SHARP

SHARP
COMPUTER
SOFTWARE

# X1turbo用
# turbo CP/M V2.2（漢字版）

CZ-130SF

## ユーザーズマニュアル

## はじめに

このたびは，シャープパソコンテレビ X1 turbo シリーズ用 turbo CP/M（CP/M™V2.2 Word Master™）をお買いあげいただきまことにありがとうございました。
ご使用になる前にこのマニュアルをよくお読みいただき，本機を正しくお使いください。**また，ソフトウエア使用権許諾契約書／マイクロプロ使用者契約書はかならずお読みください。**

〈ご使用いただけるプリンタ〉

| | |
|---|---|
| CZ-800P | CZ-80PK |
| CZ-8PD2/SP-80 | CZ-8PP2 |
| CZ-8PK2 | CZ-8PN1/CZ-8PK3(4) |
| MZ-IP06 | MZ-1P07 |
| MZ-1P08 | MZ-1P10A |
| ESC/P-ext | RP-80Ⅱ |
| FP-80 | UP-130K |
| PC-8822/PC-PR201 | M-1009X |
| M-1009 | LPR-24T |
| KP-3000 | MZ-1P17 |

〈ご使用いただけるディスプレイ〉
本機のご使用に際しましては，高解像度ディスプレイをご使用ください。

なお，商品構成は下記の通りです。

turbo CP/M マスターディスク……1枚
ユーザーズマニュアル……1冊
契約書……1通
登録カード……1通

**ご注意**

このマニュアルは，シャープパソコンテレビ X1 turbo シリーズのシステムソフトウエア CP/M™V2.2に基づいて作成されています。

(1)シャープパソコンテレビ X1 turbo シリーズではシステムソフトウエアをフロッピーディスクにより供給しています。各システムソフトウエアおよび本書の内容は，改良のために予告なく変更することがありますのでファイルバージョンナンバーには特にご注意されるようお願い致します。
(2)本書は内容について充分注意し作成しておりますが，万一ご不審な点・お気付きのことがありましたら，お買い求めの販売店もしくはもよりのシャープお客様ご相談窓口にご連絡ください。

※CP/M™V2.2は，米国デジタルリサーチ社の登録商標です。
※Word Master™は米国マイクロプロ社の登録商標です。

INDEX PAGE

## 第1章 turbo CP/Mとは


1 turbo CP/Mの特長 1
2 CP/Mの働き 2
3 turbo CP/Mの利用方法 3
4 turbo CP/Mの構成 4


## 第2章 turbo CP/Mを使用するにあたって


1 turbo CP/Mの起動方法 7
2 マスターディスクのバックアップコピー 9
3 マスターディスクに含まれる各種ファイル 13
4 作業用マスターディスクの作成 17
5 プリンタの設定 19


## 第3章 turbo CP/Mの使い方


1 キーの使い方 21
2 日本語入力の方法 23
3 ビルトインコマンドとトランジェントコマンド 24
4 マニュアル表記方法について 25


## 第4章 turbo CP/Mによるプログラム作成


1 アセンブラによるプログラム作成手順 27
2 アセンブラプログラムのフォーマット 32


## 第5章 ビルトインコマンド


1 d：(ログインディスクの変更) 39
2 DIR(ファイル名出力) 41
3 REN(ファイル名の変更) 43
4 ERA(ファイル削除) 44
5 TYPE(ファイル内容出力) 46
6 USER(ユーザーエリアの移行) 47
7 SAVE(メモリ内容のセーブ) 48

INDEX PAGE

## 第6章 トランジェントコマンド


1 STAT(ファイルおよび周辺装置の設定・状態確認コマンド) 51
2 PIP(周辺装置間データ転送コマンド) 56
3 ED(テキストエディタコマンド) 60
4 ASM(8080アセンブラ) 64
5 LOAD(HEXファイルからCOMファイルへの変換コマンド) 68
6 DDT(8080デバッガ) 69
7 DUMP(ディスクファイルの16進出力コマンド) 72
8 SUBMIT(バッチ処理コマンド) 73
9 XSUB(拡張バッチ処理コマンド) 75
10 MOVCPM(CP/Mシステムのメモリーサイズ変更コマンド) 76


## 第7章 turbo CP/M強化コマンド


1 STARTUP(各種パラメータの初期設定) 81
2 FORMAT(フロッピーディスクのフォーマット) 88
3 COPY(ディスク間のオールコピー) 89
4 SYSGEN(CP/Mシステムのコピー) 92
5 DICCOPY(日本語処理用の辞書コピー) 93
6 DICSET(辞書ディスクの割付け) 96
7 DISKSET(使用ディスクの割付け) 97
8 HDMAINT(ハードディスクの領域確保・解除) 100
9 HDBACKUP(ハードディスクのバックアップ) 105
10 HDOFF(ハードディスクの終了設定) 109
11 TERM(ターミナル・リモートモードでの通信) 110
12 CMT(ディスク・カセット間のファイル転送) 114
13 KEY(ファンクションキーの設定) 119
14 TIME(日付・時刻の設定) 121


## 第8章 コントロールコードとエスケープシーケンス


1 コントロールコードとエスケープシーケンスの使い方 123
2 コントロールコード 125
3 エスケープシーケンス一覧 126
4 エスケープシーケンス詳細 128


## 第9章 システムコール


1 システムコールの使い方 137
2 システムコール一覧 139
3 システムコール詳細 142

INDEX PAGE

## 第10章 CP/M BIOSコール


1 CP/M BIOSコールの方法 157
2 CP/M BIOSコールの種類 158
3 直接BIOSコール(BIOS ROMコール) 159


## 第11章 BASICとのリンクについて


1 BASICとのリンク方法 161
2 BASICサンプルプログラム 164


## 第12章 turbo CP/Mシステム


1 turbo CP/Mディスク構成 167
2 システムスクラッチエリアの内容およびメモリ構成 168
3 周辺機器の割あて(IOバイトの設定) 169
4 ディスクパラメータテーブル 170
5 FCB(ファイルコントロールブロック) 173
6 DMA(ダイレクトメモリアドレス)について 175


## 第13章 ワードマスター(WORD MASTER)


1 はじめに 177
2 ご使用になる前に 178
3 ワードマスターの起動 179
4 ワードマスターのビデオモード 181
5 ワードマスターのコマンドモード 186
6 ワードマスターのヘルプメニュー 192
7 ワードマスターのまとめ 193
8 エラーメッセージと情報メッセージ 196


## 第14章 付 録


1 コマンド一覧 199
2 ニーモニック一覧 205
3 プリンタの登録 211
4 エラーメッセージ 213

# 第1章

# turbo CP/Mとは

# 1 turbo CP/M の特長

turbo CP/M は米国デジタルリサーチ社の CP/M の機能に加えて X1 turbo 特有のハードウェアをサポートしたオペレーティングシステムです。標準オペレーティングシステムにビジネスユースには欠かせない日本語処理機能を付加しました。これにより漢字アプリケーションソフトの開発にもご利用いただけます。

**特　長**

①現行 CP/M（CZ-5CPM，CZ-3CPM，CZ-128SF）の機能を包含しています。※

②日本語処理機能をサポートしています。(オプションのシステムユーザー辞書の利用も可能。)

③各種ディスクドライブをサポートしています。(5 インチ2DD，2HD，8 インチ1S，2D，ハードディスク)

④内蔵の RS-232C をサポートしています。

⑤テキストのアトリビュートをコントロールできます。
テキストに色，反転，ブリンクなどがつけられるほか，倍文字，ユーザー定義のキャラクターゼネレータも使用できます。

⑥X1turbo の持つグラフィック機能(640×400ドット)をフルに使用できます。グラフィックのドット毎に色が指定できるほか，パレットやプライオリティの機能も使えます。

⑦内蔵のサウンドゼネレータをサポートしています。

⑧別売の320K バイトの外部 RAM をサポートしており，ディスクと同様に扱えます。
また，グラフィック RAM を外部記憶装置としてファイルの一時記憶などに使えます。

(ただし，⑤，⑥，⑦についてはエスケープシーケンスをご使用ください。)

また，マイクロプロ社のスクリーンエディターであるワードマスター(Word Master)を搭載しております。このワードマスターは日本語処理可能となっておりますので日本語を含むテキストファイルの作成，編集を効率よく行なうことができます。

※turbo CP/M では別売の RS-232C カード(CZ-8RS，CZ-8BM2)のサポートはしておりません。

## 2 CP/Mの働き

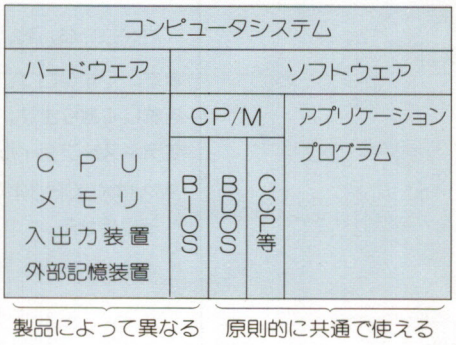
CP/M（Control Program for Microcomputer）は，80系CPU（8080，Z80，8085）を使用するコンピュータの〝OS（オペレーティングシステム）〟です。〝OS〟は〝ハードウェアをより簡単に能率よく働かせるために，アプリケーションプログラムとの間に位置し，入出力装置（キーボードやCRT画面など）の制御，プログラムファイルの管理などを行なう基本的なソフトウェア〟と言えます。CP/Mもその〝OS〟の1つで，8ビットマイクロコンピュータの〝OS〟として最も広く利用されています。（近年，16ビット用CP/Mも開発されたため8ビット用は特にCP/M−80と呼ばれることが多くなっています。）

マイクロコンピュータのハードウェア構成は，同じCPUを使用していても製品によって異なるためそれぞれに対応したプログラムが必要となります。したがって，ハードウェアの構成が異なる他の機種で実行することは通常，不可能です。

しかし，CP/Mシステム上で動作するアプリケーションプログラムは，ハードウェアが異なっていても基本的にはプログラムの変更なしで利用することが可能です。（ハードウェア特有の機能等を利用している場合は利用できないこともあります。）

コンピュータシステムは，ソフトウェアによりハードウェアを制御し目的の仕事を処理していきます。CP/Mでは，ハードウェアを直接制御する機能別のプログラムを〝BIOS（Basic Input Output System）〟といい，それらのプログラムを呼び出す方法を統一することによりソフトウェアの互換性を大幅に高めることに成功しました。

このため80系CPUを使用するマイコンシステムでは，ハードウェアの構成が変わってもCP/Mで必要とする〝BIOS〟部分のプログラムを作ることによって，他のCP/Mシステムで作られたプログラムを動作させることができるようになります。




CP/Mは1974年米国デジタルリサーチ社で，8080CPUを使用するマイコンシステムのフロッピーディスク用OSとして開発されましたが，移植が容易であったこと，CP/Mをベースとするアプリケーションプログラムの互換性が良かったことなどから急速に普及し，その後開発されたZ80，8085CPUでも使用できたことから非常にたくさんのアプリケーションプログラム（数千といわれる）が開発され有効に利用されています。

しかし，CP/Mは多くのハードウェアに使用することを前提としているため，基本的にグラフィックや音声等の機能はなく，最近のパソコンで強化されているこれらの機能を使ったプログラムは，他の機種には使えません。

## 3 turbo CP/M の利用方法

CP/M 上で走るソフトウェアとしては，各種高級言語，簡易言語，ソフトウェア開発ツール，ユーティリティソフトなど多くのアプリケーションソフトがあります。

例えば，BASIC をはじめとする FORTRAN，COBOL，PL/I，PASCAL，LISP，FORTH などの高級言語を CP/M 上で走らせることができます。シャープからも CP/M システム上で走らせることのできる初心者向けの言語シリーズを提供しています。又，様々な応用プログラムを作成することも可能です。この CP/M パッケージの中には，スクリーンエディタ（Word Master），8080アセンブラ，8080デバッガのソフトウェア開発ツールを備えていますので，8080アセンブラによるソフトウェアの開発ならば，パッケージのみで行なっていただけます。Z80用アセンブラも市販されておりますので，（マイクロソフト社の MACRO-80などがあります）Z80用ソフトウェア開発も行なえます。

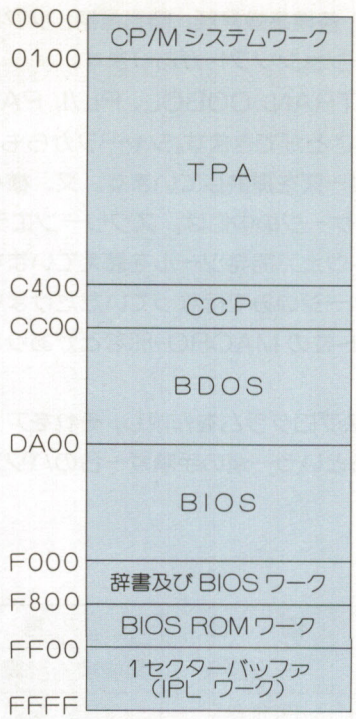
CP/M を使用すると，エディタでソースプログラムを作成し，それをアセンブルまたはコンパイルしてオブジェクトとし，デバッグもするという一連の手順が一台のパソコン上でできるため，大変便利な開発ツールといえます。

X1 ランゲージシリーズ

| 言 語 名 | 品 番 | 価 格 | 言 語 の 特 長 |
|---|---|---|---|
| FORTRAN | CZ-115LF | 13,800円 | 科学技術計算に適した言語 |
| C | CZ-116LF | 13,800円 | システム記述言語 |
| COBOL | CZ-118LF | 13,800円 | 事務処理に適した言語 |
| PROLOG | CZ-119LF | 13,800円 | 述語論理にもとずく人工知能言語 |
| LISP | CZ-120LF | 13,800円 | リスト処理のできる人工知能言語 |
| FORTH | CZ-121LF | 13,800円 | 自己増殖型，スタックオリエンテッド型言語 |
| PASCAL | CZ-125LF | 13,800円 | 教育用構造型プログラミング言語 |
| APL | CZ-126LF | 13,800円 | 表計算向き対話型言語 |

# 4 turbo CP/M の構成

## CP/M のメモリー構成




(56K CP/M の場合)

CP/M のシステムは次のように構成されています。

**①CP/M システムワーク**

CP/M システムが必要とするワークエリア

**②TPA (Transient Program Area)**

ディスク上のプログラムが入るところで，メモリーのフリーエリアです。

**③CCP (Console Command Processor)**

コンソール(キーボード)により入力されたコマンドを解析し，そのコマンドに対応した処理をするところです。

**④BDOS (Basic Disk Operating System)**

CP/M システムのディスクのファイルを管理するところで、CP/M システムの中心部といえます。

**⑤BIOS (Basic Input Output System)**

CP/M マシンの入出力装置の制御を行なう部分で，CP/M システムで唯一，ハードウエアに依存するところです。

⑥辞書及びBIOSワーク

辞書及びCP/MのBIOSを使用する際に必要なエリア

⑦BIOS ROMワーク

X1 turboのBIOS ROMを使用する際必要なエリア

⑧1セクターバッファ(IPLワーク)

ディスクの読み書きに必要なエリア

※ BIOSとBIOS ROMの違い

BIOS……CP/Mソフトウェアの一部で，CP/Mの入出力制御を行なっているもの。

BIOS ROM ……X1 turboのハードウェアの一部で，基本入出力ルーチンが入ったROM（Read Only Memory）

# 第2章

# turbo CP/Mを使用するにあたって

# 1 turbo CP/M の起動方法

turbo CP/M の起動方法は DISK BASIC の起動と同じです。順番に説明すると次のようになります。

①X 1 turbo の本体とディスプレイが正しく接続されているか確認し，まずディスプレイの電源を入れます。

②ディスクドライブのドライブ 0 に turbo CP/M のマスターディスクをセットし，コンピュータ本体の電源を入れます。

③IPL がスタートし，ドライブ 0 のアクセスが始まります。

④しばらくすると，画面上に次のようなオープニングメッセージが表示されます。

```
56K CP/M  Version 2.2 for SHARP X1 turbo [Rel. 1.0]
          Copyright (C) by Digital Research
```

⑤次に，自動的にSTARTUPプログラムが起動し，画面上に

A>■←カーソル
↑
———プロンプト

と，プロンプト（コマンド入力を受けつけるという意味のマーク）が表示され，turbo CP/M の起動が完了します。

**ログインディスク**

プロンプトが A>と表示されていることから，ログインディスクがドライブ A であることがわかります。ログインディスクとはドライブ指定を省略したときファイルの読み書きが行なわれるドライブのことです。起動時は必ず A に設定されています。

## 起動時のドライブ内容

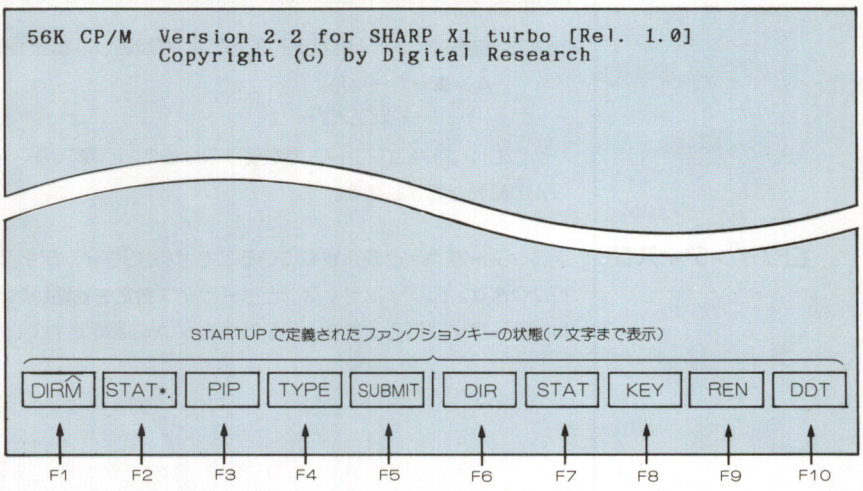
turbo CP/M 起動時のドライブ内容は下記のとおりです。

| ドライブ名 | 起動時の割り付け |
|---|---|
| A： | フロッピーディスクドライブ（2D）0 |
| B： | 〃 1 |
| C： | 〃 2 |
| D： | 〃 3 |
| E： | 外部メモリ（EMM0） |
| F： | 外部メモリ（EMM1） |
| G： | グラフィック RAM（MEM0） |
| H： | ユーザー指定のドライブ |

以後，特に指定のない限り，ドライブ名としてA:～H:を使用します。

## turbo CP/M 起動時の画面

turbo CP/Mを起動させ，STARTUPプログラムが自動的に走った後の画面は下のようになります。




ファンクションキーは全部で10個であり，起動時はそれぞれに上図のような内容が定義されています。[F6]～[F10]は[SHIFT]キーと[F1]～[F5]キーと組み合せて使います。たとえば[F6]は[SHIFT]キーを押しながら[F1]キーを押します。

[F2]キーは"STAT *.*"と表示されていますが，実際には"STAT *.* *^M"(^Mは[↵]と同じ動作をします）と登録されています。

## 2 マスターディスクのバックアップコピー

turbo CP/M をご使用になる前に，まずマスターディスクのバックアップコピーを作成します。大切なマスターディスクを事故により壊さないように，日常はバックアップしたサブマスターディスクを使用し，マスターディスクは安全な場所に保管しておいてください。
特に，ここでご使用になるFORMATコマンド，COPYコマンドはフロッピーディスクの内容を消去し，書き込みをする働きのコマンドですので，重要なフロッピーディスクの内容を誤って消去してしまうことのないように十分注意して下さい。

### ■ディスクドライブが 2 基の場合

①ドライブAにマスターディスクをセットし，turbo CP/M を起動します。この時，マスターディスクにはライトプロテクトシールを貼ったままにしておいてください。

②ドライブBにバックアップを作成するフロッピーディスクを入れます。このフロッピーディスクにはライトプロテクトシールを貼らないように注意してください。

③FORMAT（スペース）B：（リターン）と入力します。

④フォーマットユーティリティが起動し，

```
デバイスタイプは 5 inch 2D（320K），デバイス番号は 1 です。
フォーマットを開始しますか（Yes/No）？
```

と，メッセージがでますので，確認の上 Y と入力してください。

⑤フォーマットが開始されます。フォーマット中はトラック番号とサイド番号が表示され，フォーマットの経過がわかるようになっています。

⑥つづいてベリファイ（正しくフォーマットされたか検証する）が行なわれます。ベリファイ中はトラック番号が表示されています。

⑦フォーマットとベリファイが終了し，

```
－－－－終了しました－－－－
フォーマットを続けますか（Yes/No）？
```

とメッセージがでたら，N と入力します。画面にプロンプトがあらわれます。
（フォーマットを続けるときは Y と入力してください。）

⑧次に，COPY ↵と入力します。

⑨コピーユーティリティが起動し，

```
ソースドライブ名                    を入力してください ?
（使用可能なドライブ名はA～Hです。）
```

とたずねてきますので，[A] と指定します。

⑩つづいて，

```
ディスティネーションドライブ名を入力してください ?
（使用可能なドライブ名はA～Hです。）
```

とたずねてきますので，[B] と指定します。

⑪

```
ドライブ A から  ドライブ B へコピーします。

コピーを開始しますか（Yes/No）?
```

とメッセージがでますので，ドライブ名を確認し[Y]と入力します。

⑫コピーが開始されます。コピーの作業は，読み込み，書き込み，ベリファイと2トラック分ずつ行ないます。コピー中はトラック番号が表示されています。

⑬しばらくするとコピーが終了し，

```
－－－－終了しました －－－－

コピーを続けますか（Yes/No）?
```

とメッセージが表示されますので，[N] と入力します。画面にプロンプトが現われます。これでマスターディスクのコピーが完了しました。
（続けてもう1枚コピーするときは[Y]と入力してください。）

※なお，FORMAT，COPY コマンドについては第7章の2，3項でそれぞれ詳しく説明していますので参照してください。

〈ご注意〉

- 作成するサブマスターディスクは5枚以内とし，必ずお客様個人のみで使用してください。
- 作業中にエラーが出たら，まずフロッピーディスクの不良が考えられます。別のフロッピーディスクを用意して，もう1度初めから作業し直してください。
- サブマスターディスクなどを使用して，実際にCP/Mシステムをご利用になる場合は，ライトプロテクシールを貼らないように注意してください。

## ■ディスクドライブが１基の場合

①ディスクドライブにマスターディスクを入れ，turbo CP/M を起動します。
この時，マスターディスクにはライトプロテクトシールを貼ったままにしておいてください。

②FORMAT␣A：↵（スペース，リターン）と入力します。

③フォーマットユーティリティが起動し，次のようなメッセージがでます。

```
デバイスタイプは ５ｉｎｃｈ ２Ｄ（３２０Ｋ），デバイス番号は ０ です。
フォーマットを開始しますか（Yes/No）？
```

④マスターディスクをディスクドライブからぬいて，かわりにバックアップを作成するフロッピーディスクを入れ，[Y] と入力します。
このフロッピーディスクには，ライトプロテクトシールを貼らないように注意してください。

⑤フォーマットが開始されます。フォーマット中はトラック番号とサイド番号が表示され，フォーマットの経過がわかるようになっています。

⑥つづいてベリファイ（正しくフォーマットされたか検証する）が行なわれます。ベリファイ中はトラック番号が表示されています。

⑦フォーマットとベリファイが終了し，

```
－－－－終了しました－－－－
フォーマットを続けますか（Yes/No）？
```

とメッセージが表示されたら，[N] と入力します。画面にプロンプトがあらわれます。
（フォーマットを続けるときは[Y]と入力してください。）

⑧次に，マスターディスクをディスクドライブへ入れ，COPY↵と入力します。

⑨コピーユーティリティが起動し，

```
ソースドライブ名                    を入力してください？
（使用可能なドライブ名はＡ～Ｈです。）
```

とたずねてきますので，[A] と指定します。

⑩つづいて

```
ディスティネーションドライブ名を入力してください？
（使用可能なドライブ名はＡ～Ｈです。）
```

とたずねてきますので，こちらも[A] と指定します。

⑪

```
ドライブ A から  ドライブ A へコピーします。
コピーを開始しますか (Yes/No) ?
```

とメッセージがでますので，ドライブ名を確認し Y と入力します。

⑫

```
ソースディスク                    をドライブAにセットしリターンキーを押してください
```

とメッセージがでたら，↵キーを押します。するとドライブがアクセスされ，マスターディスクの内容が読み込まれます。全部を1度に処理することはできないため，2トラック分ずつ読み込みます。

⑬

```
ディスティネーションディスクをドライブAにセットしリターンキーを押してください
```

とメッセージがでたら，マスターディスクをドライブからぬき，バックアップを作成するために先程フォーマットしたフロッピーディスクを入れて，↵キーを押します。ドライブがアクセスされ，⑫でマスターディスクから読み込んだ分が書き込まれます。

⑭ ⑫と⑬の作業をあと19回繰り返して行ないます。トラック番号38～39の処理で終了するとオールコピー完了です。

⑮ コピーが終了し，

```
－－－－終了しましたー－－－
コピーを続けますか (Yes/No) ?
```

とメッセージがでたら，N と入力します。画面にプロンプトが現われます。これでマスターディスクのコピーが完了しました。(続けてもう1枚コピーするときは Y と入力してください。)

※コピーは，2トラックを1単位として行ないます。そのためディスク1枚40トラック（5インチ2D）をコピーするには，20回読み込みを繰り返します。

※なお，FORMAT，COPY コマンドについては第7章の2，3項でそれぞれ詳しく説明していますので参照してください。

〈ご注意〉

- 作成するサブマスターディスクは5枚以内とし，必ずお客様個人のみで使用してください。
- 作業中にエラーが出たら，まずフロッピーディスクの不良が考えられます。別のフロッピーディスクを用意して，もう1度初めから作業し直してください。
- サブマスターディスクなどを使用して，実際にCP/Mシステムをご利用になる場合は，ライトプロテクシールを貼らないように注意してください。

## 3 マスターディスクに含まれる各種ファイル

**turbo CP/M のマスターディスクには，各種のプログラムファイル，ソースファイル，音訓辞書ファイルなどが含まれています。**

マスターディスクのディレクトリ(ファイル一覧)を表示させてみましょう。
マスターディスクをドライブAにセットし，DIR↵と入力すると下のような画面があらわれます。

```
A>DIR ↵
A: MOVCPM   COM : CPM56    SYS : STARTUP  COM : STAT     COM
A: PIP      COM : ED       COM : ASM      COM : LOAD     COM
A: DDT      COM : DUMP     COM : DUMP     ASM : SUBMIT   COM
A: XSUB     COM : FORMAT   COM : COPY     COM : SYSGEN   COM
A: DICCOPY  COM : DICSET   COM : DISKSET  COM : HDMAINT  COM
A: HDBACKUP COM : HDOFF    COM : TERM     COM : CMT      COM
A: KEY      COM : TIME     COM : ULPT0    ASM : ULPT1    ASM
A: ULPT2    ASM : BIOS     MAC : WM       COM : WM       HLP
A: WMENGL   HLP : WMKANA   HLP : WMKANJI  HLP : ONKUN    DIC
A>
```

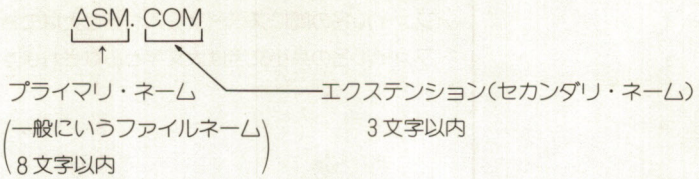
マスターディスクの中の各ファイルの内容を説明します。

| | ファイル名 | 内　　容 |
|---|---|---|
| トランジェント・コマンド | MOVCPM. COM | CP/M サイズを変更するためのプログラム |
| | STAT. COM | ファイル・周辺装置の設定と状態確認プログラム |
| | PIP. COM | 周辺装置とのデータ転送プログラム |
| | ED. COM | ASCII ファイルの作成，編集をするプログラム |
| | ASM. COM | 8080用アセンブラプログラム |
| | LOAD. COM | インテル HEX 形式のオブジェクトファイルを実行可能なトランジェント・コマンド・ファイルに変換，ディスク上にセーブするプログラム |
| | DDT. COM | 8080用デバッガプログラム |
| | DUMP. COM | ファイルの内容を16進数で表示するプログラム |
| | SUBMIT. COM | バッチ処理プログラム |
| | XSUB. COM | SUBMIT 拡張用プログラム |

| | | |
|---|---|---|
| turbo CP/M 強化コマンド | STARTUP. COM | 各種パラメータの設定および turbo CP/M 起動時の初期設定プログラム |
| | FORMAT. COM | フロッピーディスクのフォーマットを行なうプログラム |
| | COPY. COM | フロッピーディスクの内容を別のフロッピーディスクへオールコピーするプログラム |
| | SYSGEN. COM | CP/M システムのコピーを行なうプログラム |
| | DICCOPY. COM | 日本語処理用の辞書をコピーするプログラム |
| | DICSET. COM | 日本語処理用の辞書を使用可能な状態に準備するプログラム |
| | DISKSET. COM | 使用しているドライブの確認と指定をするプログラム |
| | HDMAINT. COM | ハードディスクの領域確保・解除を行なうプログラム |
| | HDBACKUP. COM | ハードディスクとフロッピーディスクとの間でファイル転送をするプログラム |
| | HDOFF. COM | ハードディスクの使用を終了するためのプログラム |
| | TERM. COM | ターミナルモード，リモートモードでの通信を行なうプログラム |
| | CMT. COM | ディスクとカセットとの間でファイル転送をするプログラム |
| | KEY. COM | ファンクションキーの設定，内容表示プログラム |
| | TIME. COM | 日付・時刻の設定，内容表示プログラム |
| その他・参考プログラム | CPM56. SYS | CP/M システムプログラム |
| | DUMP. ASM | DUMP. COM のソースプログラム |
| | ULPT0. ASM | プリンタ設定用ソースプログラム |
| | ULPT1. ASM | プリンタ設定用ソースプログラム |
| | ULPT2. ASM | プリンタ設定用ソースプログラム |
| | BIOS. MAC | BIOS のソースプログラム |
| | WM. COM | ワードマスター・プログラム（テキストファイルの作成，編集を行なう） |
| | WM. HLP | ワードマスターのヘルプメニューファイル |
| | WMENGL. HLP | ワードマスターの英文ヘルプメニューファイル |
| | WMKANA. HLP | ワードマスターのカナ文字ヘルプメニューファイル |
| | WMKANJI. HLP | ワードマスターの漢字ヘルプメニューファイル |
| | ONKUN. DIC | 音訓辞書 |

## ■ファイル名について

ファイル名は，プライマリ・ネームとエクステンション（セカンダリ・ネーム）から成っています。次に例を示します。（ファイル名に漢字を使用することはできません）




プライマリ・ネームがファイルの名前を示し，エクステンションがファイルのタイプを示します。

さて，その〝ASM〟や〝COM〟などのエクステンションには特別な意味を持つものがあります。その意味を下の表に示します。

ファイル名のエクステンション一覧表

| エクステンション | 各エクステンションの付いたファイルの意味 |
|---|---|
| ASM | アセンブル・ソース・ファイル（8080用アセンブラであるASM がアセンブルすることができるファイル） |
| COM | トランジェント・コマンド・ファイル（即実行できるファイル）。LOADコマンドで，HEX ファイルを読み込むことにより作られます。 |
| HEX | インテルHEX形式のオブジェクトファイル。ソース・ファイルをアセンブルすることにより生成されます。 |
| BAK | バックアップ・ファイル。エディタにより，オリジナル・ファイルを保存するため自動的に作られます。 |
| PRN | ソース・ファイルをアセンブルした後のリスティング・ファイル。アセンブル後，自動的に作られます。 |
| HLP | トランジェント・コマンドに関するヘルプメニュー |
| SUB | バッチ処理プログラム（SUBMIT プログラム＝作業手順を設定し自動作業をさせる）用にユーザーが作成。 |
| $$$ | テンポラリ・ファイル。エディタ実行時などに，一時生成されます。 |

この他にも，各種プログラムによって，さまざまなエクステンションが使用されています。

〈ご注意〉

・プライマリ・ネームは英数字１～８文字，エクステンションは英数字０～３文字で構成します。また，〝.〟はプライマリ・ネームとエクステンションを区別する記号です。

・ファイル名に，カナ文字，漢字，および次の特殊記号は使用できません。

〈 〉 . , ; : = ? * [ ]

・ファイル名の間にスペースを入れることはできません。

・ファイル名の英小文字は大文字とみなされます。

## 4 作業用マスターディスクの作成

turbo CP/M のマスターディスクには，多くのファイルが含まれていますが，その中には普段はあまり使わない参考プログラムファイルなどもあります。そこで，バックアップコピー作成の際には，マスターディスクと全く同じ内容のサブマスターディスクを数枚作っておくのと同時に，実際にプログラミングする際に不必要と思われるファイルを除いて，頻繁に使用するファイルだけをコピーした作業用のマスターディスクを作っておくと便利です。

■作業用マスターディスク作成手順

（この作成手順はディスクドライブ 2 基の場合に限ります。）

例として，エクステンションがCOMとHLP及びDICのファイルのみもっている作業用マスターディスクを作成します。

①作業用マスターディスクを作成するフロッピーディスクをフォーマットします。フォーマットの方法については本章の2項を参照してください。

②ドライブAにマスターディスク，ドライブBにフォーマット済みのディスクをセットし，SYSGEN ↵と入力します。SYSGEN コマンドが起動しますので，画面にでるメッセージに従って，CP/M システムをコピーします。(SYSGENコマンドについては第7章の4項を参照してください。)

③PIP␣B：＝＊.COM ↵と入力します。これで .COM とエクステンションのついたファイルが全てドライブBへ転送されます。(PIPコマンドについては第6章の2項を参照してください。)

④コピー中です。全部をコピーし終えるとプロンプトがでて，コマンド待ちとなります。

⑤PIP␣B：＝WM. HLP ↵と入力します。これでワードマスターのヘルプメニューファイルがドライブBへ転送されます。

```
A>SYSGEN↵ ————————②

A>PIP B:=*.COM↵——③

COPYING - ————————④
MOVCPM.COM
STARTUP.COM
STAT.COM
PIP.COM
ED.COM
ASM.COM
LOAD.COM
DDT.COM
DUMP.COM
SUBMIT.COM
XSUB.COM
FORMAT.COM
COPY.COM
SYSGEN.COM
DICCOPY.COM
DICSET.COM
DISKSET.COM
HDMAINT.COM
HDBACKUP.COM
HDOFF.COM
TERM.COM
CMT.COM
KEY.COM
TIME.COM
WM.COM

A>PIP B:=WM.HLP↵——⑤
```

```
A>DICCOPY↵
```

⑦

```
A>ERA B:HDMAINT.COM↵
A>ERA B:HDBACKUP.COM↵
A>ERA B:HDOFF.COM↵
A>ERA B:CMT.COM↵
```

⑧

⑦DICCOPY↵と入力します。
この後の処理は7章5項を参照ください。

⑧次にERAコマンドを使用して，不必要と思われるファイルを削除します。
（ERAコマンドについては第5章の4項を参照してください。）

⑨DIR␣B：↵と入力し，いま作成した作業用マスターディスクの内容を確認してみます。

```
A>DIR B:↵
B: MOVCPM   COM : STARTUP  COM : STAT     COM : PIP      COM
B: ED       COM : ASM      COM : LOAD     COM : DDT      COM
B: DUMP     COM : SUBMIT   COM : XSUB     COM : FORMAT   COM
B: COPY     COM : SYSGEN   COM : DICCOPY  COM : DICSET   COM
B: DISKSET  COM : TERM     COM : KEY      COM : TIME     COM
B: WM       COM : WM       HLP : ONKUN    DIC
```

※ここでいう必要なファイルというのは，あくまでも一例であり絶対的なものではありません。たとえば，使用するドライブの種類，アプリケーションの種類などにより，必要なファイルは変わります。臨機応変に使い分けるようにしてください。

## 5 プリンタの設定

バックアップコピーができたら，実際にプログラムに入る前に，まず使用するプリンタの設定をしておきます。プリンタの設定は，STARTUPコマンドを使って行ないます。一度，使用するプリンタのデータをマスターディスクに登録しておけば，それ以後は，turbo　CP/M 起動時に自動的に設定されるようになります。

①マスターディスクをドライブAにセットし，STARTUP␣＊↵と入力します。

②STARTUP ユーティリティが起動し，メニュー画面がでますので，〝プリンタ設定〟を指定します。

③〝プリンタ設定〟の画面になったら，設定したい機種を選択します。

④プリンタ設定が終了し，メニュー画面へ戻ったら，[8]（CP/M へ戻る）を入力します。

⑤画面に，〝設定したデータをディスクに登録しますか（Yes/No）?〟とメッセージがでるので，[Y]　を押します。これで再起動した際には自動的にプリンタが設定されます。

※なお，プリンタ設定については，7章の1項を参照してください。

※また，実際にプリンタ出力を行なうには以下の方法があります。

・turbo CP/M 起動後，[CTRL]＋[P]を入力します。これ以後は，CRT に出力されるものが，同時にプリンタへも出力されます。

・PIP コマンドを使用する際に，論理デバイスであるLST：を指定すると，ファイルのリストなどを出力することができます。

## 使用可能プリンタ

turbo CP/M で使用可能なプリンタとその機能は下の表のとおりです。

| プリンタ　機種名 | 漢字印字のモード | ハードコピー |
|---|---|---|
| CZ-800P | ビットイメージ | ○ |
| CZ-80PK | ビットイメージ | ○ |
| CZ-8PD2/SP-80 | ビットイメージ | ○ |
| CZ-8PP2 | コード | × |
| CZ-8PK2 | コード | ○ |
| CZ-8PN1/CZ-8PK3(4) | コード | ○ |
| MZ-1P06 | ビットイメージ | ○ |
| MZ-1P07 | ビットイメージ | ○ |
| MZ-1P08 | ビットイメージ | ○ |
| MZ-1P10A | コード | × |
| ESC/P-ext | ビットイメージ | ○ |
| RP-80II | ビットイメージ | ○ |
| FP-80 | ビットイメージ | × |
| UP-130K | コード | ○ |
| PC-8822/PC-PR201 | コード | ○ |
| M-1009X | ビットイメージ | ○ |
| M-1009 | ビットイメージ | × |
| LPR-24T | コード | ○ |
| KP-3000 | ビットイメージ | ○ |

※MZ-IP17はCZ-8PN1のモードでご使用ください。

# 第3章

# turbo CP/Mの使い方

## 1 キーの使い方

turbo CP/M システムで使用できる特殊キーは，下記の通りです

| キー操作 | 内　　容 |
|---|---|
| HTAB | 8文字単位のタブ動作を行ないます。 |
| CAPS LOCK | 大文字，小文字の切り換えをします。<br>キーが押された状態で大文字入力，上がった状態で小文字入力となります。 |
| SHIFT | SHIFT を押しながらキー入力すると，CAP SLOCK キーと逆の状態で入力されます。又，同様に，F1 ～ F5 キーをおすと，F6 ～ F10 になります。 |
| カナ | カナ キーが押されている状態では，片仮名入力となります。この状態で，SHIFT を押しながら入力するとカナ入力，カナ記号入力となります。 |
| CTRL<br>（コントロールキー） | 他のキーと組み合わせて，コントロールコードを入力します。 |
| CTRL + C<br>または<br>SHIFT + BREAK | CP/M システムの CCP，BIOS 部の読み込みを再度実行します。（リブート）<br>コマンドラインの最初に入力します。 |
| CTRL + E | 行の終わりに入力します。カーソルは次の行の左端に移動します |
| CTRL + H<br>または DEL | カーソルの左側の文字が抹消され，右側の文字列が左へ移ります。 |
| CTRL + J<br>または<br>CTRL + M<br>または<br>↵ | 入力ラインの終了を表わしており，コマンドを実行します。 |
| CTRL + P<br>または<br>COPY | 現在割り当てられているリスト・デバイスに以後のコンソール出力をプリントアウトします。これは次の CTRL + P の入力があるまであるいはリブートするまで続けられます。 |
| CTRL + R<br>または<br>SHIFT + INS | 入力ラインの最後に "#" を表示した後，行を替え入力ラインを再びタイプアウトします。 |

| | |
|---|---|
| CTRL＋S<br>またはBREAK | コンソール出力を一時停止します。再び出力させるには，もう一度，CTRL＋Sを入力します。これにより，CRT 画面の出力内容を常に確認できます。 |
| CTRL＋U | 入力ラインの最後に "#" を表示した後，行を替え，入力ラインをすべて抹消します。 |
| CTRL＋X | カーソルを入力ラインの最初まで戻し，入力ラインをすべて抹消します。 |
| CTRL＋0～7<br>（キーボード） | 画面の背景色の指定をします。<br>1で青，2で赤、3でマゼンタ，4で緑，5でシアン，6で黄，7で白，0で黒。<br>＊この設定後，パレットも変更されるためグラフィック画面が表示されます（起動状態ではしま模様が表示されます）。 |
| CTRL＋0～7<br>（テンキー） | 文字の色を指定します。<br>1で青，2で赤，3でマゼンタ，4で緑，5でシアン，6で黄，7で白，0で黒。 |
| CTRL＋／<br>（テンキー） | キャラクタゼネレータのROM/RAMを切り換えます。 |
| CTRL＋＊<br>（テンキー） | 点滅モードとノーマルモードの切り換えを行ないます。 |
| CTRL＋－<br>（テンキー） | 反転モードとノーマルモードの切り換えを行ないます。 |
| SHIFT＋COPY | プリンタにテキスト画面のハードコピーを出力します。 |
| GRAPH＋COPY | プリンタにグラフィック画面のハードコピーを出力します。 |
| CTRL＋COPY | プリンタにテキストおよびグラフィック画面のハードコピーを出力します。 |
| CTRL＋＋<br>（テンキー） | コンソールステータスを正常に返すモードにし，キーバッファをクリアします。 |
| GRAPH＋SHIFT<br>＋CTRL＋F1 | リブートし，ログインディスクをドライブAにします。なお，F1は，最後に1秒間押します。 |

＊＋はキーを同時に押すことを表わしています。

## 2 日本語入力の方法

turbo CP/M では，X1turbo のもつ日本語処理機能をサポートしています。(日本語入力に関しては，BASIC と同じですので，BASIC のユーザーズマニュアルを参照してください。)
又，音訓辞書や，システム・ユーザー辞書のカナ/ローマ字入力による漢字変換もサポートしています。(システム・ユーザー辞書はオプション) なお，辞書をご使用の場合，CP/M フォーマットディスクへの辞書のコピー (DICCOPY)，辞書ドライブの設定 (DICSET) が必要となります。

**辞書コピーの方法**

① CP/M のフォーマットをしたディスクを用意します。

② DICCOPY コマンドを使って，コピーしたい辞書を指定し，コピーします。
詳細は，第 7 章の 5 項を参照してください。

**辞書ドライブの設定**

DICSET コマンドを使って，辞書の入っているドライブを設定します。
詳細は，第 7 章の 6 項を参照してください。

**〈ご注意〉**

・ログインディスク上に辞書がある場合でも，必ず DICSET でドライブ設定を行なってからご使用ください。
・「ワードパワー」および「レキシコン」はご使用になれませんのでご注意ください。

## 3 ビルトインコマンドとトランジェントコマンド

**CP/M のコマンドは，メモリに常駐するビルトインコマンドと，実行する際ディスクから呼び出されるトランジェントコマンドの 2 つに分けられます。**

**ビルトインコマンドは，turbo CP/M システムの内部，つまりメモリの CCP の部分に常に存在しており，ディスクから呼び出されることなく即実行されます。一方，トランジェントコマンドは，コマンドを実行するために，一度ディスクからプログラムを TPA の領域に読み込んで実行します。**

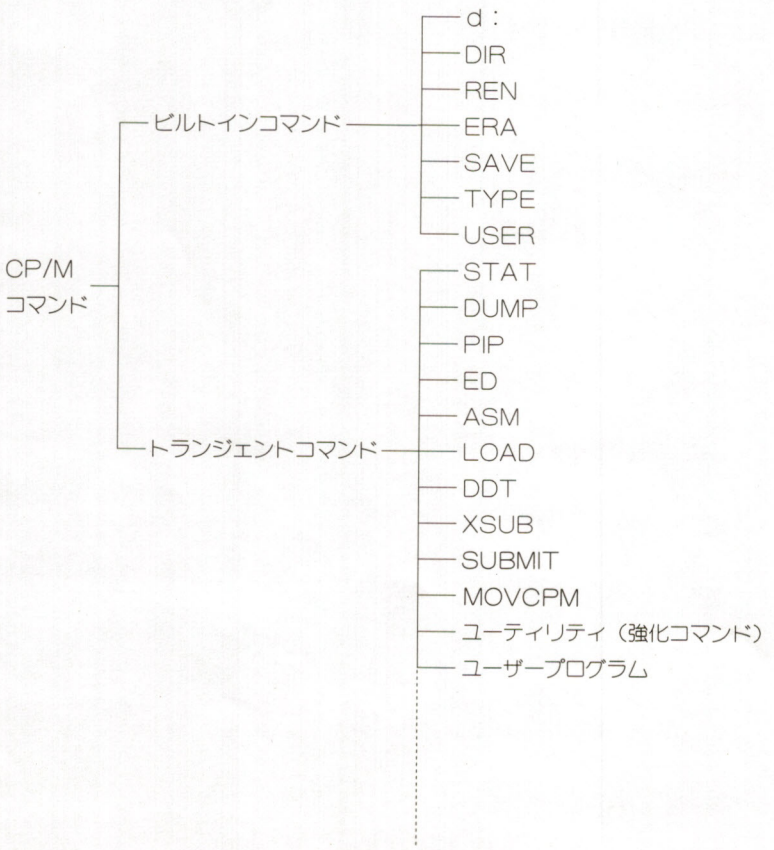
**ビルトインコマンド**

turbo CP/M には，7 つのビルトインコマンドがあります。これらのコマンドは，turbo CP/M の一部であり，どのドライブにどのディスクがセットされているかということとは無関係にいつでも使用することができます。ビルトインコマンドは，turbo CP/M の一部として，メモリに常駐しているため，トランジェントコマンドより速やかに実行することができます。

