

ファイル転送プログラムの開発

— スタンドアロン BASIC システムから CP/M システムへ —

小 田 哲 久

Development and Implementation of a File Converting Program

From the Stand Alone BASIC System to the CP/M System

Tetsuhisa ODA

Two kinds of Disk Operating Systems (DOS) are supplied by a computer manufacturer to control a floppy disk system of the personal computer : One is the stand alone Disk BASIC Language Processing System (stand alone BASIC), and the other is the CP/M Disk Operating System which is developed by Digital Research Co. Ltd. Indeed, each DOS has respective merits. For example, the stand alone BASIC is easy for the beginner to learn its command sets. On the other hand, the CP/M system can handle several language processing systems. Even so, the incompatibility between the two systems is found to be inconvenient for a user due to the fact that a disk file prepared by one system cannot be read by another system, vice versa. This Study investigated and developed the concepts of transferring a file(s) from one to another system and converting file(s) from one to another DOS in a personal computer atmosphere. This Study presented one program which can convert a file(s) from the stand alone BASIC system to the CP/M system. The program is written in the BASIC language, and can be applied to the OKI if-800 personal computer. An Application of the concepts can be easily done to any other personal computers.

1. [はじめに]

国産パーソナルコンピュータには、一般にインタープリタタイプの BASIC 言語処理系が標準搭載されており、この処理系を用いて稼動する事を前提条件としている場合が多い。特に発売開始時点では、この言語処理系が、ほとんど唯一のシステム・ソフトウェアである為に、早期に購入したユーザーは、好むと好まざるとにかかわらず、この言語処理系の上で応用プログラムを開発し、又、データを蓄積する事になる。その後、しばらくの期間を経て、オプションの DOS (一般には CP/M) が発売され、それを購入する事になるが、その場合、DOS 購入以前に作成したプログラム・データ等を利用する上で、ファイルに互換性の無い事が問題となる。本レポートでは、異機種・異 DOS で作成されたディスク・ファイルの転送対策の概要を論じた後、特定機種上で、標準搭載さ

れた BASIC 言語処理系から、CP/M で読めるファイルを作成するプログラムを提示する。

2. [スタンドアロン BASIC の特徴]

従来のコンピュータの通信では、ディスクを作動させる場合、まず DOS を稼動させ、その管理下で、何らかの言語処理系を走らせるのが常であるが、最近のパーソナルコンピュータに標準搭載された BASIC インタープリターは、ディスクを外部記憶装置として使用する場合でも、特別の DOS を必要とする訳ではなく、それ自身の内部に、簡単な DOS 機能を含んでいる。それ故、自立しているという意味で、スタンドアロン BASIC と呼ばれている。その特徴は、

(1) 特別の DOS を必要とせず、操作が簡単で、初心者向きである。

- (2) 処理言語が BASIC に限定され、又、ハードウェアも特定されるので、内蔵 EDITOR の仕様を特殊化し、色々便利な機能を持たせる事ができる。特に CRT 上でのスクリーン・エディット機能は、初心者には大変に便利である。
- (3) インタープリターなので、デバッグがしやすい。特に、エラー発生時に各変数の値を表示してくれたり、現在実行中の行番号をトレースしてくれる機能は便利である。
- (4) 特定ハードウェアを前提として設計されている為、特殊キーの利用、グラフィック機能の利用をはじめ、オーディオカセット、AD/DA コンバータ等、各種入出力装置の有効利用が可能かつ容易である。

といった所であろう。これらの長所の現れるゆえんは、DOS・EDITOR・LANGUAGE・PROCESSOR・DEBUGGER の四者が一体化され、特定ハードウェア専用に設計・製作されている事にあろう。この様に、DOS をも内包してしまった言語処理系というのは、一見、原始的に見えて、その実、「操作が容易」という本質的に重要な要素を含んでおり、未来的ともいえる。例えば、近年とみに評価の高まって来た FORTH 処理系¹⁾は、元来、DOS 機能の内包を指向した言語系である。

