

# 初心者対象のパソコン指導

宮下 英明\*

## 1 ワーキングディスクの用意

指導に先立ち、指導者はパソコン学習用のワーキングディスク(システムディスク)を学生数分あるいは二人に一枚の割で作成しておく。パソコン指導は、このワーキングディスクの上で行なう。即ち、授業開始時にこのワーキングディスクを学生に貸与し、この上で演習を行なわせ、そして授業終了後にこれを回収するというやり方を取る。

## 2 パワー・オン

滑稽に聞こえるかも知れないが、パワーオンも一つの指導である。学生が知ったのはスイッチだけではない。より重要なことは、スイッチオン、スイッチオフが何の実害ももたらすものではないということ、学生が指導者の様子から了解したということである。(実際、一度起こってしまったことに対して何かをしなければ元に戻せないと考える方が自然ではないだろうか?)

パソコンの初心者とは、何よりもまず、パソコンに対して何をしても大丈夫かを知らない者のことである。パソコン指導を一つの側面から見ると、それは彼らの危惧、動揺を一つ一つ消していく指導というものになっている。

## 3 ブラインドキータッチの練習

パワーオンの指導のつぎにくるものは、キータッチの指導である。キータッチの指導は、言うまでもなく、ブラインドタッチの指導である。ブラインドタッチは、コマンドライン入力やプログラム作成の学習等を苦痛のないものにするために、絶対必要である。

ブラインドタッチの練習は、単純な反復練習である。せいぜい動機づけの工夫が入るくらいである。

MS-DOSのコマンドラインに入力するという形では、キータッチの練習は行なえない。実際、コマンドラインの制限文字数超過を警告するピーブ音が出たり、リターンするとエラーメッセージが出たりということで、落ち着いて練習ができない。手続きの意味には目をつぶらせて

```
TYPE CON > NUL
```

と入力させ、タイプライター・ライクな環境を作らせてから、練習に入らせるようにする。

はじめに、指のホームポジションと、それぞれの指のカバーする領域を知らせ、これに忠実に従うことを約束させる。ブラインドキータッチの指導ではここが一番肝心なところであって、(学生に対するこれまでの指導の経験から言

\* 宮下 英明 金沢大学教育学部

えることであるが)練習用のテキストに気を使う必要は全くない。

ここでは、専らアルファベット26文字のブラインドタッチを練習させる(他の記号は、必要に応じて追加して覚えていくというようにして十分である)。最初は、“ABC……XYZ”を何回も繰り返させる。この練習でできるようになるブラインドタッチでは、相等部分“ABC……XYZ”の順番が慣れになっている。そこでつぎに、ローマ字で“あいうえお……”の練習を繰り返し行なう(この場合、濁音、半濁音の除外は気にしなくてよい)。さらに“いろはに……”の練習をさせてもよい。そしてまた“ABC……XYZ”に返らせる。

以上のことを一回の授業時間を費やして行なえば、学生は十分にブラインドタッチを習得する。しかしこれを強化し続けることは必要で、各授業開始時に、一定時間のブラインドキータッチの練習を課すということもやるべきである。

#### 4 ファイルの概念

MS-DOSでは、ファイルの概念は入出力の対象一般へと拡張されている。これを実証的に示すために、リダイレクトの概念を初めから取り上げることにする。

具体的には、つぎのようなリダイレクトを用いた入出力の実験によって、キーボード、ディスプレイ、プリンタがディスク上のファイルと同等であることを示す：

1) DIR > \*\*\*

ここで\*\*\*には、CON, PRN, TESTが入る。

2) TYPE \*\*\* > \*\*\*

ここで\*\*\*には、CON, PRN, TESTのうちの任意の二つ(重複を含めて)が入る。

3) TYPE CON > ANSWER

[RET]

2 [RET]

N [RET]

[STOP]