ビルトインコマンドの詳細については，第 5 章を参照してください。

**トランジェントコマンド**

turbo CP/M には，10個のトランジェントコマンドがあります。これらのコマンドはユーザーがそのコマンドを要求したときだけディスクからメモリーにロードされます。ですからこれらのコマンドは turbo CP/M のプログラムの領域を狭めないといった長所があります。しかしトランジェントコマンドを利用するためには，呼びだすためのディスクを指定しなければなりません。しかも必らずその指定ディスクに目的のコマンドファイルが存在しないと実行できないので注意が必要です。turbo CP/M のトランジェントコマンドの詳細については，第 6 章を参照してください。



## 4 マニュアル表記方法について

マニュアルにおけるコマンド表記の方法については以下のとおりです。

①大文字の英字で書かれたものは，その英字どおりに入力してください。入力に際しては，大文字で入力しても，小文字で入力してもかまいません。

②小文字の英字で書かれたものは，その意味を持つ言葉であることを示します。これらも入力に際しては，大文字で入力しても小文字で入力してもかまいません。

③小文字の英字で書かれたもののうち，特に頻繁にでる単語について説明します。

"d："はドライブ名で，省略できます。省略した場合にはログインディスクとなります(ログインドライブについては，5章を参照してください。）またドライブ名は，A～Hで表現されます。

"filename"はファイル名で，そのファイルの呼び名を最大8文字までの英数字で表現します。なお，漢字はご使用になれません。

".ext" はファイルタイプ（エクステンション）で，3文字までの英数字でファイルの種類を指定します。

ファイル名とファイルタイプには，以下の様な特殊文字を指定して，特定の条件を満足するすべてのファイルを一括して指定することができます。
"＊" ファイル名，ファイルタイプに使用し，ファイル名，ファイルタイプが何でもかまわないことを意味します。
"？" ファイル名，ファイルタイプの文字列の任意の場所の1文字の代わりに使い，"？"の位置にある文字が何であってもかまわないことを意味します。

特殊な表記をするキー操作については以下のとおりです。

"∧" コントロールキーを意味します。
"↵" リターンキー（CRキー）を意味します。
"␣" スペースキーを意味します。

# 第4章

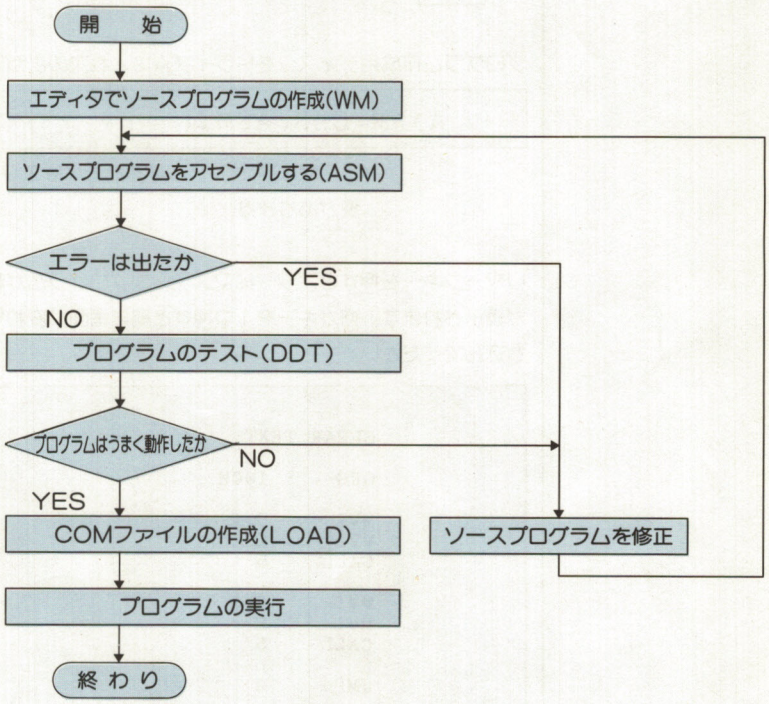
# turbo CP/Mによるプログラム作成

## 1 アセンブラによるプログラム作成手順

**CP/Mは，8080やZ80などをCPUにもつパーソナルコンピュータのためのディスクオペレーティングシステム(DOS) です。通常標準のCP/Mには，アセンブラプログラミングのためにED（エディタ），ASM[注1)]（アセンブラ），LOAD(ローダ），DDT（デバッガ）のプログラムが用意されていますが，turbo CP/Mの場合これに加えてEDよりもソースプログラムが作成しやすいマイクロプロ社のエディタソフトWM(ワードマスター)がついています。**

注1)ASMは8080アセンブラソースプログラムをアセンブルするためのもので，Z80用としてはマイクロソフト社のMACRO−80などを利用する必要があります。

以下では，WM，ASM，LOAD，DDTを用いた機械語プログラム作成の手順を説明します。




①エディタ　まず，エディタにより新たにソースプログラムを作成します。また，WMではすでにディスク上にセーブされているソースプログラムの修正や編集を行なうことも可能です。

②アセンブラ　次に，ASMを起動してソースプログラムを読み出しアセンブルします。アセンブルにより，インテルHEX形式（ソースプログラムと機械語プログラムの中間にあたるコードで表わされたもの）のファイルとリスト形式（メモリ・アドレスや機械語コード，エラーメッセージが表わされたもの）のプリントファイルが作成されます。

③デバッガ　アセンブラで作成したファイルをロードし，その修正，テストを行ないます。

④ローダ　アセンブラで作成されたインテルHEX形式のファイルを即実行可能な機械語ファイルに変換します。

それでは，テキスト画面をクリアさせる簡単なプログラムを例に実際の手順を説明します。

## 1. ソースプログラムの作成

ワードマスターによるエディット作業は，主にCTRLキーと他のキーを同時に押すことにより行なわれます。また，そのキー操作は非常に多くのものがサポートされていますが，とりあえず次の操作を覚えることでソースプログラムの作成が行なえます。

| | |
|---|---|
| CTRL + O | 文字挿入（インサート）モードの切換え |
| CTRL + Y | カーソル行の削除 |
| → ↑ ↓ ← | カーソル位置の移動 |
| DEL | 左の文字を削除 |
| TAB | カーソルを8文字右に移動 |

プログラム作成用ディスク[注2)]をドライブAに入れ次のようにタイプします。

```
A>WM CLS.ASM↵
```

※ASMでアセンブルするためにはエクステンションを"ASM"にする必要があります。

リターンキーを押すと，ワードマスターのプログラムが起動し，オープニングのメッセージが表示されます。何かキーを1つ押すと編集画面に変わりますので，以下のプログラムを打ち込んでください。

```
;
;       CLEAR TEXT
;
        ORG     100H
;
        MVI     C,2
        MVI     E,1BH
        CALL    5
;
        MVI     C,2
        MVI     E,*
        CALL    5
;
        JMP     0
```

打ち終わりましたら，ESCキーを押してください。画面の左下に＊(アスタリスク)が表示されワードマスターのコマンド待ちになります。この状態でEキーを押してリターンキーを押せば，ワードマスターのプログラムは終了しディスク上にソースファイルが作成されます。

間違ってタイプしていないかを確かめるために，次の様に TYPE コマンドを使用してリストをとってください。もし間違いがありましたら前記のようにワードマスターを再起動して修正を行なってください（作成したファイル名と同じものを指定すれば，その修正を行なうことができます）。

```
A>TYPE CLS.ASM ↵
```

注2)プログラム作成用ディスクに関しては，第2章4項を参照してください。

## 2. アセンブル

それでは，ワードマスターで作成した“CLS.ASM”というファイルを次のようにしてアセンブルします。

```
A>ASM CLS ↵          ―― ファイル名にエクステンションはつけない
CP/M ASSEMBLER - VER 2.0
E0109 1E00              MVI     E,*―― エラー行の表示
0111
000H USE FACTOR
END OF ASSEMBLY
```

MVI E,＊と書かれた行にエラーがあることを上記のメッセージは示しています。ワードマスターを再起動して次のようにこの行を変更し再度アセンブルしてください。

```
;
;       CLEAR TEXT
;
        ORG     100H
;
        MVI     C,2
        MVI     E,1BH
        CALL    5
;
        MVI     C,2
        MVI     E,'*'  ―――――――このように変更します
        CALL    5
;
        JMP     0
```

```
A>ASM CLS ↵          ―― 再度アセンブルします
CP/M ASSEMBLER - VER 2.0
0111
000H USE FACTOR
END OF ASSEMBLY
```

これでアセンブルが終了します。これによりエクステンションが“HEX”と“PRN”のファイルが作成されます。

```
A>DIR CLS.* ↵
A: CLS      ASM : CLS      PRN : CLS      HEX
```

## 3. プログラムの動作確認

BASIC などのインタープリタ言語プログラムと異なり機械語プログラムは，通常その実行を途中で止めることはできません。このため DDT のようなツールが必要となります。
DDT では L コマンドでプログラムリストを確認しながら G コマンドでプログラムを部分的に実行することができます。また，X コマンドによりレジスタの内容を表示することもできます。

```
A>DDT CLS.HEX ↵ ―――――――――― DDT を起動
DDT VERS 2.2
NEXT  PC
0111 0000
-L100,110 ↵ ―――――――――― L コマンドにより100H～110H 番地までを表示
  0100  MVI  C,02
  0102  MVI  E,1B
  0104  CALL 0005
  0107  MVI  C,02
  0109  MVI  E,2A
  010B  CALL 0005
  010E  JMP  0000
  0111
```

```
-X ↵ ―――――――――― レジスタ内容を表示
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0000 S=0100 P=0000 JMP  DA03
-G100,104 ―――――――――― 部分的に実行
*0104 ―――――――――― 104番地で実行中止
```

```
-X ↵
C0Z0M0E0I0 A=00 B=0002 D=001B H=0000 S=0100 P=0104 CALL 0005
```

レジスタ内容を表示すると B に02H，D に1BH が入っているのがわかります。

```
-G ↵ ―――――――――― 続きを実行
```

テキスト画面をクリアした後，プロンプトの状態に戻ります。
このように DDT を用いることにより，プログラムのバグ（プログラムの誤り）がどの部分にあるかを探し出すことができます。

## 4. マシンオブジェクトファイルの作成と実行

DDT により正しく動作することが確認できたら，LOAD コマンドを用いてマシンオブジェクトファイル（COM ファイル）を作成します。

```
A>LOAD CLS ↲ ―――――――――――――――― エクステンションはつけない
FIRST ADDRESS 0100
LAST  ADDRESS 0110
BYTES READ    0011
RECORDS WRITTEN 01
```

```
A>DIR CLS.* ↲
A: CLS      ASM : CLS      PRN : CLS      HEX : CLS      COM
```

これにより "COM" ファイルが作成されているのがわかります。

このファイルはプロンプトの後に直接入力することにより実行できます。

```
A>CLS ↲ ―――――――――――――――― エクステンションはつけない
```

テキスト画面がクリアされてプロンプトが表示されます。

## 2 アセンブラプログラムのフォーマット

アセンブリ言語（アセンブラでアセンブルできる言語）は，1ステートメントが次のフォーマットに従った形になっている必要があります。

| 行番号␣ラベル␣オペレーション␣オペランド；コメント↵ |
|---|
| 10　　turbo　EQU　　AA＊BB；LOOP |

※↵キーではなく"！"を使用することにより，1行にいくつものステートメントを書くこともできます。

**1. 行番号**

行番号は10進数で表わされます。turbo CP/M のアセンブラを使用する場合は，行番号を無視するために特につける必要はありません。

**2. ラベル**

ラベルは特定のステートメントにつけられる識別子で，ラベルにはその後につづくオペレーションによって次のような値が与えられています。

・EQU，SETが次にくる場合……オペランドによって与えられる値
・ニーモニックなどが次にくる場合……ラベルが付けられた所のアドレス値

※SET をのぞいて１つのステートメントには１つのラベルしか設定できず，他のステートメントに同じラベルを付けることはできません。

ラベルには次のような条件があります。

・英文字が先頭の英数字列で表わします（小文字は大文字としてみなされます）。
・16文字以内で表わします。
・読みやすくするために"$"を任意に入れることができます（アセンブラはこれを無視します）。
・ラベルとオペレーション間の区切りは "␣" の他 ":" で表わすことも可能です。

**3. オペレーション**

オペレーションフィールドは，8080ニーモニックコードあるいは，疑似命令によって構成されます。

・ニーモニックコード……付録のニーモニック一覧を参照してください。
・疑似命令……アセンブル作業の指定をするだけで，命令そのものは機械語に変換されません。疑似命令には次の９つがあります。

| | |
|---|---|
| ORG | プログラムのデータや開始番地の設定 |
| END | プログラムの終了。スタートアドレスの設定(オプション) |
| EQU | 数値の同値化 |
| SET | 数値のセット |
| IF | 条件付アセンブルの開始 |
| ENDIF | 条件付アセンブルの終了 |
| DB | データの定義(8ビット単位) |
| DW | データの定義(16ビット単位) |
| DS | メモリ領域(大きさ)の定義 |

次に各疑似命令について詳しく説明します。(なお，以下で用いられる式はオペランドにあたります。詳しくは次のオペランドの説明を参照してください。)

| 命　令 | 形　　式 | |
|---|---|---|
| ORG文 | ラベル ORG 式 | ここでラベルは任意であり，式は16ビットで評価され，このORG文の前に定義されているオペランドからなっています。<br>アセンブルでは，式で評価された値の番地から機械語に変換を始めます。また，1つのプログラムで，いくつものORG文を使用してもかまいませんが，メモリ領域が重なる場合のチェックは行ないませんので注意が必要です。<br>なお，CP/Mシステムの場合TPAが100Hから始まるため，ほとんどのプログラムが次のように書かれます。<br>ORG　100H |
| END文 | ①<br>ラベル END<br><br>②<br>ラベル END 式 | END文はあってもなくてもかまいませんが，もしEND文が出てきたときは，そのステートメントが最後のステートメントとなります。<br>①の形式でEND文を設定すると，そのステートメントでアセンブルをストップし，プログラムに与えられるスタートアドレスは0000Hとなります。<br>②の形式で設定すると，式の値がスタートアドレスの値となります。<br>END文ではラベルは任意です。<br>なお，CP/Mアセンブリ言語のプログラムでは，ほとんどの場合次のようになります。<br>END　100H |
| EQU文 | ラベル EQU 式 | ラベルは必要で，このラベルを他のステートメントに付けることはできません。EQU文は，ラベルに数値を割り当てる命令で，このステートメント以後は，その数値をラベルで表わすことができます。 |
| SET文 | ラベル SET 式 | EQU文と同様にラベルに数値を割り当てる命令です。ここでもラベルは必要です。SET文の場合は，同じラベルをプログラム中の他のSET文で用いることも可能です。したがって，EQU文の場合にはラベルに単一の値しか割り当てられませんが，SET文では1つのラベルに別の値を何度も割り当てることができます。ラベルに割り当てられた値は，次のSET文でそのラベルに別の値が割り当てられるまで有効です。 |

| 命　令 | 形　　式 | |
|---|---|---|
| IF文<br><br><br><br>ENDIF文 | IF　式<br>ステートメント#1<br>ステートメント#2<br>⋮<br>ステートメント#n<br>ENDIF | IF，ENDIFはIFとENDIFの間のステートメントを式の値により，アセンブルするかどうかを選択するものです。アセンブラは，IF文において式の値を求めます。その値が0以外であれば，ステートメント#1からステートメント#nまでをアセンブルし，0であれば，プリントファイルにリストはしますがアセンブルはしません。 |
| DB文 | ラベル DB e#1,<br>e#2,……,e#n | メモリ内容を1バイト単位で定義する場合に用います。ここではラベルは任意に付けることができます。e#1，e#2，……e#nは，8ビットとみなせる式の値(16ビットでも上位8ビットが0)か，64文字を越えない長さのアスキー文字です(1つの行にあれば，e#nの数に特別の制限はありません)。式の値は，アセンブルされたプログラムアドレスに続けて，HEXファイルの順に記録されます。文字も同様に最初から最後の文字まで順に記録されます。通常，他のプログラム部分から，ラベルによりこのデータ領域を参照します。<br>例<br>soft：DB　0，1，2，3<br>：DB　'I　love　HAL'，3+2<br>：DB　DATA，TEL　AND　77H |
| DW文 | ラベル DW e#1,<br>e#2,……,e#n | メモリ内容を2バイト単位で定義する場合に用います。ここではラベルは任意につけることができます。e#1，e#2，……，e#nは16ビットに評価される式の値か，2文字までのアスキー文字です。データは式の下位バイトがメモリの前に格納され，その後に上位バイトが格納されます。<br>例<br>Look　DW　'FU'，'ji'<br>DW　1986H,sign |
| DS文 | ラベル DS 式 | DS文は定義されないメモリ領域を確保するためのものです。アセンブラは，このステートメントのときのアドレスに式の値を加えたアドレスから次のステートメントを変換します。ラベルは任意につけることができます。 |

## 4. オペランド

オペランドフィールドは，定数，ラベル，算術式，論理式によって構成されます。式の計算はアセンブルの過程でアセンブラ自身によって行なわれます。その計算結果は16ビットの値ですが，オペレーションによっては1バイトのオペランド形式になっているものがあり，その場合，式の上位8ビットは0になる必要があります。

### [1] 定数（数値）

定数は16ビットの値で，その値が何進数であるかは，値の最後につける以下のアルファベット（radix indicator）で判断します。値はそれぞれの進数で使用できる数字でなければなりません。

B　2進数（使用できる数字　0，1）
O　8進数（使用できる数字　0～7）
Q　8進数（使用できる数字　0～7）
D　10進数（使用できる数字　0～9）
H　16進数（使用できる数字　0～9，A～F：先頭の文字は0～9になるようにします。このとき16ビットを越える部分は無視されます。）

OとQは同じく8進数を表現します。これはOが数字の0と間違いやすいために用いられたためです。また，これらのアルファベットが付いていない数字は，すべて10進数として判断されます。この他，"$"を見やすいように定数の中に入れてもかまいません。

例

| | | |
|---|---|---|
| 123 | 1111$0111$1010B | 228O |
| 1FEH | 324D | 77$44$62Q |
| 214Q | 0A3EH | 32bd |

また，既にその値や意味が定義されている登録語が次のように存在します。

・8080レジスタ名には次のような値がすでに定義されています。

| （レジスタ名） | （定義された値） |
|---|---|
| A | 7 |
| B | 0 |
| C | 1 |
| D | 2 |
| E | 3 |
| H | 4 |
| L | 5 |
| M | 6 |
| SP | 6 |
| PSW | 6 |

・ニーモニック・コードには各々の機械語の値が定義されています。

例

| | |
|---|---|
| CMA | 2FH |
| MOV | 40H |
| DCR | 05H |

※MOVなどのようにオペランドを含めて機械語になるものは，オペランドが0になった場合の機械語の値が定義されています。

・"$"がオペランドで単独に用いられる場合，そのときのアセンブルアドレスが定義されます。

例

```
place    JMP  $
```

というステートメントの場合　place＝$となります。

### [2] 定数（文字）

定数の中には前記のように数値を表わすものの他に，文字を表わす文字定数があります。文字定数は，アポストロフィー（'）で囲まれた，文字長64字以内のものです。文字定数にアポストロフィーを使用する場合には，アポストロフィーを2度続けて入力することにより表わします。

またDB疑似命令以外は，1文字か2文字の文字長となります。文字定数の中では，小文字は大文字に変換されませんので，区別して使用することができます。

例

```
'We are loving'    'for peace'
'ABC'    'abc'
```

### [3] ラベル

オペランドフィールドにラベルがある場合，アセンブラによりその値はラベルの値に置き換えられます。また，ラベルを他の定数，式と組み合わせて用いることもできます。

例

```
TOP    ENQ    FFH
       MVI    A，TOP−21H
```

### [4] 算術式，論理式

オペランドフィールドでは，次の算術および論理演算子を用いて式を表わすことができます。

| | |
|---|---|
| a＋b | 符号なしの加算 |
| a−b | 符号なしの減算 |
| ＋b | プラス（bと同じ） |
| −b | マイナス（0−bと同じ） |
| a＊b | 符号なしの乗算 |
| a／b | 符号なしの除算 |
| a MOD b | a／bの剰余 |
| NOT b | bのインバート（すべての0が1に、1が0になります。また，bは16ビットとして扱われます） |
| a AND b | 論理積 |
| a OR b | 論理和 |
| a XOR b | 排他的論理和 |
| a SHL b | bで示す数だけaを左にシフトし，残りに0を挿入 |
| a SHR b | bで示す数だけaを右にシフトし，残りの0を挿入 |

※aとbは単純なオペランド（ラベル，数値定数，1文字か2文字の文字定数）か，カッコで囲まれた式です。

また演算子には，相対的な優先順位が与えられており，カッコを書かなくても良いようになっています。したがって式は，その優先順位によって計算が行われます。また，カッコで囲まれた式は，優先順位を無視し，カッコ内の式を計算します（カッコ内では優先順位にしたがって計算されます）。

演算子の優先順位は以下のようになっています。左の数字が優先順位を示し，数の少ない方が優先順位が高くなっています。また，同じ優先順位の演算子が同じ式内にある場合には，右から左に計算を行ないます。

1．*，1，MOD，SHL，SHR
2．－，＋
3．NOT
4．AND
5．OR，XOR

## 5．コメント

“；”につづいて書かれた文字はコメントとして扱われます。コメントはプログラムを見やすくするためにつけるもので，機械語には変換されません。また，コメントはすべてのステートメントにつける必要はありません。

# 第5章

# ビルトインコマンド

## 1 d：（ログインディスク変更コマンド）

〈コマンド形式・機能〉

d：

ログインディスクの切換えを行ないます。
dはA～Hの英記号

**ログインディスクとは，プログラム上でCPUとつながっているディスク装置のことで，CP/Mユーザーが特にディスク装置を指定しないかぎりファイルの処理はログインディスク上で行なわれます。**

・ログインディスクはCP/Mシステムが入力待ちのときのプロンプトでわかります。
たとえばA>と表示されているなら，ログインディスクはドライブAということになります。

・ログインディスクの切換えは，ディスクドライブ名（A，B……H）のすぐ後にコロン：をつけてリターンキーを入力して行ないます。

A>B：↵ ――――ログインディスクをドライブAからドライブBに切換えます。
B>A：↵ ――――ログインディスクをドライブBからドライブAに切換えます。
A>

**使用できる物理デバイスとディスクタイプ**

ドライブのA～Hにはそれぞれ下記のようなディスクタイプを割付けることができます。割付けの方法については7章の7項を参照してください。

| 物理デバイス名 | デバイスタイプ |
|---|---|
| 5インチフロッピーディスク | 両面倍密度（2D）<br>両面倍密度倍トラック（2DD）<br>両面高密度（2HD） |
| 8インチフロッピーディスク | 両面倍密度（2D-256）<br>片面単密度（1S-128） |
| 5インチハードディスク | 10Mバイト |
| グラフィックRAM | 48Kバイト（MEM0，MEM1） |
| 外部RAM | 320Kバイト外部RAM EMM0～EMM9 |
| ユーザー指定のドライブ※ | |

※）ユーザー指定のドライブについては7章の1項を参照してください。

## ドライブ内容の初期設定

ドライブ内容の初期設定は次のとおりです。

| ドライブ名 | 物理デバイス名 | 備　考 |
|---|---|---|
| A： | フロッピーディスクドライブ(2D)　0 | |
| B： | 〃　1 | |
| C： | 〃　2 | |
| D： | 〃　3 | |
| E： | 外部 RAM（EMM 0） | ポート0D00H～0D03H |
| F： | 外部 RAM（EMM 1） | ポート0D04H～0D07H |
| G： | グラフィック RAM　(MEM0) | 48K バイト |
| H： | ユーザー指定のドライブ | ドライブ 1 を使用 |

・ドライブ内容の変更は，第 7 章の 7 項を参照してください。

・IPL起動はドライブＡからのみ可能です。

## 2 DIR （ファイル名出力コマンド）

〈コマンド形式・機能〉

1. DIR␣d：
   ドライブ d：上のすべてのファイル名を表示します。

2. DIR␣d：filename. ext
   指定されたファイルを捜し，存在していればそのファイル名を表示します。

注）ログインディスクを取り扱う場合は d：を省略できます。

DIRコマンドは，指定されたディスクにあるファイルをコンソール(通常は CRT 画面)にリストアウトします。

例1 ログインディスクのファイルのリストアウト

ログインディスクがドライブAの場合

```
A>DIR↵
A: MOVCPM   COM : CPM56    SYS : STARTUP  COM : STAT     COM
A: PIP      COM : ED       COM : ASM      COM : LOAD     COM
A: DDT      COM : DUMP     COM : DUMP     ASM : SUBMIT   COM
A: XSUB     COM : FORMAT   COM : COPY     COM : SYSGEN   COM
A: DICCOPY  COM : DICSET   COM : DISKSET  COM : HDMAINT  COM
A: HDBACKUP COM : HDOFF    COM : TERM     COM : CMT      COM
A: KEY      COM : TIME     COM : ULPT0    ASM : ULPT1    ASM
A: ULPT2    ASM : BIOS     MAC : WM       COM : WM       HLP
A: WMENGL   HLP : WMKANA   HLP : WMKANJI  HLP : ONKUN    DIC
A>
```

例2 ログインディスク以外のファイルのリストアウト

ログインディスクがドライブAで，ドライブBのファイルをリストアウトした場合

```
A>DIR B:↵
B: MOVCPM   COM : CPM56    SYS : STARTUP  COM : STAT     COM
B: PIP      COM : ED       COM : ASM      COM : LOAD     COM
B: DDT      COM : DUMP     COM : DUMP     ASM : SUBMIT   COM
B: XSUB     COM : FORMAT   COM : COPY     COM : SYSGEN   COM
B: DICCOPY  COM : DICSET   COM : DISKSET  COM : HDMAINT  COM
B: HDBACKUP COM : HDOFF    COM : TERM     COM : CMT      COM
B: KEY      COM : TIME     COM : ULPT0    ASM : ULPT1    ASM
B: ULPT2    ASM : BIOS     MAC : WM       COM : WM       HLP
B: WMENGL   HLP : WMKANA   HLP : WMKANJI  HLP : ONKUN    DIC
A>
```

この例ではドライブBにファイルが入っていることを示しています。

**例3**
**ファイルを指定したリストアウト**

特定ファイルを指定した場合

```
A>DIR DUMP.ASM ↵
A: DUMP      ASM
A>
```

ファイルをグループで指定した場合

```
A>DIR DUMP.*↵
A: DUMP      COM : DUMP      ASM
A>
```

・上の例ではファイル名が"DUMP"であるファイルを指定しています。

```
A>DIR *.ASM↵
A: DUMP      ASM : ULPT0     ASM : ULPT1     ASM : ULPT2     ASM
A>
```

・上の例ではファイルのエクステンションが"ASM"であるファイルを指定しています。

**ファイルのグループ指定の方法**

P1ABC. COM, P2ABC. COM, P1XY. COM, P1XY. ASM, P2Z. ASMというファイルがあるとき

```
A>DIR *.ASM ↵
A: P1XY      ASM : P2Z       ASM
```

*. ASMと指定すると
P1XY. ASMとP2Z. ASMが選ばれます。

```
A>DIR P??.ASM ↵
A: P2Z       ASM
```

P??. ASMと指定すると
P2Z. ASMが選ばれます。

```
A>DIR P1XY.* ↵
A: P1XY      COM : P1XY      ASM
```

P1XY. *と指定すると
P1XY. COMとP1XY. ASMが選ばれます。

```
A>DIR P2*.* ↵
A: P2ABC     COM : P2Z       ASM
```

P2 *. *と指定すると
P2ABC. COMとP2Z. ASMが選ばれます。

```
A>DIR P?ABC.COM ↵
A: P1ABC     COM : P2ABC     COM
```

P?ABC. COMと指定すると
P1ABC. COMとP2ABC. COMが選ばれます。

```
A>DIR *.* ↵
A: P1ABC     COM : P2ABC     COM : P1XY      COM : P1XY      ASM
A: P2Z       ASM
```

*. *と指定すると
すべてのファイルが選ばれます。

つまり，"?"は，?が位置する文字が何であってもよいことを意味し，"*"は*が現われた位置よりどのような文字があってもよいことを意味しています。

## 3 REN （ファイル名変更コマンド）

〈コマンド形式・機能〉

REN␣d：新 filename. ext=旧 filename. ext

指定ディスク上のファイル名を変更します。

注1）同じディスク上にすでに存在するファイル名には変更できません。

注2）ファイル名は1つのファイルを指定するものでないといけません。

注3）ログインディスクを取り扱う場合はd：を省略できます。

**例1 ファイル名の変更**

DIRコマンドでディスクのファイルを確認してから，RENコマンドでDUMP. ASMをDUMP. XYZという名前のファイルに書きかえます。

```
A>DIR DUMP.*↵
A: DUMP     COM : DUMP     ASM
A>REN DUMP.XYZ=DUMP.ASM↵
A>DIR DUMP.*↵
A: DUMP     COM : DUMP     XYZ
A>
```

**例2 ファイル名の変更ができない場合**

DUMP. COMをDUMP. XYZに書きかえようとしましたが、すでにファイルが存在しているので“FILE EXISTS”というエラーがでます。

```
A>DIR DUMP.*↵
A: DUMP     COM : DUMP     XYZ
A>REN DUMP.XYZ=DUMP.COM↵
FILE EXISTS
A>
```

RENコマンドは，ログインディスク上以外のファイルでも実行できますが，1度DIRコマンドでファイルを確めておいたほうが，重要なファイルを誤って変更したりするようなミスをおかしません。

## 4 ERA （ファイル削除コマンド）

〈コマンド形式・機能〉

ERA␣d：filename. ext

指定ディスク上のファイルを削除します。

注1）このコマンドを実行後，ファイルは再生できません。

注2）ログインディスクを取り扱う場合はｄ：を省略できます。

**例1 特定ファイルの削除**

DIR コマンドでファイルを確認し，WM. HLP のファイルを削除した後，ファイルが削除されたことを確認します。

```
A>DIR WM.* ↵
A: WM       COM : WM       HLP
A>ERA WM.HLP ↵
A>DIR WM.* ↵
A: WM       COM
A>
```

**例2 グループ単位でのファイルの削除**

① 〝＊〟を使って〝DUMP.〟　を満足するファイルをすべて削除します。

```
A>DIR DUMP.* ↵
A: DUMP     COM : DUMP     ASM
A>ERA DUMP.* ↵
A>DIR DUMP.* ↵
NO FILE
A>
```

② 〝＊〟を使って〝……. COM〟を満足するファイルをすべて削除します。

```
A>DIR *.COM ↵
A: MOVCPM   COM : STARTUP  COM : STAT     COM : PIP      COM
A: ED       COM : ASM      COM : LOAD     COM : DDT      COM
A: DUMP     COM : SUBMIT   COM : XSUB     COM : FORMAT   COM
A: COPY     COM : SYSGEN   COM : DICCOPY  COM : DICSET   COM
A: DISKSET  COM : HDMAINT  COM : HDBACKUP COM : HDOFF    COM
A: TERM     COM : CMT      COM : KEY      COM : TIME     COM
A: WM       COM
A>ERA *.COM ↵
A>DIR *.COM ↵
NO FILE
A>
```

③"*.*"を使うことによってすべてのファイルを削除します。このとき，本当にすべて削除するか確認してきますので，すべて削除するときは Y↵を入力します。

```
A>DIR↵
A: CPM56    SYS : DUMP     ASM : ULPT0    ASM : ULPT1    ASM
A: ULPT2    ASM : BIOS     MAC : WMENGL   HLP : WMKANA   HLP
A: WMKANJI  HLP : ONKUN    DIC
A>ERA *.*↵
ALL (Y/N)?Y↵
A>DIR↵
NO FILE
A>
```

ERA コマンドも，ログインディスク以外のファイルに対して実行できますが，誤って重要なファイルを削除してしまう恐れがありますので，DIR コマンドで確認し，かつ，ログインディスク上のファイルに対して ERA コマンドを実行したほうがよいでしょう。

## 5 TYPE （ファイル内容出力コマンド）

〈コマンド形式・機能〉

TYPE␣d：filename. ext

指定ディスク上のアスキーファイルの内容を表示します。

注1）ファイルはアスキーコードで表記されているものに限ります。

注2）ファイル名は1つのファイルを指定するものでなければいけません。（ファイルのグループ指定はできません。）

注3）ログインディスクを取り扱う場合はd：を省略できます。

TYPEコマンドは，指定されたディスクのアスキーファイルの内容をコンソールに出力します。

例1 "DUMP.ASM" というファイルのタイプアウト

```
A>TYPE DUMP.ASM↵
;       FILE DUMP PROGRAM, READS AN INPUT FILE AND PRINTS IN HEX
;
;       COPYRIGHT (C) 1975, 1976, 1977, 1978
;       DIGITAL RESEARCH
;       BOX 579, PACIFIC GROVE
;       CALIFORNIA, 93950
;
        ORG     100H
BDOS    EQU     0005H   ;DOS ENTRY POINT
CONS    EQU     1       ;READ CONSOLE
TYPEF   EQU     2       ;TYPE FUNCTION
PRINTF  EQU     9       ;BUFFER PRINT ENTRY
BRKF    EQU     11      ;BREAK KEY FUNCTION (TRUE IF CHAR READY)
OPENF   EQU     15      ;FILE OPEN
READF   EQU     20      ;READ FUNCTION
;
FCB     EQU     5CH     ;FILE CONTROL BLOCK ADDRESS
BUFF    EQU     80H     ;INPUT DISK BUFFER ADDRESS
;
;       NON GRAPHIC CHARACTERS
CR      EQU     0DH     ;CARRIAGE RETURN
LF      EQU     0AH     ;LINE FEED
;
A>
```

・ファイルの表示の途中でCP/Mシステムに戻るときには、何らかのキーを押してください。

・表示を一時停止するときには，CTRL+S（CTRLキーを押しながらSを押す），表示を再開するときにはもう一度CTRL+Sを行ないます。

## 6 USER （ユーザーエリア移行コマンド）

〈コマンド形式・機能〉

USER␣n

ユーザーエリアに移行します。
（n は0～15の10進数）

USER コマンドは，1 枚のフロッピーディスクを仮想の16枚のディスケットに分割し，あたかも16枚のフロッピーディスクを使用しているかのようにみせるものです。通常はこのコマンドを使う必要はありませんが，大容量のハードディスクを使用するときなど便利です。
CP/M 起動時は，ユーザー 0 にセットされています。

1 枚のフロッピーディスク

| ユーザー0 | ユーザー1 | ユーザー2 | | ユーザー15 |
|---|---|---|---|---|
| PIP. COM<br>STAT. COM | ASM. COM<br>ED. COM<br>PIP. COM | PIP. COM | ..... | |

たとえば，ユーザー 1 で登録したファイルはユーザー 1 でのみ利用可能でユーザー 0 やユーザー 2 では利用することができません。また，ユーザーエリアの移行はすべてのドライブに対しても同じ効果を与えます。ドライブ A がユーザーエリア 3 ならドライブ B も 3 になります。

例　他のユーザーエリアへの PIP. COM の登録及びファイルの転送方法

```
A>USER 0↵                                    ①
A>DDT PIP.COM↵
DDT VERS 2.2                                 ②
NEXT  PC
1E00 0100
-^C CTRL+Cを入力します。

A>USER 2↵                                    ③
A>DIR ↵                                      ④
NO FILE
A>SAVE 29 PIP.COM↵                           ⑤
A>DIR↵                                       ⑥
A: PIP      COM
A>PIP A:=A:STAT.COM[G0]↵                     ⑦

A>DIR↵                                       ⑧
A: PIP      COM : STAT      COM
A>STAT USR:↵                                 ⑨

Active User : 2
Active Files: 0 2
A>
```

① ユーザー 0 に移ります。
② DDT コマンド（6 章参照）で PIP. COM をメモリーにとりこみます。
③ ユーザー 0 からユーザー 2 に移ります。
④ ユーザー 2 のファイルを調べます。まだファイルがありません。
⑤ メモリーの29ページ分を PIP. COM としてドライブにセーブします。
⑥ ユーザー 2 に PIP. COM が生成されたことを確認します。
⑦ ユーザー 2 に生成された PIP コマンドにより，ユーザー 0 にある STAT.COM ファイルをユーザー2のドライブ A に転送します。（オプション〔Gn〕はユーザ n にあるファイルを現在のユーザに転送するという意味です）
⑧ ファイルが転送されたことを確認します。
⑨ STAT USR：↵で現在のユーザーエリアがわかります。
例では，現在ユーザー 2 にあり，ファイルがユーザー 0 とユーザー 2 に存在していることを示しています。

# 7 SAVE （メモリ内容セーブコマンド）

〈コマンド形式・機能〉

**SAVE␣n␣d：filename.ext**

TPAの始まりからメモリの内容をnページ分（1ページ=256バイト，nは10進数）だけ，ファイル名をつけて指定装置にセーブします。

注1）TPA（トランジェント・プログラム・エリア）は，トランジェントプログラムがロードされるエリアで，アドレス100Hから始まります。

注2）ログインディスクを取り扱う場合はd：を省略できます。

例
メモリーの2ページ分をディスクにセーブする

```
A>DIR *.COM↵──────────────────────────────────①
A: MOVCPM   COM : STARTUP  COM : STAT     COM : PIP      COM
A: ED       COM : ASM      COM : LOAD     COM : DDT      COM
A: DUMP     COM : SUBMIT   COM : XSUB     COM : FORMAT   COM
A: COPY     COM : SYSGEN   COM : DICCOPY  COM : DICSET   COM
A: DISKSET  COM : HDMAINT  COM : HDBACKUP COM : HDOFF    COM
A: TERM     COM : CMT      COM : KEY      COM : TIME     COM
A: WM       COM
A>DDT DUMP.COM↵──────────────────────────────②
DDT VERS 2.2
NEXT  PC
0300 0100 ──────────────────────────────────③
-^C ────────────────────────────────────────④
A>SAVE 2 DU.COM↵─────────────────────────────⑤

A>DIR *.COM↵──────────────────────────────────⑥
A: MOVCPM   COM : STARTUP  COM : STAT     COM : PIP      COM
A: ED       COM : ASM      COM : LOAD     COM : DDT      COM
A: DUMP     COM : SUBMIT   COM : XSUB     COM : FORMAT   COM
A: COPY     COM : SYSGEN   COM : DICCOPY  COM : DICSET   COM
A: DISKSET  COM : HDMAINT  COM : HDBACKUP COM : HDOFF    COM
A: TERM     COM : CMT      COM : KEY      COM : TIME     COM
A: WM       COM : DU       COM
A>DU DU.COM↵──────────────────────────────────⑦

0000 21 00 00 39 22 15 02 31 57 02 CD C1 01 FE FF C2
0010 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 13 02
0020 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2
0030 44 01 CD 72 01 CD 59 01 0F DA 51 01 7C CD 8F 01


A> DUMP DU.COM↵──────────────────────────────⑧

0000 21 00 00 39 22 15 02 31 57 02 CD C1 01 FE FF C2
0010 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 13 02
0020 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2
0030 44 01 CD 72 01 CD 59 01 0F DA 51 01 7C CD 8F 01


A>
```

① DIR コマンドでファイルを確認します。
② DDT コマンド（6章参照）でセーブしたいファイル（例では DUMP. COM）をメモリーにとりこみます。
③ DDT のメッセージによりメモリーの100H～300Hまで，すなわち 2 ページ分がこのファイルに使用されていることがわかります。
④ CTRL + C でCP/Mシステムに戻ります。
⑤ SAVE コマンドでメモリーの2ページ分を，"DU. COM" というファイル名でディスクに SAVE します。
⑥ DIR コマンドで "DU. COM" がセーブされていることを確認します。
⑦·⑧ で DU と DUMP を実行してみます。これにより，DUMP. COM と DU. COM は全く同じものとわかります。

# 第6章

# トランジェントコマンド

# 1 STAT （ファイルおよび周辺装置の設定・状態確認コマンド）

〈主なコマンド形式・機能〉

1. STAT␣d：
   ディスクの未使用領域を表示します。ドライブｄ：を省略するとCP/M起動後，一度でもアクセスされたすべてのディスクの未使用領域を表示します。

2. STAT␣d：filename. ext
   指定したファイルについて，ファイルの使用領域の大きさと，ファイルアクセスモードを表示します。

3. STAT␣DEV：
   周辺装置の現在の割付け状況を表示します。

4. STAT␣VAL：
   STATコマンドの一般形式を表示する一種のHELPメニューです。

5. STAT␣d：DSK：
   指定したディスクの状況を示します。ドライブｄ：を省略すると，一度でもアクセスされたすべてのディスクの状況を示します。

注）ログインディスクを取扱う場合はドライブ名ｄ：を省略することができます。

**STATコマンドは，ディスクのファイルの使用状況を調べるときや，ファイルの属性の確認および指定，周辺装置の割当て確認をする場合に使用します。**

**例1 ディスクの空きエリアの容量を調べる**

STAT ↵を入力することにより，ディスクの空きエリアを調べることができます。

```
A>STAT ↵
A: R/W, Space: 70k
B: R/O, Space: 274k

A>
```

上の例では，ドライブAが読み出し，書き込み可能で未使用領域が70Kバイト，ドライブBが読み出しのみ可能で未使用領域が274Kバイトであるということを示しています。
また，このようにドライブ名を省略した場合，一度でもアクセスされたドライブはすべて状況がリストアウトされます。

**例2**
**ファイルの状況を調べる**

STAT␣ファイル名 ↵を入力すると，使用したレコード数，バイト数などの状況を調べることができます。

```
A>STAT *.COM ↵

 Recs   Bytes   Ext Acc
   64      8k     1 R/W A:ASM.COM
   13      2k     1 R/W A:CMT.COM
   10      2k     1 R/W A:COPY.COM
   38      6k     1 R/W A:DDT.COM
   15      2k     1 R/W A:DICCOPY.COM
   10      2k     1 R/W A:DICSET.COM
   26      4k     1 R/W A:DISKSET.COM
    4      2k     1 R/W A:DUMP.COM
   52      8k     1 R/W A:ED.COM
   35      6k     1 R/W A:FORMAT.COM
   33      6k     1 R/W A:HDBACKUP.COM
   22      4k     1 R/W A:HDMAINT.COM
    2      2k     1 R/W A:HDOFF.COM
    6      2k     1 R/W A:KEY.COM
   14      2k     1 R/W A:LOAD.COM
   76     10k     1 R/W A:MOVCPM.COM
   58      8k     1 R/W A:PIP.COM
   54      8k     1 R/W A:STARTUP.COM
   41      6k     1 R/W A:STAT.COM
   10      2k     1 R/W A:SUBMIT.COM
   12      2k     1 R/W A:SYSGEN.COM
   28      4k     1 R/W A:TERM.COM
    7      2k     1 R/W A:TIME.COM
  120     16k     1 R/W A:WM.COM
    6      2k     1 R/W A:XSUB.COM
Bytes Remaining On A: 70k

A>
```

Recs：使用したレコード数を示しています。1レコードは128バイト。

Bytes：使用したバイト数を示しています。2Kバイト単位で増えていきます。（5インチ2D）

Ext　：エクステント数を示しています。CP/Mのファイル管理上，32Kバイト単位で1エクステント使用します。（5インチ2D）

Acc　：ファイルの状況を示しています。
　　R/W……ファイルに書き込み可能
　　R/O……書き込み・削除禁止

Bytes Remaining：ディスクの空きエリアを示しています。

**例3**
**周辺装置の割付けを調べる**

STAT␣DEV：↵を入力すると，現在の周辺装置の割付け状態を調べることができます。

```
A>STAT DEV:↵
CON: is CRT:
RDR: is TTY:
PUN: is TTY:
LST: is LPT:

A>
```

〈論理デバイスと物理デバイス〉

例3はCP/Mスタート時の周辺装置の割付けを示しています。周辺装置には，大きく分類して4つの論理デバイスがあり，その1つ1つに対してさらに4つの物理デバイスを割付けることができます。

論理デバイス……例3で左側のCON：，RDR：，PUN：，LST：の4つはCP/Mシステムがプログラム上で取扱っている周辺装置で，論理デバイスといいます。

CON：……システムコンソールデバイス
RDR：……紙テープ・リーダ
PUN：……紙テープ・パンチャー
LST：……リストデバイス

物理デバイス……例3で右側のCRT：，TTY：，LPT：，の3つはコンピュータシステムに接続されている実際の装置で，物理デバイスといいます。turboCP/Mの周辺装置の割付けはCON：にCRT：(X1turboの画面出力とキー入力)，RDR：とPUN：にTTY：(X1turbo内蔵RS232C)，LST：にLPT：(X1turbo用プリンタ)となっていることがわかります。

**例4**
**STATコマンドのメニュー表示**

STAT␣VAL：↵と入力することにより，STATコマンドのメニューと，各論理デバイスに対して割付け可能な物理デバイスを表示します。

```
A>STAT VAL:↵

Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS $DIR
Disk Status  : DSK: d:DSK:
User Status  : USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
A>
```