一方、スタンドアロン BASIC の欠点を列挙すると、

- (1) DOS 機能が極めて貧弱であり、ファイル転送命令すら無い場合がほとんどである。この為、ファイルの編集・保守が困難である。
- (2) DOS 機能の機種依存性が高く、特定機種上で作成したファイルの入ったフロッピーディスクを他機種のスタンドアロン BASIC のディスク装置にかけても、ディスクへの書式が異なる為、読みとれないのが普通である。
- (3) インタープリター言語である為、コンパイラ言語に比べて処理速度が遅く、科学技術計算等、高速演算を必要とする JOB に対しては、不適切である。

といった所であろう。

3. [CP/M の特徴]

CP/M は、80系 CPU(インテル8080, 同8085, ザイログ Z80)をメイン CPU とする、小型コンピュータシステムの為の汎用 DOS で、米国製のほとんどのパーソナルコンピュータに、オプション又は標準で搭載され、今や国際規格に近いものとなって来た。米国デジタル・リサーチ社の商品であるが、最近、日本製のパーソナル・コンピュータにも搭載され始めている。CP/M の下で開発されたソフトウェアは、同じく CP/M を DOS に採用しているシステムでありさえすれば、たとえ機種(ハード

ウェア)が異っていても、特別の変更なしに動作させる事ができる訳で、現に、各種 CPU の為のアセンブラーや、BASIC, FORTRAN, COBOL, PASCAL, PL/I 等々、高級言語のコンパイラーは無論の事、ワードプロセッサ、データベース等、各種のアプリケーションプログラムも、豊富に商品化されている。ユーザー側から見た場合、CP/M の最大の魅力は、充実したソフトウェア群を、比較的安価に入手できるという点だが、一方、メーカー側から見たメリットは、各種ソフトウェア(特に言語処理系)を、わざわざ製作する必要が無いという点に尽きよう。CP/M は本来、開発支援システム用の DOS として使う事を想定して組み立てられたもので、小さな機能しか備えていないし、占有メモリを極力小さく抑える為に処理速度を犠牲にしている面もあるが、シングルユーザー・シングルタスクを前提としたパーソナルコンピュータにとっては、最適とはいえないにしても、必要十分な機能を有しているといえよう。

4. [ディスク・ファイルの互換性]

現在、パーソナル・コンピュータ用のスタンドアロン BASIC 言語処理系は、ほとんどが、米国マイクロソフト社の手で開発されており、そうでない場合(例えばシャープMZシリーズパーソナルコンピュータ)でも、マイクロソフト社の BASIC 言語仕様と類似の仕様を採用している。この結果、ソースプログラムレベルでいえば、かなりのコンパチビリティが生れ、特定機種上で開発した BASIC プログラムは、特殊な、その機種独自の命令を使わないように注意すれば、他の機種へ、極めてわずかな変更で、時には一切の変更なしに、移植して動作させる事が可能となっている²⁾。しかし、同じマイクロソフト社製スタンドアロン BASIC といっても、機種ごとにディスクの書式は異なる為、プログラムの書き込まれたディスクを別の機種で直接読みとる事ができず、プログラムの移植は、キーボードからの再入力という手段によるのがふつうである。しかし、この再入力作業は非常に不能率であるし、又、バグの発生する恐れも多分にある。さらに、大量データの再入力、もはやほとんど不可能といえるほどの大変な作業となる。

このような、ディスク書式の相違による、ファイルの互換性の無さは、①異機種のスタンドアロン BASIC 言語処理系とおしの間には於いてのみならず、②異機種の CP/M どおしの間でも同様であり、又、③同一機種上での DOS の相違、すなわち、スタンドアロン BASIC 言語処理系と、CP/M との間でも同様である。

5. [ファイル転送の対策]

異機種・あるいは異なる DOS の間で、プログラムやデータのファイルを転送する手段としては、両機種を回線で結び、〈送り出し→受取り〉の規格を同一にして送るのが一般的である³⁾⁹⁾。現在のパーソナルコンピュータでは、この目的の為に、RS-232-C インターフェースを、オプション又は標準装備として用意している機種が多いが、これを CP/M がサポートしていない場合もあり、又、余り遠くまで送れないという難点もある。

