

MSX-DOS-TOOLS

目次

1	MSX-DOS TOOLSの概要	P 1
1.1	UNIXの入出力環境とMSX-DOS	P 1
1.2	MSX-DOS TOOLSの標準入出力	P 1
2	入力と出力	P 2
2.1	I/Oリダイレクション	P 2
2.2	パイプ機能	P 2
2.3	シーケンシャルプロセス	P 3
3	コマンドの書式指定	P 3
3.1	ファイルスベック	P 3
3.2	ファイル名	P 3
3.3	ドライブ名	P 3
3.4	ファイルの間接指定	P 4
3.5	西暦指定	P 5
3.6	パラメータの区切	P 5
4	ツールコマンド一覧	P 6
5	コマンドリファレンス	P 7~40
5.1	コマンドリファレンス表示オプション "/H" について	P 7
6	エラーメッセージ一覧	P 41
付録	TOOLS使用サンプル集	P 45

1

2

1 MSX-DOS TOOLSの概要

MSX-DOS TOOLSはMSX-DOS上でのプログラム開発を強力にサポートする外部コマンド群です。TOOLSを構成するソフトウェアは、28のCOM形式のファイルからなり、DOS上でコマンド行を入力することによりDOSの内部コマンドと同様の操作感覚で実行できます。

個々のコマンドはディスク管理、ファイル操作からカレンダーや電卓機能等、幅広くサポートし、また、パイプライン、I/Oリダイレクション機能（後述）によりUNIXライクな入出力操作が可能です。

1.1 UNIXの入出力環境とMSX-DOS

UNIXオペレーティングシステムがプログラムの開発環境として、その実力を高く評価される理由の一つとして、豊富なツール（UNIXではソフトウェアを何らかの目的を達成するための道具としてとらえる立場から個々の開発支援、その他のプログラムをこう呼ぶ）を機能や目的に応じて縦横に組合せることができるという点あげられます。UNIXの環境下では、あるプロセス（個々のプログラムの実行過程）と他のプロセスの間でデータの受け渡しを行ったり、複数のプロセスをそれぞれ独立にディスクファイルやコンソールから入力を行ったり、プリンターやディスクファイルにデータを出力したりしながら、順次実行していくなどといった柔軟性に富んだ入出力操作が可能です。

こうしたすぐれた入出力環境を実現する共通のインターフェースを、UNIXではOSレベルでサポートし標準入出力として各プログラムが用いることができますが、残念ながらMSX-DOSには標準入出力の概念はなく、UNIXのような入出力環境をOSのレベルではサポートしていません。

1.2 MSX-DOS TOOLSの標準入出力

MSX-DOS TOOLSでは、UNIXライクな操作環境を実現するため、UNIXの標準入出力をコマンド内部でシミュレートしています。

MSX-DOS TOOLSで用いられる標準入出力は以下のような機能をサポートしています。（各機能の詳説は、2.入力と出力をご覧ください）

a) コンソール入出力

通常、入力はキーボードから、出力はスクリーンに表示します。

b) I/Oリダイレクション

入出力をディスクファイルや他のデバイスに切り換えます。

c) パイプ機能

あるコマンドの出力から別のコマンドの入力へデータの受け渡しを行います。

d) シーケンシャルプロセス

複数のコマンドをコマンド間のデータの受け渡しなしにそれぞれ独立に実行します。

2 入力と出力

多くのツールコマンドでは前述の機能を持った標準入出力がサポートされています。コマンドへの入力をキーボードから行ったり、あるいはスクリーンに結果を表示するという通常の入出力の他に後述のリダイレクトを用いて入力をキーボードからディスク上のファイルに切り替えたり、出力をスクリーンでなく、ディスクファイルやプリンタへ転送することができます。また、パイプ機能（後述）により、あるコマンドの出力をそのまま別のコマンドの入力にあてるといったUNIXライクな入出力環境を実現することが可能です。

2.1 I/Oリダイレクション

リダイレクションにより標準入出力を通常のコンソール入出力からプリンタやファイル等に転送することができます。リダイレクトには“<”、“>”の記号を用います。

例1

```
CAL >PRN
```

では、カレンダーをプリンターへ出力し、

例2

```
SORT <LIST1 >B:LIST2
```

ではカレントドライブのLIST1というファイルをソートし、BドライブにLIST2という名でファイル出力します。

さらにPATCHコマンドでは以下のようにバイナリファイルの更新データをファイルから受け取ることにによりキー入力なしに確実なファイル更新が可能です。

```
PATCH FILE.BIN <EDIT.DAT
```

この例ではFILE.BIN というファイルをディスク上のEDIT.DATというファイル中の更新データにしたがって更新します。（詳細はPATCHコマンド参照のこと。）

2.2 パイプ機能

あるコマンドプロセスの標準出力を次のプロセスの標準入力に送りこむという操作をサポートします。

例

```
SORT <LIST1 | HEAD
```

ではカレントドライブのLIST1というファイルがソートされその出力がHEADコマンドの入力となり先頭部分が画面出力されます。

パイプ機能ではディスク上に中間ファイルが作られるためディスクがライトプロテクトされた状態で使用しますと処理を中断します。通常はカレントドライブに中間ファイルは作られますが、“|a”のようにブランクをあけずにドライブ名を指定することにより、つくられるドライブを変更することができます。

また、プロセス間の入出力の受け渡しをせずに個々のプロセスを独立に順次実行する場合には次に述べるシーケンシャルプロセスを用います。

2.3 シーケンシャルプロセス

複数のコマンドプロセスを順次実行します。パイプラインのように前のプロセスの標準出力が後のプロセスの標準入力となることはありません。

各プロセスは“;”でつなぎます。

例

```
HEAD LIST ; SORT LIST
```

では、HEADコマンドがLISTというファイルの先頭部分を画面表示したのち同じファイルLISTをあらためてディスクから読みだしソートして画面に出力します。

注) 以上述べた入出力環境は、TOOLSのコマンドレベルでサポートされているものであり、DIR、TYPE等のMSX-DOSの内部コマンドでは使用できません。

3 コマンドの書式指定

ツールコマンドの書式表記法はDOSの内部コマンドの表記に準拠しますが、以下いくつか補足します。(DOSマニュアル第2部参照)

3.1 ファイルスペック

書式中で“ファイルスペック”となっているパラメータはドライブ名、ファイル名、ピリオド、拡張子名が正しい順序で並んだものを示し、ファイル名、拡張子名はワイルドカードキャラクタ(*, ?)を含むことができます。

さらに上記の指定を“+”または“, ”でつなぐことにより複数指定可能です。(間のブランクは無視されます)

ドライブ名は常に省略可能であり、デフォルトはカレントドライブとなります。また、ファイルスペックのかわりにファイルの間接指定(3.4 ファイルの間接指定を参照のこと)を用いることが可能です。

例

```
b:*.bat+*.c+a:studio.h
```

3.2 ファイル名

“ファイル名”で示されるパラメータはワイルドカードキャラクタを含むことができないという点、複数の指定がゆるぎないという点でファイルスペックと区別されます。その他のについては同様の指定となります。