①STAT d：=R/O

ドライブd：を書き込み禁止にします。ディスクがリブートしたときに解除されます。d：にはA：～G：まで指定できます。

②STAT d：filename.ext この後に$R/O，$R/W，$SYS，$DIRいずれかを指定します。

指定ファイルに属性を設定します。ファイルの属性には次のような種類があります。

$R/O ……Read Only File。指定ファイルを書き込み禁止にして，ファイルの削除や，一部の変更などができなくなります。

$R/W ……Read/Write File。指定ファイルは書き込み可能となり，ファイルの削除や変更が自由にできるようになります。

$SYS ……System File。この属性をもつファイルは，DIR コマンドによる表示には現れず，PIP コマンドによる転送もできません。

$DIR ……Directory File。"SYS"の属性をもつファイルを，DIR コマンドにより表示される通常のファイルに戻します。

これらの属性はファイルのエクステンションとは異なりますのでご注意ください。

③STAT d：DSK：

指定ディスクの状況を調べます。（例5を参照してください。）

④STAT USR：

現在のユーザーモードを調べます。（USER コマンドを参照してください。）

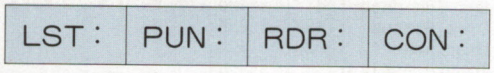
⑤STAT CON：=CRT：（TTY：，BAT：，UC1：）

論理デバイスに対して物理デバイスを割付けます。

## IOバイトについて

例4のIObyte Assignが示すように，CP/M では4つの各論理デバイスに4種類の中の1つの物理デバイスを割付けることができます。デバイスの割付けは通常 STAT コマンドで行ないますが，直接 IO バイトを操作して行なうこともできます。IO バイトは，物理デバイスの割付け状況をしまっておくところで，メモリーの3番地にあります。

IO バイトの構成は次のようになっています。




MSB　　LSB

turbo CP/M スタート時，論理デバイスの割付けは

CON：=CRT：(01)
RDR：=TTY：(00)
PUN：=TTY：(00)
LST：=LPT：(10)

のようになっています。IO バイトの変更は STARTUP コマンドによって行なうことができます。

また，IO バイトについての詳しい説明は12章を参照してください。

## 例5 ディスクの状況を調べる

STAT␣DSK：↵を入力することによりディスクの状況を知ることができます。

```
A>STAT DSK:↵

    A: Drive Characteristics ——————————①
 2432: 128 Byte Record Capacity ——————②
  304: Kilobyte Drive  Capacity ————————③
   64: 32  Byte Directory Entries ———————④
   64: Checked  Directory Entries ———————⑤
  256: Records/ Exten ———————————————⑥
   16: Records/ Block ———————————————⑦
   64: Sectors/ Track ———————————————⑧
    2: Reserved Tracks ——————————————⑨

A>
```

①ドライブAの状況であることを示しています。

②ディスクに128バイトのレコードを2432個書けることを示しています。

③上の②をバイト数に直したものです。

2432×128(バイト)÷1024=304Kバイト

④ ⑤ディスクに書けるディレクトリの数を示しています。ディレクトリとはどのようなファイルがディスクのどこに入っているかという情報を入れておくところで，この例では64個入ることを示しています。

⑥ 1エクステントの中のレコード数を示します。

256×128(バイト)÷1024=32Kバイト

つまり，1ディレクトリで指定されるのが32Kバイトまでであることを示しています。このため，ファイルの大きさが32Kバイトを超える場合，32Kバイトごとに1エクステントを使用しますので，ディレクトリに登録できるファイルの数はそれだけ少なくなります。

⑦ 1ブロック中のレコード数を示します。16×128(バイト)÷1024=2Kバイト単位で，CP/Mシステムがディスク上の領域（ブロック）を割り付けていることを示しています。

⑧ 1トラックに64セクターあることを示しています。

⑨CP/Mシステム専用に2トラック使用していることを示しています。

## 2 PIP （周辺装置間データ転送コマンド）

〈主なコマンド形式・機能〉

1．PIP

PIP プログラムを起動します。

2．PIP␣d：=s：filename. ext

ドライブｄ：上にドライブｓ：上のファイルをコピーします。ファイルのグループ指定も可能です。

3．PIP␣d：newname. ext=s：filename. ext

ドライブｄ：上にファイル名をかえてドライブｓ：上のファイルをコピーします。

注）ログインディスクを取扱う場合，ドライブ名ｄ：やｓ：は省略することができます。

PIP コマンドは，CP/M システムの管理下にある周辺装置間のファイルの転送操作を行なうものです。ディスクからディスクへのファイル転送だけでなく，ディスクから CRT ディスプレイやプリンタ装置などへも転送することができます。
また，このコマンドにはオプションコマンドもあり，その使い方によってさまざまな処理ができます。

例1 PIP コマンドの実行法

①PIP コマンドのみを単独で起動する場合

```
A>PIP ↲
*B:=ASM.COM
* ―――――――――――― ↲あるいは CTRL + C を入力

A>
```

PIP コマンドのみを起動すると，次の処理を要求する〝*〟が表示されます。
必要な回数だけ繰り返し処理することができます。この場合〝*〟に続けて PIP のコマンドラインを入力し，↲を押します。CP/M システムに戻るには ↲あるいは CTRL + C を入力します。

②PIP コマンドの起動と同時に，次の処理の実行も行なう場合

```
A>PIP B:=ASM.COM ↲

A>
```

PIP に続けてコマンドラインも入力すると，PIP プログラムの起動と同時にコマンドラインも実行して，実行が終わると CP/M システムに戻ります。

例 2
ファイルのコピー

①ファイル名を変えずに特定のファイルをコピーする場合

```
A>PIP B:=ASM.COM ↵

A>
```

ドライブBにドライブAの "ASM.COM" をファイル名を変えずに転送します。

②ファイル名を変えてコピーする場合

```
A>PIP B:ASM8080.COM=ASM.COM ↵

A>
```

ドライブAの "ASM.COM" というファイルを "ASM8080.COM" というファイル名に変えてドライブBに転送します。

③グループを指定してコピーする場合

```
A>PIP B:=*.ASM ↵

COPYING -
DUMP.ASM
ULPT0.ASM
ULPT1.ASM
ULPT2.ASM


A>
```

ドライブAのファイルのうち，エクステンションが "ASM" であるファイルをすべてドライブBに転送します。

例 3
周辺装置への
ファイル転送

①コンソールデバイスにファイルを出力する場合

```
A>PIP CON:=DUMP.ASM ↵
;       FILE DUMP PROGRAM, READS AN INPUT FILE AND PRINTS IN HEX
;
;       COPYRIGHT (C) 1975, 1976, 1977, 1978
;       DIGITAL RESEARCH
;       BOX 579, PACIFIC GROVE
;       CALIFORNIA, 93950
;
        ORG     100H
BDOS    EQU     0005H   ;DOS ENTRY POINT
CONS    EQU     1       ;READ CONSOLE
TYPEF   EQU     2       ;TYPE FUNCTION
PRINTF  EQU     9       ;BUFFER PRINT ENTRY
BRKF    EQU     11      ;BREAK KEY FUNCTION (TRUE IF CHAR READY)
OPENF   EQU     15      ;FILE OPEN
READF   EQU     20      ;READ FUNCTION
;
FCB     EQU     5CH     ;FILE CONTROL BLOCK ADDRESS
ABORTED: DUMP.ASM

A>
```

初期設定では CON：(コンソールデバイス)には CRT：(画面出力)が割付けられているのでファイルの内容が画面に表示されます。(周辺装置の割付けについては STAT を参照してください。)

②リストデバイスにファイルを出力する場合

```
A>PIP LST:=DUMP.ASM ↵
A>
```

初期設定ではLST：(リストデバイス) にはLPT：(プリンタ) が割付けられているので，ファイルはプリンタに出力されます。①と同様に何らかのキーを入力すると，PIP 処理を中止します。

③RS-232C にファイルを出力する場合

```
A>PIP TTY:=DUMP.ASM ↵
A>
```

ただし，RS-232C に出力する場合，あらかじめ通信パラメータの設定を行なう必要がありますので，第7章の STARTUP を参照してください。

## オプション機能について

PIP コマンドラインの最後にオプションを〔 〕で囲んで入力すると，指定したオプションを行ないながらファイルを転送します。次の例では，〔V〕オプションを指定して，ドライブBにファイルを転送した後，転送されたファイルを再度読みこむことにより，正確にコピーされているかを確認しています。

```
A>PIP B:=ASM.COM[V] ↵
A>
```

オプションには様々なものがあり，1回に1つだけでなく必要なものを組み合わせて使用することもあります。また，オプションの中には，データを読み込んだときに働くものと，データを書き出すときに働くものとがあります。オプション機能をまとめると次のようになります。

| オプション | 機　　能 |
|---|---|
| B<br>(Block) | ・入力ファイルよりX-off（13H）コードがくるまでバッファに読み込みます。<br>・X-off（13H）コードがくると，バッファの情報を出力ファイルに書き出します。 |
| Dn<br>(Delete) | ・1行の字数をn字で打ち切ります。 |
| E<br>(Echo) | ・転送されているデータをコンソールに表示します。 |
| F<br>(Form feed) | ・転送データから Form-Feed（0CH）コードを取り除きます。 |
| Gn<br>(Get) | ・ユーザーnにあるファイルを現ユーザーエリアに転送します。<br>（USRコマンド参照） |
| H<br>(Hex format) | ・HEXファイルを転送するときにインテル16進型式になっているかをチェックします。 |
| I<br>(Ignores null) | ・HEXファイルを転送しているときにヌルレコード（:00）で始まるレコードを無視します。 |
| L<br>(Lower case) | ・すべての英大文字を英小文字に変えて転送します。<br>（漢字ファイルでは使用できません。） |
| N<br>(Line numbers) | ・新しいファイルに行番号と：を付けて出力します。<br>・Nだけのときは，上位の0は空白になりますが，N2を指定すると，上位の0も出力されます。 |
| O<br>(Object files) | ・ASCIIファイル以外のファイルを転送するときに，ファイルの終りにあるEOF（1AH）コードを無視します。 |
| Pn<br>(Page ejects) | ・n行ごとにページ送りを入れます。nが1または指定されないときには，60行ごとに行なわれます。 |
| Q文字列^Z<br>(Quit) | ・転送データの中に指定した文字列と同じものがあると，ファイルの転送を中止します。文字列自身も転送されます。 |
| S文字列^Z<br>(Start) | ・転送されるデータから文字列をさがし，見つかると文字列自身も含めて以後のデータ転送を開始します。 |
| Tn<br>(Tab) | ・転送データの中に Tab（09H）コードが見つかると，n字にタブをセットします。 |
| U<br>(Upper case) | ・すべての英小文字を英大文字に変えて転送します。<br>（漢字ファイルでは使用できません。） |
| V<br>(Verify) | ・ファイルの転送を行なった後，ファイルが正しく書かれているかをチェックします。 |
| Z<br>(Zeros parity) | ・データの受信時に ASCII キャラクタのパリティビット（bit8）を0にします。<br>（漢字ファイルでは使用できません。） |

## 3 ED （テキストエディタコマンド）

〈コマンド形式・機能〉

ED␣d：filename. ext

アスキーファイルの作成・修正・追加・削除をします。

注）ログインディスクを取り扱う場合はd：を省略することができます。

EDコマンドは，CP/Mシステムの文編集プログラムで，アセンブラや高級言語のソースプログラムを作成することができます。ただし，EDはラインエディタですから通常はスクリーンエディタのワードマスターを使って行なう方が便利です

例
ファイルの作成

この例では新しくABC. XYZというファイルを作成します。

```
A>ED ABC.XYZ ↵ ―――――――――①

NEW FILE  ―――――――――②
     : *i ↵ ―――――――――③
    1:  New Product Information
    2:
    3:  Image Input Device for the X1 Series and
    4:  X1 Turbo Series Personal Computer TVs
    5:
    6:  Features
    7:  (1) High-speed processing of images input
    8:  (2) New built-in SCRAMBLE circuit for subtle colors
    9:  (3) Easy to use
   10:
     : *b ↵ ―――――――――④
    1: *#t ↵ ―――――――――⑤
    1:  New Product Information
    2:
    3:  Image Input Device for the X1 Series and
    4:  X1 Turbo Series Personal Computer TVs
    5:
    6:  Features
    7:  (1) High-speed processing of images input
    8:  (2) New built-in SCRAMBLE circuit for subtle colors
    9:  (3) Easy to use
    1: *2|t ↵ ―――――――――⑥
    3:  Image Input Device for the X1 Series and
    3: *k ↵ ―――――――――⑦
    3: *b#t ↵ ―――――――――⑧
    1:  New Product Information
    2:
    3:  X1 Turbo Series Personal Computer TVs
    4:
    5:  Features
    6:  (1) High-speed processing of images input
    7:  (2) New built-in SCRAMBLE circuit for subtle colors
    8:  (3) Easy to use
    1: *e ↵ ―――――――――⑨


A>DIR ABC.XYZ ↵ ―――――――――⑩
A: ABC      XYZ
A>
```

①EDで新しいファイルABC. XYZを作成します。

②新しいファイルであることを示しています。

③ｉ*1)コマンドでインサートモードに入り，文を作成します。1行の終りは ↵を入力し，インサートモードから出るには CTRL ＋ Z を入力します。

④ｂコマンドでキャラクタポインタ*2)を先頭（1：）まで戻します。

⑤ｔコマンドでキャラクタポインタの位置から指定行数だけ表示します。
#ｔコマンドを指定するとすべての行を表示します。

⑥ｌコマンドでキャラクタポインタを進めて，ｔコマンドでキャラクタポインタの示す行を表示します。この場合，2行進めて1行表示しています。

⑦Ｋコマンドでキャラクタポインタの示す位置から指定行数削除します。
この場合，"X1 Turbo Series Personal Computer TVs" の1行が削除されます。

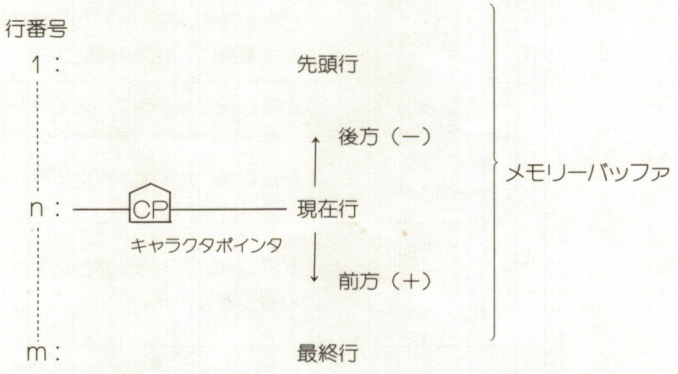
⑧キャラクタポインタを先頭に戻し，すべての行を表示して1行削除されたことを確認しています。

⑨ｅコマンドでエディタを終了し，CP/Mシステムに戻ります。

⑩新しいファイルができたことを確認しています。

＊1）エディタ内で小文字を扱う場合，コマンドは小文字で入力してください。
コマンドによっては編集文の小文字の部分が大文字に変換されてしまうからです。

＊2）キャラクタポインタとは，編集文が入っているメモリーバッファ内を移動する仮想の文字ポインタです。このポインタを基準として，エディタのオペレーションを行ないます。



## EDコマンド一覧

| EDコマンド | 機　　能 |
|---|---|
| nA | テキストn行をメモリーバッファにとりこみます。<br>（n＝#で全テキスト，n＝0でメモリーバッファの半分） |
| E | 編集終了。エディットバッファのテキストをディスクにセーブしてCP/Mに戻ります。 |
| H | 編集終了。テキストをセーブし，メモリーバッファをクリアして再び同一ファイル名でEDを起動します。 |
| O | 編集中止。一切を中断して，最初にEDを起動した状態に戻ります。 |
| Q | 編集中止。オリジナルファイルは何も変更されず，CP/Mに戻ります。 |
| nW | エディットバッファのテキストn行をテンポラリファイル“filename. $$$”にセーブします。セーブしたいn行はエディットバッファから削除されます。 |
| ±B | CPをテキストの最初または最後（−B）に移動します。 |
| ±nC | CPをn文字分前進またはバック（−nC）します。 |
| ±nL | CPを±n行移動し，その行の先頭にセットします。 |
| ±n(±Lと同じ) | CPを±n行移動し，その行をタイプアウトします。 |
| n： | CPをラインnの先頭にセットします。 |
| ↵(リターン) | CP位置の行をタイプアウトします。（Lと同じです） |
| I | CP位置にテキスト挿入を開始します。（^Zで解除します。） |
| ±nD | CP位置から±n文字削除します。 |
| ±nK | CP位置から±n行削除します。 |
| nF 文字列 | CP以後のメモリーバッファにあるテキストで，n番目の指定文字列の最後にCPを移動します |
| nN 文字列 | CP以後の全テキストに対する同上の機能です |
| nS 文字列1 ^Z 文字列2 | CP以後，文字列1を文字列2に変更するのをn回繰り返します。 |
| nJ 文字列1 ^Z 文字列2 ^Z 文字列3 | CP以後，文字列1から文字列3までを文字列2に変更するのをn回繰り返します。 |
| ±U | Uコマンド以後，入力する文字をすべて大文字に変換します。（−Uで解除） |
| ±nP | CP位置から±nページ分をコンソールにタイプアウトします。 |
| ±nT | CP位置から±n行をタイプアウトします。 |
| ±V | 行番号を表示するモード，表示しないモード（−V）にします。 |

| | |
|---|---|
| nX | CP 位置よりn行をテンポラリファイル “X$$$$$$$$. LIB” としてセーブします。通常，この後Rコマンドを使用します。 |
| R | CP 位置より，“X$$$$$$$$. LIB” ファイルの内容を挿入します。 |
| nMcommand | CP 位置より command をｎ回繰り返します。<br>（ｎを省略するとバッファの最後まで繰り返します。） |
| n：command | 行番号ｎより command を実行します。 |
| ：ncommand | 行番号ｎまで command を実行します。 |
| n1：：n2<br>command | 行番号 n1から n2まで command を実行します。 |

注）・CP はキャラクタポインタの略です。
・±ｎの＋の場合は＋記号を省略できます。
・ｎが１のとき，この１は省略できます。
・各コマンドは並べて記述することができます。例　B10T
・編集文で小文字を使用する場合，コマンドは小文字で入力してください。

ED は，コマンドが完全に実行できなかった場合，次のエラーメッセージを出力します。

BREAK ＼#＼ AT
↑――エラー種別

エラーメッセージが出力されたラインに関しては，コマンドは実行されずにもとのままの内容を保っています。
エラーメッセージには，次の種類があります。

| | |
|---|---|
| ? | コマンドが間違っている。 |
| > | エディット・バッファがいっぱいになったかF，N，Sコマンドの文字列が長すぎる。 |
| # | 指定回数分のコマンドの実行ができない。 |
| 0 | Rコマンド実行で，LIB ファイルをオープンできない。 |

# 4 ASM （8080アセンブラ）

〈コマンド形式・機能〉

ASM␣d：filename

アセンブリ言語で書かれたソースファイル（filename. ASM）をアセンブルし，生成されたHEXファイルとPRNファイルを同じディスク上にセーブします。

d：ASM␣filename. shp

“s”で指定されるドライブ上のソースファイルをアセンブルし，“h”で指定されるドライブにHEXファイルを，“p”で指定されるドライブにPRNファイルを出力します。

注）ASM. COMがログインディスク上にある場合，d：は省略できます。

例1
アセンブルの実行

この例では，DUMP. ASMというファイルをアセンブルします。

```
A>DIR DUMP.* ↲ ―――――――――――――――――――――――――①
A: DUMP     COM : DUMP     ASM
A>ASM DUMP ↲ ―――――――――――――――――――――――――②
CP/M ASSEMBLER - VER 2.0
0257 ―――――――――――――――――――――――――――――③
001H USE FACTOR ―――――――――――――――――――――④
END OF ASSEMBLY

A>DIR DUMP.* ↲ ―――――――――――――――――――――――――⑤
A: DUMP     COM : DUMP     ASM : DUMP     PRN : DUMP     HEX
A>
```

①ファイルを確認します

②アセンブルしたいファイル名（“……. ASM”に限ります。）を指定してアセンブルを実行します。（このとき，エクステンションの“ ASM”は入力しません。）

③アセンブルプログラムの最終番地より1番高いアドレスを表示しています。

④プログラムで使用しているシンボリックテーブル*1)の使用量を表示しています。

⑤アセンブルにより，DUMP. PRNとDUMP. HEXのファイルが生成されています。

＊1）シンボリックテーブルとはソースプログラム中で使用されたラベルや記号と，それに対応するアドレスを示す表です。00～FFまでの値をとります。

## HEX ファイル

インテル HEX 形式のファイルは，次の項で解説する LOAD プログラムによって，直接実行可能なマシン語コードに変換することができます。

例1　で生成された DUMP. HEX ファイルの内容を，TYPE コマンドで見ることができます。

・DUMP. HEX ファイル

```
A>TYPE DUMP.HEX ↵
:10010000210000392215023157 02CDC101FEFFC284
:10011000 1B0111F301CD9C01C351013E803213023A
:10012000210000E5CDA201E1DA5101477DE60FC2D1
:10013000 4401CD7201CD59010FDA51017CCD8F01FF
:10014000 7DCD8F01233E20CD650178CD8F01C32366
:10015000 01CD72012A1502F9C9E5D5C50E0BCD05F1
:10016000 00C1D1E1C9E5D5C50E025FCD0500C1D101
:10017000 E1C93E0DCD65013E0ACD6501C9E60FFE20
:10018000 0AD28901C630C38B01C637CD6501C9F5D6
:10019000 0F0F0F0FCD7D01F1CD7D01C90E09CD05EA
:1001A000 0C93A1302FE80C2B301CDCE01B7CAB373
:1001B000 0137C95F16003C321302218000197EB757
:1001C000 C9AF327C00115C000E0FCD0500C9E5D52A
:1001D000 C5115C000E14CD0500C1D1E1C946494CE2
:1001E000 0452044554D502056455253494F4E2031DD
:1001F000 2E34240D0A4E4F20494E50555420464966
:10020000 4C452050524553454E54204F4E204449B2
:03021000534B2429
:0000000000
```

①②③④⑤⑥

①レコードマーク……レコードの始まりを示します

②データ量……データ量をバイト単位で表わすレコード長を示します

③アドレス……ロード開始アドレス

④型式……レコードの性質を表わしますが，本 CP/M では使用しないので常に00になります

⑤データ……メモリにロードされるデータの部分

⑥チェックサム……レコード長からデータの最後までのチェックサムコード

## PRN ファイル

PRN ファイルは，ソースファイルの左に各ステートメントのマシンコードやエラーメッセージなどを付加したリストファイルです。
例 1 で生成された DUMP. PRN ファイルの内容を TYPE コマンドで見ることができます。

・DUMP. PRN ファイル

```
A>TYPE DUMP.PRN ↵

                ;       FILE DUMP PROGRAM, READS AN INPUT FILE AND PRINTS IN
                ;
                ;       COPYRIGHT (C) 1975, 1976, 1977, 1978
                ;       DIGITAL RESEARCH
                ;       BOX 579, PACIFIC GROVE
                ;       CALIFORNIA, 93950
                ;
 0100                   ORG     100H
 0005 =         BDOS    EQU     0005H   ;DOS ENTRY POINT
 0001 =         CONS    EQU     1       ;READ CONSOLE
 0002 =         TYPEF   EQU     2       ;TYPE FUNCTION
 0009 =         PRINTF  EQU     9       ;BUFFER PRINT ENTRY
 000B =         BRKF    EQU     11      ;BREAK KEY FUNCTION (TRUE IF CHAR REA
 000F =         OPENF   EQU     15      ;FILE OPEN
 0014 =         READF   EQU     20      ;READ FUNCTION
                ;
 005C =         FCB     EQU     5CH     ;FILE CONTROL BLOCK ADDRESS
 0080 =         BUFF    EQU     80H     ;INPUT DISK BUFFER ADDRESS
                ;
                ;       NON GRAPHIC CHARACTERS
 000D =         CR      EQU     0DH     ;CARRIAGE RETURN
 000A =         LF      EQU     0AH     ;LINE FEED
                ;

A>
```

## PRNファイル内のエラーメッセージ

プログラムにエラーが存在する場合は，エラーメッセージが PRN ファイルの左端にアルファベット 1 文字で表示されます。エラーのある行は，コンソールにも出力されますので，ソースをリストアウトしてエラーのある行を探す必要はありません。
エラーコードは次のとおりです。

| | |
|---|---|
| D Data エラー | データステートメント中のデータが定められたデータ領域にない場合に表示されます |
| E Expression エラー | 式の形が間違っている場合に表示されます |
| L Label エラー | ラベルが定義されていない場合または，ラベルの二重定義がある場合に表示されます。 |
| N 未使用エラー | マクロアセンブラ（たとえば MAC）に含まれていても，現在の ASM に含まれていない機能を使ったときに，メッセージを表示します。 |
| O Overflow エラー | 式が複雑すぎる(演算子が多すぎる場合など)ために計算できない場合に表示されます。 |
| P Phase エラー | 1 パスと 2 パスにおいて，同じラベルの値が用いられなかった場合に表示されます。 |
| R Register エラー | レジスタとして定義した数値がオペレーションコードで許されていない場合に表示されます |
| S Syntax エラー | ステートメントが定型外である場合に表示されます |
| V Value エラー | 式の中のオペランドが適切な形でない場合に表示されます |

## ASMコマンドの書式について

例1では，ログインディスクのファイルをアセンブルして，HEXファイルもPRNファイルもログインディスクに生成しましたが，次のような書式で指定することもできます。

d：ASM␣ファイル名．shp

d：……ASM．COMのあるドライブ名（ログインディスク上にある場合は省略できます）
s……ソースファイルが入っているドライブ名
h……HEXファイルを生成するドライブ名
　　Zを指定するとHEXファイルは生成されません
P……PRNファイルを生成するドライブ名
　　Xを指定するとPRNファイルはコンソールに出力されます
　　Zを指定するとPRNファイルは生成されません
“shp”が省略されると，すべてログインディスクが指定されたことになります
例えば，

A>B：ASM␣DUMP．ABX

と指定すると，ドライブBにあるASM．COMプログラムを起動して，ドライブAにあるソースプログラム“DUMP．ASM”をアセンブルし，HEXファイルをドライブBに生成，PRNファイルをコンソール出力するという意味になります。

## 5 LOAD （HEX ファイルから COM ファイルへの変換コマンド）

〈コマンド形式・機能〉

LOAD␣d：filename

インテル形式のオブジェクトファイル（filename. HEX）を読み込み実行可能なマシンオブジェクトファイル（filename. COM）を生成します。

注）ログインディスクを取り扱う場合はd：を省略できます。

ASM で作成された HEX 形式のファイルはそのままでは実行されません。
LOAD コマンドは HEX ファイルを実行可能なマシン語の COM 形式ファイルに変換して同一ディスク上にセーブします。

例
直接実行可能なオブジェクトファイルの作成

アセンブラで作成された DUMP. HEX のマシンオブジェクトファイルを生成します

```
A>REN D.HEX=DUMP.HEX ↲ ―――――――――――――――――①
A>DIR D.* ↲
A: D          HEX
A>LOAD D ↲ ―――――――――――――――――――――――――②

FIRST ADDRESS 0100
LAST  ADDRESS 0212
BYTES READ    0113 ―――――――――――――――――――③
RECORDS WRITTEN 03


A>DIR D.↲ ―――――――――――――――――――――――――④
A: D          HEX : D          COM
A>
```

①同一ディスク上にすでに DUMP. COM ファイルがあるので，ファイル名を“D. HEX”に変更します。

②LOAD コマンドを実行します。

③プログラムの先頭番地が，100H 番地，
プログラムの最終番地が，212H 番地，
合計で113H バイト読まれたことを示し，
最後に，ディスクの 3 レコード＝128× 3 ＝384＝180H バイトにファイルが記録されたことを示しています。このとき残り，180H－113H＝6DH バイトには00H が書きこまれています。

④D. COM というマシン語ファイルが生成されたことを確認しています。

## 6 DDT （8080デバッガ）

〈コマンド形式・機能〉

1．DDT
　DDT プログラムを起動します。

2．DDT␣d：filename. ext
　DDT プログラムを起動して指定ファイルをメモリーにロードします。

注1）HEX ファイルはファイルに指定されたロードアドレスへ，マシンオブジェクトに変換されてロードされます。

注2）ログインディスクを取り扱う場合はd：を省略できます。

**DDT プログラムは，プログラムが正しく動くかどうかを調べるためのプログラムで，次のような豊富な機能でプログラムをテストすることができます。**

DDT コマンド一覧

| コマンド | 機能 |
|---|---|
| A | メモリーにプログラムをアセンブリ言語で書き込みます。 |
| D | メモリー内容を，16進数と ASCII コードで表示します。 |
| F | メモリーの任意のエリアを任意のデータでうめます。 |
| G | ブレークポイントを入れてプログラムを実行します。 |
| H | 16進計算をします。 |
| I | テストするファイルをファイルコントロール・ブロック*に登録します。 |
| L | メモリーの内容をアセンブリ言語の形式で表示します。 |
| M | メモリーの内容をブロック転送します。 |
| R | プログラムを読み込みます。 |
| S | メモリーの値を置きかえます。 |
| T | プログラムの実行をトレースします。 |
| U | プログラムを指定ステップ実行し最終ステップの CPU の状態のみ表示します。 |
| X | CPU のレジスタの内容の表示と変更を行ないます。 |

※）ファイルコントロール・ブロックとは，CP/M がファイルの各種操作をするために必要なファイルやそのファイルのディスク上の格納場所など，ファイルに関するすべての情報をもったブロックのことです。

なお，コマンドの詳しい形式については，付録のコマンド一覧を参照してください。

## コマンド実行例 | 例として，DUMP．COMファイルをテストします。

```
A>DDT DUMP.COM ↲ ―――――――――――――――――――――――――――――――――――――――①
DDT VERS 2.2
NEXT  PC
0300 0100
-L100 ↲ ―――――――――――――――――――――――――――――――――――――――②
  0100  LXI  H,0000
  0103  DAD  SP
  0104  SHLD 0215
  0107  LXI  SP,0257
  010A  CALL 01C1
  010D  CPI  FF
  010F  JNZ  011B
  0112  LXI  D,01F3
  0115  CALL 019C
  0118  JMP  0151
  011B  MVI  A,80
-X ↲ ―――――――――――――――――――――――――――――――――――――――③
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 LXI  H,0000
-T5 ↲ ―――――――――――――――――――――――――――――――――――――――④
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 LXI  H,0000
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0000 S=0100 P=0103 DAD  SP
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0100 S=0100 P=0104 SHLD 0215
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0100 S=0100 P=0107 LXI  SP,0257
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0100 S=0257 P=010A CALL 01C1*01C1
-S100 ↲ ―――――――――――――――――――――――――――――――――――――――⑤
0100 21 00 ↲
0101 00 ↲
0102 00 ↲
0103 39 00 ↲
0104 22 00 ↲
0105 15 00 ↲
0106 02 00 ↲
0107 31 00 ↲
0108 57 00 ↲
0109 02 00 ↲
010A CD . ↲
-D100,12F ↲―――――――――――――――――――――――――――――――――――――――⑥
0100 00 00 00 00 00 00 00 00 00 00 CD C1 01 FE FF C2 ..........ﾍﾁ."(ﾂ
0110 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 13 02 ..."(ﾍ"(ﾃQ.>"(..
0120 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2 !..jO｢.b¥Q.G}"(ﾂ
-A100 ↲ ―――――――――――――――――――――――――――――――――――――――⑦
0100  JMP 0000↲
0103  MVI A,FF↲
0105  INR A ↲
0106  NOP ↲
0107 ↲
-D100,10F ↲
0100 C3 00 00 3E FF 3C 00 00 00 00 CD C1 01 FE FF C2 ﾃ..>.<....ﾍﾁ."(ﾂ
-F110,13F,FF↲―――――――――――――――――――――――――――――――――――――――⑧
-D100,13F ↲
0100 C3 00 00 3E FF 3C 00 00 00 00 CD C1 01 FE FF C2 ﾃ..>.<....ﾍﾁ."(ﾂ
0110 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
0120 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
0130 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
-G0 ↲ ―――――――――――――――――――――――――――――――――――――――⑨
```

①DDT にファイル名を入れて起動すると，プログラムがテストできる状態になります。

②Lコマンドでメモリーの内容を逆アセンブルします。この例のように最終アドレスを指定しない場合は，11ライン分逆アセンブルします。(例では100H 番地から行なっています。)
この場合，例えば Laaaa，bbbb とするとアドレス aaaaH 番地から bbbbH 番地まで逆アセンブルします。

③Xコマンドで CPU の状態を調べます。
内容はフラグレジスタ，各レジスタ，スタックポインタ，プログラムカウンタの示すメモリーの命令を表示します。
例

| C0Z0M0E010<br>フラグレジスタ | A=00 B=0000 D=0000 H=0000<br>各レジスタ | S=0100<br>スタック<br>ポインタ | P=0100<br>プログラム<br>カウンタ |
|---|---|---|---|

④Tコマンドでプログラムを1ステップずつテストします。
表示内容は，Xコマンドと同じです。例ではプログラムを5ステップ分テストしています。

⑤Sコマンドでメモリーの内容の表示と変更をします。左側に表示されている内容をキー入力した値に変更します。例では，メモリーの100H 番地から109H 番地までデータを書き込んでいます。データの入力は1バイト入力ごとに↲を入力し，スキップするときは↲のみ入力します。
コマンドからぬけ出すには .（ピリオド）↲ を入力します。

⑥Dコマンドでメモリーの内容を表示します。表示は左側が16進表示，右側がアスキー表示となっています。例では100H 番地から12FH 番地までを表示しています。
この場合，終了アドレスを省略すると12行分ダンプします。

⑦Aコマンドでメモリーにプログラムをアセンブリ言語で入力します。1つ命令を入力するごとに↲を入力し，コマンドからぬけるには↲のみを入力します。
例では100H 番地から106H 番地までアセンブリ言語で入力し，Dコマンドで入力プログラムを確認しています。

⑧Fコマンドでメモリーのある領域を指定データで満たします。例では110H番地から13FH 番地までを FFH で満たし，Dコマンドで確認しています。

⑨Gコマンドでプログラムを実行します。例のように0番地を実行するとリブートして CP/M システムに戻ります。

## 7 DUMP （ディスクファイルの16進出力コマンド）

〈コマンド形式・機能〉

DUMP␣d：filename. ext

ファイルの内容を16進コマンドでコンソールに出力します。

注）ログインディスクを取り扱う場合はd：を省略できます。

**COM 形式のファイルなどは，TYPE や PIP で内容を表示させても完全には表示されません。DUMP を使用すれば次の例のように表示させることができます。**

**DUMP コマンドで表示した例**

例では，DUMP. COM，つまり DUMP 自身のファイルを16進コード出力しています。

```
A>DUMP DUMP.COM ↵

0000 21 00 00 39 22 15 02 31 57 02 CD C1 01 FE FF C2
0010 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 13 02
0020 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2
0030 44 01 CD 72 01 CD 59 01 0F DA 51 01 7C CD 8F 01
0040 7D CD 8F 01 23 3E 20 CD 65 01 78 CD 8F 01 C3 23
0050 01 CD 72 01 2A 15 02 F9 C9 E5 D5 C5 0E 0B CD 05
0060 00 C1 D1 E1 C9 E5 D5 C5 0E 02 5F CD 05 00 C1 D1
0070 E1 C9 3E 0D CD 65 01 3E 0A CD 65 01 C9 E6 0F FE
0080 0A D2 89 01 C6 30 C3 8B 01 C6 37 CD 65 01 C9 F5
0090 0F 0F 0F 0F CD 7D 01 F1 CD 7D 01 C9 0E 09 CD 05
00A0 00 C9 3A 13 02 FE 80 C2 B3 01 CD CE 01 B7 CA B3
00B0 01 37 C9 5F 16 00 3C 32 13 02 21 80 00 19 7E B7
00C0 C9 AF 32 7C 00 11 5C 00 0E 0F CD 05 00 C9 E5 D5
00D0 C5 11 5C 00 0E 14 CD 05 00 C1 D1 E1 C9 46 49 4C
00E0 45 20 44 55 4D 50 20 56 45 52 53 49 4F 4E 20 31
00F0 2E 34 24 0D 0A 4E 4F 20 49 4E 50 55 54 20 46 49
```

アドレス部分　　データ部分

・DUMP コマンドのアドレス部分の表示は，0000H から始まります。
すべてのファイル形式について使用可能となっています。

・コンソール出力の途中で何かキーを入力すると CP/M システムに戻ります。

・表示を一時停止するには CTRL ＋ S，表示を再開するにはもう一度 CTRL ＋ S を入力します。

# 8 SUBMIT （バッチ処理コマンド）

〈コマンド形式・機能〉

1．SUBMIT␣filename

サブミットファイル "filename. SUB" の内容をバッチ処理します。

2．SUBMIT␣filename␣P1␣P2␣P3 ……

P1, P2……はサブミットファイル中の代用パラメータで S1, S2……に代入されてバッチ処理が行なわれます。

注1）バッチ処理とは，ある量の仕事を集めておいてから一括して処理する方法のことです。

注2）サブミットファイルはドライブ A になければなりません。

SUBMIT コマンドは，いくつかの CP/M コマンドや COM 形式ファイルを実行順に並べて作業手順を設定しておくことで，自動的に作業を行なってくれるコマンドです。作業手順の設定は WM（ワードマスター）あるいは ED でサブミットファイル（filename. SUB）を作成して行ないます。

**例 1 作業手順の設定**

あらかじめサブミットファイルを作成しておきます。ファイル名は任意のものをつけられますが，エクステンションは必ず "SUB" を指定します。

```
A>TYPE EX0. SUB ↲ ——①
DIR DEX. *
PIP DEX. ASM=DUMP. ASM
ASM DEX
LOAD DEX
DIR DEX. *

A>
```

①はあらかじめ WM（あるいは ED）で作成した作業手順を TYPE コマンドでタイプアウトしたものです

**例 2 SUBMIT の実行例**

作業手順 EX0. SUB を SUBMIT コマンドで自動処理している実行例です。

```
A>SUBMIT EX0 ↲


A>DIR DEX. *
NO FILE
A>PIP DEX. ASM=DUMP. ASM

A>ASM DEX
CP/M ASSEMBLER - VER 2.0
0257
001H USE FACTOR
END OF ASSEMBLY

A>LOAD DEX

FIRST ADDRESS 0100
LAST  ADDRESS 0212
BYTES READ    0113
RECORDS WRITTEN 03


A>DIR DEX. *
A: DEX      ASM : DEX      PRN : DEX      HEX : DEX      COM
A>
```

## 例3 パラメータを使用した場合

例2では特定のファイルに対してしかSUBMITを実行することができませんでしたが，この例の$1，$2のような記号を使用することにより任意のファイルに対して作業を行なうことができます。

```
A>TYPE EX1.SUB ↵
DIR $1.*
PIP $1.ASM=$2.ASM
ASM $1
LOAD $1
DIR $1.*

A>
```

$1と$2にはSUBMIT実行時に指定するファイル名が入ります。

$1に"DEX1"，$2に"DUMP"を指定して実行すると，例2と全く同じ作業を行ないます。

```
A>SUBMIT EX1 DEX1 DUMP ↵

A>DIR DEX1.*
NO FILE
A>PIP DEX1.ASM=DUMP.ASM

A>ASM DEX1
CP/M ASSEMBLER - VER 2.0
0257
001H USE FACTOR
END OF ASSEMBLY

A>LOAD DEX1

FIRST ADDRESS 0100
LAST  ADDRESS 0212
BYTES READ    0113
RECORDS WRITTEN 03


A>DIR DEX1.*
A: DEX1      ASM : DEX1      PRN : DEX1      HEX : DEX1      COM
A>
```

## 9 XSUB （拡張バッチ処理コマンド）

〈コマンド形式・機能〉

ダイレクト実行モードはありません。
SUBMIT の拡張コマンドとして，サブミットファイルの最初に宣言文のような形式で入ります。

**XSUB コマンドは，トランジェントコマンド内のキー入力手順の指示までも自動処理することができるコマンドで，サブミットファイルの最初に入れておきます。**

例
XSUB の実行例

```
A>TYPE EXX.SUB
XSUB
DDT $1
D100,13F
L100
G0

A>
```

①

①はエディタであらかじめ作成した作業手順をTYPE コマンドでタイプアウトしたものです。このファイルでは最初に XSUB を宣言したことによって，DDT 内の D，L，G といった各コマンドまで指定できるようになっています。これらのコマンドも SUBMIT コマンドで自動的に処理されます。

```
A>SUBMIT EXX DUMP.COM

A>XSUB

A>DDT DUMP.COM
DDT VERS 2.2
NEXT  PC
0300 0100
-D100,13F

0100 21 00 00 39 22 15 02 31 57 02 CD C1 01 FE FF C2 !..9"..1W.ﾍﾁ."(ﾂ
0110 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 13 02 ..."(ﾍ"(ﾃQ.>"(..
0120 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2 !..jOﾄ.b¥Q.G}"(ﾂ
0130 44 01 CD 72 01 CD 59 01 0F DA 51 01 7C CD 8F 01 D.ﾍr.ﾍY..ﾚQ.|ﾍ"(
-L100

  0100  LXI  H,0000
  0103  DAD  SP
  0104  SHLD 0215
  0107  LXI  SP,0257
  010A  CALL 01C1
  010D  CPI  FF
  010F  JNZ  011B
  0112  LXI  D,01F3
  0115  CALL 019C
  0118  JMP  0151
  011B  MVI  A,80
-G0


(xsub active)
A>
```

## 10 MOVCPM （CP/M システムのメモリーサイズ変更コマンド）

〈コマンド形式・機能〉

MOVCPM␣n␣＊

CP/M システムの BIOS サイズを変更する場合に使用するコマンドです。

n：CP/M システムのメモリーサイズ（20≦n≦56）

MOVCPM は，アプリケーションの開発において，CP/M の BIOS サイズを変更する必要がある場合などに使用するコマンドです。

MOVCPM を使用する場合には本システムの BIOS ソースプログラムである〝BIOS. MAC〟を変更する必要があります。この BIOS. MAC は Z80ニーモニック表記のアセンブラで書かれておりますので，これをアセンブルし，リロケートするためには Z80ニーモニック用のアセンブラ，リンカが必要となります。（turbo CP/M にはインテルニーモニック用アセンブラ〝ASM. COM〟のみ付属しております。）MOVCPM の具体的な使用方法は以下のとおりです。

①ワードマスターを使用して，BIOS ソースプログラムである〝BIOS. MAC〟の BDOS や CCP，BIOS のエントリーアドレス及び CP/M サイズを変更します。（ワードマスターの使用方法については，第13章「ワードマスター」を参照してください。）
変更ケ所についてはP.79を参照してください。

まず，ワードマスターを起動します。

A>WM␣BIOS. MAC ↵

と入力してください。ワードマスターのメッセージ画面がでてきますので，読み終えたら何かキーを入力してください。
BIOS. MAC のプログラムが表示され編集可能な状態になります。

②まず，Z80ニーモニック用アセンブラを使用して①で変更したBIOS.MACファイルをアセンブルし，次にZ80ニーモニック用リンカを使用して，CP/Mシステムで，実行可能なBIOS.COMファイルをディスクに作成します。
ドライブBに「MACRO−80」（マイクロソフト社）をセットした実行例を示します。

```
A>B:M80 =BIOS ↵
  ⟨
A>B:L80 /P:100,BIOS,BIOS/E/N ↵
  ⟨
```

③MOVCPMコマンドを使用して，CP/Mシステム（CCP,BDOS部分など）をTPA（トランジェントプログラムエリア）にリロケートします。ここでリロケートされたCCPはメインメモリ上の0980H〜117FHに，BDOSは，1180H〜1F7FHに配置されます。また，BIOS（現段階では使用不可能なため，④のディスクセーブの対象にはなりません。）については，1F80H以降に配置されます。

```
A>MOVCPM 52 * ↵

CONSTRUCTING 52k CP/M vers 2.2
READY FOR "SYSGEN" OR
"SAVE 34 CPM52.COM"
```

④SAVEコマンドを使用して③でリロケートされたCP/Mシステムをディスクにセーブします。なお，③でリロケートされたCP/Mシステム（BIOS部は除きます）は，メインメモリの0100H〜1F7FH(この中にはCCP，BDOSの他に，SYSGENコマンド等のプログラムエリアも含まれます）に存在しているので，セーブするファイルのページ数は34ページ（1ページ=256バイト）になります。

```
A>SAVE 34 CPM52.COM ↵
```

⑤まずDDTコマンドを起動して，④でセーブしたファイルをメモリーに読み込みます。次にDDTのIコマンドでメモリ上のFCB(ファイルコントロールブロック)に，②で作成したBIOS.COMファイルを登録します。その後，DDTのRコマンドでメインメモリ上の1F80Hより BIOS.COMファイルを読み込みます。
（ただし，リンカを使用してBIOS.COMを0100Hより実行できるようにした場合は，1E80Hよりプログラムを読み込ませます。）

以上でTPA領域にメモリーサイズを変更した新しいCP/Mシステムが生成されます。

```
A>DDT CPM52.COM ↵
DDT VERS 2.2
NEXT  PC
2300 0100
-IBIOS.COM ↵
-R1E80 ↵
NEXT  PC
2F00 0100
-G0 ↵
```

⑥⑤で生成されたCP/Mシステムを SAVE コマンドでディスクにセーブします。なお，生成されたCP/Mシステムは，CCP, BDOS, BIOS（その他に，SYSGEN コマンド等のプログラムエリアも含まれます）から成っており，セーブするファイルのページ数は52ページになります。

```
A>SAVE 52 CPM52.SYS ↵
```

⑦最後に，SYSGEN コマンドで，メモリーサイズを変更した新しいCP/Mシステムをディスクに生成することができます。

```
A>SYSGEN CPM52.SYS ↵
```

