバプログラムを動作可能な状態にする必要があります。TCP/IP マネージャおよびドライバプログラムの使用方法に関しては、HI.CommunicationEngine TCP/IP マネージャ リファレンスマニュアル、および各ドライバプログラムのマニュアルを参照してください。

Telnet を使うプログラムでは次のヘッダファイルをインクルードしてください。

```
#include "tcpip??.h"          /* TCP/IP マネージャ定義ファイル */
#include "tln02.h"           /* Telnet 定義ファイル */
```

注：??は HI.CommunicationEngine 基本セットのバージョンを表します（Ver2.0：02、Ver3.0：03）。

一般的な Telnet クライアントの使用方法を以下に説明します。

2.1 Telnet クライアントの基本フロー

- (1) 最初に TLN_init をコールする必要があります。TLN_init はシステムリセットの際などに一度のみコールします。TLN_init にて各 Telnet サービスコールが μ ITRON 仕様 OS の拡張サービスコールとして定義されます。（図 2-2 ）
- (2) TLN_start をコールすると、Telnet が使用するタスク、メモリ等のリソースが確保され、Telnet クライアントの各サービスコールが使用可能となります。（図 2-2 ）
- (3) TLN_cre_svp をコールすると Telnet サービスポイントが生成され、Telnet サービスポイントを使用した各 Telnet サービスコールが使用できるようになります。（図 2-2 ）また、Telnet サービスポイントを複数作成することによって Telnet 接続を多重化することができます。（図 2-1 ）
- (4) TLN_con_srv をコールすることで Telnet サービスポイントから Telnet サーバへ接続を行います。接続がオープンされると、サーバとの間でいくつかのオプション交渉が開始されます。この際、クライアントはキャラクタ単位モードに設定されます。キャラクタ単位モードでは、Telnet サービスポイントから送信されたデータは、通常、サーバ側でエコーされます。（図 2-2 ）
- (5) Telnet サーバからデータを受信するには TLN_get_dat を使用します。Telnet サーバにデータを送信するには TLN_put_dat を使用します。（図 2-2 ）
- (6) クライアントから切断するには TLN_sht_svp をコールします。切断された Telnet サービスポイント ID はフリーとなり、以降のサーバ接続にて再び使用できます。また、サーバ側から切断された場合も同様です。（図 2-2 ）
- (7) Telnet サービスポイントを削除するには TLN_del_svp をコールします。（図 2-2 ）
- (8) Telnet クライアントを停止する場合は、TLN_stop をコールします。Telnet が使用するタスク、メモリ等のリソースが解放されます。Telnet クライアントを停止した後であれば、再度 TLN_start をコールすることができます。（図 2-2 ）