そこで、別の方法として、オーディオカセットテープあるいはフロッピーディスクによる、オフライン転送という手段が考えられるわけであるが、このうち、オーディオカセットテープについては、物理的な書き込み方式が各社まちまちであり、又、ほとんどの場合、CP/M がこれをサポートしていないので、上記目的の為に転送メディアとしては不適當である。一方、フロッピーディスクについては、現在のパーソナル・コンピュータの多くが 5 ¼ インチサイズの、ソフトセタータイプのものを採用しており、両面か片面か、あるいは単密度か倍密度か、といった違いはあるにせよ、オーディオカセットより有利な条件にある。物理的な記録方式が違う場合には、適切な手段はないが、幸いにも、国産パーソナルコンピュータの場合、記録方式に違いはなく、ソフト的な工夫をすれば、転送が可能である。

転送上の問題点は既述した通り、ディスク書式の相違であるが、この事は、逆にいえば、**Ⓐ**受け取り側の書式にあわせて、ディスクへファイルを書き込むか、**Ⓑ**送り出し側の書式で受け手が読み取れば、転送が可能となる。この、**Ⓐ****Ⓑ**以外の対策として、岡村迪夫氏⁶⁾⁷⁾の TR ディスク構想がある。これは、**Ⓒ**特定機種あるいは特定 DOS と関係のない独得の、しかし、取り扱いの容易なファイル書式を設定して、自由にファイルを交換しようとするものである。さらに、**Ⓐ**と**Ⓑ**の複合形ともいべき方式として、野上道男氏⁹⁾の「完全共用ディスク」方式がある。これは、異機種のファイル書式が極めて類似しているにもかかわらず、ディレクトリの位置が異っている場合、受け手の規約に従ったディレクトリ情報を作成し、これを 1 個のファイルとして、受け手の DOS のディレクトリ用の位置に書き込むもので、やや特殊なケースでしか採用できないという制約はあるものの、大変興味深い方法である。

6. [CP/M ファイルの書式変換]

異機種 CP/M で作製された 5 ¼ インチディスクファイルを読もうとするとき、

①全く変更なしに読める場合。

②わずかな変更（例えばトラックをずらすだけ）で読める場合。

③大巾な書きかえをしないと全く読めない場合。

の 3 つのケースが考えられる。①のケースは、例えば、富士通 MICRO-8 と、NEC、PC-8801 (両面倍密度ディスクドライブ使用)、シャープ MZ-80B の場合がそれであり、②は、これら 3 機種と、沖 if-800/20 の場合である。③は、NEC、PC-8001 (片面倍密度ディスクドライブ使用) と、沖 if-800/20 あるいは他の両面倍密度採用機種の場合がそれに当る。小田が以前発表したプログラム⁹⁾、及び、竹迫一雄氏がこれと独立に開発・発表されたプログラムが¹⁰⁾¹¹⁾が、いずれもこのケースのファイル書式変換を中心とした理由はここにあり、その変更上の困難さは、ひとえに、ディレクトリの管理するデータブロックのサイズの相違によるものである。

7. [スタンドアロン型 BASIC によるファイルを、CP/M ファイルに書式変換して転送する方法]

スタンドアロン BASIC 言語処理系によって作成されたディスク・ファイルを、CP/M ファイルへ書式変更するプログラムは、NEC、PC-8001 用については、既に今泉 修氏によって発表されている¹²⁾。氏の方法は、CP/M (受け手)のディスクドライブ上で、N-BASIC (送り手)の書式で作成されたファイルを読み取り、別ドライブへ転送して、CP/M で読める書式のファイルを作成しようというものである。氏のプログラムはアセンブラで記述されており、正統的であるが、他機種で実現するには、その機種独自の、スタンドアロン BASIC によるファイルの書式を調査する必要がある、又、各 CP/M の BIOS の構造に依存する部分もあるので、移植が容易とはいえない面がある。そこで、本報では、if-800 model 20 上で、スタンドアロン BASIC から CP/M へのファイル転送を、スタンドアロン BASIC 上で実行する事にした。その方式は、前述した、転送対策の分類でいえば**Ⓐ**に当る。当然の事ながら、言語もスタンドアロン BASIC で書かれているので、他機種への移植も容易である。