BIOS.MACファイルは，次の4ケ所を変更します。下のリストと表を参照してください。

```
;
BDOSJP  EQU     0CC06H ----------①
CCP     EQU     0C400H ----------②
;
        .PHASE  0DA00H ----------③
;
TTLMES: DB      '56K CP/M  Version 2.2 for SHARP X1 turbo [Rel. 1.0]',13,10
        DB              Copyright (C) by Digital Research',13,10
                        ----------④
```

| CP/Mシステムサイズ n | ①BDOSJPエントリーアドレス | ②CCPエントリーアドレス | ③BIOS PHASEエントリーアドレス | ④CP/Mサイズ |
|---|---|---|---|---|
| 56 | 0CC06H | 0C400H | 0DA00H | 56K |
| 55 | 0C806H | 0C000H | 0D600H | 55K |
| 54 | 0C406H | 0BC00H | 0D200H | 54K |
| 53 | 0C006H | 0B800H | 0CE00H | 53K |
| 52 | 0BC06H | 0B400H | 0CA00H | 52K |

変更が終わりましたら，[ESC]を押した後，[E]を入力してワードマスターを終了します。

※メモリーサイズをn（n=20～56までの整数）キロバイトとすると各エントリーアドレスは次の式で計算できます。

CCP=(n−20)×1024+3400H

BDOSJP=CCP+806H

BIOS=CCP+1600H

# 第7章

# turbo CP/M強化コマンド

# 1 STARTUP（turbo CP/M 起動時の初期設定）

**STARTUP コマンドは，turbo CP/M 起動時に自動的に実行されるプログラムで，下記の 7 項目の各種パラメータの初期設定を行ないます。また，設定内容の変更も行なえます。**

1．ファンクションキーの設定
2．先行キー入力（STARTUP 実行後，自動的にタイプされるキー入力）の設定
3．RS-232C のパラメータ設定
4．プリンタ設定
5．ユーザー DPB のパラメータ設定
6．初期 IOBYTE の設定
7．非漢字モード，漢字モード（KMODE 0，1）の設定

## STARTUP の実行

STARTUP のコマンドは起動と同時に実行されますが，もし CP/M 使用中にもう 1 度初期設定が必要になった場合は，STARTUP↵と入力すれば実行されます。

## データの設定変更

設定データの変更はメニュー選択方式になっており，変更の必要な項目だけを変更できるようになっています。各項目について，変更方法を説明します。

1. STARTUP␣＊ ↵と入力します。

2. スタートアップユーティリティが起動し，下のようにメニュー画面が表示されます。

```
1・・・ファンクションキー設定
2・・・先行キー設定
3・・・RS-232Cパラメータ設定
4・・・プリンタ設定
5・・・ユーザーDPB設定
6・・・ＩＯバイト設定
7・・・漢字／非漢字モード設定
8・・・CP/Mに戻る

変更したい番号を入力してください。
```

3. 設定変更をする項目を選び [1] ～ [8] のキーを入力します。

### 3-1 ファンクションキーの設定

①メニュー画面で [1] を入力すると現在のファンクションキー設定内容が表示されます。

```
        現在の設定

    KEY 1= DIR@M
    KEY 2= STAT *.*@M
    KEY 3= PIP
    KEY 4= TYPE
    KEY 5= SUBMIT
    KEY 6= DIR
    KEY 7= STAT
    KEY 8= KEY
    KEY 9= REN
    KEY 0= DDT

  再設定したい番号を入力してください．（リターンキーでメニューに戻ります）
```

②再設定したいキーの番号を入力します。[0] ～ [9] のキーを入力してください。
例として [1] を入力し，[F1] の変更をします。

```
  KEY1  を設定します
  >
```

と，メッセージがでます。

③変更したい内容を入力します。例えば STARTUP␣ * ↵と設定する場合は

```
  KEY1  を設定します
  >STARTUP *@M
```

と入力します。リターンキー（[CTRL] + [M]）の動作を行ないたい場合は上の例のように@M と入力してください。

④変更したい内容を入力したら↵を押します。画面に表示されている KEY 1 の定義内容が変更されます。

```
        現在の設定

    KEY 1= DIR@M
    KEY 2= STAT *.*@M
    KEY 3= PIP
    KEY 4= TYPE
    KEY 5= SUBMIT
    KEY 6= STARTUP *@M
    KEY 7= STAT
    KEY 8= KEY
    KEY 9= REN
    KEY 0= DDT

  再設定したい番号を入力してください．（リターンキーでメニューに戻ります）
```

⑤他のキーも変更する場合は②の作業から行ないます。↵を入力するとメニュー画面へ戻ります。

### 3-2 先行キー入力の設定

STARTUP プログラム実行後に，続けて実行されるキー入力を設定するものです。

①メニュー画面で [2] を入力すると，

```
設定したい先行キーを入力してください。（^Zで入力終了）

A>
```

とメッセージがでます。

②例えば DIR ↵とすると，STARTUP 実行後すぐにドライブAのディスクのディレクトリが表示されるようになります。

```
設定したい先行キーを入力してください。（^Zで入力終了）

A>DIR↵
```

③ [CTRL] + [Z] で設定を終了し，メニュー画面へ戻ります。

### 3-3 RS-232C パラメータ設定

①メニュー画面で [3] を入力すると，下のような画面になります。

```
ＲＳ－２３２Ｃポートの設定

  １・・・オープン
  ２・・・クローズ

番号を選択してください。
```

1…オープンの設定では RS-232C の DTR，RTS の信号が ON の状態になり TTY：に入出力が可能となります。出荷時にはこの状態になっています。

2…クローズの設定では上記の信号が OFF の状態となり，TTY：に入出力が不可能となります。なお，TERM コマンドを実行した場合，この設定は解除されオープンの状態になります。

[1] か [2] を入力してください。

[1] を選択後，ボーレート，通信制御，ストップビット，パリティ，キャラクタ長の設定を行ないます。これらについては第7章の11項（TERM）を参照してください。

最後の項目まで設定を終えるとメニュー画面に戻ります。

[2] を選択した場合，メニュー画面に戻ります。

### 3-4 プリンタ設定

①メニュー画面で 4 を入力すると，下のように表示されます。

```
            プリンタ設定

A...CZ-800P                    K...ESC/P-ext
B...CZ-80PK                    L...RP-80II
C...CZ-8PD2/SP80               M...FP-80
D...CZ-8PP2                    N...UP-130K
E...CZ-8PK2                    O...PC-8822/PC-PR201
F...CZ-8PN1/CZ-8PK3(4)         P...M-1009X
G...MZ-1P06                    Q...M-1009
H...MZ-1P07                    R...LPR-24T
I...MZ-1P08                    S...KP-3000
J...MZ-1P10A                   T...OTHER

Z...終了

    プリンタを選択してください。
```

②A ～ T のキーを入力してプリンタを選択します。

③Z を入力するとメニュー画面に戻ります。

〈ご注意〉

・MZ-1P17は CZ-8PN1に設定してご使用ください。

・Tの OTHER には登録されていないプリンタを１機種設定することができます。ただし，登録するためにはアセンブラソースを変更する必要があります。詳しくは付録の３項を参照してください。

### 3-5 ユーザー DPB の設定

ユーザー DPB は，ユーザー指定のドライブを設定するためのものです。

①メニュー画面で 5 を入力すると，下のような画面になります。

```
ユーザーDPBの各パラメータ（１６進）を設定してください。
リターンキーでスキップします。
Secter per track          =$0040
```

②パラメータの値を変更する場合は，16進コードで入力し ↵ を押してください。（変更しない場合はそのまま ↵ を押すと次のパラメータにスキップします。）

③下の画面のように10項目設定すると自動的にメニューへ戻ります。

```
ユーザーDPBの各パラメータ（１６進）を設定してください。
リターンキーでスキップします。
Secter per track          =$0040
Block shift facter        =$04
Block mask                =$0F
Extent mask               =$01
Disk size max             =$0097
Directory entries max     =$003F
ALLOC0                    =$80
ALLOC1                    =$00
Check vecter size         =$0010
Track offset              =$0002
```

※ DPB とはディスク・パラメータ・ブロックのことです。詳しくは第12章 4 項を参照してください。

### 3-6 IOバイト設定

IOバイトの設定は，周辺装置の割付けを行なうものです。

①メニュー画面で 6 を入力すると下のように表示されます。

```
            ＩＯバイト設定

LST:=[0)..TTY:  1)..CRT:  2)..LPT:  3)..UL1:  ] ?
PUN:=[0)..TTY:  1)..PTP:  2)..UP1:  3)..UP2:  ]
RDR:=[0)..TTY:  1)..PTR:  2)..UR1:  3)..UR2:  ]
CON:=[0)..TTY:  1)..CRT:  2)..BAT:  3)..UC1:  ]

設定したい物理デバイス番号を入力してください。
```

turbo CP/M では，シアンで表示されている物理デバイスをサポートしています。

turbo CP/M 出荷時の設定は，

LST：=LPT：
RUN：=TTY：
RDR：=TTY：
CON：=CRT：

となっています。

②LST：から順に設定したい物理デバイスの番号を入力します。
（リターンキーでスキップします。）４つのデバイスを設定すると，自動的にメニューへ戻ります。

※ IOバイトについては第６章１項，第12章３項を参照してください。

### 3-7　漢字/非漢字モード設定

①メニュー画面で 7 を入力すると下のような画面になります。

```
画面モードの設定

  1・・・漢字モード
  2・・・非漢字モード


設定したいモードの番号を選択してください。
```

②どちらかのモードの番号を入力します。自動的にメニュー画面へもどります。

※漢字/非漢字モード設定は表示文字のタイプの設定です。漢字モードでは2バイトコード文字（全角文字）と，半角文字（セミグラフィック文字は除く）が表示可能です。非漢字モードでは1バイトコード文字（半角文字）が表示可能です。

### 3-8　CP/M に戻る

①設定を終了したら，メニュー画面で 8 を入力します。

```
設定データをディスクに登録しますか (Yes/No)?
```

とメッセージが表示されます。設定データをセーブするときは Y と押します。ドライブAがアクセスされ，STARTUP のデータが書きかえられます。

②書き込みが終了すると，CP/M システムへ戻り，画面上にプロンプトがあらわれます。なお，再起動した場合の初期設定はディスク上に登録された，STARTUP の設定となります。

## 2 FORMAT（フロッピーディスクのフォーマット）

FORMATコマンドは，指定ディスクドライブのフロッピーディスクのデータをすべて消去し、フロッピーディスクを初期状態にします。新しいフロッピーディスクを使用する場合や他で使用していたフロッピーディスクを再使用する場合には，まず最初に必ずFORMATコマンドを実行し，フロッピーディスクを初期化してください。

**FORMATの実行**

①FORMAT ↵と入力します。

②

```
フォーマットを行なうドライブ名を入力してください ?
（使用可能なドライブ名はA～Hです。）
```

とたずねてきますので，ドライブ名を入力します。

③例えば5インチ2Dのドライブで行なう場合，

```
デバイスタイプは 5inch 2D（320K），デバイス番号は 1 です。
フォーマットを開始しますか（Yes/No）?
```

と確認してきますので，正しければ[Y]を入力しフォーマットを実行します。誤まりがある場合は[N]と入力し，②の表示にもどります。

④フォーマット実行中はトラック番号とサイド番号が表示されており，フォーマットの経過がわかります。

⑤フォーマットが完了すると，

```
－－－－終了しました－－－－
フォーマットを続けますか（Yes/No）?
```

と表示がでます。次のフロッピーディスクをフォーマットする場合は[Y]と入力して，②以下を同様にくり返してください。フォーマットの作業を終了したいときは[N]と入力して，CP/Mシステムへ戻ります。

※コマンド入力時にドライブ名を指定する方法

FORMATコマンドでは，コマンド入力時にドライブ名を指定することもできます。

FORMAT␣ドライブ名：↵

と入力すると，上記③の状態となります。あとは同じようにメッセージに従って操作してください。

**〈ご注意〉**

・フォーマット中にエラーがでた場合は，フロッピーディスクの不良が考えられますので，別のフロッピーディスクと交換して，もう一度実行し直してください。

・FORMATコマンド実行中に中断するときは，入力待ちの状態で[CTRL]＋[C]（あるいは[SHIFT]＋[BREAK]）を押すと，ただちにCP/Mシステムへ戻すことができます。

・誤って重要なデータを消去してしまうことのないよう十分に注意しましょう。

## 3 COPY（オールコピー）

COPYコマンドは，turbo CP/M用にフォーマットされたフロッピーディスクの内容をすべて他のフロッピーディスクにコピーするコマンドです。これは，フロッピーディスク上のファイルだけでなく，turbo CP/Mのシステムも含めてコピーし(turbo CP/Mのシステムが入っている場合)，オリジナルと全く同一のコピーを作成するというものです。

### 3－1　ディスクドライブが2基の場合

**COPYの実行**

①COPY ↵と入力します。

②

```
ソースドライブ名                を入力してください ?
（使用可能なドライブ名はA～Hです。）
```

とたずねてきますので，コピーしたいソースディスク（もとになるディスク）をディスクドライブに入れ，そのドライブ名を入力します。

③

```
ディスティネーションドライブ名を入力してください ?
（使用可能なドライブ名はA～Hです。）
```

とたずねてきますので，コピーをとるフロッピーディスク（必ずフォーマットしたものを使用する）をディスクドライブに入れ，そのドライブ名を入力します。

④例えばソースディスクをドライブA，ディスティネーションディスクをドライブBにセットしてある場合，

```
ドライブ A から  ドライブ B へコピーします。
コピーを開始しますか（Yes/No）?
```

と表示されますので，ドライブ名を確認し，正しければ[Y]と入力しコピーを実行します。誤まりがある場合は[N]を入力して，②の表示へもどります。

⑤コピー実行中は，処理の経過がトラック番号で示されています。

⑥コピーが終了すると，

```
－－－－終了しました －－－－
コピーを続けますか（Yes/No）?
```

とメッセージが表示されます。コピーを続けたいときは[Y]を入力し，別のフロッピーディスクをセットして，②からの操作を再度行なってください。コピーの作業を終了したいときは[N]を入力してCP/Mシステムへもどります。

①COPY ↵と入力します。

②

```
ソースドライブ名                    を入力してください ?
（使用可能なドライブ名はA～Hです。）
```

とたずねてきますので，コピーしたいソースディスク（もとになるディスク）をディスクドライブに入れ，そのドライブ名を入力します。

③

```
ディスティネーションドライブ名を入力してください ?
（使用可能なドライブ名はA～Hです。）
```

とたずねてきますので，②と同じドライブ名を入力します。

④例えばドライブＡで行なう場合，

```
ドライブ A から  ドライブ A へコピーします。
コピーを開始しますか（Yes/No）?
```

と表示されますので，ドライブ名を確認し，正しければ[Y]を入力しコピーを実行します。誤まりがあった場合は[N]を入力して，②の表示へもどります。

⑤

```
ソースディスク                    をドライブAにセットしてリターンキーを押してください
```

とメッセージが表示されたら，ソースディスクがドライブに入っていることを確かめて，↵キーを押します。トラック番号０から１までのデータが読み込まれます。

⑥

```
ディスティネーションディスクをドライブAにセットしてリターンキーを押してください
```

とメッセージが表示されたら，コピーを作るフロッピーディスク（必ずフォーマットしたもの）をソースディスクと入れかえ，↵を入力します。トラック番号０から１にデータが書き込まれます。

⑦再び⑤のメッセージにもどります。２トラックを１単位として処理をしますので，最終トラック（39トラック）まで処理するためには⑤と⑥の作業と20回繰り返して行ないます。
います。
処理の経過とともに，トラック番号が画面に表示されます。

⑧コピーが終了すると，

```
-----終了しました -----
コピーを続けますか（Yes/No）？
```

とメッセージが表示されますので，コピーを続けたいときは[Y]を入力し，別のフロッピーディスクをセットして②からの操作を再度行なってください。コピー作業を終了したいときは，[N]を入力してCP/Mシステムへもどります。

〈ご注意〉

・ディスティネーションディスクは，必ずフォーマットしてあるフロッピーディスクを使用してください。

・COPYコマンド実行中に中断したいときは，入力待ちの状態でマスターディスクをドライブにセットし，[CTRL]＋[C]（あるいは[SHIFT]＋[BREAK]）を押してください。ただちにCP/Mシステムへもどることができます。

## 4 SYSGEN （CP/Mシステムのコピー）

SYSGEN コマンドは，CP/M システムをあらたにフロッピーディスクに生成するコマンドです。
CP/M システムは，CPM56. SYS というファイル名で入っています。

### SYSGENの実行

SYSGEN ↵と入力し，コマンドを起動します。

〈実行例〉

```
システムを読み込むドライブ名を入力してください。A ―――――――― ①
（リターンキーでスキップします。）

ドライブＡから読み込みを開始します。―――――――― ②
準備ができたなら，リターンキーを入力してください。

終了しました。 ―――――――― ③

システムを書き込むドライブ名を入力してください。B ―――――――― ④
（リターンキーでリプートします。）

ドライブＢに書き込みを行います。―――――――― ⑤
準備ができたなら，リターンキーを入力してください。

終了しました。 ―――――――― ⑥

システムを書き込むドライブ名を入力してください。―――――――― ⑦
（リターンキーでリプートします。）
```

例ではドライブＡのシステムをドライブＢにコピーします。

① システムの入っているドライブ名をきいてきますのでドライブ名を入力します。(ドライブ名はＡ～Ｈまで指定できます。)

② 入力したドライブ名を確認してきます。準備ができたら ↵を入力します。

③ システムの読み込みが終了しました。

④ システムを書き込むドライブ名をきいてきますのでフロッピーディスクをセットして，そのドライブ名を入力します。(ドライブ名はＡ～Ｈまで指定できます。)

⑤ ドライブ名を確認してきます。準備ができたら ↵を入力します。

⑥ コピーが終了したことを知らせています。

⑦ 続けてコピーを行なう場合は，フロッピーディスクを交換して書き込むドライブ名を入力してください。
終了する場合は ↵を入力してください。

※ログインディスク上に CPM56. SYS というファイルがある場合は，

SYSGEN␣CPM56. SYS↵

と入力すると SYSGEN コマンドを起動してから自動的にシステムを読み込み，実行例の手順④以後から行なうことができます。

## 5 DICCOPY （日本語処理用の辞書コピー）

**DICCOPY コマンドは，CP/M で音訓辞書，システム辞書，ユーザー辞書を使用するために，辞書のコピーを行なうコマンドです。**

turbo CP/M で使用いただける辞書は，音訓辞書，システム辞書，ユーザー辞書の 3 種類です。BASIC 上でご使用の辞書はそのままではご使用になれません。CP/M フォーマットを行なったディスクにこのコマンドを使ってコピーしたものをご使用ください。
なお，音訓辞書は "ONKUN. DIC" というファイル名で turbo CP/M マスターディスクに入っています。

〈ご注意〉

・このコマンドはドライブ 1 基ではご使用いただけません。
・DIC ファイルは PIP コマンドではなく，必ず DICCOPY で転送してください。

### DICCOPY の実行

①DICCOPY ↵と入力します。

②コピーしたい辞書をきいてきますので番号を入力します。

```
どの辞書をコピーしますか？

      1...音訓    辞書
      2...システム辞書
      3...ユーザー辞書
      4...CP/M  に戻る

番号で入力してください ？
```

③コピーの種類をきいてきます。

```
コピーの種類はどれですか？

      1...CP/M  → CP/M
      2...BASIC → CP/M
      3...CP/Mに戻る

番号で入力してください ？
```

転送元がすでにこのコマンドを使用し CP/M フォーマットで作成された辞書であれば，[1]と入力し，BASIC 上で使用している辞書であれば[2]と入力してください。

④辞書を読み込むドライブと書き込むドライブを順にきいてきます。それぞれA～Hのドライブを入力して ↵ を押します。この場合，転送元と転送先を同じデバイスにすることはできません。ドライブ名を入力するとコピーを開始するかどうかきいてきますので設定の変更がなければ Y と入力してください。
N と入力すると手順①に戻ります。

```
辞書を読み込むドライブ名を入力してください ? A
辞書を書き込むドライブ名を入力してください ? B
コピーを開始しますか (Yes/No)
```

上の例では転送先をドライブBに指定しています。この場合，ドライブAには，コピーしたい辞書の入っているフロッピーディスクをセットしてドライブBにはCP/Mフォーマットを行なっているフロッピーディスクをセットしてください。

⑤コピー中は次のように表示されます。

```
コピー中です。しばらくお待ちください。
```

⑥終了すると次のように表示されます。

```
－－－－終了しました－－－－
処理を続けますか (Yes/No) ?
```

再度コピーを行なう場合は Y と入力してください。　手順①から繰り返します。
終了する場合は N と入力してください。 CP/Mシステムに戻ります。(この場合，ドライブAにCP/Mのシステムディスクが入っていないとリブートエラーになりますのでご注意ください。)

## エラーメッセージ

※転送元に指定した辞書が入っていないと

```
辞書が見つかりません。
    処理を続けますか (Yes/No) ?
```

と表示されます。 Y と入力すると手順①に，N と入力するとCP/Mシステムに戻ります。

※転送先のフロッピーディスクの容量が足りないと，

```
ディスクの容量がたりません。
    処理を続けますか（Yes/No）？
```

と表示されます。（CP/M マスターディスクには容量の関係でシステム辞書をコピーすることはできません。）処理を続けるときは，別のフロッピーディスクと差しかえて Y と入力し手順①に，入力するときは N と入力して CP/M システムに戻ります。

※転送先に指定した辞書がすでに存在している場合は，

```
辞書がすでに存在しています。
    処理を続けますか（Yes/No）？
```

と表示されます。Y と入力すると手順①に，N と入力すると CP/M システムに戻ります。

## 6 DICSET （辞書ディスクの割付け）

DICSET コマンドは，CP/M で音訓辞書，システム辞書，ユーザー辞書を使用する際，辞書の入っているドライブを設定するコマンドです。

### DICSET の実行

①DICSET ↵と入力します。

②辞書の入っているドライブを順にきいてきます。

設定の必要のない辞書についてはリターンキーを押し，スキップしてください。最後に，設定の確認を行ないますので正しければ Y と入力してください。N と入力すると再度設定することができます。（CTRL +C で CP/M に戻ります。）

```
辞書の入っているドライブを指定してください。
        （リターンキーでスキップします。）

音訓     辞書の入っているドライブ...?
システム辞書の入っているドライブ...?
ユーザー辞書の入っているドライブ...?

設定はよいですか（Yes/No）?
```

③②で Y と入力すると指定のディスクに正しい辞書がセットされているかどうか調べ設定が正しければ設定の終了を知らせます。

指定したディスクに辞書がセットされていないと，例えば次のように表示されます。

```
音訓     辞書（ONKUN.DIC） がみつかりません。

再設定しますか（Yes/No）?
```

このとき，Y を入力すると②に戻り，N を入力するとCP/M に戻ります。

## 7 DISKSET （使用ディスクの割付け）

**DISKSET コマンドは，A〜H のドライブに物理デバイス，デバイス番号，デバイスタイプを割付けるコマンドです。**

turbo CP/M で使用できる物理デバイスとデバイスタイプについては，第5章のd：（ログインディスク変更コマンド）の項で述べたとおりですが，これらの変更を行なうときは以下のような手順で行なってください。

### DISKSET の実行

①DISKSET ↵と入力します。

②メニュー画面が表示されます。

```
        1 . . . . . 設定状況一覧
        2 . . . . . 設定変更
        3 . . . . . CP/Mに戻る

  メニュー番号を入力してください
```

まず，現在の設定状況を確認するために 1 を入力します。

③設定状況の一覧が表示されます。

```
CP/Mドライブ    物理デバイス    デバイス番号    タイプ
    A:          5inch FD          0          2D
    B:          5inch FD          1          2D
    C:          5inch FD          2          2D
    D:          5inch FD          3          2D
    E:          EMM               0          2D
    F:          EMM               1          2D
    G:          MEM               0          -----
    H:          5inch FD          1          user

  何かキーを入力してください
```

確認したら，何かキーを入力してください。

④例ではドライブＡを５インチの２ＨＤに変更します。メニュー画面で 2 を入力します。

```
設定変更したいドライブ名を入力してくださいA          ―――― 4－1
   （使用可能なドライブ名は  Ａ～Ｈ  です）

物理デバイス番号を下の番号より選んでください3       ―――― 4－2
[1･･･MEM  2･･･EMM  3･･･5inch FD  4･･･8inch FD  5･･･HARD DISK]

デバイス番号を入力してください（０～３）？0         ―――― 4－3

デバイスタイプを下の番号より選んでください3         ―――― 4－4
 [1･･･2D  2･･･2DD  3･･･2HD  4･･･USER]

ドライブ Ａ を 5inch FD  の 2HD  に変更します      ―――― 4－5
 変更してもよいですか（Yes / No）？
```

4－1　設定変更したいドライブ名を入力します。

4－2　物理デバイス番号を入力します。

4－3　デバイス番号を入力します。

4－4　デバイスタイプの番号を入力します。※

4－5　以上の設定でよければ Y を入力します。N を入力するとメニュー画面に戻ります。

4－1から4－5で ↵だけを入力するとメニュー画面に戻ります。

⑤ 4－5で Y を入力すると変更した設定が表示されます。

```
CP/Mドライブ    物理デバイス    デバイス番号    タイプ
    A:          5inch FD          0           2HD
    B:          5inch FD          1           2D
    C:          5inch FD          2           2D
    D:          5inch FD          3           2D
    E:          EMM               0           2D
    F:          EMM               1           2D
    G:          MEM               0           -----
    H:          5inch FD          1           user

 何かキーを入力してください
```

ドライブＡのデバイスタイプは２ＨＤに変更されていることがわかります。
何かキーを入力するとメニュー画面に戻ります。

⑥変更が終了したらメニュー画面で 3 を入力します。

※４－４を入力した時点で同一デバイスがすでに存在している場合は

```
同一デバイスが存在しています。
処理を続けますか（Yes/No）？
```

と表示されます。Y を入力すると，同一デバイスは削除され，設定がシフトします。
N を入力するとメニュー画面に戻ります。

⑦変更データをシステムトラックに登録するかどうかきいてきます。[N]を入力するとCP/Mに戻ります。[Y]を入力すると以下のように確認してきます。

```
変更データをシステムトラックに登録しますか(Yes / No)?Y
ドライブ名を入力してください。A
書き込みを開始しますか(Yes/No)?Y
終了しました。
```

ドライブ名（A～H）を入力すると書き込みを開始するかどうかきいてきます。
[N]を入力すると⑦の始めに戻ります。
[Y]を入力すると書き込みを開始します。

## 8 HDMAINT （ハードディスクの領域確保・解除）

HDMAINT コマンドは，CP/M システムでハードディスクを使用する場合，ハードディスクの使用領域を設定するときに使います。

現在発売されている X1 turbo 純正のハードディスクとして，以下の機種があります。

| 機種名 | 標準価格 | 記憶容量 | タイプ |
|---|---|---|---|
| CZ-500H | 348,000円 | 10MB（メガバイト） | 5″ウィンチェスター型ハードディスク |

なお，ハードディスクに付属している取扱説明書も十分お読みいただいてからご使用ください。

〈ご注意〉

HDMAINT を実行する前に以下の事項にご注意ください。

ハードディスクを初めて使用する場合

ハードディスクを購入後，初めて使用する場合は HDMAINT を実行する前に，ディスクのフォーマットとマッピング（領域設定）を行なう必要があります。これらの方法につきましては，ハードディスクの取扱説明書と，コンピュータ本体に付属している〝アプリケーションソフトの説明〟のディスクユーティリティ　3. ハードディスクフォーマットユーティリティと，4. ディスクマップ設定を参照してください。

ディスクマップ設定を行なう場合,10M バイトの容量を 1 M バイトの領域10個に分割し，それぞれの領域を BASICで使用するか他のOSで使用するか指定するようになっています。CP/M で使用したい領域は，〝他の OS〟で使用するように指定してください。

すでに BASIC などでハードディスクを使用している場合

ハードディスクの10M バイトの容量のうち CP/M で使用できる領域は，ハードディスクに付属している HD MAP. Uty（ディスクマップ設定）で指定した〝他の OS〟の領域になります。すでに BASIC で領域確保されている領域を〝他の OS〟に指定しなおす場合は，一度使用領域解除を行なってください。

### HDMAINT の実行

〝他の OS〟用に確保した領域のうち，turbo CP/M で使用する領域の設定（領域確保と解除）を行なうのが HDMAINT です。領域設定は以下の手順で行なってください。

①CP/M のコマンド待ちの状態で

HDMAINT ↵

と入力します。

②領域設定を行なうドライブ番号をきいてきますので，0～3 の数字を入力してください。（工場出荷時，ハードディスクのドライブ番号は 0 に設定されています。）

③ドライブ番号を入力すると，メニュー画面が表示されます。現在のハードディスク使用状況も確認できます。

```
CP/Mのハードディスク使用状況
識別番号          容量(Mﾊﾞｲﾄ)          ｼｽﾃﾑﾄﾗｯｸ




残り容量･･･････ １０Mﾊﾞｲﾄ
選択する項目の番号を入力してください。
１・・・・・CP/M領域の確保
２・・・・・CP/M領域の解除
３・・・・・CP/Mに戻る
```

上の例では残り容量が10Mバイトになっています。HDMAINTの実行前の注意事項で述べましたディスクマップの設定で10Mバイトすべてを“他のOS”で使用するように指定した場合の画面です。

## CP/M領域の確保

メニュー画面で[1]を入力するとCP/M領域の確保を行ないます。確保したいCP/M領域の識別番号(ID)，容量，システムトラックを設けるかどうかを順に指定していきます。
識別番号0の領域を2Mバイト確保してシステムトラックを設けるように設定した例です。

```
CP/Mのハードディスク使用状況
識別番号          容量(Mﾊﾞｲﾄ)          ｼｽﾃﾑﾄﾗｯｸ




残り容量.....１０ Mﾊﾞｲﾄ
確保したいCP/M領域の識別番号を入力してください。0
何Mバイト確保しますか（0～8）2
システムトラックを設けますか（Yes/No)?Y
確保してもよいですか（Yes/No)?
```

①識別番号(ID)……ハードディスクでは大容量のデータを扱うため，〝他のOS〞で確保した領域の中でturbo CP/Mで使用する領域に識別番号をつけて使用します。番号を入力したら ↵ を押します。

②容　　量……1 Mバイト単位で指定することができます。残り容量を考慮しながら1つの領域に8 Mバイトまで指定できます。

③システムトラック……IPL，リブートの際に必要なシステムを格納するためのトラックです。

システムトラックを設けるときは Y，必要ないときは N を入力します。

(①～③の設定の際に ↵ のみを入力するとメニュー画面に戻ります。)

①～③の設定が終了すると，領域を確保してよいかきいてきます。

Y を入力すると次のように表示されます。

```
CP/Mのハードディスク使用状況

識別番号          容量(Mﾊﾞｲﾄ)          ｼｽﾃﾑﾄﾗｯｸ
   0                 2                   有




残り容量･････････  8 Mﾊﾞｲﾄ

選択する項目の番号を入力してください。

1・・・・・CP/M領域の確保
2・・・・・CP/M領域の解除
3・・・・・CP/Mに戻る
```

## CP/M 領域の解除

下の図のような使用状況のときに，識別番号を解除する例を説明します。

```
CP/Mのハードディスク使用状況

識別番号          容量(Mﾊﾞｲﾄ)          ｼｽﾃﾑﾄﾗｯｸ
    0                 2                  有
    1                 1                  有




残り容量･･･････  7 Mﾊﾞｲﾄ

選択する項目の番号を入力してください。

1・・・・・CP/M領域の確保
2・・・・・CP/M領域の解除
3・・・・・CP/Mに戻る
```

まず，メニュー画面で [2] を入力します。

```
CP/Mのハードディスク使用状況

識別番号          容量(Mﾊﾞｲﾄ)          ｼｽﾃﾑﾄﾗｯｸ
    0                 2                  有
    1                 1                  有




残り容量･･･････  7 Mﾊﾞｲﾄ

 解除したいCP/Mの識別番号を入力してください。1

 解除してもよいですか (Yes/No)?
```

①解除したい識別番号を入力して ↵を押します。

②解除してもよければ [Y] を入力します。[N] を入力すると，メニュー画面に戻ります。

③②で [Y] を入力すると，次のように表示されます。

```
CP/Mのハードディスク使用状況

識別番号        容量(Mﾊﾞｲﾄ)        ｼｽﾃﾑﾄﾗｯｸ
    0               2                有



残り容量･･･････  8 Mﾊﾞｲﾄ

選択する項目の番号を入力してください。

1・・・・・CP/M領域の確保
2・・・・・CP/M領域の解除
3・・・・・CP/Mに戻る
```

### CP/M に戻る

メニュー画面で 3 を入力すると，CP/M 領域の確保または解除を実行してハードディスクの領域を変更した場合は，そのデータに基づいて実際にハードディスクに領域設定を行なってもよいかどうかきいてきます。

```
変更データをディスクに登録します。

変更してもよいですか (Yes/No)?
```

変更してもよければ Y，変更する必要がなければ N を入力してください。N を入力すると CP/M に戻ります。Y を入力すると処理終了後 CP/M に戻ります。

変更後は必ずメニュー画面で 3 を入力してから HDMAINT を終了してください。

# 9 HDBACKUP （ハードディスクのバックアップ）

**HDBACKUP コマンドは，不慮の事故によるデータの破壊に備えて，ハードディスク内のデータをフロッピーディスクでバックアップをとるためのコマンドです。また，万一，ハードディスク内のデータが壊れた場合にはフロッピーディスクのバックアップファイルをハードディスクに戻すこともできます。**

HDBACKUPでバックアップをとると，数Kバイトのファイルでもフロッピーディスク1枚分すべてを使用します。1枚のディスケットにおさまるファイルに関しては，PIP コマンド（第6章参照）で行なってください。

各デバイスタイプのフロッピーディスク1枚分にコピーできる容量は次のとおりです。

| デバイスタイプ | バックアップ容量 |
|---|---|
| 5 inch 2D | 300K バイト |
| 5 inch 2DD | 620K バイト |
| 5 inch 2HD | 960K バイト |
| 8 inch 2D | 960K バイト |

※ HDBACKUP では DIC ファイルを扱うことはできません。

## HDBACKUP の実行

HDBACKUP ↵ と入力すると最初に下のようなメニュー画面が表示されます。

```
1・・・・・バックアップ（Hard   disk ---> Floppy disk）
2・・・・・リストア     （Floppy disk ---> Hard   disk）
3・・・・・バックアップ・ディスクのディレクトリ確認
4・・・・・CP/Mに戻る

メニュー番号を入力してください。
```

1．バックアップ……ハードディスクからフロッピーディスクにバックアップをとるためのモードです。対象となるファイルは300K バイト以上のものですので複数枚のフロッピーディスクが必要になります。この場合，フロッピーディスクは CP/M のフォーマット済のものをご使用ください。また，別のファイルが記録されている場合，そのファイルはすべて消えてしまいますので注意してください。

2．リストア　……ハードディスク内のデータが壊れた場合などに，バックアップをとってあったフロッピーディスクからハードディスクにファイルを戻すためのモードです。

3．バックアップ・ディスクのディレクトリ確認
……バックアップをとったフロッピーディスクのディレクトリを確認するためのモードです。

## バックアップ

メニュー画面で 1 を入力するとバックアップのモードになります。例ではドライブFのハードディスクの〝SAMPLE・FIL〟というファイルをドライブAのフロッピーディスクにバックアップします。

```
ソースドライブ名を入力してください。F
ディスティネーションドライブ名を入力してください。A
ファイル名を入力してください。SAMPLE.FIL
ファイルのバックアップを開始しますか (Yes/No)?
```

①ソースドライブ名（ハードディスクドライブ名）を入力します。
②ディスティネーションドライブ名（フロッピーディスクドライブ名）を入力します。
③ファイル名を入力して，↵を押します。
④バックアップを開始してもよければ Y を入力します。（ N を入力すると，メニュー画面に戻ります。）

＊ ①～③で何も入力しないで↵を押すとメニュー画面に戻ります。

⑤

```
SAMPLE .FIL のバックアップを開始します。
  1番目のディスクを入れてリターンキーを押してください。
```

フォーマット済のディスクをセットして↵を入力します。

⑥

```
SAMPLE .FIL をバックアップしています。
```

バックアップ中は上のように表示されます。

⑦

```
SAMPLE .FIL をバックアップしています。
  2番目のディスクを入れてリターンキーを押してください。
```

1番目のディスクが一杯になると上のように表示されます。2番目のディスクをセットして↵を入力します。

⑧

```
SAMPLE .FIL をバックアップしています。
バックアップを続行します。
```

バックアップ中は上のように表示されます。

3番目以後のディスクが必要な場合は，⑦，⑧を同様にくり返してください。

⑨

```
終了しました。
何かキーを押してください。
```

バックアップが終了すると上のように表示されます。何かキーを入力すると，メニュー画面に戻ります。

＊バックアップをとったフロッピーディスクはあくまでも保存用のものです。単独で使用することはできません。

## リストア

メニュー画面で 2 を入力するとリストアのモードになります。
リストアを行なう場合は，バックアップをとった順番でディスクをセットしてください。何枚目のディスクかを確認するためには，次で説明するバックアップディスクのディレクトリを行なってください。例では，ドライブAのバックアップディスクの "SAMPLE.FIL" というファイルをドライブFのハードディスクに戻します。

```
ソースドライブ名を入力してください。A
ディスティネーションドライブ名を入力してください。F
ファイルのリストアを開始しますか（Yes/No)?
```

①ソースドライブ名（バックアップディスクの入っているドライブ名）を入力します。
②ディスティネーションドライブ名（ハードディスクドライブ名）を入力します。
③リストアを開始してもよければ Y を入力します。（ N を入力すると，メニュー画面に戻ります。）
＊①～③で何も入力しないで↵を押すとメニュー画面に戻ります。

④

```
SAMPLE .FIL をリストアしています。
  2番目のディスクを入れてリターンキーを押してください。
```

バックアップの場合と同様に1番目のディスクのリストアが終了すると2番目と差し換えるようにメッセージが表示されます。

⑤

```
SAMPLE .FIL をリストアしています。
リストアを続行します。
```

リストア続行中は上のように表示されます。

⑥

```
終了しました。
何かキーを押してください。
```

リストアが終了すると上のように表示されます。何かキーを入力すると，メニュー画面に戻ります。

## バックアップディスクのディレクトリ

メニュー画面で[3]を入力すると，ディレクトリのモードになります。

```
ドライブ名を入力してください。A
準備はよろしいですか (Yes/No)?
```

バックアップディスクの入っているドライブ名を入力して，準備ができたら[Y]を入力します。([N]を入力するとメニュー画面に戻ります。また，何も入力しないで↵を押すと，メニュー画面に戻ります。)

[Y]を入力すると，バックアップディスクのファイル名，バックアップした時間，ディスク番号（何枚目のディスクか）を表示します。

```
以下にディスクのディレクトリを示します
ファイル名              :SAMPLE  .FIL
バックアップした時間:86/03/01 SAT 10:15:20
ディスク番号（枚目）：  1
何かキーを押してください
```

何かキーを押すと，もう1度ドライブ名の入力に戻ります。もう確認する必要がなければ，そのまま↵を押してください。メニュー画面に戻ります。

また，ディレクトリを確認したフロッピーディスクが最後のディスクである場合は次のように表示されます。

```
以下にディスクのディレクトリを示します
ファイル名              :SAMPLE  .FIL
バックアップした時間:86/03/01 SAT 10:15:20
ディスク番号（枚目）：  2   （これが最後のディスクです）
何かキーを押してください
```

HDBACKUPを終了するときはメニュー画面で[4]を入力してください。CP/Mに戻ります。

## 10 HDOFF （ハードディスクの終了設定）

HDOFF は，CP/M システムでハードディスクの使用を終了し，ハードディスクの電源を切る前に必ず実行していただくコマンドです。HDOFF を実行することにより，ハードディスクのヘッドをディスク上のデータ領域外に移動させ，データの破壊を防止します。
HDOFF の実行には，以下の 3 通りの入力方法があります。

①HDOFF↵
ドライブ 0 に対して HDOFF を実行します。

②HDOFF␣d↵（d＝0, 1, 2, 3）
指定のハードディスクのドライブ番号(0〜3)に対して HDOFF を実行します。

③HDOFF␣*↵
CP/M でアクセスしたすべてのドライブに対して HDOFF を実行します。

## 11 TERM （ターミナル・リモートモードでの通信）

TERMコマンドは，X1turboのRS−232Cインターフェイスを使って，他の機器と接続し通信を行なう場合に使用するもので，RS−232Cのパラメータの設定と動作モードの設定をする必要があります。

### RS-232Cパラメータの設定

RS−232Cパラメータの設定は次の8項目について行ないます。

1．ボーレート

データ転送速度のことで，1秒間に伝送できるビット数を表わします。turbo CP/Mでは150〜9600ボーが設定可能です。

2．通信制御

データ送受信時に受信側のデータ受信処理が間に合わない場合，送信側に対して送信の一時停止を要求し，受信可能な状態になると送信を要求することをいいます。

RTS　ON/OFF ……RTS制御信号線をOFFで送信一時停止，ONで送信再開を要求します。

XON/XOFF　…… XOFFコード（13H）を送って送信一時停止，XONコード(11H)を送って送信再開を要求します。

3．ストップビット

各データを送るとき，1文字分ずつのデータの区切りを示す信号で，1ビット，1.5ビット，2ビットの3種類あります。

4．パリティ

各データが正しく送られたかどうかをチェックするものです。偶数パリティと奇数パリティがあり，〝なし〟を選択するとチェックを行ないません。

5．キャラクタ長

各データ（1文字分）を何ビット使用して送るかの設定で，5〜8ビットに設定可能です。

6．カナ表現

7ビットデータでカナを送るかどうかを選択します。

SI/SO　…7ビットデータでカナを送ることができます。
SO（シフトアウトコード0EH）があると以後のデータをカナとみなし，SI（シフトインコード0FH）があると英数字とみなされます。これにより，キャラクタ長7ビットでカナ文字の転送が可能となります。

な　し　…7ビットデータではカナを送受信できません。

7．受信改行コード

CR……CR コード（0DH)を受信し，復帰＋改行コードとして処理します。

CR＋LF……CR＋LF(OAH)コードを復帰＋改行コードとして処理します。

8．送信改行コード

CR…… CR コード（0DH）を復帰＋改行コードとして送信します。

CR+LF……CR コード(0DH)＋LF コード(0AH)を復帰＋改行コードとして送信します。

**動作モード**

動作モードとは次の2つのモードです。

1．ターミナルモード……X1turbo をターミナルとして使用するモード

2．リモートモード　……X1turbo をホストコンピュータとし，外部の機器から X1turbo を操作するモード

**設定方法**

①TERM ↵と入力します。

②下のように表示されます。

```
現在の設定状況は次のとおりです。

ボーレート      ... 300 BPS
通信制御        ... XON/XOFF
ストップビット  ... 1   bit
パリティ        ... なし
キャラクタ長    ... 8 bit
カナ表現指定    ... なし
受信改行コード  ... CR/LF
送信改行コード  ... CR

設定データの変更をしますか（Yes/No)?
```

データの変更を行なう場合は Y を入力します。

N を入力すると⑧の画面になります。

③データの変更を行なう場合はまずボーレイトの変更を行ないます。
ボーレイトは150～9600BPS の中から選択することができます。1～7の番号を入力してください。変更する必要がないときは↵を入力するとスキップして，次の設定項目へ移ります。(以下同様です。)

```
設定データの変更

ボーレート

  1.....9600 BPS
  2.....4800 BPS
  3.....2400 BPS
  4.....1200 BPS
  5..... 600 BPS
  6..... 300 BPS
  7..... 150 BPS

番号を選んでください。
```

④次に通信制御の変更を行ないます。1～3の番号を入力してください。

```
設定データの変更

通信制御

  1.....なし
  2.....RTS ON/OFF
  3.....XON/XOFF

番号を選んでください。
```

⑤以後，各項目について同様に設定を行ないます。

⑥8項目全ての設定変更が終ると，自動的に②の画面に戻ります。このとき，設定状況は変更後のデータになっています。

⑦設定状況を確認し，これでよければ N を入力します。まだ変更が必要なときには Y を入力し，もう1度ボーレイトの設定から同じ操作をしてください。

⑧⑦で N を選ぶと次のような画面が表示され，動作モードの選択へ入ります。

```
モードの選択

  1...ターミナル　モード（全二重）
  2...ターミナル　モード（半二重）
  3...リモート　モード
  4...ＣＰ／Ｍに戻る

番号を入力してください。
```

⑨モードの選択をして番号を入力します。各モードへ入り，送受信待機の状態となります。
モード選択の時点で送受信の必要のないときは 4 を入力して CP/M へ戻ります。

## ターミナルモード

動作モード設定で，ターミナルモードを指定すると X 1 turbo を端末として他のコンピュータと接続し，操作することができます。
ターミナルモードには，全二重と半二重の 2 つのモードがあり，次のようなちがいがあります。

全二重モード……　このモードでは，キーボードから入力された文字は画面上には出力されず，受信した内容だけが画面に表示されます。

半二重モード……　このモードでは，キーボードから入力された内容と，受信内容の両方が画面に表示されます。

相手の通信方式にあわせて変更します。

ターミナルモードから CP/M システムのコマンド待ちの状態に戻るには，
CTRL ＋ GRAPH ＋ SHIFT ＋ F 1
の 4 つのキーを同時に押します。

## リモートモード

動作モード設定でリモートモードを指定すると，X1turbo をホストマシンとして他のコンピュータから X1turbo を操作することができます。

リモートモードから CP/M システムのコマンド待ちの状態へ戻るには
CTRL ＋ GRAPH ＋ SHIFT ＋ F 1
の 4 つのキーを同時に押します。

## 12 CMT （ディスク・カセット間のファイル転送）

CMT コマンドは，ディスクのファイルをカセットテープにセーブしたり，カセットテープのファイルをディスクにセーブするユーティリティプログラムです。

1．ディスクのファイルをカセットにセーブ

```
CMT␣装置名：ディスクファイル名␣カセットファイル名
```

と入力することにより，ディスクのファイルをカセットテープにセーブすることができます。カセットファイル名は，ディスクファイル名と同じように，ファイル名 8 文字以内，ファイルの形式 3 文字以内でかならず指定してください。もし，カセットファイル名を指定しなかった場合，カセットのファイルを読み込む動作になってしまいます。なお，ディスクのファイルの形式により，カセットテープにセーブされるファイルの形式がちがってきます。