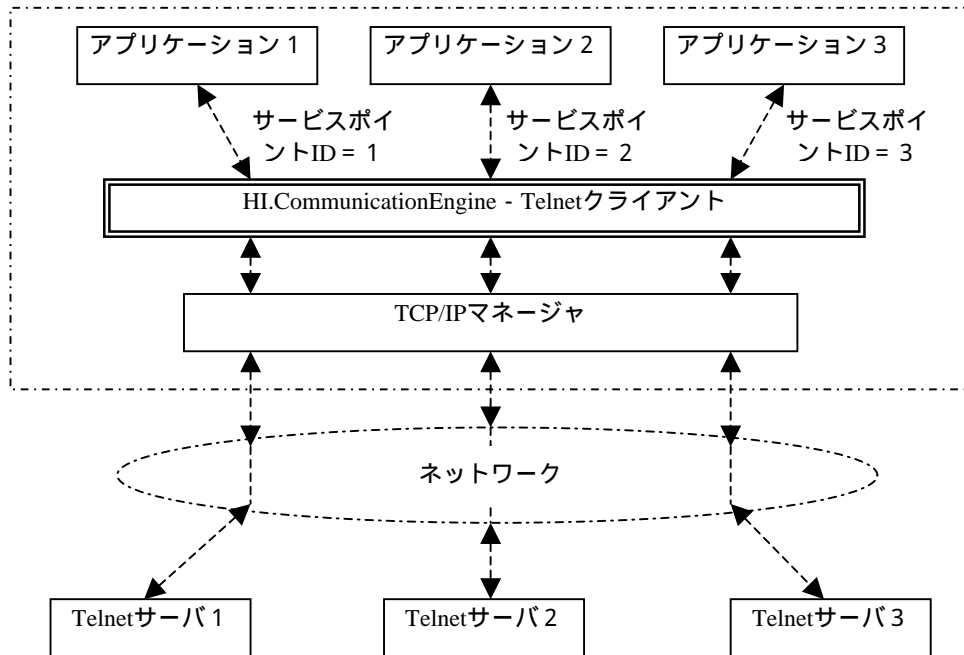


図2-1 TelnetサービスポイントIDと多重化

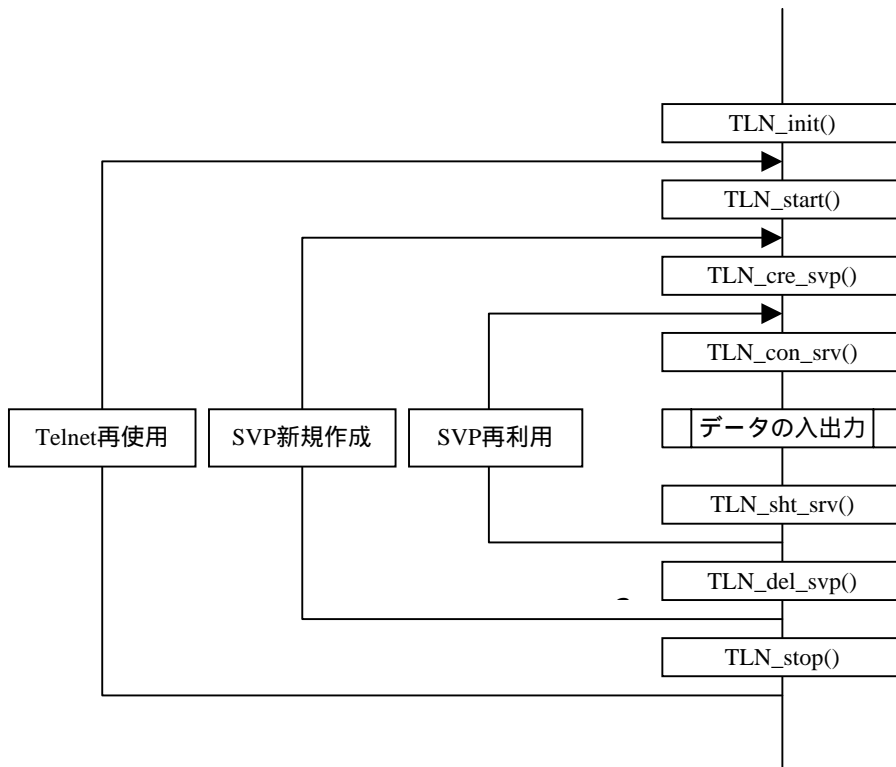


図2-2 Telnetクライアントの基本フロー

2.2 Telnet クライアントの状態遷移図

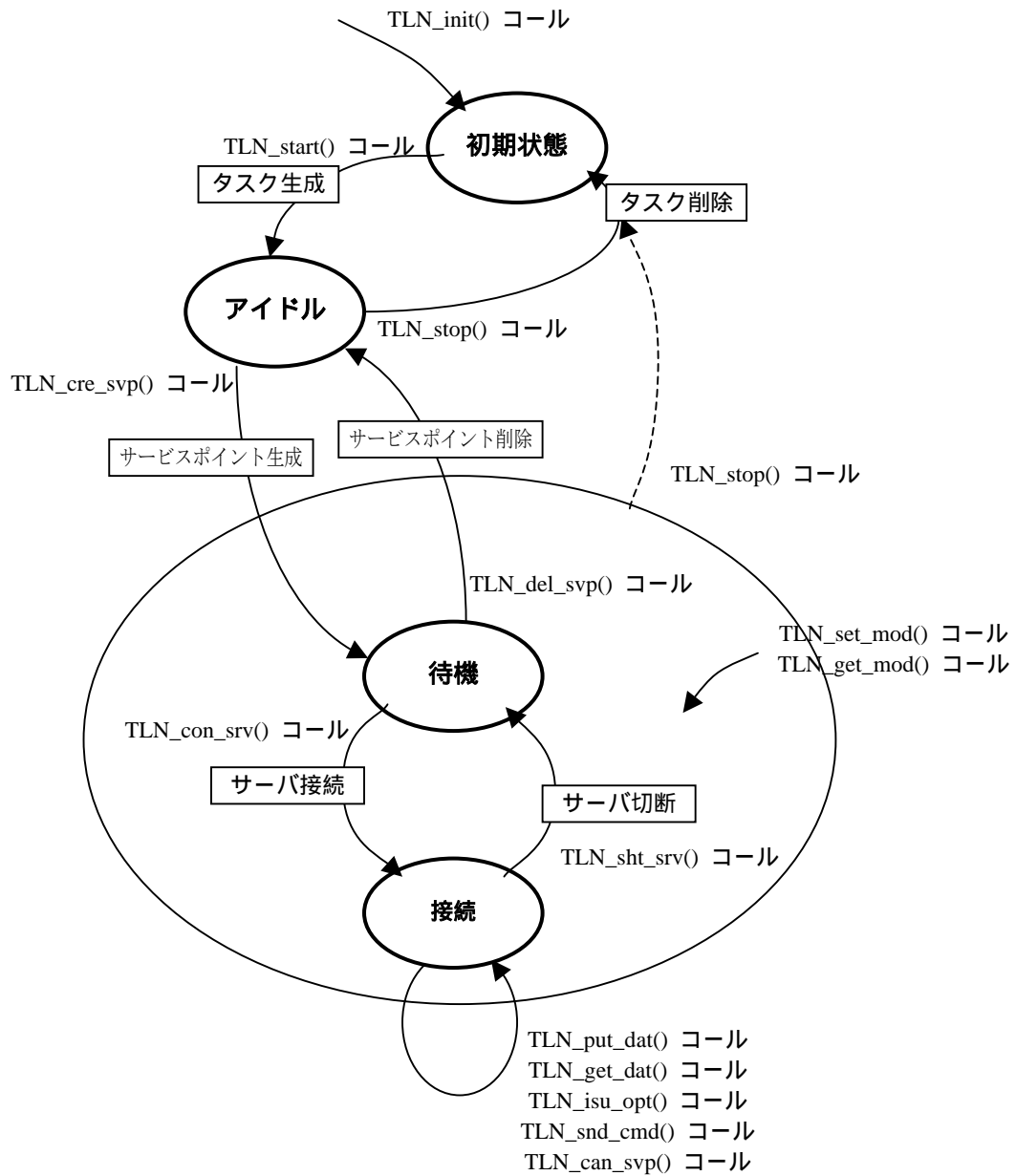


図2-3 Telnetクライアント状態遷移図

2.3 端末エミュレーションの基本処理

2.3.1 キャラクタ単位モードの場合

- (1) コンソールにより入力されたデータを TNL_snd_dat によりサーバに逐次送信します。通常、データはサーバ側でエコーされる為、ローカルでエコーする必要はありません。
- (2) TLN_snd_dat で送信されたデータは、サーバによりエコーされます。エコーされたデータと共にサーバから送信されたデータを TLN_rcv_dat により取得し、画面等に出力してください。
- (3) パスワード等の入力のため、サーバにエコーをさせたくない場合は、TELNET ECHO オプションをサーバへ送信します(TLN_isu_opt の optname パラメータを DONT 指定で発行する)。
- (4) パスワード等の入力が終わったら、サーバにエコーを開始するよう、再度 TELNET ECHO オプションをサーバへ送信します(TNL_isu_opt の optname パラメータを DO 指定で発行する)。

2.3.2 擬似ラインモード、およびリアルラインモードの場合

- (1) コンソールにより入力されたデータを編集し、行単位に TNL_snd_dat によりサーバに送信してください(必ずしも行単位に送信する必要はありません)。ラインモードの場合は、サーバはデータをエコーしません。コンソールにより入力されたデータはローカルでエコー^{*1}する必要があります。これによりクライアントとサーバの間でやり取りされるセグメントの数を減らすことができます。
- (2) サーバから送信されたデータは TLN_rcv_dat により取得し、画面等に出力してください。
- (3) パスワード等の入力でエコーが不要な場合はローカルエコー^{*1}を停止してください。
- (4) パスワード等の入力が終わったら、ローカルエコー^{*1}を開始してください。

1 : ローカルエコーは、Telnet クライアント側のユーザプログラムで行う必要があります。本 Telnet クライアントではローカルエコー処理は行いません。

3. Telnet サービスコール

各サービスコールについての説明は次の形式で行います。

No.	サービスコール名	機能	【発行可能なシステム状態 ^{*1} 】
C 言語インタフェース			
サービスコール呼出し形式			
パラメータ			
型	パラメータ	パラメータの意味	
・	・	・	
・	・	・	
・	・	・	
リターンパラメータ			
型	パラメータ	パラメータの意味	
・	・	・	
・	・	・	
パケットの構造			
リターン値 / エラーコード			
リターン値またはニモニック	リターン値またはエラーコードの意味		
・	・		
・	・		
・	・		
解 説			
.....			

*1発行可能なシステム状態を以下のアルファベットで示します

- T : タスク実行状態
- D : ディスパッチ禁止状態
- L : CPUロック状態
- I : 非タスク部実行状態

なお、各状態の詳細は各μITRON仕様OSのユーザズマニュアルを参照してください。

発行可能なシステム状態以外の状態でサービスコールを発行した場合、システムの正常な動作は保証されません。

3.1 初期化サービスコール

3.1.1 TLN_init Telnet クライアントを初期化する

【 T / D / I 】

C 言語インタフェース

void TLN_init (void);

パラメータ

なし

リターンパラメータ

なし

リターン値 / エラーコード

なし

解 説

Telnetクライアントで使用する内部変数を初期化し、各TelnetサービスコールをμITRON仕様OSの拡張サービスコールに登録します。

Telnetクライアントの各サービスコールを使用する前に、必ず本サービスコールを呼び出す必要があります。

拡張サービスコールの機能コード不足等で拡張サービスコールが登録できなかった場合でもエラーコードは返りません。この場合、TLN_startをコールすると拡張サービスコールが未定義であるエラーコード(E_RSFN)、またはシステムエラー(E_SYS)が返されます。

3.2 基本サービスコール

3.2.1 TLN_start Telnet クライアントのサービスを開始する

【T】

C 言語インタフェース

```
ER ercd = TLN_start ( T_TLN_STA *pk_sta );
```

パラメータ

T_TLN_STA	*pk_sta;	Telnetクライアント情報を格納した領域の先頭アドレス
-----------	----------	------------------------------

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

パケットの構造

```
typedef struct{          Telnetクライアント情報
    T_TLN_RSC            resource;   リソース情報
    T_TLN_OSC            osconf;     OSコンフィグレーション情報
    H                    Perrno;     発生したエラーの詳細を示す番号
} T_TLN_STA;
```

```
typedef struct{          リソース情報
    ID                  maxsvpid;    最大TelnetサービスポイントID
    UW                  *memadr;     使用するメモリの先頭アドレス
    W                   memlen;     使用可能なメモリの長さ
    UH                  minport;    自動生成する最小のポート番号
    UH                  maxport;    自動生成する最大のポート番号
    UH                  curport;    自動生成するポート番号の初期値
} T_TLN_RSC;
```

```
typedef struct{          OSコンフィグレーション情報
    ATR                 tskatr;     Telnetクライアントタスク属性
    ID                  tskid;      TelnetクライアントタスクID番号
    PRI                 tskpri;     Telnetクライアントタスク優先度
    INT                 stksz;      Telnetクライアントタスクスタックサイズ
    ID                  mbxid;      TelnetクライアントメールボックスID番号
    ID                  evfid;      TelnetクライアントイベントフラグID番号
} T_TLN_OSC;
```

リターン値 / エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (pk_staおよびmemadrが0または4の倍数以外、maxsvpid 0、maxsvpid > 10、minport > maxport、curport < minport、curport > maxport)
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態エラー (すでにサービスが開始している)
E_SYS	システムエラー (OSのリソース確保に失敗)

解 説

Telnetクライアントのサービスを開始します。

maxsvpidには使用するTelnetサービスポイントIDの最大値を指定します。TelnetサービスポイントIDは同時に利用するTelnet接続の数分だけ必要となります。(注1)

Telnetサービスコールでは、必要となるワークメモリをmemadrで指定されたメモリアドレスから切り出して使用します。切り出したメモリの長さをmemlenに返します。最大TelnetサービスポイントIDに比例して使用するワークメモリが多くなります。

