
★
★ AZAS
★ Reference Manual
★ 1. DOS COMMAND
★
★ MINI-DOS Ver0. B対応

★★ 1. DOS COMMAND ★★
★★ MINI-DOS Version 0. B(16進番号)に対応 ★★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★ 1 — 0. MINI-DOSの立ち上げ方 ★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

MINI-DOS Version 0. Bが立ち上るのは以下のハードです。

本体 PC-8001MK2/SR

PC-8801/MK2/SR

(RAMが64KバイトあるPC-8001でも使えます。

PC-8001Ver1.1とMulti-Cardのシステムで

実動中です。バンク切り替えのポートが異なる場合ソース“boot.aza”
を変更する必要があります。

DISK 5インチ両面倍密度2ドライブ以上必要です。

8インチのサポートはいまのところなし。ただし、エクスパンド・コマン
ドとして8インチ・ドライバーを作れるようになっていきます。

CRT 2000文字以上表示できればカラーでもモノクロでもかまいません。

プリンター なくてもかまいません。一行80文字以上印字できれば使えます。

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★ 1 — 1. AZASのファイル管理 ★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

MINI-DOSはNEC N88/N80/N-DISK-BASICとコンパチブルなディスク・フォーマットを使用しています。なお、つぎのファイル名やファイル・タ

イプはMINI-DOSまたはAZASで特別な意味を持っています。

特別なファイル名

“autoex.bat”……DOSを立ち上げたとき自動的に実行されるバッチ・ファイル。

特別なファイル・タイプ

“.trc”……トランジェント・コマンド(実行時ロード・コマンド)

“.flc”……フィルター・コマンド(実行時ロード・DOS結合コマンド)

“.ovr”……オーバーレイ・コマンド(一時ロード・コマンド)

“.exp”……エクスパンド・コマンド(DOS拡張コマンド)

“.bat”……バッチ・コマンド(コマンド自動入力・コマンド・ファイル)

“.img”……イメージ・ファイル(BSAVE/BLOAD タイプのファイル)

“.fky”……ファンクション・キー・ファイル(DOSのSK/LK コマンドで使用)

AZASでのみ特別なファイル・タイプ(MINI-DOSを他の目的で使用する時は特別でない)

“.aza”……AZAアセンブラーのソース・ファイル

“.bak”……エディターの作るバック・アップ・ファイル

“.crf”……クロス・リファレンス・インフォメーション・ファイル

これらのファイル・タイプは“.crf”以外強制するものではありません。例えばアセンブラのソース・ファイルのファイル・タイプが“.e”でも、ファイル・タイプを省略せずに入力すればアセンブラは受け付けますし、エディターの作るバック・アップ・ファイルのファイル・タイプはコマンド一つで変更できます。

また、イメージ・ファイルが“.img”というのも、単にファイル・タイプを省略したBSAV/E/BLOAD タイプのファイルにそう付けているだけで、“.trc”や”.flc”, “.ovr”などもBSAV/E/BLOAD タイプです。

このように、特別な、と言ってもユーザーが変えられるものが多くあります。ですから、気にいらなければどんどん変えて使ってください。

MINI-DOSには入出力機器をファイルとして扱う機能があります。(リダイレクト)。ディスク・ファイルを含めつぎの機器が標準でサポートされています。

入力機器

I : DISK DRIVE

2: 同上
3: 同上 (拡張)
4: 同上 (〃)
C: Console Input (キーボード入力)

出力機器

1: DISK DRIVE
2: 同上
3: 同上 (拡張)
4: 同上 (〃)
C: Console Output (CRT/テキスト画面出力)
L: Line Printer Output (プリンター出力)

★ ★★★★★★★★★★★★★★★★★★★★★★★★ ★ ★ 1 - 2. DOSのコマンド ★ ★ ★★★★★★★★★★★★★★★★★★★★★★★★ ★

(1-2-1……16.) は DOS 自身が持っているレジデント・コマンドです。コマンドを入力した時点でコマンド・ファイルを DISK からロードしません。

コマンド名は大文字/小文字を区別しません。

1-2-1. ** D ** 【ディレクトリ】

書式 D [n]

ドライブ n のディスクのディレクトリをコンソールへ出力します。

1-2-2. ** D S ** 【詳細なディレクトリ】

書式 D S△[fn]

特定のファイル(群)の詳しい情報をコンソールへ出力します。

ファイル名にワイルド・カードを使用できます。"*, *" とすればすべてのファイルを対象にできます。

ファイル名を省略すると、残りバイト数のみ出力します。

ファイルのアトリビュートとして出力される略記号の意味は以下の通り。ただし、b または i の記号が出力されないファイルはアスキーファイルを意味します。

p: Write Protect

r: Read After Write

b: Binary BASIC Text (中間言語形式 BASIC プログラム)

i: Image File (BSAVE/BLOAD タイプ/あとに表示されるのはアドレス)

1-2-3. ** D R ** 【デフォルト・ドライブの変更】

書式 D R△n

デフォルト・ドライブをドライブ n へ変更します。

1-2-4. ** K I L L ** 【ファイル削除】

書式 K I L L△fn

ディスク・ファイルを削除します。

MINI-DOS Ver. B ではワイルド・カードを使用しても一回に一つのファイルしか削除できません。トランジエント・コマンドの "kill.trc" を使ってください。

1-2-5. ** N A M E ** 【ファイル名の付け替え】

書式 NAME△old-fn, new-fn

ファイル名を付け替えます。

1-2-6. ** T ** 【アスキー・ファイルのタイプアウト】

書式 T△fn

アスキー・ファイルをコンソールへ出力します。

アセンブラーのソースなどの内容を確認するとき使うと、エディターをロードしたりする時間もかからず、便利です。

1-2-7. ** S ** 【メモリー・イメージのセーブ】

書式 S fn (img), start-hex, end-hex [, real-start-hex]

BASICのBSAVEとほぼ同じ機能です。

real-start-hexを省略するとstart-hexと同じとみなされ、start-hexからend-hexまでのメモリー内容がセーブされます。（オフセット無しセーブ）

real-start-hexを省略しない時、スタート・アドレスstart-hexとエンド・アドレスend-hexをファイルに記録し、メイン・メモリーのアドレスreal-start-hexからメモリー内容をセーブします。（オフセット付きセーブ）

1-2-8. ** L ** 【イメージ・ファイルのロード】

書式 L fn (img) [, load-hex]

BASICのBLOADとほぼ同じ機能です。

イメージ・ファイルをメイン・メモリーへロードします。

load-hexを省略すると、記録されたアドレスからロードし、load-hexを指定するとそのアドレスからロードします。

1-2-9. ** SK ** 【ファンクション・キー定義のセーブ】

書式 SK fn (fky)

ファンクション・キーの定義を(f · 1 から f · 10 までまとめて)セーブします。

なおこのコマンドでセーブしたファイルは、つぎのLKコマンド以外ではアクセスできません。（ファイル内容が特殊なフォーマットにまっています）

MINI-DOS Ver.0. Bではファンクション・キーを定義するコマンドがありません。1-2-14. のBコマンドでN-BASICに移行してからkeyステートメントで定義してください。

1-2-10. ** LK ** 【ファンクション・キー定義のロード】

書式 LK fn (fky)

1-2-9. のSKコマンドでセーブしたファイルをロードします。

1-2-11. ** > ** 【コンソール出力リダイレクト】

書式 1) > fn

2) >

1) はコンソールの出力をファイルに出力するよう切り替えます。（そのとき、CRT／テキスト画面にも出力します。）

ファイルは“L:”のようにプリンターでもよいので、プリンターへコンソールの出力をエコーバックさせることもできます。

2) はコンソール出力中のファイルをクローズし、コンソール出力をCRT／テキスト画面に戻します。

1-2-12. ** < ** 【コンソール入力リダイレクト】

書式 < fn

コンソールの入力をファイルから行います。ファイルがEOF（ファイル末／コントロール-Z）になるか、STOPキーによるBreakでファイルよりの入力は終了しキーボードからの入力に戻ります。

1-2-13. ** RESET ** 【I/O機器のリセット】

書式 RESET

システムに異常があるとき入力する。もし、それでもだめならRESETボタンしかない。

このコマンドで実行すること。（実行順）

 フィルター・コマンドのクローズ。

 全ファイル強制クローズ。（出力中ファイルの内容は保証せず）

 ディスク・ユニットのイニシャライズ。

 CRTのイニシャライズ。

 コンソール入出力をキーボード／CRTとする。

1-2-14. ** B ** 【N-BASIC】

書式 B

N-BASICのコマンド待ちに移行します。

この機能のため256バイトのメモリーが割り当てられていて、その範囲内でBASICでできることなら何をやってもかまいません。

だいたい、次のような目的で使用します。

- (1) PRINT文で電卓の代用。(clearと入力して文字領域をすべて演算スタックにすると複雑な式でもout of memoryが出ない)
- (2) ファンクション・キーの定義。
- (3) N-BASIC・romのモニターを使う。
- (4) LPRINT文でプリンターのモード設定をする。

MINI-DOSに戻るには、“OK”のプロンプトに、

SET

と入力して下さい。(SET文でMINI-DOSに戻る)

