

計算機システムの環境整備への パーソナルコンピュータの応用

本 田 道 夫

I はじめに

パーソナルコンピュータ（以後パソコンと呼ぶ）は、その初期の段階から、一つの計算機システムとして使用するだけでなく、別の計算機システム（以後ホスト計算機と呼ぶ）のターミナルとしても使用することが考えられてきた。それに、計算機システムとはいいい難いようなもの、例えば機能の少ないエディタ及びアセンブラ程度の機能を ROM に備えたワンボード・マイクロプロセッサシステムのターミナルとしても用いることも可能であった。このようなこともあり、現在では、ほとんどのパソコン・メーカーが基本ソフトシステムとして採用している BASIC システムの中にターミナルシミュレーションのための機能が備えられてきている¹⁾ また一方では、MS-DOS, CP/M-86 等のオペレーティングシステム（以後 OS と呼ぶ）上で動作するターミナルシミュレータのソフトが計算機関連の雑誌にも掲載されているし²⁾ ソフトハウスからも同様のものが多数販売されている。

しかし、これらのソフトは、ターミナルとして用いるのに必要最小限の機能しか備えていないものであるか、特定の有名なターミナルの完全なシミュレーション機能のみを実現したものであるか、あるいは、それらにファイル転送の機能を追加した程度のものがほとんどである。つまり、パソコンとホスト計算機のデータのやりとりのための機能は備えてはいるものの、従来のターミナルの機能の域を出ていないものが多いのが現状である。

一方、計算機システムにおける使用環境、及びソフト開発環境を見た場合、

最近では Unix のような、ある程度環境が整備され、かつ整備のための新たな機能追加も比較的容易におこなえる OS も出現してきているが、一般の計算機システムでは、まだまだ充分な環境が整っておらず、また環境整備のための機能追加を図ろうとしても、目的とする機能によっては、ホスト計算機の OS 及びその上の既存のソフトウェアという枠組の中では、不可能または簡単には実現できない場合がある。

このような現状に対し、本論文では、パソコン及びホスト計算機を一つのシステムとして捉え、パソコン側に単なるターミナルシミュレーション機能だけでなく、ホスト計算機の枠組の中だけでは実現困難であるが、ある程度のメモリ、ディスクドライブ等を備えたパソコンをターミナルとして用いるが故に容易に実現できる機能を盛り込むことにより、ホスト計算機の使用環境、ソフト開発環境の整備を図ることを提案し、そのような目的で作成されたソフトウェア (ACTIVE と命名 … Aircorp Constructed Total Interface in Various Environments) について述べる。

尚、ACTIVE は、現在、日本電気製パソコン PC-9801 の上で本学の計算センターの三菱電機製 COSMO-700S、及び管理科学科の東芝製 UX-300、ザイログ社製 System-8000 をホスト計算機として有用に機能している。また、このようなソフトウェアは、ホスト側(というよりはパソコンに対する相手側)が、ワンボード・マイクロプロセッサシステム等の場合にも、そのソフトウェア開発において非常に威力を発揮することが確認されている。ACTIVE の総ての機能についての使用法については、「ACTIVE (計算機システムの環境整備のためのターミナル・ソフトウェア) の使用法」として、別の機会に記載の予定である。

II ACTIVE の機能

ACTIVE は、以下の機能を備えている。但し、(1)、(3)及び(4)は、従来のターミナル・ソフトウェアに備えられている機能であり、パソコンをターミナルとして用いる場合必要最小限の機能というべきものであるが、一応 active の機

能として述べておく。(5)の Batch 機能, (6)の Photo機能が本論文で提案したい「パソコンを用いることにより容易に実現可能であり, ホスト計算機の使用環境, 及びソフト開発環境の整備を図る機能」として現在実現されているものである。