3.3 ドライブ名

“ドライブ名”で示されるパラメータは接続されているドライブのA~Hまでの各ドライブ名にコロンの“:”をつけたもので示します。

3.4 ファイルの間接指定

ファイルの間接指定とは、複数のファイルを同時に指定可能なコマンドについて、前述の“ファイルスペック”を用いず必要なファイル名をまとめてひとつのファイルにして、コマンド行ではその“ファイル名のファイル”を間接的に指定することです。

ファイルの間接指定により複数のファイル指定を簡単なコマンド行の入力でおこなうことができます。

(1) 間接指定の方法

通常のファイルスペックを指定するかわりに下記のフォーマットにしたがったファイル名のリストファイルを“[]”で囲んで指定します。コマンド側は“[]”で囲まれたファイルの間接指定用のリストファイルとしてオープンしその内容を通常のファイル指定と同様に処理します。

間接指定の“[]”のなかでワイルドカードを用いたり複数のリストファイルを指定することはできません。また、間接指定をもちいた場合はそのみが有効であり、同時にファイルスペックやファイル名の指定をしても無効となります。

(2) ファイル名のリストファイルについて

ファイルの間接指定のためのファイル名のリストファイルは以下のフォーマットで作ります。

- ・必要なファイル名を1行にひとつだけ指定します。指定できるファイル数に制限はありません。
- ・ファイル名にワイルドカードはもちいられません。
- ・各行頭のスペース、タブは無視され最初のキャラクターからファイル名とみなされます。

(3) 間接指定の用例

間接指定の用例としてディスク上の拡張子“bat”のついたすべてのファイルをファイルサイズの更新日の新しい順にそのヘッダー部分を見ながら必要なら別のディスクにコピーするという操作を以下に解説します。

- 1) LSコマンドのリダイレクトによりリストファイルをつくる。

```
ls /t/c *.bat >b:bat.lst
```

LSコマンドの/T、/C オプションによりカレントドライブのディスク上の拡張子“bat”をもったすべてのファイルが最新更新日の新しい順にソート(/Tオプションによる)され一行に一ファイル名のリストが(/Cによる) Bドライブにbat.lstという名でリダイレクトされます。

- 2) bat.lstをファイル間接指定のリストファイルに用いてMENUコマンドにより必要なファイルをコピーする。

```
menu /t [b:bat.lst]
```

これにより拡張子“bat”をもったファイルが更新日の新しい順にプロンプト表示され以下MENUコマンドのTコマンドによりそのヘッダー部分を見なが

ら、必要ならCコマンドでコピーすることができます。

3.5 西暦指定

西暦年 (yyyy) の入力は、4桁のうち上2桁が省略可能でありデフォルトは20世紀となります。

00~99 ...1900~1999年

3.6 オプションスイッチ

“/” にアルファベット1文字をつけた (“/S100” のようにパラメータを伴うこともある) アーギュメントは、コマンドの各種のオプション機能を実行するためのスイッチを意味します。アルファベットは大文字でも小文字でも有効であり、また複数のオプションスイッチを指定する場合、スペース、タブ等で区切らずに続けて指定可能です。

例

① GREP/X/N/A----

② g r e p / x _ / n _ / a ----

①、②どちらも有効です。

3.7 アーギュメントの区切り

ツールコマンドでは、書式中 “_” であらわされるアーギュメントの区切りにはスペースまたはタブのみが有効です。カンマを用いることはできません。

但し “/X” のようなオプションスイッチを複数指定する場合各オプション間のスペースは省略できます。

例

g r e p / x / n / a _ “----” _ < x x x . c

4 ツールコマンド一覧

BEEP	BEEP音の発生
BIO	バイオリズムの出力
BODY	ファイルの一部の切出し
BSAVE	HEXファイルのバイナリファイルへの変換
CAL	カレンダーの表示
CALC	簡易電卓
CHKDSK	ディスクの状態の表示
CLS	クリアスクリーン
DISKCOPY	バックアップコピー及び照合
DUMP	ファイルの16進ダンプ
ECHO	コマンド行の表示
EXPAND	スペース・タブの変換
GREP	任意の文字列の検索
HEAD	ファイルの先頭部分の表示
HELP	コマンドリファレンスの表示
KEY	ファンクションキーの設定
LIST	BASICの中間言語ファイルのソースファイルへの変換
LS	ディレクトリの詳細情報の表示
MENU	ディレクトリのファイルの諸操作
MORE	ファイルの表示
PATCH	バイナリファイルの更新
SLEEP	アラーム機能
SORT	ソート (クイックソート)
TAIL	ファイルの末尾部分の表示
TR	文字列の置き換え
UNIQ	重複行の削除
VIEW	ファイルの表示 (スクロール)
WC	語数、行数、ページ数カウント

5 コマンドリファレンス

5.1 コマンドリファレンス表示オプション “/H” について

すべてのツールコマンドで “/H” オプションを用いることによりコマンドリファレンスを表示、参照することができます。但しこの場合コマンドは実行されません。

以下ツールコマンドをアルファベット順に解説します。

BEEP

書式

BEEP [/H]

機能

BEEP音を発生します。

解説

BEEP音を発生します。

バッチ処理中などで用いることにより、処理の進行の目安とすることができます。
また、BEEPコマンドでは、リダイレクトによりファイルにBELコード（16進の07h）を送ることも可能です。

例

```
A>beep␣  
Beeeeeeeeeeep!  
A>
```

B I O

書式

B I O [/ H] [< y y y y / m m / d d >] [{ [[y y y y /] m m /] d d }]

機能

バイオリズムを出力します。

解説

出力は標準出力です（通常はCRTディスプレイ）

生年月日を指定すると、当日と前後20日のバイオリズムが表示されます。生年月日の指定は 西暦年, 月, 日の順にスラッシュで区切り指定します。

オプションとして、年月日を指定すると、指定の日と前後20日間のバイオリズムが出力されます。この場合、年, 月のデフォルトは当年, 当月で省略可能ですが、月のみの省略は許されません。

例

A>bio 1965/3/30 ␣

BODY

書式

```
BODY [/H] [/N] [/S 〈開始行〉] [/E 〈終了行〉] _  
    [ 〈ファイルスペック〉 ]
```

機能

入力の一部を切り取って出力します。

解説

入出力は通常標準入出力が用いられますが、入力については〈ファイルスペック〉による、1つまたは複数のファイルの指定が可能です。

出力には入力の先頭行からの行番号、ファイル名がつきます。

出力の開始行、終了行は以下のスイッチにより指定します。

“/S” 入力の切り出し開始を〈開始行〉で指定します。

“/E” 切り出し終了行を〈終了行〉で指定します。

以上2つのデフォルトはそれぞれ、入力の先頭、末尾行となります。

“/N”を指定すると、行番号、ファイル名等の出力が省かれ、行本体のみの出力となります。

例

```
A>body test.c /s100 /e120 ↵  
TEST.C:  
100: VOID fnk1(arg1,arg2,arg3)  
101:      int arg1,arg2,arg3;  
  
120: } /*end of fnk1 */  
A>
```

BSAVE

書式

BSAVE [/H] [<ファイル名1> [<ファイル名2>]

機能

HEX形式のファイルをバイナリファイルに変換します。

解説

アセンブラ、リンカー等により生成したHEX形式のファイルを入力することにより、BASICの“BLOAD”命令でロード、実行が可能なバイナリファイルを生成します。これにより、従来の煩雑なファイル操作なしに、マシン語のルーチンがBASICの環境下で実行可能となります。