1-2-15. “BOOT” 【システム・ディスクの立ち上げ】

書式 BOOT

ドライブ1に入っているシステム・ディスクを立ち上げます。

N-BASIC・romのリセット時のブート・ルーチンへ実行を移すだけなので、N-BASICモードで立ち上がらないシステムをブートすることはできません。(そういう時はRESETボタンを使って下さい)

1-2-16. “CUT” 【DOS拡張機能の切り放し】

書式 CUT

最後にリンクしたDOS拡張コマンド(Expand Command)を切り放します。このコマンドによりCUTした拡張コマンドの分だけユーザー・エリア(UCA)が増えます。当然ですが、何も拡張されていない時はエラーとなりユーザー・エリアは増えません。

1-2-17. “……名前無し……” 【コマンド・ファイルの実行】

書式 1) :fn

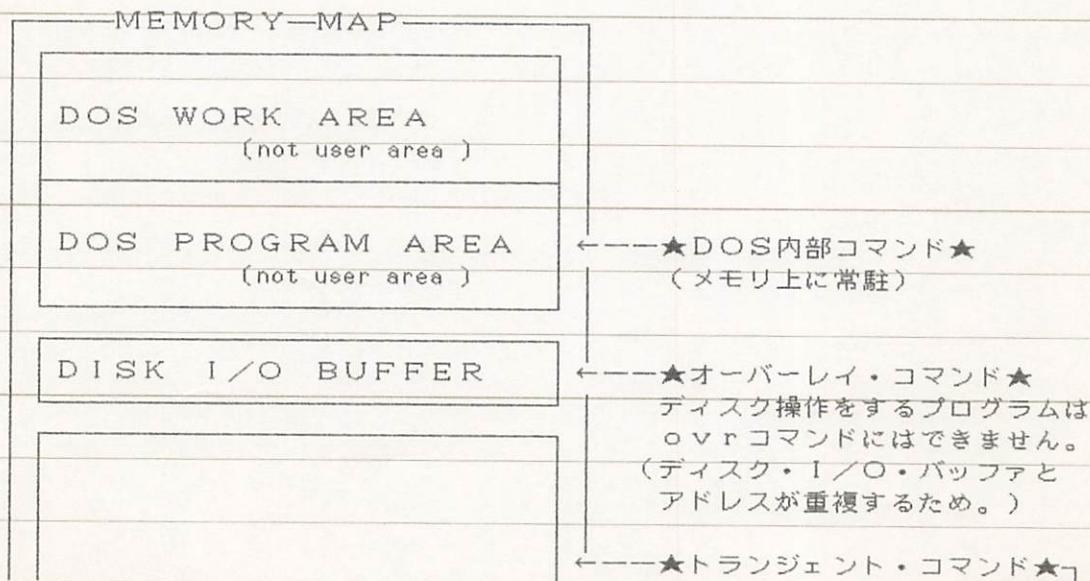
2) n:fn

コマンド・ファイルを実行します。書式1)のようにコマンド・ファイルのあるドライブ番号を省略する時は“:”を付加します。

各コマンドの説明は次のそれぞれの項目(1-2-17-x)にまかせますが、省略したとき暗黙に指定されるファイル・タイプは以下の通りです。

- “.trc” (1-2-17-1) トランジエント・コマンド
- “.flc” (1-2-17-2) フィルター・コマンド
- “.ovr” (1-2-17-3) オーバーレイ・コマンド
- “.exp” (1-2-17-4) エクスパンド・コマンド
- “.bat” (1-2-17-5) バッチ・コマンド

☆☆☆☆☆ それぞれのコマンドのメモリ上配置 ☆☆☆☆☆



USER/COMMAND AREA (UCA: ユーザー／コマンド・エリア)	←—★フィルター・コマンド★—
	これらのコマンドはつねに一つまでしかロードしておくことができません。
	←—●ユーザー・プログラム● MINI-DOS下で実行させるときこのエリアにロードします。

1-2-17-1. 【トランジェント・コマンド】

トランジェント・コマンドは“.trc”というファイル・タイプを持つコマンド・ファイルで、アセンブラーやフォーマット・ユティリティ等があります。

USER/COMMAND兼用エリアにロードされ、すぐに実行されます。実行終了とともにメイン・メモリー上から忘れ去られるため、一度にまとめて仕事をするアセンブラーやコンパイラ等に適しています。

1-2-17-2. 【フィルタ・コマンド】

フィルタ・コマンドは“.flc”というファイル・タイプを持つコマンド・ファイルで、エディタや対話型のファイル転送ユティリティ等があります。

USER/COMMAND兼用エリアにロードされますが、すぐに実行されるのは前処理部分（エディタでいうと、テキストのロード）だけで、すぐにDOSへ戻ります。

このときフィルタ・コマンドとDOSのコマンド入力待ちとはコマンド・フィルターというメカニズムによりリンクされていて、DOSコンド待ち（\$マークのプロンプト）に、フィルタ・コマンドへの命令を入力すると、あたかもDOSのコマンドが拡張されたかのように、フィルタ・コマンドへの命令が実行されます。

この状態は終了処理（エディタでいうと、テキストのセーブとバックアップ）が行われるまで続きます。終了処理はコマンドで実行させる場合と、他のトランジェントまたはフィルタ・コマンドを入力することによる自動終了により実行させる場合の2通りがあります。自動終了が発生するのは、他のコマンドがUCA（ユーザー・コマンド・エリア）を必要とするとDOSが判断したときです。

1-2-17-3. 【オーバーレイ・コマンド】

オーバーレイ・コマンドは“.ovr”というファイル・タイプを持つコマンド・ファイルで、モニターや日付時間の表示設定等があります。

フィルタ・コマンドを自動終了させないので（1-2-17-2. 参照）、エディタ等を使用中でも自由に利用できます。ディスク・I/O・バッファにロードされるためオーバーレイ・コマンドがディスクをアクセスすることはできませんが、サイズが2Kbytes以内なので数秒でディスクよりロードでき、DOS自身が初めから持っているコマンドのような感覚で使用できます。

1-2-17-4. 【エクスパンド・コマンド】

エクスパンド・コマンドは“.exp”というファイル・タイプを持つコマンド・ファイルで、MINI-DOS Ver.0, Bでは先行入力機能やプリンターへ送るTABコードのスペースへの変換機能等があります。

このコマンドはDOSの機能を拡張します。そのため、拡張される機能の分だけUCA（ユーザー・コマンド・エリア）が減少します。

エクスパンド・コマンドはいくつでも同時に使用できますが、ものによっては同時に使用できない組合せもあります。

拡張した機能を切り放すにはCUTコマンド（1-2-16.）を使います。

1-2-17-5. 【バッチ・コマンド】

バッチ・コマンドは“.bat”というファイル・タイプを持つコマンド・ファイルです

あたかもキーボードからコマンドを入力したかのように指定したコマンド列を実行します。

他のコマンド・ファイルと違いアスキイ・ファイル形式になっていてユーザーがエディタを使って簡単に作成することができます。

このバッチ・コマンドは仮引数を指定できません。また、(1-2-12.)のコンソール入力リダイレクトとほぼおなじです。(そちらも参照してください)

DOSがBOOT時に実行する“autoex.bat”ファイルもバッチ・コマンドです。

★ ★★★

★ 1 — 3. ユティリティ・コマンド・ファイル ★

★ ★★★

MINI-DOS Ver 0. Bでは次のコマンド・ファイルが標準的なユティリティとして用意されています。

1-3-1.	“mon.ovr”	簡易モニター
1-3-2.	“time.ovr”	日付時刻の表示設定
1-3-3.	“raw.ovr”	リード・アフター・ライトの設定
1-3-4.	“xf.flc”	ファイル転送
1-3-5.	“attr rtc”	ファイル・アトリビュートの設定
1-3-6.	“format trc”	ディスクケットのフォーマット
1-3-7.	“backup trc”	ディスクケットのバック・アップ
1-3-8.	“kill trc”	複数のファイルの削除
1-3-9.	“senko exp”	先行入力機能拡張
1-3-10.	“lpttab exp”	プリンターTABコード変換

1-3-1. “mon.ovr” 【簡易モニター】

N-BASIC.romのモニターとほぼ同じ機能の簡易モニターです。

: mon

と入力すると起動し、

MON]

というプロンプトが表示されます。

以下のコマンドが使えます。

“D”	(1-3-1-1.) ダンプ・メモリー
“S”	(1-3-1-2.) セット・メモリー
“X”	(1-3-1-3.) レジスタの表示
“G”	(1-3-1-4.) 任意番地から実行
“!”	または [ESC] + キー	(1-3-1-5.) DOSへ戻る。
“E”	(1-3-1-6.) 特定番地から実行
“^ A”	(コントロール・A)	(1-3-1-7.) I.D. の表示

1-3-1-1. “D” 【ダンプ・メモリー】

書式 D [start-hex[, [end-hex]]]

メイン・メモリーのダンプ・リストを出力します。

スタート・アドレス (start-hex) のみ指定の時 128 バイト出力します。

区切りのカンマの後にエンド・アドレスを省略すると、メモリー・エリアの最後まで出力します。

なお、アスキイ・ダンプ部はコンソール出力をリダイレクトしていると、コントロール・キャラクターはスペースに変わります。(プリンター等にコントロール・キャラクターのフォントがないため。4-2-27. “CPRDIR”, 4-2-43. “PUTDIR” 参照)

1-3-1-2. “S” 【セット・メモリー】

書式 S [start-hex]

