PC-9800 シリーズが普及した時代の初代 CPU である 8086 は、1MB のメモリ空間を利用できることは、前述のとおりです。ここでは、この 1MB のメモリ空間について、もうすこし深く記述しましょう。
ある情報を 1MB の空間のどこかに書いたとしましょう。どこに書いたのか、それを示す単位をアドレスといいます。一般にアドレスは 16 進数で表現します。16 進数とは、16 まで数えると 1 桁繰りあがる数字の数え方で、末尾に h（ヘキサ）をつけて表現します。とはいっても数字は 0,1,2,3,4,5,6,7,8,9 までしかありません。そこで 16 進数では 9 の次は A と数えます。0h,1h,2h,3h,4h,5h,6h,7h,8h,9h,Ah,Bh,Ch,Dh,Eh,Fh まで数えて次が 10h になります。このことを踏まえて、次のステップへ進みます。
1MB は 1024KB ですが、それを 16 づつ分割して考えましょう。1 つのブロックは 64KB になります。本書では、このひとつひとつのブロックをセグメントと呼ぶことにします。下位から 0 セグメント、1 セグメント ¥¥¥A セグメント、B セグメント ¥¥¥F セグメントまでです。0 セグメントの先頭は、00000h といアドレス（絶対アドレス・5 桁の 16 進数）で表現します。ちなみに、下位から 640KB のところは、A セグメントの先頭で絶対アドレスが A0000h、一番上位のアドレスは、F セグメントの最後で絶対アドレスは FFFFFh になります。640KB のコンベンショナルメモリは、0 セグメントから 9 セグメントまで、絶対アドレスで表現すると 00000h から 9FFFFh になります。
さて、640KB 以上の部分は、どのように使われているのでしょうか。まず、A セグメントと B セグメント。ここには、V-RAM といって、画面に色や文字を写すための情報が収納されています。次に E セグメントと F セグメント。ここにも文字を写すための情報の一部と、あとはパソコンの基本動作（リセットしたら立ち上がるとか）や環境（メモリスイッチを有効にするなど）が記録されています。
残った C セグメントと D セグメントは、一般に拡張 ROM エリアと呼ばれています。このエリアは、ユーザがあとからパソコンにオプション機能を追加できるように、わざわざ開けてある領域です。例えばハードディスクをつなげると、D セグメントの一部に BIOS-ROM と呼ばれるプログラムが出現します。ハードディスクと CPU は、このプログラムを仲介してアクセスします。他にも EMS 拡張や LAN などが、この拡張 ROM エリアを使って機能します。

8086 は 1MB のメモリ空間しか利用できませんでした。次に発表された 80286 と呼ばれる CPU は、16MB の空間を利用できるように改良されました。その後発表された CPU の i386 は、80286 の機能に仮想 86 モードという、その後のパソコンの発展にとってかかせない、画期的な動作モードを搭載しました。
仮想 86 モードとは、1 つの i386 をたくさんの 8086 に見せかけるモードです。このモードを使うと、ひとつの CPU でありながら、8086 と 1MB のメモリがいくつもいくつもあるように動作します。8086 と 1MB のメモリがたくさんあるのですから、たくさんのパソコンがあるのと同じことになります。そして、複数のプログラムが同時に進行するマルチタスクが可能になったわけです。もちろんこのモードは、i486 や Pentium にも継承されています。
話題の WINDOWS は、i386 以上のパソコンでマルチタスクを実現します。これは、仮想 86 モードを利用しているからに他なりません。

EMS 拡張は、思い起こせば MS-DOS が全盛の時代、メモリ不足に悩んだパソコン業界があみ出した、MS-DOS で 1MB 以上のメモリを無理やり使う方法です。EMS とは Expanded Memory Spicification の略で拡張メモリ規格という意味です。Lotus-Intel-Microsoft の 3 社が提唱したこともあって、LIM 規格とも呼ばれます。
EMS 拡張では、どのようにして 1MB 以上のメモリを使うのでしょうか。まず、1MB 以上のメモリを（拡張メモリとよぶ）16KB づつの単位に分割します。この単位を 1 ページといいます。次に、読み書きしたいページを、C セグメントに差し込みます。C セグメントは 64KB ですから、1 度に 4 ページ差し込めるわけです。他のページに読み書きしたいときは、C セグメントのページを差し換えます。こうして EMS 拡張は、最大 16MB の拡張メモリを、C セグメントに入れ換え差し替えして、読み書きするわけです。
では、MELWARE は、この動作とどのように関係するのでしょう。拡張メモリの各ページを、C セグメントに差し換え作業をするのが、MELWARE に付属の MELEMM.386 というメモリマネージャです。アプリケーションソフトは、メモリマネージャに EMS メモリをとってきてくれ、と依頼します。メモリマネージャは、拡張メモリから未使用のページを C セグメントに差し込みます。アプリケーションソフトは、ここに読み書きを実行し、終了したら、ページをメモリマネージャに返却します。必要があれば、さっきのページをもう一度使うように、メモリマネージャにリクエストできます。EMS 対応アプリケーションソフトと、メモリマネージャがあってはじめて、EMS 拡張が利用できるわけです。.386 は、仮想 86EMS ドライバとも呼ばれます。MELEMM.386 は、仮想 86 モードを利用してページの移動を行っているからです。ですから、MELEMM.386 は i386 以上の CPU を搭載したパソコンで、プロテクトモード増設メモリがあるときに動作します。1993 年頃から、パソコンもメモリもこの条件にあてはまるものばかりになりました。今では MELEMM.386 も、標準的な機能です。

XMS とは、マイクロソフト、インテル、AST リサーチ、ロータスが提唱したメモリ管理規格で eXtended Memory Specification の略です。EMS 拡張の語源とそっくりですが、よくみると一文字違います。
8086 は、1MB のメモリしか管理できませんでしたが、その後開発された 80286 は 16MB までのメモリを管理できるように改良されました。この、80286 になってから初めて使えるようになったメモリのことを、プロテクト増設モードメモリといいます。i386、i486 は、プロテクトモード増設メモリを含めて 4GB のメモリを取り扱いできます。

ただし、パソコンで実際に使えるメモリ量は、パソコンの仕様によって制限があります。i386を搭載した多くのPC-9800シリーズは、コンベンショナルメモリ（0.6MB）とプロテクト増設モードメモリ（14MB）をあわせて、14.6MBのメモリを管理できました。1994年頃から、64MBなど、より広大なメモリを管理できるパソコンが増えてきています。
さて、どんなにたくさんのメモリをパソコンが管理できても、MS-DOSを使用している以上、アプリケーションソフトなどで自由に使えるメモリは、コンベンショナルメモリ（640KB）しかありません。そこで、パソコンが管理できるメモリのうち、コンベンショナルメモリを除く部分をなんとか活用するために考案されたのが、XMS拡張です。
XMS拡張では、メモリを次の3つに分け、それぞれの特徴を活かした使い道を定めています。
UMB（Upper Memory Blocks）は、8086の管理できる1MBのメモリの内、コンベンショナルメモリを除いた部分です。この領域の使い方は、少し複雑なので、4-1-4 UMB拡張のしくみで説明することにしましょう。
HMA（High Memory Area）は、プロテクト増設モードメモリの内、先頭の約64KBをさします。この部分に常駐するプログラムは、比較的簡単に作成できることから、他のプロテクト増設モードメモリと区別されています。1992年頃から辞書デバイスなどが積極的にHMAを利用しました。また、MS-DOS Ver.5は、CONFIG.SYSにDOS=HIGHを指定すると、システムの1部（MSDOS.SYS）が自動的にHMAへ常駐する仕組みになっています。
ただし、HMA領域は64KBと小さい上に、ひとつのプログラムしか入らない（いわゆる早い者勝ち）という欠点があります。MS-DOS Ver.5がHMAに対応してから、それ以外のプログラムがここを利用することはできなくなってしまいました。
EMB（Extended Memory Blocks）は、広大なプロテクト増設モードメモリそのものです。ここを活用してはじめて、拡張メモリを使ったと言えるでしょう。しかし実際のところ、EMBを活用するアプリケーションソフトは、さほど開発されませんでした。XMSの規格そのものが、EMBにはアプリケーションソフトのデータ部分を常駐するように規定していたからです。開発者にとっては、EMS拡張に対応するほうが、簡単で都合がよかったのです。
さて、有名無実だったXMS（EMB）を積極的に活用したのが、WINDOWSです。WINDOWSは、起動時にあるったけのEMBを確保します。そして、あとは自分の天下とばかりと、自分のやり方（実はDPMI拡張という方式）でメモリを管理します。アプリケーションソフトが動こうとすると、WINDOWSは必要な分だけのメモリをEMBからアプリケーションに分け与えます。EMBが足りなくなったら、ハードディスクのメモリまで分け与えます。（スワップ）WINDOWS上では、WINDOWSがまとめてメモリを管理するので、WINDOWS対応のアプリケーションソフトはメモリのことなど考慮せずにプログラムできます。プログラマにとっては、この上なく開発が容易なわけです。
さて、XMS拡張にも、メモリマネージャが存在します。アプリケーションソフトの要求をうけてHMAやEMBを分け与える、XMS拡張メモリの番人です。もちろんメルコのメモリドライバMELEMM.386は、XMSメモリマネージャの機能をもっています。ただし、仮想86モードをを利用したEMSメモリマネージャとして開発されました。その後、時代の要求があって、XMSメモリマネージャの機能を追加した経緯があります。そこで、次のように登録すると、XMSメモリマネージャの機能が付加される仕様になっています。