例

```
A>CMT  B:GAME.COM  GAME.CPM ↵
```

例は，ドライブBにある GAME.COM ファイルを，カセットテープに GAME.CPM というファイル名でセーブしています。

※ディスクのファイルが"……. COM"ファイルの場合
ディスクのファイルは，カセットテープに機械語形式のファイルとしてセーブされます。すなわち，セーブされたファイルは，IPL ロードにより直接実行可能になります。しかし，プログラム先頭アドレスおよび実行アドレスは，100H番地固定となります。なぜなら，"……. COM"ファイルは，メモリーの100H番地からロードされ，100H番地より実行されるからです。

※ディスクのファイルが"……. COM"ファイル以外の場合
ディスクのファイルは，カセットテープに 1 ブロック256バイトのブロックごとに，アスキー形式のファイルとしてセーブされます。（1 ブロックごとに，カセットのヘッドを上下させ，テープに記録します）

2．カセットテープのファイルをディスクにセーブ

```
CMT␣装置名：ディスクファイル名
```

と入力することにより，カセットテープのファイルをディスクに指定ファイル名でセーブします。ディスクにセーブできるカセットテープのファイル形式は，機械語形式とアスキー形式で，X1 turbo付属のBASIC※1)で作成したプログラムが，アスキー形式でセーブされていない場合，CMTコマンドでディスクにセーブすることはできません。

例

```
A>CMT  TAPE.COM ↵
```

例は，カセットテープのファイルをログインディスク（この場合，ドライブA）に，TAPE.COMというファイル名でセーブしています。

※カセットテープのファイルが機械語形式の場合

カセットテープのファイルは，ディスクに機械語形式のファイルとしてセーブされます。ただし，プログラム先頭アドレスおよび実行アドレスは無視されますので，ディスクファイル名を"…….COM"とする場合，カセットテープのファイルのプログラム先頭アドレスおよび実行アドレスは100H番地でなければなりません。

※カセットテープのファイルがアスキー形式の場合

カセットテープのファイルは，ディスクにアスキー形式のファイルとしてセーブされます。なお，1行の終わりは，"CR"+"LF"(0DH, 0AH)のコードを送ってください。X1 turbo付属のBASICでは，1行の終わりには"CR"（0DH）のコードのみしか送られません。

※1)

X1 turbo付属のBASICとは
CZ8CB02　CZ8FB02
のことを指します。

〔例1　CMT コマンド実行例〕

```
A>CMT DUMP.COM DUMP.COM ↵ ―――――――――――――――――― ①
COMPLETE

A>CMT B:DEX.COM ↵ ―――――――――――――――――― ②
FOUND DUMP          COM
COMPLETE

A>B:DEX DUMP.COM ↵ ―――――――――――――――――― ③

0000 21 00 00 39 22 15 02 31 57 02 CD C1 01 FE FF C2
0010 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 13 02
0020 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2
0030 44 01 CD 72 01 CD 59 01 0F DA 51 01 7C CD 8F 01
0040 7D CD 8F 01 23 3E 20 CD 65 01 78 CD 8F 01 C3 23
0050 01 CD 72 01 2A 15 02 F9 C9 E5 D5 C5 0E 0B CD 05
0060 00 C1 D1 E1 C9 E5 D5 C5 0E 02 5F CD 05 00 C1 D1
0070 E1 C9 3E 0D CD 65 01 3E 0A CD 65 01 C9 E6 0F FE


A>
```

①ディスクの DUMP. COM をカセットに同じファイル名でセーブします。（この場合，カセットテープには機械語形式のファイルがセーブされます。）

②カセットのファイル（DUMP.COM）を，ディスクのドライブBに DEX.COM というファイル名でセーブします。

③DEX.COM を実行し，DUMP.COM ファイルの16進コード出力をします。DEX.COM ファイルは，もともとファイルの16進コード出力をする DUMP. COM ファイルを，ディスク→カセット→ディスクとファイルをうつして作ったものですから，DEX. COM と DUMP. COM は同じといえます。ここで注意することは，ディスクのファイルをカセットにうつしても，ファイルの内容は全く変化しないということです。

〔例2　CMT コマンド実行例〕

```
A>TYPE B:ABC.XYZ ↵ ―――――――――――――――――――――――― ①
New Product Information

Image Input Device for the X1 Series and
X1 Turbo Series Personal Computer TVs

Features
(1) High-speed processing of images input
(2) New built-in SCRAMBLE circuit for subtle colors
(3) Easy to use

A>CMT B:ABC.XYZ ABC.XYZ ↵ ―――――――――――――――――― ②
COMPLETE


A>
```

①TYPE コマンドで ABC.XYZ の内容をタイプアウトします。

②ドライブBにある ABC.XYZ ファイルを，カセットに同じファイル名でセーブします。（この場合，カセットテープにはアスキー形式のファイルでセーブされます。）

```
10 WIDTH 80
20 OPEN "I",#1,"CAS:"
30 LINPUT #1,A$
40 PRINT MID$(A$,INSTR(A$,CHR$(&HA))+1)
50 IF INSTR((A$,CHR$(&H1A))<>0 THEN CLOSE #1 ELSE GOTO30
```

③ここで一度電源を切り，次に X1 turbo 付属の BASIC をロードし，例のようなプログラムを作ります。ここで行番号40は"LF"コードを無視して一行を表示し，行番号50は，ファイルの終わり（エンドオブファイル）を調べています。そして先ほどカセットにセーブしたファイルの先頭にテープを巻き戻し，プログラムを実行しますと，画面に次のように表示されます。

```
New Product Information

Image Input Device for the X1 Series and
X1 Turbo Series Personal Computer TVs

Features
(1) High-speed processing of images input
(2) New built-in SCRAMBLE circuit for subtle colors
(3) Easy to use
```

以上のようにして，CMT コマンドで作成したカセットのファイルを BASIC のプログラムで確認することができます。

〔例3　CMT コマンド実行例〕

```
10 A$=CHR$(&HD,&HA)
20 OPEN "O",#1,"CAS:TEST"
30 PRINT #1,"New Product Information";A$;
40 PRINT #1,A$;
50 PRINT #1,"Image Input Device for the X1 Series and";A$;
60 PRINT #1,"X1 Turbo Series Personal Computer TVs";A$;
70 PRINT #1,A$;
80 PRINT #1,"Features";A$;
90 PRINT #1,"(1) High-speed processing of images input";A$;
100 PRINT #1,"(2) New built-in SCRAMBLE circuit for subtle colors";A$;
110 PRINT #1,"(3) Easy to use";A$;
120 CLOSE #1
```

X1 turbo 付属の BASIC で上例のプログラムを作り，プログラムを実行しますと，カセットテープにアスキー形式のファイルが作成されます。ここで，CHR$(&H0D，&H0A)は CP/M の1行の終了を示しており，CHR$(&H1A)は，CP/M のファイルの終わり（エンドオブファイル）を示しています。

```
A>CMT B:TEST ↵ ―――――――――――――――― ①
FOUND TEST
COMPLETE

A>TYPE B:TEST↵ ―――――――――――――――― ②
New Product Information

Image Input Device for the X1 Series and
X1 Turbo Series Personal Computer TVs

Features
(1) High-speed processing of images input
(2) New built-in SCRAMBLE circuit for subtle colors
(3) Easy to use

A>
```

①セーブしたファイルの先頭にテープを巻き戻し，次に CP/M を起動し，CMT コマンドにより，先ほど作成したカセットファイル "TEST" をドライブBに，同じファイル名でセーブします。

②TYPE コマンドにより，ディスクにセーブしたファイルの内容を確認します。以上の作業により，X1 turbo 付属の BASIC で，CP/M ファイルを作ることができます。

## 13 KEY （ファンクションキーの設定）

KEY コマンドは，ファンクションキーに対して文字列を定義したり，ファンクションキーの定義状態を表示したりします。また，〝……KEY〞ファイルからファンクションキーに定義することもできます。

### 例1 ファンクションキーの定義状態の表示

KEY ↵ と入力すると，ファンクションキーの定義状態を表示します。この例は，CP/M 起動時の定義状態です。

```
A>KEY↵
1, DIR@M
2, STAT *.*@M
3, PIP
4, TYPE
5, SUBMIT
6, DIR
7, STAT
8, KEY
9, REN
0, DDT

A>
```

たとえば

1，DIR@M

はファンクションキー F 1 の定義状態を示しています。

@M は，CTRL ＋ M，すなわち↵の意味です。

また，

6，DIR

は SHIFT ＋ F 1 を押す定義状態を，

0，DDT

は SHIFT ＋ 5 を押す定義状態を表わしています。

### 例2 文字列の定義

KEY␣n，文字列（n＝0～9）↵

を入力することにより，ファンクションキーに文字列を定義することができます。

例では，ファンクションキー F 5 に TIME@M を定義しています。なお，設定できる文字数は15文字までです。

```
A>KEY 5,TIME@M↵

A>KEY
1, DIR@M
2, STAT *.*@M
3, PIP
4, TYPE
5, TIME@M
6, DIR
7, STAT
8, KEY
9, REN
0, DDT

A>
```

**例3**
**先行キーバッファ**
**への予約**

KEY␣*，文字列↵
と入力することにより，63文字分の先行キーバッファへの書き込みができます。

```
A>KEY *,DIR DUMP.*@M

A>DIR DUMP.*
A: DUMP      COM : DUMP      ASM
A>
```

この例では，DIR　DUMP.＊　↵を先行キーバッファに書き込み，KEY コマンド終了後，予約した文字列を出力しています。

**例4**
**ファイルからの**
**文字列の定義**

KEY コマンドでは，ディスクの〝……．KEY〟ファイルからファンクションキーに文字列を定義することができます。

```
A>TYPE EXK.KEY——①
1,TIME
2,ED
*,KEY@M

A>
```

①は，ED コマンドで作成した EXK.KEY ファイルの内容を TYPE コマンドでリストアウトしたものです。（ファイル名のエクステンションはかならず〝KEY〟とします。）
ここでは F 1 を TIME，F 2 を ED，そして，先行キーバッファに KEY ↵を定義しています。

```
A>KEY EXK————②

A>KEY
1,TIME
2,ED
3,PIP
4,TYPE
5,SUBMIT
6,DIR
7,STAT
8,KEY
9,REN
0,DDT

A>
```

EXK.KEY を KEY コマンドで実行したのが②です。文字列の定義を行なってから，先行キーバッファに予約していた文字列を出力しています。

ファンクションキーの状態は，画面の最下行に表示させたり表示させなかったりすることができます。通常は表示させるモードになっています。（第8章「コントロールコードとエスケープシーケンス」を参照してください。）

## 14 TIME （日付・時刻の設定）

TIMEコマンドは，X1turbo内蔵の時計の内容をコンソール画面に表示したり，データを設定するときに使います。

### TIMEの実行

TIME ↵ と入力すると，年，月，日，曜日，時，分，秒の順に表示されます。

```
A>TIME ↵
86/03/01 SAT 11:30:15

A>
```

例では，86年3月1日土曜日，11時30分15秒を表わしています。

### データ設定方法

データの設定方法として3通りの方法があります。
それぞれ，TIMEと入力して，1つスペースをあけてから

①年，月，日，時，分，秒を入力して ↵ を押します。（曜日は自動的に設定されます。）
②年，月，日を入力して ↵ を押します。（曜日は自動的に設定されます。）
③時，分，秒を入力し ↵ を押します。

データは“YY/MM/DD␣HH：MM：SS”の書式で入力します。
例えば，86年3月1日土曜日，11時30分15秒は，86/03/01␣11：30：15となります。

上の設定方法のうち①と③は時間設定のタイミングを測るために，
↵ を押すと

```
Hit any key to set time
```

と表示されます。ここで何かキーを押した瞬間に指定した時間が設定されます。
キーを入力すると，

```
Data set OK
```

と表示されます。

①年，月，日，時，分，秒を設定する例

```
A>TIME 86/03/01 11:49:30 ↵

Hit any key to set time
Data set OK

A>
```

②年，月，日を設定する例

```
A>TIME 86/03/01 ↵

 Data set OK

 A>
```

③時，分，秒を設定する例

```
A>TIME 11:50:20 ↵

Hit any key to set time
Data set OK

A>
```

秒データの入力は省略することができます。この場合，秒データは00にセットされます。

```
A>TIME 12:01 ↵

Hit any key to set time
Data set OK

A>
```

設定データがまちがえていると，次のようにエラー表示されます。再度，正しいデータを入力してください。

```
A>TIME 86/3/1 ↵

Invaild format or parameter

A>
```

左の例では，86/03/01と入力するところを86/3/1と入力しているために，エラーとなっています。

# 第8章

# コントロールコードとエスケープシーケンス

# 1 コントロールコードとエスケープシーケンスの使い方

**コントロールコードあるいはエスケープシーケンスを画面に対して出力することにより画面の制御を行なうことができます（通常プログラム中でのみ使用可能です）。**

コントロールコード……アスキーコードのうち，文字を表わすコード以外に各種コントロールを割りつけたもの。

エスケープシーケンス……各種のコントロールをエスケープコード（1BH）と他の文字を組み合わせることによって表わしたもの。

コントロールコードとエスケープシーケンスを利用するには，各コードをCRT：のデバイスに出力するか，コンソールに直接出力する必要があります。

**使用例**

システムコールを使用してコンソールに各コードを出力する例を以下にあげます。
（システムコールについては第9章を参照してください。）

(1)コントロールコードでビープ音を出す場合

```
ORG     100H

MVI     C，2        ┐
MVI     E，07H      ├ コンソールにコントロールコードの7Hを
CALL    5H          ┘ 出力します

MVI     C，0        ┐
CALL    5H          ┘ システムリセットを行ないプロンプト表示に戻ります
```

⑵エスケープシーケンスでファンクションキーの表示を消す場合

```
ORG     100H

MVI     C, 2
MVI     E, 1BH      ｝コンソールにエスケープコード（1BH）を
CALL    5H          ｝出力します

MVI     C, 2
MVI     E, 'f'      ｝コンソールに f(66H)を出力します
CALL    5H

MVI     C, 0        ｝システムリセットを行ないプロンプト表示
CALL    5H          ｝に戻ります
```

## 2 コントロールコード

コントロールコードを使用して，カーソルを移動したりビープ音を出したりすることができます。これらのコードをディスプレイ装置に出力すれば，画面制御等を行なえます。

| 内　　容 | コード（HEX） |
|---|---|
| ビープ音を出します | 07H |
| カーソルを１つ前にもどします | 08H |
| 改行します | 0AH |
| カーソルを画面左上端へ移します | 0BH |
| テキスト画面をクリアします | 0CH |
| 復帰します（リターンキーと同じ機能） | 0DH |
| グラフィック画面をクリアします | 15H |
| グラフィック画面とテキスト画面をクリアします | 16H |
| エスケープシーケンスコード※ | 1BH |
| カーソルを次へ移動します | 1CH |
| カーソルを１つ前にもどします | 1DH |
| カーソルを上へ移動します | 1EH |
| カーソルを下へ移動します | 1FH |

※注　エスケープシーケンスは，エスケープコード（1BH）の後に続けてコードを送ることによって様々な機能を実行させることができます。

## 3 エスケープシーケンス一覧

turbo CP/M では，エスケープシーケンスで以下のような特有の機能がサポートされています。

| | 内　　容 | ESC に続く文字 | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | …… |
| 1 | テキスト画面のクリア | * | | | | |
| 2 | カーソル以下の一行クリア | T | | | | |
| 3 | カーソル以下の画面クリア | Y | | | | |
| 4 | カーソルを表示させるモード | B | | | | |
| 5 | カーソルを表示しないモード | b | | | | |
| 6 | カーソルのダイレクト指定 | = | Y | X | | |
| 7 | 現在のカーソル位置をかえします | ? | | | | |
| 8 | 文字の反転　ON | ( | | | | |
| 9 | 文字の反転　OFF | ) | | | | |
| 10 | 文字のブリンク　ON | { | | | | |
| 11 | 文字のブリンク　OFF | } | | | | |
| 12 | 横倍　ON | < | | | | |
| 13 | 横倍　OFF | > | | | | |
| 14 | 縦倍　ON | ^ | | | | |
| 15 | 縦倍　OFF | _ | | | | |
| 16 | カラー　0（黒） | 0 | | | | |
| 17 | 　　1（青） | 1 | | | | |
| 18 | 　　2（赤） | 2 | | | | |
| 19 | 　　3（マゼンタ） | 3 | | | | |
| 20 | 　　4（緑） | 4 | | | | |
| 21 | 　　5（シアン） | 5 | | | | |
| 22 | 　　6（黄） | 6 | | | | |
| 23 | 　　7（白） | 7 | | | | |
| 24 | パレットの設定 | P | $H_1$ 0〜7 | $H_2$ 0〜7 | | |
| 25 | ファンクションキーを表示させるモード | F | | | | |
| 26 | ファンクションキーを表示しないモード | f | | | | |
| 27 | キーの先行入力不可のモード | + | | | | |
| 28 | キーの先行入力可のモード | - | | | | |
| 29 | 非漢字モード | U | 0 | | | |
| 30 | 漢字モード | U | 1 | | | |
| 31 | ROM キャラクターゼネレータモード | ] | | | | |
| 32 | RAM キャラクターゼネレータモード | [ | | | | |

| | 内　　容 | ESC に続く文字 | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | …… |
| 33 | ROM キャラクターゼネレータの内容をかえします(8×8フォント) | C | 0 | $H_1H_1$ 00～FF | | |
| 34 | RAM キャラクターゼネレータの内容をかえします | C | 1 | $H_1H_1$ 00～FF | | |
| 35 | ROM キャラクターゼネレータの内容をかえします(16×8フォント) | C | 2 | $H_1H_1$ 00～FF | | |
| 36 | ユーザーキャラクターの定義 | D | HH 00～FF | $H_1H_1$ | …… | $H_{24}H_{24}$ |
| 37 | 画面モードの初期化 | I | | | | |
| 38 | 行，解像度の設定（25行・200ライン） | V | 0 | | | |
| 39 | （12行・192ライン） | V | 1 | | | |
| 40 | （20行・グラフィック表示不可） | V | 2 | | | |
| 41 | （10行・グラフィック表示不可） | V | 3 | | | |
| 42 | （25行・400ライン） | V | 4 | | | |
| 43 | （12行・384ライン） | V | 5 | | | |
| 44 | 指定されたグラフィック位置に点を表示 | S | $H_1H_1H_1$ 000～27F | $H_2H_2H_2$ 000～18F | $H_3$ 0～7 | |
| 45 | 指定されたグラフィック位置の色データをかえします | G | $H_1H_1H_1$ 000～27F | $H_2H_2H_2$ 000～18F | | |
| 46 | プライオリティの設定 | p | $H_1H_1$ 00～FF | | | |
| 47 | スクリーンモードの変更（GRAM0：表示用・書き込み用） | v | 0 | | | |
| 48 | （GRAM0：表示用，GRAM1：書き込み用） | v | 1 | | | |
| 49 | （GRAM1：表示用，GRAM0：書き込み用） | v | 2 | | | |
| 50 | （GRAM1：表示用・書き込み用） | v | 3 | | | |
| 51 | KEY クリック ON | K | | | | |
| 52 | KEY クリック OFF | k | | | | |
| 53 | サウンドゼネレータにデータを書きます | W | $H_1$ 0～F | $H_2H_2$ 00～FF | | |
| 54 | サウンドゼネレータのレジスタの内容をかえします | R | $H_1$ 0～F | | | |

## 4 エスケープシーケンス詳細

エスケープシーケンスとは，エスケープコード（1BH）の後に続くあるコードをコンソールに送ることにより，特定の機能を実行させることをいいます。
turbo CP/M には様々なエスケープシーケンスが用意してあり，X1 turbo の機能を十分に活用することができます。以下，各エスケープシーケンスについて説明します。

〈表の見方〉

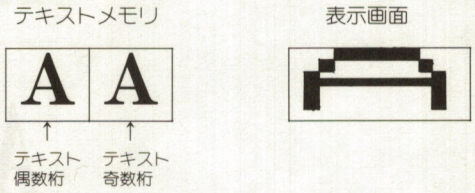
H は 0 ～F までの16進数を表わしており，h はその数字に対する ASCII コードを表現しています。なお，この部分は各種コントロールを行なう際に使われる数値変数を表わしています。

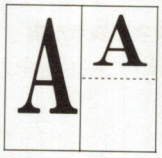
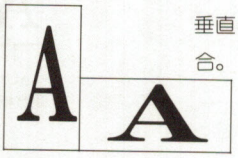
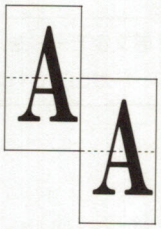
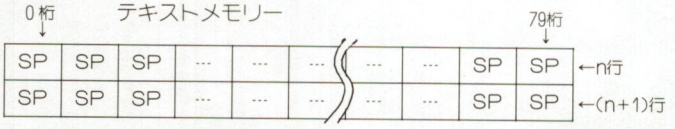
対応表

| 16進数字 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| アスキーコード | 30H | 31H | 32H | 33H | 34H | 35H | 36H | 37H | 38H | 39H | 41H | 42H | 43H | 44H | 45H | 46H |

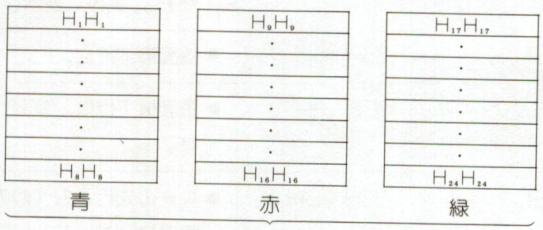
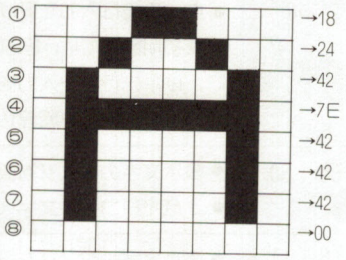
例　H が数字の 9 であればそれに対応する h は39H で表わされます。

| | エスケープコード | 16進コード | 機　　能 |
|---|---|---|---|
| 1 | ESC * | 1BH 2AH | ● 画面をすべてクリアします。カーソルは画面左上隅に移動します。 |
| 2 | ESC T | 1BH 54H | ● カーソルの位置から右 1 行をクリアします。<br>● カーソルの位置は変化しません。 |
| 3 | ESC Y | 1BH 59H | ● カーソルの位置から画面右下端までクリアします。<br>● カーソルの位置は変化しません。 |
| 4 | ESC B | 1BH 42H | ● カーソルを表示させるモードにします。 |
| 5 | ESC b | 1BH 62H | ● カーソルを表示しないモードにします。 |
| 6 | ESC = Y X | 1BH 3DH<br>y x | ● カーソルを Y 行 X 桁の位置まで移動します。<br>● 一番上の行を第 0 行，左端の桁を第 0 桁とします。<br>● x，y には，実際の行数，桁数に20H を加えた値を設定します。<br>たとえば，カーソルを Y＝5 行，X＝13桁へと移動させるには<br>y=05H+20H=25H，x=0DH+20H=2DH とします。 |
| 7 | ESC ? | 1BH 3FH | ● キー入力バッファをクリアした後，現在のカーソル位置を，キー入力バッファの最初の 2 バイトに書き込みます。<br>● 1 バイト目が行数，2 バイト目が桁数を表わし，データは実際の行数，桁数に20H を加えた値となっています。たとえば，カーソルが10行，25桁にあるとき，このコードを出力すれば，キー入力バッファには，0AH+20H=2AH，19H+20H=39H が書き込まれます。<br>（ダイレクトカーソルアドレシングの座標の項を参照してください。） |
| 8 | ESC ( | 1BH 28H | ● 文字表示モードを，リバースモードにします。<br>● このコードが出力された後の表示文字は反転文字で表示されます。 |
| 9 | ESC ) | 1BH 29H | ● リバースモードをノーマルモードにします。<br>● このコードが出力された後の表示文字は標準文字で表示されます。 |

| | エスケープコード | 16進コード | 機　　能 |
|---|---|---|---|
| 10 | ESC { | 1BH 78H | ● 文字表示モードをフラッシングモード（標準文字と反転文字を交互に繰り返す）にします。<br>● このコードが出力されたあとの表示文字はフラッシングします。 |
| 11 | ESC } | 1BH 7DH | ● フラッシングモードをノーマルモードにします。<br>● このコードが出力されたあとの表示文字はフラッシングしない文字です。 |
| 12 | ESC < | 1BH 3CH | ● 表示文字の大きさを水平 2 倍文字にします。<br>● このコードが出力された後の文字は水平 2 倍文字となります。<br>● 表示文字はかならず偶数桁から出力し，次の桁も同じ文字にしてください。一行中にノーマル文字があってもかまいません。<br>● **横倍文字の表示方法**<br>ESC<のコードを出力した後，<br><br> |
| 13 | ESC > | 1BH 3EH | ● 水平 2 倍文字モードをノーマル文字モードにします。<br>● このコードが出力された後の表示文字は標準文字となります。 |

| | エスケープコード | 16進コード | 機　　能 |
|---|---|---|---|
| 14 | ESC ∧ | 1BH 5EH | ● 表示文字の大きさを垂直2倍文字にします。<br>● このコードが出力された後，同じ桁2行に同じ文字を出力すれば，垂直2倍文字となります。<br>● 表示文字の2行中に，ノーマル文字および水平2倍文字があってはなりません。<br>したがって，垂直2倍文字を表示するにあたって，ESC∧のコードを送った後，文字を表示する2行全体にわたってあらかじめスペースコードを書く必要があります。<br>● **表示できない文字の組み合せ**<br>(1)  垂直2倍文字表示桁にノーマル文字がある場合。<br>(2)  垂直2倍文字表示桁に水平2倍文字がある場合。<br>(3)  垂直2倍文字を1行ずらして表示する場合。<br>● **縦倍文字の表示方法**<br>ESC∧のコードを出力した後<br><br>※垂直2倍文字を表示する行全体にスペース（SP）コードを書きます。<br> |
| 15 | ESC _ | 1BH 5FH | ● 垂直2倍文字モードをノーマル文字モードにします。<br>● このコードが出力された後の表示文字は標準文字となります。 |

<table>
<tr><th></th><th>エスケープコード</th><th>16進コード</th><th>機　　能</th></tr>
<tr><td>16<br>〜<br>23</td><td>ESC 0〜7</td><td>1BH 30H〜37H</td><td>●キャラクターカラーをセットします。<br>●このコードが出力された後の表示文字は指定したカラーで表示されます。カラーコードは次の通りです。<br><br>
<table>
<tr><th>カラーコード</th><th>16進コード</th><th>色</th></tr>
<tr><td>0</td><td>30H</td><td>黒</td></tr>
<tr><td>1</td><td>31H</td><td>青</td></tr>
<tr><td>2</td><td>32H</td><td>赤</td></tr>
<tr><td>3</td><td>33H</td><td>マゼンタ</td></tr>
<tr><td>4</td><td>34H</td><td>緑</td></tr>
<tr><td>5</td><td>35H</td><td>シアン</td></tr>
<tr><td>6</td><td>36H</td><td>黄</td></tr>
<tr><td>7</td><td>37H</td><td>白</td></tr>
</table>
</td></tr>
<tr><td>24</td><td>ESC P $H_1$ $H_2$</td><td>1BH 50H $h_1$ $h_2$</td><td>●パレットコード $H_1$ (0〜7) で指定されるパレットに，カラーコード $H_2$ (0〜7) で指定されるカラーを設定します。すなわち，基本色の番号が $H_1$，実際に表示したい色の番号が $H_2$ となります。</td></tr>
<tr><td>25</td><td>ESC F</td><td>1BH 46H</td><td>● 画面最下行に，ファンクションキーの状態を表示するモードにします。</td></tr>
<tr><td>26</td><td>ESC f</td><td>1BH 66H</td><td>● 画面最下行に，ファンクションキーの状態を表示しないモードにします。</td></tr>
<tr><td>27</td><td>ESC +</td><td>1BH 2BH</td><td>● キーの先行入力（他の処理を行なっている間にキー入力すること）を受けつけないモードにします。</td></tr>
<tr><td>28</td><td>ESC −</td><td>1BH 2DH</td><td>● キーの先行入力を受けつけるモードにします。<br>※初期設定では，ESC+のモードになっています。ESC−のモードは，先行キー入力を正しく保っていたいときに使用します。ただし画面出力の一時停止機能は動作しません。</td></tr>
<tr><td>29</td><td>ESC U 0</td><td>1BH 55H 30H</td><td>● 表示文字のタイプを設定します。(BASICのKMODE0)<br>● 1バイトコード文字（半角文字）が表示可能になります。</td></tr>
<tr><td>30</td><td>ESC U 1</td><td>1BH 55H 31H</td><td>● 表示文字のタイプを設定します。(BASICのKMODE1)<br>● 2バイトコード文字（全角文字）と半角文字（セミグラフィック文字を除く）が表示可能になります。</td></tr>
<tr><td>31</td><td>ESC ]</td><td>1BH 5DH</td><td>● 文字表示モードを ROM キャラクターゼネレータモードにします。<br>● このモードが出力された後の表示文字は，通常の文字（ROM キャラクターゼネレータの文字）になります。</td></tr>
<tr><td>32</td><td>ESC [</td><td>1BH 5BH</td><td>● 文字表示モードを RAM キャラクターゼネレータにします。<br>● このコードが出力された後の表示文字は，あらかじめ定義されたユーザー定義キャラクターとなります。</td></tr>
</table>

| | エスケープコード | 16進コード | 機　　能 |
|---|---|---|---|
| 33 | ESC C 0 $H_1H_1$ | 1BH 43H 30H<br>$h_1h_1$ | ●キー入力バッファをクリアした後，キャラクターコード $H_1H_1$ のROMキャラクターゼネレータのデータ8バイト（8×8フォント）を，キー入力バッファの最初から16バイトに書き込みます。データはすべて16進数字で与えられます。 |
| 34 | ESC C 1 $H_1H_1$ | 1BH 43H 31H<br>$h_1h_1$ | ●キー入力バッファをクリアした後，キャラクターコード $H_1H_1$ のRAMキャラクターゼネレータのデータ24バイトを，キー入力バッファの最初から48バイトに書き込みます。<br>● 青のデータが最初の16バイト，次の16バイトが赤のデータ，最後の16バイトが緑のデータとなっており，データはすべて16進数字で与えられます。 |
| 35 | ESC C 2 $H_1H_1$ | 1BH 43H 32H<br>$h_1h_1$ | ●キー入力バッファをクリアした後，キャラクターコード $H_1H_1$ のROMキャラクターゼネレータのデータ16バイト（16×8フォント）を，キー入力バッファの最初から32バイトに書き込みます。データはすべて16進数字で与えられます。 |
| 36 | ESC D<br>HH $H_1H_1 \cdots H_{24}H_{24}$ | 1BH 44H<br>hh $h_1h_1 \cdots h_{24}h_{24}$ | ●RAMキャラクターゼネレータのキャラクターコードHHに，ユーザーキャラクターを定義します。データ入力は次のようにします。<br><br><br><br><br>例えば，図のような白色の文字をキャラクターコード41Hに定義するには，<br>ESC D 41 18 24 42 7E 42 42 42 00<br>18 24 42 7E 42 42 42 00<br>18 24 42 7E 42 42 42 00<br>という文字を出力します。 |

<table>
<tr><th></th><th>エスケープコード</th><th>16進コード</th><th>機　　能</th></tr>
<tr><td>37</td><td>ESC I</td><td>1BH 49H</td><td>● X1 turbo を初期状態に戻します。すなわち，次のことを行ないます。<br>① COLOR 7（白）に設定します。<br>② アトリビュートを07（ROM キャラクターゼネレータ，標準サイズ）に設定します。<br>③ パレットコードをすべて黒に設定します。<br>④ テキスト画面をグラフィック画面に対し，最優先に設定します。<br>⑤ サウンドを OFF にします。<br>⑥ キーのクリックを ON にします。<br>⑦ カーソルを表示させます。<br>⑧ ファンクションキーの状態を最下行に表示します。</td></tr>
<tr><td>38<br>〜<br>43</td><td>ESC V 0〜5</td><td>1BH 56H<br>30H〜35H</td><td>● テキスト画面に表示する文字の行数と，グラフィック画面の解像度を設定します。<br><table><tr><th>続く文字</th><th>テキスト画面</th><th>グラフィック画面</th></tr><tr><td>0</td><td>25行</td><td>200ライン</td></tr><tr><td>1</td><td>12行</td><td>192ライン</td></tr><tr><td>2</td><td>20行</td><td>表示不可能</td></tr><tr><td>3</td><td>10行</td><td>表示不可能</td></tr><tr><td>4</td><td>25行</td><td>400ライン</td></tr><tr><td>5</td><td>12行</td><td>384ライン</td></tr></table>テキスト画面を10行か20行に設定した場合，グラフィック画面を使用することはできません。また10行の指定は標準ディスプレイモードの場合のみ使用可能です。解像度はその時の RESOLUTION スイッチの状態によります。</td></tr>
<tr><td>44</td><td>ESC S<br>$H_1H_1H_1$<br>$H_2H_2H_2$<br>$H_3$</td><td>1BH 53H<br>$h_1h_1h_1$<br>$h_2h_2h_2$<br>$h_3$</td><td>● グラフィック画面に点を表示します。<br>● X 座標を $H_1H_1H_1$，Y 座標を $H_2H_2H_2$，パレットコードを $H_3$の16進数字で与えます。<br>値のとりうる範囲は，<br>$H_1H_1H_1$が000H〜27FH　(0〜639)<br>$H_2H_2H_2$が000H〜18FH　(0〜399)<br>$H_3$　が　0H〜7H　(0〜7)<br>となります。<br>注）初期状態では，パレットがすべて黒に設定されグラフィック画面が表示されません。表示のためにはパレット(ESC P $H_1$ $H_2$)を設定する必要があります。</td></tr>
<tr><td>45</td><td>ESC G<br>$H_1H_1H_1$<br>$H_2H_2H_2$</td><td>1BH 47H<br>$h_1h_1h_1$<br>$h_2h_2h_2$</td><td>● キー入力バッファをクリアした後，座標（$H_1H_1H_1$，$H_2H_2H_2$）にある画面の点のパレットコードを最初の1バイトに書き込みます。データはすべて16進数字で与えられます。</td></tr>
</table>

<table>
<tr><th></th><th>エスケープコード</th><th>16進コード</th><th>機　　能</th></tr>
<tr><td>46</td><td>ESC p $H_1H_1$</td><td>1BH 70H $h_1h_1$</td><td>テキスト画面のグラフィック画面に対する優先順位を設定します。Hには，ASCII 文字の 0～9 および A～F が入り，16進数字を構成します。よって，HH の 2 バイトで00H～FFH までの数字で表わします。<br>● 優先順位の設定は，次のビットパターンに従って行ないます。<br><br>ビット　7 6 5 4 3 2 1 0<br>白 | 黄 | シアン | 緑 | マゼンダ | 赤 | 青 | 黒<br><br>0 のとき　テキスト優先<br>1 のとき　グラフィック優先</td></tr>
<tr><td>47<br>～<br>50</td><td>ESC v 0～3</td><td>1BH 56H<br>30H～33H</td><td>● グラフィック画面の解像度が 200 ラインか 192 ラインの場合のみ，2 つのグラフィック RAM（GRAM 0 と 1）を表示用か，書き込み用かに指定できます。<br><br>
<table>
<tr><th>Vの値</th><th>表　示</th><th>書き込み</th></tr>
<tr><td>0</td><td>GRAM0</td><td>GRAM0</td></tr>
<tr><td>1</td><td>GRAM0</td><td>GRAM1</td></tr>
<tr><td>2</td><td>GRAM1</td><td>GRAM0</td></tr>
<tr><td>3</td><td>GRAM1</td><td>GRAM1</td></tr>
</table>
例えば，V＝1 の場合，画面には，GRAM0の内容が表示され，グラフィックに対して書き込みを行なった場合，GRAM1に書き込まれます。</td></tr>
<tr><td>51</td><td>ESC K</td><td>1BH 4BH</td><td>● キーのクリック音を出すモードにします。<br>● このコードが出力された後，キーボードのキーを押すと「クリッ」という音が出ます。</td></tr>
<tr><td>52</td><td>ESC k</td><td>1BH 6BH</td><td>● キーのクリック音を出さないモードにします。</td></tr>
</table>

| | エスケープコード | 16進コード | 機　　能 |
|---|---|---|---|
| 53 | ESC W $H_1H_2H_2$ | 1BH 57H $h_1h_2h_2$ | ●サウンドゼネレータに，データを書き込み，音を出します。$H_1$にレジスタ番号，$H_2H_2$にデータを、16進数で書きます。<br>サウンドゼネレータ用LSI(AY 38910)へ直接データを送りこむことによって音を出します。 |

下の図でレジスターとデータとの関係を示します。

| レジスタ番号 | ビット / レジスターの内容 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | チャンネルAの周波数 | 8ビット　FTA | | | | | | | |
| 1 | （音程） | 未　使　用 | | | | 4ビットCTA | | | |
| 2 | チャンネルBの周波数 | 8ビット　FTB | | | | | | | |
| 3 | （音程） | 未　使　用 | | | | 4ビットCTB | | | |
| 4 | チャンネルCの周波数 | 8ビット　FTC | | | | | | | |
| 5 | （音程） | 未　使　用 | | | | 4ビットCTC | | | |
| 6 | ノイズの周波数 | 未使用 | | | 5ビット データ | | | | |
| 7 | チャンネルの設定 | IN/OUT | | ノイズ | | | トーン | | |
| | | IOB | IOA | C | B | A | C | B | A |
| 8 | チャンネルAの音量 | 未　使　用 | | | M | L3 | L2 | L1 | L0 |
| 9 | チャンネルBの音量 | 未　使　用 | | | M | L3 | L2 | L1 | L0 |
| A | チャンネルCの音量 | 未　使　用 | | | M | L3 | L2 | L1 | L0 |
| B | エンベロープ周期 | 8ビット　CT | | | | | | | |
| C | | 8ビット　FT | | | | | | | |
| D | エンベロープ形状 | 未　使　用 | | | | E3 | E2 | E1 | E0 |
| E | I/OポートA※ | 8ビット（ジョイスティック1） | | | | | | | |
| F | I/OポートB※ | 8ビット（ジョイスティック2） | | | | | | | |

※

| レジスター7のD7 | レジスター7のD6 | 内　　容 |
|---|---|---|
| 0 | 0 | ポートA，ポートB共に出力 |
| 0 | 1 | ポートAは出力，ポートBは入力 |
| 1 | 0 | ポートAは入力，ポートBは出力 |
| 1 | 1 | ポートA，ポートB共に出力 |

| | | | |
|---|---|---|---|
| 54 | ESC R $H_1$ | 1BH 52H $h_1$ | ●キー入力バッファをクリアした後，サウンドゼネレータのレジスタ $H_1$ のデータ内容をキー入力バッファの最初から2バイトに書き込みます。<br>データはすべて16進数字で与えられます。 |

## ■ダイレクトカーソルアドレシングの座標

（ESC＝，ESC ？ 使用時）

| X，Y | コード | | X，Y | コード | | X，Y | コード | |
|---|---|---|---|---|---|---|---|---|
| | アスキー文字 | 16進コード | | アスキー文字 | 16進コード | | アスキー文字 | 16進コード |
| 0 | SP(スペース) | 20H | 27 | ; | 3BH | 54 | V | 56H |
| 1 | ! | 21H | 28 | < | 3CH | 55 | W | 57H |
| 2 | " | 22H | 29 | = | 3DH | 56 | X | 58H |
| 3 | # | 23H | 30 | > | 3EH | 57 | Y | 59H |
| 4 | $ | 24H | 31 | ? | 3FH | 58 | Z | 5AH |
| 5 | % | 25H | 32 | @ | 40H | 59 | [ | 5BH |
| 6 | & | 26H | 33 | A | 41H | 60 | ¥ | 5CH |
| 7 | ▼ | 27H | 34 | B | 42H | 61 | ] | 5DH |
| 8 | ( | 28H | 35 | C | 43H | 62 | ^ | 5EH |
| 9 | ) | 29H | 36 | D | 44H | 63 | _ | 5FH |
| 10 | * | 2AH | 37 | E | 45H | 64 | ` | 60H |
| 11 | + | 2BH | 38 | F | 46H | 65 | a | 61H |
| 12 | , | 2CH | 39 | G | 47H | 66 | b | 62H |
| 13 | − | 2DH | 40 | H | 48H | 67 | c | 63H |
| 14 | . | 2EH | 41 | I | 49H | 68 | d | 64H |
| 15 | / | 2FH | 42 | J | 4AH | 69 | e | 65H |
| 16 | 0 | 30H | 43 | K | 4BH | 70 | f | 66H |
| 17 | 1 | 31H | 44 | L | 4CH | 71 | g | 67H |
| 18 | 2 | 32H | 45 | M | 4DH | 72 | h | 68H |
| 19 | 3 | 33H | 46 | N | 4EH | 73 | i | 69H |
| 20 | 4 | 34H | 47 | O | 4FH | 74 | j | 6AH |
| 21 | 5 | 35H | 48 | P | 50H | 75 | k | 6BH |
| 22 | 6 | 36H | 49 | Q | 51H | 76 | l | 6CH |
| 23 | 7 | 37H | 50 | R | 52H | 77 | m | 6DH |
| 24 | 8 | 38H | 51 | S | 53H | 78 | n | 6EH |
| 25 | 9 | 39H | 52 | T | 54H | 79 | o | 6FH |
| 26 | : | 3AH | 53 | U | 55H | | | |

# 第9章

# システムコール

# 1 システムコールの使い方

システムコールとは，ファンクションコールとも呼ばれ，CP/M システムがサブルーチンとしてもっている各周辺機器との入出力や，ディスク上にあるファイル操作などの機能を，ユーザープログラムで利用して行なうことです。このシステムコールを利用すると CP/M システムを利用してのプログラム作成の負担が軽くなり，各プログラム間の互換性が保てるようになります。
本 CP/M システムでは39個のシステムコールが使用できます。

## システムコールの基本的な使用方法

| | | |
|---|---|---|
| 入力 | 入力パラメータが1バイトのとき | MVI C，ファンクション No.<br>（ファンクション No.をCレジスタにロード）<br>MVI E，入力パラメータ<br>（入力パラメータをEレジスタにロード）<br>CALL 0005H<br>（BDOS ジャンプベクトルが格納されている 0005H をコール） |
| | 入力パラメータが2バイトのとき | MVI C，ファンクション No.<br>LXI D，入力パラメータ<br>（入力パラメータを DE レジスタにロード）<br>CALL 0005H |
| 出力 | 出力パラメータが1バイトのとき | Aレジスタに格納されます |
| | 出力パラメータが2バイトのとき | HL レジスタに格納されます |

※出力に際しては，他のレジスタが破壊されることがあります。

## サンプルプログラム

具体的なシステムコールの利用方法としてシステムコール No. 6とシステムコール No. 15についてサンプルプログラムを示します。

①システムコール No. 6

```
INPUT   ORG     100H
        MVI     C,06H
        MVI     E,0FFH
        CALL    0005H
        CPI     19H
        JZ      0000H
        MVI     C,06H
        MOV     E,A
        CALL    0005H
        JMP     INPUT
        END
```

- MVI C,06H / MVI E,0FFH / CALL 0005H：コンソール(CON：)からの入力指定
- CPI 19H / JZ 0000H：CTRL + Y が入力されると終了して CP/M システムに戻ります。
- MVI C,06H / MOV E,A / CALL 0005H：Aレジスタに格納されている文字をコンソール(CON：)へ出力指定
- JMP INPUT：ループ

このプログラムを実行するとキーボードから入力された文字が CRT ディスプレイ上に出力され，CTRL + Y を入力すると終了します。

②システムコール No. 15

まず，DIR を実行すると次のように画面上にリストアウトされます。

```
A>DIR
A: MOVCPM   COM : CPM56    SYS : STARTUP  COM : STAT     COM
A: PIP      COM : ED       COM : ASM      COM : LOAD     COM
A: DDT      COM : DUMP     COM : DUMP     ASM : SUBMIT   COM
A: XSUB     COM : FORMAT   COM : COPY     COM : SYSGEN   COM
A: DICCOPY  COM : DICSET   COM : DISKSET  COM : HDMAINT  COM
A: HDBACKUP COM : HDOFF    COM : TERM     COM : CMT      COM
A: KEY      COM : TIME     COM : ULPT0    ASM : ULPT1    ASM
A: ULPT2    ASM : BIOS     MAC : WM       COM : WM       HLP
A: WMENGL   HLP : WMKANA   HLP : WMKANJI  HLP : ONKUN    DIC
```

ディレクトリコード 0　ディレクトリコード 1　ディレクトリコード 2　ディレクトリコード 3

次に DDT を起動し，システムコール No. 15を実行するために以下のような形で入力していきますと，そのファイルに応じたディレクトリコードがAレジスタに格納されていきます。

```
A>DDT↵
DDT VERS 2.2
-A2000
2000  MVI C,0F↵
2002  LXI D,005C↵
2005  CALL 0005↵
2008  NOP↵
2009  ↵
-IPIP.COM↵
-G2000,2008↵
*2008
-XA ↵
A=00
-IABC.XYZ ↵
-G2000,2008↵
*2008
-XA ↵
A=FF
-
```