ワークメモリはTLN_stopをコールした後、解放されます。解放前にワークメモリの内容を書き換えた場合の動作は保証しません。

指定したmemlenが必要なメモリサイズみ満たない場合は、エラーコードとしてE_NOMEMを返します。

Telnetクライアントが必要とするワークメモリのサイズは以下の計算式にて算出できます。

$$\text{ワークメモリサイズ} = 452 + \text{maxsvpid} \times 4300 \quad (\text{バイト})$$

minport、maxportには、Telnetクライアント用に自動生成するポート番号（TLN_con_srvでmyportに0を指定した場合）の範囲を指定します。この範囲内の任意のポート番号を自動生成します。minportおよびmaxportに0を指定すると1025から65535の範囲のポート番号をデフォルトとして使用します。

curportには、自動生成するポート番号の初期値を指定します。curportに0を指定した場合、前回使用したポート番号 + 1 を初期値として使用します。

tasktr、tskid、tskpri、stkszには、Telnetクライアントタスクのタスク属性、タスクID番号、タスク優先度、タスクスタックサイズを指定します。tskidに0を指定した場合、OSのID番号自動割付け機能によりID番号を割り付け、割り付けたID番号をtskidに格納します。

mbxid、evfidには、Telnetクライアントが生成するメールボックスのID番号、およびイベントフラグのID番号を指定します。mbxid、evfidに0を指定した場合は、OSのID番号自動割付け機能によりID番号を割り付け、割り付けたID番号をmbxid、evfidに格納します。

指定したパラメータが不正の場合は、perrnoに詳細エラーコードを格納し、エラーコードとしてE_PARを返します。

OSのリソース確保に失敗した場合は、perrnoに詳細エラーコードを格納し、エラーコードとしてE_SYSを返します。

TLN_initコール時に拡張サービスコールの登録に失敗している場合には、perrnoに登録に失敗した拡張サービスコールの機能コードを格納し、エラーコードとしてE_SYSを返します。

すでにTelnetクライアントが開始している状態で本サービスコールをコールした場合、エラーコードとしてE_OBJを返します。

リターン値がE_PAR、またはE_SYSの場合、発生したエラーの詳細を示す番号がperrnoに格納されています。perrnoに格納される番号の種類を以下に示します。ただし、pk_staが不正なアドレスの場合、perrnoには格納されません。

リターン値がE_PARの場合：

perrno	エラー内容
-1	maxsvpidが11以上、または0以下
-2	memadrが0または4の倍数以外
-3	minportがmaxportより大きい、またはcurportが不正
-255	その他のパラメータエラー

リターン値がE_SYSの場合：

perrno	エラー内容
-1	メールボックスの生成に失敗
-2	イベントフラグの生成に失敗
-3	タスクの生成に失敗
-4	タスクの起動に失敗
-255	その他のシステムエラー
1以上	TLN_initにて登録に失敗した拡張サービスコールの機能コード

（注1）TCP/IPマネージャの最大通信端点数、最大受付口数、およびTCP/IPマネージャを使用している他プログラムの通信端点、受付口の使用状況によって同時に生成できるサービスポイントの数が最大TelnetサービスポイントIDより少なくなる場合があります。

3.2.2 TLN_stop

Telnet クライアントのサービスを終了する

【T】

C 言語インタフェース

```
ER ercd = TLN_stop ( void );
```

パラメータ

なし

リターンパラメータ

ER ercd リターン値またはエラーコード

リターン値 / エラーコード

E_OK 正常終了

解 説

Telnetクライアントのサービスを終了し、TLN_startで切り出したワークメモリを解放します。
クライアントがまだ接続されている場合は全て強制切断されます。

全てのTelnetサービスポイントに対し、ペンディング中の処理がある場合は強制終了します。強制終了されたサービスコールは、エラーコードとしてE_RLWAIを返します。

3.2.3 TLN_cre_svp Telnet クライアントを生成する

【 T 】

C 言語インタフェース

```
ER_UINT ercd = TLN_cre_svp ( ID svpid , T_TLN_CRE *pk_cre );
```

パラメータ

ID	svpid	TelnetサービスポイントID
T_TLN_CRE	*pk_cre	Telnetクライアントの生成情報を格納した領域のアドレス

リターンパラメータ

ER_UINT	ercd	リターン値またはエラーコード
H	pk_cre->perrno	発生したエラーの詳細を示す番号

パケットの構造

```
typedef struct{
    T_TLN_WBUF wbuf;          Telnetクライアントのウィンドウバッファ情報
    FP        callback;      イベント通知用コールバックルーチンのアドレス
    H         perrno;        発生したエラーの詳細を示す番号
} T_TLN_CRE;

typedef struct{
    VP        sbuf;          Telnetクライアント送信用ウィンドウバッファの先頭アドレス
    INT       sbufsz;        Telnetクライアント送信用ウィンドウバッファサイズ
    VP        rbuf;          Telnetクライアント受信用ウィンドウバッファの先頭アドレス
    INT       rbufsz;        Telnetクライアント受信用ウィンドウバッファサイズ
} T_TLN_WBUF;
```

リターン値 / エラーコード

正の値	正常終了 (TelnetサービスポイントID)
E_PAR	パラメータエラー (pk_creが4の倍数以外、各ウィンドウバッファの先頭アドレスが奇数、各ウィンドウバッファのサイズが2047以下、callbackが奇数)
E_ID	不正ID番号 (svpid < 0、svpid > maxsvpid)
E_NOID	ID番号不足 (TelnetサービスポイントID、またはTCP通信端点IDが不足)
E_OBJ	オブジェクト状態不正 (指定したTelnetサービスポイントIDは既に生成済み)

解 説

Telnetクライアントを生成します。

正常終了するとリターン値としてTelnetサービスポイントIDを返します。以降のTelnetサービスを使用する場合に必要となります。

svpidには新たに生成するTelnetサービスポイントIDを指定します。svpidに0を指定した場合は、未生成のTelnetサービスポイントIDを検索して生成します。

sbuf、sbufsz、rbuf、rbufszには、TelnetクライアントにてTCP/IPマネージャが使用する送信用ウィンドウバッファの先頭アドレスとサイズ、および受信用ウィンドウバッファの先頭アドレスとサイズを指定します。sbufsz、rbufszの推奨値は共に2Kバイト以上です。

各ウィンドウバッファは、Telnetサービスポイント毎に異なるエリアを指定する必要があります。指定したウィンドウバッファが異常の場合、正常な動作は保証しません。

callbackには、イベント通知用コールバックルーチンのアドレスを指定します。全てのTelnetサービスコールにおいて、システムイベントやTelnetサーバからのオプション受信イベントが発生した場合、このコールバックルーチンにより通知されます。callbackはTelnetサービスタスクレベルにてコールされます。callbackに0を指定した場合、コールバックによる通知は行いません。コールバックルーチンの詳細に関しては「4 コールバック」の項を参照してください。

svpidに0を指定した場合でTelnetサービスポイントIDに空きがない場合は、エラーコードとしてE_NOIDを返します。

リターン値がE_PARの場合、発生したエラーの詳細を示す番号がpernoに格納されています。pernoに格納される番号の種類を以下に示します。ただし、pk_creが不正なアドレスの場合、pernoには格納されません。

perno	エラー内容
-1	sbufが奇数、またはsbufszが2047以下
-2	rbufが奇数、またはrbufszが2047以下
-5	callbackが奇数
-255	その他のエラー

3.2.4 TLN_del_svp Telnet クライアントを削除する

【 T 】

C 言語インタフェース

```
ER ercd = TLN_del_svp ( ID svpid );
```

パラメータ

ID	svpid	TelnetサービスポイントID
----	-------	------------------

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値 / エラーコード

E_OK	正常終了
E_ID	不正ID番号 (svpid = 0、svpid > maxsvpid)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態不正 (指定したTelnetサービスポイントがサーバに接続中)

解 説

Telnetクライアントを削除します。

svpidには削除するTelnetクライアントに対応したTelnetサービスポイントIDを指定します。この値はTLN_cre_svpで返された値を使用します。

サーバに接続中のTelnetサービスポイントを削除しようとした場合、エラーコードとしてE_OBJを返します。

3.2.5 TLN_con_srv Telnet サーバに接続する

【 T 】

C 言語インタフェース

```
ER ercd = TLN_con_srv ( ID svpid , T_TLN_CONN *pk_conn , TMO tmout );
```

パラメータ

ID	svpid	TelnetサービスポイントID
T_TLN_CONN	*pk_conn	Telnetサーバ接続情報を格納した領域のアドレス
TMO	tmout	タイムアウト値

リターンパラメータ

ER	ercd	リターン値またはエラーコード
H	pk_conn->perrno	発生したエラーの詳細を示す番号

パケットの構造

```
typedef struct{
    T_TLN_IPADR  ipaddr;      IPアドレスおよびポート番号
    H            Perrno;      発生したエラーの詳細を示す番号
} T_TLN_CONN;

typedef struct{
    UW          myaddr;       TelnetクライアントのIPアドレス
    UH          myport;       Telnetクライアントのポート番号
                                (0の場合、ポート番号自動割付)
    UW          svraddr;       接続するTelnetサーバのIPアドレス
    UH          svrport;       接続するTelnetサーバのポート番号
                                (0の場合、デフォルトポート番号23を使用)
} T_TLN_IPADR;
```

リターン値 / エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (pk_connが4の倍数以外、IPアドレスが不正、tmout < -1)
E_ID	不正ID番号 (svpid = 0、svpid > maxsvpid)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態不正 (指定したTelnetサービスポイントは既に接続済み、またはポート番号既使用)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除
E_CLS	Telnetサーバに接続できない