《CONFIG¥SYS》

UMBはUpper Memory Blocksの略です。8086が管理できる1MBのメモリのうち、640KBのコンベンショナルメモリを除く部分を、UMB領域といいます。
前書したとおり、PC-9800シリーズの場合、この領域はいろいろな用途で使われています。が、全部の領域が使い尽くされているわけではありません。このいくらかの空き領域を、無駄なく骨の髄まで使い尽くそうというのが、UMB拡張です。
そもそもメモリ不足は、640KBのコンベンショナルメモリの不足が原因で起きます。コンベンショナルメモリをできるだけ節約すれば、メモリ不足を回避できます。そこでUMB拡張では、本来コンベンショナルメモリに常駐する小さなプログラムを、UMBの空き領域（未使用の部分）に移動します。結果として、コンベンショナルメモリが節約できるわけです。
一般的に、UMBの空き領域に移動するプログラムは、PRINT.SYSやMOUSE.SYSなどCONFIG.SYSに登録するプログラム（デバイスドライバ）が多くあります。

さて、UMBの空き領域はどれぐらいあるものでしょうか。前書のとおり、8086の管理できる1MBのメモリ空間のうち、拡張ROMエリア（CセグメントとDセグメント）は、パソコンにオプション機能を追加するためにもうけられた空き領域です。実際にパソコンを使うとき、この空間はいろいろなオプションで虫食い状態に使われています。とはいっても、全く隙間がない、ということはまずありません。
拡張ROMエリアを、詳しく紹介しましょう。この空間は、全部で128KBあります。これを、16KBづつ8つのブロックに分割します。下から4つのブロック（C0〜CC）は、メモリ（EMS拡張）が利用するエリアです。メモリを利用しているユーザは、この領域は使用されています。パソコン内蔵型のハードディスクは、通常5つめのブロック（D0）の一部を使います。また、SCSIインターフェースボードを取りつけ、外づけのハードディスクやMOドライブを利用しているときは、SCSIインターフェースボードが一番上のブロック（DC）の一部を使っていることが多くあります。メルコのSCSIボードは、ユーザが希望すれば、使用するブロックを変更できるようになっています。その他、パソコンの拡張スロットに取りつけるタ

イプのインターフェースボードは、往々にして拡張 ROM エリアを使うことが多くあります。各々のインターフェースボードのマニュアルには、拡張 ROM エリアを使うかどうか、どこを使うか、が書いてあるので、機会があったら調べてみましょう。ちなみに、メルコのウィンドウアクセラレータやイメージスナッチャー、サウンドボードは、拡張 ROM エリアを使用しないように設計されています。
さて、パソコンの拡張 ROM エリアに存在する、空き領域は、どのような原理で使われるのでしょうか。ここで、メルコのメモリマネージャ MELEMM.386 が登場します。MELEMM.386 は、EMS 拡張の機能を持っています。この機能を応用して、拡張メモリから何ページかを UMB の空き領域にあてはめる。これで、空き領域が、ただ空いているのではなく、読み書き可能な状態になったわけです。

```
DEVICE=MELEMM.386 /M D4,D8 /HM
```

さらに、内蔵ハードディスクや SCSI インターフェースのプログラムを、UMB 領域以外に移動することもできます。A セグメントの後半（A5000h〜AFFFFh）の 12KB の空間に移動するのです。すると、D0 や DC のブロックが空きます。結果として活用できる UMB 領域が広がるわけです。

```
DEVICE=MELEMM.386 /M D0,D4,D8,DC /SW1 /HM
```

次に、本来 640KB のコンベンショナルメモリに常駐するプログラムを、UMB 領域に移動します。常駐プログラムの多くは、CONFIG.SYS に登録されている PRINT.SYS や MOUSE.SYS などのデバイスドライバです。メルコのディスクキャッシュ（HYPERDSK.EXE）や RAM ディスク（EXDISK.EXE）も、UMB 領域へ移動できます。ひとつひとつは小さくても、チリもつもれば山となってコンベンショナルメモリの不足をまねきます。これらを小まめに UMB 領域に移動するのが、UMBLOAD.SYS です。

```
DEVICE=MELEMM.386 /M D0,D4,D8,DC /SW1 /HM
DEVICE=UMBLOAD.SYS HYPERDSK.EXE C:2048 CW:2048 S
DEVICE=UMBLOAD.SYS EXDISK.EXE 2048/128
DEVICE=UMBLOAD.SYS PRINT.SYS
DEVICE=UMBLOAD.SYS MOUSE.SYS
```

MS-DOS Ver.5 以上を利用しているときは、UMBLOAD.SYS のかわりに、DEVICEHIGH というコマンドを利用できます。DEVICEHIGH は MS-DOSVer.5 以上の標準機能です。もちろん MELEMM.386 といっしょに使えます。

```
DOS=HIGH,UMB
DEVICE=MELEMM.386 /M D0,D4,D8,DC /SW1 /HM
DEVICEHIGH=HYPERDSK.EXE C:2048 CW:2048 S
DEVICEHIGH=EXDISK.EXE 2048/128
DEVICEHIGH=PRINT.SYS
DEVICEHIGH=MOUSE.SYS
```

UMB 領域に移動できるプログラムは、デバイスドライバだけではありません。常駐型コマンドも、UMB 領域に移動が可能です。常駐型コマンド、とは一度実行すると、いつでも呼び出し可能なコマンドのこと。TSR ともいいます。例えば、ハードディスクメニューや DOSSHELL などが、常駐型コマンドです。しかし、ハードディスクメニューや DOSSHELL は大きすぎて、一般に UMB 領域に収まりません。もっと小さい常駐型コマンドを使っていれば、UMB に移動できます。常駐型で、かつサイズの小さいコマンドは滅多にありませんが、MS-DOS の DOSKEY.COM はその条件にあてはまる数少ないコマンドのひとつで DOSKEY.COM を UMB 領域に移動するには、UMBLOAD.COM を使います。次のようにコマンド入力すると。DOSKEY.COM がコンベンショナルメモリではなく UMB 領域上で実行されます。

```
A>UMBLOAD DOSKEY
```

DOSKEY.COM を常日頃から利用している人は、AUTOEXEC.BAT に次のように登録すると便利です。

```
UMBLOAD DOSKEY
```

さて、ここまで UMB 拡張の仕組みと登録について説明してきましたが、UMB 領域を使いこなすには、相当の知識が必要だと感じた方も多いでしょう。事実、以前は一部のパワーユーザだけが、UMB 拡張を利用していました。しかし新しい MELWARE には、UMB 拡張を有効活用するためのバカチョンプログラムがついています。名前はオプティマイザ（OPTIMIZE.COM）といいます。ユーザはパソコンの環境についていちいち調べなくても、オプティマイザを実行すれば良いのです。
オプティマイザは、パソコンの環境に合わせて最大限に UMB 拡張を利用するように環境を整えるプログラムです。UMB 領域が不連続なときは、大きいサイズのプログラムを連続した領域に、小さいサイズのプログラムを不連続な領域に移動してくれます。
オプティマイザを起動する前に、ディップスイッチ SW2-5 を ON する必要があります。これで、メモリスイッチの内容が初期化されないようになります。

また、ハードディスクを２つ以上のパーティションに分けているときも注意が必要です。オプティマイザを実行すると、３回のリセットがかかります。ハードディスク起動メニューで起動ドライブを選ぶように設定していると、オプティマイザが実行中に止まってしまいます。FORMAT.EXE を起動して、ハードディスクの状態変更で自動起動を設定しましょう。オプティマイザを実行した後に、Ｓキーを押しながらリセットすれば、自動起動を解除できます。
次に、MELWARE をインストールしておきます。メモリマネージャ（MELEMM.386）が起動し、UMB 領域が存在していることが、オプティマイザを実行する条件です。
実際にオプティマイザを実行してみましょう。次のように入力します。

```
A>A:\MEL4WIN\OPTIMIZE
```

メッセージが表示され、「リセットが３回繰り返される」と表示されます。１回目のリセットは、UMB 領域の大きさと、常駐プログラムの大きさを計測しするために行われます。その結果を参考に、CONFIG.SYS や AUTOEXEC.BAT が書き換えられます。２回目のリセットは、指定そおりに UMB 領域が使われているかどうかを、チェックするために行われます。問題があれば、再度 CONFIG.SYS や AUTOEXEC.BAT を変更します。３回目のリセットがかかり、パソコンは効率よく UMB 拡張が使える状態になります。動作中にしばらく画面が止まったように見えますが、これは最適化の計算中のためですので、あわててリセットスイッチを押さないでください。
オプティマイザを実行すると、/UA:1 とか /UA:2 といったオプションスイッチが追加されることがあります。これは、それぞれのプログラムが、UMB 領域の何番目に常駐するか、を指定するオプションスイッチです。
このスイッチは UMB 領域がトビトビに存在するときに、設定されることが多くあります。普通（/UA がついていなければ）プログラムは、UMB 領域の１番目から順番に常駐しますが、それが効率的とは限りません。16KB 以上のサイズのプログラムは、UMB 領域が２つ以上連続した部分でないと常駐できないからです。そこで、小さいプログラムは連続していない UMB 領域に、大きいプログラムは連続している UMB 領域に常駐させると効率的、ということになります。
オプティマイザは必要があれば、プログラムが常駐する位置も入れ換えてくれます。入れ換えのおきた証拠が、/UA オプションです。このスイッチの利用は、大変難しいので、オプティマイザに任せるようにしましょう。
ディスクキャッシュは、ハードディスクと CPU のデータ転送を仲介するキャッシュメモリです。その原理は、ハードディスクから情報を読むときと、ハードディスクへ情報を書き込むときとで、若干異なります。
ハードディスクから情報を読むとき、１回よんだ情報はディスクキャッシュに記憶されます。もし同じ情報をもう一度読むことがあれば、次はディスクキャッシュから読み込むので速い、ということになります。
とはいえハードディスクの容量は 100MB を越えるのに、ディスクキャシュを 2MB や 4MB 確保してどうなるのでしょう。2MB や 4MB のディスクキャッシュは、すぐにいっぱいになってしまいます。その後新しい情報がディスクキャッシュに入ってきたら、ディスクキャッシュは再読み込み回数の少ない情報を捨てていきます。そのうちに、ディスクキャッシュの中には何度も読み込む情報だけが蓄積されます。そしてディスクキャッシュの効果が発揮されるわけです。ただし、ディスクキャッシュの中の情報は、リセットとともに消えてしまいます。

