

# パソコン上でのファイル変換

吉 村 卓

## File Conversion from B16-BASIC to N88-BASIC on PC-9801

Takashi YOSHIMURA

(平成元年10月31日受理)

### 1. 目 的

昭和48年度に導入された HITAC-8250 は15年間の永きにわたって本校の情報処理教育に貢献してきたが、この度ようやくその任を終え、平成2年3月を期して新システムに更新されることになった。機種選定の作業も一段落し、日本データゼネラル社製の ECLIPSE MV/2500DC 2 台から成るシステムが導入されることとなった。

このシステムには、RS-232C接続により、あるいは Ethernet LANを介し NEC製の PC-9801 シリーズ等が接続可能であるが、現在、本校計算機室に設置されている日立製の B16は接続できない。それ故 B16で開発されたソフトはそのままでは利用できないこととなる。そこで、B16-BASICのプログラムを N88-BASICのプログラムに変換しようというのが本稿のめざすところである。

### 2. 変換方法の概要

B16とPC-9801がRS-232Cケーブルで接続できる距離のところにあれば、アスキーセーブされた B16-BASIC ファイルを N88-BASIC に通信利用のファイル転送する方法が最も簡単であるが、この2機種がいつでも併置されているとは限らないので、ここでは、この方法は採らないことにする。

ところで、B16-BASIC は MS-DOS の管理下で動作し、データ転送は MD2HD で1セクタ 1024 バイト単位で行なわれる。一方、PC-9801 に標準装備の N88-BASIC は スタンドアローン・タイプで、ファイル転送は1セクタ 256 バイト単位でなされる。それ故、PC-9801 で B16 のディスクを直接読むことはできない。そのため、MS-DOS ファイルを N88-BASIC ファイルに転送するプログラムを用意しなければならない。

また、B16-BASIC も N88-BASIC もともに、Microsoft 社系であるが、非アスキー形式のファイルは機種により、その中間言語の命令コードが異なるので、変換作業を容易にするために、アスキー形式でセーブされた BASIC プログラムファイルの転送を扱うことにする。

なお、N88-BASIC にはディスク内のセクタを直接読み書きできる DSKI\$ 関数と DSKO\$ という便利なコマンドが用意されているが、これらの命令は1セクタ 256 バイトのディスクフォーマットでないと使用できないので、B16-BASIC から N88-BASIC へのファイル転送のためには、ディスクドライブ機械語ルーチンを用意する必要がある。

ところで、B16の5インチ2HDからN88-BASICへファイルメディア変換されたBASICファイルはFILESコマンドでファイル名を表示することができる。しかし、ファイル名の表示が正常になされたとしても、必ずしもそのままテキストファイルとして利用できるとは限らない。単にメディア変換されただけのファイルをloadしようとしてもDirect statement in fileという応答が返ってきて、正常にファイルを読み込むことができない。それは、B16-BASICプログラムの行の先頭に記録されている数バイトの文字列のためにPC-9801では正常に読めないのである。それ故、メディア変換されたファイルの内容の変更が必要となる。

### 3. 変換の道具とプログラミング手法

#### 3.1 ディスク・ダンプとディスク・エディタ

図1はB16-BASICの、図2はN88-BASICの同じ内容のプログラムのダンプ・リストの先頭部分である。いずれもアスキー形式でセーブしたものである。B16-BASIC、N88-BASICいずれもマイクロソフト系のBASICなので、アスキーセーブしたファ

