

書 式	CALL BINLOAD(<番号> [, <機能依存フラグ>] [, <展開アドレス>] [, <バイト数>])
解 説	<p>バイナリシステムの仕様にに基づき、<番号>の連結データを指定されたメモリへ転送し、実行します。</p> <p><番号>は1バイトの整数で、以下のフラグを格納します。</p> <p>bit0~5 格納データ番号 (0~63)</p> <p>bit6 写しモード (0:OFF 1:ON)</p> <p>bit7 一気モード (0:OFF 1:ON)</p> <p>通常は<番号>の連結データの連結情報をDMシステム2のインフォメーションエリアへ書き出して処理を終了します。“0”を指定するとバイナリテーブルのヘッダ情報を書き出します。</p> <p>「写しモード」は、<番号>の連結データをバイナリテーブル構築時に設定したアドレスへ転送を行います。BPE圧縮された連結データは自動的に展開します。</p> <p>「一気モード」は、バイナリテーブル構築時に設定したアドレスへデータ転送を行った上で、そのデータを実行します。実行の動作はデータの種類により異なります。BPE圧縮されたデータは自動的に展開します。</p> <p><機能依存フラグ>は1バイトの整数で、バイナリテーブル構築時と同様のフラグを代入します。省略するとバイナリテーブルに内蔵されたフラグを参照します。</p> <p><展開アドレス><バイト数>は2バイトの整数で、バイナリシステムと同様の数値を代入します。COPY形式のVRAMデータは上位1バイトを<x座標>、下位1バイトを<y座標>として扱います。省略するとバイナリテーブルに内蔵された数値を参照します。</p>

2.3.12 美味しい命令

MSX-BASICではなかなか難しい、かゆいところに手が届く小粒な命令です。

メモリ操作関連

CALL BLOCK

機 能	メモリ内容をブロック転送します。
-----	------------------

書式 CALL BLOCK(<転送元アドレス>, <転送先アドレス>, <長さ>)

解説 <転送先アドレス>で示されるメモリから<長さ>分を<転送先アドレス>へコピーします。

<転送元アドレス><転送先アドレス>ともに、先頭に“@”を付けると VRAM 指定になります。

<転送元アドレス><転送先アドレス>に 65535 以上の RAM のアドレスを指定すると、マップ RAM 指定になります。65535+<利用可能なセグメント数>×16384 が上限となり、これを超える値を指定すると“Out of memory”エラーになります。

CALL PEEK

機能 RAM の 1 バイトの値を読みます。

書式 CALL PEEK(<アドレス>, <変数>)

解説 <アドレス>で示される RAM を読み込み、<変数>へ返します。

<アドレス>の先頭に“@”を付けると VRAM 指定になります。

8000h 未満の RAM は裏 RAM になります。

CALL PEEKS

機能 メモリの連続したバイトの値を文字変数に変換して読みます。

書式 CALL PEEKS(<アドレス>, <長さ>, <文字変数>)

解説 <アドレス>で示されるメモリから<長さ>分を読み込み、<文字変数>へ返します。

<アドレス>の先頭に“@”を付けると VRAM 指定になります。

8000h 未満の RAM は裏 RAM になります。

各バイトは 0~255 の半角文字に変換され、<文字変数>へ返します。

CALL PEEKW

機能	RAM の 2 バイトの値を読みます。
書式	CALL PEEKW(<アドレス>, <変数>)
解説	<アドレス>で示される RAM から 2 バイトを読み込み、<変数>へ返します。 <アドレス>の先頭に “@” を付けると VRAM 指定になります。 8000h 未満の RAM は裏 RAM になります。 2 バイトの表現は “ザイログ式” に基づき、下位 8 ビット・上位 8 ビットの順で読み込み、<変数>に返します。

CALL POKE

機能	RAM に 1 バイトの値を書きます。
書式	CALL POKE(<アドレス>, <数値>)
解説	<アドレス>で示される RAM に<数値>を書きます。 <アドレス>の先頭に “@” を付けると VRAM 指定になります。 8000h 未満の RAM は裏 RAM になります。