次に、ハードディスクへ情報を書き込むときの話をしましょう。ディスクキャッシュが設定し、ファイルの保存を実行します。意外と速く、ファイルの保存が完了します。キー入力やマウスの移動ができるようになり、ユーザは次の作業をはじめます。が、ハードディスクのアクセスランプをみると点滅しており、ハードディスクはまだ書き込み中です。これは、どういうことでしょう。
ハードディスクへ情報を書き込むと、ジコジコと音がします。少し大きめのデータファイル保存するときは、何秒間か時間がかかるものです。どんなに速いといわれるハードディスクでも、機械がジコジコいっている以上、電気で書き込みを行うメモリにくらべると、だんぜん遅いのです。
さて、ディスクキャッシュが設定されているとき、CPU から出力された情報は、ディスクキャッシュに書き込まれます。ディスクキャッシュはメモリですから、あっというまに書き込みが終了します。CPU は情報をさっさと出力してしまい、ハードディスクへの書き込みは完了したと考え、次の作業にうつります。ところで、ディスクキャッシュはハードディスクの書き込みスピードにあわせて、ゆっくり情報を出力するわけです。このようなキャッシュのしくみを、遅延書き込みとかライトバックキャッシュとかいいます。
特に WINDOWS を動かすときは、遅延書き込みを利用すべきです。WINDOWS は、ときどき唐突に現状を保存します。遅延書き込みが設定していないと、その度に WINDOWS の動作が停止するからです。これではイライラします。

便利な遅延書き込みですが、使い方を間違えると危険なこともあります。作業は完了した、と思った後も、ハードディスクにデータを書き込んでいることがあるからです。特にパソコンをリセットしたり電源 OFF するときは、要注意です。書き込み中にリセットや電源 OFF すると、データが壊れてしまいます。必ず STOP キーを押す習慣をつけましょう。^C が表示されれば、書き込みは終了しています。それから、リセットなり電源 OFF なりを実行すれば安全です。
さて、MELWARE のディスクキャシュ（HYPERDSK.EXE）を使う上の注意点を紹介しましょう。
ハードディスクを初期化するとき、512 バイト／セクタと 256 バイト／セクタという方式があります。HYPERDSK.EXE は、このうち 512 バイト／セクタに対応しています。リセット時に「サポートできないドライブがあります」と表示されるときは、256 バイト／セクタで初期化したハードディスクが存在しているわけです。ハードディスクを初期化しなおさななければ、残念ながらディスクキャッシュの恩恵には預かれません。
ディスクキャッシュが設定されていると、正しく動作できないプログラムもあります。ハードディスクの最適化（ファイルの順番を入れ換えて、ハードディスクのアクセス効率を高くするプログラム）は、ディスクキャッシュが設定されていると、ハードディスクの内容が壊れることがあります。ノートン・ユーティリティーズの SPEEDSK.EXE や、アドミラルシステムの Newton-PRO98 などが、そうです。また、圧縮ドライブを作成するプログラムも危険です。DiskXII の DXUT.EXE や MS-DOS Ver.6 の WDRIVE.EXE がそうです。このようなプログラムを実行するときは、ディスクキャッシュを無効にしましょう。また、すでに 512 バイト／セクタでフォーマットしてハードディスクや MO ディスクを、再度フォーマットするときも、ディスクキャッシュを無効にしなければなりません。
次のようにコマンド入力すると、ディスクキャッシュが無効になります。

```
A>HYPERDSK D
```

WINDOWS の設定について紹介しましょう。パソコンにはハードディスク BIOS といって、ハードディスクのアクセスを司るプログラムが用意されています。普通のアプリケーションソフトは、このプログラムを使って、ハードディスクに読み書きするようプログラムされています。HYPERDSK.EXE は、BIOS を行き来するデータを待ちかまえて、動作しています。
ところで、ハードディスクとアクセスする方法は、BIOS が唯一の手段ではありません。パソコンのハードウェアに詳しい開発者なら、他の方法をとることもできます。果敢にも、そのようにプログラムされたアプリケーションソフトを使うと、ディスクキャッシュの効果がでないということになります。
特に WINDOWS は、BIOS を使わないで、ハードディスク（スワップファイル）にデータを読み書きできます。そのまま使うと、大変まずいことになります。そこで、WINDOWS がちゃんと BIOS を使うように指定しなければなりません。

グループ「メイン」のアイコン「コントロールパネル」をひらき、その中から「エンハンスドモード」を起動します。
「スワップファイルの設定(V)」をクリックすると現在のスワップファイルの設定が表示されます。「変更」をクリックすると、ダイアログボックスの一番下に「BIOS を経由しないでスワップファイルを利用(U)」が表示されます。ここのチェックが外れていれば OK です。
RAM ディスクは、メモリを媒体とするディスクドライブです。ハードディスクや MO ディスクのように、ドライブとして扱われます。ドライブ名は、ハードディスクやフロッピーディスク、MO ディスクの後になります。
ハードディスクと比べると、RAM ディスクのアクセススピードは驚くほど速さです。ハードディスクは、メディアが回転し、ヘッドと呼ばれる部分が伸縮して、情報を読み書きします。ハード的な駆動部分がある限り、ハードディスクはどんなに改良されても、電気信号だけで情報を読み書きするメモリに勝てっこありません。もし、100MB ぐらいの RAM ディスクを作って、そこで WINDOWS を動かしたら、WINDOWS はすごいスピードで動くでしょう。（筆者も一度挑戦してみたいと常々思っております。）
ところが、電気で動く RAM ディスクは、電気が止まったらおしまいです。パソコンの電源を OFF にしたら、RAM ディスクの内容はきれいにクリアされてしまいます。リセットでも、消えてしまいます。アクセス速度は速いが、リセットすると消えてしまう、この 2 極性を理解して RAM ディスクを活用したいものです。
一般に WINDOWS は、できるだけ多くのメモリ（XMS 拡張）がある方が好ましいです。WINDOWS は高機能なアプリケーションソフトを動作させるためにメモリを大量に浪費します。WINDOWS を使うときは、RAM ディスクを使用しないで、その分 XMS 拡張メモリに回す方が好ましいでしょう。
これに対し、MS-DOS 版のアプリケーションソフトはそれほど多くの拡張メモリを必要としない場合が多くあります。ですからある程度 EMS 拡張メモリを確保したら（4MB 程度）、あとはディスクキャッシュや RAM ディスクに使うと良いでしょう。
MELWARE の RAM ディスク（EXDISK.EXE）は、MS-DOS が起動した後でも、RAM ディスクを追加できます。WINDOWS と DOS 版アプリケーションソフトの両方を使用している人は、DOS 版アプリケーションソフトを使用するときのみ RAM ディスクを作成すると良いのでしょう。
さて、RAM ディスクはどのようにして活用したら良いでしょうか。RAM ディスクの大きさが十分有れば、アプリケーションソフトをまるごと RAM ディスクにコピーして使うと効果抜群です。ですが、限り有るときは、効率よくアクセスの多い部分を RAM ディスクコピーすると効果的です。ワープロなら辞書。データベースならデータファイル。プログラマならソー

スファイルをRAMディスク上でつくりコンパイルすると良いでしょう。いずれのファイルも、MS-DOSのCOPYコマンドでコピーできます。
問題は、RAMディスクの内容を、リセットする前にハードディスクへ書き戻す必要があることです。これを忘れると、せっかく学習した辞書もデータファイルも、パアになってしまいます。そこでRAMディスクを活用するユーザには、各自でBATファイルを作成することをお勧めします。アプリケーションソフト起動時にファイルをRAMディスクにコピーし、アプリケーションソフトが終了したら書き戻すように設定します。これなら、ハードディスクに書き戻し忘れることも無いでしょう。

桐 Ver.5で、データファイルをRAMディスク（ドライブDとする）上で使用する例

```
ramdisk.bat
echo off
cls
EXDISK DEVICE = X 2048/128      :RAMディスク（2048KB）を作成する
COPY A:\*.TBL D:\                       :桐のデータファイルを、RAMディスクにコ
                                         ピーする
KIRI                                    :桐を起動する
COPY D:\*.TBL A:\                       :桐のデータファイルを、ハードディスクに
                                         書き戻す
EXDISK R                                :RAMディスクを解除する
ECHO ON
```