FORMAT B : < ANSWER

なお2)では、>CONは省略できることを知らせる。

3)では、“FORMAT B :”が“FORMAT B : <CON”と同じであるということも、指摘しておく。

#### 5 エスケープシーケンス、アスキー制御コード

エスケープシーケンス、アスキー制御コードも、重要な指導内容としてここでは位置づける。実際、これを用いることによって、テキスト画面やプリンタの制御が簡単に行なえるようになる。

アスキー制御コードでは、07の“SOUND BELL”のみを取り上げる(他のコードはエスケープシーケンスでカバーできる)。実行プログラムの終了をBEEP音で知らせるというような応用が考えられるからである。

コード07は、^G(CTRLキーを押しながらGキーを押す)で入力される。そこで、BEEP音を発生させるには、

ECHO ^G

のようにすればよい。

同じことは、このコードで成るアスキーファイルを例えば、

ECHO ^G > BELL

のようにしてつくり、TYPE コマンドでこのファイルを出力する形でもできる：

TYPE BELL

あるいはまた、AUTOEXEC. BAT 内で環境変数にこのコードをセットしておけば——例えば、“SET BELL= ^G”——バッチプログラムにおいて

ECHO %BELL%

のようにしてこのコードを出力することができる。

エスケープシーケンスについては、これの一覧を示し、コマンドライン上でのエスケープコード ^[ の入力の仕方——ESC キーを押してから R キーを押す——を知らせる。つづいて、エスケープシーケンスを用いたテキスト画面制御を色々と実験させる。

例： ECHO ^[[7;36mABCDEFGG ^[[m (“ABCDEFGG” の文字列が水色にリバーズされて表示される)

プリンタに対する制御の方法も知らせる——これについては次節で述べる。

## 6 プリンタ

エスケープコードをプリンタに出力することによるプリンタセットアップを、ここで指導しておく。

一つは必須の知識として与える。それは、印字幅のセットである。

印字幅のセットは、エスケープコード(アスキーコード 1B)と、左マージンか右マージンかを示す文字(それぞれ、L と /)と、そして印字位置を示す 3 桁の数値をプリンタに出力すること

で実現される。用紙はどのサイズも中央にセットするというにすれば、最後の数値は例えばつぎのようになる：

	左マージン (L)	右マージン (/)
B 4 版	025	111
A 4 版	034	102
B 5 版	037	099

(但し、用紙を縦に使う場合)

プリンタセットの文字列をプリンタに出力するには、(エスケープコードを書ける)エディタを用いて、文字列 ^[L025, ^[/111 の 2 行で成るファイル——ファイル名を例えば B4TATE とする——を作っておき、

TYPE B4TATE>PRN

とすればよい。勿論、コマンドラインから直接

ECHO ^[L025>PRN

ECHO ^[/111> PRN

としてもよい。

同じことを BASIC 環境化するときには LPRINT 文を用いる。例えば、左マージン 25、右マージン 111 なら、

LPRINT CHR \$ (&H1B);”L”;”025”

LPRINT CHR \$ (&H1B);”/”;”111”

なお、初心者対象の場合、プリンタ用紙は、連続用紙よりも単票の方が使いやすい。BASIC プログラムにしても、バッチプログラムにしても、単票の単位でプリントアウトするのが——ファイルサイズの点からも保存の点からも——丁度よいし、融通が効く。

印字モードのセットも、知識として与えておいてよい。——^[N, ^[H, ^[Q, ^[E, ^[P をプリンタに出力することによって、印字モードがそれぞれ高速パイカ、高密度パイカ(標準)、コンデンス、エリート、プロポーショナル

にセットされる。

練習は、

TYPE CON>PRN

として、タイプライタ・ライクな状態で行なわせるとよいだろう。同一行中でもエスケープシーケンスをはさむことで文字種を何回でも変え得るといようなことを、確認させる。

## 7 フロッピーディスク

フロッピーディスクについては、その使用のみが問題である場合にも、初心者には結構色々なことを知っておく必要が出てくる。

まず、規格である。3.5インチ、5インチ、8インチといったサイズの違い、2D、2DD、2HDといった記録密度の違いがあることを、知らなければならない。

ディスクを物理的に壊さないために知らせておくべきことがある：磁性体であるから磁気から遠ざけるといったことは常識であるが、その他に、熱や直射日光を避けること、圧力をかけないこと、水に濡らしてしまったらもうダメなこと、ヘッドウィンドウを指で触れないこと、等々。