システムコール No. 15の設定
（FCB の先頭アドレスは005CH）
PIP. COM の FCB をセット
システムコール No. 15の実行
Aレジスタの内容確認
ディレクトリコード 0
ABC. XYZ の FCB をセット
システムコール No. 15の実行
Aレジスタの内容確認
ディレクトリコード0FFH(該当ファイルなし)

※ディレクトリコードについては，後述のシステムコール詳細のファイルのオープン（システムコール15）を参照してください。

## 2 システムコール一覧

### システムコール(ファンクションコール)の一覧表

| システムコールNo<br>(ファンクションコールNo)<br>(Cレジスタ) | 機能 | 入力パラメータ<br>(Eレジスタまたは DE レジスタ) | 出力パラメータ<br>(レジスタ) |
|---|---|---|---|
| 0 | システムリセット | | |
| 1 | コンソール(CON：)からの入力<br>＝キーボードからの入力 | | A＝入力文字 |
| 2 | コンソール(CON：)への出力<br>＝CRT ディスプレイへの出力 | E＝出力文字 | |
| 3 | リーダ(RDR：)からの入力 | | A＝入力データ |
| 4 | パンチ(PUN：)への出力 | E＝出力データ | |
| 5 | リスト(LST：)への出力 | E＝出力データ | |
| 6 | コンソール(CON：)への直接入出力 | E＝FFH(入力指定)<br>E＝出力文字(出力指定) | A＝入力文字(入力指定) |
| 7 | IO バイトのデータ取り出し | | A＝IO バイトのデータ |
| 8 | IO バイトのデータセット | E＝IO バイトの データ | |
| 9 | 文字列のプリントアウト | DE＝文字列の格納先頭アドレス | |
| 10 | コンソールバッファへの読みこみ | DE＝コンソールバッファ先頭アドレス | |
| 11 | コンソール(CON：)<br>ステータスチェック | | A＝FFH(入力あり)<br>A＝00H(入力なし) |
| 12 | バージョン番号の取り出し | | H＝00H(CP/M)<br>H＝01H(MP/M)<br>L＝00H(Version2.0以下)<br>L＝20H(Version2.0)<br>L＝22H(Version2.2) |
| 13 | ディスクシステムのリセット | | |
| 14 | ディスクドライブの選択 | E＝ドライブ番号<br>(0＝A：, 1＝B：, …, 7＝H：) | |
| 15 | ファイルのオープン | DE＝FCB（ファイルコントロールブロック）格納先頭アドレス | A＝ディレクトリコード(0〜3)〔正常なオープン〕<br>A＝FF H(ファイルのオープン不可)<br>FCB＝オープンされたファイルのディレクトリ |

| | | | |
|---|---|---|---|
| 16 | ファイルのクローズ | DE＝FCB 格納先頭アドレス | A＝ディレクトリコード(0〜3)<br>〔正常なクローズ〕<br>A＝FF H(ファイルのクローズ不可) |
| 17 | 最初のファイルサーチ | DE＝FCB 格納先頭アドレス | A＝ディレクトリコード(0〜3)<br>〔正常なファイルサーチ〕<br>A＝FF H(ファイルサーチ不可)<br>DMA（ダイレクトメモリアドレス）＝該当ファイルが含まれる4つのディレクトリ |
| 18 | 次のファイルサーチ | | A＝ディレクトリコード(0〜3)<br>〔正常なファイルサーチ〕<br>A＝FF H(ファイルサーチ不可) |
| 19 | ファイルの削除 | DE＝FCB 格納先頭アドレス | A＝ディレクトリコード(0〜3)<br>〔正常なファイル削除〕<br>A＝FF H(ファイル削除不可) |
| 20 | シーケンシャルリード | 同上 | A＝00H(正常なリード)<br>A＝00H 以外(ファイルの終了) |
| 21 | シーケンシャルライト | 同上 | A＝00H(正常なライト)<br>A＝00H 以外<br>（書きこみ動作不可） |
| 22 | ファイルの作成 | 同上 | A＝ディレクトリコード(0〜3)<br>〔正常なファイル作成〕<br>A＝FF H(ファイルの作成不可) |
| 23 | ファイル名の変更 | 同上 | A＝ディレクトリコード(0〜3)<br>〔正常なファイル名変更〕<br>A＝FFH（ファイル名の変更不可） |
| 24 | ログインベクトルの取り出し | | HL＝ログインベクトル |
| 25 | ログインディスク番号の取り出し | | A＝ログインディスク番号<br>(0＝A：,1＝B：,…,7＝H：) |
| 26 | DMA アドレスのセット | DE＝DMA 先頭アドレス | |
| 27 | アロケーションアドレスの取り出し | | HL＝アロケーションベクトルの格納先頭アドレス |
| 28 | ディスクのライトプロテクト | | |
| 29 | リードオンリーベクトルの取り出し | | HL＝リードオンリーベクトル |
| 30 | ファイルの属性セット | DE＝FCB 格納先頭アドレス | A＝ディレクトリコード(0〜3)<br>〔正常な属性セット〕<br>A＝FFH(属性セット不可) |

| | | | |
|---|---|---|---|
| 31 | ディスクパラメータアドレスの取り出し | | HL＝DPB（ディスクパラメータブロック）の格納先頭アドレス |
| 32 | ユーザーコードのセットまたは取り出し | E＝ユーザーコード（ユーザーコードのセット）〔E＝00Hのとき USER0, …, E＝0FHのとき USER15〕<br>E＝FF H（ユーザーコードの取り出し） | |
| 33 | ランダムな読み出し | DE＝FCB 格納先頭アドレス（FCB の r0, r1, r2はアクセスレコード番号セット） | A＝00H（正常な読み出し）<br>A＝00H 以外（読み出し不可）〔エラーコード〕 |
| 34 | ランダムな書きこみ | 同上 | 同上 |
| 35 | ファイルの大きさの計算 | DE＝FCB 格納先頭アドレス | FCB の r0, r1, r2にファイルの大きさ（レコード数）をセット |
| 36 | ランダムレコードのセット | DE＝FCB 格納先頭アドレス | FCB の r0, r1, r2にランダムレコード番号をセット |
| 37 | ディスクドライブのリセット | DE＝ドライブベクトル | |
| 40 | 0（ゼロ）書きこみを伴うランダムな書きこみ | DE＝FCB 格納先頭アドレス | A＝00H（正常な書きこみ）<br>A＝00H 以外（書きこみ不可）〔エラーコード〕 |

※システムコール No.12の「MP/M」は，デジタルリサーチ社の OS（オペレーティングシステム）です。

# 3 システムコール詳細

## 1 システムコール0：システムリセット

**入力パラメータ**

Cレジスタ：00H

**機　能**

CP/Mシステムのリブートを行ないます。

## 2 システムコール1：コンソールからの入力

**入力パラメータ**

Cレジスタ：01H

**リターン時の値**

Aレジスタ：入力文字

**機　能**

コンソール（キーボード）からの入力をAレジスタに格納すると同時にコンソール（画面）へエコーバックします。

入力があるまで内部でループします。

## 3 システムコール2：コンソールへの出力

**入力パラメータ**

Cレジスタ：02H

Eレジスタ：出力文字

**機　能**

Eレジスタに格納されている文字をコンソール（画面）に出力します。

## 4 システムコール3：リーダからの入力

**入力パラメータ**

Cレジスタ：03H

**リターン時の値**

Aレジスタ：入力データ

**機　能**

リーダ（RDR：）からの入力データを，Aレジスタに格納します。入力があるまで内部でループします。

## 5 システムコール4：パンチへの出力

**入力パラメータ**

Cレジスタ：04H

Eレジスタ：出力データ

**機　能**

Eレジスタに格納されているデータをパンチ（PUN：）に出力します。

## 6 システムコール5：リストへの出力

**入力パラメータ**

Cレジスタ：05H

Eレジスタ：出力データ

**機　能**

Eレジスタに格納されているデータをリスト（LST：）に出力します。

## 7 システムコール6：コンソールへの直接入出力

**入力パラメータ**

Cレジスタ：06H

Eレジスタ：FF H（入力を指定する場合）

：出力文字（出力を指定する場合）

**リターン時の値**

Aレジスタ：入力文字（入力を指定した場合）

**機　能**

EレジスタにFF Hを格納し，このシステムコールを実行した場合は，コンソール入力となり，FF H以外の文字を格納した場合は，その文字がコンソールに出力されます。

Aレジスタにはコンソールからの入力文字を入れて戻し，もし入力がないときは，00Hを入れて戻ってきます。

このシステムコールでは，内部でループせず，さらに CTRL +P，CTRL +S などのコントロール機能は無視されます。

## 8 システムコール7：IOバイトのデータ取り出し

**入力パラメータ**

Cレジスタ：07H

**リターン時の値**

Aレジスタ：IOバイトのデータ

**機　能**

メモリ上の0003H番地にあるIOバイトのデータをAレジスタに格納します。

## 9 システムコール8：IOバイトのデータセット

**入力パラメータ**

Cレジスタ：08H

Eレジスタ：IOバイトのデータ

**機　能**

Eレジスタに入っているIOバイトのデータをメモリ上の0003H番地に書き込みます。

## 10 システムコール 9 ：文字列のプリントアウト

**入力パラメータ**

Cレジスタ：09H

DEレジスタ：文字列の格納されている先頭番地

**機　能**

DEレジスタで示された先頭番地より格納された文字列を，文字列の最後を示す"$"（24H）の文字が来るまで，コンソールに出力します。

## 11 システムコール10：コンソールバッファへの読み込み

**入力パラメータ**

Cレジスタ：0AH

DEレジスタ：コンソール入力バッファの先頭番地

**リターン時の値**

コンソールからの入力が入力バッファに格納

**機　能**

コンソールからの入力を，（DEレジスタで示された入力バッファの先頭番地＋2）番地より読み込みます。このファンクションコールから戻るのは，↵（キャリジ・リターン）が入力されるか，入力バッファがオーバーフロー（256文字以上の入力）した時です。入力バッファは，次のような構成となっています。

| mx | nc | $C_1$ | $C_2$ | $C_3$ | ····· | ····· | ····· | Cn | ····· | ····· | ····· | ····· | $C_{254}$ | $C_{255}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

↑

**DEレジスタで示される番地**

mx：あらかじめセットした，入力バッファに保持できる最大文字数（1文字〔01H〕～255文字〔FFH〕）が入ります。

nc：システムコールから戻ったとき，実際に入力された文字数がセットされます。

$C_1$～$C_{255}$：コンソールからの入力文字

## 12 システムコール11：コンソールステータスチェック

**入力パラメータ**

Cレジスタ：0BH

**リターン時の値**

Aレジスタ：コンソールステータス

**機　能**

コンソールからの入力の有無をチェックします。

入力があった場合には，AレジスタにFF Hを格納し，入力がない場合には，00Hを格納します。

## 13 システムコール12：バージョン番号の取り出し

**入力パラメータ**

Cレジスタ：0CH

**リターン時の値**

HLレジスタ：バージョン番号

**機　能**

オペレーティングシステムが，CP/Mであるか MP/Mであるか，またCP/Mである場合のバージョン番号をHLレジスタに格納します。

Hレジスタ＝00Hのとき CP/M
Hレジスタ＝01Hのとき MP/M
Lレジスタ＝00Hのときバージョン2.0以前
Lレジスタ＝20Hのときバージョン2.0
Lレジスタ＝22Hのときバーション2.2

※ MP/Mとはデジタル・リサーチ社が出しているOS（オペレーティング・システム）です。

## 14 システムコール13：ディスクシステムのリセット

**入力パラメータ**

Cレジスタ：0DH

**機　能**

すべてのディスクシステムを初期状態にセットします。

すべてのドライブは"R/W"（リード・ライト可能）となり，そして，ドライブAをオンライン状態にし，ログインディスクに戻ります。また，DMA先頭番地は，0080Hにセットされます。

※オンラインについて

オンライン状態とは，ログインディスクのチェンジなどにより，ディスクのディレクトリ情報が，CP/Mシステムに取り込まれた状態のことです。

## 15 システムコール14：ディスクドライブの選択

**入力パラメータ**

Cレジスタ：0EH

Eレジスタ：ドライブ番号（0＝A：，1＝B：，……，7＝H：）

**機　能**

各種ファイル操作を，Eレジスタにより指定されたディスクドライブ上で行ないます。しかし，FCBのdrフィールドの値が00H以外の場合は，直接選択されたドライブで，ファイル操作が行なわれます。

## 16 システムコール15：ファイルのオープン

**入力パラメータ**

Cレジスタ：0FH

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

先頭番地がDEレジスタで示されるFCBに格納されているファイルを，該当ディスク上のディレクトリから探し，同じファイルがあると，そのディレクトリのディスクアロケーションマップのデータをFCBのd0からd15フィールドにコピーし，今後のファイルの読み出し，書き込み機能ができるようにします。なお，ファイルのオープンが正常にできると，Aレジスタには，0から3までのディレクトリコードが入り，ファイルのオープンができなかったときは，AレジスタにFF Hのディレクトリコードが入ります。

※ディレクトリコード

BDOSは，4つのディレクトリを1単位として，ファイルを探していきます。そして，ファイルがみつけられると4つのディレクトリの何番目にファイルがあるかということ，すなわち，0から3までの値をAレジスタに格納します。一方，ファイルがみつからないときは，FF HをAレジスタに格納します。このようにして，Aレジスタに格納される値をディレクトリコードと呼びます。一般に，ディレクトリコードは，前述のサンプルプログラム②で示したようにDIR ↵ でリストアウトされるファイルの順番に関係しています。

## 17 システムコール16：ファイルのクローズ

**入力パラメータ**

Cレジスタ：10H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

ディスク上のディレクトリに最新のFCBの内容を記録し，ディスク上のディレクトリのデータを更新します。ただし，ファイルの読み出しのみ行なった場合は，ファイルのクローズの必要はありません。また，ファイルのクローズが正常にできた時はAレジスタには，0から3までのディレクトリコードが入り，ファイルのクローズができなかったときは，Aレジスタには，FF Hのディレクトリコードが入ります。

## 18 システムコール17：最初のファイルサーチ

**入力パラメータ**

Cレジスタ：11H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

DEレジスタで示されるFCBにあるファイルと一致するファイルをディスク上のディレクトリから探します。ファイルが見つからないときは，Aレジスタに FF Hのディレクトリコードが入り，ファイルが見つかったときは，Aレジスタに0から3までのディレクトリコードが入ります。また，ファイルが見つかったときは，DMAバッファに，該当ファイルが含まれる4つのディレクトリ（128バイト）が格納されます。

この場合該当ファイルのディレクトリ部の先頭番地は，

DMAの先頭番地＋ディレクトリコード×32

で求められます。

たとえば，DMAの先頭番地が0080H（リブートしたときの設定値），ディレクトリコードが2のとき，該当ファイルのディレクトリ部の先頭番地は，

0080H＋20H×2＝00C0H

となります。すなわち，4つのディレクトリの各ディレクトリ部の先頭番地は，0080H，00A0H，00C0H，00E0Hの値をとることになります。このことは，1つのディレクトリが，32バイトで構成されていることを示します。

## 19 システムコール18：次のファイルサーチ

**入力パラメータ**

Cレジスタ：12H

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

システムコール17では，ディスク上のディレクトリの最初のほうからファイルをサーチしますが，システムコール18では，前回のファイルのサーチで一致したファイルの後のディレクトリを対象にファイルをサーチします。（たとえば，ファイルをグループで指定した場合などに使います。）なお，ディレクトリコードの意味は，システムコール17と同じです。

## 20 システムコール19：ファイルの削除

**入力パラメータ**

Cレジスタ：13H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

先頭番地がDEレジスタで示されるFCBにあるファイルと一致したディスク上のファイルを削除します。該当ファイルが削除された場合は，Aレジスタに0から3までのディレクトリコードが入り，削除されなかった場合には，AレジスタにFFHのディレクトリコードが入ります。

## 21 システムコール20：シーケンシャルリード

**入力パラメータ**

Cレジスタ：14H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：00Hまたは00H以外の値

**機　能**

先頭番地がDEレジスタで示されるFCB内のcrフィールドが示す1レコードをDMAバッファに格納します。その後，crの値を1だけ増やし，次のレコードを示すようにします。もし，crがオーバーフローするときは，つぎのロジカルエクステントをオープンし，crを00Hにセットします。

読み出し動作が正常に行なわれたときは，Aレジスタに00Hの値が入り，次のレコード位置にデータがないときは，Aレジスタに00H以外の値が入り，ファイルの終了を知らせます。なお，このシステムコールを行なう前に，ファイルのオープンをしておかなければなりません。

※ロジカルエクステントについて

turbo CP/Mでは，1つのディレクトリで32Kバイトまでの領域を，ディスク上で指定することができます。ロジカルエクステントとは，1つのディレクトリで指定できる32Kバイトの領域を示しており，1つのファイル容量が32Kバイトを越える場合，32Kバイトごとに1ロジカルエクステント（ディレクトリ）を使用します。

## 22 システムコール21：シーケンシャルライト

**入力パラメータ**

Cレジスタ：15H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：00H又は00H以外の値

**機　能**

先頭番地がDEレジスタで示されるFCB内のcrフィールドが示すレコードに，DMAバッファの128バイトのデータを書き込みます。crの値は自動的に1だけ増え，次のレコードを示します。また，crがオーバーフローしたときは，自動的に次のロジカルエクステントをオープンし，crを00Hにセットします。書き込み動作が正常に行なわれたときは，Aレジスタに00Hの値が入り，書き込み動作ができなかったときは，Aレジスタに00H以外の値が入ります。

なお，このシステムコールを行なう前に，ファイルのオープンをしておかなければなりません。

## 23 システムコール22：ファイルの作成

**入力パラメータ**

Cレジスタ：16H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

先頭番地がDEレジスタで示されるFCBのファイルをディスク上のディレクトリに登録します。登録ファイル名は，同一ディスク上にあるファイルと同じファイル名であってはいけません。ファイルが作成されますと，そのファイルはオープンされており，読み出し，書き込み動作が可能になっています。なお，ファイルの作成が正常に行なわれたときには，Aレジスタに0から3までのディレクトリコードが入り，ファイルの作成ができなかったときは，AレジスタにFF Hのディレクトリコードが入ります。

## 24 システムコール23：ファイル名の変更

**入力パラメータ**

Cレジスタ：17H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

FCB内での，最初の16バイトで示されるディスク上のファイル名を次の16バイトで示されるファイル名に変更します。ファイル名が正常に変更されたなら，Aレジスタに0から3までのディレクトリコードが入り，ファイル名が変更されないときは，AレジスタにFFHのディレクトリコードが入ります。

## 25 システムコール24：ログインベクトルの取り出し

**入力パラメータ**

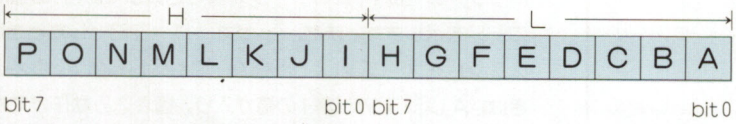
Cレジスタ：18H

**リターン時の値**

HLレジスタ：ログインベクトル

**機　能**

ログインベクトルをHLレジスタに取り込みます。各ドライブ(A：～H：)に対するオンライン，オフラインのビット対応は次のようになっています。




それぞれのビットが { 0のときオフライン / 1のときオンライン

※X 1 turboではドライブAからドライブHまで，サポートしております。

## 26 システムコール25：ログインディスク番号の取り出し

**入力パラメータ**

Cレジスタ：19H

**リターン時の値**

Aレジスタ：ログインディスク番号

**機　能**

現在のログインディスク番号をAレジスタに格納します。A=00Hのときドライブ A，A=01Hのときドライブ Bというようになります。

## 27 システムコール26：DMA アドレスのセット

**入力パラメータ**

Cレジスタ：1AH

DEレジスタ：DMA 先頭番地

**機　能**

DEレジスタで示される先頭番地よりDMAを設定します。DMAアドレスは，リブートもしくはディスクシステムのリセット（システムコール13）を行なうと，DMA先頭番地は，0080Hにセットされます。

## 28 システムコール27：アロケーションアドレスの取り出し

**入力パラメータ**

Cレジスタ：1BH

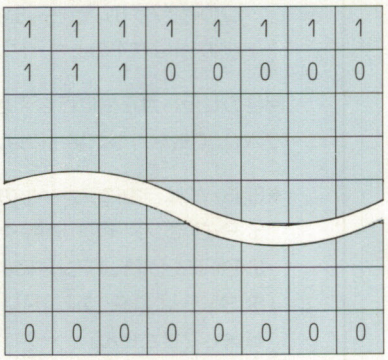
**リターン時の値**

HLレジスタ：アロケーションベクトル先頭番地

**機　能**

ログインされているディスクのアロケーションベクトルの先頭番地をHLレジスタに格納します。アロケーションベクトルとは，フロッピーディスクの使用，未使用エリアを示すもので，メモリのBIOS部にアロケーションベクトルのデータが記憶してあります。なお，turbo CP/Mでは，1バイトのデータで2Kバイトのデータブロック8個の使用，未使用状況を表わすことができます。

HLレジスタに格納されている番地 ⇨




アロケーション・ベクトルのバッファエリア

メモリーのそれぞれのビットが { 0のとき未使用のエリア / 1のとき使用中のエリア } を表わします。

## 29 システムコール28：ディスクのライトプロテクト

**入力パラメータ**

Cレジスタ：1CH

**機　能**

現在のログインディスクを，書き込み禁止にします。

書き込み禁止の状態は，リブートもしくはディスクシステムのリセット（システムコール13）を行なうと，解除されます。

## 30 システムコール29：リードオンリーベクトルの取り出し

**入力パラメータ**

Cレジスタ：1DH

**リターン時の値**

HLレジスタ：リードオンリーベクトル

**機　能**

書き込み禁止となったディスクドライブをHLレジスタに取り込みます。形式は，システムコール24の形式と同じです。すなわち各ビットが0のとき，R/W（読み出し，書き込み可），1のときR/O（書き込み禁止）となります。

## 31 システムコール30：ファイルの属性セット

**入力パラメータ**

Cレジスタ：1EH

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：ディレクトリコード

**機　能**

FCBで示されるディスクファイルの属性をセットします。属性のセットは，FCBのt1，t2のビット7により行なわれます。

t1のbit7が0のとき　$ R/W(リード/ライト)を与えます。
t1のbit7が1のとき　$ R/O(リード/オンリー)を与えます。
t2のbit7が0のとき　$ DIRを与えます。
t2のbit7が1のとき　$ SYSを与えます。

属性のセットが正常に行なえた場合，Aレジスタに0から3までのディレクトリコードが入り，できなかった場合には，AレジスタにFF Hのディレクトリコードが入ります。

※$ R/Wは自由にファイルの削除や一部の変更ができます。
$ R/Oはファイルの削除や一部変更ができません。
$ DIRはDIRコマンドにより表示される通常のファイルです。
$ SYSはDIRコマンドによる表示には現われません。また，PIPコマンドによる転送もできません。

## 32 システムコール31：ディスクパラメータアドレスの取り出し

**入力パラメータ**

Cレジスタ：1FH

**リターン時の値**

HLレジスタ：ディスクパラメータブロックの先頭番地

**機　能**

BIOS部分にあるディスクパラメータブロックの先頭番地を，HLレジスタに格納します。ディスクの諸元を知るときや，パラメータを変更するときに使用します。

## 33 システムコール32：ユーザーコードのセットまたは取り出し

**入力パラメータ**

Cレジスタ：20H

Eレジスタ：ユーザーコード(ユーザーコードのセット)
　　：FF H（ユーザーコードの取り出し）

**リターン時の値**

Aレジスタ：現ユーザーコード

**機　能**

Eレジスタに示されるユーザーエリア(E=00Hのとき USER0，E=01Hのとき USER1，…，E=0FHのとき USER15）へ移ります。また，Eレジスタに FFHをセットした場合は，現ユーザーコードをAレジスタに格納します。

## 34 システムコール33：ランダムな読み出し

**入力パラメータ**

Cレジスタ：21H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：リターンコード

**機　能**

FCBで示されるディスクのファイルの任意の指定レコードを，DMAバッファに取り込みます。指定するレコード番号は，FCBのr0，r1，r2フィールドに入れます。通常，r2フィールドには00Hを入れ，r0，r1で0から65535までの数値を表わします。このレコード番号をセットするにはr0に下位データ，r1に上位データを与えます。たとえば，レコード番号511（10進）をセットする場合，r0にFFH，r1に01H，r2に00Hを与えます。また，このシステムコールを行ないますと，Aレジスタには0から6までのリターンコードが入り，次のような意味を表わします。

A＝00H　読み出し正常終了

A＝01H　書き込まれていないレコードを読み出そうとした

A＝02H　ランダムモードにならない

A＝03H　現在のロジカルエクステントをクローズできない

A＝04H　書き込まれていないロジカルエクステントへ読み出しを行なった

A＝05H　読み出しモードにならない

A＝06H　ディスクの物理的限界を越えて読み出しを行なった

## 35 システムコール34：ランダムな書き込み

**入力パラメータ**

Cレジスタ：22H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：リターンコード

**機　能**

DMAバッファにあるデータをFCBのr1，r2フィールドで示されているディスクのレコードに書き込みます。すなわち，システムコール33と逆の働きをします。ただし，リターンコードが5のときは，ディレクトリがオーバーフローしたことを表わしています。

## 36 システムコール35：ファイルの大きさの計算

**入力パラメータ**

Cレジスタ：23H

DEレジスタ：FCBの先頭番地

**リターン時の値**

FCBのr0，r1，r2にファイルの大きさをセット

**機　能**

先頭番地がDEレジスタで示されるFCBにあるファイルの大きさを計算します。すなわち，最終レコード番号までにあるレコード数を，r0，r1，r2に格納します。例えば，最終レコード番号が511（10進）のとき，レコード数は512（10進）レコードありますので，r0=00H，r1=02H，r2=00Hとなります。なお，ランダムファイルの場合，データを書き込んでいない場所があるので，実際に書き込まれたレコード数よりこのシステムコールで求められたレコード数のほうが大きな値となっています。

## 37 システムコール36：ランダムレコードのセット

**入力パラメータ**

Cレジスタ：24H

DEレジスタ：FCBの先頭番地

**リターン時の値**

FCBのr0，r1，r2に，ランダムレコード番号をセット

**機　能**

シーケンシャルに，リード／ライトしたレコード番号に1を加えたレコード番号をFCBのr0，r1，r2フィールドにセットし，ランダムなリード／ライトを可能にします。

## 38 システムコール37：ディスクドライブのリセット

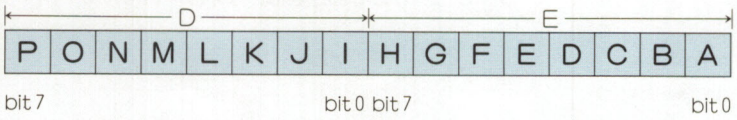
**入力パラメータ**

Cレジスタ：25H

DEレジスタ：ドライブベクトル

**機　能**

DEレジスタの16ビットのビットパターンによって任意のディスクドライブをリセットします。




ビットが1のとき，そのディスクドライブがリセットされます。

## 39 システムコール40：ゼロ書き込みを伴うランダムな書き込み

**入力パラメータ**

Cレジスタ：28H

DEレジスタ：FCBの先頭番地

**リターン時の値**

Aレジスタ：リターンコード

**機　能**

システムコール34と同じですが，書き込みを行なうブロックをすべて0で満たしてから，指定レコードにデータを書き込みます。

# 第10章

# CP/M BIOSコール

## 1 CP/M BIOS コールの方法

CP/M のディスク等の入出力は，CP/M BDOS により、CP/M BIOS をとおして行われます。この BIOS の先頭には目的の機能を実行するためのサブルーチンへ導く17個のジャンプベクトル(飛び先番地）が並んでおり，このジャンプベクトルは，同じ CP/M サイズであれば，そのアドレス，順序共，すでに決められています。

特に，CP/M BIOS コールはシステムコールと違い，CP/M BDOS 内でのルーチン処理が省かれるので，その分高速になるほか，任意のセクタのリード/ライトが可能であるというメリットがあります。

具体的な CP/M BIOS コールの方法としては，入力パラメータの必要なものは，それぞれのレジスタに入力パラメータをセットして，それぞれのジャンプベクトルの絶対アドレスをコールします。

例．サブルーチン「CONOUT」コールについてのサンプルプログラムを示します。

```
ORG     100H
MVI     C，41H
CALL    0××0 CH
JMP     0000H
END
```

turbo CP/M は標準56k CP/M ですので，××には DA が入ります。このプログラムを実行するとAという文字が表示されます。

## 2 CP/M BIOSコールの種類

| エントリーアドレス | ジャンプベクトル名 | 機　　能 | 入力パラメータ | 出力パラメータ |
|---|---|---|---|---|
| XX00H | BOOT | コールドブート（IPL起動） | | C＝00H |
| XX03H | WBOOT | ウォームブート | | C＝ドライブNo. |
| XX06H | CONST | コンソールの状態チェック | | A＝0FFH（読みこみ可）<br>A＝00H（読みこみ不可） |
| XX09H | CONIN | コンソールからの入力 | | A＝入力文字 |
| XX0CH | CONOUT | コンソールへの出力 | C＝出力文字 | |
| XX0FH | LIST | リストへの出力 | 同　　上 | |
| XX12H | PUNCH | パンチへの出力 | 同　　上 | |
| XX15H | READER | リーダからの読みこみ | | A＝入力文字 |
| XX18H | HOME | セレクトされているディスクのヘッドをトラック00の位置へ戻す | | |
| XX1BH | SELDSK | ディスクドライブの選択 | C＝ドライブNo. | HL＝ディスクパラメータヘッダアドレス |
| XX1EH | SETTRK | トラック番号のセット | BC＝トラックNo. | |
| XX21H | SETSEC | セクタ番号のセット | BC＝セクタNo. | |
| XX24H | SETDMA | DMAアドレスのセット | BC＝DMAアドレス | |
| XX27H | READ | 指定セクタからの読みこみ | | A＝00H（エラー無し）<br>A＝01H（エラー） |
| XX2AH | WRITE | 指定セクタへの書きこみ | | 同　　上 |
| XX2DH | LISTST | リスト装置の状態チェック | | A＝FFH（出力可能）<br>A＝00H（出力不可） |
| XX30H | SECTRAN | 論理セクタから物理セクタへの変換 | BC＝論理セクタ<br>DE＝セクタ変換テーブルアドレス | HL＝物理セクタ |

※turbo CP/Mは，標準で56kCP/Mですので，XXには，DAが入ります。

## 3 直接 BIOS コール（BIOS ROM コール）

turbo CP/M では、X1turbo の BIOS ROM を有効に使用する目的で，BIOS ROM コールルーチンを F600H に用意しています。この BIOS ROM コールルーチンは，CP/M BIOS の WBOOT サブルーチンを実行することにより，メインメモリ上の F600H のアドレスから生成されます。

特に，この BIOS ROM には、システム本体の基本的な入出力ルーチンが集まっており，IPL，キー入力，画面表示，プリンタ出力，カセット入出力，ディスクのリード/ライト，グラフィックルーチン，浮動小数点演算，関数，モニタ機能などのルーチンも，もっています。

具体的な BIOS ROM コールの方法としては，まず，各レジスタ，ワーク等に必要な入力パラメータをセットし，次に BC レジスタにコールしたい BIOS ROM のエントリーアドレスを入れ，F600H をコールします。（もし，BIOS ROM 内でエラーが発生した場合，キャリーフラグが 1，エラーが発生しなかった場合，キャリーフラグが 0 でサブルーチンより戻ってきます。）

但し，キャリーフラグを判断材料とするような BIOS ROM サブルーチンを利用する場合は，使用できません。

例. BIOS ROM 内のサブルーチン「BEEP」コールについてのサンプルプログラムを示します。

```
ORG     100H
LXI     B，1B41H
CALL    0F600H
JMP     0000H
END
```

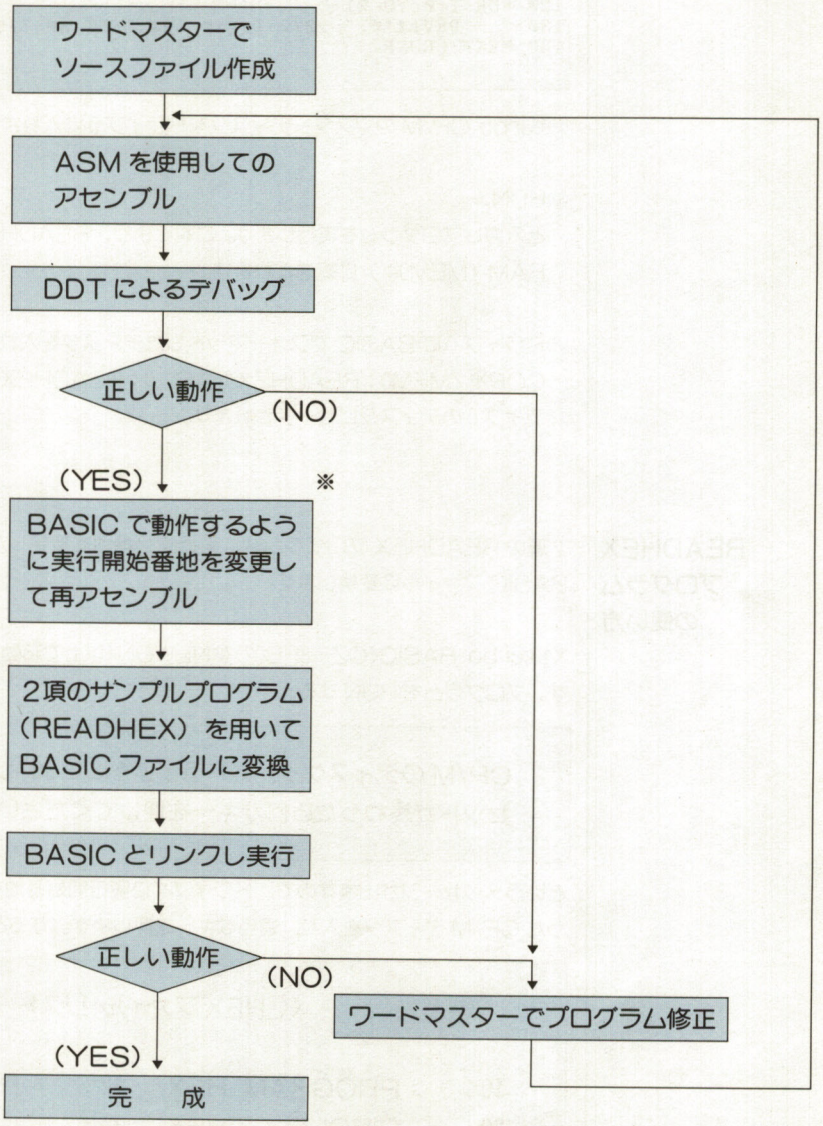
・このプログラムを実行すると，BEEP 音がなります。

# 第11章

# BASICとのリンクについて

## 1 BASICとのリンク方法

X1 turbo BASICを使用してプログラムを組む時に，実行速度をさらに高速化させたり，BASICにはない独自の処理ルーチンをつける必要が生じた場合には，その部分をマシン語のサブルーチンプログラムとして組まなければなりません。この場合，BASICのモニタ機能を利用してマシン語を直接入力することも可能ですが，数Kバイト以上の大きなプログラムになるとあまり効率のよい方法とはいえません。そこで，次のようなフローチャートで，まずturbo CP／Mを用いてマシン語プログラムを作成し，BASICとリンクすることができます。




※CP/M上では，マシン語プログラムは通常100Hから実行されますが，turbo BASICではこの番地からマシン語を配置することができません。このため，ソースプログラム中の"ORG 100H"の部分をWMを再起動して番地変更し，再アセンブルしたHEXファイルを作成する必要があります。具体的には，X1 turbo BASIC（CZ-8FB02）のCLEAR文で宣言したアドレスからF3FFHの間の任意のアドレスに変更する必要があります。なお，READHEXプログラムで扱えるCP/M上のHEXファイルは32Kバイトまでです。

## READHEX プログラムディスクの作成

本章2項のサンプルプログラム（READHEX）はディレクトリでは表示されませんが，turbo CP/Mのマスターディスクの空きエリアにデータとして記録されており，直接入力しなくてもご使用いただけます。

①X1 turbo BASIC（CZ-8FB02）を起動後，下記のプログラムを入力してください。

READHEX取り出しプログラムリスト

```
100 SCREEN:OPTIONSCREEN4:INIT"MEM:"
110 OPEN"R",#1,"MEM:READHEX":FIELD #1,128 AS A$,128 AS B$
120 FOR I=0 TO 21
130     DEVI$"0:",1279-I,Z1$,Z2$:LSET A$=Z1$:LSET B$=Z2$:PUT #1,I
140 NEXT:CLOSE
```

②turbo CP/Mのマスターディスクをドライブ0に入れます。

③RUN↵

と入力しプログラムを実行します。これにより，READHEXプログラムがグラフィックRAM（MEM0：）に転送されます。

④ドライブ0にBASICでフォーマットしたディスクを入れます。

COPY "MEM0：READHEX" AS "0：READHEX" ↵と入力することにより，ドライブ0のディスクにセーブされます。

## READHEX プログラムの使い方

2項のREADHEXプログラムは，実行開始番地を変更したHEXファイルを次の手順でBASICファイルに変換します。

X1 turbo BASIC(CZ−8FB02)をNEWON4以上で起動し，READHEXをロードします。プログラムをRUNすると，

```
CP/Mのディスケットをドライブ1にセットしてください
セットが終わったら何かキーを押してください
```

というメッセージが出ますので，ドライブ1に実行開始番地を変更したHEXファイルの入ったCP/Mディスクを入れ，適当なキーを押します。すると画面上に，

```
<< HEXファイル リスト >>

10：    PROGRAM.HEX   12：    PAINT     .HEX
20：    CIRCLE  .HEX  34：    LINE      .HEX
          |                       |

ファイル番号を指定してください……
```

とディスクに含まれるHEXファイルが表示されてキー待ちになります。

ロードしたいHEXファイルの左に表示された番号を入力し，リターンすると，

```
HEXファイルをディスクから読み込んでいます
```

というコメントが出てHEXファイルを読み，ファイルに指定されているアドレスのメモリ番地へその内容を書き込んでいきます。終了すると指定したファイル名に"．COM"を付け，例えば，

```
PROGRAM. COMをセーブしていいですか（Y/N）？
```

と表示されます。Y↵を入力すると，ドライブ0にPROGRAM，COMという名前で，LOADしたデータをSAVEし，

```
PROGRAM. COMをディスクに書き込みました
アドレスは&HXXXX から&HYYYY です
```

というコメントが出て終了します
なお，ドライブが1基しかない場合には，リストの1030行めを

DR＝0

と書き直し，ディスクを差し換えながらご利用ください。

## BASICからの利用方法

E000番地からマシン語プログラムを実行する場合の例を示します。

```
10 CLEAR &HE000 ―――――――――― マシン語領域の確保
20 LOADM "PROGRAM. COM" ―――― マシン語ルーチンをメインメモリへロード
        ⋮
100 CALL &HE000 ―――― マシン語ルーチンの呼び出し
```

100行目のコール文の番地は，ソースプログラムで変更した"ORG"の番地を指定します。マシン語サブルーチンの呼び出しは，CALL文でなく，USR関数を用いても行なえます。BASICでマシン語を利用する方法の詳細は，本体付属のユーザーズマニュアルを参照してください。

## 2 BASIC サンプルプログラム

以下にREADHEXプログラムのリストを示します。使用方法は前項を参照してください。

```
1000 '**** X1 turbo 漢字 CP/M HEX File Read Program ****
1010 '                    *
1020 INIT:CLEAR &HA7FF:DEFINT A-Z:OPTION SCREEN 0:WIDTH 80,25,0,2:CLS4:SCREEN
1030 DR=1
1040 GOSUB "SETMACHINE"
1050 '
1060 GOSUB"CPMIN":GOSUB "FILES"
1070 IF FLAG=0 THEN 1740
1080 PRINT:PRINT SPC(18);:INPUT "ファイル番号を指定してください ...  ",A :PRINT
1090 '
1100 GOSUB"LOADINIT"
1110 IFNM2$<>"HEX"THEN1750
1120 CLS:COLOR4:CFLASH1:LOCATE 16,10
1130 PRINT "ＨＥＸファイルをディスクから読み込んでいます":CFLASH0:COLOR 7
1140 '
1150 POKE &HE000,dr,0                    'drive
1160 POKE &HE006,&HFF,&HA7,&HFF,&HF3     'store address
1170 MEM$(&HE00A,17)=GA$+CHR$(0)         'directory
1180 CALL &HE100
1190 '
1200 A=PEEK(&HE001)                      'error code
1210 IF A=1 THEN 1750
1220 IF A=2 THEN 1760
1230 IF A<>0 THEN 1780
1240 GOSUB "HEXSAVE":INIT:END
1250 '
1260 LABEL"LOADINIT"
1270 BT=(A MOD 4)*32+1:I=((A MOD 8)AND 4)/4
1280 REC=64+INT(A/8):DEVI$ STR$(DR)+":",REC,R$(0),R$(1)
1290 GA$=MID$(R$(I),BT+16,16)
1300 NM1$=MID$(R$(I),BT+1,8):NM2$=MID$(R$(I),BT+9,3)
1310 RETURN
1320 '
1330 LABEL"FILES"
1340 CLS:COLOR5:PRINT SPC(20);"<<    ＨＥＸ ファイル  リスト  >>"
1350 PRINT:COLOR7:FCNT=0:LIN=0:REC=64:FLAG=0
1360 DEVI$ STR$(DR)+":",REC,R$(0),R$(1)
1370 I=0
1380 BT=1
1390 CPM$=MID$(R$(I),BT,13)
1400 IFLEFT$(CPM$,1)<>CHR$(0) THEN1460
1410 IFMID$(CPM$,10,3)<>"HEX" THEN 1460
1420 A$=RIGHT$(CPM$,1):IF A$<>CHR$(0) AND A$<>CHR$(1) THEN 1460
1430 FLAG=1:PRINT SPC(9) ;:PRINTUSING "####";FCNT;
1440 PRINT ":    ";MID$(CPM$,2,8);".";MID$(CPM$,10,3);"     ";
1450 IF LIN=1 THEN LIN=0:PRINT ELSE LIN=1
1460 FCNT=FCNT+1:BT=BT+32
1470 IF BT>127 THEN I=I+1 ELSE 1390
1480 IF I>1 THEN REC=REC+1 ELSE 1380
1490 IF REC>71 THEN PRINT:RETURN ELSE 1360
1500 '
1510 LABEL"CPMIN"
1520 CLS :COLOR4:CFLASH1 :LOCATE15,10
1530 PRINT"ＣＰ／Ｍのディスケットをドライブ";AKCNV$(STR$(DR));
1540 PRINT"にセットして下さい":COLOR7:CFLASH0
1550 LOCATE 15,12:PRINT"セットが終わったら何かキーを押して下さい"
1560 A$=INKEY$:IF A$="" THEN 1560
1570 CLS :RETURN
1580 '
1590 LABEL"HEXSAVE"
1600 LOCATE 5,10:PRINT AKCNV$(NM1$);
1610 INPUT".ＣＯＭをセーブしていいですか  ( Y/N )     ";A$
1620 IF A$<>"Y" AND A$<>"y" THEN END
1630 SADR=CVI(CHR$(PEEK(&HE002),PEEK(&HE003)))
```

```
1640 EADR=CVI(CHR$(PEEK(&HE004),PEEK(&HE005)))
1650 KADR=VARPTR(MOVE$)
1660 CALL STRPTR+CVI(CHR$(PEEK(KADR+1),PEEK(KADR+2)))
1670 SAVEM NM1$+".COM",SADR,EADR
1680 CLS:LOCATE 15,10:PRINT AKCNV$(NM1$);". ＣＯＭをディスクに書き込みました"
1690 LOCATE 15,12:PRINT"アドレスは＆Ｈ";AKCNV$(HEX$(SADR));
1700 PRINT"から＆Ｈ";AKCNV$(HEX$(EADR));"です"
1710 CLS 0:RETURN
1720 '
1730 'ERROR MESSAGE
1740 PRINTSPC(20);"ＨＥＸファイルがありません":BEEP:CLS 0:END
1750 PRINT:PRINT SPC(14);"ＨＥＸファイル以外はロードできません":BEEP:CLS0:END
1760 LOCATE 13,10
1770 PRINT"ＢＡＳＩＣではロードできないアドレスが指定されています":BEEP:CLS0:END
1780 LOCATE 13,10:PRINT"ＨＥＸファイルが正しくありません":BEEP:CLS0:END
1790 '
1800 LABEL "SETMACHINE"
1810 'SUB MACHINE CODE
1820 M1$=CHR$(&HED,&H4B,2,&HE0,&H2A,4,&HE0,&HB7,&HED,&H42,&H23,&HEB,&H2A)
1830 M2$=CHR$(2,&HE0,&HED,&H78,&H77,&H23,3,&H1B,&H7B,&HB2,&H20,&HF6,&HC9)
1840 MOVE$=M1$+M2$
1850 '
1860 FOR I=0 TO &H184
1870 READ A:POKE &HE100+I,A
1880 NEXT
1890 RETURN
1900 '
1910 'MAIN MACHINE CODE
1920 DATA &HED,&H73,&H29,&HE1,&H31,&H05,&HEB,&H21,&HFF,&HFF,&H22,&H02
1930 DATA &HE0,&H23,&H22,&H04,&HE0,&HAF,&H32,&H01,&HE0,&H21,&H0A,&HE0
1940 DATA &H22,&HE5,&HE1,&H01,&H85,&HE2,&HCD,&H2B,&HE1,&H32,&H01,&HE0
1950 DATA &HED,&H7B,&H29,&HE1,&HC9,&H00,&H00,&HCD,&HF0,&HE1,&H06,&H80
1960 DATA &HCD,&H8C,&HE1,&H20,&H02,&H10,&H03,&H3E,&H01,&HC9,&HFE,&H3A
1970 DATA &H20,&HF2,&HCD,&HE7,&HE1,&HCD,&HFF,&HE1,&H38,&H3D,&HB7,&HC8
1980 DATA &H47,&HCD,&HE7,&HE1,&HCD,&HFF,&HE1,&H38,&H32,&HC5,&H47,&HCD
1990 DATA &HE7,&HE1,&HCD,&HFF,&HE1,&H38,&H27,&H6F,&H60,&HC1,&H22,&H8A
2000 DATA &HE1,&HCD,&HE7,&HE1,&HCD,&HE7,&HE1,&HCD,&HFF,&HE1,&H38,&H17
2010 DATA &HC5,&HED,&H4B,&H8A,&HE1,&HCD,&H57,&HE2,&H38,&H10,&HED,&H79
2020 DATA &H03,&HED,&H43,&H8A,&HE1,&HC1,&H10,&HE4,&H18,&HAC,&HC1,&H3E
2030 DATA &H03,&HC9,&HC1,&H3E,&H02,&HC9,&H00,&H00,&H3A,&HE4,&HE1,&HB7
2040 DATA &H3E,&H1A,&HC0,&HE5,&HD5,&HC5,&H2A,&HE2,&HE1,&H2B,&H22,&HE2
2050 DATA &HE1,&H7D,&HB4,&H28,&H16,&H2A,&HDC,&HE1,&H7E,&HFE,&H1A,&H28
2060 DATA &H09,&H23,&H22,&HDC,&HE1,&H4F,&HAF,&H79,&H18,&H01,&HB7,&HC1
2070 DATA &HD1,&HE1,&HC9,&H21,&H00,&H08,&H22,&HE2,&HE1,&H2A,&HDE,&HE1
2080 DATA &H22,&HDC,&HE1,&HED,&H5B,&HE0,&HE1,&HCD,&HD2,&HE1,&H30,&HD5
2090 DATA &H3E,&H01,&H32,&HE4,&HE1,&HC9,&HC5,&HE5,&HD5,&HCD,&H1E,&HE2
2100 DATA &HD1,&HE1,&HC1,&HC9,&H00,&H01,&H00,&H00,&H00,&H00,&H01,&H00
2110 DATA &H00,&H00,&H00,&HCD,&H8C,&HE1,&H67,&HCD,&H8C,&HE1,&H6F,&HC9
2120 DATA &HED,&H43,&HDE,&HE1,&HAF,&H32,&HE4,&HE1,&H21,&H01,&H00,&H22
2130 DATA &HE2,&HE1,&HC9,&H7C,&HCD,&H0E,&HE2,&H07,&H07,&H07,&H07,&H4F
2140 DATA &H7D,&HCD,&H0E,&HE2,&HB1,&HC9,&HD6,&H30,&HD8,&HFE,&H0A,&H38
2150 DATA &H07,&HD6,&H07,&HFE,&H0A,&HD8,&HFE,&H10,&H3F,&HC9,&H3E,&H07
2160 DATA &H32,&HDD,&HFA,&H3A,&H00,&HE0,&H32,&HDE,&HFA,&H21,&HB4,&HFA
2170 DATA &H85,&H6F,&H36,&H00,&H2A,&HE5,&HE1,&H7E,&HB7,&H37,&HC8,&HC6
2180 DATA &H08,&H6F,&H26,&H00,&H29,&H29,&H29,&HEB,&H21,&HE5,&HE1,&H34
2190 DATA &H2A,&HDE,&HE1,&H01,&H00,&H1D,&HED,&H79,&H3E,&H08,&HCD,&H9D
2200 DATA &H73,&H01,&H00,&H1E,&HED,&H79,&HC9,&HE5,&H2A,&H06,&HE0,&HB7
2210 DATA &HED,&H42,&H3F,&H38,&H22,&H2A,&H08,&HE0,&HB7,&HED,&H42,&H38
2220 DATA &H1A,&H2A,&H02,&HE0,&HB7,&HED,&H42,&H38,&H04,&HED,&H43,&H02
2230 DATA &HE0,&H2A,&H04,&HE0,&HB7,&HED,&H42,&HE1,&HD0,&HED,&H43,&H04
2240 DATA &HE0,&HB7,&HC9,&HE1,&HC9
```