入力には必ずディスク上のHEXファイルを指定し、出力は標準出力または<ファイル名2>の指定によりディスクファイルへの出力も可能です。

入力にHEX形式以外のファイルを指定すると、“<ファイル名> is not a hex file”のエラーとなります。

例

```
A>BSAVE TEST.HEX >TEST.BIN ↵
```

```
complete
```

```
A>
```

CAL

書式

CAL [/H] _ [/Y <y y y y> | <[y y y y/] mm>]

機能

カレンダーを出力します。

解説

出力は標準出力が用いられ、通常画面表示となります。

以下オプションの指定とその機能について解説します。

- ① 月を指定することにより、指定の月とその前後1ヶ月のカレンダーを出力します。
月の指定は <y y y y / mm> で年、月をスラッシュで区切って行います。
各々のデフォルトは当年当月で省略可能ですが、月のみの省略は出来ません。
- ② “/Y” スイッチで年を指定した場合、<y y y y> 年の1年分のカレンダーを出力します。
年の指定は上2桁を省略出来ます。

例

A>CAL 2 ↵

Jan 1987						Feb 1987						Mar 1987								
Su	M	Tu	W	Th	F	Sa	Su	M	Tu	W	Th	F	Sa	Su	M	Tu	W	Th	F	Sa
				1	2	3	1	2	3	4	5	6	7	1	2	3	4	5	6	7
4	5	6	7	8	9	10	8	9	10	11	12	13	14	8	9	10	11	12	13	14

CALC

書式

CALC [/H] _ [〈演算式〉]

機能

簡易電卓機能を提供します。

解説

四則演算を通常の演算書式による計算式の入力によりおこなうことができます。

CALCコマンドでは演算式の入力に二つの方法をサポートしています。

1) コマンド行で式を指定しない場合 (連続計算可)

“CALC”と入力しコマンドを起動したのち、計算したい数式を入力しリターンキーをおすと改行し計算結果を表示するとともに次の計算式の入力待ちの状態となり以下CTRL-C、CTRL-Z、“Q”または“q”によりコマンドをぬけるまで連続して計算をおこなうことができます。

2) コマンド行で式を指定する場合

コマンド行で計算式を入力することも可能です。但しこの場合1回の計算でコマンドを終了します。

CALCコマンドでは、十進十二桁の精度での浮動小数点演算をサポートし、 $-9.999999999999E+99$ ～ $9.999999999999E+99$ の範囲の数値を扱うことができます。

例

```
A>CALC 12+1222.2+345/2、
```

```
1406.7
```

```
A>
```

CHKDSK

書式

CHKDSK [/H] [/F] [/M] _ [<ドライブ名 >]

機能

ディスク上のファイルの状況、ディスクの残り容量および、オプションによりメディアに関する情報を表示します。

解説

<ドライブ名> で指定したドライブ (省略はデフォルトドライブ) のディスクをチェックし、以下の内容について報告します。

- ・ 全ディスク容量
- ・ ディスク上のファイル数とトータルファイルサイズ
- ・ 使用可能ディスク容量
- ・ ディスク上のファイル状況に誤りがあると、それを表示します。

つぎに示すエラーは、“/F” スイッチを指定した場合、自動的に修復されます。

```
<ドライブ名> has invalid cluster, file truncated
```

つぎのエラーは、ユーザーが処理を選択してください。

```
<XXXX> lost clusters found in <YYY> chains
```

“/F” スイッチが指定されていると、

```
Convert lost chains to files (Y/N)
```

と表示されます。

この表示に対して、“Y” を選択した場合には
ファイル名 FILEnnnn. CHK を作成し、次のような表示をします。

```
<nnnn> bytes disk space freed
```

また、“N”を選択した場合には、

```
(nnnn) bytes disk space would be freed
```

と表示されます。

```
(ファイル名) is cross linked on cluster (nnnn)
```

と表示されたら、それぞれ必要なファイルのコピーを作成した後、重複してリンクしている元のファイルを消去してください。

さらに“/M”スイッチにより

- ・ クラスタサイズ及び数
- ・ セクタサイズ及び数
- ・ ディレクトリエントリーの数

などのメディアに関する詳細を表示します。

例

```
A>chkdsk b: /m
512 bytes per sector
1* first FAT sector
3 sectors per FAT
2 copies of FAT

112* first directory sector
112 directory entries

1024* first data sector
1024 bytes per cluster

713 total clusters

713K total disk space
683K in 79 user files
75K available disk space
```

A>

CLS

書式

CLS [/H]

機能

スクリーン画面をクリアします。

解説

CLSコマンドでは標準出力にCtrl-L (0CH) コードを送ることにより画面をクリアします。そのためリダイレクトによりファイルに0CHコードを送ることも可能です。

DISKCOPY

書式

DISKCOPY [/H] [/V] [/C] _ [<ドライブ名1>] _
[<ドライブ名2>]

機能

ディスク単位でのバックアップコピー及び照合を行います。

解説

<ドライブ名1> で指定されたドライブのディスクの内容をすべて <ドライブ名2> のディスクにコピーします。

DISKCOPY コマンドにより出来た 2 枚のディスクは同一のファイル内容を持ちます。COPY コマンドと違い、不連続のファイルを連続したファイルにコピーすることは出来ません。また、<ドライブ名2> のディスクの元のファイルは、DISKCOPY によってすべて失われてしまいます。

属性の異なるメディア間でのコピーは出来ません。

ディスクドライブ名のデフォルトは 1. 2 ともそれぞれカレントドライブとなります。

- ・ “/V” スイッチを指定するとディスクのコピーではなく照合のみを行います。
- ・ “/C” スイッチを指定するとコピーの実行に続いてディスクの照合を行います。

例

```
A>diskcopy a: b: ↵
Source disk a:
destination disk b:
strike a key when ready ↵
all data and file on destination disk
will be erased!
  O K ? ↵
complete!
copy other disk (Y/N) ?N ↵
A>
```

DUMP

書式

DUMP [/H] [/N] _ [/B (1行のバイト数)] _ [/S (開始バイト)]
_ [/E (終了バイト)] _ [(ファイルスペック)]

機能

ファイルを16進及びキャラクタによりダンプします。

解説

入出力には通常標準入出力が用いられますが、入力には (ファイルスペック) による1つまたは複数のファイルが指定可能です。

ダンプには、16進数とASCIIキャラクターが用いられ、16バイト毎に改行し、標準出力に表示されます。さらに、“/B”スイッチにより行毎のバイト数をユーザーが指定可能です。

“/S” “/E” の各スイッチにより、ダンプの開始バイト、終了バイトをそれぞれ16進数で指定出来ます。デフォルトはファイルの先頭及び末尾です。

“/N” はファイルのアドレス番地及びキャラクターダンプを省き16進数ダンプリストのみを出力します。“/B” オプションとの組合せで1行のバイト数を変更することにより、例えば1行に1バイトずつ出力しリダイレクトによりファイルにするなどの操作が可能です。