解 説

Telnetサーバに接続します。

svpidにはサーバと接続するTelnetサービスポイントIDを指定します。

myaddrには、TelnetクライアントのIPアドレスを32ビット (4バイト) で指定します。

myportには、Telnetクライアントのポート番号を指定します。myportに0を指定すると、自動生成したポート番号を使用します (自動生成についてはTLN_startの解説を参照してください)。

svraddrには、TelnetサーバのIPアドレスを32ビット (4バイト) で指定します。

svrportには、Telnetサーバのポート番号を指定します。svrportに0を指定すると、デフォルトポート番号23を使用します。

tmoutには、Telnetサーバからの応答待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間(単位はms)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。ただし、本サービスコールはTelnetサーバからの応答メッセージを待つため、必ず待ち状態になります。そのため、ポーリング指定では処理を行わず、直ちにエラーコードとしてE_TMOUTを返します。

リターン値がE_PARの場合、発生したエラーの詳細を示す番号をpernoに格納します。pernoに格納される番号の種類を以下に示します。ただし、pk_connが不正なアドレスの場合、pernoには格納しません。

perno	エラー内容
-1	myaddrが0または0xffffffff
-2	svraddrが0または0xffffffff
-4	tmoutが-2以下
-255	その他のエラー

3.2.6 TLN_sht_srv Telnet クライアントの接続を切断する

【 T 】

C 言語インタフェース

```
ER ercd = TLN_sht_srv ( ID svpid , TMO tmout );
```

パラメータ

ID	svpid	TelnetサービスポイントID
TMO	tmout	タイムアウト値

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値 / エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (tmout < -1)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (サービスが開始していない、または接続状態でない)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除

解 説

接続しているサーバから切断します。

svpidには切断するTelnetサービスポイントIDを指定します。

tmoutにはクライアントからの応答待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間(単位はms)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。

ポーリング指定、および待ち時間指定によりタイムアウトが発生した場合は、RST送信による強制切断を行い、リターン値としてE_TMOUTを返します。その場合、ペンディング中の処理はすべて強制終了します。強制終了されたサービスコールは、エラーコードとしてE_RLWAIを返します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

指定したTelnetサービスポイントIDが未生成の場合は、エラーコードとしてE_NOEXSを返します。

接続状態以外のサービスポイントに対して本サービスコールを発行した場合は、エラーコードとしてE_OBJを返します。

3.2.7 TLN_put_dat Telnet サーバにデータを送信する

【 T 】

C 言語インタフェース

```
ER_UINT sdatlen = TLN_put_dat ( ID svpid , VP data , INT len , TMO tmout );
```

パラメータ

ID	svpid	TelnetサービスポイントID
VP	data	送信するデータの先頭アドレス
INT	len	送信したいデータの長さ
TMO	tmout	タイムアウト値

リターンパラメータ

ER_UINT	sdatlen	リターン値またはエラーコード
---------	---------	----------------

リターン値 / エラーコード

正の値	正常終了 (送信バッファに入れたデータの長さ)
E_PAR	パラメータエラー (dataが0、len 0、tmout < -1)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (サービスが開始していない、または接続状態でない)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除
E_CLS	接続切断

解 説

dataで指定されたアドレスからlenの長さ分のデータをTelnetサーバに送信します。

送信データが送信バッファに入った時点で、本サービスコールからリターンします。

空いている送信バッファ長が送信しようとしたデータ長よりも短い場合、送信バッファが一杯になるまで送信バッファにデータを入れ、送信バッファに入れたデータの長さを返します。送信バッファに空きがない場合には、空きが生じるまで待ち状態になります。

svpidには送信に使用するTelnetサービスポイントIDを指定します。

tmoutには送信バッファが空くまでの待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間 (単位はms)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。

送信バッファに空きが無い状態で、ポーリングを指定した場合、および待ち時間指定によるタイムアウトが発生した場合は、データの送信は行わずにエラーコードとしてE_TMOUTを返します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なTelnetサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

指定したTelnetサービスポイントが接続状態以外の場合は、エラーコードとしてE_OBJを返します。

TCPコネクションが切断された場合は、エラーコードとしてE_CLSを返します。この場合、受信データがあればTLN_get_datにてデータを取得することができますが、TLN_put_datにてデータを送信することはできません。

本サービスコールがペンディング状態でない場合に、コールバックにより送信バッファの空きイベントを通知することができます。コールバックルーチンの詳細に関しては「4 コールバック」の項を参照してください。

3.2.8 TLN_get_dat Telnet サーバからデータを受信する

【 T 】

C 言語インタフェース

```
ER_UINT rdatlen = TLN_get_dat ( ID svpid , VP data , INT len , TMO tmout );
```

パラメータ

ID	svpid	TelnetサービスポイントID
VP	data	受信データを格納する領域の先頭アドレス
INT	len	受信データを格納する領域の長さ
TMO	tmout	タイムアウト値

リターンパラメータ

ER_UINT	rdatlen	リターン値またはエラーコード
---------	---------	----------------

リターン値 / エラーコード

正の値	正常終了 (受信したデータのサイズ)
0	データ終結 (接続状態でFINを受信し、受信バッファが空)
E_PAR	パラメータエラー (dataが0、len 0、tmout < -1)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (サービスが開始していない、または接続状態でない)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除
E_CLS	接続切断

解 説

Telnetサーバからデータを受信し、dataで指定されたアドレスに格納します。

受信バッファに入ったデータを取り出した時点で、本サービスコールからリターンします。

受信バッファに入っているデータ長が受信しようとしたデータ長よりも短い場合、受信バッファが空になるまでデータを取り出し、取り出したデータの長さを返します。受信バッファが空の場合には、データを受信するまで待ち状態となります。

svpidには、受信に使用するTelnetサービスポイントIDを指定します。

tmoutには、Telnetサーバからの受信待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間(単位はms)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。

受信データが無い状態で、ポーリングを指定した場合、および待ち時間指定によるタイムアウトが発生した場合は、エラーコードとしてE_TMOUTを返します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なTelnetサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

指定したTelnetサービスポイントが接続状態以外の場合は、エラーコードとしてE_OBJを返します。

TCPコネクションが切断された場合は、エラーコードとしてE_CLSを返します。

サーバからFINを受信し、受信バッファにデータがない場合は、リターン値として0を返します。

本サービスコールがペンディング状態でない場合、コールバックによりデータの受信イベントを通知することができます。コールバックルーチンの詳細に関しては「4 コールバック」の項を参照してください。

3.2.9 TLN_set_mod Telnet クライアントの動作モードを設定する

【 T 】

C 言語インタフェース

```
ER ercd = TLN_set_mod ( ID svpid , INT modname , VP modval , TMO tmout);
```

パラメータ

ID	svpid	TelnetサービスポイントID
INT	modname	設定する動作モードの種類
VP	modval	設定内容を格納した領域の先頭アドレス
TMO	tmout	タイムアウト値

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値 / エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (modnameの値が不正、modvalが0、または4の倍数以外、modvalの内容が不正、tmout < -1)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (サービスが開始していない、または設定できない状態)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除
E_CLS	接続切断
EV_PROT	サーバが要求を拒否

解 説

Telnetクライアントのサービス、または各Telnetサービスポイントの動作モードを変更します。サーバとTelnetオプション交渉が必要な場合、交渉が終了するまで待ち状態となります。

svpidには変更するTelnetサービスポイントIDを指定します。svpidに0を指定した場合は、Telnetクライアントのデフォルト設定を変更します。(デフォルト設定を変更した場合でも、すでに接続されているTelnetサービスポイントには影響しません。)

tmoutには、Telnetサーバからの応答待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間(単位はms)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。
無効なTelnetサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

modnameには、以下に示す値が指定できます。

(1) modname = TLN_MOD_SVP (0x00000001) (Telnetクライアント通信モードの設定)

Telnetサービスポイントの動作モードを指定します。デフォルト設定では、sockMSL, sockTIMEWAITはTCP/IPマネージャの構築情報としてtcpipConfTbl に設定したtcpMSL (最大セグメント寿命) を基準に、sockMSL = tcpMSL (秒)、sockTIMEWAIT = tcpMSL × 2 (秒) となっています。

svpidに0を指定した場合、以降に作成されるTelnetサービスポイントのデフォルト値として設定します。(すでに生成されているTelnetサービスポイントには影響しません。)

svpidにTelnetサービスポイントIDを指定した場合、そのTelnetサービスポイントがアイドル状態の場合のみ設定できます。

modvalに指定するパケットの構造 :

```

typedef struct{
    H                sockMSL;           最大セグメント生存時間タイマ(秒)
    H                sockTIMEWAIT;     TIMEWAIT(時間待機)状態タイマ(秒)
}T_TLN_CEPMODE