メモリーの参照と書き換えをします。

N88-BASICのモニターのEコマンドと似ています。

メモリーの値を半バイトずつ変更できます。

“S” モードに入ると 3 行分 (48 バイト) がダンプされ、カーソルは中央の行に来ま

。カーソルが次の行に行くとスクロールが発生します。

このコマンドは、スクロールするとき表示をリフレッシュするため、コンソール出力をCRT以外にリダイレクトすると正常に動作しません。

“S”モードでは次のキーが使えます。

□……9	半バイト書き換え。
A……F	リ
. . . [RETURN] = + *	A … Fに相当。（テンキーだけで16進数が入力可）
カーソル・キー	カーソル移動
[DEL] キー	カーソルを半バイト戻す
[HOME] キー	“S”モードに入った時のアドレスに戻す。
[ESC] キー	その時のアドレスを次のSコマンドのアドレスのデフォルトとして記憶し、“S”モードを終了。
[STOP] キー	“S”モードの終了（デフォルトは変えない）

1-3-1-3. ** X ** 【レジスターの表示】

書式 X (リターン・キー不用)

マシン語のプログラムがRST \$38などで、DOSにブレークした時、DOSは各レジスターの値を保存します。このXコマンドは、そのレジスターの値を表示します。それ以上の機能はありません。

表示されるレジスター名のうちTop, Secはスタック・トップとセカンドで、通常TopはPC(プログラム・カウンター)に当たります。

1-3-1-4. ** G ** 【任意番地から実行】

書式 Gstart-hex

指定したアドレスからマシン語プログラムを実行します。

PC以外のレジスターの値は不定です。

1-3-1-5. ** ! ** または [ESC] キー 【DOSへ戻る】

書式 ! または [ESC] (リターン・キー不用)

モニターを終了して、DOSへ戻ります。

エディターでバッチ・ファイルを作るとき、モニターを終了させるための[ESC]キーが入力できないので“!”を使います。

1-3-1-6. ** E ** 【特定番地から実行】

書式 E

DOS内部のALOMAP(アロケーション・マップ)のEXECUTE_Wの示すアドレスからマシン語プログラムを実行します。4. DOSサービスを参照してください。

※ AZASでは通常何の意味も持ちません。※

1-3-1-7. ** ^ A ** (コントロール・A) 【I.D.の表示】

書式 ^ A (コントロール・A)

モニターのコピーライト・メッセージとバージョン番号を表示します。

1-3-2. ** t i m e . o v r ** 【日付時刻の表示設定】

機能1 …… 日付時刻を表示させたいとき。

: time

機能2 …… 時刻を設定したいとき。(HH時MM分SS秒)

: time△HH:MM:SS

機能3 …… 日付を設定したいとき。(YY年MM月DD日)

: time△YY/MM/DD

機能4 …… 年のみ設定したいとき。(YY年)

: time△@YY

PC-8001/8801シリーズのカレンダー・クロックのバッテリー・バックアップは“年”を保存しません。そのため“autoex.bat”の中で機能4により“年”的設定をする必要があります。年が明けたら“autoex.bat”的 : time△@YYという部分は書き直さなくてはなりません。

1-3-3. “xf. flc” 【リード・アフター・ライトの設定】
このコマンドで設定するのは一つひとつのファイルのリード・アフター・ライト・アトリビュートではなく、全てのDISK書き込みでリード・アフター・ライトをするかどうかのDOS内部のスイッチです。

機能1 ……リード・アフター・ライト・スイッチON
: rawon

機能2 ……リード・アフター・ライト・スイッチOFF
: rawoff

機能3 ……リード・アフター・ライト・スイッチの表示(ONかOFFか)
: raw

1-3-4. “xf. flc” 【ファイル転送】

“xf. flc”はファイル転送を行うフィルタ・コマンドでいくつかの使用法があります。

使用法1 ファイル転送

: xf to-n : = from-n : fn

from-nというドライブのfnというファイル(ワイルド・カード可)をto-nというドライブへ転送します。

例 : xf 2 : = 1 : * . aza

ドライブ1のファイル・タイプ“.aza”的ファイルを全てドライブ2へ転送します。

使用法2 ファイル複写

: xf to-fn = from-fn

from-fnというファイルの複製をto-fnというファイル名で作ります。もちろん、from-fnとto-fnが同じドライブ上でも異なるドライブ上でもかまいません。

例 : xf 2 : new. aza = 2 : old. aza

ドライブ2の“old.aza”というファイルの複製をドライブ2に“new.aza”という名前で作ります。

使用法3 対話式

: xf

と入力すると、*xf readyというプロンプトを表示してDOSのコマンド待ちに戻ります。(当然ここでDOSのコマンドも実行できます)。そこで以下のコマンドをつかって転送や複写をします。

#to-n=from-n : fn ファイル転送(使用法1と同様)

#to-fn = from-fn ファイル複写(使用法2と同様)

/A AskモードON

/N AskモードOFF

HELP コマンドHELP

! このコマンドを終了させる。

AskモードをONにすると、ファイル一つずつに転送するかどうかの確認(Y/N)をとります。例えば、“#2 : = 1 : *.*”のように入力して転送したいファイルには“Y”を押し、したくないファイルには“N”を押せば、いちいちファイル名を入力しなくても転送したいファイルだけ転送できます。

1-3-5. “attr. trc”

【ファイル・アトリビュートの設定】

機能1 ……ファイルにライト・プロテクト・アトリビュートを設定する。
: attr fn, p

機能2 ……ファイルにリード・アフター・ライト・アトリビュートを設定する。
: attr fn, r

機能3 ……ファイルに設定されたライト・プロテクトまたはリード・アフター・ライトのアトリビュートを解除する。
: attr fn

ファイル名の指定にワイルド・カードが使用できます。

ファイルに設定されたアトリビュートはDOSの“DS”コマンド(1-2-2.)で調べることができます。

1-3-6. "format. trc"

【ディスクケットのフォーマット】

物理的フォーマットと論理的フォーマットを行います。

このコマンドでサポートされている機能は以下の通りです。

機能1 …… MINI-DOSのシステム・ディスクをフォーマットする。

機能2 …… データ・ディスクをフォーマットする。

(MINI-DOSとN/N80/N88-BASIC共通)

機能3 …… ユーザーが書き変えたBOOT-LOADERをディスクに書きこむ。
(フォーマットはしません)

1-3-7. "backup. trc"

【ディスクケットのバック・アップ】

ディスクケットのバック・アップをします。

: backup

と入力しメッセージが出たら、ドライブ1に元になるディスクケットをいれ、ドライブ2に新しいディスクケットをいれ、“Y”をキー・インします。

次に、新しいディスクケットに物理的なフォーマットが必要なら“Y”、必要でないなら“N”をキー・インします。

これで、バック・アップを始めます。未使用のトラックは自動的にスキップします。

※ コマンドを起動する時に、

: backup△/a

と入力すると、全てのトラックをコピーします。

1-3-8. "kill. trc" 【複数ファイルの削除】

DOSのKILLコマンドにワイルド・カードを使用できるようにしたものです。

: kill△fn

と入力します。該当するファイル一つひとつに確認(Y/N)をとってきます。また、ライト・プロテクト・アトリビュートの付いたファイルはスキップします。

1-3-9. "senko. exp" 【先行入力機能拡張】

DOSに先行入力機能を拡張します。

: senko

と入力すると、DOSが拡張され、

senon

とDOSのコマンド待ちで入力すると、先行入力ができるようになります。このとき、ファンクション・キー・ウインドウの左に“+”マークが表示され、先行入力状態になっていることを示します。

先行入力バッファは30文字分あり、バッファの最後の7文字がファンクション・キー・ウインドウの右に表示されます。

先行入力をやめるには、

senoff

と入力するか、DOSの“CUT”コマンドで“senko. exp”そのものをDOSから切り放して下さい。

※ “senko. exp”がDOSに拡張されていると、1行入力用のスクリーン・エディターが次のように改良されます。(もともとはN-BASICのものと同じ)

☆ キーのリピートがはやくなる。

☆ [INS] キーはインサート・モードの切り替えになる。

☆ ^F (コントロール・F) で^N (コントロール・N)と同じことができる。

1-3-10. "ipttab. exp"

【プリンターTABコード変換】

使用法

: ipttab

と入力します。

このDOSの拡張機能は、プリンターに送るTABコードをスペースに変換してからプ

プリンターに出力します。この機能により、TABコードを受け付けないプリンターでも正常にプリントできます。

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★ 1 — 4 . BOO T時のコマンドの自動実行機能 ★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

"autoex.bat" というバッチ・コマンド・ファイルがBOOTしたときドライブ1にあれば、DOSはそれを自動的に実行します。この機能により自分用に設定されたDOSをBOOT後ただちに使えたり、特定のMINI-DOSアプリケーションをRESETボタンだけで立ち上げるバカチョン的な使いかたもできます。ちなみにTOKYO NAMPA STREETの(AZASの作者のきたが移植した)PC-8801バージョンとX1バージョン(両方ともDISK版)ではMINI-DOSから"autoex.bat"によってゲームを立ち上げています。PC-8801版ではBOOTに手を加え、DMA禁止のまま立ち上がるようになっているので、見てもわかりませんが、X1版は見ていると"autoex.bat"がゲームをスタートさせているのがわかります。

AZASのマスター・ディスクでは"autoex.bat"で次のコマンドを実行させています。
(t△autoex.batと入力すればすぐに見られます)

☆ :senko	先行入力をDOSに拡張(1-3-9.)
☆ senon	先行入力ON
☆ :lpttab	プリンターTABコード変換をDOSに拡張(1-3-10.)
☆ :time△@85	カレンダーの年を設定。(1-3-2.)
☆ t△open.msg	オープニング・メッセージを出力。(1-2-6.)
☆ d	ディレクトリを出力。(1-2-1.)