磁化は恒久的なものではなくホケる、したがって定期的なバックアップが必要であるということ。そして、事故はつねに起こるものと考えて、バックアップをとる習慣をつけておくといったこと、等。

フォーマットの知識が要る：FORMATを起動すること、/S オプション付きでこれを起動すること、また、このとき問われるディスクの記録容量(640KB/1MB)に答えられること。/S オプションを忘れてフォーマットしたディスクをシステムディスクにする方法、等々。

以上は、ディスクを自ら使っていくために最低必要となる知識である。

さて、ディスクに対する知識は、その扱い方の知識で十分か。

“ディスクは使えればよい；その構造がどうなっているかは実用上問題にはならない。”——確かにこのように言い得る場合はある。しかし、現実には多くの場合、構造までも知っていることが要求されるのである。

例えば、ディスクのフォーマットにしても、“どのようなときにフォーマットするのか(してはだめなのか、する必要がないのか)?”の問いに答える知識は、構造の知識になる。

また、通常の使用の上でのエラーメッセージ。——“何のことか?” “内容が壊れてしまったのか?” “どう対処したらいいのか?” “no system file”——“何だこれは?” “システムが転送できないといっている! 今度に限って何故できないのだ? コンピュータの故障か?”

簡単なことでもわかっていなければ、散々振り回されることになる。

学習者に対し、フロッピーディスクをブラックボックスのままにしていつまでも中途半端に不可解なものにしておくべきではない。

ディスクの仕組み(構造)の主題の下に、ここではディスクの物理的形狀——ヘッドウィンドウ、インデックスホール、セントラルホール、ライトプロテクトノッチ——の意味と、ディスクの構造(次節)の両方を、指導していくことにする。

## 8 フロッピーディスクの構造

〈フォーマットでつくられる構造〉以前の構造の概念としては、サーフィス、トラック(シリ

ング)がここで取り上げられる。フォーマットでつくられる構造の概念としては、セクタ(読み込み・書き込みの単位)とクラスタ(ファイルの記録の単位)の概念；IPL, FAT, ディレクトリ, データの4領域の概念が, 取り上げられる。

指導ではこれらの図式的な説明を行なうとともに, SYMDEBを使って実際にディスクの中身を覗いてみることを学生にさせる。

IPL, FAT, ディレクトリの意味を理解させる目的から, わざと書き換えもしてやることにするので, ワーキングディスクをコピーしたものをを使わせる。——実際このディスクは実験の中で壊される。

具体的にはつぎのようなことをする。

システムディスクとデータディスクのIPLを見比べる。

ディレクトリを覗く。併せてファイルの属性の概念を知らせる。属性を書き換えて, DIRコマンドによるファイルの表示がどのようになるかを確かめさせる。DELコマンドによるファイルの削除がどのような処理であるかを確かめさせ, このとき削除したファイルを復活させる。

FATの仕組みを知らせる。アスキーファイルに対してFATのそれに対応する部分を書き換え, これを短くするとか, 他のアスキーファイルとつなぐといったことをやらせてみる。

## 9 VRAM

メモリの主題の一環として, テキストVRAM, グラフィックVRAMを主題化する。

これについては, メモリ処理がそのまま画面に表示されるので, 指導効果は大きい。——実際, SYMDEB.EXE を用いてVRAMを操作

し, 手作業で文字や色模様の表示をするということを見せてみる。

BASIC言語でのPRINT文やLINE文は, この手作業をサブルーチン化したもの呼び出すコマンドに翻訳されて, 所期の表示をもたらす。初心者には, 手作業の延長としてプログラム中のコマンドを感得させ, プログラムを神秘的レベルから人のレベルに降ろしてやるのが大切である。