例

```
A>DUMP ABC.COM /BF8 ␣
0000 : C3 26 12 11 F9 FF CD 8A 26 21 13 00 39 5E 23 56      .&..y...&!..9^#V
0010 : 21 01 00 CD F5 27 CA 02 02 21 13 00 39 5E 23 56      !...u'...!..9^#V
0020 : 21 02 00 CD 21 28 CA 7C 01 21 01 00 EB 21 08 00      !...!(. |!...!...
.
.
.
00F0 : 39 E5 CD 25 0C D1 D1 22 39                          9..%...9
```

ECHO

書式

ECHO [/H] [〈アークユメント〉]

機能

コマンドアークユメントを標準出力に出力します。

解説

コマンド行で指定したアークユメントをそのまま標準出力に出力します。

例

```
A>echo hello↵  
hello  
A>
```

EXPAND

書式

EXPAND [/H] [/R] [/F] _ [<ファイルスペック>]

機能

スペース・タブの変換をします。

解説

オプションを省略した時、タブをスペースに変換します。

“/R” オプションにより、反対にスペースをタブにまとめることが出来ます。さらに、“/F” オプションをもちいることにより各行頭から最初のキャラクター（スペース、タブ以外の）間のみで変換をおこなうことが可能です。これにより例えば、プログラム中の文字列データ内のスペース、タブが変換されてしまいコンパイル後、実行時に予想外の表示がなされるといったトラブルを防ぐことができます。

書式

```
GREP [/H] [/X] [/C] [/N] [/A] _ "〈文字列〉" _
      [ 〈ファイルスペック〉 ]
```

機能

入力から任意の文字列を検索し、出力します。

解説

入出力は通常標準入出力が用いられますが、入力には〈ファイルスペック〉による1つまたは複数のファイルが指定可能です。

文字列はダブルクォーテーションで囲み必ずファイルスペックの前で指定します。

入力中の検索文字列を含む行を先頭からの行番号をつけて出力します。入力がファイルスペックの場合、ファイル単位でファイル名とともに出力されます。

文字列中で大文字、小文字は区別されません。

GREPコマンドには多くのスイッチがあり、これらを有効に組み合わせることにより、コマンドを効果的に用いることが出来ます。

- ・ “/X” 指定した文字列を含まない行を返します。
- ・ “/C” 指定した文字列を含む行数のみを返します。
- ・ “/N” 行番号、ファイル名を省きます。
- ・ “/A” 大文字、小文字を区別して検索します。

例

```
A>grep "fprintf(" FILE
A:FILE
7:      fprintf(stderr, "%s", MESS1);
8:      fprintf(stderr, "%d line contain ...
      |
      |
21:      fprintf(stderr, "Unknown switch...
      9 Lines contain "fprintf("
A>
```

HEAD

書式

HEAD [/H] [/N] [/L <出力行数>] _ [<ファイルスペック >]

機能

入力の先頭から指定された行を切り取って出力します。

解説

入出力は通常標準入出力が用いられますが、入力については <ファイルスペック > で1つまたは複数のファイルが指定可能です。

入力の先頭行から指定の行数を切り取り、行番号、残り行数、入力がファイルスペックである場合、ファイル名をつけて出力します。この際各行は、コンソール画面幅を超えた部分については出力されません。

出力の内容は以下のスイッチにより変更可能です。

- “/L” 先頭から何行出力するかを <出力行数> で指定します。指定のない場合、先頭から10行分の出力となります。
- “/N” スイッチを指定すると、行番号、ファイル名等すべて省かれ、行本体のみの出力となります。この場合の出力は画面幅と無関係に全て表示されます。

例

```
A>head test.doc ↵
```

```
TEST.DOC:
```

```
1: 1st line
```

```
2: 2nd line
```

```
⋮
```

```
10: 10th line
```

```
A>
```

HELP

書式

HELP <コマンド名>
または、
<コマンド名> /H

機能

ツールコマンドのリファレンスを出力します。

解説

MSX-DOSツールのコマンドすべてについてHELPによって参照することができます。

(但し、HELPコマンドでMSX-DOSの内部コマンドを参照することはできません)

また、各コマンドで“/H” オプションを指定することのよっても同様のヘルプメッセージを表示します。

例

```
A>help key
MSX-DOS TOOLS HELP KEY
Displays or erases function ..
'ON/OFF' ..
...
Usage: key /h .....
```


LIST

書式

LIST [/H] □ (ファイル名)

機能

BASICの中間言語ファイルをソースファイルに変換し出力します。

解説

DOS上でBASICのプログラムリストを見ることが出来ます。

入力ファイルには(ファイル名)により必ずBASICのテキスト(中間言語)ファイルを指定してください。それ以外のファイルを指定すると、

“<ファイル名> is not a basic binary file”
のエラーとなります。

例

```
A>list TEST.BAS ↵
100 CLEAR 1024 , &HD7FF -----
110 FOR L=0 TO -----
      |
150 END
A>
```

書式

LS [/H] [/L] [/A] [/B] [/C] [/D] [/S|/T]
 [/R] [(ファイルスペック) | (ドライブ名)]

機能

ディレクトリの詳細情報を報告します。

解説

オプションを指定しない場合、カレントドライブのディレクトリ下のファイル名をアルファベット順にソートし、スクリーン幅に合わせて改行し表示します。

“/L” オプションにより以下の情報を付加し、各ファイルごとに改行して表示します。

- ・最新更新日時
- ・ファイルサイズ

以下のスイッチにより表示の内容、形式を変更することが出来ます。

- ・ “/A” 一行に一ファイルずつファイル名とファイル属性 を表示
- ・ “/B” “ ファイルサイズ ”
- ・ “/D” “ ファイルの日付 ”
- ・ “/C” 表示の形式を一行に一ファイルにする

ファイルの表示順序は以下のスイッチにより変更出来ます。

- ・ “/T” 最新更新日の新しい順に表示
- ・ “/S” ファイルサイズの大きい順に表示
- ・ “/R” 上記2つのスイッチとそれぞれ組み合わせることにより、各々逆に表示します。単独の場合、アルファベット順の逆順となります。

尚、“/T” “/S” が共に指定された場合 “/T” のみが有効となり日付順に表示されます。

例

A>ls /b/d/t b: 』 (日付の新しい順にソート)

ATEXT.DOC 1987-01-23 12:30 232345Bytes

BTEXT.DOC 1987-01-22 10:00 23358Bytes

CTEXT.DOC 1986-12-24 11:30 345112Bytes

MENU

書式

MENU [/H] [/T|/S] [/R] _ [<ファイルスペック> | <ドライブ名>]

機能

ディスク上のファイルについて、連続してデリート、リネーム、タイプ、ダンプなどの処理を行うことが可能です。

解説

“MENU” と入力すると、カレントドライブのディスク上のファイル名が、アルファベット順にプロンプト表示されます。下記のサブコマンドにより各ファイルにつきコピー、リネーム等様々な処理を行うことができます。

またカレントドライブについてのみでなくドライブ名やファイルスペックの指定により他のドライブや特定ファイルの指定が可能です。

以下各コマンド操作について解説します。

- ・ ESC or “Q”, ^Z or ^C : コマンド処理の中止
- ・ SPACE or ␣ : 現在表示されているファイルについて何も処理しません。次のファイルについて処理命令待ちの状態になります。
- ・ “E” : 表示中のファイルを消去します。
- ・ “R” : 表示中のファイルをリネームします。新しいファイルネームをひき続き入力します。
- ・ “T” : ファイルの先頭の10行をタイプアウトします。
- ・ “D” : ファイルの先頭から1画面分のダンプリストを表示します。
- ・ “C” : ファイルを他のファイルにコピーします。“C”を入力後、ひき続きコピー先の<ファイル名>を入力します。
- ・ BS : 1つ前のファイルにプロンプトが戻ります。