```

sockMSL, sockTIMEWAITに0を指定すると、以前に設定された値を使用します。
 ただし、sockMSLに0以外の値を指定し、sockTIMEWAITに0を指定すると、sockTIMEWAITにsockMSL
 ×2の値を設定します。

(2) modname = TLN_MOD_CBK (0x00000002) (コールバックイベントの設定)
 Telnetクライアントがコールバックにより通知するイベントの有効/無効を指定します。デフォルト設定
 では、cbkmode = 0 (すべて無効)となっています。
 コールバックイベントの設定はsvpidに0を指定した場合のみ設定可能です。(すでに接続されている
 Telnetサービスポイントには影響しません。)
 コールバックルーチンの詳細に関しては「4 コールバック」の項を参照してください。

modvalに指定するパケットの構造：

```

typedef struct{
    UW                cbkmode;         コールバック通知オプション
}T_TLN_MODCBK;

```

コールバック通知オプション(cbkmode)には、次の値を組み合わせ設定します。

ビット1：データ受信イベント

TLN_CBK_RCVOFF (H'00000000)	データ受信イベント通知無し
TLN_CBK_RCVON (H'00000001)	データ受信イベント通知有り

ビット2：送信バッファ空きイベント

TLN_CBK_SNDOFF (H'00000000)	送信バッファ空きイベント通知無し
TLN_CBK_SNDON (H'00000002)	送信バッファ空きイベント通知有り

ビット3：Telnetオプション変更イベント

TLN_CBK_OPTOFF (H'00000000)	Telnetオプション変更イベント通知無し
TLN_CBK_OPTON (H'00000004)	Telnetオプション変更イベント通知有り

ビット4：接続切断イベント

TLN_CBK_DISCOFF (H'00000000)	サーバ接続切断イベント通知無し
TLN_CBK_DISCON (H'00000008)	サーバ接続切断イベント通知有り

(3) modname = TLN_MOD_ASC (0x00000003) (クライアントテキスト形式の設定)
 クライアント側のテキスト形式を設定します。デフォルトではTLN_ASC_CRLFに設定されています。
 サーバから送られたデータは、クライアントのテキスト形式に自動変換します。また、サーバに送ら
 れるデータはネットワークの標準形式に自動変換します。ただし、バイナリモードが有効の場合は変換しま
 せん。
 svpidに0を指定した場合、以降に作成されるTelnetサービスポイントのデフォルト値として設定します。
 (すでに生成されているTelnetサービスポイントには影響しません。)
 svpidにTelnetサービスポイントIDを指定した場合、そのTelnetサービスポイントがアイドル状態の場合の
 み設定できます。

modvalに指定するパケットの構造：

```

typedef struct{
    UW                ascmode;         テキスト形式
}T_TLN_MODASC

```

テキスト形式(ascmode)には、次の値が指定できます。

TLN_ASC_CRLF (H'00000000)	改行文字はCR+LF
TLN_ASC_LF (H'00000010)	改行文字はLF
TLN_ASC_CR (H'00000020)	改行文字はCR

(4) modname = TLN_MOD_TERM (0x00000004) (ローカル端末情報の設定)

ローカル端末の各種情報を設定します。

svpidに0を指定した場合、以降に作成されるTelnetサービスポイントのデフォルト値として設定します。

(すでに作成されているTelnetサービスポイントには影響しません。)

modvalに指定するパケットの構造 :

```
typedef struct{
    UH          cols;          ウインドウサイズ幅
    UH          rows;         ウインドウサイズ行
    UW          tspeed;       端末送信スピード
    UW          rspeed;       端末受信スピード
    UB          *ttype;       端末タイプ情報 ( 最大40文字 )
}T_TLN_MODTERM
```

ウインドウサイズ幅 (cols)、ウインドウサイズ行 (rows) にはローカル端末の画面サイズを設定します。

端末送信スピード (tspeed)、端末受信スピード (rspeed) にはローカル端末の速度を指定します。

端末タイプ情報 (ttype) には、端末タイプ文字列 (最大40文字まで有効) の先頭アドレスを設定します。

初期状態では、端末タイプ情報が“vt100”、その他の項目は未設定となっています。未設定の状態の場合、サーバへの値の通知は行いません。また、0が設定された項目や、現在の設定値と同じ値が設定された項目に関しても、サーバへの通知は行いません。

新しい値が設定された場合は、それぞれ対応するTelnetオプションによりサーバへ通知されます。ただし、端末スピード、端末タイプ情報に関してはサーバから送信要求があった場合のみ、通知します。

(5) modname = TLN_MOD_BINARY (0x00000005) (バイナリモードの設定)

入力・出力に対するバイナリモードの有効 / 無効を設定します。8ビットデータを送受信する場合に使用します。新しいモードが設定された場合、TELNET BINARYオプションによりサーバに通知されます。指定したTelnetサービスポイントが接続状態の場合、Telnetサーバからの応答メッセージを待つため、必ず待ち状態になります。そのため、ポーリング指定では処理を行わず、直ちにエラーコードとしてE_TMOUTを返します。

modvalに指定するパケットの構造 :

```
typedef struct{
    UW          binmode;      バイナリ設定モード
}T_TLN_MODBIN
```

バイナリ設定モード(binmode)には、次の値を設定します。

TLN_BIN_DISABLE (H'00000000)	入力・出力ともバイナリモードを無効にする。
TLN_BIN_INPUT (H'00000001)	入力のバイナリモードを有効にする。
TLN_BIN_OUTPUT (H'00000002)	出力のバイナリモードを有効にする。
TLN_BIN_INOUT (H'00000003)	入力・出力ともバイナリモードを有効にする。

(6) modname = TLN_MOD_LM (0x00000006) (ラインモードの設定)

ラインモードの有効 / 無効を設定します。デフォルトではキャラクタ単位モードに設定されています。

新しいモードが設定された場合、TELNET LINEMODEオプションによりサーバに通知されます。

指定したTelnetサービスポイントが接続状態の場合、Telnetサーバからの応答メッセージを待つため、必ず待ち状態になります。そのため、ポーリング指定では処理を行わず、直ちにエラーコードとしてE_TMOUTを返します。

modvalに指定するパケットの構造 :

```
typedef struct{
    UW          lmmode;      設定モード
}T_TLN_MODLM
```

設定モード (Immode)には、次の値を組み合わせて指定できます。

TLN_LM_CHAR (H'00000000)	ラインモードを無効にします。(キャラクタ単位モードに移行します。)
TLN_LM_LINE (H'00000100)	リアルラインモードを有効にします。サーバがリアルラインモードをサポートしていなければ、擬似ラインモード、またはキャラクタ単位モードに移行します。
TLN_LM_EDIT (H'00000001)	このビットがセットされる場合、クライアント側にて全ての編集処理を行い、完全な行のみ送ることをサーバに通知します。(この場合、編集処理はユーザが行う必要があります。)このビットがセットされない場合、クライアント側は出力を何も処理せずにそのまま送ることをサーバに通知します。(この場合、編集処理はサーバ側が行います。)リアルラインモードの場合のみ有効です。
TLN_LM_TSIG (H'00000002)	このビットがセットされる場合、IP、BRK、ATY、ABORT、EOF、SUSP等の割り込み/シグナルを、Telnetコマンドに変換してサーバに送ります。このビットがセットされない場合、割り込み/シグナルを通常のアスキー値にてサーバに送ります。リアルラインモードの場合のみ有効です。
TLN_LM_SOFTTAB (H'00000008)	このビットがセットされる場合、クライアントは水平タブコード (ASCII : 0x09) を次の水平タブストップまでの適当なスペースに変換して送ることをサーバに通知します。(この場合、変換処理はユーザが行う必要があります。)このビットがセットされない場合、クライアントは水平タブコードを処理しないで送ることをサーバに通知します。リアルラインモードの場合のみ有効です。
TLN_LM_LITECHO (H'00000010)	このビットがセットされる場合、ユーザがタイプした印刷不可文字をローカルエコーする場合は、そのままの文字をエコーすることをサーバに通知します。このビットがセットされない場合は、クライアント側が任意の方法で文字をローカルエコーすることをサーバに通知します。(例えば、ASCII 0x01は「^A」をエコー等) 注：ローカルエコーは、Telnetクライアント側のユーザプログラムで行う必要があります。本Telnetクライアントでは行いません。

(7) modname = TLN_MOD_SLC (0x00000007) (リアルラインモードでの特殊文字の設定)

リアルラインモードで使用される特殊文字を設定します。特殊文字は、Telnetコマンドや行編集文字にマップされるコードです。新しいコードが設定された場合、TELNET LINEMODEオプションによりサーバに通知されます。