CALL POKES

機能	メモリの連続したバイトの値を文字変数から書きます。
書式	CALL PEEKS(<アドレス>, <文字変数>)
解説	<アドレス>で示されるメモリから<文字変数>分を書きます。 <アドレス>の先頭に “@” を付けると VRAM 指定になります。 8000h 未満の RAM は裏 RAM になります。 <文字変数>の半角文字は文字コード (0~255) に変換されます。

CALL POKEW

機能	RAM に 2 バイトの値を書きます。
----	---------------------

書式 CALL POKEW(<アドレス>, <数値>)

解説 <アドレス>で示される RAM に 2 バイトの <数値> を書きます。
 <アドレス>の先頭に “@” を付けると VRAM 指定になります。
 8000h 未満の RAM は裏 RAM になります。
 2 バイトの表現はザイログ式に基づき、下位 8 ビット・上位 8 ビットの順で書き込みます。

マシン語ルーチンの呼び出し

CALL CALL

機能 マシン語ルーチンをコールします。

書式 CALL CALL(<アドレス> [, <A>][, <HL>][, <DE>][, <BC>][, <IX>][, <IY>])

解説 <A><HL><DE><BC><IX><IY> をそれぞれのレジスタに代入して <アドレス> をコールします。
 マシン語から復帰した状態のレジスタの値は DM システム 2 のインフォメーションエリアに格納します。
 <A> に 16 ビットの値を指定すると、上位 8 ビットを F レジスタに代入します。
 マシン語ルーチンをコールした時点ではページ 0 はメイン ROM、それ以外のページは RAM になっています。

CPU モードの切り替え

CALL CHGCPU

機能 CPU モードを変更・獲得します。

書式 CALL CHGCPU([<CPUモード>][, <変数>])

解説

＜変数＞へ現在の CPU モードの数値を返し、＜CPU モード＞のモードに切り替えます。＜変数＞の値は turbo R と同名の BIOS と同じ 1 バイトの値です。

bit0～1 CPU モード (0:Z80 1:R800-ROM 2:R800-DRAM)

bit2～6 無効

bit7 高速モード LED の反映 (0:OFF 1:ON)

この命令は MSX2/2+/fMSX では無視します。＜変数＞には 0 を返します。

マップ RAM 保護のため、R800-ROM モードで駆動している状態で DMシステム2 をインストールすると、R800-DRAM モードへの変更はできません。

ディスク/ファイル操作関連

CALL CHGDRV

機能

カレントドライブを変更・獲得します。

書式

CALL CHGDRV([＜ドライブ番号＞][, ＜変数＞])

解説

カレントドライブを＜変数＞へ返し、＜ドライブ番号＞に切り替えます。

＜ドライブ番号＞＜変数＞は 1～8 でそれぞれ A:～H:を表します。

CALL FILES

機能

ファイル名を獲得します。

書式

CALL FILES(＜検索ファイル＞, ＜変数＞)

解説

＜検索ファイル＞に該当するファイルを検索し、該当数を＜変数＞へ返します。

＜検索ファイル＞はワイルドカードでの指定も可能です。

該当したファイル名は DMシステム2 のワークエリアである 7A00h から 12 バイト単位で格納します。実行後速やかに CALL PEEKS 命令や CALL BLOCK 命令で取り出すようにしてください。

CALL FSIZE

機能 ファイルのサイズを獲得します。

書式 CALL FSIZE(<ファイル名>, <変数>)

解説 対象ファイルのサイズが極端に大きいとき、<変数>が倍精度実数型でないと“Overflow”エラーを引き起こす場合があります。

CALL LOAD

機能 ファイルの内容を強引にメモリへ転送します。

書式 CALL LOAD(<ファイル>, <転送アドレス> [, <長さ>] [, <オフセット>])

解説 <ファイル>の<オフセット>から<長さ>分の内容を<転送アドレス>へ転送します。
<転送アドレス>の先頭に“@”を付けるとVRAM指定になります。
<長さ>は転送するバイト数で、省略するとファイル全体を転送します。
<オフセット>はファイルの先頭からのステップ値で、省略するとファイルの先頭(0バイト目)になります。