“T” 及び “D” を実行した場合、プロンプトのファイル名は実行前のものがひき続き表示されます。従って、ファイルの内容を確認した後、デリート、等の処理を行うことが可能です。

ファイルの表示順序は以下のスイッチにより変更できます。

“/T” ファイルの日付の新しい順に表示

“/S” ファイルサイズの大きい順に表示

“/R” “/S” “/T” と組合せてそれぞれの逆に表示、単独ではアルファベットの逆順に表示します。

“/T” “/S” を同時に指定した場合 “/S” は無視され日付順に表示されます。

例

```
A>menu ↵  
COMMAND.COM: ↵  
MSXDOS.SYS: ↵  
TEST.C:Rename  
NEW FILE NAME :TEST1.C ↵  
Complete!  
TEST2.C: Quit  
A>
```

MORE

書式

MORE [/H] [/P <行数>] _ [<ファイルスペック>]

機能

標準入力から標準出力へ1画面分ずつ区切りながら表示します。

解説

標準入力からの、または指定のファイルの内容を標準出力（通常はCRTディスプレイ）に1画面分表示します。入力がさらに続く場合、画面最下部に

“====MORE?====”

と表示されキー入力待ちの状態となります。

さらに、スペースキーをおすことにより、次の1画面分、リターンキーをおすことにより次の1行が表示され、再度キー入力待ちとなります。

以下、入力（ファイルまたは標準入力）の終わりまで同様の操作により表示することが出来ます。

また、

“/P <行数>”により1画面の出力行数を任意にかえることが出来ます。

“Q”（Quit）キーを押すことによりコマンドは中止されます。

“S”キーにより、現在表示中のファイルから次のファイルにスキップして出力します。入力が標準入力の場合や1ファイルのみ指定の時などはコマンドの終了となります。

例

A> MORE TEST.DOC ↵

THIS IS TEST FILE FOR MORE COMMAND

==== MORE ?====

この状態でスペースKeyをおすことにより
出力が再開されます。

PATCH

書式

PATCH [/H] [/S <ダンプ開始バイト>] _ <ファイル名>

機能

バイナリファイルを更新します。

解説

コマンド行の入力により指定のファイルがスクリーンに16進で表示され、パッチモード（ファイル更新モード）となります。カーソルを自由に動かしてスクリーン上の数値を変更することにより、指定のバイナリファイルを更新することができます。ダンプ開始バイトは“/S”オプションにより4桁以内の16進数で指定可能です。指定しない場合ファイルの先頭（0バイト）からとなります。

さらにESCキーによりコマンドモードに入り、ファイルの任意のバイトを指定し更新することが可能です。以下各モードで使用できるキー、コマンドについて解説します。

パッチモード

“A~F” “1~9” の各キー

更新したいバイトの16進表示上にカーソルを移動し各キーを入力することによりメモリー上のファイルイメージを変更します。

“P”、“N”

それぞれ1画面分前（Pキー）、後（Nキー）へダンプ表示を進めます。

カーソルキー

カーソルを任意のバイト上に移動します。

スペースキー

カーソルを右に一つ移動。

バックスペースキー

カーソルを左に一つ移動。

ESCキー または、“\$”

コマンドモードへ移行。

コマンドモード

“P” ↓

パッチモードに戻ります。

“X” ↓

更新したメモリー上のファイルをセーブしてコマンドを終了します。

“Q” ↓

更新したメモリー上のファイルをセーブせずにコマンドを終了します。

<ダンプ開始バイト> の指定

ダンプ開始バイトを16進数4桁以内で指定しリターンキーを押すことにより指定のバイトからダンプ表示されパッチモードに戻ります。

SORT

書式

SORT [/H] [/A] [/R] [/S <桁数>] _ [<ファイルスペック >]

機能

文字列データをソート（並べ替え）します。

解説

入出力は通常標準入出力が用いられますが、入力については <ファイルスペック> でファイルの指定が可能です。

入力されたアスキーコードのデータを各行の一文字目からを対象に判断し、各行を特殊文字 1. 2. 3. ..., A. B. C. ..., ア. イ. ウ. エ, 順に並べ替えをします。アルファベットはアルファベット順にソートされた後、大文字、小文字の順に出力されます。

“/R” オプションをつけることにより上記の逆順にソートします。

“/S” オプションでは各行の何桁目からを判断の対象とするかを桁数で指定します。TABは相当数のスペースとして数えます。

2つのオプションは同時に指定出来ます。

“/A” オプションによりアスキーコード順のソートをします。

“/R” と “/A” を同時に指定した場合、アスキーコード順の逆ソートとなります。

例

```
A>sort test.dat
1 2 3
4 5 6
A B C
a b c
B B B
b b b
ア イ ウ
エ オ カ
A>
```

TAIL

書式

TAIL [/H] [/N] [/S <開始行> | /L <出力行数>]
_ [<ファイルスペック >]

機能

入力の末尾から指定された行を切り取って出力します。

解説

入出力は通常標準入出力が用いられますが、入力については <ファイルスペック> で1つまたは複数のファイルが指定可能です。

入力の末尾から指定の長さを切り取り、行番号、入力がファイルの場合、ファイル名をつけて出力します。出力の開始位置は以下のスイッチにより指定します。

“/S” 出力の開始行を <開始行> で指定します。

“/L” 末尾の何行を指定するかを <出力行数> で指定します。

以上どちらも用いない場合、末尾10行の出力となります。

“/N” スイッチを指定すると、行番号、ファイル名が省かれ、行本体のみの出力となります。

例

```
A>tail b:test.doc ↵
```

```
B:TEST.DOC
```

```
1225: line 1225
```

```
⋮
```

```
1234: line 1234
```

```
A>
```

書式

```
TR [/H] [/W] [/D] [/S] [/I] [/C] [/N] _
    "〈文字列1〉" _ [ "〈文字列2〉" ] _ [〈ファイルスペック〉]
```

機能

文字、文字列の置き換えを行います。

解説

TRコマンドでは、入出力は標準入出力ですが、〈ファイルスペック〉によりひとつまたは複数のファイルを指定可能です。また、“/W”スイッチにより、2種類の置き換えが選択可能です。“/W”を指定しない場合、文字単位の置き換えを行います。〈文字列1〉に含まれる文字が入力中に見つかると、その文字に対応する〈文字列2〉の文字に置き換えます。

また、“/W”スイッチを指定しない時にのみ、“/C”スイッチが使用出来ます。“/C”スイッチは〈文字列1〉に含まれない入力中の文字を〈文字列2〉に置き換えます。

“/W”を指定した場合、単語単位での置き換えを行います。〈文字列1〉と完全に一致する文字列が入力中に検出されると、〈文字列2〉に置き換え出力します。

①文字列の指定について

- ・文字列はダブルコーテーションで囲みます。
- ・〈文字列1〉は省略出来ません。また“/D”スイッチ指定時には〈文字列2〉は指定できません。
- ・文字列は255文字以下で指定し途中でヌルキャラクターを含んではいけません。
- ・次の三つのキャラクターは文字列中で機能コードとして特別の意味をもちます。