- (1) ターミナル・ソフトウェアとしての基本的機能である PC-9801 をホスト計算機のキャラクタ・ターミナルとして用いる機能。但し, ターミナルとして使用中, 一時的に使用を中断し, PC-9801 の OS に戻り, その OS 上で作業をしたのち, ホスト計算機で LOGON(login) し直すことなく, 再度ターミナルとして使用できるよう設計されている。
- (2) グラフィック・ターミナルとしての機能。
PC-9801 を 640×400 (あるいは 640×200) のグラフィック・ターミナルとして用いる機能。但し, 単なるドット表示機能だけでなく, ホスト側からのコマンドにより簡単に, 直線, 矩形, 円, 楕円等の図形の表示, 領域塗り潰し, 漢字, アルファベット等の文字の表示, 及びそれらのプリンターへのハードコピーが行えるようになっている。
- (3) PC-9801 と ホスト計算機との間のファイル転送機能。
- (4) ホスト計算機のターミナルとして使用中, ホスト計算機の出力の任意の部分を PC-9801 に接続したプリンターに出力するリスト機能 (List 機能)。
- (5) ホスト計算機のターミナルとして使用中, PC-9801 のディスクファイルに書かれたホスト計算機に対する一連のコマンドやデータ等を, あたかもキーボードからの入力のように, 順次ホスト計算機に送り実行させる機能 (Batch 機能)。
- (6) ホスト計算機のターミナルとして使用中, ホスト計算機の出力の任意の部分を PC-9801 のディスクファイルに記録する機能 (Photo 機能)。

(2)のグラフィック機能についてであるが, パソコンの 640×400 のドット数では, 1240×1240 程度のグラフィック画面を持つ本格的なグラフィック端末

のシミュレーションには能力不足であり、あまりグラフィック機能は考慮されていないようである。(但し、PC-9801のBASICシステムに用意されているターミナル機能の場合は、BASICのグラフィックと同程度の機能は用意されている。)従って、ターミナル・ソフトウェアのグラフィック機能といっても、シミュレーション対象のキャラクタ・ターミナルが有している特殊文字による疑似グラフィック機能を実現しているのが大部分である。しかし、パソコンのグラフィック関連ソフトウェアが640×400のグラフィック画面で実現されていることを考慮すれば、640×400のものでもある程度は実用になるであろうと思われ、インテリジェントなグラフィック機能を実現するため、各種図形の描画機能、領域塗り潰し機能、及びハードコピー機能等のためのコマンドを用意した。(尚、本ソフトウェアの大部分はPC-9801のBASICシステムのグラフィック命令処理の部分を利用している。)

以下、第III章でBatch機能、第IV章でPhoto機能について述べる。また、第V章では、上記(3)-(6)の機能をキーボード以外(ホスト計算機側から、あるいはBatch機能用ファイル中から)呼び出す機能について述べるが、これもパソコンであるが故に実現できた機能である。

III Batch 機能

Batch機能とは、ホスト計算機のターミナルとして使用中、PC-9801のディスクファイルに書かれたホスト計算機に対する一連のコマンドおよびデータを、あたかもキーボードからの入力のように、順次ホスト計算機に送り実行させる機能である。この機能は、以下のような利用法および動機に基づき作成した。

- (1) デバッグ段階では毎回同じコマンド及びデータを繰り返し入力することが多く冗長であるが、そのような場合のキーボードからの入力の手間をできるだけ少なくする。このことを、COSMO-700Sの場合を例にとり説明する。次のようなホスト計算機の使用を考える。
 - (a) ファイル PROG:F に作成しているフォートランのプログラムをコン

パイルしオブジェクト・ファイル PROG:R を作成する。

(b) それをリンクしてロードモジュールPROG:L を作成する。

(c) オブジェクト・ファイル PROG:R を削除する。

(d) PROG:L を実行する。このとき、"READ 文"に対するデータとして、
1, 2, 3を入力する。

この場合、キーボードからのコマンド入力は次のようになる。(下線の部分がユーザ入力の部分である。)

!EXTFORT PROG:F OVER PROG:R

><CR>

>END

!LYNX PROG:R OVER PROG:L;.EXTLIBE

!DELETE PROG:R

!S PROG:L

?1

?2

?3

(但し、"!)"は COSMO-700S の top-level のプロンプト、">"は FORTRAN コンパイラのプロンプト、"*****"は COSMO-700S の出力、"?)"は、プログラム中の "READ 文" に対するプロンプトである。また、<CR>は、キャリッジ・リターンのみを入力を示す。)

上記の入力は大した分量ではないが、しかし、この入力をデバッグ中に何度もキーボードから行うのは面倒である。それに、一般には、もっと入力データが多いものと思われる。そこで、このようにあらかじめ分かっている一連のコマンドを、これもまたあらかじめ分かっているデータに対して実行させる場合、キーボードから入力すると同様な次のような内容のファイルをパソコン上に作成しておき、以後はそのファイル名を指定するだけでその内容をあたかもキーボードからの入力のようにホスト計算機に送ることができればキーボードからの冗長なコマンド及びデータ入力を避けることができる。(このようなファイルを以後 batch-file と呼ぶ。)

```
!EXTFORT PROG:F OVER PROG:R
>
>END
!LYNX PROG:R OVER PROG:L;EXTLIBE
!DELETE PROG:R
!S PROG:L
?1
?2
?3
```