### 9.1 テキストVRAM

初めに, メモリとテキスト画面との対応関係を知らせておく:

	(0)	(79)
(0)	0000 0001	..... 009E 009F
(1)	00A0 00A1	..... 013E 013F
	⋮	⋮
(24)	0F00 0F01	..... 0F9E 0F9F

(数値は, セグメント値A000に対するオフセット値)

そして画面を一旦クリアしてからSYMDEBを起動する。(画面をクリアしてから起動するのは, 以下の操作で画面をスクロールさせないためである。)

先ず,

```
-D A000:0 L140
```

として, 画面とメモリの内容が対応していることを見させる。このとき併せて, 半角文字が《表示位置に対応するメモリにこの文字のアスキーコードと00が書き込まれていることで》表示されている, ということも押さえるわけである。

Eコマンドの形式を知らせて, 半角文字の表示を実際に行なわせる。それからつぎに, 漢字の表示の実践へと移る。

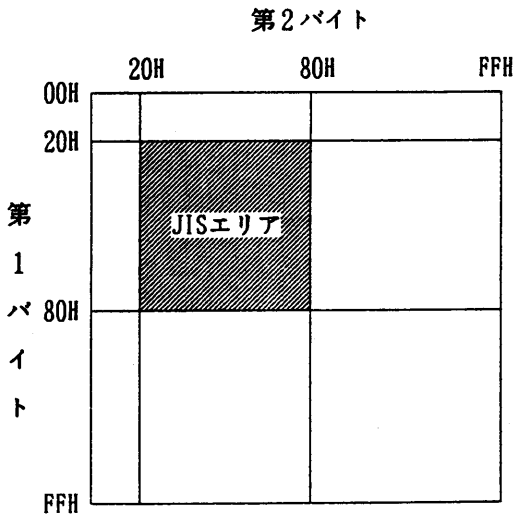
漢字の表示は, その漢字のJISコードに対

し

- (J I Sコードの第1バイト) - 20H,
- (J I Sコードの第2バイト),
- (J I Sコードの第1バイト) - 20H,
- (J I Sコードの第2バイト) + 80H

の4バイトを表示位置に対応するメモリに書き込むことで実現される。これを知らせるときには、書き込む4バイトの形式の理屈についても少し触れないわけにはいかないだろう。

漢字J I Sコードのエリアは



のようになっている。特に漢字J I Sコードの第2バイトは00にはならないので、上の形式の4バイトによる漢字の表示は、半角表示と交じることはない。そして4バイト目の +80H で、4バイトの一まとまりがどこからどこまでであるかがわかる。

つぎに、アトリビュートVRAMに属性を書き込むことをさせてみる。メモリとテキスト画面との対応関係は、セグメント値を除けばVRAMの場合(本節の最初の図)と同じである——アトリビュートVRAMではセグメント値が

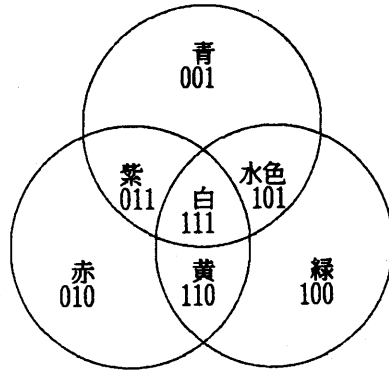
A200。

画面をスクロールさせないために、一旦MS-DOSに戻り画面をクリアしてから、再びSYMDEBを起動させる。属性の指定の形式:

ビット番号

	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	
ビット 値	1	G	R	B	垂線	アンダー ライン	リパス	フリック	ノーマル
	0				ノーマル	ノーマル	ノーマル	ノーマル	シークレット

と色の関係:



(数値は、GRBのビット値)