★
★ AZAS
★ Reference Manual
★ 2. SOCE /Screen Oriented Character Editor
★
★ SOCE Version 1.42対応

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★

★ 2-0. SOCEの特徴 ★

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★

- ☆ アセンブラーのソースの作成編集を主目的とするエディターです。
- ☆ その他の言語やアスキー・ファイルの作成編集にも使用できます。
- ☆ 1文字単位でエディットするキャラクター・エディターです。
- ☆ エディット・モードでは、コマンドは特殊キー(コントロール・キー)を使用し、テキスト入力とコマンド入力のモードの区別はありません。
- ☆ 画面を二つに分割し、テキストの別の箇所を同時に表示できる。分割は縦にも横にもできます。
- ☆ アセンブラーのソースには自動的にスペースをTABコードに置きかえてフォーマットする機能があります。
- ☆ ユーザーが各種のモードの初期設定を変更できます。
- ☆ エディット・モードでのキーの対応を1対1で変更できます。

SOCEは『そーす』と発音します。

SOCEは“ed. filc”というフィルター・コマンドになっています。

SOCEを起動すると、まずDOSのコマンド・フィルターがはたらくコマンド・モードに入る。コマンド・モードではエディタのいくつかのモードの設定ができます。

テキストを編集するためには、そこから更にエディット・モードに入らなくてはなりません。エディット・モードではフル・スクリーン・エディターとして動作します。

★ ★★★★★★★★★★★★★★★★★★★★★★★★

★ 2-1. コマンド・モード ★

★ ★★★★★★★★★★★★★★★★★★★★★★★★

コマンド・モードはDOSのコマンド・フィルター機能を利用しています。

SOCEを使用するにはまずコマンド・モードに入らなくてはなりません。

SOCEを起動してコマンド・モードに入るには、

: e d △ fn (aza)

と入力します。fnは編集したいファイルの名前で、デフォルトのファイル・タイプの“.aza”はSETFTコマンド(2-1-20.)で変更できます。

新しいファイルを作りたいときは、SAVEするファイル名を入力してください。このとき、SOCEは、

***** NEW FILE *****
SURE (Y/N) ?

と確認をとりますので、“Y”を押して下さい。(もし間違えて存在しないファイルをエディットしようとした場合は、“N”を押してエディタから抜けます。)

コマンド・モードに入ると、

* e d △ READY.

というプロンプトが出るようになります。それと分かります。

SOCEを終了させるときには、一定のシーケンスが自動的に行われます。そのシーケンスを発生させるコマンドは次の二つです。

2-1-1. “……名前なし……”

2-1-2. “* ! ”

※例外は(2-1-5.) CLEARコマンドで、何もしないでSOCEから抜けだします。

このとき発生するシーケンスは、(エディット中のファイル名を“xxx.aza”とします。)

1. “xxx.bak”があれば削除します。
 2. “xxx.aza”があれば“xxx.bak”にリネームします。
 3. “xxx.aza”というファイル名でテキストをセーブします。
- ファイル・タイプが“.aza”以外のファイルをエディットしたときは、そのファイルの

ファイル・ラインに置きかえて読んでください。

また、".bak"というファイル・タイプもSETBAKコマンド(2-1-21.)で変更できます。

コマンド・モードで使用できるコマンド。

2-1-1.	"...名前なし..."	SOCEの自動終了
2-1-2.	"!"	SOCEの手動終了
2-1-3.	"SAVE"	テキストのセーブ
2-1-4.	"LOAD"	別のファイルをエディット
2-1-5.	"CLEAR"	テキストを捨てる
2-1-6.	"BUF"	-----
2-1-7.	"."(ピリオド)	各種情報を表示
2-1-8.	"E"	エディット・モードへ入る
2-1-9.	"F"	"
2-1-10.	"HDIV"	画面横分割
2-1-11.	"VDIV"	画面縦分割
2-1-12.	"TON"	オート・フォーマッタON
2-1-13.	"TOFF"	" OFF
2-1-14.	"SON"	補助音出力ON
2-1-15.	"SOFF"	" OFF
2-1-16.	"MARK"	エラー発生行セット
2-1-17.	"KEYMAC"	キー・マクロ定義
2-1-18.	"KEYLIST"	定義されたキー・マクロの出力
2-1-19.	"KEYCLEAR"	定義されたキー・マクロを消去
2-1-20.	"SETFT"	ファイル・タイプのデフォルトの設定
2-1-21.	"SETBAK"	バック・アップのファイル・タイプの設定
2-1-22.	"CONFIG"	エディターの再設定
2-1-23.	"NECAON"	NECコンパチEOFセーブ・モード
2-1-24.	"NECAOFF"	高速テキスト・セーブ・モード

2-1-1. **名前なし.....** 【SOCEの自動終了】

このコマンドは、本当はSOCEのコマンドではなくDOSの機能なのですが、機能の実行はSOCE自身が行い、またユーザーからもSOCEのコマンドのように見えるのでコマンドに含めました。

アセンブラー等を起動するコマンドを入力すると、自動的にテキストをセーブする終了シーケンスが発生し、それから、入力されたコマンドを実行します。

これにより、終了コマンドを入力しセーブが終わるのを待ってアセンブル・コマンドを入力する、という手間のかかるなどをしなくても、アセンブル・コマンドを入力するだけで、エディターの終了も行われます。

2-1-2. ** ! ** 【SOCEの手動終了】

書式 !

エディターの終了シーケンスを発生させ、ノーマルなDOSのコマンド待ちに戻ります。

2-1-3. ** SAVE ** 【テキストをセーブする】

書式 SAVE△[fn]

エディット中のテキストをセーブする。通常は終了シーケンスでセーブされるので使用する必要はないコマンドです。

ドライブ、ファイル名、ファイル・タイプは省略すると、エディット中のファイルと同じになります。全て省略すると同じファイル名でセーブされます。

エディット中にディスク上に複製を作りたいときに使ってください。

2-1-4. ** LOAD ** 【別のファイルをエディットする】

書式 LOAD△fn (aza)

現在エディット中のファイルに終了シーケンスを発生させ、その後fnというファイルをロードします。

2-1-5. “CLEAR” 【テキストを捨てる】

書式 CLEAR

エディット中のファイルを捨てて、エディターを終了します。

このコマンドで終了した場合、ディスク上のファイルはもとのまま残っています。

2-1-6. “BUF” 【-----】

※※※ 使うべきではないコマンド。次のバージョンでは削除します。 ※※※

2-1-7. “-” (ピリオド) 【各種情報の表示】

書式 -

次の情報／モードを表示します。

テキストのサイズ、エディット・バッファのサイズ

CTRL-Yによるキャピタル・ロック

補助音のスイッチ

画面分割の縦横

オート・フォーマッタが有効か

キー・マクロ定義数

ファイル・タイプのデフォルト

バック・アップ・ファイルのファイル・タイプ

セーブはNECタイプか高速タイプか

2-1-8. “E” 【エディット・モードに入る】

書式 1) E

2) Estr

書式1) はそのままエディット・モードへ入ります。

書式2) は行頭のstrという文字列をテキストの最初からサーチし、その行にポインターをセットしてエディット・モードに入ります。見つからなかったときは、ポインターは変えないでエディット・モードに入ります。

2-1-9. “F” 【サーチ&エディット・モードに入る】

書式 Fstr

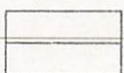
ポインターのあるところからstrという文字列をサーチし、見つけた行にポインターをセットしてエディット・モードに入ります。見つからなかったときは、ポインターは変えないでエディット・モードに入ります。

2-1-8. “E” コマンドの書式2) とは、サーチを始める場所 (Eコマンドはテキストの先頭から) とサーチする対象 (Eコマンドは行頭のみ) が異なります。

2-1-10. “H D I V” 【画面横分割】

書式 HDIV

エディット・モードで画面を2分割するとき、横に分割します。

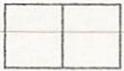


2-1-11. “V D I V” 【画面縦分割】

書式 VDIV

エディット・モードで画面を2分割するとき、縦に分割します。

初期状態ではVDIVになっています。



2-1-12. “TON” 【オート・フォーマッタON】

書式 TON

エディット・モードのオート・フォーマッタ機能をONにします。

アセンブラーのソースをエディットするときに使用します。

初期状態ではTONになっています。

2-1-13. “TOFF” 【オート・フォーマッタOFF】

書式 TOT
エディット・モードのオート・フォーマッタ機能をOFFにします。
アセンブラーのソース以外をエディットするときに使用します。

2-1-14. “SON”【補助音出力ON】

書式 SON

エディット中の補助音が出るようにします。
初期状態ではSONになっています。

2-1-15. “SOFF”【補助音出力OFF】

書式 SOFF

エディット中の補助音が出ないようにします。
なお、SOFFしてもエラー音や警告音は出ます。

2-1-16. “MARK”【エラー発生行セット】

書式 MARK

アセンブラー等で発生したエラーのファイル上の場所にポインターをセットします。当然、最後に発生したアセンブル・エラーのあったファイルがロードされていなければなりません。MARKコマンドを実行するとポインターはエラー行の次の行にセットされます。AZAアセンブラーを使う時の注意。