CALL SAVE ①

機能 メモリの内容をファイルに保存します。

書式 CALL SAVE(<ファイル>, <保存開始アドレス>, <長さ>)

解説 <保存開始アドレス>から<長さ>分の内容を<ファイル>として新規保存します。同名のファイルが存在する場合は、古いファイルの内容は完全に失われます。
<保存開始アドレス>の先頭に“@”を付けるとVRAM指定になります。

CALL SAVE ②

機能 メモリの内容を既存ファイルに更新します。

書式 CALL SAVE(<ファイル>, <保存開始アドレス>, <長さ>, <オフセット>)

解説 <保存開始アドレス>から<長さ>分の内容を<ファイル>の<オフセット>先へ追加保存します。
<保存開始アドレス>の先頭に“@”を付けると VRAM 指定になります。
<オフセット>はファイルの先頭からのステップ値で、メモリの内容をファイルの任意位置へ書き込みます。対象ファイルの長さより大きな値を指定すると、内容不定のデータ列で隙間が埋められます。

三角関数

CALL COS

機能 コサイン関数 (余弦値) を獲得します。

書式 CALL COS(<変数>, <角度>, <数値>)

解説 <角度>のコサイン値に<数値>をかけた値を 2 バイト整数で<変数>へ返します。角度は 360 度で一回転します。

CALL SIN

機能 サイン関数 (正弦値) を獲得します。

書式 CALL SIN(<変数>, <角度>, <数値>)

解説 <角度>のサイン値に<数値>をかけた値を 2 バイト整数で<変数>へ返します。角度は 360 度で一回転します。

各種処理待ち

CALL INTWAIT

機能 DM システム 2 の各種割り込み処理が終了するまで待ちます。

書式 CALL INTWAIT

解説 この命令を実行すると、割り込み処理がすべて終了するまで BASIC の処理を中断します。
BGM のフェードアウトや VDP マクロの終了確認などの用途に便利です。
待っている途中で **Ctrl**+**STOP** を押すと強制的に BASIC の処理を再開し、次の命令の実行に移ります。

CALL PAUSE

機能 1/1000 秒単位のウェイトをかけます。

書式 CALL PAUSE(<時間>)

解説 機能的には turbo R の CALL PAUSE 命令と同等です。<時間>は整数で指定し、<時間>× 1/1000 秒間何もせずに待ちます。
MSX2/2+/fMSX では待ち時間が 1/60 秒単位となるように自動的に換算します。端数は四捨五入となります。
待っている途中で **Ctrl**+**STOP** を押すと処理を中断します。

CALL VDPWAIT

機能 VDP コマンドの処理が終了するまで待ちます。

書式 CALL VDPWAIT

解説 VDP の処理が終了するまで何もせずに待ちます。
待っている途中で **Ctrl**+**STOP** を押すと処理を中断します。

CALL WAIT

機能 1/60 秒単位のウェイトをかけます。

書式 CALL WAIT(<時間>)

解説 <時間>は整数で指定し、<時間>× 1/60 秒間何もせずに待ちます。
待っている途中に **Ctrl**+**STOP** を押すと処理を中断します。

文字列の小文字→大文字変換

CALL UPPER

機能 文字列中の小文字英数を大文字に変換します。

書式 CALL UPPER(<文字変数>, <文字列>)

解説 <文字列>を大文字に変換して、<文字変数>に代入します。

グラフィックアキュムレータの設定

CALL XY

機能 グラフィック使用時の座標を設定します。

書式 CALL XY(< X 座標>, < Y 座標>)

解説 単なる座標設定で、MSX システムワークの“グラフィックアキュムレータ”へ座標を書き出します。
DM システム 2 の漢字表示機能を利用する際、この命令で < Y 座標 > に 0 未満か 256 以上の数値を設定しておく、アクティブページ以外のページを設定することができます。