(行の最初の文字が、ホスト側のプロンプトになっており、キーボードから入力すべき文字列だけではないことに注意。これについての説明は第IV章 3 でおこなう。)

- (2) あらかじめ分かっているコマンド及びデータを、順次ホスト計算機に送るのであれば、それらのコマンド及びデータをファイルに作成しておき、あとの確認のために、Photo 機能、あるいは List 機能により記録をとりな

がら、そのファイルの内容を自動的に送ることにすれば、ユーザはその間、キーボード(パソコン)の前にいる必要はなくなるし、別の作業をすることも可能である。

- (3) ROM に簡単な(機能の少ない)エディタ、及びアセンブラ程度の機能を備えたワンボードマイクロコンピュータシステムのソフトウェア開発において次のように使用する。まず強力なエディタを用いてパソコン側でプログラムの作成・修正を行う。次に、ワンボードシステムでエディタを起動し入力状態にしておき、パソコンのディスク上のそのプログラムを Batch 機能を用いてワンボードマイコン側に送る。次にエディタを終了し、アセンブラを起動し、最後に実行する。

尚、初期の Batch 機能は上記仕様のものではあったが、例えば、(1)の場合のようにロードモジュールを作成することは、一つのプログラムだけでなく、複数のプログラムについて行なうのが普通であるがそのような場合、各プログラムの入ったファイル名に対し、上記のような、batch-file を個々に作成しておくのは種々の点から得策ではない。そこで、現在はプログラムの入っているファイル名を Batch 機能起動時にパラメータとして与えることにより、一つの batch-file により複数のプログラムに対するロードモジュール作成が可能であるように機能拡張がなされている。その場合、batch-file は次のようになる。(\$1 の部分に Batch 機能起動時にパラメータとして与えられる文字列の最初のものが代入されてホスト計算機へ送られる。)

```
!EXTFORT $1:F OVER $1:R
>
>END
!LYNX $1:R OVER $1:L;EXTLIBE
!DELETE $1:R
```

IV Photo 機能

Photo 機能とは、ホスト計算機のターミナルとして使用中、ホスト計算機の出力の任意の部分を PC-9801 のディスクファイルに記録する機能である。この機能は、以下のような利用法および動機に基づき作成した。

(1) ユーザとホスト計算機との会話をディスクファイルへ記録し、その記録を後で調べることにより、デバッグの効率化をはかる。この機能の代わりに、ACTIVE の機能のうち、List 機能によるプリンタ出力を用いてのデバッグも考えられるが、その方法では、結果的にはデバッグに本質的に必要な部分は少なくても、あらかじめその部分が予測できないため総ての出力をプリントすることになり、もし出力が非常に多い場合には、以下のような問題がある。

(a) ホスト計算機との会話の速度が、プリンタの速度により制限されるため会話における待ち時間が多くなる。

(b) (a)に述べたように、パソコン側での通信処理速度がプリンタの速度に制限されるため、パソコン側の受信バッファがオーバフローし、コントロールS/コントロールQによるフロー制御が可能でない場合は、ホスト側からの情報が失われることがある。

(c) プリンタ出力から、デバッグに必要な部分を探すのが大変である。

これに対し、記録をディスクファイルにとることにより、会話の速度は Photo 機能を用いないときとほとんど変わらず、(a)、(b)の問題はなくなる。但し、(b)については、ホスト側からの一連の出力がディスクシステムの一記録単位より長い場合、ディスクへの書き出しが起ころ情報失われる場合がある。しかし、この場合でも、ディスクとして、メモリディスクを用いれば、(b)の問題は解決する。(c)については、ディスクファイルに記録した場合、強力なエディタ等がパソコンに備わっていれば、容易にデバッグに必要な部分を探すことが可能であり、必要ならば、その部分だけ切り出してプリンタに