インクルードされたソース・ファイルの中で発生したエラーはメイン・モジュール内のINCLUDE疑似命令の場所をMARKポインターにセットします。すなわち、インクルードされたファイルの中でのエラーの場所はMARKコマンドでセットすることはできません。

2-1-17. “KEYMAC”【キー・マクロ定義】

書式 1) KEYMAC△n1, n2

2) KEYMAC△n1

KEYMACによってエディット・モードでのキーの対応を1対1で変えることができます。

書式1)は定義でn1というキーを押すと、n2をSOCEは受け取ります。

n1とn2は10進数のアスキーコードで指定します。

定義は40個までできます。

例：KEYMAC△8, 7とするとDELキーを押すとSOCEはコントロール・Gを受け取りカーソル位置の文字の削除を行うようになります。(FM-7のDELキーのように。)

書式2)はn1というキーの定義を消去します。

SOCEが終了すると、この定義は消えてしまいます。KEYLIST(2-1-18.)でセーブするか、CONFIG(2-1-22.)でエディタに設定してしまって下さい。

2-1-18. “KEYLIST”

【定義されたキー・マクロの出力】

書式 1) KEYLIST

2) KEYLIST△fn

書式1)は定義されたキー・マクロのリストをコンソールへ出力します。

書式2)は定義をファイルへセーブします。このファイルをロードするには、DOSのコンソール入力リダイレクトを使用します。（<fnとします）

特にファイル・タイプにデフォルトはありませんが、いちおう“.kmc”をkeymacroファイルの標準ファイル・タイプとします。

2-1-19. “KEYCLEAR”

【定義されたキー・マクロを消去】

書式 KEYCLEAR

キー・マクロの定義を全て消去します。

2-1-20. “SETFT”

【ファイル・タイプのデフォルトの設定】

書式 SETFT△ft

ロードするファイルのファイル・タイプのデフォルトを設定します。

2-1-21. "SETBAK"

【バック・アップのファイル・タイプの設定】

書式 SETBAK△ft

バック・アップ・ファイルを作る時のファイル・タイプを設定します。

CONF1G(2-1-22.)とペアで使います。

2-1-22. "CONFIG"

【エディターの再設定】

書式 CONFIG△fn

現在の各種設定を持ったエディターをセーブする。

CONFIGで設定できるのは以下の通りです。

(2-1-10, 11)	H DIV/V DIV
(2-1-12, 13)	TON/TOFF
(2-1-14, 15)	SON/SOFF
(2-1-23, 24)	NEC△ON/NEC△OFF
(2-1-17)	KEYMAC
(2-1-20)	SETFT
(2-1-21)	SETBAK
(2-2-15)	コントロール・Y(キャピタル・ロック)

例えば、バッチ・コマンド専用の"eb.flc"を作りたい時、

:ed△dmy

などとダミー・ファイルでエディターを起動します。

toff

でオート・フォーマット機能をOFFします。

setft△bat

でファイル・タイプのデフォルトを".bat"にします。

config△eb.flc

と最後に入力し、"eb.flc"を作成します。

clear

でダミー・ファイルを捨てるのをお忘れなく。

2-1-23. "NEC△ON"

【NECコンパチブルEOFセーブ・モード】

書式 NEC△ON

テキストをセーブするとき、EOF(エンド・オブ・ファイル)をNECのディスク・ベーシックとコンパチブルにします。

2-1-24. "NEC△OFF"

【高速テキストセーブ・モード】

書式 NEC△OFF

テキストを高速にセーブします。そのため、EOF(エンド・オブ・ファイル)がNECのディスク・ベーシックと完全に同じにはなりません。MINI-DOSではEOFコード(\$1A)が入っているとファイルの途中でもEOFとみなしますが、ディスク・ベーシックでは最後のセクターのEOFコードのみをEOFとします。

初期状態ではNEC△OFFになっています。

★ ★★★★★★★★★★★★★★★★★★★★★★ ★

★ 2-2. エディット・モード ★

★ ★★★★★★★★★★★★★★★★★★★★★★★★ ★

エディット・モードでは入力されたキーによって、以下のようなファンクションに分類できます。

1. 1キーのエディット・コマンド (2-2-x)
2. [CLR]キーを前に押す拡張エディット・コマンド (2-3-x)
3. 通常の文字の入力

通常の文字を入力すると、カーソル位置への一文字入力となります。インサート・モード

（）とソルバー、セイドを切り替えられます。（2-3-1. INS/REP）

以下カーソルとポインターという言葉がでてきますが両者はほぼ同じものです。
画面上ではポインター（メイン）のある行はアンダーラインで示し、サブ・ポインター
のある行はブリンクで示します。ブロックに指定されたエリアはリバースで示します。

2-2-1. “cursor keys” 【カーソル移動】

使用キー ↑↓←→：カーソル・キー
上下左右にカーソルを移動させます。

2-2-2. “ESC” 【コマンド・モードへ戻る】

使用キー ESC：エスケープ・キー

2-2-3. “CTRL-B” 【カーソルを10行戻す】

使用キー CTRL-B

2-2-4. “CTRL-C” 【入力1行キャンセル】

使用キー STOPキー または CTRL-C

カーソルが他の行に行く前なら1行分の入力をキャンセルできます。必ず使えるとは限
りません。

2-2-5. “CTRL-E” 【1行クリア】

使用キー CTRL-E

カーソル位置から行末までクリアします。

2-2-6. “CTRL-F” 【強制フォーマット】

使用キー CTRL-F

カーソルのある行にオート・フォーマットを作用させます。通常は1行入力ごとにオー
ト・フォーマットは働くので、この機能は使いません。

2-2-7. “CTRL-G” 【カーソル位置の1文字を削除】

使用キー CTRL-G

FM-7の[DEL]キーのような機能です。

2-2-8. “CTRL-H” 【カーソルの手前の1文字削除】

使用キー DELキー または CTRL-H

2-2-9. “CTRL-I” 【カーソルのTABごと移動】

使用キー TABキー または CTRL-I

TAB位置ごとカーソルを移動させます。

2-2-10. “CTRL-K” 【ポインターをテキスト先頭へ】

使用キー HOMEキー または CTRL-K

ポインターをテキストの先頭にセットします。

2-2-11. “CTRL-L” 【拡張コマンド・プリフィックス】

使用キー CLRキー または CTRL-L

(2-3-x) 参照

2-2-12. “CTRL-M” 【リターン・キー】

使用キー RETURNキー または CTRL-M

インサート・モードのとき行を分割します。

リプレース・モードのときは単に改行します。

2-2-13. “CTRL-N” 【カーソルを10行進める】

使用キー CTRL-N

2-2-14. “CTRL-X” 【カーソルを行末へ移動】

使用キー CTRL-M

2-2-15. "CTRL-Y" 【キャピタル・ロック】

使用キー CTRL-Y

押すごとに反転します。

初期状態ではONになっています。PC-8801でCAPSキーを使ってキャピタル

・ロックしたい人は解除して下さい。

2-2-16. "CTRL-Z" 【ポインターをテキスト末へ移動】

使用キー CTRL-Z

★ ★★ ★

★ 2-3. エディット・モード／拡張エディット・コマンド ★

★ ★★ ★

[CLR]キーを押した後に指定された文字キーを押して実行させます。

ファンクション・キーに定義されているコマンドはファンクション・キーを使うとキー
・リピートがききます。

2-3-1. "1" 【INS/REP】

使用キー f・1 または CLR+"1"

インサート・モードとリプレース・モードを切り替えます。

2-3-2. "2" 【ポインター交換】

使用キー f・2 または CLR+"2"

ポインター(メイン)とサブ・ポインターを交換する。サブ・ポインターを操作するには、この機能でポインター(メイン)としてから操作します。

2-3-3. "3" 【単語単位でカーソル左移動】

使用キー f・3 または CLR+"3"

2-3-4. "4" 【単語単位でカーソル右移動】

使用キー f・4 または CLR+"4"

2-3-5. "5" 【次行先頭へカーソル移動】

使用キー f・5 または CLR+"5"

2-3-6. "6" 【1行削除】

使用キー f・6 または CLR+"6"

2-3-7. "7" 【画面2分割ON/OFF】

使用キー f・7 または CLR+"7"

2-3-8. "8" 【1行コピー】

使用キー f・8 または CLR+"8"

サブ・ポインターのある行をポインターの手前にコピーします。

それぞれのポインターは次の行に進む。

2-3-9. "9" 【1行転送】

使用キー f・9 または CLR+"9"

サブ・ポインターのある行をポインターの手前に転送します。

それぞれのポインターは次の行に進む。

2-3-10. "O" 【サブ・ポインターをポインターに一致させる】

使用キー f・10 または CLR+"O"

2-3-11. "B" 【ブロック開始点の指定】

使用キー CLR+"B"

ポインターのある行をブロックの開始点とします。

2-3-12. "E" 【ブロック終了点の指定】

使用キー

ポインターのある行の手前までをブロックとします。

2-3-13. ** H ** 【ブロック指定を消す】

使用キー CLR + "H"