*CONFIG.SYSに次の行を登録しておく
```
        LASTDRIVE=Z
```

自信が無い人は、RAMディスクをあきらめて、ディスクキャッシュを活用する方が間違い無いことを、書き加えておきましょう。

　拡張メモリをどのように使えば良いか、については、1980年代後半に試行錯誤が繰り返されました。その中で、EMS拡張やXMS拡張が生まれたわけですが、他にもVCPIやDPMIという拡張規格があります。
　プロテクト増設モードメモリが世にお目見えした当初、その広大なメモリ空間を使う機能としてDOSエクステンダーが考案されました。DOSエクステンダーは、いわばプロテクト増設モードメモリの使い方（プログラム方法）のきまり、のようなもので、各アプリケーションソフトの機能の一部に取り込まれました。（プログラムが難しかったので、結果的にあまり流行しませんでした。）
　そして間もなく、プログラムのしやすさも手伝って、EMS拡張が流行しはじめました。すると、プロテクト増設モードメモリがDOSエクステンダー対応アプリケーションソフトと仮想86EMS対応のアプリケーションソフト同志で競合する恐れが発生しました。そこで規格されたのが、VCPIです。
　VCPIは、DOSエクステンダー対応アプリケーションソフトと、仮想86EMS対応アプリケーションソフトで、仲よくプロテクト増設モードメモリを分けて使いましょう、という規格です。例えば、あるときはDOSエクステンダー対応アプリケーションソフトを、あるときは仮想86EMS対応アプリケーションソフトを使用するときに、必要です。
　DPMIは、VCPIの規格を踏まえて、さらにメモリを活用しやすく機能アップした規格です。有名なところでは、WINDOWSがDPMI方式でメモリを利用しています。
　といってもDPMIが設定されていなければWINDOWSが起動しない、ということではありません。WINDOWSは、先にも書いたように、XMS拡張メモリがあれば起動します。これはどういうことでしょうか。
　WINDOWSは起動するとき、XMSメモリマネージャに、XMSメモリをリクエストします。そして、あるったけのXMSメモリを確保して起動します。一度起動してしまえば、あとは確保したメモリを自分の使いたい方法で使いたいように使うのですが、その方法がDPMIです。WINDOWSの場合、WINDOWS内部にDPMIの機能を持っていて、その方法でメモリを活用しているのです。
MS-DOS Ver.6.2は、DOUBLE SPACE（ハードディスクの容量を増やす機能）を採用した新しいOSです。MELWAREを使用するにあたって、MS-DOS Ver.6を採用すると、どんな利点があるでしょうか。
メモリ管理方法に関して、MS-DOS Ver.6はVer.5と同じ方法を受け継いでいます。CONFIG.SYSにDOS=HIGHを登録することで、HMA領域にMS-DOSのシステムの一部（MSDOS.SYS）を常駐させることもできるし、DEVICEHIGHを利用して小さな常駐プログラムをUMB拡張に移動することもできます。
MELWARE（for WINDOWS Ver.2）付属のメモリマネージャ（MELEMM.386）は、MS-DOS Ver.6に対応しています。MELWARE Ver.5、MELWARE for WINDOWS Ver.1などは、MS-DOS Ver.6上に完全対応できないので、注意が必要です。メモリマネージャは、パソコン稼働中は常に動いているソフトウェアです。是非、MELWARE forWINDOWS Ver.2を採用したいものです。
MELWARE for WINDOWS Ver.2付属のメモリドライバ（MELEM.386）やディスクキャッシュ（HYPERDSK.EXE）は、DOUBLE SPACEで容量が増えたハードディスクにも対応しています。まずは、ハードディスクをMS-DOS Ver.6でフォーマットし、DOUBLE SPACEを実行して容量を増やします。その後、普通にMELWAREをインストールすればよい。なお、さきにMELWAREをインストールしてからDOUBLE SPACEを設定するのは避けましょう。ハードディスクに対してディスクキャッシュが有効な状態でDOUBLE SPACEを設定すると、ハードディスクがクラッシュすることがあるからです。また、DOUBLE SPACEで容量を大きくしたハードディスクへのアクセスは、どうしても遅くなってしまうことを書き添えておきましょう。できれば、DOUBLE SPACEを採用するより、ハードディスク事態を増設する方が効率的です。

CD-ROMの増設するときは、CONFIG.SYSとAUTOEXEC.BATの両方に書き換えが必要です。CONFIG.SYSにはCD-ROM用デバイスドライバを登録します。デバイスドライバは、CD-ROMドライブを購入すると付属しているはずです。AUTOEXEC.BATには、CD-ROM用拡張ドライバMSCDEXを登録します。MSCDEXドライバがかなり大きなサイズを占めるので、気をつけないとすぐ

メモリ不足（コンベンショナルメモリの不足）になってしまいます。CD-ROMを使うときは、UMB領域を大きく広げ、UMBに追い出すようにしたいものです。
DOS版アプリケーションソフトとWINDOWSを交互に使用するときなどは、WINDOWS専用のAUTOEXEC.BATやCONFIG.SYSを作った方がよいでしょう。WINDOWS用のAUTOEXEC.BATやCONFIG.SYSには、余分なデバイスドライバの登録をしないで、シンプルな構成にするようにしましょう。

ネットワークを使うときは、LANボードそのものが拡張ROMエリアを使用します。また、LANドライバやSHARE.EXEがコンベンショナルメモリを圧迫するので、上手に設定しないとメモリ不足が発生しがちです。
LANドライバはデバイスドライバの形態で提供され、CONFIG.SYSに登録して使います。UMBの空き領域さえあれば、LANドライバをUMB拡張に追い出すことも可能です。また、排他制御を行うためのSHARE.EXEも、AUTOEXEC.BATにしばしば登録されます。このコマンドも常駐型プログラム（TSR）なので、UMB領域に移動できます。
メルコの簡単WEBは、インストールすると自動的にUMB拡張を使用するように設計されています。それ以外のネットワークOSを使うときは、オプティマイザを実行しましょう。LANドライバがUMB拡張に常駐するようになります。
注意しなければならないのは、ノートパソコンでPCMCIA規格のLANカードを導入するとき。これは、カードソケットドライバというカード専用のドライバソフトが追加されます。

何かしようと思うと、すぐ不足になるコンベンショナルメモリ。8086系のパソコンでは、どんなにメモリを増やしたところで、コンベンショナルメモリがいっぱいになってしまうと動かなくなってしまうケースが多くあります。このようなパソコンでは、いかにしてコンベンショナルメモリの消費を押さえるか、が拡張メモリを有効に使うカギになります。
コンベンショナルメモリの消費を押さえるには、まず第一に、使わないで済むプログラムを常駐させないことです。ハードディスクメニューやDOSSHELLなどを、使用をやめると、大きな効果があります。辞書デバイスの追加／削除（ADDDRV）も、コンベンショナルメモリを圧迫する大きな要因です。その他、不必要なデバイスドライバをCONFIG.SYSから削除したり、BUFFERSの数を少なくすることも、効果があります。また、WINDOWSでは、DOSアプリケーションで使っていた、ATOK8のような日本語フロントエンドプロセッサ(FEP)や、RS-232Cドライバ(RSDRV.SYS)、プリンタドライバ(PRINT.SYS)は必要ないので、これらのものが組み込まれていたら削除します。
次に、どうしても外せないプログラムを、出来るだけHMAやUMBに移動させることです。HMAは約64KB、UMBは環境にもよるが大きくて64KB、あわせて128KBあります。コンベンショナルメモリがもともと640KBしかないことを考えると、微量ながらも頼もしいメモリといえるでしょう。

MS-DOSはOSです。WINDOWSも、ある種のOSだ、といわれています。両者は、いったいどういう関係にあるのでしょうか。
そもそもOSとは、アプリケーションソフトから（もしくは人間から直接）命令を受け取り、パソコンの機械部分を動作させる基本ソフトウェアです。アプリケーションの方を向いてはソフトウェア語を話し、機械部分を向いては機械語をはなす、いわば同時通訳のようなものです。
そういう意味では、MS-DOSこそが本当のOSです。MS-DOS版アプリケーションソフトから命令を受け取り、メモリを利用したり、画面を写したり、ハードディスクに読み書きしたり、と機械部分を動かします。
WINDOWS（Ver3.1やWINDOWS NT）は、MS-DOS上で動くアプリケーションソフトの一種です。WINDOWSを動かす為には、必ずMS-DOSが必要です。証拠に、MS-DOSの入力モード（A>）からWINと入力すると起動します。
その半面、WINDOWSはOSの側面ももっています。証拠にWINDOWSの上では、WINDOWS版アプリケーションソフトを起動します。WINDOWSは、アプリケーションソフトからの命令を受け取り、機械部分を動かしているわけです。
このようにWINDOWSは、MS-DOSの方を向けばMS-DOS版アプリケーションソフト、アプリケーションソフトの方を向けばOSといった、両面性を持っています。
さて、次世代のWINDOWS（WINDOWS 95）は、MS-DOSを必要としなくなります。従来MS-DOSが受け持っていた機能を、WINDOWS 95に集約した為です。これをもって、WINDOWSは真のOSに昇格することになりそうです。逆にWINDOWS 95上で、MS-DOSを起動させることもできます。MS-DOSとWINDOWSの立場は、逆転することになります。

WINDOWSアプリケーションを動作させていて、アプリケーション実行エラー「メモリ不足のため実行できません。いくつかのWindowsアプリケーションを終了してから、やり直してください。」というエラーメッセージに遭遇した人もあるでしょう。
「自分の使っているパソコンには、16Mバイトもメモリを搭載したのに、更に追加しなければならないの？」と、疑問に思われるかもしれません。
このエラーは、何らかの形でWINDOWSの管理するメモリがいっぱいになってしまったときに表示されます。エラーを手っ取り早く解消するには、たくさん開いている窓を閉じて、WINDOWSで使用できるメモリを増やしてやることです。では、実際にWINDOWSがどんなメモリを管理しているのかを見ていくことにしましょう。
WINDOWSの管理するメモリは、大きく分けて、WINDOWSメモリとシステムリソースの2種類が存在します。
WINDOWSメモリは、WINDOWSがアプリケーションを実行するのに使用するメモリです。実際には、XMSメモリといわれる1Mバイト超のメモリと、スワップファイルといわれるハードディスク上に設定した仮想メモリ、DOSのコンベンショナルメモリの3つをあわせたものです。中でも、DOSのコンベンショナルメモリは特別な役割を担っていて、WINDOWSのタスク管理用に使用されています。
WINDOWSは、これらのメモリを一括してWINDOWSメモリとして管理しているが、どれか1つでも不足すると、「メモリ不足のため・・・」というエラーメッセージが出てしまいます。とりわけ、コンベンショナルメモリは、640KBと決まっているので、不足しがちです。そこで、UMB領域を活用し、コンベンショナルメモリを節約することが、ポイントになってきます。

WINDOWSメモリの使用状態は、WINDOWSユーティリティの環境モニタで観察できます。システムメモリのグラフ部分をマウスクリックすると、実メモリ（XMSメモリ）やスワップファイル、コンベンショナルメモリの使用状況が表示されます。
「メモリ不足のため・・・」が出るときは、どの部分が不足なのかを知るパロメータにするとよいでしょう。
システムリソースとは、WINDOWSのシステム（カーネル）が使用するメモリのことです。WINDOWSユーティリティの環境モニタに表示されるGDIリソースやUSERリソースがこれに相当します。GDI(Graphics Device Interface)リソースは、WINDOWSのディスプレイ表示や印刷出力のために使用されるメモリ。USERリソースは、マウスやキーボード等のユーザーインターフェースのために使用されるメモリ。ウィンドウの移動やアイコンの作成等もUSERリソースに関わってきます。実は、これらのメモリは、それぞれ64Kバイトに制限されていて、あまり多いとはいえません。これらのメモリが不足しても、「メモリ不足のため・・・」というエラーメッセージが出てしまいます。