出力すればよい。

- (2) 学生等のためにホスト計算機の使用方法的資料を作成する場合、マニュアル、あるいは記憶を頼りに使用方法を記述するよりも、実際に使用しているそのままの記録をディスクファイルに記録し、それに対し、必要ならばパソコンのエディタを用いて注釈等を付けて作成する。この場合も、List 機能を用いてもよいように思われるかもしれないが、作成する使用方法の資料の量が多いときには、しばしば途中でミスをする可能性があること、注釈の記入の必要があること等を考慮すると、Photo 機能を用いた方がはるかに効率的である。
- (3) 次の例(3-1), (3-b) のように、ホスト計算機の出力をファイルに記録したものを修正し、Batch 機能を用いて、ホスト計算機へ以降の入力として送る。
- (3-a) 会話型でかつ頻繁にデータ入力を要求するたぐいのプログラムのデバッグの場合、同じデータに対し何度もテストランをすることが多い。この場合、もし最初から要求されるデータが分かっているならば、前章の(1)で述べたように、それをパソコンの batch-file に作成しておき、Batch 機能によりキーボードからの代わりにホスト計算機に送れば冗長な入力が避けられる。しかし、最初の方に入力するデータにより以後のプログラムの流れが左右され、従って入力要求の内容も異なってくるような複雑なプログラムの場合、要求される総ての入力をプログラムを実行することなくあらかじめ知ることは容易でないことが多い。このような場合、最初のデバッグではキーボードから入力し、その実行の様子を Photo 機能により、パソコンのディスクファイルにその記録をとり、パソコン上でそのファイルからデータ入力の部分のみ取り出し適当に修正しておけば、二回目以降の入力については、前章で述べた Batch 機能を用いればキーボードからの冗長なデータ入力を避けることができる。また、Batch 機能による入力は、人間によるキーボードからの入力よりも高速であるため、デバッグもより効率よくおこなえる。

(3-b) ホスト計算機の不要のファイルを削除するとき、削除対象ファイルが多い場合は、ファイル名一覧を表示させて、それを見ながら削除コマンドを入力するのが一般的である。しかし、ある程度以上の数のファイルを削除する場合はその入力はかなり量の量になる。そこで、ファイル名一覧を表示するとき Photo 機能によりパソコンのディスクファイルにその記録をとり、パソコン上でエディタによりそのファイルをもとに削除コマンドを並べたファイルを作成しておけば、前章で述べた Batch 機能を用いて簡単に削除が実行できる。

V キーボード以外からの ACTIVE の機能の呼び出し

ACTIVE では、第 II 章の(3)から(6)で列挙した機能を、ユーザが直接キーボードから起動するだけでなく、次の(1)、(2)に述べるように、batch-file 中、あるいはホスト計算機で実行されるプログラム中からも起動できる。これにより、より整備されたホスト計算機の使用環境、ソフトウェアの開発環境が実現できている。

(1) 例えば、ホスト計算機に強力なエディタがない場合、PC-9801で作成・修正したフォートランプログラムをホスト計算機に送り、コンパイル、リンクの後、実行する手続きがデバッグ期間中繰り返されるであろう。このような場合、第 III 章で述べたように、ホスト計算機に対するコマンドであるコンパイル、リンク、実行の部分だけを batch-file に用意するだけでなく、ACTIVE の機能の一つであるところの「プログラムをホスト側に転送する機能」の呼び出しも batch-file 中に用意できれば、ユーザのキーボード入力の手間はそれだけ省かれる。そのような目的のための batch-file は以下のようなものとなる。1 行目が、パソコンのファイル \$1.FOR をホスト計算機のファイル \$1:F に転送するファイル転送機能を呼び出す部分である。但し、"S" は、コントロール・アンダーラインと S であり、キーボード

から、ACTIVE のファイル転送機能呼び出すときのコマンドと同じものである。また \$1 は、第III章で述べた Batch 機能起動時にパラメータとして与えられる文字列の最初のもので代入されてホスト計算機へ送られるパラメータを指定するものである。

```

!/\_S $1.FOR $1:F
!EXTFORT $1:F OVER $1:R
>
>END
!LYNX $1:R OVER $1:L;EXTLIBE
!DELETE $1:R
!S $1:L
?1
?2
?3

```

(2) ホスト計算機の上で開発しているプログラムのデバッグのために、そのプログラムの出力が必要である場合、ACTIVE の機能のうち、Photo 機能あるいは List 機能を用いて記録をとればよい。しかし、出力が非常に多く、かつ必要な出力は最後のいくらかの部分である場合には、通常は、