を知らせた上で、文字の表示位置に対応するアトリビュートVRAMのメモリに属性を書き込ませ、実際に所期の効果が文字の上に現われることを確かめさせる。

## 9.2 グラフィックVRAM

テキストVRAMの場合と同様、SYMDEBを用いたメモリ書き込みの方法によって、グラフィック画面表示とグラフィックVRAMとの対応を知らせることが、ここの目的である。

グラフィックモードはカラー640×400とし

て、この場合のメモリとグラフィック画面との対応関係：

	(0-7) (8-15)		(632-639)
(0)	0000	0001	000F
(1)	0050	0051	009F
	⋮	⋮	⋮
(399)	7CB0	7CB1	7CFE

(数値は、セグメント値 A800,B000,B800 のそれぞれに対するオフセット値)

を先ず知らせる。

直接MS-DOSからSYMDEBを起動するやり方では、画面表示の開始、400ライン表示モードの設定を、I/Oポートを介して制御データを出力するという形で行なう必要がある。そしてこの方法をとるときには、手続きの意味に触れないわけにはいかない。

これをしないで済ますために、ここでは(カラー640×400表示モードがデフォルトになっている)BASIC環境からチャイルドプロセスでSYMDEBを起動する方法をとる。このチャイルドプロセスでは、カラー640×400表示モードが保存されるからである。

SYMDEBを起動して行なわせることは、EコマンドやFコマンドを用いて、グラフィックVRAMにデータを書き込み、それに対応して表示がどのようになるかを確認することである。具体的には、ビットが画面上の点と対応すること；メモリのアドレスと表示色および画面上の位置の対応；青、赤、緑が互いに重なったときの色、等の確認である。

## 10 SYMDEB

ハードを理解させる(あるいはむしろ、ハード

に親しませる)ことの一環として、ディスクやメモリの中身を積極的に覗いてみるということここではやっている。そのためのツールとして、SYMDEBを早々に取り上げ、活用してきた。

この場合、SYMDEBの使用するサブコマンドは、D、E、F、L、Q、W程度であり、SYMDEBの使用そのものに学生が躓くことはない。但しこの場合、メモリの論理アドレス(セグメント・オフセットアドレス)の概念や、16進数、ビット、バイト等の概念を併行して指導していくことになる。

なお、SYMDEBではMS-DOSのプレート機能が使用でき、実際これの便利さが最もよく感得されるのは、SYMDEBの使用時であると思われる。プレート機能は何れにせよ指導しておくべき内容であるが、その最も指導しやすい場面として、このときの指導を位置づけておく。

## 11 ディスプレイ

ディスプレイ(CRT)は、テキストVRAMおよびグラフィックVRAMの内容を表示するだけの装置である。しかし初心者は、ディスプレイの中に画像のモトがあるというように錯覚しやすい。実際、ディスプレイのスイッチをオフにすると画像のモトがなくなってしまう。特に、一度描いた図は、一旦スイッチオフで消してしまうと初めから描き直さなければならないというように思ってしまう。

この錯覚は、ことばでそれと知らせて正すこともできるし、VRAMの説明によって理詰めでも正すこともできる。何れにしても、早々に済ませておくべきである。—実際、これにはディスプレイの“寿命”の問題が掛かっている。“焼

けつけ”の事実をしらせ、上手にスイッチオフを使うことを併せて指導しておく必要がある。

## 12 OSの概念

OSを直接主題化するかどうかは、パソコン指導の目的に依る。

例えば、パソコン指導の目的が或るソフト(ワープロ、リレーショナルデータベース等)を使いこなさせることであれば、ディスク処理を含めた全ての作業をメニュー選択ないし会話形式で進めることのできる一括処理プログラムを“システムディスク”の形態で提供することで、OSの主題化は必要でなくなる。しかしここでの指導のように、パソコンを汎用機として——即ち、用途がオープンなものとして——学習させることが主眼であるときには、OSの主題化は避けて通れない。