ブロック指定をするとき、前回のブロック指定の場所によっては、開始点と終了点の関係が悪いという意味のエラー音とともに拒否されることがあります。そういう時に使用して下さい。

2-3-14. ** Y ** 【ブロック指定されたテキストを削除】

使用キー CLR + "Y"

2-3-15. ** C ** 【ブロック指定されたテキストを転送】

使用キー CLR + "C"

ブロック指定されたテキストをポインターの手前へ転送します。

2-3-16. ** W ** 【ブロック指定されたテキストをセーブ】

使用キー CLR + "W"

このコマンドを入力すると、ファイル名を聞いてきます。(デフォルトは".aza")

ここで、"L : dummy"とファイル名を指定するとプリンターへ出力できます。

2-3-17. ** R ** 【ファイルを読みこんでインサート】

使用キー CLR + "R"

このコマンドを入力すると、ファイル名を聞いてきます。(デフォルトは".aza")

ファイルを読みこんでポインターの手前にインサートします。

2-3-18. ** F ** 【文字列サーチ】

使用キー CLR + "F"

SEARCH FOR?にサーチしたい文字列を入力し、OPTION?にオプションを入力します。(2-3-AP1参照)

2-3-19. ** A ** 【文字列置きかえ】

使用キー CLR + "A"

SEARCH FOR?にサーチしたい文字列を入力して、REPLACE WITH?に置きかえる文字列を入力し、OPTION?にオプションを入力します。(2-3-AP1参照)

2-3-20. ** G ** 【サーチまたは置きかえの再実行】

使用キー CLR + "G"

前回のCLR + "F"またはCLR + "A"と同じ指定で再実行します。

2-3-AP1.

SEARCH FOR?に入力する文字列には次の特殊文字を使用できます。

CTRL-A (^A) 全ての文字に合います。

CTRL-Ox (^Ox) x以外の文字に合います。

CTRL-N (^N) 改行に合います。

CTRL-I (^I) TABに合います。

REPLACE WITH?に入力する文字列には次の特殊文字を使用できます。

CTRL-N (^N) 改行になります。

CTRL-I (^I) TABになります。

OPTION?に入力できる機能は以下の通り。いくつ指定してもかまいません。

指定したとき

指定しないとき

G 英大小文字を区別しません。

区別します。

W 単語ごとサーチします。

1 文字ごとサーチします。

N 置きかえのとき、置きかえる前に

発見するとすぐに置きかえます。

確認(Y/N)をとります。

G テキストの最後まで繰り返します。

一回実行したら止まります。

★★
★ ★
★ AZAS
★ Reference Manual
★ 3. AZA/Aki's Zilog Mnemonic Assembler
★
★ AZA Version 3. 3対応
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★★ 3. AZA / Aki's Zilog Mnemonic Assembler ★★
★★ AZA Version 3. 3対応 ★★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★ 3-1. アセンブラー・コマンド ★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

アセンブラーを起動するコマンドは、

: as △fn (aza), option, …, option

となります。fnはアセンブルしたいファイルの名前です。オプションは、幾つ指定しても、一つも指定しなくても、どちらでもかまいません。以下のオプションがあります。

- | | | |
|--------|-------------|-------------------|
| 3-1-1. | "L", "AL" … | アセンブル・リスト出力 |
| 3-1-2. | "Q" … | ラベル・リストのみ出力 |
| 3-1-3. | "X" … | エラー・ストップ |
| 3-1-4. | "F" … | オブジェクト出力 |
| 3-1-5. | "C" … | クロス・リファレンス情報出力 |
| 3-1-6. | "N" … | 無効行をリストしない |
| 3-1-7. | "@" … | @で始まるラベルも出力有効にする。 |
| 3-1-8. | "Y" … | シンボル・ファイル出力 |

3-1-1. ** L ** , ** AL ** 【アセンブル・リスト出力】

- | | | |
|-------|--------|---------|
| 書式 1) | L | AL |
| 2) | L + | AL + |
| 3) | L + fn | AL + fn |

"AL" が "L" 違うのは、行番号を付けてリストする所にあります。"AL" で行番号を付けると 1 行 80 文字のプリンターでは 80 文字を越えて改行してしまうことが良く起こります。(縮小モードにするとうまくいきます。)

書式 1) は CRT ヘアセンブル・リストを出力します。

書式 2) はプリンターヘアセンブル・リストを出力します。

書式 3) は fn というファイル名でアセンブル・リスト・ファイルを作ります。

3-1-2. ** Q ** 【ラベル・リストのみ出力】

- | | |
|-------|--------|
| 書式 1) | Q |
| 2) | Q + |
| 3) | Q + fn |

書式 1) は CRT ヘラベル・リストを出力します。

書式 2) はプリンターヘラベル・リストを出力します。

書式 3) は fn というファイル名でラベル・リスト・ファイルを作ります。

なお、(3-1-1.) L, AL でリストすれば最後に必ずラベル・リストを出力します。この Q オプションを使わなくてはならないのは、特にラベル・リストだけが必要な時だけです。

3-1-3. ** X ** 【エラー・ストップ】

- | | |
|----|---|
| 書式 | X |
|----|---|

エラーが発生したら、アセンブルを中止します。エラーにはアセンブルを継続できないエラーもあり、それらは X オプションにかかわらずエラーが発生するとアセンブルを中止します。

3-1-4. ** F ** 【オブジェクト出力】

- | | |
|-------|--------|
| 書式 1) | F - |
| 2) | F + fn |

AZA は F オプションの指定のない時、作成するファイルの名前を TITLE 疑似命令、無ければソースのファイル名(ファイル・タイプを ".img" とします)にもとめます。

書式 1) を指定すると、オブジェクト・ファイルの出力を止めます。(エラー・チェックだけします。)

書式 2) は作成するオブジェクト・ファイルの名前を fn とします。

オブジェクト・ファイルのファイル名は次の順番で決められます。

1. F + fn による指定。
2. F + オプション指定がなければ、T I T L E 疑似命令による指定。
3. 上の二つともない時、ソース・ファイル名のファイル・タイプを ".img" とする。

3-1-5. ** C ** 【クロス・リファレンス情報出力】

- 書式 1) C
 2) Cn:
 3) Cfn

クロス・リファレンサ "crf.trc" へ入力するためのクロス・リファレンス情報ファイルを作成します。出力されるファイルのタイプは必ず ".crf" になります。変えることも変わることもありません。このオプションで作られるファイルはクロス・リファレンサ以外では何の意味ももたないフォーマットをとっているからです。

書式 1) はファイルをソース・ファイルと同じファイル名で作ります。

書式 2) は書式 1) と同様にドライブn:に作ります。

書式 3) はfnというファイル名で作ります。

あらかじめ、AL オプションにより行番号付きリストを取っておかないとクロス・リファレンス・リストは利用できません。

(3-4.) クロス・リファレンサを参照して下さい。

3-1-6. ** IN ** 【無効行をリストしない】

- 書式 N

I F 疑似命令で無効となり (IGNORE) と出力される行をアセンブル・リストに出力しません。AL オプションで行番号も出力するとき、リストされない行にも番号はわりふられ、リスト上では番号は不連続になります。

3-1-7. ** @ ** 【@で始まるラベルも出力有効とする】

- 書式 @

@で始まるラベルは通常クロス・リファレンスにもラベル・リストにも出ませんが、@オプションを指定すると、出力されます。

3-1-8. ** Y ** 【シンボル・ファイル出力】

- 書式 1) Y
 2) Yn:
 3) Yfn

ラベル名とアドレスの対応をシンボル・ファイルとして出力します。出力されるファイルのタイプは必ず ".sym" になります。変えることも変わることもありません。このオプションで作られるファイルは、いまのところ何の使用目的もありません。便利な応用を考えてみて下さい。(未作成のシンボリック・デバッガーへの入力に使う予定です。)

書式 1) はファイルをソース・ファイルと同じファイル名で作ります。

書式 2) は書式 1) と同様にドライブn:に作ります。

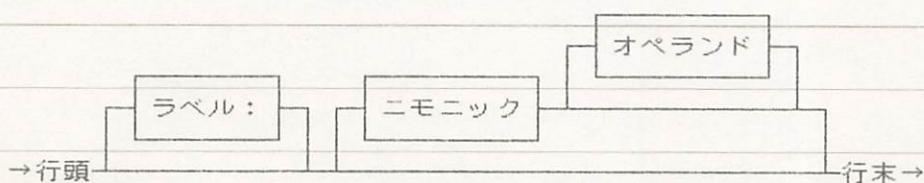
書式 3) はfnというファイル名で作ります。

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★ ★
 ★ 3-2. ソース・ファイル書式 ★
 ★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★ ★

3-2-1. ★★ ソース・ファイル形式 ★★

通常のアスキイ・ファイルであればOKです。

3-2-1-1. 【1行のフォーマット】



ラベルは必ず行頭に書かなくてはなりませんが、その他の間には任意個の（一つは必ずいれます）スペースまたはTABコード入れられます。

例外として、コメントは行のでこからでも書くことができます。
コメントは“;”（セミ・コロン）から行末までです。

3-2-1-2. 【ラベル】

ラベルの長さ（文字数）は制限ありません。しかし、ラベル・リストは14文字までしか表示できません。ラベルに使用できる文字は、

- 1文字め “@” または “A” … “Z”
- 2文字め以降、“@”、“A” … “Z”、“口” … “9”
 - “.” (ピリオド)
 - “_” (アンダー・ライン)
 - “\$”
 - “?”