8. [転送プログラムの使用方法と機能]

本プログラムの使用は極めて簡単であり、次の手順に従って行なえば、会話式で実行される。

手順 1. BASIC インタープリターを起動させる。

手順 2. 転送プログラムを LOAD する。

手順 3. #1 ドライブには、スタンドアロン BASIC で作られたプログラムやデータのファイルが記録されたディスクを挿入する。又、#2 ドライブ

には、アドレス・フォーマットがずみで、何も記録されていないディスクを挿入する。

手順4. 手順3に間違いの無い事を確認したら、RUNする。

手順5. 自動的に、#1ドライブに挿入されているディスクの全ファイル名と空白領域のサイズが画面に表示され、PLEASE INPUT FILE NAMEという表示が行なわれるので、転送したいファイルの名前をキーインし、RETURNキーを押す。名前をキーインせずに、RETURNキーのみを押せば、実行終了となる。

手順6. キーインされたファイル名は、ただちに、プリンター上に出力される。

手順7. DATA or BASIC or OTHER LANGUAGE ……D or B or Oと画面表示されるので、該当するキーを押す、RETURNキーを押す。

手順8. 手順7でキーインされた文字がDの場合は、ファイルの型はDATとなり、Bの場合はBASとなる。Oの場合、BASICの文番号が消されて転送されるが、この場合に限りFILE TYPE: ASM or FOR or PAS or PLIと表示され、該当用語の3文字のファイルの型名をキーインするよう要求される。

手順9. 以上の手順が完了すると、転送が開始され、#1ドライブ上の該当ファイルの中身が画面表示されつつ、#2ドライブ上に、CP/Mで読める形に記録されてゆく。ただし、ファイルの型名がD(すなわちデータ)以外の場合は、一旦、#1ドライブ上の空白領域へ、CP/Mのエディターで読み書きできる書式に変換して記録し、これが再度#2ドライブへ転送される。この手順はすべて自動的に行なわれるので、使用上、特別な操作を必要とするわけでは無いが、#1ドライブ中のディスクの空白領域が充分にないと、エラー表示される。このバッファ領域は反復使用されるので、転送したいプログラムファイルのサイズよりも大きい程度で充分である。

手順10. 該当ファイルの転送が完了すると手順5へもどる。

手順11. 手順5でRETURNキーのみが押されると、#2ドライブ上へ、CP/M用のディレクトリ情報が書き込まれるが、この情報は、ディスクへの書き込みに先だって、プリンター上へ印刷される。そしてすべての作業が完了した後、画面上へJOB ENDと表示され、実行終了を知らせる。この時、#2ドライブ上には、CP/Mで読め

る形のファイルが完成している。

9. [他機種で実行する場合の修正点と制約]

本報で呈示したプログラムは、アルゴリズムは単純なので、ソースリストを読むだけで内容が把握できよう。他機種上で実行される場合、78行目から80行目の式に含まれる3という数値を、その機種のCP/Mのシステム用リザーブトラック数に直していただきたい。その他の修正点としては、18行目の1000という数値を、その機種のCP/Mで利用可能な、ディスク一枚当りの最大レコード数に変更する事であるが、これは特に変更を加えなくても、一応動作するはずである。本プログラムは、現在、両面倍密度ミニフロッピーディスクを使う場合の主流となっている、256バイト/セクター、16セクター/トラック、2KB/ブロックを前提としている。ただし、トラック数はif-800では35トラックであるが、この数には若干の融通性がある。

10. [結び]

