

Title	端末エミュレータ
Author(s)	片山, 正昭
Citation	大阪大学大型計算機センターニュース. 76 P.37-P.44
Issue Date	1990-02
Text Version	publisher
URL	http://hdl.handle.net/11094/65865
DOI	
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/repo/ouka/all/>

端末エミュレータ

大阪大学大型計算機センター 研究開発部

片山 正昭

katayama@center.osaka-u.ac.jp

1 はじめに

端末エミュレータとは、計算機の専用端末機で実現している機能をパソコンのソフトウェアでエミュレート（模倣）する¹ものです。例えば、本特集で紹介されている `h t e r m` は、DEC社の端末機VT-220の機能をパソコン上で実現します。また、ETOS52Gエミュレータ²とPC9800シリーズパソコンの組合せは、センターの第一端末室にあるACOS専用端末N6500と同等の能力を発揮します。

また、端末エミュレータは、単にパソコンを専用端末として使う機能以外にも、多くの便利な機能を持っています。例えば、パソコン上のエディタなどで編集したプログラムをセンターの計算機に送ったり、逆にセンターでの計算結果を手元のフロッピーディスクに保存したりすることも可能です。また漢字入力時に、パソコン上の仮名漢字変換機能、例えばATOKや松茸、VJEなどを利用することもできます。

具体的な端末エミュレータの使い方は、本特集の別稿に譲り、本稿は端末エミュレータを使ってセンターに接続する際に利用者が設定すべき項目（パラメータ）を中心に解説します。なお本特集の「資料」に、無料で入手できるエミュレータソフトウェアが紹介されています。併せてご覧下さい。

2 パラメータ設定

端末エミュレータを実際に使うためには、まず利用者がいくつかのパラメータ項目を設定する必要があります。そこで、それらのパラメータの意味と、具体的にどう設定すれば良いかを解説します。ちょっとむづかしい話は脚注に回しました。興味の無い方は、無視して頂いても差し支えありません。

¹もっとも、特定の専用端末機のエミュレートではなく、パソコンを端末として利用する為のソフトウェアも、広い意味では端末エミュレータと呼ばれます。端末ソフト、通信ソフトなどと呼ばれる場合もあります。これらはすべて、同じものと考えて良いでしょう。

²センターでも提供しています。

2.1 回線速度

データレートなどとも呼ばれます。単位はbps³[ビーピーエス:bit per second]、回線上を流れるデータの伝送速度です。普通の端末エミュレータでは、300、1200、2400、4800、9600などから選ぶようになっています。300だと、センターからの出力される文字が1つずつ眼で追える程度です。2400だとちょっと眼では追いつけないとおもいます。スクリーンエディタを使う場合、9600ならばまったく問題ありませんが、それより遅いと、カーソルの動きが鈍くなったような感じがするはずで、1200以下で、スクリーンエディタを使うには忍耐が必要でしょう。

使い勝手の面からは、回線速度は速いにこしたことはありません。しかし、

- 高速の伝送が可能なモデムは高価である。(本特集「資料」参照)
- 通信速度が速いほど、文字化け⁴が起きやすい。

という制約があります。つまり回線速度は、使っているモデムと回線の品質で決ってくるわけです。センターの交換回線は、各種速度の電話番号を準備していますので、端末エミュレータ、モデムの速度に合わせたものをお使い下さい。

なお、普通は端末とモデム間のデータ通信速度は、回線速度と同じです。しかしながら、最近のモデム(MNPなど)では、端末とのデータ通信速度が、回線上でのデータの伝送速度と異なっているものもあります。モデムの説明書をよく御覧下さい。

一般に、受信したデータの速度が、端末エミュレータで設定しているものと違う場合は、たとえば、キャリッジリターンを打つ度に、訳のわからない文字列が出力されてしまいます。

2.2 データビット長

1文字を何ビットで表わすかということです。キャラクタ長ともいいます。センターでは、ACOS、ワークステーション共に、

データビット長=7ビット

です。

³俗にボーといいますが、厳密には誤ったいいかたです。

⁴端末やホスト計算機から送られた文字が、ホストや端末についたときに雑音の影響で別の文字になってしまう現象。

2.3 パリティビット

実際にデータが通信回線上で伝送されるときは、上のデータビット長だけのビットの他に、余分なビットをつけて送られる場合があります。たとえばセンターでは、データビットの7ビットの後ろに更に1ビットをつけて送受信するようにしています。そして、この1ビットによって、これら8ビット中で1の値をとるものの数が、常に偶数となるように調整しています⁵。この余分な1ビットのことをパリティビットと呼びます。つまりセンターとの通信のためには、

パリティビット有り。 偶数(even)パリティ

と設定する必要があります。

なお、本センターに限らず

データビット長=7ビット→ 偶数(even)パリティ

データビット長=8ビット→ パリティビットなし。

とするのが普通⁶です。

2.4 ストップビット長

データの1かたまり（1文字分）の終了を表わすために、パリティビットの後ろに挿入されるビット（常に1）です。歴史的な理由で、この長さが2とか1.5などと設定できる端末エミュレータもありますが、本センターの場合も、それ以外の場合も、常に