「広大なプロテクト増設モードメモリを使いほうだい」と思っていたWINDOWSでも、メモリ不足がおきます。それは、メモリがいくつかのブロックに別れていて、そのうちどれかひとつのブロックでメモリ不足がおきれば、すなわちメモリ不足となるからです。メモリ不足を解消するには、どの部分が不足しているか正確に把握し、メモリを増やす（または節約する）努力をすべきです。不足箇所は、WINDOWSユーティリティの監視モニタで把握できます。ここでは、コンベンショナルメモリ、WINDOWSメモリ、システムリソースを増やす方法を考えましょう。

コンベンショナルメモリを増やすには
WINDOWSを使う上でも、コンベンショナルメモリは特別な役割を果たしています。ここが不足したら、どんなにプロテクト増設モードメモリが余っていても、メモリ不足になってしまいます。そこで、WINDOWSを使う上でも、コンベンショナルメモリはできる限り節約する必要があります。まずは、MS-DOSのレベルで、不必要な常駐プログラムを出来る限り外すこと。次にオプティマイザを使って、小さなプログラムをできる限りUMB領域に移動すること。UMB領域を制することが、コンベンショナルメモリの空き領域を増やすテクニックです。詳細は、4-2-4「640KBコンベンショナルメモリの不足を避けるには」を参照して欲しい。

WINDOWSのメモリを増やすには
もっとも手っ取り早いのが、メモリを増設することです。WINDOWSメモリは、XMSメモリ（プロテクト増設モードメモリ）が多ければ、比例して大きくなります。

システムリソースを節約するには
WINDOWS Ver3.1のシステムリソースは、GDIリソース、USERリソース共64Kバイトに制限されているので、残念ながら増やすことができません。ですから、節約するしか方法がありません。そのためには、いくつものアプリケーションを開けっぱなしにするのでなく、使っていないアプリケーションは終了するようにしましょう。

環境モニタで測定したGDIリソース、USERリソースの値

WINDOWSを使用中に、ちょっとだけMS-DOSを呼び出して、MS-DOS版アプリケーションソフトを使用することができます。おなじみの、グループ「メイン」のアイコン「MS-DOSプロンプト」です。これをクリックすると、MS-DOSのウィンドウが開きます。（俗にDOS窓と呼ぶ）
DOS窓には、MS-DOSの入力モード（A>）が表示されています。普通にMS-DOSのコマンドを入力すれば、アプリケーションソフトが起動します。ウィンドウのサイズでは動作しないソフトウェアも存在しますが、多くはGRPF+TABキーを押してフルスクリーンにすれば動き出します。ただし、あくまでWINDOWSを経由したMS-DOSで動作しているので、普段より動作が遅くなることは避けられません。
さて、MS-DOS版アプリケーションソフトがEMS拡張に対応しているときは、どうなるでしょう。WINDOWSはメモリをXMS拡張として管理しているはずです。
DOS窓でEMSを利用するアプリケーションでは、WINDOWSのPIFファイルにEMSの必要量を記述すればよいのです。WINDOWSが、自分の管理しているXMSメモリからDOS窓を介してEMSメモリを作ります。EMS対応のアプリケーションソフトは、動作できるわけです。

本章では、MELWARE for WINDOWS Ver.2に含まれる、MS-DOS関連のファイルの仕様を解説します。MS-DOS関連のファイルとはCONFIG.SYSに登録して使用する各種デバイスドライバ、および、MS-DOSのコマンドレベルからコマンド入力して実行する補助プログラムです。
これらのファイルは、MELWAREのオリジナルフロッピー（主にセットアップディスク）のサブディレクトリ\BINの中に収納されています。ファイルは圧縮された状態で収納されているので、そのままハードディスクにコピーして使用することは出来ません。MELWAREのインストーラを実行すると、ファイルの圧縮を解除し、ハードディスクのサブディレクトリ\MEL4WINにコピーされます。
本章では、次の順で各ファイルの仕様を説明します。

機能

i386/i486/Pentiumを搭載したパソコンで、プロテクト増設モードメモリをEMS拡張メモリとして管理する、EMSメモリマネージャ。i386以上のCPUがもつ、仮想86モードとページング機能を応用しています。オプションスイッチを設定すれば、UMB拡張やXMS拡張の機能を追加できます。

条件

CPUが386以上で、プロテクト増設モードメモリが存在すること。

使用法

形態

デバイスドライバ。MS-DOSのCONFIG.SYSにデバイス登録して使用します。

書式

```
DEVICE=[パス]MELEMM.386 [オプション1][オプション2] [オプション3]...
```

例）

```
DEVICE=A:\MEL4WIN\MELEMM.386 /HM /M D0,D4,D8,DC /SW1 /NECWIN
```

オプションスイッチ

/SD	EMSページフレームアドレスの指定
/P	EMSページ数の指定
/P	EMSページフレームドレスの絶対指定
/HM	XMS拡張の指定
/M	UMB領域の確保
/SW1	ハードディスクBIOSのスワップ
/T	MS-DOS Ver.5.0の拡張タスクスワップ機能指定
/NC	EMSファンクションの高速化
/XMS	XMSメモリ容量の上限設定
/H	EMSハンドル数の指定
/BE	仮想86バンクエミュレーション
/CX	Cx486DLC/SLC用キャッシュコントローラの動作指定
/SD	EMSページフレームアドレスの指定

概要

EMSページフレームとは、拡張ROMエリアの未使用領域に設定される64KBの空間。EMS拡張では、この領域に頁単位（1ページは16KB）で区切ったEMSメモリを差し替えます。

EMSページフレームは、通常C0000hからCFFFFh（絶対アドレス）の64KB空間に設定されるが、まれにこの領域が使用されてる場合があります。その為に、EMSページフレームの開始アドレスを変更するオプション（/SD）が用意されています。例えば、C0000hからC3FFFhが他のインターフェースボードで利用されているとき、このオプションを用いればEMSページフレームをC4000hからD3FFFhに移動できます。

書式　/SD nnnn

nnnn=EMSページフレームの先頭アドレス。16進数。（セグメントアドレス）

C000:C0000h(絶対アドレス)～:C000h（セグメントアドレス）（デフォルト値）
C400:C4000h(絶対アドレス)～:C400h（セグメントアドレス）
C800:C8000h(絶対アドレス)～:C800h（セグメントアドレス）
CC00:CC000h(絶対アドレス)～:CC00h（セグメントアドレス）
D000:D0000h(絶対アドレス)～:D000h（セグメントアドレス）

例）

EMSページフレームをC4000hから開始するとき

```
DEVICE=MELEMM.386 /SD C400
```

【注意】

EMSページフレームに指定した領域に、ハードディスクBIOSなどが存在するとき、MELEMM.386は競合する領域をEMSページフレームとして確保しません。このようなときは、指定よりも少ないページフレームを確保し、EMSが起動します。3ページしかEMSページフレームを確保できなかった場合、4ページのＥＭＳページフレームを必要とするアプリケーションソフトは動作しないことがあります。

/Pn　EMSページ数の指定

概要

EMSページ数とは、ページフレームに表示できるEMSメモリのページ数のこと。1ページは16KB。このオプションを指定したとき、ページフレームは連続して確保されます。通常は4ページ（64KB）を設定します。これは、EMS対応アプリケーションソフトの殆どが、4ページのEMSページフレームに対応しているため。もし、2ページや8ページのEMSページフレームを必要とするアプリケーションソフトを使用することがあれば、/PnオプションでページNを変更します。

書式　/Pn

n=ページフレームのページ数

2:2ページ
4:4ページ（デフォルト値）
8:8ページ

例）

EMS ページフレームを C4000h から 4 ページ（64KB）確保するとき
    DEVICE=MELEMM.386 /SD C000 /P4

【注意】
EMS ページフレームに指定した領域に、ハードディスク BIOS などが存在するとき、MELEMM.386 は競合する領域を EMS ページフレームとして確保しません。このようなときは、指定よりも少ないページフレームを確保し、EMS が起動します。3 ページしか EMS ページフレームを確保できなかった場合、4 ページのＥＭＳページフレームを必要とするアプリケーションソフトは動作しないことがあります。

/P　EMS ページフレームドレスの絶対指定

概要
通常 EMS ページフレームは連続した 4 ページを確保します。これは、EMS 対応アプリケーションソフトの多くが、連続した 4 ページの EMS ページフレームに対応しているからです。しかし、本来の EMS 規格は、ページフレームが必ずしも連続していなくても良しとしています。そこで、もし不連続な EMS ページフレームを設定するために、/P オプションが用意されました。/P オプションでは、EMS として確保するアドレスを直接指定します。

書式　/P XX,XX,XX,XX
XX=ページフレームのアドレス（16 進数。セグメントアドレスの先頭 2 桁）
C0:C0000h(絶対アドレス)～:C000h（セグメントアドレス）（デフォルト値）
C4:C4000h(絶対アドレス)～:C400h（セグメントアドレス）
C8:C8000h(絶対アドレス)～:C800h（セグメントアドレス）
CC:CC000h(絶対アドレス)～:CC00h（セグメントアドレス）
D0:D0000h(絶対アドレス)～:D000h（セグメントアドレス）
例）
  C0000h～C7fffh と D0000h～D7000h に、EMS ページフレームを作成します。
    DEVICE=MELEMM.386 /P C0,C4,D0,D4