今回呈示したプログラムは、当初の目的どおりに動作し、処理速度も使用上充分であった。しかし、本来この種の機能はメーカーサイドで用意すべきものであるのに、それがなされていない事は、大きな問題である。パーソナル・コンピュータ市場は今後一層の拡大をするに違いないが、これまでの様な、ユーザーのソフト開発能力に依存した政策は、早晚変更をせまられる事であろう。現在、16ビットマシンの開発・発表があいついで行なわれているが、この機会に、ディスクの書式が統一される事を願うものである。

参考文献

- 1) 井上外志雄：拡張性のある言語 FORTH, bit, 13 (14), 20-23, 1981.
- 2) 土井政則：パーソナルコンピュータの BASIC 徹底比較——前編, ASCII, 6 (3), 186-200, 1982.
- 3) 有澤 博：マイコンの新しい使い方——ページックマスター・レベル3による汎用インテリジェント端末の製作——, bit, 14 (2), 7-18, 1982.
- 4) KEY LAB 中之島：MICRO-8 ⇄ PC-8001 パーソナルコンピュータ間のファイル転送——RS232C インターフェースで異機種を直結, マイコン, 5(6), 187-192, 1982.
- 5) KMC PLANET Project Team：コンピュータ・ネットワーク, ASCII, 5(7), 82-107, 1981.
- 6) 岡村迪夫：異機種コンピュータでデータやプログラムを読むための、転送用ディスク・フォーマットと

- その用法, Interface, 8 (8), 231-239, 1982.
- 7) 岡村迪夫: super BASIC, 77-85, CQ 出版社, 東京, 1982.
- 8) 野上道男: PC/FM 完全共用ディスクの作成, I/O, 7 (7), 345-353, 1982.
- 9) 小田哲久: PC-8001用 CP/M → if800用 CP/M ファイル書式変換プログラム, Interface, 8 (7), 242-249, 1982.
- 10) 竹迫一雄: マイクロ・コンピュータ フロッピーディスク・オペレーティングシステム CP/M*のファイル構造の機種による相違および異機種間ファイル変換プログラム, 福岡大学理学集報, 12(2), 71-82, 1982.
- 11) 竹迫一雄: CP/M ファイルの構造と異機種間の相互変換プログラム=PC, MZ, IF, FM=, Interface, 9 (1), 203-212, 1983.
- 12) 今泉 修: N-BASIC to CP/M ファイル転送プログラム, Interface, 8 (12), 284-288, 1983.
- 13) 沖電気: if-800 model 20 OKI-BASIC 文法解説書, 1980.
- 14) 沖電気: if-800 標準 CP/M 解説書, 1980.
- 15) 沖電気: if-800 if-CP/M ユーザーズマニュアル, 東京, 1980.
- 16) Rodney Zaks: 標準 CP/M ハンドブック, アスキー出版社, 東京, 1981.
- 17) 石田 芳: CP/M の使い方, 産報出版社, 東京, 1981.

(受理 昭和58年1月16日)