パソコン上でのファイル変換

<< DISK EDITOR >>

PAGE: 1

DRIVE: 2 TRACK: 54(36H) SURFACE: 0 SECTOR: 8(08H)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0	0A	31	30	20	27	20	2A	2A	2A	2A	2A	20	4D	41	54	52	0 F10 ' ***** MATR
1	49	58	20	49	4E	56	45	52	53	49	4F	4E	20	2A	2A	2A	1 IX INVERSION ***
2	2A	2A	0D	0A	0A	32	30	20	27	20	44	41	54	41	20	49	2 **RFF20 ' DATA I
3	4E	50	55	54	0D	0A	0A	33	30	20	49	4E	50	55	54	20	3 NPUTRFF30 INPUT
4	22	4E	3D	22	3B	4E	0D	0A	0A	34	30	20	44	49	4D	20	4 "N=";NRFF40 DIM
5	41	28	4E	2C	4E	29	2C	58	28	4E	2C	4E	29	0D	0A	0A	5 A(N,N),X(N,N)CLFF
6	35	30	20	46	4F	52	20	49	3D	31	20	54	4F	20	4E	0D	6 50 FOR I=1 TO NR
7	0A	0A	36	30	20	46	4F	52	20	4A	3D	31	20	54	4F	20	7 LFF60 FOR J=1 TO
8	4E	0D	0A	0A	37	30	20	50	52	49	4E	54	20	22	41	28	8 NRFF70 PRINT "A(
9	22	3B	49	3B	22	2C	22	3B	4A	3B	22	29	3D	22	3B	0D	9 ";I";";";J;")=";C
A	0A	0A	38	30	20	49	4E	50	55	54	20	41	28	49	2C	4A	A LFF80 INPUT A(I,J
B	29	0D	0A	0A	39	30	20	4E	45	58	54	20	4A	0D	0A	0A	B )RFF90 NEXT JRFF
C	31	30	30	20	58	28	49	2C	49	29	3D	31	0D	0A	0A	31	C 100 X(I,I)=1CLFF
D	31	30	20	4E	45	58	54	20	49	0D	0A	0A	31	32	30	20	D 10 NEXT IRFF120
E	50	52	49	4E	54	0D	0A	0A	31	33	30	20	50	52	49	4E	E PRINTRFF130 PRIN
F	54	20	22	41	3D	22	0D	0A	0A	31	34	30	20	50	52	49	F T "A="RFF140 PRI

図1 B16-BASICのダンプ・リスト

Track	30	Surface	0	Sector	1		
0000	31	30	20	27	20	2A	2A 2A 2A 2A 20 4D 41 54 52 49 10 ' ***** MATRI
0010	58	20	49	4E	56	45	52 53 49 4F 4E 20 2A 2A 2A 2A X INVERSION ****
0020	2A	0D	0A	32	30	20	27 20 44 41 54 41 20 49 4E 50 * 20 ' DATA INP
0030	55	54	0D	0A	33	30	20 49 4E 50 55 54 20 22 4E 3D UT 30 INPUT "N=
0040	22	3B	4E	0D	0A	34	30 20 44 49 4D 20 41 28 4E 2C ";N 40 DIM A(N,
0050	4E	29	2C	58	28	4E	2C 4E 29 0D 0A 35 30 20 46 4F N),X(N,N) 50 FO
0060	52	20	49	3D	31	20	54 4F 20 4E 0D 0A 36 30 20 46 R I=1 TO N 60 F
0070	4F	52	20	4A	3D	31	20 54 4F 20 4E 0D 0A 37 30 20 OR J=1 TO N 70
0080	50	52	49	4E	54	20	22 41 28 22 3B 49 3B 22 2C 22 PRINT "A(";I";";
0090	3B	4A	3B	22	29	3D	22 3B 0D 0A 38 30 20 49 4E 50 ";J;")="; 80 INP
00A0	55	54	20	41	28	49	2C 4A 29 0D 0A 39 30 20 4E 45 UT A(I,J) 90 NE
00B0	58	54	20	4A	0D	0A	31 30 30 20 58 28 49 2C 49 29 XT J 100 X(I,I)
00C0	3D	31	0D	0A	31	31	30 20 4E 45 58 54 20 49 0D 0A =1 110 NEXT I
00D0	31	32	30	20	50	52	49 4E 54 0D 0A 31 33 30 20 50 120 PRINT 130 P
00E0	52	49	4E	54	20	22	41 3D 22 0D 0A 31 34 30 20 50 RINT "A=" 140 P
00F0	52	49	4E	54	20	22	98 22 3B 53 50 43 28 31 32 2A RINT "r";SPC(12*

図2 N88-BASICのダンプリスト

イルの格納状態は殆んど同じである。B16-BASICのアスキー・ファイルでは、プログラム行の区切り符号として、各行の先頭に 0Ah なるコードがつけられているが、N88-BASIC ファイルにはこのコードはついていない。また各行の終りには B16-BASIC, N88-BASIC 両者ともに 0Dh, 0Ah なるコードが、さらに、ファイルの終りには共に 1Ah なるコードがつけられている。B16-BASIC の各行の先頭にあるコード 0Ah のために、B16-BASIC ファイルを PC-9801 で

は正常に読み込むことができなかったのである。したがって、このコードを消去すれば、PC-9801 の標準形式のテキストとして正常に読み込むことが可能となる。

このようにファイルの内容を観察する目的のためにディスクダンプ・プログラムが必要となる。

ところで、B16-BASIC プログラムの各行の先頭にある 0Ah を消去すると当然ファイルの長さが変わって以後の取り扱いが面倒になる。0Ah を 20h (空白)

で置き換えるのが簡単である。短いプログラムならば人手でこの置き換え操作を行なうことによって B16 プログラムを N88-BASIC のプログラムとして利用できることになる。