われわれの指導では、OSの主題化は図式的なものにとどめる。即ち、ハード、CPUの違いを吸収するものとしてのOSを図式的に説明する。アプリケーションソフトが、直接ハードではなく、OSとコンタクトする形をとる(例えばBIOSコールとかMS-DOSのファンクションコールの形で)ことによるメリットがここで理解されればよい。——それで十分であるという意味において、かつそれしかできないという意味において。

## 13 環 境

指導する側が用意作成したワーキングディスクは、学生にとっては全くのブラックボックスである。このブラックボックスの中身の解説の形をとって、さらにMS-DOSのファイル管理の概念、環境の概念、バッチファイルの概念

等の指導を行なう。

環境の概念では、具体的に、CONFIG.SYS ファイルの内容になる DEVICE, FILES, BUFFER, SHELL の語の意味と用法、そして環境変数の概念、特に PATH の語の概念と用法を知らせる。

但しこの段階では、FILES, SHELL, 環境変数一般についての説明は実効しない。何故なら、これに対応する用途は、まだ学生のものではないからである。

しかし、その他の DEVICE, BUFFER, PATH については、実証的に、これの価値を掴ませることができる。

## 14 バッチプログラム

バッチプログラムの作成も、重要な指導内容の一つである。この場合、外部コマンドを連ねる形のものよりは、内部コマンドだけでできるもの、さらにおもしろい画面効果をつくり出すものとか、ゲーム感覚のものの方がよい。軽快であり、作業も楽しいものになるということもあるが、つぎのような教育的効果があるからである。即ち、色々なテクニックが自分の問題として無理なく考えられるようになること、そして、限られた語法によっても豊かな内容のプログラムは可能であるということが感得されるようになること。

なお、勿論、以上の指導の一環として AUTOEXEC.BAT も取り上げられる。AUTOEXEC.BAT では、パスの設定も併せて指導する。

## 15 自習環境

何の学習についても言えることであるが、パ



ソコンの学習でも〈馴れ〉が大きなウェイトを占める。また、授業の限られた時間枠内では、取り上げる内容も自ずと限定される。したがって、不足を補いつつ、かつ個々の主題について習熟していくということが、学生の課外の実践としてどうしても要求されることになる。

このように、パソコン指導は、学生の自習環境という問題を閉却しては成立しない。そしてこの問題は、独り設備の問題にとどまるのではない。設備の管理・運営の問題が絡んでくる。パソコン指導の本当の難しさは、正にこの点に存していると言ってよい。パソコン指導は、授業で完結するものとして考えることはできないのである。

## 16 LAN

今日、一つの教室をパソコン演習室として設備する場合、パソコンをネットワークで連絡するところまで行なうようである。さらにこの場合、ネットワークは固定した目的を見て導入されているわけではない。パソコン指導が汎用機としてのパソコンの指導であるとき、ネットワークを目的に対して開いているものとして導入するのは当然であろう。

しかし、ネットワークの導入がこのようであるときには、“ためにする”問題として、“ネットワークの効果的な活用法”の問題が、指導者の側に無理な形で課せられることになる。

一斉指導の形の授業では、資源へのアクセスが同時になるために、待ち時間の問題が生じる。この場合には、ネットワークは快適な環境とは言いがたい。そのために、“後退”した形態をとることも一つの工夫になる。——例えば、ファイルをワークステーションの側に所持させる目的で

それを資源として供給する場合、ファイルサイズが大きいときには、アクセスさせるワークステーション数を限定する。そして、アクセスして得たファイルを残りの者にコピーするようにさせる。

筆者のネットワーク使用の実践は、ワーキングディスクをあらかじめ貸与した上での、サンプルプログラムの提供という形が専らであった。プログラム作成が授業の形態であったために、ネットワークの使用が自ずとこのようなものにとどまったわけであり、大きな問題も生じることがなかった。しかしこれに対し、ワープロ環境の共有、データベースの共有といった使い方を考えるときには、ソフトの対応の問題とか、アクセスタイムの問題とかで、課題は多く、かつ深刻になるであろう。