付録 プログラム・リスト

```

1 'BASIC FILE → CP/M FILE
2 ' (DRIVE 1) → (DRIVE 2)
3 '
4 OPTION BASE 1:FALSE=0:TRUE=-1
5 DEFINT I-N
6 ON ERROR GOTO 163
7 DIM X1$(8),X2$(8)
8 DIM FB1$(64),FB2$(64),FC$(64),FD$(64),FE$(64,16) : 'directory-informations
9 DIM A$(8),B1$(8),B2$(8),C$(8),D2$(8),D$(8),E$(8,16)
10 FOR I=1 TO 8:K=(I-1)*32
11   FIELD #0,K AS D1$,1 AS A$(I),8 AS B1$(I),3 AS B2$(I),1 AS C$(I),2 AS
D2$(I),1 AS D$(I)
12   FOR J=1 TO 16:L=K+J+15
13     FIELD #0,L AS D3$,1 AS E$(I,J)
14   NEXT J
15 NEXT I
16 FIELD #0,128 AS CPMBF1$,128 AS CPMBF2$
17 FIELD #1,128 AS Z1$,128 AS Z2$
18 N=1000 : 'maximum record size
19 CLN%=2
20 MM=1
21 TJ=1
22 UN%=0 : 'USER NUMBER
23 DRBSC%=1
24 DRCPM%=2
25 'get next file name
26 FILES:BUFA%=DSKF(DRBSC%):PRINT "Disk Free Area : ";BUFA%;" Clusters"
27 EDFL=FALSE
28 DFULL=FALSE
29 INPUT "PLEASE INPUT FILE NAME";FLN$:DRFLN%=CHR$(DRBSC%+&H30)+" : "+FLN$
30 LPRINT "FILE NAME : ";FLN$
31 IF FLN%="" THEN CLOSE #1:GOSUB 114:PRINT "JOB END":END : 'Program end
32 'check FILE NAME
33 I=1
34 WHILE (I<64) AND (FB1$(I)<>FLN%) AND (FB1$(I)<>"")
35   I=I+1
36 WEND
37 IF FB1$(I)=FLN% THEN PRINT "Already Transferd":GOTO 29
38 '
39 II=0
40 JJ=0
41 EXT%=-1
42 LL=0
43 RCSZ%=0
44 '
45 'DATA or BASIC or OTHER LANGUAGE
46 INPUT "DATA or BASIC or OTHER LANGUAGE ----- D or B or O";DBO$
47 IF DBO$="D" THEN FLTP$="DAT"
48 IF DBO$="B" THEN FLTP$="BAS"
49 IF DBO$="O" THEN INPUT "File Type : ASM or FOR or PAS or PLI ";FLTP$
50 IF DBO$<>"D" THEN GOSUB 169 :IF DFULL=TRUE THEN GOTO 29
51 OPEN DRFLN$ AS #1
52 J=1
53 WHILE (TJ<=N) AND (MM<64) AND (NOT EDFL)
54   I=0
55   WHILE (I<8) AND (TJ<=N) AND (NOT EDFL)
56     GET #1,J
57     J=J+1 : 'increment record number
58     TJ=TJ+1 : 'total record number
59     RCSZ%=RCSZ%+2 : 'increment record size for directory
60     I=I+1

```

```

61     X1$(I)=Z1$
62     X2$(I)=Z2$
63     WEND
64 '                                     LPRINT "TRANSFER :REC.NO. ";J-I;" TO ";J-1
65     IF I<>0 THEN GOSUB 77:'write DATA for CP/M
66     WEND
67     CLOSE #1
68     IF DBO$(">"D" THEN KILL BUFF$
69     IF (TJ<=N) AND (MM<=64) THEN GOTO 25
70     IF (MM>64) THEN PRINT "***Directory Full**"
71     IF (TJ>N) THEN PRINT "***Disk memory Overflow**"
72     PRINT "Transfer error : ( File ;:FLN$;" )"
73     GOSUB 114
74     PRINT "JOB ABNORMAL END"
75     END
76 'main Program end.
77 'sub : write data for CP/M
78     NT=INT(CLN%/4)+3
79     SD%=INT((CLN%-4*(NT-3))/2)
80     NS=(CLN%-4*(NT-3)-2*SD%)*8
81     K=0
82 ' LPRINT "CLN%";CLN%,"SD";SD%,"NT";NT,"NS";NS
83     WHILE K<I
84         K=K+1
85         PRINT X1$(K);X2$(K);
86         NS=NS+1
87         LSET CPMBF1$=X1$(K)
88         LSET CPMBF2$=X2$(K)
89         DSKO$ DRCPM%,SD%,NT,NS
90     WEND
91     GOSUB 94
92     CLN%=CLN%+1
93     RETURN
94 'sub : prepare DIRECTORY-INFORMATIONs for CP/M
95     LL=LL+1
96     II=II+1
97     FE%(MM,LL)=CLN%                                     : 'memory CLUSTER NUMBER
98     IF (II<8) AND (NOT EDFL) THEN RETURN
99     II=0
100    JJ=JJ+1
101    EXT%=EXT%+1                                         : 'increment EXTENT NUMBER
102    IF (JJ<2) AND (NOT EDFL) THEN RETURN
103    IF JJ=2 THEN RCSZ%=RCSZ%-128
104    FB1$(MM)=FLN$                                       : 'store FILE NAME
105    FB2$(MM)=FLTP$                                       : 'store FILE TYPE
106    FC$(MM)=EXT%                                         : 'store EXTENT NUMBER
107    FD$(MM)=RCSZ%                                       : 'store RECORD SIZE
108    II=0
109    JJ=0
110    LL=0
111    MM=MM+1
112    RCSZ%=0
113    RETURN
114 'sub : write directory for CP/M
115    FOR I=1 TO MM-1
116        LPRINT "FILE : ";FB1$(I);". ";FB2$(I),"EXT ";FC$(I),"REC ";FD$(I)
117        FOR J=1 TO 16
118            LPRINT FE%(I,J);
119        NEXT J : LPRINT
120    NEXT I
121 '        RETURN
122 '
123    E5$=CHR$(&HE5)
124    E5H2$=STRING$(&H2,E5$)
125    E5H3$=STRING$(&H3,E5$)