‘-’ (ハイフン)

二つの文字または数字を‘-’でつなぐことにより文字または数字の範囲の指定となります。

例 “I-N”と指定すると“I J K L M N”の指定と同等となります。

‘*’ (アスタリスク)

文字*N (Nは数字)で同一文字N個の文字列を意味します。

例 “Z*5”は“Z Z Z Z Z”と同等となります。

‘¥’ (円記号)

タブやキャリッジリターンなどの特殊コードやハイフン、アスタリスクなどを文字列の要素として指定したい場合‘¥’の後に以下のような指定をす

ることにより可能となります。

- “¥n” : ラインフィード (0Ah)
- “¥r” : キャリッジリターン (0Dh)
- “¥t” : タブコード (09h)
- “¥x (16進2桁)” : ‘x’ の後に2桁の16進数でコードを指定
- “¥ (8進3桁)” : 3桁の8進数で指定 例. “¥102” は “A”
- “¥ (文字)” : 円記号の後の文字はそのまま文字列の要素となります。たとえば “¥¥” のように円記号を文字列に指定可能です。

② オプションスイッチについて

TRコマンドには以下のようなオプションスイッチが指定可能です。

- “/H” ヘルプメッセージを出力します。
- “/W” 単語単位での置換 (デフォルトは文字単位) をします。
- “/C” (文字列1) に含まれない文字を置換します。(“/W” 指定時は無効)
- “/N” ファイル名を出力から省きます。
- “/D” (文字列1) の文字または語を入力中から削除して出力します。
- “/S” 同一の置き換えが連続する場合、1回の置き換えにまとめて出力します。
(“/W”、“/D” のどちらかが指定されている場合 “/S” は無効)
- “/I” 複数の連続したスペース、タブを1スペースとしてマッチングを行います。

③ エラーメッセージ、警告メッセージ

TRコマンドでは文字列、オプションの指定が多様であるため発生する様々なエラーにユーザーが適切に対応可能なよう以下のようなメッセージを出力します。

・エラーメッセージ

コマンド処理の続行不可能なエラーであり処理を中止します。

“Illegal use of ‘-’ ”

文字列指定のための ‘-’ の使用法に誤りがあります。(‘-’ の前後に文字がなかったり同じ文字である、など)

“Illegal use of ‘*’ ”

‘*’ の前に文字がない、後に数字がないなどのエラー。

“String2 not necessary”

“/D” 指定時に、文字列2は指定出来ません。

"Missing string#"

String#の指定がありません。(＃は1または2)

・警告メッセージ

メッセージの後何らかのコマンド処理を実行します。

"Warning: multi definition '?'"

キャラクタ?が文字列中に2回以上指定されています('?' は実際指定された文字が表示されます)。最初の出現時の対応で置き換えをおこないます。

"Warning: string2 is longer
Rest of string2 is ignored"

単語単位の置き換えのとき文字列2が長く文字列1中に対応する文字が存在しません。文字列2の長すぎる部分を無視して置き換えを行います。

"Warning: string1 is longer
Rest of string1 will deleted"

単語単位の置き換えで文字列1が長すぎ文字列2中に対応する文字が存在しません。文字列1の長すぎる部分に対応する入力中の文字は削除されます。

例

A> TR "a-z" "A-Z" text.c 

(ファイルTEXT. Cのアルファベットの小文字を大文字に置き換えコンソールに出力します。)

A>TR /W /N "printf(" "fprintf(stderr" CTEXT.C >CTEXT1.C 

(置き換えの結果をファイルにリダイレクトする場合)

UNIQ

書式

UNIQ [/H] [/N] [/C] [<ファイルスペック >]

機能

入力中の隣接する2行を比較し、それが同じである場合、2行目を出力しません。

解説

入出力は標準入出力が用いられます。

UNIQの機能はSORTとの組合せで特に有効となります。SORTの結果をUNIQフィルターにかける事により、同一行の連続出力を避けることができ、見易い出力になるでしょう。

“/C” スイッチにより行頭にその行の出現回数が表示されます。

“/N” スイッチによりファイル名、行の出現回数を省きます。

例

```
A>sort <file1 >file2
```

```
A>uniq <file2 >list1
```

(以上のオペレーションによりfile1の各行が辞書順にソートされ同一内容の行が重複しないリストがlist1としてディスク上につくられました。)

VIEW

書式

VIEW [/H] _ (ファイル名)

機能

ファイルの内容を画面表示し、カーソルキー、その他のキーにより画面外も自由に見ることができます。

解説

“VIEW” コマンドを実行すると、(ファイル名) で指定されたファイルが先頭から1画面分表示されます。以下カーソルキーの操作により、画面を上下自由にスクロールすることができ、1画面に納まらないファイルの内容をすべて見る事が可能です。

カーソルアップキー	5行上を表示
ダウンキー	5行下を表示
ライトキー	8桁右を表示
レフトキー	8桁左を表示

カーソルキーの他に

“P”	前ページを表示
“N”	次ページを表示
“H”	ファイルの先頭を表示
“T”	ファイルの末尾を表示

の各キーにより表示部分を自由に移動できます。

また、“?” キーによりVIEWコマンドのサブコマンドのヘルプが表示されコマンドを起動中に参照することができます。

“Q”、“q”、“^C” 又はESCキーによりVIEWコマンドから抜けられます。

WC

書式

WC [/H] [/P <行数>] _ [<ファイルスペック >]

機能

入力中の語数、行数、ページ数をカウントします。

解説

入出力には通常標準入出力が用いられますが、入力には <ファイルスペック > により、1つまたは複数のファイルの指定が可能です。

[/P <行数 >] により1ページ毎の行数を指定出来ます。デフォルトは60行です。

WCコマンドにおいては、文字、数字のブランクを含まない列(但し、“ ” “-” 等が間にあっても連続とみなす)を1ワードとします。

それ以外のものは無視されます。

例

A>wc *.doc

	Pages	Lines	Words	Characters
A.DOC	12	116	441	1543
B.DOC	1	12	25	85
C.DOC	12	110	500	1834

A>

6 エラーメッセージ一覧

ツールコマンドではすべてのエラーメッセージはコンソール画面にのみ出力されます。プリンターやファイルにエラーメッセージが出力されることはありません。

Bad argument (+usage)

アーギュメントの数に誤りがあったり、不要なアーギュメントがコマンドライン中にあった場合このメッセージとともにユーセージを表示しコマンドを終了します。

Bad field specified

BODY、DUMPコマンドで出力の開始行(バイト)の指定が終了行(バイト)より大きい場合このメッセージで処理を終了します。

Bad hexadecimal

16進数の指定に誤りがあります。

例 tr "¥0x" "0a" (xは16進数として不適切なためエラーとなる)

Excessive argument (+usage)

アーギュメントの数が多すぎます。複数のファイル名を '+' または ';' でつながずに指定したような場合このメッセージが表示されます。

例 ls *.c *.com *.rel

File cannot be copied onto itself

MENUコマンドの "C" (コピー) コマンドでコピー先のファイル名がコピー元と同一に指定された場合このメッセージを表示します。コピーはおこなわれずプロンプトのファイル名はそのまま次のコマンド待ちとなります。