指定したTelnetサービスポイントが接続状態で、リアルラインモードが有効の場合、Telnetサーバからの応答メッセージを待つため、必ず待ち状態になります。そのため、ポーリング指定では処理を行わず、直ちにエラーコードとしてE_TMOUを返します。

modvalに指定するパケットの構造 :

```
typedef struct{
    UW          slcmode;      特殊文字設定モード
    UB          slc[30];      設定する特殊文字の値 ( slcmodeに
                                TLN_SLC_EXPORTを指定した場合のみ有効 )
}T_TLN_MODSLC;
```

特殊文字設定モード(slcmode)には、次の値を設定します。

TLN_SLC_EXPORT (H'00000001)	特殊文字をslc[30]に指定した値に設定する。
TLN_SLC_IMPORT (H'00000002)	特殊文字をリモートのデフォルト値に設定する。

特殊文字 (slc[30]) には、端末の各特殊文字のコードを指定します。0を指定した場合はその特殊文字は機能しません。slcmodeにTLN_SLC_EXPORTを指定した場合のみ有効です。特殊文字には以下の種類があります。

特殊文字一覧

	名 称	デフォルト値	説 明
1	synch	0x00	未読の入力データを破棄させる
2	brk	0x00	ブレーク
3	ip	0x03	プロセス割り込み
4	ao	0x0f	出力データのフラッシュ
5	ayt	0x14	相手存在確認
6	eor	0x00	エンド・オブ・レコード
7	abort	0x1c	プロセス中断
8	eof	0x04	エンド・オブ・ファイル
9	susp	0x1a	プロセスの一時停止
10	ec	0x7f	文字消去。入力ストリームから1文字を消去する
11	el	0x15	行消去。現在の入力行全体を消去する
12	ew	0x17	単語消去。入力ストリームから1単語を消去する
13	rp	0x12	行の再印字
14	lnext	0x16	次の文字は処理せず、そのまま送信する
15	xon	0x11	出力再開。ユーザ端末への出力を再開する
16	xoff	0x13	出力一時停止。ユーザ端末への出力を停止する
17	forw1	0x00	1行中の送信開始位置
18	forw2	0x00	1行中の送信終端位置
19	mcl	0x00	1文字左にカーソルを移動
20	mcr	0x00	1文字右にカーソルを移動
21	mcwl	0x00	1単語左にカーソル移動
22	mcwr	0x00	1単語右にカーソル移動
23	mcbol	0x00	行の開始にカーソルを移動
24	mceol	0x00	行の終わりにカーソルを移動
25	insrt	0x00	挿入モード
26	over	0x00	上書きモード
27	ecr	0x00	右側キャラクタの消去
28	ewr	0x00	右側単語の消去
29	ebol	0x00	行の開始まで消去
30	eeol	0x00	行の終端まで消去

3.2.10 TLN_get_mod Telnet クライアントの動作モードを取得する

【 T 】

C 言語インタフェース

```
ER_UINT modlen = TLN_get_mod ( ID svpid , INT modname , VP modval);
```

パラメータ

ID	svpid	TelnetサービスポイントID
INT	modname	取得する動作モードの種類
VP	modval	動作モードを格納する領域の先頭アドレス

リターンパラメータ

ER_UINT	modlen	リターン値またはエラーコード
---------	--------	----------------

リターン値 / エラーコード

正の値	正常終了 (取得した動作モードのデータ長)
E_PAR	パラメータエラー (modnameが不正、modvalが0、または4の倍数以外)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (サービスが開始していない、または取得できない状態)

解 説

Telnetサービスポイントの現在の動作モードを取得します。
現在の動作モードを取得できた場合、リターン値として取得した動作モードのデータ長を返します。

svpidには取得するTelnetサービスポイントIDを指定します。
指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。
無効なTelnetサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

modnameには、以下に示す値が指定できます。

- (1) modname = TLN_MOD_BINARY (0x00000005) (バイナリモードの状態取得)
現在のバイナリモードの状態を取得します。
指定したTelnetサービスポイントが接続状態の場合のみ取得できます。

modvalに指定するパケットの構造 :

```
typedef struct{
    UW          binmode;          バイナリ設定モード
}T_TLN_MODBIN
```

バイナリ設定モード(binmode)には、以下の値を返します。

TLN_BIN_DISABLE (H'00000000)	入力・出力ともバイナリモードが無効
TLN_BIN_INPUT (H'00000001)	入力のバイナリモードが有効。
TLN_BIN_OUTPUT (H'00000002)	出力のバイナリモードが有効。
TLN_BIN_INOUT (H'00000003)	入力・出力ともバイナリモードが有効。

- (2) modname = TLN_MOD_LM (0x00000006) (ラインモードの状態取得)
現在のラインモードの状態を取得します。
指定したTelnetサービスポイントが接続状態の場合のみ取得できます。

modvalに指定するパケットの構造 :

```
typedef struct{
    UW          lmmode;          設定モード
}T_TLN_MODLIM
```

設定モード(lmmode)には、現在のラインモードの設定モードが返されます。詳細は「3.2.9 TLN_set_mod Telnetクライアントの動作モードを設定する」項の「(6) ラインモードの設定」を参照してください。

(3) modname = TLN_MOD_SLC (0x00000007) (リアルラインモードでの特殊文字の取得)
 リアルラインモードでの現在の特殊文字を取得します。
 指定したTelnetサービスポイントが接続状態で、リアルラインモードが有効の場合のみ取得できます。

modvalに指定するパケットの構造 :

```
typedef struct{
    UW          slcmode;          特殊文字設定モード
    UB          slc[30];          特殊文字
}T_TLN_MODSLC;
```

特殊文字設定モード(slcmode)には、次の値が返されます。

TLN_SLC_EXPORT (H'00000001)	特殊文字はローカルの値を使用。
TLN_SLC_IMPORT (H'00000002)	特殊文字はリモートのデフォルト値を使用。

特殊文字 (slc[30]) には、現在の各特殊文字のコードが返されます。詳細は「3.2.9 TLN_set_mod Telnetクライアントの動作モードを設定する」項の「(7) リアルラインモードでの特殊文字の設定」を参照してください。

(4) optname = TLN_MOD_OPT (0x00000008) (Telnetオプションの取得)
 現在のTelnetオプションの状態を取得します。
 指定したTelnetサービスポイントが接続状態の場合のみ取得できます。

以下に示すパケットの各Telnetオプションには、次の値が返されます。

- 0 : サーバ/クライアント共にオプションが無効
- 1 : サーバ側のみオプションが有効
- 2 : クライアント側のみオプションが有効
- 3 : サーバ/クライアント共にオプションが有効

modvalに指定するパケットの構造

```
typedef struct{
    UB          neg_options[40];
}T_TLN_MODALOPT;
```

neg_options[0]	TRANSMIT-BINARY (バイナリ転送)
neg_options[1]	ECHO (エコーバック)
neg_options[2]	RCP (再接続) ¹
neg_options[3]	SUPPRESS-GA (進行抑止)
neg_options[4]	NAMS (メッセージ長の調停) ¹
neg_options[5]	STATUS (Telnetオプション状態通知)
neg_options[6]	TIMING-MARK (タイミングマーク)
neg_options[7]	RCTE (送信&エコーのリモート制御) ¹
neg_options[8]	NAOL (行幅出力の調停) ¹
neg_options[9]	NAOP (ページサイズ出力の調停) ¹
neg_options[10]	NAOCRD (キャリッジリターン出力の調停) ¹
neg_options[11]	NAOHTS (水平タブストップ出力の調停) ¹
neg_options[12]	NAOHTD (水平タブキャラクタ処理の調停) ¹
neg_options[13]	NAOFFD (フォームフィード出力処理の調停) ¹
neg_options[14]	NAOVTS (垂直タブストップ出力の調停) ¹
neg_options[15]	NAOVTD (垂直タブ出力処理の調停) ¹
neg_options[16]	NAOLFD (ラインフィード出力の調停) ¹
neg_options[17]	EXTEND-ASCII (拡張ASCII) ¹
neg_options[18]	LOGOUT (強制ログアウト)
neg_options[19]	BM (バイトマクロ) ¹
neg_options[20]	DET (データエントリ端末) ¹

neg_options[21]	SUPDUP (SUPDUPディスプレイプロトコル) ¹
neg_options[22]	SUPDUP-OUTPUT (SUPDUP出力) ¹
neg_options[23]	SEND-LOCATION (ロケーションID送信) ¹
neg_options[24]	TERMINAL-TYPE (ターミナルタイプ)
neg_options[25]	END-OF-RECORD (エンドオブレコード) ¹
neg_options[26]	TUID (TACACSユーザ識別) ¹
neg_options[27]	OUTMRK (マーク出力) ¹
neg_options[28]	TTYLOC (端末ロケーション番号) ¹
neg_options[29]	3270-REGIME (TELNET3270様式) ¹
neg_options[30]	X3-PAD (X.3 PAD) ¹
neg_options[31]	NAWS (ウィンドウサイズの調停)
neg_options[32]	TERMINAL-SPEED (端末スピード)
neg_options[33]	TOGGLE-FLOW-CONTROL (フロー制御)
neg_options[34]	LINEMODE (リアルラインモード)
neg_options[35]	XDISPLOC (Xディスプレイロケーション) ¹
neg_options[36]	ENVIRON (環境変数) ¹
neg_options[37]	AUTHENTICATION (認証) ¹
neg_options[38]	ENCRYPTION (暗号化) ¹
neg_options[39]	NEW-ENVIRON (新環境変数) ¹