【注意】
/SD /Pn オプションとは併用できません。
/P オプションで指定したアドレスに、ハードディスクの BIOS などが存在していても、MELEMM.386 は強制的にこの領域を使用します。拡張 ROM が競合すると、パソコンは正常に動作できません。/P オプションを使用するときは、拡張 ROM エリアの競合が起きないよう、注意が必要です。

/HM　XMS 拡張の指定

概要
MELEM.386 に XMS メモリマネージャの機能を追加します。プロテクト増設モードメモリがあれば、XMS 拡張メモリ（HMA と EMB）が利用できるようになります。MS-DOS Ver.5 の DOS=HIGH を使用するとき、WINDOWS を使用するときは、/HM オプションが必要です。省略時は、XMS 機能を追加しません。

書式　/HM
例）
    DEVICE=MELEMM.386 /HM

【注意】
MELEMM.386 のエンハンスドモードを利用するときは、ルートディレクトリまたは MELEMM.386 の存在するディレクトリに MELEMM.VXD も必要です。

/M　UMB 領域の確保

概要
UMB 領域の空き領域に、EMS 拡張メモリからメモリを埋め込みます。メモリを埋め込んだ空間は、UMB 拡張として、UMBLOAD.や MS-DOS Ver.5 の DEVICEHIGH などで利用できます。省略時は UMB 領域を確保しません。

書式　/M XX,XX,XX,XX
XX=メモリを埋め込むアドレス
C0:C0000h(絶対アドレス)～:C000h（セグメントアドレス）（デフォルト値）
C4:C4000h(絶対アドレス)～:C400h（セグメントアドレス）
C8:C8000h(絶対アドレス)～:C800h（セグメントアドレス）
CC:CC000h(絶対アドレス)～:CC00h（セグメントアドレス）
D0:D0000h(絶対アドレス)～:D000h（セグメントアドレス）
D4:D4000h(絶対アドレス)～D400h（セグメントアドレス）
D8:D8000h(絶対アドレス)～:D800h（セグメントアドレス）
DC:DC000h(絶対アドレス)～:DC00h（セグメントアドレス）
例）D0000h～DFFFFh の空間にメモリを埋め込みます。
    DEVICE=MELEMM.386 /M D0,D4,D8,DC

【注意】

/M オプションで指定した領域に、ハードディスク BIOS などが存在するとき、MELEMM.386 は競合する領域にメモリを埋め込みません。このようなときは、指定よりも少ない UMB 領域が確保されます。
MS-DOS Ver.5 の DEVICEHIGH を使用するときは、CONFIG.SYS に DOS=UMB を登録する必要があります。

  /SW1   ハードディスク BIOS のスワップ

概要
D0000h～DFFFFh に存在する BIOS（主にハードディスク BIOS）を、4KB 単位で最大 12KB まで A5000h～AFFFFh へ移動します。これにより、D0000h 以降の空間が空き、/M オプションで UMB 領域に設定できるようになります。省略時はスワップしません。

書式　/SW1
例）内蔵ハードディスクの BIOS（D0000h～D3FFFh）を移動し、そこを
　UMB 領域として使います。
　　DEVICE=MELEMM.386 /SW1 /M D0,D4,D8,DC

【注意】
ノートパソコンでは、/SW1 オプションを設定してはいけません。RAM ドライブの BIOS が移動してしまい、パソコンが動作しなくなることがあります。なお、IDE ハードディスクの BIOS は移動できません。

  /NC   EMS ファンクションの高速化

概要
EMS ページフレームの差し替え時に、一部の手順（EMS ページのデアロケート・リアロケート時に、メモリの初期化する機能）を省略し、EMS の動作（ファンクション No.6 と No.18）を速くします。省略時は高速化しません。

書式　/NC
例）
　　DEVICE=MELEMM.386 /NC

【注意】
一部の EMS 対応アプリケーションソフト（LOTUS 1-2-3 R2.1J など）は、/NC を使用すると動作できません。が、1995 年現在、このようなソフトウェアは殆どみられなくなりました。
  /XMS   XMS メモリ容量の上限設定

概要
/HM オプションを使用すると、MELEMM.386 は、すべてのプロテクト増設モードメモリを EMS でも XMS でも使用できる状態に整えます。ところで、MS-DOS Ver.5 の DOS シェルは、起動と同時にありったけのメモリを XMS として確保します。DOS シェル上で EMS 対応アプリケーションソフトを動かそうとしても、EMS メモリは残っていません。
/XMS オプションスイッチは、プロテクト増設モードから確保する XMS 拡張の容量の上限を指定します。すると、残りのメモリは EMS 拡張として残されます。
省略時は上限を設定しません。

書式　/XMS nnnn
nnnn=XMS メモリの上限容量（KB 単位、16KB 単位で切り捨て）
例）
　プロテクト増設モードメモリのうち、2048KB を XMS メモリとし、残り
　を EMS メモリに設定します。
　　DEVICE=MELEMM.386 /HM /XMS 2048

【注意】
/HM オプションと併せて指定すること。

  /T MS-DOS Ver.5.0 の拡張タスクスワップ機能指定

概要
MS-DOS Ver.5 の DOS シェルで、拡張タスクスワップ機能を使用するときに、/T オプションを使用します。/T オプションでは、拡張タスクスワップ機能を提供するデバイスドライバ（MS-DOSVer.5.0A では EXTDSWAP.SYS）をパスを含めて指定します。

書式　/T [PASU][デバイスドライバ名]
例）
　A:\DOS に EXTDSWAP.SYS が存在するとき
　　DEVICE=MELEMM.386 /HM /XMS 2048 /T A:\DOS\EXTDSWAP.SYS

【注意】
/HM および/XMS オプションと併せて指定すること。

/H nnn　EMSハンドル数の指定

概要
EMSハンドルとは、EMS対応アプリケーションソフトがEMSメモリを使うときに、メモリマネージャから渡される予約番号です。デフォルトでは予約番号は64とおり用意されています。EMS対応アプリケーションソフトが一度に64以上の予約が必要なとき、/Hオプションでハンドル数を増加します。

書式　/H nnn
nnn=EMSハンドル数（10進数で2〜255、デフォルト値は64）
例）
EMSハンドル数を128にします。
DEVICE=MELEMM.386 /H 128

【注意】
ハンドル数を大きくすると、EMSの動作が若干遅くなります。

/BE　仮想86バンクエミュレーション

概要
仮想86EMSメモリの一部を、バンク切換えメモリとして設定します。MS-DOSから見た場合、IOポート00Echにバンク番号を出力することで、バンクメモリにアクセスできます。/BEオプションを使用すると、旧来のバンク切換えメモリ対応アプリケーションソフトやデバイスドライバが利用可能になります。省略時はバンク切換えメモリを設定しません。

書式　/Bennnn
nnnn=バンク切換えとして利用するバンク数。1バンク=128KB（バンク単位、10進数）
例）
プロテクト増設モードメモリから1024KB（8バンク）をバンク切換えメモリに設定します。
DEVICE=MELEMM.386 /BE8

【注意】
実際に存在するプロテクト増設モードメモリより大きい容量をバンク切換えメモリに指定すると、EMS（またはXMS）拡張で使用できるメモリが無くなるので、ご注意ください。
/BEオプションで指定したバンクメモリをRAMディスクやキャッシュディスクで使用するときは、MELWARE Ver.5付属のデバイスドライバを利用してください。

/CX Cx486DLC/SL用キャッシュコントローラの動作指定

概要
80286または386を搭載したパソコンに、Cx486DLC/SLを採用したCPUアクセラレータを取りつけるときに、/CXオプションを指定します。Cx486DLC/SLのCPUキャッシュが有効になります。デフォルトでCPUにキャッシュされる領域は、以下のとおり。省略時はCPUキャッシュは無効です。
＜ノーマルモード＞
8086で管理できる1MBの空間の内、VRAM(A0000h〜BFFFFhとE0000h〜E7FFFh)及びハードディスクBIOS領域(D0000h〜DFFFFh)を除く部分。
＜ハイレゾモード＞
8086で管理できる1MBの空間の内、VRAM(C0000h〜DFFFFhとE0000h〜E7FFFh)及びハードディスクBIOS領域(E8000h〜EFFFFh)を除く部分。

書式　/CX [サブオプション] [サブオプション]
サブオプション
CCR0=03:640KB〜1MBの空間、および1MBごとの先頭64KBを、CPUにキャッシュしません。この領域をキャッシュすると、一部のアプリケーションソフトで動作が不安定になる場合がある為です。
NCR1=00F00,128K：絶対アドレスF00000hから128KBの領域を、CPUにキャッシュしません。この領域はNEC製のウィンドウアクセラレータなどが使用しているとき、このオプションスイッチ指定すると動作が安定します。
例）
Cx486DLC/SLを採用したCPUアクセラレータを取りつけるとき
DEVICE=MELEMM.386 /CX
CPUアクセラレータを使うと動作が不安定になるときの対策
DEVICE=MELEMM.386 /CX /CCR0=3 NCR1=00F00,128K

概要
MELEMM.383の管理するXMS拡張メモリからディスクキャッシュを配分し、ハードディスクの読みだし／書き込みを高速化します。CONFIG.SYSにデバイスドライバ登録するか（形態①）、またはMS-DOSの入力モードから起動する（形態②）方法があります。