<ファイル名> is not a hexfile

BSAVEコマンドでHEXファイルでないファイルを入力したときにこのメッセージでコマンドを終了します。

Illegal combination of switch

オプションスイッチの指定に無効な組合せがあります。その後の処理はコマンドにより異なります。(BODY、MENU、LSの各解説を参照してください)

例 ls /s/t

Invalid month specified (+usage)

CALコマンドで1~12の範囲以外で月の指定があった場合のメッセージ。

Invalid character in '/x' (+usage)

オプションスイッチ中に不要なパラメータがあります。

例えば '/H' オプションなどのようにパラメータの不要なオプションにおいて '/H1' などのように無効な文字や数字が指定されている場合ユーセージとともにこのメッセージを表示しコマンドを終了します。

Invalid key number (+usage)

KEYコマンドにおいて1~10以外のファンクションキーナンバーの指定があった場合このメッセージでコマンドを終了します。複数のキーの指定のなかでひとつでも誤りがあった場合このエラーとなり処理を中止します。

Invalid minute or second specified (+usage)

SLEEPコマンドで60以上の分または秒の指定があったときにこのエラーとなります。ただし、時のパラメータ省略時の分、時及び分のパラメータ省略時の秒の指定には60を超えた値も許されます。

missing [[hh] mm] ss (+usage)

SLEEPコマンドの時刻指定（/A指定時）で時、分のパラメータが共に省略されたときこのエラーとなります。

'x' is not digit (+usage)

数字でパラメータを指定しなければならないオプションスイッチで数字以外を指定した場合このエラーメッセージが表示されます。

例 HEAD /S12FG

の場合 “'F' is not digit” と表示されます。

Missing parameter (+usage)

HEADコマンドの '/L' オプションのように必ずパラメータが必要なところで指定がないとこのメッセージでエラー終了となります。

例 HEAD/L では “Missing parameter in '/L' ” となります。

Not enough memory for buffer

コマンドの実行に十分なメモリーを確保できなかった場合にこのメッセージを表示しコマンドを終了します。

File not found: <ファイル名>

指定のファイルがディレクトリ中に見つかりません。

Rename error

MENUコマンドの 'R'（リネーム）コマンドにおいて指定された新しいファイル名がすでにディスク上に存在するファイルと同一である場合にリネームエラーとなりプロンプトのファイル名はそのまま次のコマンド入力持ちとなります。

String too long (+usage)

TRコマンドで255文字を超える文字列の指定があった場合、またはKEYコマンド

のファンクションキーへの設定文字列が15文字を超えた場合このメッセージを表示しコマンドの実行を終了します。

Syntax error (+usage)

KEYコマンドで文字列設定の書式にまちがいがある場合のエラーメッセージです。

例 KEY F1. DIR

ではファンクションキーナンバーと設定文字列の区切りにカンマでなくピリオドが使われているためこのエラーとなります。

Unknown switch (+usage)

無効なオプションスイッチが指定された場合のエラーメッセージです。

例 WC /K

では、WCコマンドのオプションに/Kは存在しないためこのエラーとなりコマンドを終了します。

付録. MSX-DOS TOOLS 使用サンプル集

MSX-TOOLSの、2つ以上のコマンドを組み合わせた例を、サンプルとしてここに挙げます。

ここでは、あらかじめフロッピーに入っている、
sample1.doc
sample2.doc
sample3.doc
および、MSX-DOS立ち上げ時に使われる、
welcome.doc
が、必要です。

また、ここで使用するフロッピーは、必ず、マスターからコピーしたものを、用いて下さい。また、ライトプロテクトノッチはoff（書き込み可能）としておいて下さい。2つ以上のTOOLSのコマンドを用いた場合、コマンドからコマンドヘッダを引き渡すのに、中間ファイルを作ります。従って、ライトプロテクトスイッチをon（書き込み不可）としておくと、中間ファイルが作れずにエラーとなってしまいます。

まず、サンプルとして付いてくるファイルの内容を見てみましょう。

```
A>type sample1.doc
```

と、タイプして見て下さい。

```
4  
2  
7  
5  
8  
1  
3  
6
```

```
A>
```

と、表示されると思います。では、このファイルをソートしてみましょう。

```
A>sort sample1.doc
```

```
1  
2  
3  
4  
5  
6  
7  
8
```

```
A>
```

では、もうちょっと複雑なことをしましょう。
いまソートしましたが、たとえばソートの結果のうち、最初から3つめまでを
表示させるには、次の様にします。

```
A>sort sample1.doc | head /L3
```

```
1: 1  
2: 2  
3: 3
```

```
A>
```

ここで出てきたheadコマンド（ファイルの先頭（はじめ）から、指定行だけ表示する。）について、もうすこし説明しますと、
まず、

```
A>dir *.doc
```

で、拡張子が「doc」のファイルを見てみます。おそらく、

```
WELCOME.DOC  
SAMPLE1.DOC  
SAMPLE2.DOC  
SAMPLE3.DOC
```

の4つだと思います。そこで、たとえば

```
A>head /L3 welcome.doc
```

とすると、

```
1:
2: ようこそ MSX-DOS/TOOLS の せかいへ
3:
```

のように表示されます。これは、このファイルのうへ（最初）から3行を表示したものです。では、つぎに、

```
A>head /L5 *.doc
```

としますと、これはこのフロッピー上の（この場合はAドライブの）全てのファイルのうち、拡張子にdocを持つもののファイルの初めから5行を、順に表示します。

```
welcome.doc
```

```
1:
2: ようこそ MSX-DOS/TOOLS の せかいへ
3:
4: このたびは、MSX-TOOLSを おかいあげ
5: いただき まことに ありがとう ございます。
```

```
sample1.doc
```

```
1: 4
2: 2
3: 7
4: 5
5: 8
```

```
sample2.doc
```

```
1: 1234567890
2: 2234567890
3: 3234567890
4: 4234567890
5: 5234567890
```

```
sample3.doc
```

```
1: 2222
2: 3333
```

3: 4444
4: 1111
5: 2222

A>

いまは、これだけ（4つのファイルだけ）ですが、もっとファイルが多くなって、1画面に入らずに、スクロールアップして画面の上に消えて行ってしまう場合、

A>head /L5 *.doc | more

とすれば、1画面1画面ごとにスクロールが止まって、見やすくなります。

さて、tail（ファイルの後ろ（おしまい）から指定行だけ表示する。）でも指定の仕方は同じです。

A>tail /L5 *.doc

で、該当するファイル（ここでは、sample1.doc、sample2.doc、sample3.doc、welcome.docの4つのはずです）のファイルの
おわりから5行を順に表示します。

```
welcome.doc
14: と して あります。 これらの コメント"を おつかい
15: の うえ、より たのしい MSX-LIFEを おすすめ
16: し ください。
17:
18: -----ASCII MSXとうかつぶ
```

```
sample1.doc
4: 5
5: 8
6: 1
7: 3
8: 6
```

```
sample2.doc
4: 4234567890
5: 5234567890
6: 6234567890
```