# 第12章

# turbo CP/Mシステム

# 1 turbo CP/M ディスク構成

turbo CP/M のディスクは，標準で 5 インチ両面倍密度（256/バイト/セクタ）のフロッピーディスクを使用しています。

## turbo CP/M のディスク割りあて

ディスクの割りあては次のようになっています。

| 物理的なレコード番号 | 内　容 |
|---|---|
| 0000H | Information Block |
| 0001H～0008H | CCP |
| 0009H～0016H | BDOS |
| 0017H～003FH | BIOS |
| 0040H～0047H | ディレクトリエリア |
| 0048H～04FFH | データエリア |

5 インチ両面倍密度のフロッピーディスクの場合，物理的には 1 ドライブあたり40トラック（1 セクタ＝256/バイト，1 トラック＝32セクタ）のフォーマットになっていますが，CP/M システムで使用する場合，論理的に 1 ドライブあたり40トラック（1 セクタ＝128/バイト，1 トラック＝64セクタ）として扱います。

## 2 システムスクラッチエリアの内容およびメモリ構成

turbo CP/M の TPA（トランジェントプログラムエリア）領域は，メインメモリ上の0100 Hから始まっており，特に0000H～00FFH までの256バイトの領域はシステムスクラッチエリアと呼ばれ，CP/M システムが働くためのワークエリア（ジャンプベクトルのエントリーや各バッファに利用されている）に使用されています。

turbo CP/M システムスクラッチエリア（0000H～00FFH）の内容

| メインメモリのアドレス | 内　　容 |
|---|---|
| 0000H～0002H | CP/M BIOSのウォームブート（WBOOT）へのジャンプ命令 |
| 0003H | IOバイト |
| 0004H | 上位4ビット ／ USER No. ／ 下位4ビット ／ 現在使用中のログインディスク No. |
| 0005H～0007H | CP/M BDOSをコールするためのジャンプ命令（システムコール，エスケープシーケンスにて利用） |
| 0008H～0037H | 未使用（RST 1～6） |
| 0038H～003AH | RST7（DDTコマンドのリスタートとして使用） |
| 003BH～005BH | 未使用 |
| 005CH～007FH | FCB（ファイルコントロールブロック）デフォルト値<br>ただし，007CHにはシーケンシャルファイルのレコードNo.が，007DH～007FHにはランダムアクセスファイルのランダムレコードNo.が格納されます。 |
| 0080H～00FFH | DMA（ダイレクトメモリアドレス）デフォルト値<br>〔ディスクとメモリ間の128 バイトの入出力エリア〕 |

※FCB 及び DMA は，ユーザーが特に指定しない限り，この値に設定されています。

turbo CP/M メモリ構成（turbo CP/M の標準56Kシステムの場合）

| メインメモリのアドレス | 内　　容 |
|---|---|
| 0000H ～ 00FFH | CP/Mシステムワーク（システムスクラッチエリア） |
| 0100H～C3FFH | TPA（トランジェントプログラムエリア） |
| C400H～CBFFH | CCP（Console Command Processor） |
| CC00H～D9FFH | BDOS（Basic Disk Operating System） |
| DA00H～EFFFH | BIOS（Basic Input Output System） |
| F000H～F7FFH<br>F800H～FEFFH<br>FF00H～FFFFH | CP/Mシステム（BIOS）ワークおよび辞書ワーク<br>BIOS ROMワーク<br>IPL ワークあるいはCP/Mの1セクタバッファ |

## 3 周辺機器の割りあて（IOバイトの設定）

CP/Mでは，ディスクドライブのほかに，キーボード，CRTディスプレイ，プリンタなどの各種周辺機器（物理デバイス）を接続することができます。turbo CP/Mでは，これらの周辺機器とCP/Mシステムとの間でデータのやりとりを行なう場合4つの論理デバイスを使用します。

### 論理デバイス

| 論理デバイス | 内　　容 |
|---|---|
| CON：(CONSOLE) | キーボード(入力)，CRTディスプレイ(出力)のようにユーザーがCP/Mシステムと対話するための入出力機器 |
| RDR：(READER) | 紙テープ・リーダやテレタイプなどの入力機器 |
| PUN：(PUNCH) | 紙テープ・パンチャなどの出力機器 |
| LST：(LIST) | プリンタなどリスト類を出力する出力機器 |

これらの論理デバイスには，それぞれについて4種類の周辺機器（物理デバイス）を割りつけることができます。

### 論理デバイスと物理デバイスの割りつけ

| 論理デバイス | 物理デバイス | IOバイトのビットパターン | 物理デバイスの内容 |
|---|---|---|---|
| CON：〔入出力〕（IOバイトのビット0,1に該当） | TTY： | 00 | 内蔵RS-232C（受信，送信） |
| | CRT： | 01 | キーボード(入力)，CRT(出力) |
| | (BAT：) | 10 | 未使用 |
| | (UC1：) | 11 | 未使用 |
| RDR：〔入力〕（IOバイトのビット2,3に該当） | TTY： | 00 | 内蔵RS-232C（受信） |
| | (PTR：) | 01 | 未使用 |
| | (UR1：) | 10 | 未使用 |
| | (UR2：) | 11 | 未使用 |
| PUN：〔出力〕（IOバイトのビット4,5に該当） | TTY： | 00 | 内蔵RS-232C（送信） |
| | (PTP：) | 01 | 未使用 |
| | (UP1：) | 10 | 未使用 |
| | (UP2：) | 11 | 未使用 |
| LST：〔出力〕（IOバイトのビット6,7に該当） | TTY： | 00 | 内蔵RS-232C（送信） |
| | CRT： | 01 | CRTディスプレイ |
| | LPT： | 10 | プリンタ½（セントロニクス社準拠） |
| | (UL1：) | 11 | 未使用 |

- 上記の表で物理デバイスが(　)で囲まれたものは，turbo CP/Mでは使用していません。
- CP/M起動時の設定は，各論理デバイスに対して，CON：＝CRT：，RDR：＝TTY：(受信)，PUN：＝TTY：(送信)，LST：＝LPT：となっていますので，IOバイトの値は，10000001B（81H）となります。

## 4 ディスクパラメータテーブル

CP/M の BIOS 内においては，第10章「CP/M　BIOS コール」で説明しました17個のジャンプベクトルの他に，ディスクパラメータテーブルという各ディスクドライブごとの情報管理と，CP/M の BDOS オペレーションに対するスクラッチパッド領域を与えるためのエリアが設けられています。

このディスクパラメータテーブルは，DPH（ディスクパラメータヘッダ）と DPB（ディスクパラメータブロック）などから構成されています。ただし，スキュー等のように論理セクタと物理セクタの変換が必要な場合は，その変換テーブルが追加されます。

以下に，DPH と DPB についての詳細を説明します。

①DPH（ディスクパラメータヘッダ）……16バイトで構成されています。

| XLT（2バイト） | 0000H（2バイト） | 0000H（2バイト） | 0000H（2バイト） | DIRBUF（2バイト） | DPB（2バイト） | CSV（2バイト） | ALV（2バイト） |
|---|---|---|---|---|---|---|---|
| Ⓐ | Ⓑ | Ⓒ | Ⓓ | Ⓔ | Ⓕ | Ⓖ | Ⓗ |

Ⓐセクタトランスレートテーブル

各ディスクドライブにおける論理セクタと物理セクタのセクタ変換テーブルの格納先頭アドレスを示します。ただし，セクタ変換の必要のないときは，0000H が入ります。また同一ドライブの場合は共用可能です。

Ⓑ，Ⓒ，Ⓓ

CP/M の BDOS 内で使うスクラッチパッド値が入ります。初期値は，それぞれ0000H が入ります。

Ⓔディレクトリバッファ

CP/M の BDOS 内のディレクトリ操作に使用される128バイトのスクラッチパッド領域の先頭アドレスを示します。

Ⓕディスクパラメータブロック

各ディスクドライブのDPB(ディスクパラメータブロック)の格納先頭アドレスを示します。同一ドライブの場合は共用可能です。

Ⓖチェックベクトル

ディスケットを変えるときに，ディレクトリ情報をソフトウエアチェックするためのスクラッチパッド領域の先頭アドレスを示します。ただし，各ドライブによって異なります。

Ⓗアロケーションベクトル

ディスクの使用状況を管理するために，CP/M の BDOS によって使われるスクラッチパッド領域の先頭アドレスを示します。ただし，各ドライブによって異なります。

## ②DPB（ディスクパラメータブロック）

| SPT (2バイト) | BSH (1バイト) | BLM (1バイト) | EXM (1バイト) | DSM (2バイト) | DRM (2バイト) | AL 0 (1バイト) | AL 1 (1バイト) | CKS (2バイト) | OFF (2バイト) |
|---|---|---|---|---|---|---|---|---|---|
| Ⓐ | Ⓑ | Ⓒ | Ⓓ | Ⓔ | Ⓕ | Ⓖ | Ⓗ | Ⓘ | Ⓙ |

Ⓐ Sectors per track

1トラックあたりの全論理セクタ数（1セクタ＝128バイト）を示します。

Ⓑ Block shift factor

データブロックのアロケーションサイズによって決定される値を示します。

BLS＝128×$2^n$

※ n＝BSH（Block shift factor）

BLS＝データブロックアロケーションサイズ（CP/Mが管理するディスク上のファイルの最小単位）

Ⓒ Block mask

データブロックのアロケーションサイズによって決定される値を示します。

BLM＝（データブロックのセクタ数）－1

Ⓓ Extent mask

データブロックのアロケーションサイズとディスクのブロック数によって決められる値を示します。

Ⓔ Disk size max

DSM＝（ディスクのデータブロックの最大数）－1

BLSとBSH，BLM，EXMとの対応表

| BLS (データブロックアロケーションサイズ) | BSH (ブロックシフトファクター) | BLM (ブロックマスク) | EXM(エクステントマスク) | |
|---|---|---|---|---|
| | | | DSM (255以下) | DSM (256以上) |
| 1,024 | 3 | 7 | 0 | — |
| 2,048 | 4 | 15 | 1 | 0 |
| 4,096 | 5 | 31 | 3 | 1 |
| 8,192 | 6 | 63 | 7 | 3 |
| 16,384 | 7 | 127 | 15 | 7 |

ⒻDirectory entries max

DRM＝（ディスクに収容可能なディレクトリエントリーの数）－1

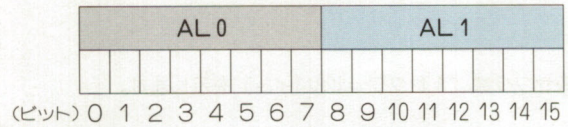
※各ファイルは，そのファイルに関する情報をもったFCB（ファイルコントロールブロック）をメインメモリ上にもっており，通常これは，そのファイルが存在するディスク上のディレクトリエリアに格納されています。すなわち，ディスク上のディレクトリとはFCBの一部を意味し，ファイルごとに32バイト単位で構成されています。（ただし，実際にメインメモリ上に存在しているFCBは，36バイト単位で構成されています。）

Ⓖ ALLOC 0

Ⓗ ALLOC 1

AL 0とAL 1の各ビットは，ディレクトリエントリーに使用するデータブロックを示します。

AL 0とAL 1の構成




BLSとディレクトリエントリーの総数

| BLS | ディレクトリエントリーの総数 |
|---|---|
| 1,024 | (AL 0とAL 1のビット〝1〟の数) × 32 |
| 2,048 | ( 〃 ) × 64 |
| 4,096 | ( 〃 ) ×128 |
| 8,192 | ( 〃 ) ×256 |
| 16,384 | ( 〃 ) ×512 |

Ⓘ Check vector size

CKS=(DRM+1)／4

※ただし，これは，フロッピーディスクドライブのように，ディスクがメディア交換可能な場合の算出式で，ハードディスクのようにディスクが固定式の場合は，0000Hにセットされます。

Ⓙ Track offset

システムトラックをスキップするための値を示します。

X1 turboの標準5〝両面倍密度フロッピーディスクのDPB（ディスクパラメータブロック）を次の表に示します。

X 1 turboの標準5〝両面倍密度フロッピーディスクのDPB

| DPB（ディスクパラメータブロック）の項目 | X 1 turbo 標準5〝両面倍密度フロッピーディスクのDPB |
|---|---|
| ⒶSPT | 0040H（1トラック64セクタ） |
| ⒷBSH | 04H〔2048(バイト)=128(バイト)×$2^n$ゆえにn＝4〕 |
| ⒸBLM | 0FH〔2048(バイト)=16(セクタ)ゆえに16－1＝15〕 |
| ⒹEXM | 01H |
| ⒺDSM | 0097H〔{(128(バイト)×64(セクタ))×(40(トラック)－(2(トラック))/2048}－1・＝151〕 |
| ⒻDRM | 003FH〔64－1＝63〕 |
| ⒼAL 0 | 80H |
| ⒽAL 1 | 00H |
| ⒾCKS | 0010H〔(63＋1)/4＝16〕 |
| ⒿOFF | 0002H |
| BLS | 2048(バイト) |

## 5 FCB（ファイルコントロールブロック）

FCB（File Control Block）とは，CP/Mシステムがディスク上のファイルを処理するのに必要な情報が入っているブロックのことをいいます。FCBは，通常ディスクのディレクトリエリアに格納されており，一つのディレクトリに対し一つのFCBが対応しています。しかし，実際はディレクトリが32バイト単位で扱われるのに対し，FCBは36バイト単位で構成されています。
CP/Mシステムがファイル処理を行なうときの過程としては，まず，ディスクのディレクトリをFCBとしてメインメモリに取りこみ，さらに，ファイルの処理を行なうたびにFCBを更新していきます。そして，ファイルの処理後は，メインメモリのFCBを再びディスクのディレクトリエリアに戻します。

以下に，このFCBの詳細について説明します。

FCB（ファイルコントロールブロック）……36バイトで構成され，通常メインメモリ上の005CH～007FHに格納されます。

| | dr | f1 | f2 | f3 | f4 | f5 | f6 | f7 | f8 | t1 | t2 | t3 | ex | s1 | s2 | rc | d0 | d1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| （バイト） | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| | Ⓐ | | | | Ⓑ | | | | | | Ⓒ | | Ⓓ | Ⓔ | | Ⓕ | | |

| | d2 | d3 | d4 | d5 | d6 | d7 | d8 | d9 | d10 | d11 | d12 | d13 | d14 | d15 | cr | r0 | r1 | r2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| （バイト） | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| | | | Ⓖ | | | | | | | | | | | | Ⓗ | Ⓘ | | Ⓙ |

Ⓐディスクドライブコード

00H＝ログインディスクドライブ
01H＝ドライブA：
02H＝ドライブB：
03H＝ドライブC：
04H＝ドライブD：
05H＝ドライブE：
06H＝ドライブF：
07H＝ドライブG：
08H＝ドライブH：

Ⓑファイル名（8文字）

ファイル名を格納します。各バイトともビット7が0のため，文字としてはアルファベット（A～Z）と数字（0～9）のみ使用できます。

Ⓒファイルエクステンション名（ファイルタイプ名）（3文字）

COM，ASM，HEX，PRN，BAKなどのファイルタイプを格納します。使用できる文字は，Ⓑのファイル名で扱える文字と同様です。特に，t1～t3のうち，t1，t2のビット7は次のような意味をもちます。

t1，t2のビット7に対応するファイル属性

| t1とt2のビット7の状態 | ファイル属性 | 意　味 |
|---|---|---|
| t1（ビット7＝0） | $ R/W | 自由にファイルの削除や一部の変更が可能 |
| t1（ビット7＝1） | $ R/O | ファイルの削除や一部の変更が不可 |
| t2（ビット7＝0） | $ DIR | DIRコマンドにより表示される通常のファイル |
| t2（ビット7＝1） | $ SYS | DIRコマンドによる表示にはあらわれないし，またPIPコマンドによる転送も不可。 |

Ⓓロジカルエクステント番号（00H～3FH）

現FCBのロジカルエクステント番号（ファイルが複数個のディレクトリをもつ場合，おのおののディレクトリにつける番号）を示します。初期値は，00Hが入ります。

Ⓔ CP/Mシステムが使用します。

Ⓕロジカルエクステントの総レコード数（00H～80H）

現ロジカルエクステントの総レコード数（128/バイト単位のデータの集まり）を示します。

Ⓖディスクアロケーションマップ

CP/Mシステムが使用するもので，d0からd15までのそれぞれの値が，ディスク上の1データブロックの位置を示し，d0～d15で16個のデータブロックを表わせます。

Ⓗシーケンシャルファイルのリード／ライト処理が行なわれている場合，その時点でのレコード番号を示します。

①ランダムアクセスファイルを使用する場合のランダムレコード番号を示し，0000H～FFFFH（0～65535）の範囲の値をとり，r0が下位の桁，r1が上位の桁になります。

Ⓙ通常は，00Hが入りますが，ランダムレコード番号がオーバーフローすると，00H以外の値が入ります。

※ただし，このFCBは，メインメモリ上では36/バイトの構成になっていますが，ディスク上のディレクトリエリアで使用されるときは，上述のⒽ，①，Ⓙを除く32/バイトのディレクトリとして扱われます。

※turbo CP/Mでは，1レコード＝128/バイト，1データブロック＝16レコード＝2048/バイト，1ロジカルエクステント＝16データブロック＝32768/バイトとして扱われます。

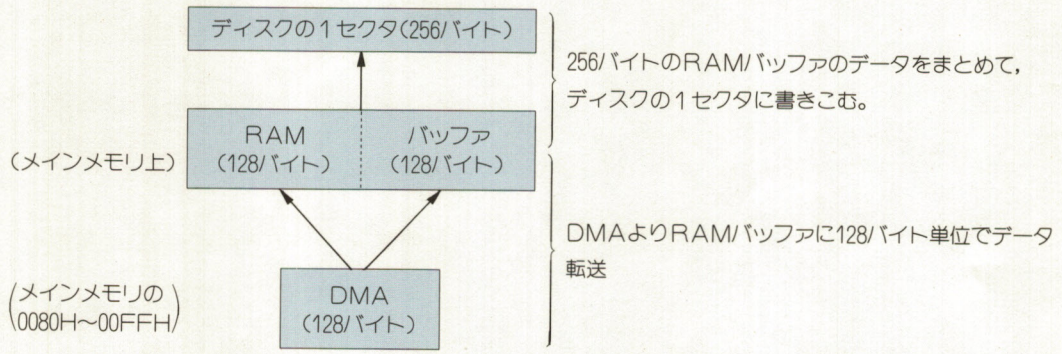
## 6 DMA(ダイレクトメモリアドレス)について

DMA（Direct Memory Address）とは，ディスクへのリード/ライト操作を行なう時に必要な128/バイトのメモリバッファ（通常は，メインメモリ上の0080H～00FFHに配置されています）のことをいいます。CP/MシステムのBDOSはこのDMAバッファを通してディスクへのリード/ライト操作を行ないます。

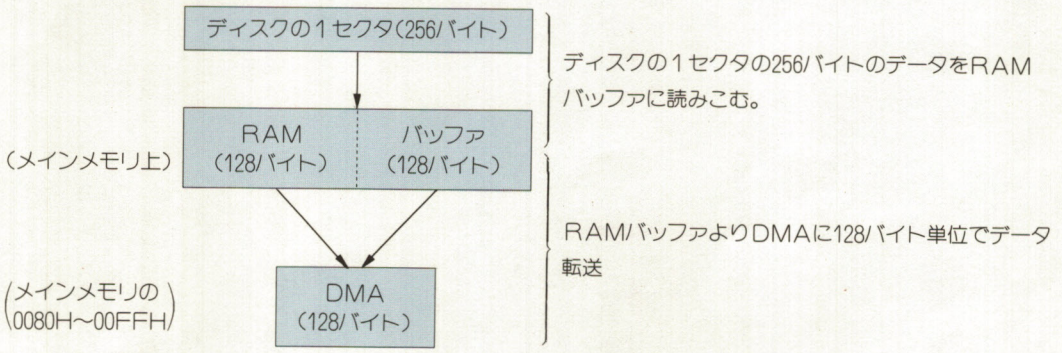
CP/MシステムのBDOSは，論理的に1セクタを128/バイトとして扱っていますが，X1 turboの5インチ2Dタイプのディスクフォーマットとしては，1セクタ=256/バイトになっています。すなわち，ディスクへのリード/ライト操作を行なう時に，DMAとディスクの1セクタが，直接データのやりとりを行なうのは，セクタサイズの違いから不可能であり，メインメモリ上にディスクの1セクタとDMAとの仲介をするためのRAM/バッファを設ける必要が生じてきます。CP/Mシステムは，このRAM/バッファをDMAとディスクとの間に配置し，ブロッキングとデブロッキングという方法を使って，ディスクのリード/ライト操作を行ないます。

以下で，ブロッキングとデブロッキングの方法について説明します。

**❶ブロッキング**（DMAからRAM/バッファを経由してディスクの1セクタに書きこむ場合）




**❷デブロッキング**（ディスクの1セクタからRAM/バッファを経由してDMAに読みこむ場合）



# 第13章

# ワードマスター(WORD MASTER)

## 1 はじめに

ワードマスターは，テキスト（アスキーファイル）の編集，作成に大きな力を発揮します。本CP/MのEDコマンドがラインごとの編集機能しか持たないのに対して，ワードマスターはBASICの編集機能と同様にカーソルを移動してスクリーン編集を行なうことができます。

ワードマスターは，ビデオモードとコマンドモードという2種類の動作モードを備えています。ビデオモードでは，カーソルを4方向に移動させ，ファイルを作成，編集することができます。一方，コマンドモードでは，コマンドを入力することにより，検索・置き換え・ループ・テキストの移動などを行なうことができます。

ワードマスターの起動時は常にビデオモードで始まり，ビデオモードからコマンドモードへは[ESC]キーを押すことにより，またコマンドモードからビデオモードへはV↵と入力することにより，移ります。

この2つのモードを使用することによって，文書の編集，作成を容易にかつ能率的に行なうことができます。

さらに，turbo CP/Mは，すぐれた日本語処理機能をもっていますので，ワードマスターでも簡単に漢字を入力することができます。入力方法はBASICと同様ですので，詳しくは「BASICマニュアル」及び第3章の2項を参照してください。

なお，ワードマスターを使用するためにはログインディスクに次のファイルが必要です。

WM. COM
WM. HLP
WMENGL. HLP
WMKANA. HLP } ヘルプメニュー変更時のみ必要
WMKANJI. HLP

## 2 ご使用になる前に

ワードマスターのヘルプメニューファイルには，英文・カナ文字・漢字の 3 種類のファイルがあります。ヘルプメニューファイルには，ビデオモード・コマンドモードの説明が入っていますが，ワードマスターを使用する前に，3 種類のメニューファイルの中からどれか 1 つを選ぶ必要があります。そのためには，使用したいヘルプメニューファイルを WM・HLP というファイル名に変更します。出荷時には漢字ヘルプメニューファイルが WM・HLP となっており，変更したい場合には PIP コマンドを使用します。
例えば，英文メニューファイルを使用したい場合には，次のように入力します。

```
A>PIP␣WM. HLP=WMENGL. HLP ↵
```

以下にヘルプメニューファイルを示します。

| | |
|---|---|
| WM. HLP | ……ワードマスターのヘルプメニューファイル |
| WMENGL. HLP | ……ワードマスターの英文ヘルプメニューファイル |
| WMKANA. HLP | ……ワードマスターのカナ文字ヘルプメニューファイル |
| WMKANJI. HLP | ……ワードマスターの漢字ヘルプメニューファイル |

ヘルプメニューの呼び出し方法については，本章 6 項を参照してください。

## 3 ワードマスターの起動

```
A>WM␣d1:filename.ext␣d2: ↵
```

と，入力するとワードマスターが起動します。
ここで，d1にはオリジナルファイル（編集するファイル）のあるディスクドライブを指定して，省略された場合にはログインディスクになります。
d2には，編集されたファイルをセーブするためのドライブを指定し，省略するとd1と同じドライブになります。
起動後，ワードマスターは入力された“filename.ext”と同じファイルをd1で指定したドライブへさがしに行きます。そのファイルが見つからないと，ワードマスターはNEW FILEをつくります。
オリジナルファイルがあった場合（つまりNEW FILEでない場合）は，Eコマンド（編集終了コマンド）で編集を終えたとき，新たに編集したテキストはここで入力した“filename.ext”となり，もとのオリジナルファイルは“filename.BAK”となります。
編集するファイルがNEW FILEの場合は，編集が終了しても“BAK”ファイルはありません。
なお，ワードマスター起動時はビデオモードになっています。

注1）ファイル名には日本語は使用できません。
注2）編集したファイルをセーブするディスクにはプロテクトシールを貼らないでください。

それでは例文を作成しながら学んでいきましょう。画面を見ながら，まず次のように入力します。

```
A>WM␣EXAMPLE.ASM ↵
```

これは，ワードマスターを呼び出し，“EXAMPLE.ASM”というファイルを編集することを宣言しています。ワードマスターが起動すると，オープニングメッセージが表示され次にこのファイルがNEW FILEであることを知らせてきます。次に画面はクリアされカーソルが画面左上に移動し，テキスト（文）が入力できる状態になります。（ワードマスターを起動してすぐ編集に入りたいときは，画面にメッセージが表示されたときに何かキーを押してください。ただちに画面が変わり，編集を行なうことができます。）

BASICの日本語処理と同様に，SHIFT ＋ XFER で日本語入力モードに入り，次の例文を入力してください。

```
ワードマスターには，ビデオモードとコマンドモードという
２つの動作モードがあります。この２つのモードを使用する
ことにより，文（アスキーファイル）の作成，編集を容易に
行なうことができます。
```

入力が終了したら SHIFT ＋ XFER で日本語入力モードをぬけてください。
これからは，この例文を使用して編集の練習を行ないますのでまずこの例文をいつでもよびだせるようにディスクにセーブしておきましょう。
セーブするには ESC を入力してから H を入力します。ESC を入力すると画面の左下に＊が表示されます。続いて H を入力したら ↲ を押します。

以上のような処理を以後

```
$H ↲
```

と表記します。〝$〟は ESC キーの意味です。

入力すると，2，3秒後に画面がクリアされ，先ほどの文章が再び現われます。
今の作業は，例文をメモリに入れた後，$でコマンドモードに変え，H コマンドでメモリ内の例文をディスクにセーブしたものです。現在，このディスクの中には先ほどの例文が〝EXAMPLE. ASM〟というファイル名でセーブされています。

ここで，キー入力指示の表記※の見方について説明しておきます。
たとえば，ビデオモードの説明で

```
^D ^D →^S ←^X ^X ↓^E ↑
```

となっている場合，
〝^〟は CTRL キーを意味していますので CTRL キーを押しながら D を２回入力し次に〝→〟はカーソルキーの → を意味していますので → を１回入力します。以後同様に行ないます。（〝←〟は ← キー，〝↓〟は ↓ キー，〝↑〟は ↑ キーをそれぞれ表わしています。）ここでは [ ] の部分をその通りに入力していくことによってカーソル移動の方法がわかるようになっています。このように，画面でのカーソルの動き等を確認しながら指示通りに入力を行なってください。

※キー入力指示の表記には，日本語変換フィールド内でのキー操作は記述していません。

## 4 ワードマスターのビデオモード

ビデオモードでは，カーソルを移動してテキスト（アスキーファイル）を作成，編集することができます。ワードマスターが起動した直後は，このビデオモードになっています。
まず，先ほどの例文を画面に出してください。出ていない場合は

```
$O↲Y↲
```

と入力して，ディスクからファイルを呼びだします。

### ①カーソルの移動

#### A．4方向のカーソル移動

```
^D ^D → ^S ← ^X ^X ↓ ^E ↑
```

と入力してください。カーソルが4方向に移動しました。現在カーソルは2行目の〝つ〟の位置にあります。
→（または^D）は，カーソルを1桁右へ移動させる働きがあります。
←（または^S）は，カーソルを1桁左へ移動させる働きがあります。
↑（または^E）は，カーソルを1行上へ移動させる働きがあります。
↓（または^X）は，カーソルを1行下へ移動させる働きがあります

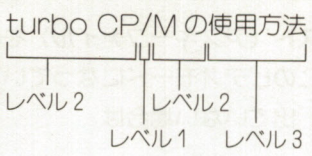
#### B．ワードタブ

```
^F ^F ^F ^A ^A
```

と入力してください。現在カーソルは2行目の〝動〟の位置にあります。^Fを押すと，カーソルは次のワードの先頭へ飛びます。また^Aを押すと，カーソルは逆方向のワードの先頭へ飛びます。
^Fは，ワード単位にカーソルを右に移動させ，^Aは，ワード単位にカーソルを左に移動させます。

以下にワードの定義についての説明をします。
ワードマスターでは扱う文字はすべて4段階のレベルにわけています。テキスト中でレベルが同じ文字かあるいは低い文字が続いている場合は1つのワードの続きとみなし，前の文字より高いレベルの文字があらわれたときにこれを新しいワードの始まりとみなします。
各文字のレベルは次の通りです。
レベル0……リターンコード，スペース，制御文字（^N，^Lなど）
レベル1……特殊記号（＋，－，？など）
レベル2……英数字，カナ（全角，半角）
レベル3……漢字

例をあげてみましょう。




「turbo CP/M の使用方法」の各文字のレベルは左のようになります。
したがってワードは"turbo CP/"，"Mの" "使用方法" の 3 種になります。

### C．行タブ

```
^B  ^B  ^B
```

と入力してください。カーソルは同じ行の左端と右端の間を移動します。現在カーソルは 2 行目の先頭にきています。
^Bは行タブです。最初の入力でカーソルは行の左端へ動き，次の入力でカーソルは行の右端へ動きます。

### D．スクリーンタブ

```
^V  ^V  ^V
```

と入力してください。最初に^Vを入力するとカーソルは表示ファイルの左上端へ移動し，2 度目の入力では表示ファイルの最終行の最初の文字へ移動します。現在カーソルはファイルの先頭へきています。
^Vはスクリーンタブです。カーソルを画面のファイルの左上端と左下端へ動かす働きがあります。

### E．カーソルが移動する場所

それでは，カーソルを 3 行目の行の終端 "に" に移動させて，次のように入力してください。

```
↓
```

カーソルはすぐ下の位置ではなく，次の行の終端に移動します。カーソルが移動することができるのは，行の最後の文字の右（リターンコードの位置）までです。以下に，カーソルが移動しない場所についてまとめます。

1．行の最後の文字の右（リターンコード）以降
2．コントロール文字（^）記号の位置
3．タブでスキップした位置

## A．文字の挿入

```
^V  ^O  文字を挿入します。^O
```

と入力してください。 最初の^Oで画面下の INSERT ON の文字が INSERT OFF に変わり，カーソルの前の文字を挿入できる状態になります。日本語処理モードにして〝文字を挿入します。〟と入力してください。さらに2回目の^Oで再び INSERT OFF に変わり，もとの状態にもどります。
^Oは挿入モードを ON/OFF にする働きをもっています。

## B．行の挿入

```
^V  ^N  ^N  ワードマスター ↲  シャープ  ↑
```

と入力してください。最初の^N ^Nで2つの空白行が挿入され，その挿入した1行目と2行目に〝ワードマスター〟と〝シャープ〟がそれぞれ書かれます。カーソルは〝ワードマスター〟の〝ス〟の位置にあります。

```
^N
```

と入力してください。〝ワードマスター〟の〝マ〟の右側の文字が全て1行下の左端へ移動し，それ以下の行もそれぞれ1行ずつ下にさがりました。また，

```
^V  ^O  ↲  ^O
```

と入力すると，画面の一番上に1行挿入されます。挿入モードが ON のとき ↲ を入力すると^Nと似た働きをします。ただし，^Nのときにはカーソルの位置はかわりませんが，挿入モードが ON のときに ↲ を入力するとカーソルはその行の最初の文字へ移動します。

## C．文字の削除

```
↲  ↲  ↲     ^G  →  DEL
```

と入力すると，4行目の〝文字を挿入します〟の〝文字〟の部分を消すことができます。削除した文字の右側の文字はすべて左へずれます。カーソルは〝を〟の位置にあります。
^Gは，カーソル位置にある文字を削除し，
DEL は，カーソルの左側の文字を削除します。

```
↑  ↑  ↑  ^B  ^B  ^G
```

と入力してください。〝ワードマ〟と〝スター〟の2つの行が1つの行につながりました。これは，〝ワードマ〟の後のリターンコードを削除したためです。

### D．ワードの削除

```
↲ ↲ → ^T ^L
```

と入力してください。カーソルは4行目の〝挿〟に移動し，^Tを入力すると〝挿入します〟が削除され，次に^Lを入力すると〝を〟の文字が削除されました。

^Tは，カーソル以下右のワードを削除し，

^Lは，カーソル以下左のワードを削除します。

ただし，ワードの中間にカーソルがある場合には，そのワードの右側（^T）あるいは左側（^L）のみが削除されます。またカーソルと削除しようとするワードとの間にリターンコードがある場合には，リターンコードも削除されその行と次の行がつながります。

### E．行の削除

```
^V ^Y
```

と入力してください。1番上の空白行がなくなり，以下の行がそれぞれ1行ずつ上にあがりました。

^Yは，カーソルの位置する行（リターンコードを含む）をすべて削除します。続いて，

```
→ → → ^U
```

と入力してください。〝ワード〟の文字，つまりカーソルの左側がすべて消えました。

```
^K
```

と入力すると，カーソルの右の文字（カーソルを含む）がすべて消え，その行には何も残っていません。

^Uは，カーソルの左側の行を削除します。

^Kは，カーソルの右側の行（カーソルを含む）を削除します。

```
^Y ^Y
```

と入力してください。最初に入力した通りの画面にもどりました。

(注)ファイル1行の長さは，画面の幅より長くてもかまいません。画面の幅よりも長い場合には，複数行にわたって表示します。その際，〝>>〟が継続している行の左端に表示され，この行が継続行であることを示します。このとき，画面の左端の桁は使用されていません。

### ③繰り返しコマンド

```
^V  ^@  →
```

と入力してください。まずカーソルは画面の左上端へ移動し，そして右に4文字動きました。^@を使用すると，その後の→が4回繰り返し実行されます。さらに

```
↵  ^@  ^@  →
```

と入力すると，今度はカーソルが16文字分右へ移動しました。つまり，^@　^@　→　で→を16回繰り返したことがわかります。

^@は，次の文字入力を4回繰り返す働きがあります。ただし，^@を続けて入力できるのは3回までです。

^@　……　次の文字入力を4回繰り返します。
^@　^@　……　次の文字入力を4×4＝16回繰り返します。
^@　^@　^@　……　次の文字を4×4×4＝64回繰り返します。

### ④画面のスクロール

```
^V  ^V  ^@  ^@  ↵  ^@  ↵  2ページ ↵
^@  ^@  ↵  ^@  ↵  ↵  ↵  3ページ
```

と入力してください。1画面には23行分のテキストを表示することができます。そして2ページ目，3ページ目の最初の行にページ番号を示す文字を入力しました。

```
^R  ^R
```

と入力すると，テキストが2ページ分ファイルの初め方向に動きました。さらに

```
^C  ^C
```

と入力すると，テキストが2ページ分ファイルの終り方向に動きました。

```
^R  ^R  ^Z  ^Z  ^Z  ^Z
```

と入力すると，^Zを入力するたびにファイルが1行上に移動します。

```
^@  ^W
```

と入力すると，ファイルは4行下に移動し，もとの状態にもどりました。

^Rは，ファイルを画面上で1ページもどします。
^Cは，ファイルを画面上で1ページ次へ動かします。
^Wは，ファイルを画面上で1行下へ動かします。
^Zは，ファイルを画面上で1行上へ動かします。

### ⑤コマンドモードへの移行

ESC キーを押してください。画面上左下端に＊が表示されます。現在の動作モードは，コマンドモードで，コマンドの受け付け状態になっています。

## 5 ワードマスターのコマンドモード

コマンドモードでは，コマンドを入力し文字列の検索，置き換え，ループ，ファイルへの入出力を行なうことができます。ワードマスターのコマンドモードにおけるコマンドは，エディタ（ED）コマンドと非常に似ています。

なお，コマンド名は，大文字，小文字どちらでも実行できます。

さて，現在コマンドモードになっているか確認してください。コマンドモードのときは，画面の左下端に〝＊〟が表示されています。表示されていない場合は，ESC キーを押してください。

### ①文字コマンドおよび行コマンド

```
B ↲
```

と入力してください。画面上では見えませんが，文字ポインタが*）ファイルの先頭へ移動します。ビデオモードで確認してみましょう。

＊文字ポインタとは，編集している文字の位置を示すもので，ビデオモードの場合カーソルとカーソルの前の文字との間に文字ポインタがあります。

```
V ↲
```

ビデオモードにもどり，カーソルがファイルの先頭にあることが確認できます。

```
$−BV ↲
```

と入力すると，今度は文字ポインタがファイルの終り（3ページの後）に移動し，ビデオモードになります。＋−Bは，文字ポインタをファイルの先頭（−の場合はファイルの終り）に動かします。では，次のように入力してください。

```
$B6CV ↲
```

文字ポインタが6文字分（半角）動きました。

＋−nCは，文字ポインタをn文字分（半角）動かすコマンドです。＋はファイルの終り方向を，−はファイルの先頭方向を表わしています（省略した場合は＋）。

```
$2LV ↲
```

と入力してください。文字ポインタが2行動きました。

＋−nLは，文字ポインタをn行動かすコマンドです。ポインタは行の先頭へ移動します。

```
$B6C2T↵
```

と入力すると，画面下側に〝マスターには……を使用する〟という2行が表示されます。これは，6Cで文字ポインタを〝マ〟の位置に移動させ，2Tでポインタ以降の2行分を表示させたものです。

＋－nT は，文字ポインタ以降のn行を表示させる働きがあります。

```
B2↵
```

と入力すると，3行目のテキストが表示されます。

＋－n ↵は，文字ポインタをnへ動かし，その1行を表示させる働きがあります。

```
B14DV↵
```

と入力してください。〝ワードマスター〟の文字が削除されました。

＋－nD は，n文字分（半角）を削除する働きがあります。

```
$2KV↵
```

と入力すると，上の2行が削除されました。

＋－nk は，n行を削除する働きがあります。

ここで，新しくファイルを作ります。一旦，ワードマスターを終了し，再びワードマスターを起動しましょう。

```
$Q↵Y↵
```

Qは〝Quit〟の略です。ワードマスターが終了しました。Qを入力した場合，それまで編集されたテキストはセーブされません。Qを入力すると，〝Abort（Y/N）?〟と確認してきます。（詳しくは，⑥ファイルの入出力の項を参照してください。）

```
A>WM␣EXAMPLE2.ASM↵
```

と入力してください。ワードマスターが再起動され，〝EXAMPLE 2〟というファイルが新しくつくられました。現在はビデオモードです。以下のようにテキストを打ち込んでください。

```
The little girl takes a walk in the park every morning.
The boy with a dog is her brother.
```

## ②スクラッチパッドコマンド

```
$BQPV ↲
```

と入力してください。ビデオモードから最初の1行が消えました。これは QP コマンドで，最初の1行（文字ポインタのある位置から1行）をファイルから削除し，スクラッチパッドに格納したためです。スクラッチパッドとは，ファイルを一時的に格納するメモリのようなものです。

nQP は，文字ポインタよりファイルの終り方向にn行のテキストをスクラッチパッドメモリに入れ，テキストを削除するコマンドです。それでは次のように入力してください。

```
$QT ↲
```

先ほどスクラッチパッドに入れた1行が表示されました。QT は，スクラッチパッドの内容を表示する働きがあります。

```
B5QGV ↲
```

スクラッチパッドの内容が，5回ファイルにコピーされています。

これは，文字ポインタをファイルの先頭へ移し，5 QG というコマンドでスクラッチパッドの内容をポインタの位置に5回コピーし，最後にビデオモードにもどしたものです。

nQG は，スクラッチパッドの内容をn回文字ポインタの位置に挿入します。

```
$/QP
```

と入力してください。

n/QP は，ファイルからn行削除し，スクラッチパッドに追加するコマンドです。

```
QT ↲
```

と入力して内容を確認してみますと，"The boy……her brother." の文が，最後に追加されているはずです。さて，次のように入力してください。

```
QKQT ↲
```

画面には＊しか表示されていません。

QK は，スクラッチパッドメモリの内容を消去します。現在，スクラッチパッドの中には何も入っていません。それでは次のように入力してください。

```
QL She has blue eyes and brown hair. ^N$ ↲
```

QL テキスト$はテキストをスクラッチパッドの中に入れ，現在のスクラッチパッドの内容と書きかえる働きがあります。テキストの終りの^Nは，このテキストの最後のリターンコードを追加するためのものです。

```
QT ↲
```

と入力してください。今入力したテキストが表示されるはずです。さらに，

```
4／QGV ↲
```

と入力してみてください。今のテキストが4回ファイルにコピーされているはずです。

### ③テキストの検索

```
∧V $Fblue $V ↲
```

と入力してください。文字ポインタは，最初の〝blue〟のすぐ後にあります。
＋－nF 文字列$は，文字ポインタ以降のファイルからn番目の文字列を捜し，見つけた文字列の直前に文字ポインタをおきます。

```
$ 3 Feyes $V ↲
```

と入力すると，先ほどのポインタ位置から数えて3番目の〝eyes〟の後に文字ポインタがあるはずです。それでは次のように入力してみて下さい。

```
$F green $ ↲
```

〝# # Fgreen$〟というメッセージが出てきます。これは〝green〟という文字列が見つからなかったために，検索の失敗を知らせるメッセージが出てきたのです。