- (a) 不要な出力を気にせずに、出力すべてについて記録をとる
- (b) 不要な出力を避けるため、プログラムを修正する

等が行われる。しかし、(b)の場合は、プログラムの正しい部分を変更することになり、新たな間違いを生じる可能性もある。そこで、プログラム中のデバッグのために必要とされる出力を行う部分（そのような部分は、本質的に修正が必要な部分に近いであろうと思われる）の直前に、ACTIVE に対し Photo 機能あるいは List 機能呼び出す出力を行う命令を付け加えておくことができれば、不要部分の出力を記録にとらなくてよくなる。

VI ACTIVE のインプリメンテーション

1. ACTIVE の処理手順

ACTIVE の基本的な処理手続きは次のようになっている。

- (a) ホスト計算機から文字が送られてきているか否かを調べる。送られてきていなければ(e)へ行く。送られてきていれば、その文字を受け取る。
- (b) その文字が ACTIVE の機能を呼び出すものであれば、そのための処理を行い、(e)へ行く。
- (c) Photo 機能実行中であれば、指定されたパソコンのファイルへその文字を書き込む。
- (d) List 機能実行中であれば、プリンタへその文字を印字する。
- (e) ホスト計算機から、入力停止のフロー制御の信号が送られてきている状態か否かを調べる。もしその状態であれば、(a)へ行く。
- (f) Batch 機能実行中か否かを調べる。実行中でなければ(j)へ行く。実行中であれば、batch-file から一文字読み出す。
- (g) その文字が batch-file の終わりを示すものであれば、Batch 機能実行中の状態を解除し、(a)へ行く。
- (h) その文字が ACTIVE の機能を呼び出すものであれば、そのための処理を行い、(a)へ行く。
- (i) その文字を、ホスト計算機へ送り、(a)へ行く。
- (j) キーボードから文字が入力されているか否かを調べる。
- (k) 入力されていなければ、(a)へ行く。入力されていれば、その文字を受け取る。
- (l) その文字が ACTIVE の機能を呼び出すものであれば、そのための処理を行なった後(a)へ行く。
- (m) その文字を、ホスト計算機へ送り、(a)へ行く。

尚、現在の ACTIVE プログラムは、通信速度 4800 ボーで PC-9801 を

ターミナルとして用い、かつホスト計算機のスクリーン・エディタを用いる都合上、処理速度を考慮しアセンブラ言語で作成している。但し、CPU が 8086 以外の機種への移植のことを考慮し、処理速度の関係からどうしてもアセンブラ言語によるプログラムの必要な部分以外は、可能な限り移植の容易な言語（例えば、C言語）で記述することを考えている。

2. PC-9801 の OS のモニタコールについて

通常プログラムは、アセンブラ言語で書く場合でも、移植性を考慮して可能なかぎり、OS (MS-DOS, CP/M-86 等) のモニタコールを利用すべきであるし、ACTIVE のインプリメンテーションでも極力そのように努力している。しかし、上記処理手順の(a)で、ホスト計算機から文字が送られてきているか否かを調べるときは、OS の標準のモニタコールだけでは不十分なため、PC-9801 に固有の拡張モニタコールを利用している。さらに、(e)でホスト計算機から XOFF の信号が送られてきている状態か否かを調べる機能は、OS のモニタコール及び PC-9801 の拡張モニタコールにも見当たらないため、OS 領域でそのような情報を管理している番地を直接参照している。また、7ビットでホスト計算機と通信する場合、シフトイン・シフトアウトの状態を調べるためにも、OS 領域を直接参照している。そのため、MS-DOS であっても、バージョンが異なれば参照番地も異なり、本質的には総てのバージョンに対しそれぞれ作成する必要がある。但し、現在の3種類のバージョンについては、偶然にも参照番地がRS-232C のハードウェア割り込みの番地（この番地は固定番地にある割り込みベクタを参照することにより決められる）から同じオフセットとなっているため、1種類のものですんでいる。このあたりは、是非モニタコールあるいは PC-9801 の拡張モニタコールとして、是非とも実現してもらいたい機能である。