ストップビット長=1

と設定する⁷とあって大丈夫です。

2.5 スタートビット長

データの1かたまり（1文字分）の開始を表わすために、データビットの前に挿入されるビット（常に0）です。常に

スタートビット長=1

と設定する⁸とあって大丈夫です。多くの端末エミュレータでは1に固定されています。

⁵これによって、1文字あたり奇数個の伝送誤りを検出できます。

⁶調歩同期式の場合。同期式でデータビット長が7ビットなら奇数（odd）パリティとするのが普通です。

⁷調歩同期式の場合のみ。同期式なら不要

⁸調歩同期式の場合のみ。同期式なら不要

2.6 フロー制御

例えば、ACOSで大きなファイルのLISTを実行した場合、大量の文字が一気に出力されて、端末パソコンでは、受信と表示が間に合わなくなる場合があります。このような時に、「ちょっと待って」「もういいよ」ということを端末から指示するのがフロー制御です。逆に端末からのデータ転送にホストがついて行けないときにも使われます。フロー制御のやりかたにも色々ありますが、センターのACOS/ワークステーションは、XON/OFFというやり方を使っています。これはCTRL-Sを「ちょっと待って」の意味で、CTRL-Qを「もういいよ」の意味で使う制御のやり方です。つまりセンターでは、

フロー制御 有り (XON/OFF制御)

2.7 ローカルエコー

端末のキーボードから入力した文字は、センターに送られて行きます。このとき、その文字を端末のスクリーンにも同時表示するかどうかの設定です。ワークステーションでは、端末から送られた文字は、そのまま折返し(エコーバック)して端末に送信されます。従って端末は、受信した文字だけをスクリーンに表示するように設定しておけば大丈夫です。つまり

ワークステーションではローカルエコー不要(エコーバックあり)

です。

一方ACOSは、このような文字のエコーバックはしていません。端末が自分で、送信した文字をスクリーンにも出す必要があります。つまり、

ACOSではローカルエコー要(エコーバックなし)

です。

なお、ポートセクタもACOS同様エコーバックをしていませんので、ワークステーション用に設定された端末では、ENTER CLASSに対して応答(5とか6とか)しても画面には表示されません。でも、ちゃんと送信されているはずですのでご心配なく。その証拠に、GOというメッセージが出力され、さらにキャリッジリターンを入力すると、

```
*** Osaka univ. ccsun01-ccsun05 , ccews01-ccews05 , ccews08-ccews10 ***
telnet>
```

が出力されます。ここからは、エコーバックが有りますから、打った文字はスクリーンに表示されます。

なお「ローカルエコー不要」にすべき時に間違えて「要」とした場合(たとえばACOS用設定でワークステーションに接続した場合)は、端末で打った文字が、2つずつスクリーンに表示されます。

2.8 文字コード

文字を通信回線に送出する際のコード体系は、J I Sに準拠すべき⁹です。本センターでもJ I Sに準拠したコードを使っています。もっともJ I S準拠というだけでは、コード体系を一意に決定することはできません。そこでここでは一般に広く使われている、J I Sのサブセット¹⁰について解説します。

J I S規格 (JIS X0208, ex C6226) は1つの漢字 (俗に全角文字) を、2バイトで表わす事を定めています。一方、普通のローマ字は、1文字1バイトです。そこで通信ではこのふたつの文字体系をなんらかの方法で切り替える必要が生じます。この切り替えの為の特別なマークがエスケープシーケンスと呼ばれるものです。