この目的のためにはディスクエディタ・プログラムを用意する必要がある。

ディスクダンプ、ディスクエディタいずれにしても、CRT 上あるいはプリンタに出力すべきファイルを B16-BASIC 用にフォーマットされたディスクレットから読み込んでこなければならないが、先述したように B16-BASIC は MS-DOS の管理下にあり、ディスクとの読み書きは 1 セクタ 1024 バイト単位で行なわれるので、N88-BASIC で利用可能な DSKI\$ は使用できない。したがって、少なくともディスクレットからの読み込みルーチンは機械語でプログラミングしなければならない。その際 DISK BIOS を利用することにする。それには、AH レジスタに BIOS の内部識別コード(データ読み出しの場合 56h)を、AL レジスタに PDA (物理デバイスアドレス)を、BX レジスタにデータ領域の長さを、CH レジスタにセクタ長 (1024 バイトの場合 03)を、CL レジスタにシリンダ番号を、DH レジスタにヘッド番号を、DL レジスタにセクタ番号を、ES レジスタにデータ領域のセグメントベースを、BP レジスタにデータ領域のオフセットアドレスをそれぞれ格納して CALL 1Bh を行なえばよい。

### 3・2 ファイル転送

B16-BASIC 用のディスクレットから N88-BASIC 用のディスクレットへファイル転送するには、MS-DOS のディスク・ファイルの管理方法を知る必要がある。

MS-DOS では、ファイルの名前とそのファイルが格納されている場所を別々のテーブルで管理している。前者のテーブルはディレクトリと呼ばれ、後者のテーブルは FAT と呼ばれる。

ディレクトリには、ファイル名とそのファイルの格納場所の入り口及びファイル属性が記録されている。そして、1 つのファイルの情報のまとまりはディレクトリ・エントリと呼ばれるが、1 つのディレクトリ・エントリはディレクトリの中で 32 バイトからなる。先頭の 8 バイトにはファイル名が、続く 3 バイトには拡張子が入られる。12 バイト目にはファイル属性が、27 バイト目からの 2 バイトにはそのファイルがディスクの中のどこに格納されているかを示す FAT のエントリ番号が入り最後の 4 バイトにはファイルの長さがバイト単位で記録されている。

ファイル名の 1 バイト目が E5h の場合は削除されたファイルであることを表わし、またここが 00h の場合は未使用のディレクトリ・エントリであることを意味し、これ以降にはディレクトリ・エントリはない。

ところで、ファイル領域はクラスタ単位で管理される。ディレクトリ・エントリは最初のクラスタ番号のみを指し示す。2 番目以降のクラスタ番号は、FAT (ファイルアロケーションテーブル) の中に書かれている。すなわち、大きさが 1 クラスタ以内のファイルは、ディレクトリ・エントリの 27 バイト目のクラスタ番号によりそのファイルのディスク上の位置を知ることができるが、複数のクラスタから構成されているファイルの場合、最初のクラスタはディレクトリ・エントリのクラスタから知ることができても、2 番目以降のクラスタは FAT をみなければ、それがディスク上のどこに存在するかを知ることができない。

FAT 内の各エントリは 1.5 バイト (12 ビット) ずつが各々のクラスタに対応して、後に続くクラスタの番号を指し示している。そして、FFFh が現われたところでそのファイルは終りである。すなわち、FAT の内部は 3 バイトの領域を 1.5 バイトずつに分解してこれを 1 エントリとして管理しているので、後続クラスタ番号を読み取るには次のようにする。

- (1) 3 バイトのデータを後ろから 1 バイトずつ逆に並べる。
- (2) 1.5 バイトずつに分ける。
- (3) 最初の 1.5 バイトと 2 番目の 1.5 バイトを入れ替える。

また、求めるエントリが FAT の何バイト目から記録されているかを調べるには次式による。

開始番地 (FAT 内のオフセット番地) = (エントリ番号  $\times$  2)  $\times$  3

実際のファイル転送の作業手順は次のようにする。

- (1) 原フロッピーディスク (MS-DOS) から転送されるファイルのクラスタ番号を得る。
  - (2) そのクラスタのセクタ番号を計算する。
  - (3) 上の計算に基づいて、原フロッピーディスクから 1 セクタ (1024 バイト) を MS-DOS 用のバッファ (#0~3 に割り当てる) に読み出す。
  - (4) 256 バイトずつ N88-BASIC 用のバッファ (#4 に割り当てる) を経由して目的フロッピーディスクに書き込む。
  - (5) ファイルの終りまで上記(1)~(4)を繰り返す。
- 上の(2)のクラスタ番号からセクタ番号等を算出す