3. Batch 機能の実現について

Batch 機能を実現するにあたって、生じた問題、及びその解決法について報告しておく。

- (1) Batch 機能を実現した最初のバージョンは、ユーザがキーボードから入力するものだけを batch-file に用意するようにしていたが、batch-file が大きい場合、ホスト計算機のバッファがいっぱいになり、送った文字が無視される場合が生じた。(ホスト計算機がコントロール S / コントロール Q によるフロー制御を行っていても、Batch 機能による文字送信の速度が速すぎて、送った文字が無視される場合が生じた。) そこで、ホスト計算機が入力を求める場合、プロンプトが出されることが多いことを利用して、batch-file 中の行の最初の文字がプロンプトとして指定されている文字であれば、そのプロンプト文字がホスト計算機から送られてきた後、その行の最初のプロンプト以外の文字がホスト計算機へ送られるように変更された。デフォルトのプロンプト文字としては "!", ">", "%", "#", "?" が指定されているが、ホスト計算機が異なれば、プロンプトの文字も異なる場合が多いこと、行の最初の文字として、どうしてもプロンプトであるような文字を送らざるを得ないことがある等の場合を考慮し、「Batch 機能に対する補助的設定機能」を用いて、ユーザがプロンプト文字を指定することができるよう配慮されている。
- (2) また、第 III 章の最後の部分で述べたような batch-file 中の形式パラメータを示す文字(デフォルトは、\$1, \$2, ... の '\$' である)についても、場合によっては、そのものをホスト計算機に送る必要が生じる。そこで、そのような場合、「Batch 機能に対する補助的設定機能」を用いて、文字 '\$' に代わるものをユーザが指定できるようにも配慮されている。
- (3) ホスト計算機のエディタによっては、入力行の処理のため次の行の入力

がしばらく待たされ、しかも次の行の入力のためのプロンプトを出さないものがある。そのようなホスト計算機のエディタを起動しておき、パソコンから Batch 機能によりファイルを送る場合、連続して何行かを送ると、文字が無視されることがある。このようなことを防ぐためには、一行送った後しばらく間隔をあけた後、次の行を送らなければならない。一方、このような間隔の最適値(送った文字が無視されず、しかも余計に待つことはない値)は、ホスト計算機、あるいはその上のソフトウェア(例えばエディタ)によりそれぞれ異なるのが普通である。そこで、場合に応じて、ユーザが、間隔を設定できるよう配慮されている。

このような問題は、Batch 機能を最初に実現したときは予想していなかったことであり、今後も Batch 機能に限らず、ホスト計算機及びその上のソフトウェアによっては、まだまだ生じることも予想される。

VII 最後に

実は、上記の Photo 機能は、DEC System-20 の OS (TOPS-20) 上での Photo 機能と同様なものであり、一方 Batch 機能は、Unix マシン上のコマンドファイルあるいは MS-DOS 上のバッチ機能の拡張であり、決して ACTIVE におけるオリジナルな着想ではない。それらの機能の有用性を実感していたため、本学計算センターの COSMO-700S の上で、それらの実現を試みたが、最初に述べたように、OS 及び既存のソフトウェアという枠組みの中では困難なことであった。しかし、パソコンをターミナルとして用いる場合、パソコン側にそれらの機能を持たせることは比較的容易に実現できたことが最初の出発点であった。

そして、第V章で述べたように、それらを含めた ACTIVE の機能を、ホスト計算機あるいは batch-file から呼び出せるようにすることによって、COSMO-700S (及び System-800, UX-300) 上でのソフトウェア開発は、従来に較

べかなり楽になり、効率もよくなった。また、MC-68000 を用いたワンボードマイコンに対するソフト開発においては、それらの機能の威力が十二分に発揮されている。さらに、現在はまだ実現していないが、パソコンのメモリの余裕を利用して、ホスト計算機からの出力を溜めておき、必要があればスクロールされてしまった画面でも巻戻して見ることができるような機能も考えている。もちろん、これらの機能以外にも、ホスト計算機の使用環境、及びソフト開発環境を整えるに役立つ機能は多々存在すると思われる。

ところで、最後に確認しておくが、本論文で提案したいことは、「個々の有用な機能」というよりは、むしろ、「ホスト計算機の枠組の中だけでは実現困難ではあるが、その使用環境、ソフト開発環境の改善を図るためには是非望ましい機能を、ターミナルとして用いるパソコン側で実現することを試みる」という方法である。そして、そのようなパソコン側で実現できる機能というものは、マン・マシン・インターフェースをつかさどり、ホスト計算機とユーザの間の入出力を加工する機能であるだけに、そのような機能を充実することは、ユーザの負担軽減に直接つながる場合が多いことが期待できるものと信じている。

参 考 文 献

- 1) 日本電気『PC-9801(E) User's Manus』
- 2) 早ノ瀬信彦, 小島一郎『MS-DOS 用ターミナル・プログラム』インターフェース No. 87, 1984,8