たとえば、「abc あ de」という文字列を送った場合、

```
a b c ESC $ B $ " ESC ( J d e
~~~~~          ~~~~~
```

いう文字列¹¹が回線上を送られます。ここで、「ESC \$ B」と「ESC (J」がエスケープシーケンスと呼ばれるもので、それぞれ「この後ろは2バイトで1文字、83年発行の漢字表をみてね」、「この後ろは1バイトで1文字、J I Sのローマ文字表ですよ」との意味を持ちます。もちろん「\$ "」は、J I S漢字表によれば「あ」を意味します。

さて、日本語に関係のあるエスケープシーケンスを以下にまとめます。

ア. ESC \$ @ :

2バイトを1文字として、1978年設定の漢字表 (JIS6226-78) を使うと指示。

イ. ESC \$ B :

2バイトを1文字として、1983年設定の漢字表 (JIS6226-83) を使うと指示。

ウ. ESC (J :

1バイトを1文字として、J I Sのローマ文字表を使うと指示。

エ. ESC (B :

1バイトを1文字として、ASCII¹²のローマ文字表を使うと指示。

⁹しかし商用BBSなどで、本来パソコンの内部コードであるシフトJ I Sを、そのまま通信用コードに使っている例も散見されます。

¹⁰アナウンサESC SPC @によって、G0集合の呼び出しが固定されている、そして合意に基づきアナウンサは省略されていると理解することができます。

¹¹データビットは7ビットしかないはずですが..... 大丈夫、回線上を伝送されるものは16進表示で00から7F、つまり7ビットで表わせる範囲内です。

¹²J I Sローマ文字とはほぼ同じ、違いは、J I Sの「〒」が「\」になっていることなど

オ. ESC (H :

正しくは、「1バイトを1文字として、スウェーデン名前用文字表¹³を使うと指示。」ですが、歴史的な理由で、ウとの混同・誤用が広く行なわれています。

なおアとイを漢字インと呼び、ウ、エ、オを漢字アウトと呼ぶ¹⁴こともあります。また特にアとエの組み合わせを旧J I Sと呼び、イとエの組み合わせを新J I Sと呼ぶ場合があります。しかし、アとオの組合せを旧J I Sと呼ぶ場合もあります。このように、現状では、エスケープシーケンスの呼称や意味の理解に関しては混乱が見られます。また市販のものを含めていくつかの端末エミュレータは、一部のエスケープシーケンスしかサポートできません。従って現状では、残念ながら、ケースバイケースで設定を行なわざるを得ません。

2.8.1 ACOSの場合

ACOS本体では、ACOS専用のコード体系を使っています。そして、端末との通信に際してコード変換をおこなっています。この際、どのコード体系に変換するかは、センターに接続するためのコマンドによって以下のように定まります。

\$\$\$CON,TSS,,NJS :

この場合は、イとウつまり、

ESC \$ B : 漢字の初め
ESC (J : ローマ字の初め

の組み合わせを端末が利用しているものとして、コード変換が行われます。1990年以降にセンターからお渡ししているASTERでは、予めそのように設定されています¹⁵。

\$\$\$CON,TSS,,KNJ :

この場合は、アとオつまり、

ESC \$ @ : 漢字の初め
ESC (H : ローマ字の初め

の組み合わせを端末が利用しているものとして、コード変換が行われます。1989年末までにセンターから配布したASTERもそのように設定してあります。

¹³ J I S ローマ文字とはほぼ同じ。違いは、J I Sの「J」が「A」になっていることなど

¹⁴ 漢字イン/アウトというのは厳密には間違い。少なくとも漢字アウトではなく、ローマ字インというべき。これはシフトイン/アウトとの混同による誤りでしょう。

¹⁵ 変更可能です。

2.8.2 ワークステーションの場合

sunのOSは、漢字を単なるデータとして扱います。コンソールで漢字を表示するためには、ウィンドー上のktermやwtermを利用する必要があります。そして、これらの上では、ア～オの全てのエスケープシーケンスが使えます。(但しローマ字は総てASCIIで表示されます。) もっとも、junetでメールを送るときには、オの「ESC (H)」の使用は避けましょう。これはJUNETでの約束事です。

一方、EWS4800では、OSレベルで漢字を扱うことができます。従って、ACOSの場合と同様に、内部コードと端末用コードの変換が必要になります。この変換は実際にはOSがterm変数で設定された端末に合わせて自動的に行います。現在、hterm, pc-vt, ddy880がサポートされています。具体的には、

h t e r m , p c - v t :

この場合は、イとウつまり、

ESC \$ B : 漢字の初め
ESC (J : ローマ字の初め

の組み合わせを端末が利用しているものとして、コード変換が行われます。

d d y 8 8 0 :

この場合は、アとオつまり、

ESC \$ @ : 漢字の初め
ESC (H : ローマ字の初め

の組み合わせを端末が利用しているものとして、コード変換が行われます。

なお、junet, ktermやwtermを利用するときは、sunのときと同様です。

2.9 送信行末処理

キャリッジリターンキーを押したときに回線に送出されるコードの指定です。ACOSでもワークステーションでも、

復帰コード (CR) のみを送出

するように設定して下さい。

2.10 受信行未処理

復帰コードや改行コードを受信した時の処理です。

復帰コード (CR) と改行コード (LF) の両方を受信したとき改行するよう設定して下さい。

3 おわりに

以上、駆け足で、端末エミュレータとその設定について解説しました。本来なら、ここに書かれたようなことは知らなくてもセンターを利用できるべきなのですが、現状ではそれはまだ難しいようです。計算機と特にその通信回りの技術が、まだまだ未成熟であるということでしょう。ということは、逆に今後どんどん便利になって行くことが期待できるともいえます。センターでは、これからも、本体から端末までを1つのシステムととらえて、使いやすい環境を提供するよう努力して行きます。しかし使いやすいシステムの実現は、利用者の方々の建設的な意見無しには不可能です。なにとぞ今後も、センターに御要望とそして暖かい御支援をお寄せ下さるようお願いいたします。