(1) : 本バージョンでは未サポート

クライアントがリアルラインモードにあるかを判断するには、neg_options[34] (LINEMODE) を参照してください。LINEMODEが有効であればリアルラインモードにあります。

クライアントが擬似ラインモードにあるか、キャラクタ単位モードにあるかを判断するには、neg_options[1] (ECHO) およびneg_options[3] (SUPPRESS-GA) を参照してください。サーバ側のECHO、SUPPRESS-GAがともに有効であればキャラクタ単位モードにあります。サーバ側のどちらかが無効であれば擬似ラインモードにあります。

オプション値と現在モードの対応表

オプション値 現在のモード	neg_options[1] (ECHO)				neg_options[3] (SUPPRESS-GA)			
	0	1	2	3	0	1	2	3
キャラクタ単位モード								
キャラクタ単位モード								
キャラクタ単位モード								
キャラクタ単位モード								
擬似ラインモード								
擬似ラインモード								
擬似ラインモード								
擬似ラインモード								
擬似ラインモード								
擬似ラインモード								
擬似ラインモード								
擬似ラインモード								

3.2.11 TLN_snd_cmd Telnet コマンドを送信する

【 T 】

C 言語インタフェース

```
ER ercd = TLN_snd_cmd ( ID svpid , UB cmd , TMO tmout );
```

パラメータ

ID	svpid	TelnetサービスポイントID
UB	cmd	送信するTelnetコマンド名称
TMO	tmout	タイムアウト値

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値 / エラーコード

E_OK	正常終了
E_NOSPT	未サポート
E_PAR	パラメータエラー (tmout < -1)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (サービスが開始していない、または接続状態でない)
E_TMOUT	ポーリング失敗またはタイムアウト
E_CLS	接続切断

解 説

指定されたTelnetコマンドをサーバに送信します。

svpidには対象となるにTelnetサービスポイントIDを指定します。

cmdには送信するTelnetコマンド名称を指定します。以下に示すコマンドが指定できます。

Telnetコマンド一覧

名 称	Telnetコマンド	説 明
TLCMD_EOF(236)	TELNET EOF	ファイル終端
TLCMD_SUSP(237)	TELNET SUSP	プロセス停止
TLCMD_ABORT(238)	TELNET ABORT	プロセス中断
TLCMD_EOR(239)	TELNET EOR	レコード終端
TLCMD_NOP(241)	TELNET NOP	何もしない
TLCMD_DM(242)	TELNET DM	データマーク
TLCMD_BRK(243)	TELNET BREAK	ブレーク
TLCMD_IP(244)	TELNET IP	プロセス割り込み
TLCMD_AO(245)	TELNET AO	出力中止
TLCMD_AYT(246)	TELNET AYT	相手存在確認
TLCMD_EC(247)	TELNET EC	文字を削除
TLCMD_EL(248)	TELNET EL	行を削除
TLCMD_GA(249)	TELNET GA	進行

未サポートのTelnetコマンド名称を指定した場合は、エラーコードとしてE_NOSPTを返します。

接続状態以外のサービスポイントに対して本サービスコールを発行した場合、エラーコードとしてE_OBJを返します。

3.2.12 TLN_isu_opt 任意の Telnet オプションを送信する

【 T 】

C 言語インタフェース

```
ER ercd = TLN_isu_opt ( ID svpid , INT optname , VP optval , TMO tmout );
```

パラメータ

ID	svpid	TelnetサービスポイントID
INT	optname	送信するTelnetオプション番号
VP	optval	送信内容を格納した領域の先頭アドレス
TMO	tmout	タイムアウト値

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値 / エラーコード

E_OK	正常終了
E_NOSPT	未サポート
E_PAR	パラメータエラー (optvalが4の倍数以外、optnameが0~255以外、tmout < -1)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (サービスが開始していない、または接続状態でない、またはすでに送信済みか送信する必要がない)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除
E_CLS	接続切断
EV_PROT	サーバが要求を拒否

解 説

指定されたTelnetオプションの要求をサーバに送信します。

svpidには対象となるにTelnetサービスポイントIDを指定します。

optnameには送信するTelnetオプション番号 (0 ~ 255) を指定します。

optvalに指定するパケットの構造 :

```
typedef struct {
    UW          flag;
} T_TLN_ISUOPT;
```

flagには送信するTelnetオプションの要求を指定します。

- 0 (WILL) : 「送信元 (クライアント側) のオプションを有効にしたい事」について、受信先 (サーバ側) に同意を求めます。
- 1 (DO) : 送信元 (クライアント側) が、「受信先 (サーバ側) のオプションを有効にしたい」という事を要求します。
- 2 (WONT) : 「送信元 (クライアント側) のオプションを無効にしたい」事について、受信先 (サーバ側) に同意を求めます (受信先は同意しなければなりません)。
- 3 (DONT) : 送信元 (クライアント側) が、「受信先 (サーバ側) のオプションを無効にしたい」という事を要求します (受信先は同意しなければなりません)。

のみ指定できます。

tmoutにはサーバからの応答待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間 (単位は ms)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。ただし、本サービスコールはサーバからの応答を待つため、必ず待ち状態になります。そのため、ポーリング指定では処理を行わず、直ちにエラーコードとしてE_TMOUTを返します。

送信したTelnetオプションの要求に対して、それをサーバが承諾した場合、リターンコードとしてE_OKを返します。サーバが要求に対してそれを拒否した場合、エラーコードとしてEV_PROTを返します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

未サポートのTelnetオプション番号を指定した場合は、エラーコードとしてE_NOSPTを返します。

接続状態以外のサービスポイントに対して本サービスコールを発行した場合、またはすでに指定した状態となっていてTelnetオプションを送信する必要がない場合、エラーコードとしてE_OBJを返します。

主なTelnetオプションを次に示します。

主なTelnetオプション一覧

オプション番号	説明
TLOPT_BINARY (0)	通常の7ビットデータではなく、8ビットバイナリとしてデータを受信する
TLOPT_ECHO (1)	エコーバックを行う
TLOPT_SGA (3)	送受信を切り替えるGO AHEADコマンドの送信を抑制する
TLOPT_STATUS (5)	Telnetオプション状態を送るように要求する
TLOPT_TM (6)	コネクションの双方の同期を取る際に使用される
TLOPT_LFLOW (33)	フロー制御を行う

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なTelnetサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

未サポートのTelnetオプション番号を指定した場合は、エラーコードとしてE_NOSPTを返します。

TCPコネクションが切断された場合は、エラーコードとしてE_CLSを返します。

3.2.13 TLN_can_svp ペンディングしている処理をキャンセルする

【 T 】

C 言語インタフェース

```
ER ercd = TLN_can_svp(ID svpid, FN fnccd);
```

パラメータ

ID	svpid	TelnetサービスポイントID
FN	fnccd	キャンセルするサービスコールの機能コード

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値 / エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (fnccdが指定可能な機能コードでない)
E_ID	不正ID番号 (TelnetサービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (svpidのTelnetサービスポイントIDが存在しない)
E_OBJ	オブジェクト状態エラー (指定したTelnetサービスポイントIDにfnccdで指定した処理がペンディングしていない)

解説

svpidで示されたTelnetサービスポイントに対し、fnccdで示されたペンディング中の処理をキャンセルします。

キャンセルされたタスクには、エラーコードとしてE_RLWAIを返します。

キャンセル可能な処理のサービスコール名と、それを指定する機能コードは次の通りです。また、TFN_TLN_ALL(0)を指定すると、指定したTelnetサービスポイントにペンディングしているすべての処理をキャンセルすることができます。

サービスコール名	機能コード
TLN_con_srv	TFN_TLN_CON_SRV (-0x1506)
TLN_sht_srv	TFN_TLN_SHT_SRV (-0x1507)
TLN_put_dat	TFN_TLN_PUT_DAT(-0x1508)
TLN_get_dat	TFN_TLN_GET_DAT(-0x1509)
TLN_set_mod	TFN_TLN_SET_MOD(-0x150A)
TLN_isu_opt	TFN_TLN_ISU_OPT(-0x150C)
TLN_snd_cmd	TFN_TLN_SND_CMD(-0x150D)
すべて	TFN_TLN_ALL(0)