## パソコン上でのファイル変換

るには次式による。

$$T = (C + 9) \div 16$$

$$H = ((C + 9) \div 8) \text{ MOD } 2$$

$$S = ((C + 9) \text{ MOD } 8) + 1$$

ここに C: クラスタ T: トラック H: ヘッド  
S: セクタである。

なお、MS-DOSのMD2HDのFAT位置はT=0,  
H=0, S=2~3であり、ディレクトリの位置は  
T=0, H=0, S=6~8である。

### 3・3 テキスト検索・置き替え

B16-BASICとN88-BASICとではコマンドや文  
に次のような違いがある。

まず、B16-BASICはMS-DOSの管理下にあるの  
でDOSのコマンドとしてCHDIR, MKDIR,  
RMDIR, ENVIRON, SHELL, SYSTEMなどがあ  
るが、独立BASICであるN88-BASICでは当然こ  
れらは使えない。

また、B16-BASICのERDEV\$, KTN\$, SCREEN  
などはN88-BASICの文に該当するものがない。

しかし、以下に示すように、容易にN88-BASIC  
の文で置き替え可能なものもある。括弧内はN88-  
BASIC文である。LCOPY (COPY), PALETTE  
(COLOR), RESET (CLOSE), VIEW PRINT  
(CONSOLE), PMAP (MAP), CDBL\$ (AKCNV\$)  
CSNG\$ (KACNV\$), INPUT (KINPUT), KPOS  
(KINSTR), GET (GET@), PUT (PUT@) など。

また、B16-BASICとN88-BASICとではパラメー  
タの書き方に少しだけ違いのあるものもある。

CLS, COLOR, LOCATE など。

さらに、KEY ON, KEY OFFはB16とN88とで  
は意味が異なるが、B16-BASICでの意味はN88-  
BASICではCONSOLE文で表わすことができる。

ファイル転送後、機種によるBASIC言語の僅かな  
違いを目的のBASICに適合するよう、ファイルの内  
容を変換するためのユーティリティが必要である。  
テキスト検索プログラムはASCIIセーブされたプロ  
グラムの中に指定された任意の文字列がどこに含ま  
れているかをCRTやプリンタに表示する。すなわち、  
BASICプログラムの中に使用されているコマンドや  
変数名などがどの行で使用されているのかを表示する。

そのためには、BASICプログラムをシーケンシャ  
ルファイルとして1行ずつ読み込み、その中の変数  
名と行番号をテーブルに記憶しておき、ソーティン  
グを行ない、重ならないように順に表示すればよい。

このプログラムに、探し出した文字列と置き換え  
たい文字列とを交換する部分を付け加えれば異機種  
間の簡単なBASICファイル変換プログラムとなる。

たとえば、LOCATE文のようにX座標とY座標の  
位置が逆になっているような1対1の変換だけだと  
容易にファイル内容の変換が可能である。

### 3・4 漢字コード変換

セクタ単位で一括転送されたファイルに漢字コー  
ドが含まれている場合、転送後のファイル内の漢字  
コードを変換する必要が生ずる。

N88-BASICでは1バイトコードにはASCIIを、  
2バイトの漢字コードにはJIS漢字コードを使ってい  
る。この組み合わせではコードが重なる(21h~7Eh)  
ので切換えコードが必要である。

すなわち、漢字文字列の最初には漢字イン(KI)  
が、最後には漢字アウト(KO)が付けられる。これ  
らのコードには1B4Bh(ESC Kと呼ばれる)と1B48h  
(ESC Hと呼ばれる)が用いられる。

一方、B16-BASICでは1バイトコードはASCIIを  
使っているが、2バイトの漢字コードにはシフト  
JISコードを使っている。この組み合わせの場合は  
切換えコードを必要としない。というのは、アスキー  
コードは80hから9FhとE0hからFFhが未定義  
となっているので、この領域を漢字コードに用いる  
ことにすれば、漢字開始や終了を指示しなくてもこ  
れらのコードが現われたときそれが漢字であること  
が分るからである。

ところで、JISコードとシフトJISコードでは切  
換えコードの存在によって漢字コードが含まれるフ  
ァイルの大きさが異なってくることに注意しなけれ  
ばならない。アスキー形式でセーブされたファイル  
はシーケンシャルファイルとして行単位で読み書きで  
できるので、1行ごとに変換すればよい。B16-BASIC  
からN88-BASICに変換するには、コード80h~9Fh  
E0h~FFhが現われたら1B4Bhを前置してJISコー  
ドに変換し、上記以外のコードが現われたら1B48h  
を後置する。

### 参考文献

1. 押野崇芳 MS-DOS構造解析  
ナツメ社 1985年
2. 竹迫一雄, 二之宮弘 パソコンファイル変換法  
森北出版 1985年