```

```

126 E5H8$=STRING$(&H8,E5$)
127 E5H80$=STRING$(&H80,E5$)
128 BL2$=STRING$(2,CHR$(0))
129 UNC$=CHR$(UN%)
130 I=1
131 L=1
132 WHILE I=<(INT((MM-1)/8)+1)
133   FOR J=1 TO 8
134     IF L<MM THEN GOSUB 141 ELSE GOSUB 152
135     L=L+1
136   NEXT J
137   DSK0$ DRCPM%,0,3,I
138   I=I+1
139 WEND
140 RETURN
141 'sub : case L<MM :write No.L directory
142   LSET A$(J)=UNC$
143   LSET B1$(J)=FB1$(L)
144   LSET B2$(J)=FB2$(L)
145   LSET C$(J)=CHR$(FC$(L))
146   LSET D2$(J)=BL2$
147   LSET D$(J)=CHR$(FD$(L))
148   FOR K=1 TO 16
149     LSET E$(J,K)=CHR$(FE$(L,K))
150   NEXT K
151 RETURN
152 'sub : case L>=MM :write E5
153   LSET A$(J)=E5$
154   LSET B1$(J)=E5H8$
155   LSET B2$(J)=E5H3$
156   LSET C$(J)=E5$
157   LSET D2$(J)=E5H2$
158   LSET D$(J)=E5$
159   FOR K=1 TO 16
160     LSET E$(J,K)=E5$
161   NEXT K
162 RETURN
163 ' sub : error Processing
164 IF ERR=53 THEN PRINT "NO File":RESUME 29
165 IF ERR<>61 THEN PRINT "ERROR CODE = ";ERR,"ERR.LINE ";ERL:END
166 PRINT "INPUT PAST END CONDITION"
167 EDFL=TRUE
168 RESUME 63
169 'sub : save BASIC & OTHER LANGUAGE
170 CRLF$=CHR$(&HD)+CHR$(&HA)
171 BUFF$=CHR$(DRBSC%+&H30)+": "+"BUFF-CPM"
172 OPEN DRFLN$ FOR INPUT AS #1
173 IF LOF(1)>=BUFA%*8 THEN CLOSE #1:PRINT "Disk Full":DFULL=TRUE:RETURN
174 OPEN BUFF$ FOR OUTPUT AS #2
175 WHILE (NOT EOF(1))
176   LINE INPUT #1,SORCE$
177   IF DBO$="0" THEN IXSP%=INSTR(SORCE$," "):SORCE$=RIGHT$(SORCE$,LEN
(SORCE$)-IXSP%)
178   PRINT #2,SORCE$:CRLF$;
179 WEND
180 CLOSE #1,#2
181 DRFLN$=BUFF$
182 RETURN
183 '

```