条件
XMS拡張メモリが存在すること。MELEMM.386には、次のオプションスイッチが必要です。
```
DEVICE=MELEMM.386 /HM
```

【注意】
フロッピディスク、MOディスクには未対応です。256バイトセクタフォーマットのハードディスク、PCMCIAのハードディスクにも未対応です。

使用法①
形態
デバイスドライバ。MS-DOSのCONFIG.SYSにデバイス登録して使用します。
書式
DEVICE=HYPERDSK.EXE [オプション1] [オプション2] [オプション3]
例）
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048 CW:1024 S
```

使用法②
形態
MS-DOS上のコマンド。MS-DOSの入力モード（A>）からコマンド入力して使用します。
書式
HYPERDSK.EXE [オプション1] [オプション2] [オプション3]
例）
```
DEVICE=MELEMM.386 /HM
A>HYPERDSK.EXE C:2048 CW:1024 S
```

オプションスイッチ

C:	MS-DOS使用時のディスクキャッシュ容量
CW:	WINDOWS動作時のディスクキャッシュ容量
S	遅延書き込み指定
W	リアルタイム書き込み指定
E	ディスクキャッシュ機能の有効化
D	ディスクキャッシュ機能の無効化
T	遅延書き込みの遅延時間設定

C:　MS-DOS使用時のディスクキャッシュ容量

概要
MS-DOS動作時のディスクキャッシュ容量を指定します。省略時は、XMS拡張メモリの約50%を確保します。

書式　C:nnnn
nnnn=MS-DOS動作時のディスクキャッシュ容量。（10進数、KB単位）
例）2MBのディスクキャッシュを設定します。
①デバイスドライバとして使用するとき
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048
```
②MS-DOS上のコマンドとして登録するとき
```
DEVICE=MELEM.386 /HM
A>HYPERDSK C:2048
```

CW:　WINDOWS動作時のディスクキャッシュ容量

概要
WINDOWS動作時のディスクキャッシュ容量を指定します。WINDOWSはできるだけ多くのXMS拡張メモリを必要とします。CW:オプションは、WINDOWS起動時に、ディスクキャッシュとして確保したメモリの一部をXMS拡張メモリに解放する為に使用します。省略時は0KB。C:オプションより大きい値は設定できません。

書式　CW:nnnn
nnnn=WINDOWS動作時のディスクキャッシュ容量。（10進数、KB単位）
例）MS-DOS起動時は2MB、WINDOWS起動時は1MBのディスクキャッシュを確保します。
①デバイスドライバとして使用するとき
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048 CW:1024
```
②MS-DOS上のコマンドとして登録するとき
```
DEVICE=MELEM.386 /HM
A>HYPERDSK C:2048 CW:1024
```

【注意】
CW:オプションを未設定時は、WINDOWS起動時にディスクキャッシュは確保されません。

S　遅延書き込み指定

概要
ハードディスクに書き込みが発生したとき、書き込む情報をいったんメモリにキャッシュし、後でハードディスクへの書き込み作業を行います。ライトバックキャッシュともいいます。省略時は遅延書き込みを行いません。

書式　S
例）遅延書き込みを指定します。
①デバイスドライバとして使用するとき
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048 CW:1024 S
```
②MS-DOS上のコマンドとして登録するとき
```
DEVICE=MELEM.386 /HM
A>HYPERDSK C:2048 CW:1024 S
```

【注意】
画面上はハードディスクへの書き込みが終了していても、実際には遅延書き込みが行われているときがあります。パソコンの電源を切る直前には、STOPキーを押して、書き込みが完全に終了したことを確認すること。

W　リアルタイム書き込み指定

概要
ハードディスクへの書き込みが発生した毎に、実際の書き込みを行います。（遅延書き込みを行わない）省略時は、遅延書き込みを行いません。

書式　W
例）遅延書き込みを行いません。
①デバイスドライバとして使用するとき
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048 CW:1024 W
```
②MS-DOS上のコマンドとして登録するとき
```
DEVICE=MELEM.386 /HM
A>HYPERDSK C:2048 CW:1024 W
```

【注意】
Sオプションとは、併用できません。

E　ディスクキャッシュ機能の有効化

概要
ディスクキャッシュ機能を有効にします。省略時はディスクキャッシュは有効です。

書式　E
例）ディスクキャッシュを有効にする
①デバイスドライバとして使用するとき
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048 CW:1024 E
```
②MS-DOS上のコマンドとして登録するとき
```
DEVICE=MELEM.386 /HM
A>HYPERDSK C:2048 CW:1024 E
```

【注意】
Dオプションとは、併用できません。

D　ディスクキャッシュ機能の無効化

概要
ディスクキャッシュ機能を無効にします。HYPERDSK.EXEをEオプションを用いてコマンド入力すれば、再び有効にできます。省略時はディスクキャッシュは有効です。

書式　D
例）ディスクキャッシュを無効にする
①デバイスドライバとして使用するとき
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048 CW:1024 D
```

② MS-DOS 上のコマンドとして登録するとき
```
DEVICE=MELEM.386 /HM
A>HYPERDSK C:2048 CW:1024 D
```

【注意】
E オプションとは、併用できません。

T　遅延書き込みの遅延時間設定

概要
遅延書き込みが設定されているとき、キャッシュメモリに書き込みが開始されてから、実際ハードディスクへの書き込みが発生するまでの、時間差を指定します。

書式　T:n
n=0〜30（デフォルト値は 1、1 は約 0.5 秒）
例）約 1 秒遅れて、ハードディスクへの書き込みを行います。
①デバイスドライバとして使用するとき
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE C:2048 CW:1024 T:2
```
② MS-DOS 上のコマンドとして登録するとき
```
DEVICE=MELEM.386 /HM
A>HYPERDSK C:2048 CW:1024 T:2
```

【注意】
W オプションとは併用できません。

概要
MELEMM.383 の管理する XMS 拡張メモリ（または EMS 拡張）から RAM ディスクを配分し、高速なディスクドライブを生成します。CONFIG.SYS にデバイスドライバ登録するか（形態①）、または MS-DOS の入力モードから起動する（形態②）方法があります。

条件
MELEMM.386 が登録されていること。
```
DEVICE=MELEMM.386 /HM
```

【注意】
MS-DOS 上のコマンドとして利用するときは、CONFIG.SYS に次の設定が必要です。
LASTDRIVE=n　n=A〜Z（RAM ディスクのドライブ名が含まれる値にすること）

使用法①
形態
デバイスドライバ。MS-DOS の CONFIG.SYS にデバイス登録して使用します。
書式
DEVICE=EXDISK.EXE [X または E][容量]/[ディレクトリ数]
例）
```
DEVICE=MELEMM.386 /HM
DEVICE=HYPERDSK.EXE X 2048 /128
```

オプションスイッチ
X または E　使用するメモリ領域の指定
容量　RAM ディスクの容量指定
ディレクトリ数　ルートディレクトリ数の指定

X または E　使用するメモリ領域の指定

概要
RAM ディスクとして使用する拡張メモリの領域を指定します。

書式　X=XMS 拡張メモリ　E=EMS 拡張メモリ
省略時は XMS 拡張メモリを指定
例）XMS 拡張メモリから 2MB を RAM ディスクに指定
```
DEVICE=MELEMM.386 /HM
DEVICE=EXDISK.EXE X 2048
```

【注意】
X 指定時は、MELEMM.386 に/HM オプションが必要です。

容量　RAM ディスクの容量指定

概要
RAM ディスクとして使用するメモリ容量を指定します。

書式　nnnn
nnnn=メモリ容量。10 進数。KB 単位。
　例）4MB を RAM ディスクに指定します。
```
DEVICE=MELEMM.386 /HM
DEVICE=EXDISK.EXE 4096
```

【注意】
E オプション併用時に容量指定を省略すると、メモリが確保されません。X オプションを併用時に容量指定すると、すべての XMS メモリが確保されます。

ディレクトリ数　ルートディレクトリ数の指定

概要
RAM ディスクのルートディレクトリの数を指定します。省略時は 128。

書式　/nnn
nnn=ルートディレクトリ数。10 進数。32～480。
　例）RAM ディスク(4MB)のルートディレクトリ数を 256 にします。
```
DEVICE=MELEMM.386 /HM
DEVICE=EXDISK.EXE 4096/256
```

使用法②
形態
　MS-DOS 上のコマンド。MS-DOS の入力モード（A>）からコマンド入力して使用します。
書式
　HYPERDSK.EXE [X または E][容量]/[ディレクトリ数] [オプション 1] [オプション 2] [オプション 3]
　　例）
```
DEVICE=MELEMM.386 /HM
A>EXDISK.EXE X 2048/128
A>EXDISK R
```

オプションスイッチ
X または E	使用するメモリ領域の指定
容量	RAM ディスクの容量指定
ディレクトリ数	ルートディレクトリ数の指定
R	コマンド起動による RAM ディスクの常駐解除
V	RAM ディスクの状態表示
M	RAM ディスクの容量変更
C	RAM ディスクの内容クリア

X または E　使用するメモリ領域の指定

概要
RAM ディスクとして使用する拡張メモリの領域を指定します。

書式　X=XMS 拡張メモリ　E=EMS 拡張メモリ
省略時は XMS 拡張メモリを指定します。
　例）XMS 拡張メモリから 2MB を RAM ディスクに指定します。
```
DEVICE=MELEMM.386 /HM
A>EXDISK.EXE X 2048
```

【注意】
X 指定時は、MELEMM.386 に/HM オプションが必要です。

容量　RAM ディスクの容量指定

概要
RAM ディスクとして使用するメモリ容量を指定します。

書式　nnnn

nnnn=メモリ容量。10進数。KB単位。
　例）4MBをRAMディスクに指定します。
　　DEVICE=MELEMM.386 /HM
　　A>EXDISK.EXE 4096

【注意】
Eオプション併用時に容量指定を省略すると、メモリが確保されません。Xオプションを併用時に容量指定すると、すべてのXMSメモリが確保されます。

　ディレクトリ数　　ルートディレクトリ数の指定

概要
RAMディスクのルートディレクトリの数を指定します。省略時は128。

書式　/nnn
nnn=ルートディレクトリ数。10進数。32～480。
　例）RAMディスク(4MB)のルートディレクトリ数を256にします。
　　DEVICE=MELEMM.386 /HM
　　A>=EXDISK.EXE 4096/256