検索コマンドには，Fコマンドのほかにnコマンドがあります。Fコマンドは「短い検索」，Nコマンドは「長い検索」を行ないます。短い検索とは，文字ポインタから2000文字以内あるいは，RAN内で検索をします。長い検索とは，文字ポインタからファイルの終端あるいは，先頭までの検索をします。
FとNは非常に似た機能をもっていますが，文字列が見つからなかった場合，Fでは文字ポインタの位置が変化しません。一方，Nでは文字ポインタがファイルの終りか初めに移動します。

### ④テキストの検索と置き換え

ワードマスターでは，文字列を捜し出すだけでなくそれを文字列に置き換えることもできます。次のように入力してみてください。

```
B ↲
S brown $black $V ↲
```

ファイルの最初の〝brown〟が〝black〟に置き換えられています。文字ポインタは，その後にあります。

＋－nS文字列$新しい文字列$は，現在の文字ポインタ位置より文字列を検索し，新しい文字列で置き換えることをｎ回繰り返します。文字ポインタは，ｎが正数の場合置き換えた文字列の直後に移動し，ｎが負数の場合置き換えた文字列の直前に移動します。
また，検索コマンドと同様，検索置換コマンドも短い検索置換と長い検索置換があります。Sは短い検索置換を，Rは長い検索置換を行ないます。
さらに，検索置換コマンドで，検索，削除することもできます。

```
$－S blue $$V ↵
```

ファイルの最初の〝blue〟が削除されています。検索置換コマンドにおいて，新しい文字列の部分を省略すると文字列が文字ポインタよりｎ個削除されます。

※検索（置換）コマンド（F，N，S，R）の中の文字列（S，Rでは最初の文字列）内の特殊文字の意味は，以下の通りです。

^A …… 任意の文字
^S …… セパレータ（文字や数字以外の任意の文字）
^O x…… x以外の文字

### ⑤ループ

```
$ <Sblue$green$> ↵
BV ↵
```

と入力してください。ファイル中の〝blue〟の文字列がすべて〝green〟にかわりました。〝<〟と〝>〟の間に検索置換コマンドがあります。
ｎ<コマンド列>で，〝<>〟内のコマンド列をｎ回繰り返します。ここでは注意したいのは今までｎが省略された場合ｎ＝１でしたが，ループコマンドの場合にはｎ＝65,535になります。実際には，ｎを省略した場合にはループ内のコマンド列はファイルの終りまで実行されると考えることができます。では次のように入力してください。

```
$<Slittle$pretty$V> ↵
```

〝little〟を〝pretty〟に置き換えると画面はビデオモードにかわります。文字ポインタは〝pretty〟のすぐ後にあるはずです。ESCキーを入力すると，入力するごとに，続きが実行されます。すべての〝little〟を〝pretty〟に置き換えると〝＃＃Slittle$pretty$V>〟というメッセージが出て，もう検索できないことを知らせてきます。

### ⑥ファイルの入出力

```
BYEXAMPLE.ASM$ ↲
V ↲
```

と入力してください。ファイルの先頭に〝EXAMPLE〟のファイルがすべて挿入されています。Yコマンドによって，指定したファイルすべてを編集しているファイルの文字ポインタ位置に挿入します。文字ポインタは，挿入したファイルのすぐ後に移動します。

```
$ 8 WTEST$ ↲
```

と入力してください。これは，今文字ポインタがあった位置から8行を新しいTESTというファイルに書き込んだものです。それでは，ワードマスターの編集操作を中止し，TESTというファイルを呼び出してみましょう。

(注)Y，Wコマンドを使用するとき，ファイル名のあとにピリオド（.）とエクステンションを入力しない場合にはファイル名の最後には自動的に〝.LIB〟が付きます。

```
Q ↲ Y ↲
WM␣TEST. LIB ↲
```

TESTというファイルに先ほどの8行が書き込まれているはずです。
nWコマンドは，文字ポインタからn行を指定されたファイルに書き込みます。文字ポインタは，書き込みされたテキストの最後の文字の次にきます。
W，Yコマンドを使用することよって，ファイルの分割，併合をすることができます。

### ⑦ワードマスターの編集終了と更新

```
$E
```

と入力してください。画面は，CP/Mにもどっています。テキストはTESTというファイルネームでディスクに書き込まれました。

ワードマスターの編集終了コマンドには以下のようなものがあります。
E……ディスクに書き込み，CP/Mへもどる
Q……ディスクに書き込まず編集を終了し，CP/Mへもどる
H……ディスクに書き込み，編集を再開
O……編集したテキストを無効にし，編集を継続

以上の編集コマンドは，単独で使用してください。他のコマンドと組み合わせて，例えばHVのように使用することはできません。

## 6 ワードマスターのヘルプメニュー

メニューファイル　WM. HLP は，＾Ｊを押すことによっていつでも画面に呼び出すことができます。メニューファイルの中には，ビデオモード，コマンドモードで使用できるコマンドの説明が入っています。

ビデオモードでは，＾Ｊを入力するといつでもメニューファイルを見ることができます。＾Ｊを押すたびに画面がかわり，合計で５回＾Ｊを入力すると編集画面にもどります。コマンドモードのときは，＾Ｑ ↵を入力した後，＾Ｊを押します。途中でもとのモードに戻るときは，↵を押してください。

# 7 ワードマスターのまとめ

## 1．ビデオ編集のまとめ

| 制御コード | 16進コード | 機　　能 |
|---|---|---|
| ^O | 0FHI | 挿入モードを ON/OFF にします。 |
| ^Sまたは← | 13H,1DH | カーソルを１文字左へ動かします。カーソルが左端にあるときは前の行の右端に動かします。 |
| ^Dまたは→ | 04H,1CH | カーソルを１文字右へ動かします。カーソルが行の右端にあるときは次の行の左端に動かします。 |
| ^Xまたは↓ | 18H,1FH | カーソルを１行下へ動かします。 |
| ^Eまたは↑ | 05H，1EH | カーソルを１行上へ動かします。 |
| ^A | 01H | カーソルを次のワードの先頭に動かします。 |
| ^F | 06H | カーソルを１つ前のワードの先頭に動かします。 |
| ^Q | 11H | カーソルを次のタブストップへ動かします。タブストップは８桁ごとにあります。 |
| ^V | 16H | カーソルを画面の左上端へ，次に左下端に動かします。 |
| ^B | 02H | カーソルを行の左端へ，行の右端に動かします。 |
| ^W | 17H | 画面上でファイルを１行上端に追加します。 |
| ^Z | 1AH | 画面上でファイルを１行下端に追加します。 |
| ^R | 12H | ファイルを画面上で１ページ分戻します。 |
| ^C | 03H | ファイルを画面上で１ページ分次へ動かします。 |
| ^G | 07H | カーソル位置の文字を削除します。 |
| DEL | 08H | カーソルの左側の文字を削除します。 |
| ^L | 0CH | カーソルの左のワードを削除します。 |
| ^T | 14H | カーソルの右のワードを削除します。 |
| ^U | 15H | カーソルから行の左側を削除します。（ただし復帰は削除しません。） |
| ^K | 0 BH | カーソルから行の右側を削除します。（ただし復帰は削除しません。） |
| ^Y | 19H | カーソルのある行を復帰を含めて削除します。 |
| ^I | 09H | タブ。ファイル文字として扱われます。 |
| ^N | 14H | リターンコードを挿入し，カーソルをその前に置きます。 |
| <CR> | 0BH | リターンコード。<br>挿入モードが ON のとき，復帰をファイルに挿入します。<br>挿入モードが OFF のとき，カーソルを次の行の先頭に動かします。<br>ファイルの終わりのとき，ファイルに追加します。 |
| ^@ | 00H | 次のコマンド文字を４回繰り返します。 |
| ^P | 10H | 次の文字（制御文字を含む）をファイルに挿入します。 |
| ESC | 1BH | ビデオモードからコマンドモードに移ります。 |

## 2．ワードマスターコマンドのまとめ

| | |
|---|---|
| +− | +または−。省略した場合は+。一般に+はファイルの終端方向，−は先端方向を表わしています。 |
| $ | ESC，∧Zまたは復帰。 |
| n | 0から65535までの整数。省略した場合は1，#は65535。 |
| +−nC | ポインタをn文字動かします。 |
| +−B | ポインタをファイルの先頭（+）終端（−）に動かします。 |
| +−nD | n文字削除します。 |
| +−nL | ポインタをn行動かします。 |
| +−nK | n行削除します。 |
| +−nT | n行表示します。 |
| +−nP | nページ（1ページ23行）ポインタを動かし，表示します。 |
| +−n ↵ | ポインタをn行動かし，1行表示します。 |
| nIテキスト$ | ポインタ位置にテキストをn回挿入します。 |
| nAテキスト$ | n回1Lを実行するごとにテキストを挿入します。 |
| +−nFキー$ | n回，キーで短い検索をします。 |
| +−nNキー$ | n回，キーで長い検索をします。 |
| +−nSキー$新しいテキスト$ | n回キーで短い検索をし，見つかったらそれを新しいテキストで置き換えます。 |
| +−nRキー$新しいテキスト$ | n回キーで長い検索をし，見つかったらそれを新しいテキストで置き換えます。 |
| /F,/N,/S,/R | それぞれ，F，N，S，Rと同様です。ただし，検索が失敗したときは，ループやQXコマンドの処理を終え，実行を続けます。 |
| n<…> | ループ。n回…を繰り返します。nを省略したときは，nを65535とみなします。 |
| Y〔d：〕ファイル名．タイプ$ | 指定されたファイル全てを，ポインタ位置に挿入します。 |
| nW〔d：〕ファイル名．タイプ$ | ポインタからn行を指定したファイルに書き込みます。 |
| nQP | ポインタからn行のテキストをファイルから削除し，スクラッチパッドに入れます。 |
| n/QP | ポインタからn行のテキストをファイルから削除し，スクラッチパッドに追加します。 |
| nQG | スクラッチパッドの内容をn回ファイルのポインタ位置に挿入します。 |
| QT | スクラッチパッドの内容を表示します。 |
| QK | スクラッチパッドの内容を消去します。 |
| QX | スクラッチパッドの内容を実行します。 |
| QLテキスト$ | テキストをスクラッチパッドに入れます。 |
| n/QLテキスト$ | テキストをn回スクラッチパッドに追加します。 |
| nZ | n秒待ちます。 |
| V | ビデオモードになります。 |
| n! | 文字コードnをポインタ位置に挿入します。 |
| ; | これに続く文字を注釈とみなします。 |
| E | ファイルをセーブし，編集を終了します。 |
| H | ファイルをセーブし，編集を継続します。 |
| Q | それまでの編集を無効にし，編集を終了します。 |
| O | それまでの編集を無効にし，編集を継続します。 |

＊ "テキスト" と "キー" の中には以下の特殊文字を含むことができます。

"テキスト" 中の特殊文字： ∧N（復帰改行），∧Y（ESC），

"キー" 中の特殊文字 ： ∧S（セパレーター），∧A（任意の文字）
∧O x（x以外の文字），∧N（復帰改行）
∧Y（ESC）

## 8 エラーメッセージと情報メッセージ

■エラーメッセージ

次のエラーは回復可能です。コマンド行待ちの状態となります。

| エラーメッセージ | 意　　味 |
|---|---|
| # #コマンド行 | F，N，R，S コマンドによる検索失敗。失敗したコマンドとコマンド行の残りが # #のあとに表示されます。 |
| ???コマンド行 | コマンド行の中に未定義コマンド。該当コマンドとコマンド行の残りの部分が???の後に表示されます。下のメッセージと共に表示されることもあります。 |
| QX?コマンド行 | QXコマンド実行中，スクラッチパッドに未定義コマンド。スクラッチパッドの残りの部分が表示されます。 |
| DISK FULL | ディスケットの空領域不足<br>回復：ポインタを後退方向に動かしたため DISK FULL エラーが起こることがあります。そのときは，カーソルを前方に動かすか，E コマンドを使うことによって回復します。ファイルの不要な行を削除してセーブしなおすことも可能です。<br>DISK FULL エラーの防止：編集を始める前に，新しいファイルの入る十分な領域があるか確かめてください。 |
| DIRECTORY FULL | ディスケットのファイルディレクトリー領域不足。 |
| Q-BUF FULL | スクラッチパッドの領域不足。QL，QP，QA コマンドは中断します。 |
| MEMORY SHORTAGE,<br>TRY CLEARING QBUF | スクラッチパッドがフルのとき，このメッセージがでます。QK コマンドをタイプしてから処理を続けてください。 |
| FILE LINE LONGER<br>THAN ENTIRE SCREEN | 画面上に改行のないテキストをビデオ編集しようとしたとき出ます。ビデオ編集の前に I コマンドで復帰を挿入してください。 |
| PUTOUR ERR | 内部エラー。 |
| A 〈CR〉 AND I 〈CR〉 ARE<br>NO LONGER SUPPORTED.<br>USE VIDEO MODE | A 改行，I 改行は WM2.0K バージョンからはサポートされていません。ビデオモードをご使用下さい。 |

次のエラーは回復不能です。ワードマスターの開始時に発生し，OS にすぐ戻ります。

| エラーメッセージ | 意　　味 |
|---|---|
| FILE EXISTS | ソースディスクと宛先ディスクが異なり，与えられた名前のファイルが宛先ディスクにすでにあるときにでます。ディスクのファイルを削除するか，リネームしてください。 |
| NO FILE NAME | OS のコマンドとして WM ↵ をタイプしたときにでます。"WM"の次にファイル名がありません。ワードマスターを開始するには，コンソールで編集したり，作成したりするファイルの名前が必要です。 |

**■情報メッセージ**

| エラーメッセージ | 意　　味 |
|---|---|
| NEW FILE | ワードマスターで指定したファイルが存在しないときタイプされます。 |
| INTERRUPTED | ^C コマンドをアボートしたときタイプされます。 |
| EOF!! | コマンドモードや^P^Z（ビデオモード）で end of-file の識別子を挿入したとき表示されます。特別の意図がなければ削除した方がよいでしょう。 |

# 第14章

# 付　録

# 1 コマンド一覧表

ビルトインコマンド，トランジェントコマンド，およびturbo CP/M強化コマンドを，ABC順に一覧表にしてあります。

## 1．ビルトインコマンド

| コマンド | 形　式 | 機　能 |
|---|---|---|
| DIR | DIR d：<br>DIR d：filename. ext | 指定ディスクのファイルのリストアウト<br>指定ファイルのリストアウト（グループ指定可能） |
| ERA | ERA d：filename. ext | 指定ファイルの削除 |
| REN | REN d：newfilename. ext=oldfilename. ext | ファイル名の変更 |
| SAVE | SAVE n d：filename. ext | メモリー100H番地からnページ分の内容をディスクにセーブ。<br>（1ページ=256バイト，nは10進） |
| TYPE | TYPE d：filename. ext | アスキーファイルの内容表示 |
| USER | USER n | ユーザー番号nを指定（nは10進の0～15） |

## 2．トランジェントコマンド

| コマンド | 形　式 | 機　能 |
|---|---|---|
| ASM | ASM d：filename<br>ASM filename. sol | ファイル名filename. ASMのアセンブル<br>ファイル名filename. ASMのアセンブル<br>s　filename. ASMのあるドライブ<br>o　HEXファイルのできるドライブ（Zで生成せず）<br>l　PRNファイルのできるドライブ<br>（Xでコンソール出力，Zで生成せず） |
| DDT | DDT<br>DDT d：filename. ext | DDT起動<br>DDT起動後，指定ファイルをメモリーにロードします。 |
| | Aaaaa<br>Daaaa, bbbb<br><br>Faaaa,bbbb,cc<br>Gaaaa,$b_1b_1b_1b_1$,$b_2b_2b_2b_2$<br><br>Haaaa, bbbb<br>I filename. ext<br>Laaaa, bbbb | アドレスaaaaH番地から，1ステップずつラインアセンブルします。<br>アドレスaaaaH番地からbbbbH番地まで，メモリーダンプします。<br>（bbbbを省略すると1行16バイトを12行表示）<br>アドレスaaaaH番地からbbbbH番地まで，データCCHで満たします。<br>プログラムをアドレスaaaaH番地から実行します。<br>$b_1b_1b_1b_1$，$b_2b_2b_2b_2$はブレークポイント。<br>16進計算aaaaH+bbbbH，aaaaH−bbbbHを表示します。<br>ファイルfilename. extをファイルコントロールブロックに登録します。<br>アドレスaaaaH番地からbbbbH番地まで，逆アセンブルします。<br>（bbbbを省略すると11行分逆アセンブル） |

| コマンド | 形　　式 | 機　　能 |
|---|---|---|
| DDT | Maaaa，bbbb，cccc | アドレスaaaaH番地からbbbbH番地までのデータを，アドレスccccH番地からブロック転送します。 |
| | Raaaa | ファイルコントロールブロックに登録されたファイルを，メモリーにロードします。aaaaHは，正のオフセット値（省略可能）。 |
| | Saaaa | アドレスaaaaH番地からのメモリー内容表示を変更します。 |
| | Tn | 現プログラムカウンターから，プログラムをn(10進)ステップトレース表示します。 |
| | Un | 現プログラムカウンターから，プログラムをnステップ実行，最終ステップをトレース表示します。 |
| | X | CPUの状態を表示します。 |
| | Xr | 指定レジスタ，もしくはフラグの内容表示と変更を行ないます。<br>※r……C　キャリー<br>Z　ゼロ　　BC<br>M　マイナス　　DE<br>E　偶数パリティ　　HL<br>I　auxキャリー　　S（スタックポインタ）<br>A　　P（プログラムカウンタ） |
| DUMP | DUMP d：filename. ext | 指定ファイルの内容を16進表示します。 |
| ED | ED filename. ext | ファイルの作成，編集開始 |
| | nA | テキストn行を，メモリーバッファに取り込みます。<br>（n＝#で全テキスト，n＝0でメモリバッファの半分） |
| | E | 編集終了，テキストをセーブしCP/Mに戻ります。 |
| | H | 編集終了，テキストをセーブし，メモリーバッファをクリアします。 |
| | O | 編集中止，メモリーバッファをクリアします。 |
| | Q | 編集中止，ファイルの変更なしにCP/Mに戻ります。 |
| | nW | バッファ先頭よりn行を，テンポラリファイルにセーブします。 |
| | ±B | CPをテキストの最初または最後（−B）に移動します。 |
| | ±nC | CPを±n文字移動します。 |
| | ±nL | CPを±n行移動し，その行の先頭にセットします。 |
| | ±n | ±nLTと同じ |
| | n： | CPをラインnの先頭にセットします。 |
| | ↵ | LTと同じ機能です。 |
| | I | CP位置に，テキスト挿入開始します。(CTRL+Z で解除) |
| | I文字列 | CP位置に，文字列挿入します。 |
| | ±nD | CP位置から，±n文字削除します。 |
| | ±nK | CP位置から，±n行削除します。 |

| コマンド | 形　　式 | 機　　能 |
|---|---|---|
| ED | nF文字列 | CP以後のメモリーバッファにあるテキストで，n番目の指定文字列の最後に，CP移動します。 |
| | nN文字列 | CP以後の全テキストに対する同上の機能です。 |
| | nS文字列１<br>^Z文字列２ | CP以後，文字列１を文字列２に変更するのをn回，繰り返します。 |
| | nJ文字列１<br>^Z文字列２<br>^Z文字列３ | CP以後，文字列１から文字列３までを文字列２に変更するのをn回繰り返します。 |
| | ±U | Uコマンド以後，入力する文字をすべて大文字に変換します。<br>（－Uで解除） |
| | ±nP<br>±nT<br>±V<br>0V<br>nZ<br>nX<br>0X<br>R<br>Rfilename | CP位置から，±nページ(1ページ23行)をタイプアウトします。<br>CP位置から，±n行をタイプアウトします。<br>行番号を表示するモード，表示しないモード（－V）にします。<br>メモリーバッファ容量を，空きエリア数／全エリア数で表示します。<br>コンソール出力ディレイ（n＝１で約1/4秒）<br>CP位置より，n行をX$$$$$$$$.LIBファイルにセーブします。<br>X$$$$$$$$.LIBファイルを空にします。<br>CP位置よりX$$$$$$$$.LIBファイルの内容を挿入します。<br>CP位置よりfilename.LIBファイルの内容を挿入します。 |
| | nMsubcommand<br><br>n：subcommand<br>：nsubcommand<br>n1：n2subcommand | CP位置よりsubcommandをn回，繰り返します。<br>（nを省略するとバッファの最後まで）<br>行番号nよりsubcommandを実行します。<br>行番号nまでsubcommandを実行します。<br>行番号n1からn2までsubcommandを実行します。 |
| | | （※CPはキャラクターポインタ<br>±の＋は省略可能<br>n＝1のとき１は省略可能） |
| LOAD | LOAD d：filename | filename.HEXファイルからfilename.COMファイルを作成します。 |
| MOVCPM | MOVCPM n * | サイズnK CP/Mシステムを生成します。 |

| コマンド | 形　式 | 機　能 |
|---|---|---|
| PIP | PIP<br>PIP command line | PIPを起動します。<br>PIPを起動し，コマンドを実行した後CP/Mに戻ります。 |
| | d：=s：filename. ext<br>d：newfilename. ext<br>=s：filename. ext<br>d：newfilename. ext<br>=s：filename1. ext<br>，filename2. ext | s：からd：へのファイルの転送（グループ指定可能）<br>s：からd：へファイル名を変更して転送<br><br>sのfilename1. extとfilename2. extを連結して転送 |
| | device：=s：filename. ext<br>d：filename. ext=device：<br>device 0：=device 1： | ファイルを指定デバイスへ出力<br>デバイスのデータをファイルとして転送<br>デバイス１からデバイス０へのデータ転送 |

■PIP特別デバイス

| 形　式 | 機　能 |
|---|---|
| EOF： | エンドオブファイルデバイス，1AH（CTRL＋Z）を出力する。 |
| NUL： | ヌルコードデバイス，40個の00H（NULL）を出力する。 |
| PRN： | LST：デバイスに，行番号，８文字タブ，60行ごとのページ送り機能をつけたもの |
| INP：　OUT： | ユーザーがPIP内で新たに定義した入力／出力デバイス |

| オプション | 機　能 |
|---|---|
| 〔B〕 | X-off(13H)が入力されるまでブロック受信を行ない，その後データ転送をします。 |
| 〔Dn〕 | １行の字数をｎ字で打ち切ります。 |
| 〔E〕 | 転送内容をコンソールに出力します。 |
| 〔F〕 | 転送データから，フォームフィードコード(0CH)を削除します。 |
| 〔Gn〕 | ユーザーｎからのファイル転送をします。 |
| 〔H〕 | HEXファイルのインテルフォーマットチェックです。 |
| 〔I〕 | 〔H〕の機能およびヌルコード(00H)を無視します。 |
| 〔L〕 | 大文字を小文字に変換して転送をします。 |
| 〔N〕 | 行番号：を付けて転送をします。(N2とした場合，空白桁に０表示) |
| 〔O〕 | ファイルのEOFコード(1AH)を無視します。 |
| 〔Pn〕 | ｎ行ごとにページを送ります。(ｎ＝１，または省略したときは60行) |
| 〔Q文字列^Z〕 | データ転送中に，指定文字列があると転送終了します。(文字列を含む) |
| 〔S文字列^Z〕 | 指定文字列が見つかると，データ転送開始します。(文字列を含む) |
| 〔R〕 | $SYSの属性のついたファイルの転送をします。 |
| 〔Tn〕 | タブをｎ文字に設定します。 |
| 〔U〕 | 小文字を大文字に変換して転送をします。 |
| 〔V〕 | 転送ファイルの確認をします。 |
| 〔W〕 | 転送先のR/O属性無視をします。 |
| 〔Z〕 | 入力データのビット７を０にします。 |

| コマンド | 形　　式 | 機　　能 |
|---|---|---|
| STAT | STAT | アクセスされたディスクの未使用エリアと属性の表示 |
| | STAT d: | 指定ディスクの未使用エリアの表示 |
| | STAT d:DSK: | 指定ディスクの諸元の表示 |
| | STAT VAL: | STATコマンドのメニューの表示 |
| | STAT DEV: | 入出力装置の割付けの表示 |
| | STAT USR: | 現在のユーザー番号と使用されているユーザー番号の表示 |
| | STAT d:filename. ext $S | 指定ファイルの大きさの表示（グループ指定可能）<br>$Sはサイズ表示指示（省略可能） |
| | STAT d:=R/O | 指定ドライブをR/O（読み出し専用）に設定<br>（d：省略不可） |
| | STAT d:filename. ext<br>$atr | 指定ファイルの属性変更（グループ指定可能）<br>$atr……$R/O 読み出し専用　$SYS システムファイル<br>$R/W 書き込み可能　$DIR　ディレクトリ表示可能ファイル |
| | STAT logical:=physical: | 入出力装置の割付けの変更 |
| SUBMIT | SUBMIT subfile<br>SUBMIT subfile $P_1$ $P_2$… | subfile. SUBファイルの内容を連続実行します。<br>代用パラメータを，実パラメータ($P_1$，$P_2$，……)に替えて，<br>subfile. SUBファイルの内容を連続実行します。<br>＊XSUBによって，トランジェントコマンド内のキー入力手順を指示できます。 |

## 3．turbo CP/M強化コマンド

| コマンド | 機　　能 |
|---|---|
| CMT | ディスクのファイルをカセットテープにセーブしたり，カセットテープのファイルをディスクにセーブします。 |
| COPY | フロッピーディスクの内容すべてを他のフロッピーディスクにコピーします。 |
| DICCOPY | 辞書のコピーをします。 |
| DICSET | 音訓・システム・ユーザー辞書の入っているドライブを指定します。 |
| DISKSET | デバイスの設定，確認および設定の変更を行ないます。 |
| FORMAT | フロッピーディスクを初期状態にします。 |
| HDBACKUP | ハードディスクとフロッピーディスク間で，ファイルのバックアップを行ないます。 |

| コマンド | 機　　能 |
|---|---|
| HDMAINT | ハードディスクの領域をCP/Mがどのように使用するかを決定したり，確認したりします。 |
| KEY | ファンクションキーに対して，文字列を定義したり，定義状態を表示したりします。 |
| STARTUP | CP/M起動時に種々の設定｛ファンクションキー<br>先行キー入力<br>RS-232Cパラメータ<br>プリンタ<br>ユーザー指定のドライブパラメータ<br>初期IOバイト<br>漢字・非漢字モード｝を行ないます。 |
| SYSGEN | CP/Mシステムをコピーします。 |
| TERM | RS-232Cインターフェイスを介して，他の機器とデータ通信を行ないます。 |
| TIME | 内蔵しているタイマーの時間設定を行ないます。 |

## 2 ニーモニック一覧表

8080命令セットを，それに対応する16進コード，及びZ-80命令セットとともに，ABC順に一覧表にしてあります。

turbo CP／MのASMは，8080ニーモニックをアセンブルします。

nn　＝8ビット引数
nnnn＝16ビット引数

| 8080命令セット | | | Z-80命令セット | |
|---|---|---|---|---|
| ニーモニック | オペランド | HEX（16進） | ニーモック | オペランド |
| ACI | nn | CE nn | ADC | A, nn |
| ADC | A | 8 F | ADC | A, A |
| ADC | B | 8 8 | ADC | A, B |
| ADC | C | 8 9 | ADC | A, C |
| ADC | D | 8 A | ADC | A, D |
| ADC | E | 8 B | ADC | A, E |
| ADC | H | 8 C | ADC | A, H |
| ADC | L | 8 D | ADC | A, L |
| ADC | M | 8 E | ADC | A, (HL) |
| ADD | A | 8 7 | ADD | A, A |
| ADD | B | 8 0 | ADD | A, B |
| ADD | C | 8 1 | ADD | A, C |
| ADD | D | 8 2 | ADD | A, D |
| ADD | E | 8 3 | ADD | A, E |
| ADD | H | 8 4 | ADD | A, H |
| ADD | L | 8 5 | ADD | A, L |
| ADD | M | 8 6 | ADD | A, (HL) |
| ADI | nn | C 6 nn | ADD | A, nn |
| ANA | A | A 7 | AND | A |
| ANA | B | A 0 | AND | B |
| ANA | C | A 1 | AND | C |
| ANA | D | A 2 | AND | D |
| ANA | E | A 3 | AND | E |
| ANA | H | A 4 | AND | H |
| ANA | L | A 5 | AND | L |
| ANA | M | A 6 | AND | (HL) |
| ANI | nn | E 6 nn | AND | nn |
| CALL | nnnn | CD nnnn | CALL | nnnn |
| CC | nnnn | DC nnnn | CALL | C, nnnn |
| CM | nnnn | FC nnnn | CALL | M, nnnn |
| CMA | | 2 F | CPL | |
| CMC | | 3 F | CCF | |

| 8080命令セット | | HEX（16進） | | Z-80命令セット | |
|---|---|---|---|---|---|
| CMP | A | BF | | CP | A |
| CMP | B | B8 | | CP | B |
| CMP | C | B9 | | CP | C |
| CMP | D | BA | | CP | D |
| CMP | E | BB | | CP | E |
| CMP | H | BC | | CP | H |
| CMP | L | BD | | CP | L |
| CMP | M | BE | | CP | (HL) |
| CNC | nnnn | D4 | | CALL | NC, nnnn |
| CNZ | nnnn | C4 | | CALL | NZ, nnnn |
| CP | nnnn | F4 | | CALL | P, nnnn |
| CPE | nnnn | EC | | CALL | PE, nnnn |
| CPI | | FE | nn | CP | nn |
| CPO | | E4 | nnnn | CALL | PO, nnnn |
| CZ | | CC | nnnn | CALL | Z, nnnn |
| DAA | | 27 | | DAA | |
| DAD | B | 09 | | ADD | HL, BC |
| DAD | D | 19 | | ADD | HL, DE |
| DAD | H | 29 | | ADD | HL, HL |
| DAD | SP | 39 | | ADD | HL, SP |
| DCR | A | 3D | | DEC | A |
| DCR | B | 05 | | DEC | B |
| DCR | C | 0D | | DEC | C |
| DCR | D | 15 | | DEC | D |
| DCR | E | 1D | | DEC | E |
| DCR | H | 25 | | DEC | H |
| DCR | L | 2D | | DEC | L |
| DCR | M | 35 | | DEC | (HL) |
| DCX | B | 0B | | DEC | BC |
| DCX | D | 1B | | DEC | DE |
| DCX | H | 2B | | DEC | HL |
| DCX | SP | 3B | | DEC | SP |
| DI | | F3 | | DI | |
| EI | | F8 | | EI | |
| HLT | | 76 | | HALT | |
| IN | nn | DB | nn | IN | A, (nn) |
| INR | A | 3C | | INC | A |
| INR | B | 04 | | INC | B |
| INR | C | 0C | | INC | C |
| INR | D | 14 | | INC | D |
| INR | E | 1C | | INC | E |
| INR | H | 24 | | INC | H |
| INR | L | 2C | | INC | L |

| 8080命令セット | | HEX（16進） | | Z-80命令セット | |
|---|---|---|---|---|---|
| INR | M | 3 4 | | INC | (HL) |
| INX | B | 0 3 | | INC | BC |
| INX | D | 1 3 | | INC | DE |
| INX | H | 2 3 | | INC | HL |
| INX | SP | 3 3 | | INC | SP |
| JC | nnnn | DA | nnnn | JP | C, nnnn |
| JM | nnnn | FA | nnnn | JP | M, nnnn |
| JMP | nnnn | C 3 | nnnn | JP | nnnn |
| JNC | nnnn | D 2 | nnnn | JP | NC,nnnn |
| JNZ | nnnn | C 2 | nnnn | JP | NZ, nnnn |
| JP | nnnn | F 2 | nnnn | JP | P,nnnn |
| JPE | nnnn | EA | nnnn | JP | PE, nnnn |
| JPO | nnnn | E 2 | nnnn | JP | PO, nnnn |
| JZ | nnnn | CA | nnnn | JP | Z, nnnn |
| LDA | nnnn | 3 A | nnnn | LD | A, (nnnn) |
| LDAX | B | 0 A | | LD | A, (BC) |
| LDAX | D | 1 A | | LD | A, (DE) |
| LHLD | nnnn | 2 A | nnnn | LD | HL,(nnnn) |
| LXI | B, nnnn | 0 1 | nnnn | LD | BC, nnnn |
| LXI | D, nnnn | 1 1 | nnnn | LD | DE, nnnn |
| LXI | H, nnnn | 2 1 | nnnn | LD | HL, nnnn |
| LXI | SP, nnnn | 3 1 | nnnn | LD | SP, nnnn |
| MOV | A, A | 7 F | | LD | A, A |
| MOV | A, B | 7 8 | | LD | A, B |
| MOV | A, C | 7 9 | | LD | A, C |
| MOV | A, D | 7 A | | LD | A, D |
| MOV | A, E | 7 B | | LD | A, E |
| MOV | A, H | 7 C | | LD | A, H |
| MOV | A, L | 7 D | | LD | A, L |
| MOV | A, M | 7 E | | LD | A, (HL) |
| MOV | B, A | 4 7 | | LD | B, A |
| MOV | B, B | 4 0 | | LD | B, B |
| MOV | B, C | 4 1 | | LD | B, C |
| MOV | B, D | 4 2 | | LD | B, D |
| MOV | B, E | 4 3 | | LD | B, E |
| MOV | B, H | 4 4 | | LD | B, H |
| MOV | B, L | 4 5 | | LD | B, L |
| MOV | B, M | 4 6 | | LD | B, (HL) |
| MOV | C, A | 4 F | | LD | C, A |
| MOV | C, B | 4 8 | | LD | C, B |
| MOV | C, C | 4 9 | | LD | C, C |
| MOV | C, D | 4 A | | LD | C, D |
| MOV | C, E | 4 B | | LD | C, E |

| 8080命令セット | | HEX（16進） | | Z-80命令セット | |
|---|---|---|---|---|---|
| MOV | C, H | 4C | | LD | C, H |
| MOV | C, L | 4D | | LD | C, L |
| MOV | C, M | 4E | | LD | C, (HL) |
| MOV | D, A | 5 7 | | LD | D, A |
| MOV | D, B | 5 0 | | LD | D, B |
| MOV | D, C | 5 1 | | LD | D, C |
| MOV | D, D | 5 2 | | LD | D, D |
| MOV | D, E | 5 3 | | LD | D, E |
| MOV | D, H | 5 4 | | LD | P, H |
| MOV | D, L | 5 5 | | LD | D, L |
| MOV | D, M | 5 6 | | LD | D, (HL) |
| MOV | E, A | 5F | | LD | E, A |
| MOV | E, B | 5 8 | | LD | E, B |
| MOV | E, C | 5 9 | | LD | E, C |
| MOV | E, D | 5A | | LD | E, D |
| MOV | E, E | 5B | | LD | E, E |
| MOV | E, H | 5C | | LD | E, H |
| MOV | E, L | 5D | | LD | E, L |
| MOV | E, M | 5E | | LD | E, (HL) |
| MOV | H, A | 6 7 | | LD | H, A |
| MOV | H, B | 6 0 | | LD | H, B |
| MOV | H, C | 6 1 | | LD | H, C |
| MOV | H, D | 6 2 | | LD | H, D |
| MOV | H, E | 6 3 | | LD | H, E |
| MOV | H, H | 6 4 | | LD | H, H |
| MOV | H, L | 6 5 | | LD | H, L |
| MOV | H, M | 6 6 | | LD | H, (HL) |
| MOV | L, A | 6F | | LD | L, A |
| MOV | L, B | 6 8 | | LD | L, B |
| MOV | L, C | 6 9 | | LD | L, C |
| MOV | L, D | 6A | | LD | L, D |
| MOV | L, E | 6B | | LD | L, E |
| MOV | L, H | 6C | | LD | L, H |
| MOV | L, L | 6D | | LD | L, L |
| MOV | L, M | 6E | | LD | L, (HL) |
| MOV | M, A | 7 7 | | LD | (HL), A |
| MOV | M, B | 7 0 | | LD | (HL), B |
| MOV | M, C | 7 1 | | LD | (HL), C |
| MOV | M, D | 7 2 | | LD | (HL), D |
| MOV | M, E | 7 3 | | LD | (HL), E |
| MOV | M, H | 7 4 | | LD | (HL), H |
| MOV | M, L | 7 5 | | LD | (HL),L |
| MVI | A, nn | 3E | nn | LD | A, nn |

| 8080命令セット | | HEX（16進） | | Z-80命令セット | |
|---|---|---|---|---|---|
| MVI | B, nn | 0 6 | nn | LD | B, nn |
| MVI | C, nn | 0 E | nn | LD | C, nn |
| MVI | D, nn | 1 6 | nn | LD | D, nn |
| MVI | E, nn | 1 E | nn | LD | E, nn |
| MVI | H, nn | 2 6 | nn | LD | H, nn |
| MVI | L, nn | 2 E | nn | LD | L, nn |
| MVI | M, nn | 3 6 | nn | LD | (HL), nn |
| NOP | | 0 0 | | NOP | |
| ORA | A | B 7 | | OR | A |
| ORA | B | B 0 | | OR | B |
| ORA | C | B 1 | | OR | C |
| ORA | D | B 2 | | OR | D |
| ORA | E | B 3 | | OR | E |
| ORA | H | B 4 | | OR | H |
| ORA | L | B 5 | | OR | L |
| ORA | M | B 6 | | OR | (HL) |
| ORI | nn | F 6 | nn | OR | nn |
| OUT | nn | D 3 | nn | OUT | (nn), A |
| PCHL | | E 9 | | JP | (HL) |
| POP | B | C 1 | | POP | BC |
| POP | D | D 1 | | POP | DE |
| POP | H | E 1 | | POP | HL |
| POP | PSW | F 1 | | POP | AF |
| PUSH | B | C 5 | | PUSH | BC |
| PUSH | D | D 5 | | PUSH | DE |
| PUSH | H | E 5 | | PUSH | HL |
| PUSH | PSW | F 5 | | PUSH | AF |
| RAL | | 1 7 | | RLA | |
| RAR | | 1 F | | RRR | |
| RC | | D 8 | | RET | C |
| RET | | C 9 | | RET | |
| RLC | | 0 7 | | RLCA | |
| RM | | F 8 | | RET | M |
| RNC | | D 0 | | RET | NC |
| RNZ | | C 0 | | RET | NZ |
| RP | | F 0 | | RET | P |
| RPE | | E 8 | | RET | PE |
| RPO | | E 0 | | RET | PO |
| RRC | | 0 F | | RRCA | |
| RST | 0 | C 7 | | RST | 0 |
| RST | 1 | CF | | RST | 8 |
| RST | 2 | D 7 | | RST | 10H |
| RST | 3 | DF | | RST | 18H |

| 8080命令セット | | HEX（16進） | | Z-80命令セット | |
|---|---|---|---|---|---|
| RST | 4 | E 7 | | RST | 20H |
| RST | 5 | EF | | RST | 28H |
| RST | 6 | F 7 | | RST | 30H |
| RST | 7 | FF | | RST | 38H |
| RZ | | C 8 | | RET | Z |
| SBB | A | 9 F | | SBC | A, A |
| SBB | B | 9 8 | | SBC | A, B |
| SBB | C | 9 9 | | SBC | A, C |
| SBB | D | 9 A | | SBC | A, D |
| SBB | E | 9 B | | SBC | A, E |
| SBB | H | 9 C | | SBC | A, H |
| SBB | L | 9 D | | SBC | A, L |
| SBB | M | 9 E | | SBC | A, (HL) |
| SBI | nn | DE | nn | SBC | A, nn |
| SHLD | nnnn | 2 2 | nnnn | LD | (nnnn), HL |
| SPHL | | F 9 | | LD | SP, HL |
| STA | nnnn | 3 2 | | LD | (nnnn), A |
| STAX | B | 0 2 | | LD | (BC), A |
| STAX | D | 1 2 | | LD | (DE), A |
| STC | | 3 7 | | SCF | |
| SUB | A | 9 7 | | SUB | A |
| SUB | B | 9 0 | | SUB | B |
| SUB | C | 9 1 | | SUB | C |
| SUB | D | 9 2 | | SUB | D |
| SUB | E | 9 3 | | SUB | E |
| SUB | H | 9 4 | | SUB | H |
| SUB | L | 9 5 | | SUB | L |
| SUB | M | 9 6 | | SUB | (HL) |
| SUI | nn | D 6 | nn | SUB | nn |
| XCHG | | EB | | EX | DE, HL |
| XRA | A | AF | | XOR | A |
| XRA | B | A 8 | | XOR | B |
| XRA | C | A 9 | | XOR | C |
| XRA | D | AA | | XOR | D |
| XRA | E | AB | | XOR | E |
| XRA | H | AC | | XOR | H |
| XRA | L | AD | | XOR | L |
| XRA | M | AE | | XOR | (HL) |
| XRI | nn | EE | nn | XOR | N |
| XTHL | | E 3 | | EX | (SP), HL |

## 3 プリンタの登録

以下の操作を行なうことにより登録プリンタ以外に一機種プリンタを登録することができます。

〈ご 注 意〉

＊使用しているプリンタにビットイメージ印字コードが無い場合は画面コピーはできません。また，ビットイメージ印字コードと漢字モードのどちらも無い場合は，漢字印字は行なえません。その他，プリンタに関する制限事項については，X1 turbo BASIC(CZ-8FB02) のプリンタユーティリティと同様になります。

＊プリンタ登録のためには，次のファイルが入っているディスクが必要です。
STARTUP.COM，ASM.COM，DDT.COM，WM.COM，WM.HELP と ULPT0.ASM，ULPT1.ASM，ULPT2.ASM のいずれか。

■登録手順

1．WMによるアセンブラソースファイルの変更

登録するプリンタがビットイメージ印字，漢字モードを持っているかどうかによって WM で変更するファイルを下記の3つの中から選択します。WM を使ってソースファイルに書きこまれたコメントに従いデータを変更してください。変更内容に関しては X1 turbo BASIC "アプリケーションソフトの説明" のプリンタユーティリティを参照してください。＊

データ変更用のソースファイル

ULPT0.ASM …… ビットイメージ印字あり，漢字モードあり
ULPT1.ASM …… ビットイメージ印字あり，漢字モードなし
ULPT2.ASM …… ビットイメージ印字なし，漢字モードあり

なお，変更は次のように擬似命令 EQU の各データに対して行なってください。

```
;改ページコード
;
FFCODE EQU 00CH
           └→このデータを変更
```

＊プリンタユーティリティで変更内容が省略されている場合、データ長は0にしてください。

2. STARTUPに登録

次に変更したソースファイルをアセンブルした後，DDTを使用して次のようにSTARTUPに変更データを登録してください。

＊以下の例はULPT0.ASMで変更した場合を示しています。

```
A>ASM ULPT0↵──────────①
CP/M ASSEMBLER - VER 2.0
0134
003H USE FACTOR
END OF ASSEMBLY

A>DDT STARTUP.COM↵────②
DDT VERS 2.2
NEXT  PC
1F80 0100
-IULPT0.HEX↵
-R↵
NEXT PC
1F80 0000
-G0↵

A>SAVE 32 STARTUP.COM↵─③
```

①まず変更したソースファイルをアセンブルします。

②STARTUPを変更します。

＊NEXT，PCの値はその時の状態により変化します。

③変更したSTARTUPをディスクにセーブします。

以上の作業により，STARTUPに新しいプリンタが登録されプリンタの選択画面でOTHERを選択することでプリンタ設定を行なうことができます。（7章のSTARTUPを参照してください。）

# 4 エラーメッセージ

turbo CP/M のエラーメッセージには，CP/M に共通なものと，各コマンドやプログラムに特有なものがあります。通常起こりやすいエラーの原因は次のようなものが考えられます。

- ○未フォーマットのディスケットを使用した場合。
- ○ディスケットを交換した後，CTRL + C を実行せずに書き込みを行なおうとした場合。
- ○リードオンリ状態のディスケットに書き込みや削除を行なおうとした場合。
- ○ディスクの全メモリ容量をオーバーして書き込みを行なおうとした場合。
- ○ファイル名やドライブ名を誤まって指定した場合。
- ○ディスケットが損傷した場合。

エラーによって処理がストップした場合，通常は CTRL + C を実行してリブートし，CP/M に戻すのが最も安全です。エラーの種類によっては，CTRL + C 以外のキーインを行なっても処理を続行できるものもありますが，その場合は後で必ずデータのチェックをする必要があります。

具体的に BDOS エラーメッセージと，BIOS エラーメッセージの原因と対策を一覧にします。

● BDOS エラーメッセージ

| エラーメッセージ | 原 因 ・ 対 策 |
|---|---|
| BAD SECTOR | ディスクの読み書き中に起こるエラーです。<br>ディスク・コントローラやディスケットの状態をチェックする必要があります。<br>回復させるには CTRL + C をキーインしてリブートしてください。リターンキーを入力すると，それがディレクトリ・ライト時のエラーであった場合，そのディスケットを壊す恐れがあります。 |
| SELECT | 規定外のドライブを指定した場合に表示されます。<br>回復させるには何かキーインしてください。自動的にリブートされます。 |
| READ ONLY | リード・オンリ状態のディスケットに書き込みを行なおうとしたとき表示されます。<br>回復させるには何かキーインしてください。自動的にリブートされます。 |

● BIOS エラーメッセージ

| エラーメッセージ | 原 因 ・ 対 策 |
|---|---|
| Reboot error | リブートして CP/M システムに戻すことができない状態です。<br>ディスク・コントローラやディスケットの状態をチェックしてください。 |
| DPH work overflow | ワークエリアが足りなくなった場合に表示されます。 |

MEMO

MEMO

MEMO

MEMO

MEMO



シャープ株式会社

本　　社　〒545 大阪市阿倍野区長池町22番22号　電話 (06)621-1221(大代表)
東京支社　〒162 東京都新宿区市谷八幡町8番地　電話 (03)260-1161(大代表)
テレビ事業部　第4商品企画部

T0938-A ③