@で始まるラベルは@オプションの指定がない限りラベル・リスト、クロス・リファレンスには出力されません。

3-2-1-3. 【ニモニック】

(3-3-×.)で説明する疑似命令とZ-80の全命令（ザイログ・ニモニックで定義されているもの）を使用できます。

\$ED, \$70という命令は、

IN A, (HL)

と書くことによって作成できます。

※ 本アセンブラーのZ-80命令はNEC uCOM(ミューコム)-82インストラクション活用表によりました。

3-2-1-4. 【オペランド】

オペランドが二つ以上あるときは、“, ”カンマで区切って下さい。

以下オペランドで特に注意を要するものと、AZAに独特のものを説明します。

3-2-1-4-1. 【プリ／ポスト・インク／デクリメント】

オペランドの中の次のレジスタ名の前後に“+”，“-”を任意個付けると、その命令の前または後ろにINCまたはDEC命令を+，-の数だけ作成します。

A, B, C, D, E, H, L, BC, DE, HL, IX, IY

例1 LD A, (HL+) → LD A, (HL)
 INC HL

例2 LD (--DE), A → DEC DE
 DEC DE
 LD (DE), A

3-2-1-4-2. 【数式】

(3-3-×.) 疑似命令でexpと略されている定数式や、オペランドに書ける定数は次のような機能が使用できます。なお、計算は無符号16bitで行います。

→ 数値 ↑	→
	-+ (加算) _____
	-- (減算) _____
	* (乗算) _____
	/ (除算) _____
	-MOD (除算の余り) _____
	.MOD. () _____
	-AND (bitごとの論理積) _____
	.AND. () _____
	-OR (bitごとの論理和) _____
	.OR. () _____
	-XOR (bitごとの排他的論理和) _____
	.XOR. () _____

計算の優先順位は、

高↑ * , /, MOD

+,-

低↓ AND, OR, XOR

当然括弧はこれらに優先されます。

数値 →	“一任意の1文字” →	文字定数(アスキーコード)
	'一任意の1文字'	" "
	10進定数	数値定数
	\$—16進定数	" "(注1)
	16進定数—H	" "
	%—2進定数	" "(注2)
	ラベル名	ラベルの値(注3)
	*	リファレンス・カウンターの値(注4)
	\$	" "
	(—式—)	括弧はアドレス指定と混同しないように ()は使わない方がよい。
	[—式—]	() [] {} この3種は同じ意味。
	{—式—}	b i t 反転する。
	NOT—数値	上位バイトをとりだす。
	HIGH—数値	下位バイトをとりだす。
	LOW—数値	右シフト
	SHR—数値	左シフト
	SHL—数値	

注1 …… 16進定数は“\$”マークを前に置くスタイルをMINI-DOSでは標準にしていますが、強制するものではありません。

注2 …… 2進定数には“\$”マークを混ぜてもかまいません。“\$”マークは無視されます。

注3 …… PHASE, ORG, EQU等のPASS1で値の確定していなければならぬ疑似命令で使うラベルは、それ以前に値が確定していないかもしれません。もし、確定していないラベルを使うと、AZAはエラー(Label undefined before)とみなします。

注4 …… アセンブル中の論理的なアドレスを示します。

3-2-1-4-3. 【インデックス・アドレッシング】

インデックス・アドレッシングのオフセットは、-128~+127の範囲内にして下さい。(IX+\$FF)のような表現はできません。やるとすれば、(IX+\$FFFF)か(IX-1)となります。この場合だけ例外的に数値を2の補数表現符号付として扱っていますので、注意して下さい。

3-2-2. ★★ オブジェクト・ファイル形式 ★★

オブジェクトはイメージ(BSAVE/BLOAD)・ファイルとして作成されます。

アドレスの範囲は、ORG疑似命令で指定したアドレスからEND疑似命令のあったアドレス-1までです。

3-2-3. ★★ リスト・ファイル形式 ★★

アセンブル・リストは、ページごとフォーマットされて出力されます。

各ページの先頭のページ・ヘッダに出力されるのは、

オブジェクト・ファイルの名前

日付

ページ・ナンバー

サブ・タイトル(SUBTITLE疑似命令による)

です。日付はカレンダー・クロックのものを使います。

1ページの行数60行ですが、PAGE疑似命令で変更できます。

FF(フォーム・フィード・コード:\$0C)で改ページします。このコードを受け付けないプリンターではうまく出力できません。

★ ★★★★★★★★★★★★★★★★
 ★ 3-3. 疑似命令 ★
 ★ ★★★★★★★★★★★★★★★★

AZAで使用できる、疑似命令は、以下の通りです。

3-3-1.	"ORG"	【オリジン】
3-3-2.	"DEFB", "DB", "DEFM", "DM"	【バイト定数】
3-3-3.	"DEFW", "DW"	【ワード定数】
3-3-4.	"DEFS", "DS"	【領域確保】
3-3-5.	"EQU"	【ラベル定義】
3-3-6.	".Z80"	【ダミー】
3-3-7.	"END"	【終了】
3-3-8.	".PHASE"	【アドレス変移】
3-3-9.	".DEPHASE"	【" " 終了】
3-3-10.	"IF"	【条件付アドレス】
3-3-11.	"ELSE"	【条件非成立の時】
3-3-12.	"ENDIF"	【条件付アセンブルの終了】
3-3-13.	"INCLUDE"	【ソース・ファイルの取り込み】
3-3-14.	"INCLIMG"	【イメージ・ファイルの取り込み】
3-3-15.	"INCLTWN"	【ソース/イメージ両ファイルの取り込み】
3-3-16.	".LIST"	【リスト出力停止】
3-3-17.	".LIST"	【" " 再開】
3-3-18.	"TITLE"	【オブジェクト・ファイル名指定】
3-3-19.	"SUBTTL"	【サブ・タイトル指定】
3-3-20.	"PAGE"	【改頁/1ページの行数】
3-3-21.	".XCRF"	【CRF情報出力停止】
3-3-22.	".CRF"	【" " 再開】
3-3-23.	"LOOP"	【ループ命令開始】
3-3-24.	"LEND"	【" " 終了】
3-3-25.	"REPEAT"	【リピート命令開始】
3-3-26.	"UNTIL"	【" " 終了】
3-3-27.	"KEEP"	【レジスタ保存開始】
3-3-28.	"KEEPOUT"	【" " 終了】
3-3-29.	"SRV"	【DOSサービス呼び出し】
3-3-30.	"ROMCALL"	【N-ROMコール】

3-3-1. **ORG** 【開始番地】

書式 ORG△exp

アセンブルする開始アドレスを指定します。

ORGは必ず一つなければなりません。また、二つ以上あってはなりません。

ORGはオブジェクトを作るどの命令よりも前になくてはなりません。

3-3-2. **DEFB**, **DB**

DEFM, **DM** 【バイト定数】

書式 DEFB△str/exp, str/exp, …, str/exp

バイト単位の定数を作ります。

DEFBはDB, DEF M, DMのどれでもかまいません。

str/expは数式または、"か'でくくられた任意の長さの文字列を書くことができます

。

3-3-3. **DEFW**, **DW** 【ワード定数】

書式 DEFW△exp, exp, …, exp

16bitの定数をつくります。

3-3-4. **DEFS**, **DS** 【領域確保】

書式 DEFS△exp

exp バイトのメモリーを確保します。

3-3-5. "EQU" 【ラベル定義】

書式 1) label△EQU△exp

2) label:△EQU△exp

label に exp という値で定義します。

書式 1) を AZA ではスタンダードにしています。

3-3-6. ".Z80" 【ダミー】

書式 .Z80

なにもしません。

3-3-7. "END" 【終了】

書式 END

必ずソース・リストの最後に入れてください。また、ソースの途中に END があった場合そこでアセンブルを終わってしまいます。

3-3-8. ".PHASE" 【アドレス変移】

3-3-9. ".DEPHASE" 【リ 終了】

書式 .PHASE△exp

:

.DEPHASE

.PHASE と .DEPHASE にはさまれた部分のアセンブル・アドレスを変移させます。

出力されるオブジェクトでは .PHASE/.DEPHASE があっても連續になっていて、.DEPHASE の後再開されるアドレスは .PHASE 命令のアドレスに .PHASE/.DEPHASE で出力されたオブジェクトのバイト数を加えたものになります。

すなわち、別の場所に転送されて実行されるルーチンをプログラムに埋めこむときなどに使います。

.PHASE と .DEPHASE はネスティングしていなければなりません。

例 アドレス データ

ORG	\$100	→ 0100:
DEFW	*	→ 0100:00 01
.PHASE	\$200	→ 0200:
DEFW	*	→ 0200:00 02
.DEPHASE		→ 0104:
DEFW	*	→ 0104:04 01
END		→ 0106:

このソースからできるオブジェクト・ファイル。

0100:00 01

0102:00 02

0104:04 01

※ .PHASE ではアドレスをみかけの上でだけ変えます。

3-3-10. "IF" 【条件付アセンブル】

3-3-11. "ELSE" 【条件非成立の時】

3-3-12. "ENDIF" 【条件付アセンブルの終了】

書式 IF△exp

:

[ELSE]

:

ENDIF

exp という式が〇でない時 IF に続く命令群をアセンブルし、〇の時は ELSE 以降の命令群をアセンブルします。ENDIF が現れると通常のアセンブルにもどります。

ELSE は省略できます。

IF-ELSE-ENDIF はネスティングしていなくてはなりません。

3-3-13. "INCLUDE" 【ソース・ファイルの取り込み】

書式 INCLUDE△fn (aza)

ソース・ファイルをインクルード（取り込み）します。インクルードされたファイルの

なかで INCLUDE 疑似命令は使用できません。

例 ファイル "smpl.aza" の内容が次のようにとします。

A_SUB:	LD	A, 12
	CALL	PRCHR
	RET	NC
	JP	ERROR

その時、次のソース・ファイルは、

	CALL	A_SUB
	JP	DOS_ENTRY
	INCLUDE	smpl
ERROR:	LD	HL, MSG
	CALL	PRSTR
	JP	DOS_ENTRY

次のようにアセンブルされます。

A_SUB:	CALL	A_SUB
	JP	DOS_ENTRY
	LD	A, 12
	CALL	PRCHR
	RET	NC
	JP	ERROR
ERROR:	LD	HL, MSG
	CALL	PRSTR
	JP	DOS_ENTRY

3-3-14. "INCLUDE"

【イメージ・ファイルの取り込み】

書式 INCLUDE fn(img)

イメージ・ファイルをインクルードします。イメージ・ファイルのアドレス情報は無視し、アセンブル中のロケーションにインクルードされます。

3-3-15. "INCLTWN"

【ソース/イメージ両ファイルの取り込み】

書式 INCLTWN fn(aza)

1命令で INCLUDE と INCLIMG の両方を行います。

INCLUDEするファイルの名前は指定されたものがそのまま使用されます。INCLIMGされるファイルの名前は指定されたもののファイル・タイプを ".img" に変えたものが使用されます。

3-3-16. ".XLIST"

【リスト出力停止】

書式 .XLIST

3-3-17. ".LIST"

【リスト出力再開】

書式 .LIST

アセンブル・リストの出力の指定がある時の意味をもちます。

.XLISTはアセンブル・リストの出力を停止させます。

.LISTは.XLISTで停止された出力を再開させます。

3-3-18. "TITLE"

【オブジェクト・ファイル名指定】

書式 TITLE fn(img)

オブジェクト・ファイルの名前を指定します。(3-1-4. Fオプション参照)

3-3-19. "SUBTTL"

【サブ・タイトル指定】

書式 SUBTTL str

アセンブル・リストの各頁の2行めに出力される、サブ・タイトルを指定します。strは60文字以内とします。SUBTTL 疑似命令は一つのソースの中に幾つあってもかまいません。

例 SUBTTL nantoka game version-2

3-3-20. "PAGE"

【改頁/1ページの行数】

書式 1) PAGE

2) PAGEΔexp

アセンブル・リストを出力する時のみ有効です。

書式1) の時、改頁をおこないます。

書式2) では、改頁のあと、1ページの行数をexp 行に設定します。無指定の時は60 行に設定されています。また、設定できるのは10行~255行の範囲内です。

3-3-21. ** X C R F ** 【CRF情報出力停止】

書式 . XCRF

3-3-22. ** C R F ** 【CRF情報出力再開】

書式 . CRF

クロス・リファレンス情報出力指定がある時のみ有効です。

. XCRFはクロス・リファレンス情報の出力を停止します。

. CRFは. XCRFで停止した出力を再開します。

3-3-23. ** L O O P ** 【ループ命令開始】

3-3-24. ** L E N D ** 【ループ命令終了】

一定回数繰りかえす命令を生成します。

※8bitレジスターでカウントするとき。

書式 LOOPΔreg[, exp]

:

LEND

reg はA, B, C, D, E, H, Lが使用できます。

以下のような命令に展開されます。

LD reg, exp → exp が無ければ生成されない。

rep :

:

DEC reg

JR NZ, rep →注1

注1 : JRでとどかないときは、JPに置きかえられます。

reg がBの時はDJNZを利用します。

※16bitレジスターでカウントするとき。

書式 LOOPΔreg[, exp]

:

LENDΔA

reg はBC, DE, HLが使用できます。ENDのオペランドにはAレジスタを使用する意味で、必ずAと書いてください。

以下のような命令に展開されます。

LD reg, exp → exp が無ければ生成されない。

rep :

:

DEC reg

LD A, reg-high

OR reg-low

JR NZ, rep →注1

注1 : JRでとどかないときは、JPに置きかえられます。

3-3-25. ** R E P E A T ** 【リピート命令開始】

3-3-26. ** U N T I L ** 【" " 終了】

書式 REPEAT

:

UNTILΔcond

condというフラグが成立するまでくりかえします。

condには次のものが使用できます。

Z, NZ, C, NC, PO, PE, M, P

以下のように展開されます。

rep :

:

J P Δnot-cond, r e p

not-condはcondの逆の条件です。

3-3-27. "KEEP"

【レジスタ保存開始】

3-3-28. "KEEPOUT" 【" 終了】

書式 KEEPΔreg , reg , ... , reg

:

KEEPOUT

KEEPは複数のPUSH命令を生成します。

KEEPOUTは対応するKEEPでPUSHしたレジスタ・ペアのPOP命令を生成します。必ず同じレジスタにPOPされ、またKEEPとKEEPOUTは必ず1対1になります。KEEPとKEEPOUTの対応はソース・リスト上の流れによります。(プログラム実行の順序などアセンブラーにはわからないので)

regは幾つ指定してもかまいません。

regに指定できるのは、

AF, BC, DE, HL, IX, IY

です。

例	KEEPΔHL, DE →	PUSHΔHL
		PUSHΔDE
	:	:
	KEEPOUT →	POPΔDE
		POPΔHL

3-3-29. "SRV" 【DOSサービス呼び出し】

書式 SRVΔexp

MINI-DOSのサービス・ファンクション・コールを行う命令を生成します。

expはサービス番号ですが、“dosequ.aza”というファイルにサービス番号とシンボルの対応がEQUで定義されているので、そのファイルをインクルードすると簡単にSRV疑似命令を使用できます。

SRVΔexpは以下のように展開されます。(2バイト)

RSTA\$28

DEFBΔexp

3-3-30. "ROMCALL" 【N-ROMコール】

書式 ROMCALLΔexp

N-BASIC・rom内のexp番地をエントリー・アドレスとするサブルーチンを呼び出す命令を生成します。この命令はMINI-DOSの機能を利用するもので、MINI-DOSの管理下でなければ利用できません。

サブルーチン実行中はアドレス\$0000～\$7FFFがROMになるためスタックやメモリー上のパラメータがこのアドレスと重なってはいけません。(ROMCALL命令は重なっていても大丈夫です。)

ROMCALLΔexpは以下のように展開されます。(3バイト)

RSTA\$20

DEFWΔexp

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★ ★

★ 3-4. クロス・リファレンサ “crf.flc” ★

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★ ★

クロス・リファレンサ “crf.flc”は、クロス・リファレンス・リストを出力する、AZAのユーティリティです。

使用法

: crf Δinput-fn (crf) [,output-fn]

ファイル・タイプ“.crf”的(3-1-5.) “C”オプションで出力されたファイルを入力し、クロス・リファレンス・リストを出力します。出力ファイル名を省略するとコンソールへ出力します。実際には、出力先として、

L:

とプリンターを指定することが殆どだと思います。

出力するリストの1ページの行数は、アセンブラー・リストでの行数と同じになります。(PAGE疑似命令で指定、無ければ60行/ページ)

リストの行番号で“#”マークが付いているのは、その行で定義されていることを示します。

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★ 3-A P 1 . エラーについて ★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★★★

AZAのエラーには、継続してアセンブルするものと、強制的にアセンブルを打ち切るものとの2種類があります。(3-1-3.) “X”オプションを使うときはどちらも、打ち切りになります。

以下の命令はきちんとネスティングしている必要があります。少しでもおかしいとエラーになります。

. PHASE/. DEPHASE
IF/ELSE/ENDIF
LOOP/LEND
REPEAT/UNTIL
KEEP/KEEPOUT
INCLUDE/インクルードされたファイルのEOF(エンド・オブ・ファイル)

例 KEEPABL, BC

:

LOOPAB, 10

:

KEEPOUT

:

LEND



←エラーになります。

★ ★★★★★★★★★★★★★★★★★★★★★★★★★★
★ 3-A P 2 . 予約語 ★
★ ★★★★★★★★★★★★★★★★★★★★★★★★★★

以下の名前はラベルに付けられません。

ニモニック名

レジスタ名

その他、疑似命令名、演算子名、etc.

ORG, DEFB, DEFW, DEFS, EQU, .Z80, END, .PHASE,
.DEPHASE, IF, ENDIF, ELSE, INCLUDE, TITLE, SUB
TTL, .LIST, .XLIST, PAGE, DEFN, ROMCALL, SRV, I
NCLIMG, INCLTWN, LOOP, LEND, DB, DW, DS, REPEAT
, UNTIL, KEEP, KEEPOUT, NOT, LOW, HIGH, MOD, SHR
, SHL, .CRF, .XCRF, DM (順不同)