## 17 BASIC言語の指導

初心者へのパソコンの指導は、パソコンの上で一つの目的をもった仕事をするという形態で進められる。少なくとも、実効する指導の形態はそうである。そして一般的には、それはプログラム言語の指導になるであろう。さらに、採用される言語は、BASICに落ち着くであろう。

理由は、いくつかの観点から述べられる。

一つの観点は、BASICがインタープリタ言語であってコンパイラ言語ではないということである。

コンパイラ言語は、やはり概念自体が難しい（というよりむしろ、インタープリタ言語ではその仕組みを意識しないで済んでしまう）。実行ファイル完成までの段階の手続きが煩瑣である——このために、一括処理のバッチファイルを

組んだり、作業のメニューファイルを作ったりすることが、現実に必要な。また、作成途上のプログラムをとにかく実行させてみるということができない(文法上のエラーがあるうちは、実行プログラムに至れない; 実行プログラムにエラーがあるときには、悪くすると、暴走してしまう)。

また、コンパイラ言語では、プログラム開発の段階にもっていくまでに、インストール、環境設定の準備が要る。コンパイル、リンクの作業にRAMディスクは不可欠である—RAMディスク無しの作業(延々と続くディスクアクセス)は耐えられない。

プログラム作成の観点から言えば、BASICの起動がそのままプログラム開発環境の起動になっていることも、BASICを初心者に指導しやすい理由になる。特に、エディタを別途用意する必要がない。

変数の型や、配列、ストリングの扱いに神経を使わないで済むということも、初心者対象の言語指導にBASICが採用される大きな理由の一つになるであろう。

さらに、学習の動機づけという観点からすると、BASICでは豊富な機能が出来合いで用意されているということがある。特に、グラフィック機能が挙げられる。(例えばC言語では、グラフィックスはそれ用のライブラリを別途用意しなければならない。) 経験的事実として言えることであるが、グラフィックスは初心者にとって学習の一番の動機づけになる。

勿論、BASICに対しても、その欠点は様々に指摘されている。例えば、付属のエディタがラインエディタであり使いにくいという指摘。

しかしこのことに対しては、(やろうと思えば)次のように対処できる。即ち、二つの機能:

1) アスキーセーブされたBASICプログラムファイルから行番号を削除する;

2) アスキーファイルに行番号を付す  
を選択して実行できる実行プログラムを別途作っておく。これによって、BASICプログラムを市販のエディタによって行番号なしで書いていくことができる。勿論、バッチ処理として、これを一括処理にしておけばよい。—ただしこのときには、GOTO文等では行番号を用いずにラベルを用いる必要がある。しかし本来ラベルの使用が推奨されるべきなので、このことは少しも欠点にはならない。

また、BASIC言語が構造化されていないという指摘もある。しかしBASICでも、ラベルを用いたサブルーチンの活用で、構造化されたプログラムを書くとはできる。BASICで書けば必然的にスパゲティプログラムになってしまうというわけではない。スパゲティプログラムにしない工夫はあり、実際これは指導の内容になる。これに関連して、インデントを多用した見やすいプログラムを作ることも指導されるべきである。

ローカル変数の概念がないということも、BASICに対して指摘される。確かに不自由であるが、逆に初心者にはこの概念が躓きになるであろう。

再帰的処理ができないという指摘。しかし初心者対象の指導ではこのことは先ず問題にならない。

## 17.1 BASICエディタ

BASICエディタの約束、機能の或るもの

を知らないために、ひどく無駄な労力をつぎこんでしまうということはよくある。エディタの約束、機能は、理屈ではなく単に知っているか知らないかの問題であるから、機に応じて指導していく必要がある。

例えば、初心者は、カーソルが文末にある状態でリターンするのでなければ、その文は確定されないと考えやすい。カーソルは文の所在する行のどの位置にあってもよいということを知らせる必要がある。また、既存の文に対する書き足し、部分的削除が有効であること、特に、行番号を書き換えることでその文が複写されるということ。