　R　　コマンド起動によるRAMディスクの常駐解除

概要
コマンド入力で作成したRAMディスクを、解除します。これにより、RAMディスクのドライブは消去されます。省略時は解除しません。

書式　R
例）RAMディスクを解除します。
　　A>EXDISK R

　V　　RAMディスクの状態表示

概要
RAMディスクの状態を表示します。省略時は表示しません。

書式　V
例）RAMディスクの状態を表示します。
　　A>EXDISK V

　M　　RAMディスクの容量変更

概要
RAMディスクのメモリ容量を変更します。省略時は変更しません。

書式　Mnnnn
nnnn=RAMディスクのメモリ容量。10進数。KB単位。
　例）RAMディスクのメモリ容量を2048KBに変更します。
　　A>EXDISK.EXE M2048

【注意】
MS-DOS起動後、最初に確保したRAMディスクの容量より増やすことはできません。RAMディスク容量を変更すると、RAMディスクの中のファイルがクリアされることがあります。

　C　　RAMディスクの内容クリア

概要
RAMディスクの内容をクリアします。省略時はクリアしません。

書式　C
　例）RAMディスクの内容をクリアします。
　　A>EXDISK C

概要
本来640KBのコンベンショナルメモリに常駐するデバイスドライバを、MELEMM.386の/Mオプションで確保したUMB領域に移動します。

条件
MELEMM.386 の/M オプションで確保した UMB 領域が存在すること。

【注意】
一部のデバイスドライバでは、動作できないことがあります。特に、常駐サイズよりも実行サイズが大きいデバイスドライバ（FEP などに多い）は、動作できないことが多くあります。

使用法
形態
　デバイスドライバ。MS-DOS の CONFIG.SYS にデバイス登録して使用します。
書式
　DEVICE=UMBLOAD.SYS [/UA:n] [移動するデバイスドライバ名]
 [移動するデバイスドライバのオプションスイッチ]
例）HYPERDSK.EXE を UMB 領域に移動します。
　　DEVICE=MELEMM.386 /HM
　　DEVICE=UMBLOAD.SYS HYPERDSK.EXE C:2048 CW:1024 S

オプションスイッチ
　/UA:移動先 UMB 領域の指定

　/UA:　移動先 UMB 領域の指定

概要
デバイスドライバの移動する UMB 領域のアドレスを指定します。

書式　/UA:n
n=UMB 領域の番号。10 進数。1～n。番号についての詳細は、A>UMBLOAD で参照できます。
　例）HYPERDSK.EXE を UMB 領域の No.1 に移動します。
　　DEVICE=MELEMM.386 /HM
　　DEVICE=UMBLOAD.SYS /UA:1 HYPERDSK.EXE C:2048 CW:1024 S

【注意】
UMB 領域のアドレス、および使用可能な容量は、A>UMBLOAD などで参照できます。本オプションは OPTIMIZE.COM 実行によって作成されたときのみ、使用すると良いのです。

概要
本来 640KB のコンベンショナルメモリに常駐する小型のコマンド（TSR）を、MELEMM.386 の/M オプションで確保した UMB 領域に移動します。

条件
MELEMM.386 の/M オプションで確保した UMB 領域が存在すること。

【注意】
一部のコマンドでは、動作できないことがあります。

使用法
形態
　MS-DOS 上のコマンド。MS-DOS の入力モード（A>）からコマンド入力
　して使用します。
書式
　UMBLOAD [/UA:n] [移動するコマンドのファイル名] [移動するファ
　イルのオプションスイッチ]
　　例）EXDISK.EXE を UMB 領域に移動します。
　　DEVICE=MELEMM.386 /HM
　　A>UMBLOAD EXDISK.EXE X 2048/128

オプションスイッチ書式
/UA:移動先 UMB 領域の指定
　/UA:　　移動先 UMB 領域の指定

概要
デバイスドライバの移動する UMB 領域の番号を指定します。

書式　/UA:n
n=UMB 領域の番号。10 進数。1～n。番号についての詳細は、A>UMBLOAD
で参照できます。
　例）EXDISK.EXE を最も下位のアドレスに移動します。

```
DEVICE=MELEMM.386 /HM
A>UMBLOAD /UA:1 EXDISK.EXE X 2048/128
```

【注意】
UMB領域のアドレス、および使用可能な容量は、A>UMBLOADで参照できます。本オプションはOPTIMIZE.COM実行によって作成されたときのみ、使用すると良いのです。
UMBLOAD.COMによって常駐したコマンドは、そのコマンドを終了することで常駐解除できます。複数のプログラムを常駐させた場合、常駐したときの逆の順番で解除すること。

概要
UMB領域やEMS領域の使用状態を表示します。

条件
MELEMM.386が設定されていること。

使用法
形態
  MS-DOS上のコマンド。MS-DOSの入力モード（A>）からコマンド入力して使用します。
書式
  UMBLOAD [オプション]
例）
```
A>UMBLOAD /E
```

オプションスイッチ
```
/E      EMS拡張の状態表示
/L      UMB領域とコンベンショナルメモリのリンク／リンク解除
/M      コンベンショナルメモリとUMB領域のメモリマップ表示
/U      UMB領域のメモリマップ表示
/V      デバイスドライバの常駐状況表示
```

  /E   EMS拡張の状態表示

概要
EMS拡張メモリの量（ページ数）、EMSページフレームの状態、EMS領域の予約状態などを表示します。XMS拡張が確保され²õているかどうかも参照できます。

書式　/E
例）EMS拡張の状態を表示します。
```
A>UMBSTAT /E
```

  /L UMB領域とコンベンショナルメモリのリンク／リンク解除

概要
UMB領域とコンベンショナルメモリを連結する／連結解除します。例えば、コンベンショナルメモリの空き容量がわずかなとき、UMB領域と連結すれば、MS-DOSプログラムをコンベンショナルメモリとUMB領域にまたがって常駐させることができます。ただし、現実的にこのような動作モードに対応したアプリケーションソフトは存在していません。

書式　/L
初回実行時、連結します。
連結時に実行すると、連結解除します。
  例）
```
A>UMBSTAT /L
```

  /M コンベンショナルメモリとUMB領域のメモリマップ表示

概要
コンベンショナルメモリやUMB領域について、どこがどれだけ使用されているか、を表示します。

書式　/M
例）
```
A>UMBSTAT /M
```

  /U UMB領域のメモリマップ表示

概要

UMB領域について、どこがどれだけ使用されているか、を表示します。

書式　/U
例）
　　A>UMBSTAT /U

/V　デバイスドライバの常駐状況表示

概要
常駐しているデバイスドライバの種類とアドレスを表示します。

書式　/V
例）
　　A>UMBSTAT /V
概要
パソコンの状態に併せて、最も効果的にUMB領域を活用するように、UMBLOAD.SYSやUMBLOAD.COMを設定するプログラムです。UMB領域が不連続なときは、サイズの大きなプログラムを連続したUMB領域へ、小さいプログラムは不連続な領域へ、というように、移動先を指定してくれます。

条件
MELEMM.386が設定されており、UMB領域が存在すること。OPTIMIZE.COMと同じディレクトリに、DOS.CMSが必要です。

【注意】
変更前のCONFIG.SYS/AUTOEXEC.BATは、CONFIG.OPT/AUTOEXEC.OPTに保存されます。カードサービス（SC.EXE）を使用しているときは、実行してはなりません。
実行中は3度のリセットがかかります。詳細は、「4-:-:オプティマイザのしくみ」を参照。

使用法
形態
　MS-DOS上のコマンド。MS-DOSの入力モード（A>）からコマンド入力
　して使用します。
書式
　A>OPTIMIZE
例）ドライブAに対して最適化を実行します。
　　A>OPTIMIZE

オプションスイッチ
　　[対象ドライブ]　最適化する対象ドライブ

[対象ドライブ]　　最適化する対象ドライブ

概要
最適化する対象ドライブを指定します。省略時は、カレントドライブを最適化します。

書式　[対象ドライブ]
例）ドライブBに対して最適化を実行します。
　　A>OPTIMIZE B

概要
プロテクト増設モードメモリにハード的な故障がないか検査します。メモリをチェックする際に、検査データをメモリに書き込み、書き込み内容に誤りがないかどうか比較（ベリファイ）を行います。

条件
i386以上を搭載したパソコンで、プロテクト増設モードメモリが存在すること。そのメモリが、EMS拡張やXMS拡張で使用されていないこと。

【注意】
「XMS又はEMSドライバの常駐を解除して実行してください。」とエラーがでたら
本プログラムは、XMSまたはEMSメモリマネージャが存在しているときは実行できません。CONFIG.SYSから次の行を解除し、リセット後、再度実行すること。
DEVICE=MELEMM.386
メモリのチェック中にエラーが発生するときは、メモリボードの装着に誤りがあるか、または故障している可能性があります。装着方法に誤りが確認できなければ、インフォメーションセンターへご連絡ください。

使用法
形態

MS-DOS上のコマンド。MS-DOSの入力モード（A>）からコマンド入力して使用します。

書式

MEMCHK

例）

```
A>MEMCHK
```

概要

パソコンの基本的な性能を計測します。CPUアクセラレータなどでシステムを強化したときに、参考にすると便利。結果は画面にグラフ表示され、ファイルへの保存も可能。測定できる性能は下記のとおりで、いずれも計測値が大きいほど高性能であることを意味します。

- CPU性能
- メモリアクセス性能
- ハードディスクアクセス性能

条件

i386以上（CPUアクセラレータでも可）を搭載したパソコンであること。

使用法

形態

MS-DOS上のコマンド。MS-DOSの入力モード（A>）からコマンド入力して使用します。

書式

SYSBMK [オプション] [オプション]

例）

```
A>SYSBMK A:
```

オプションスイッチ

N	モノクロディスプレイ対応
[ドライブ名:]	計測するドライブ名

N: モノクロディスプレイ対応

概要

ノートパソコンなどモノクロディスプレイの場合に指定します。省略時はカラーディスプレイ対応。

書式 N

例）

```
A>SYSBMK N
```

[ドライブ名:] 計測するドライブ名

概要

性能を計測するドライブ名を指定します。省略時はカレントドライブ。

書式 [ドライブ名:]

例）ドライブAを計測します。

```
A>SYSBMK A:
```