4. コールバック

コールバックは、Telnetクライアント内で発生した事象をアプリケーションプログラムに伝えるために用いられます。コールバックルーチンはユーザが作成し、そのアドレスのみがTelnetクライアントに渡されるため、モジュール名はユーザ任意となります。ここでは仮称として*callback*とします。

4.1 コールバックルーチン

4.1.1 *callback* Telnet クライアントイベントを通知する

【T】

C 言語インタフェース

```
ER ercd = callback ( ID svpid , UW evcd , VP parblk );
```

パラメータ

ID	svpid;	TelnetサービスポイントID
UW	evcd;	コールバックイベント番号
VP	parblk;	パラメータを格納した領域の先頭アドレス

リターンパラメータ

ER	ercd	リターン値またはエラーコード (未使用)
----	------	----------------------

解 説

Telnetクライアントのイベントが発生した場合にコールされます。コールバックルーチンはTelnetクライアントタスクレベルにてサブルーチンコールされます。ただし、TLN_cre_svpにてコールバックルーチンのアドレスに0を指定した場合、およびTLN_set_optにてコールバックイベントの設定を行っていない場合はコールされません。

コールバックルーチン内にてTelnetクライアントのサービスコールは発行できません。コールバックルーチン内にてTelnetクライアントのサービスコールを発行した場合の動作は保証しません。

svpidにはイベントが発生したTelnetサービスポイントIDが渡されます。
コールバックイベントには以下の種類があります。

コールバックイベント一覧

イベント種類	イベント名称	イベント番号	説明
システム イベント	TEV_TLN_DISCONN	0x1581	サーバの接続切断
	TEV_TLN_RCVDAT	0x1582	データの受信
	TEV_TLN_SNDEMP	0x1583	送信バッファ空き
Telnetオプション 変更イベント	TEV_TLN_BINARY	0x1500	バイナリモードの変更を受信
	TEV_TLN_ECHO	0x1501	エコーバックの変更を受信
	TEV_TLN_LFLOW	0x1521	リモートフロー制御の変更を受信
	TEV_TLN_LINEMODE	0x1522	ラインモードの変更を受信

(1) TEV_TLN_DISCONN Telnet クライアントの接続切断イベント

コールバックイベント番号 (evcd)

TEV_TLN_DISCONN (0x1581)

パラメータブロックの構造 (parblk)

無し

解 説

サーバからTelnetクライアントが切断されたことを通知します。
svpidには切断されたTelnetサービスポイントIDが渡されます。

このイベントでのparblkは無効のため、0を渡します。

(2) TEV_TLN_RCVDAT データの受信イベント

コールバックイベント番号 (evcd)

TEV_TLN_RCVDAT (0x1582)

パラメータブロックの構造 (parblk)

```
typedef struct{
    INT      rtncd;          受信データの長さ
} T_TLDPB_RCVDAT;
```

解 説

クライアントからのデータを受信したことを通知します。
svpidには受信したTelnetサービスポイントIDが渡されます。
本イベント通知は、TLN_set_modにてコールバックイベント通知の設定を行った場合のみ発生します。
また、TLN_rcv_datがペンディング状態の場合は発生しません。

コールバックルーチン内にてTLN_rcv_datをコールし、受信データを取得することも可能です。

(3) TEV_TLN_SNDEMP 送信バッファの空きイベント

コールバックイベント番号 (evcd)

TEV_TLN_SNDEMP (0x1583)

パラメータブロックの構造 (parblk)

```
typedef struct{
    INT      rtncd;          空き送信バッファのサイズ
} T_TLDPB_SNDEMP;
```

解 説

送信バッファに空きが発生したことを通知します。

svpidには空きが発生したTelnetサービスポイントIDが渡されます。

本イベント通知は、TLN_set_modにてコールバックイベント通知の設定を行った場合のみ発生します。
また、TLN_snd_datがペンディング状態の場合は発生しません。

コールバックルーチン内にてTLN_snd_datをコールし、データを送信することも可能です。

(4) TEV_TLN_BINARY バイナリモード変更イベント

コールバックイベント番号 (evcd)

TEV_TLN_BINARY (0x1500)

パラメータブロックの構造 (parblk)

```
typedef struct{
    UW      flag;          オプション要求フラグ
} T_TLNPB_OPT;
```

解 説

サーバから、BINARYオプションの要求を受け付けて、モードが変更されたことを通知します。
svpidには受信したTelnetサービスポイントIDが渡されます。
parblk->flagには次の値が渡されます。

- 0 (WILL) : サーバ側がバイナリデータを送信することを受け付けた。
- 1 (DO) : クライアント側がバイナリデータを送信することを受け付けた。
- 2 (WONT) : サーバ側がアスキーキャラクタを受信することを受け付けた。
- 3 (DONT) : クライアント側がアスキーキャラクタを受信することを受け付けた。

本イベント通知は、TLN_set_modにてコールバックイベント通知の設定を行った場合のみ発生します。

(5) TEV_TLN_ECHO エコーバック変更イベント

コールバックイベント番号 (evcd)

TEV_TLN_ECHO (0x1501)

パラメータブロックの構造 (parblk)

```
typedef struct{
    UW      flag;          オプション要求フラグ
} T_TLNPB_OPT;
```

解 説

サーバから、ECHOオプションの要求を受け付けて、モードが変更されたことを通知します。
svpidには受信したTelnetサービスポイントIDが渡されます。
parblk->flagには次の値が渡されます。

- 0 (WILL) : サーバ側がエコーバックすることを受け付けた。
- 1 (DO) : クライアント側がエコーバックすることを受け付けた。
- 2 (WONT) : サーバ側がエコーバックしないことを受け付けた。
- 3 (DONT) : クライアント側がエコーバックしないことを受け付けた。

本イベント通知は、TLN_set_modにてコールバックイベント通知の設定を行った場合のみ発生します。

(6) TEV_TLN_LFLOW リモートフロー制御の変更イベント

コールバックイベント番号 (evcd)

TEV_TLN_LFLOW (0x1521)

パラメータブロックの構造 (parblk)

```
typedef struct{
    UW      flag;          オプション要求フラグ
} T_TLNPB_OPT;
```

解 説

サーバから、LFLOWオプションの要求を受け付けて、モードが変更されたことを通知します。
svpidには受信したTelnetサービスポイントIDが渡されます。
parblk->flagには次の値が渡されます。

- 0 (OFF) : フロー制御の無効を受け付けた。
- 1 (ON) : フロー制御の有効を受け付けた。
- 2 (RESTART-ANY) : フロー制御を有効にし、任意のキャラクタ (XOFF以外) にて送信を再開することを受け付けた。
- 3 (RESTART-XON) : フロー制御を有効にし、XONキャラクタにて送信を再開することを受け付けた。

本イベント通知は、TLN_set_modにてコールバックイベント通知の設定を行った場合のみ発生します。

(7) TEV_TLN_LINEMODE ラインモードの変更イベント

コールバックイベント番号 (evcd)

TEV_TLN_LINEMODE (0x1522)

パラメータブロックの構造 (parblk)

```
typedef struct{
    UW      flag;           オプション要求フラグ
} T_TLNPB_OPT;
```

解 説

サーバから、LINEMODEオプションのモード設定要求を受け付けて、モードが変更されたことを通知します。

svpidには受信したTelnetサービスポイントIDが渡されます。

parblk->flagには次の値が渡されます。

- 1 (DO) : クライアント側がラインモードを有効にすることを受け付けた。
- 3 (DONT) : クライアント側がラインモードを無効にすることを受け付けた。

ラインモードの状態取得に関しては、「3.2.10 TLN_get_mod Telnetクライアントの動作モードを取得する」の項の「(2)ラインモードの状態取得」を参照して下さい。

本イベント通知は、TLN_set_modにてコールバックイベント通知の設定を行った場合のみ発生します。

5. サービスコール エラーコード一覧

表5-1 Telnetサービスコール エラーコード一覧

エラーコード	説明
E_SYS	システムエラー
E_RSFN	拡張サービスコールが登録されていない
E_PAR	パラメータエラー
E_ID	不正ID番号
E_NOMEM	メモリ不足
E_NOID	ID番号不足
E_OBJ	オブジェクト状態不正
E_NOEXS	オブジェクト未生成
E_RLWAI	処理のキャンセル、待ち状態の強制解除
E_TMOUT	ポーリング失敗、またはタイムアウト
E_CLS	接続できない、または接続が切断された
E_BOVR	バッファオーバーフロー
E_ILUSE	不正なサービスコール呼び出し
EV_PROT	プロトコルエラー

HI.CommunicationEngine
Telnetクライアント リファレンスマニュアル
CE7000TLN02J-4

発行年月 2006年 3月 第4版
発行 株式会社ルネサス北日本セミコンダクタ
編集 株式会社ルネサス北日本セミコンダクタ

©株式会社ルネサス北日本セミコンダクタ 2006