また、ファンクションキーに登録されている機能。特に、最初の打ち込み環境としてのAUTOモード(AUTO[RET]で実現；行番号が自動的に付く)、エディット環境としてのエディットモード(EDIT . [RET] で実現；ROLL UP, ROLL DOWNでテキスト画面がスクロールする)。

CTRL+E, CTRL+F等の使い方も、指導の内容になる。

## 17.2 メインルーチン, サブルーチン

BASICプログラムを書くときの心掛けとして早々に指導すべきことの一つは、プログラムをメインルーチンとサブルーチンのセットの形に構造化するということである。

“メインルーチンは全てGOSUB文で成るようにする”と指導しても、決して極端ではない。短いプログラムでも、

```
*MAIN
```

```
GOSUB *SHOKIKA
```

```
GOSUB *JIKKO
```

```
GOTO *END.
```

```
*SHOKIKA
```

```
WIDTH 80,25: CONSOLE 0,25,0,1
```

```
:
```

```
:
```

```
*JIKKO
```

```
:
```

```
:
```

```
*END.
```

```
END
```

のように書くことを指導する。実際こうしておけば、このプログラムに限って言っても、後で機能を付け足したり、部分的に修正するのが易しいという利点がある。

## 17.3 ラベルの使用

“プログラム文の中には行番号を一切現わさない”というように心掛けさせる。

行番号の使用は、プログラムを著しく見渡しにくいものにしてしまう。また、それぞれ独立して作成したプログラムをマージして一つのプログラムにするということをやるときには、行番号の符合に神経を使わなければならない、気軽にRENUMコマンドを使用できない。さらに、MS-DOS上でプログラム文だけをエディタで作成し、そしてユーティリティプログラムを使って行番号を付して仕上げるという方法も採れない。

行番号を使わずにラベルを使う。

ラベルはGOSUB文で使われるときに、最も所を得た感じになる。このときには、ラベルはサブルーチン名のようにになっている。実際この

場合、われわれはサブルーチンのプログラム中の役割がわかるようなラベル名を工夫する。

但し、

```
FOR J=1 TO 10
```

```
:
```

```
IF X>A GOTO * NEXT.
```

```
:
```

```
*NEXT. : NEXT J
```

のようなラベルの使い方は、煩瑣に感じられるであろう。実際、このときにはラベル名を組織的につける方法にアタマを悩ますことになる。しかしそれでも、将来的に考えるならば、ラベルを使うべきである。しかも、ラベル名を考へたり、命名の規則を考へることで、プログラムが構造的に考へられることになり、プログラムの作成に計画性が加わることになる。

#### 17.4 変数の使用

変数の効果的な使い方も、主題になる。

“プログラム中に数字を生で現わすのは変数への代入のときだけ”というように指導しても、決して極端にはならない。変数をこのような形で使用することで、値の調整が代入式の一箇所済むようになる。あるいは、各ルーチンが汎用的なものになる。

#### 17.5 チャイルドプロセス

BASICを終了させずにMS-DOSに戻り、そこで或る操作をして再びBASICに戻るということが、チャイルドプロセスの形で実現できる。

このような手続を踏みたい場面は、実際よく起こる。例えば、いまロードしているファイルを、あるドライブのディスクのあるディレクト

りにセーブしたいときなど。あるいは、このような事情から、ディレクトリを確認、または作成したいときなど。

チャイルドプロセスは、BASIC環境化では

```
CHILD "コマンド名",M
```

の書式で行なえる。この場合コマンドが実行されると直ちにBASICに帰ってしまうが、

```
CHILD "COMMAND",M
```

として COMMAND.COMを起動すれば、EXITで帰るまでは、MS-DOS環境化に留ることができる。学生にはチャイルドプロセスのこの形式を特に教えておくとよいだろう。