7: 7234567890
8: 8234567890

sample3.doc
7: 3333
8: 6666
9: 1111
10: 8888
11: 7777

A>

これらのhead、tailコマンドは、「どのファイルだったかなー」というときに、非常に役に立ちます。

また、ファイルの最初の方に、ファイルを作った日にちや、バージョン、作者の名前などを書いておけば、headコマンドにより、すぐに見つけ出す事が出来ます。

ここまで読んでこれたあなたなら、次の例も容易に理解できるでしょう。

A>SORT SAMPLE3.DOC | HEAD /L4 | MORE

つぎに、同様にファイルの中を見るコマンドで、DUMPコマンドを使った例をあげましょう。

A>DUMP sample1.doc

```
0000 : 34 0D 0A 32 0D 0A 37 0D 0A 35 0D 0A 38 0D 0A 31 4..2..7..5..8...  
0010 : 0D 0A 33 0D 0A 36 0D 0A 1A ..3..6..
```

A>

このように、ファイルの内容を16進とアスキーコードで表示します。上の例では、34が数字の4、0Dがキャリッジリターン（改行）、0Aがラインフィード（1行送り）...となっています。1Aはファイルの終わりを表します。そのため、つぎのバイトから先は表示されません。先ほどのHEADコマンドと組み合わせると、下の様になります。

A>DUMP sample1.doc | HEAD /L1

sample1.doc

1: 0000 : 34 0D 0A 32 0D 0A 37 0D 0A 35 0D 0A 38 0D 0A 31 4..2..7..5..8..1

A>

最後に、M80、L80、BASICを組み合わせ使用した例を示します。

ここでは、

```
SEARCH. MAC , M80. COM , LIST. COM
SEARCH. BAT , L80. COM , BSAVE. COM
SEARCH. BAS
```

の7つのファイルが必要となります。

これは、SEARCH. BATと言うバッチファイルを使って実行するようになっています。まず、使用するファイルの内容を見てみましょう。

```
A>TYPE SEARCH. BAT      動作の説明 (これはファイルには入ってません)
m80 =search             search.macをアセンブルする
180 /p:d000,search,search/n/x/e  アセンブルした結果をリンカーに通す
bsave search.hex >search.bin   リンカーの出力をバイナリーファイルにする
basic search.bas          basic上でsearch.basを実行する
```

A>

最初の3行で、SEARCH. BINが作られ、basicプログラムのSEARCH. BASの中で、これがメモリーにロードされ、使われます。ですから、L80. COMを実行する際に、&hD000hからロードするように指定しています。(/p:d000の部分)

つぎに、SEARCH. BASの内容を見てみましょう。このSEARCH. BASは、BASIC上の中間言語ファイルとなっていますので、このままTYPEコマンドで見ることは出来ません。そこで、LISTコマンドを使って、内容を見ます。(ちなみに、中間言語ファイルの方が、ディスク上のスペースを取りません。)

A>LIST SEARCH. BAS

```
110 CLEAR 300,&HCFFF: BLOAD"SEARCH.BIN": DEF USR=&HD000
120 DIM A$(100): N=0: I=0: A$(N)=MID$(USR(CHR$(0)+"?????????DOC"),2,8)
130 IF A$(N)<>" " THEN N=N+1: A$(N)=MID$(USR(" " ),2,8):GOTO 130
140 IF A$(I)=" " THEN BEEP: END
```

```

150 GOSUB 170
160 I=I+1: GOTO 140
170 '
180 B$=A$(I)+".DOC": OPEN B$ FOR INPUT AS #1
190 PRINT SPC(8);" ファイル名:";B$
200 IF EOF(1) THEN CLOSE: PRINT SPC(8);B$;" の内容:"; RETURN
210 INPUT #1,C$
220 PRINT C$
230 GOTO 200

```

A>

さて、SEARCH. BASも、説明しておきましょう。このプログラムは、2つの部分に分けられます。それは、110行から160行の部分と、170行から終りまでです。このうち、110行から160行までがSEARCH. BINを使用した部分で、このBASICプログラムの重要な部分です。110行は、機械語のエリアを確保して、(&HD000hから上)そこにSEARCH. BINをロードします。そしてSEARCH. BINの実行開始番地を&HD000とします。

120行では、文字配列A\$を宣言し、SEARCH. BINに"????????DOC" (つまり"*.DOC")の指定をして、ディスクの中から最初に該当するファイル名をA\$(0)に代入しています。またIとNに0を代入して、初期設定をしています。

130行では、ディスクのファイル名の中から、"????????DOC"に該当するファイル名をすべて、文字配列A\$に順次代入します。ファイル名を得るために、SEARCH. BINを繰り返し実行しています。(USR(" ")) SEARCH. BINは" "を指定されると、後続のファイル名を順次返すようになっています。また該当するファイルがない場合には、" " (ヌルストリング)を返すようになっています。

この時点で、文字配列A\$の中は次のようになっています。

```

A$(0) = "WELCOME"
A$(1) = "SAMPLE1"
A$(2) = "SAMPLE2"
A$(3) = "SAMPLE3"
A$(4) = " "

```

140行では、文字配列A\$の中身が" " (ヌルストリング)であれば、このプログラムを終了します。今ここを最初に通るときには、I=0ですから、A\$(0)="WELCOME"となり、先へと進みます。I=4のときに、A\$(4)=" "となり、終了します。

150行は、170行以下のサブルーチンを呼んでいます(170行へいき、RETU

RN文で帰ってくる)

160行では、Iの値を1増やし、140行へと飛びます。140行と160行により、Iの値を0、1、2、...と増やして、つまり文字配列A\$のファイル名をつぎつぎと変えながら実行していくわけです。

170行以降は、まとめて説明しますと、B\$に文字配列A\$からファイル名を代入して、そのファイルをシーケンシャルファイルとしてオープンし、ファイルの内容を画面に表示して、表示する内容が無くなったらRETURN文により150行に帰る、というものです。

従って、120行の"????????DOC"と、170行以降を変える事により、「ディスクのファイル名を得て、それを元にして何かする」というプログラムが簡単に出来るのです。

では、実行です。と言っても、ただ見ているだけで、アセンブルからBASICの実行まで、全部バッチファイルがやってくれます。

お買い上げ頂いたマスターフロッピーからのコピーをお使いください。

A>SEARCH

(SEARCH.BATを実行)

A>m80 =search

No Fatal error(s)

A>180 /p:d000,search,search/n/x/e

MSX.L-80 1.00 01-Apr-85 (C) 1981, 1985 Microsoft

Data D000 D0A7 < 167>

43403 Bytes Free

[0000 D0A7 208]

A>bsave search.hex >search.bin
complete

A>basic search.bas

(ここで画面が消えて、BASICが立ち上がる)
SEARCH.BASを実行する。

どうでしたか。あまりにすいすいと実行してしまうので、よくわからなかったと思います。そのときは、CALL SYSTEMとタイプして、MSX-DOSにもどり、もう一度SEARCH. BATを実行してみてください。また、SEARCH. BASについて、もっと詳しく動作を追って見たい方は、BASIC上でTRONコマンドを実行した上で、RUNしてみてください。SEARCH. BINはもう作られていますので、今度は、

A>BASIC

(BASICが立ち上がる)

LOAD" SEARCH. BAS"

⋮

⋮

⋮

から 始めることができます。

