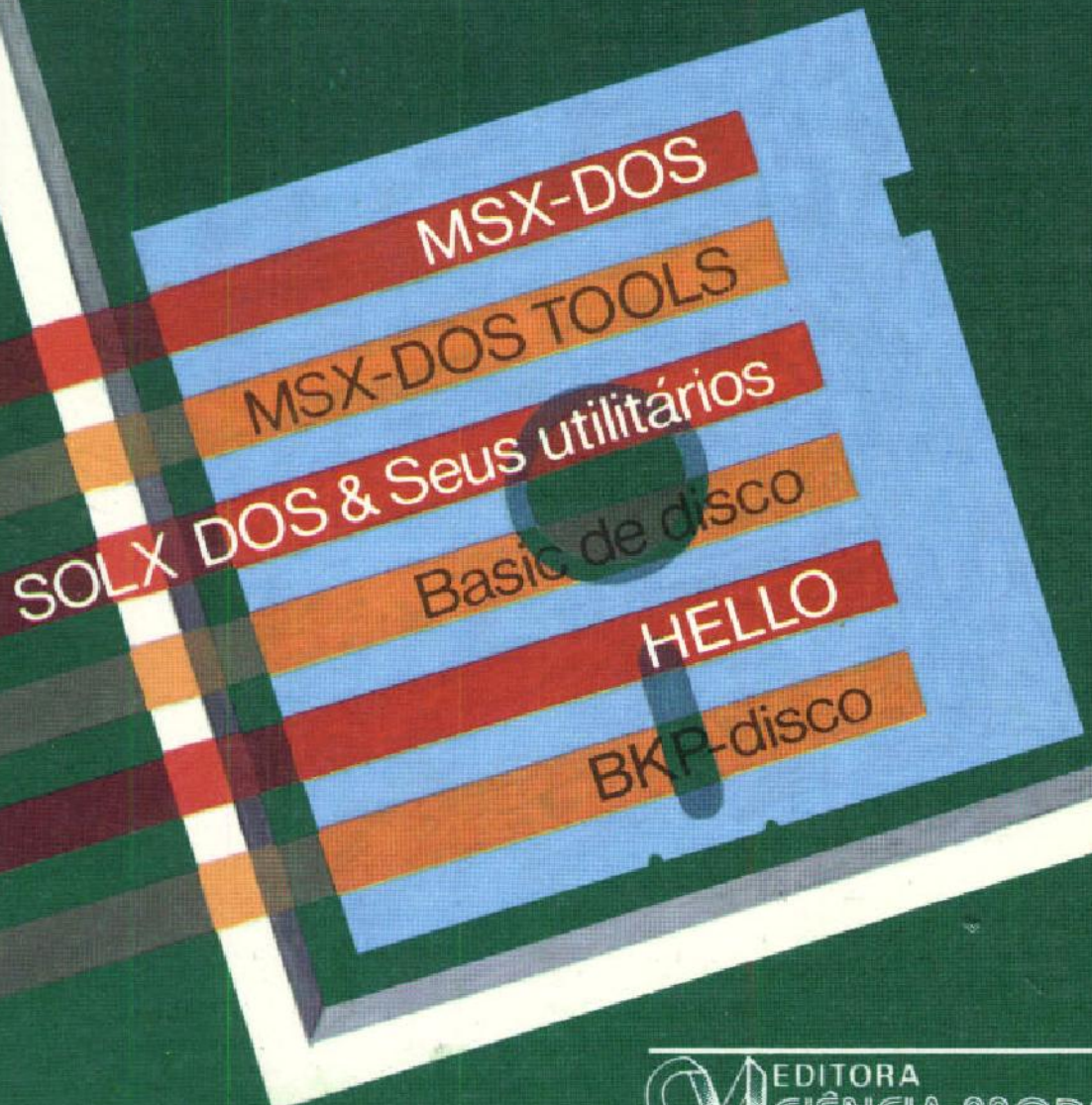
sistemas operacionais do MSX & suas ferramentas

Sérgio Guy Pinheiro Elias
Paulo Roberto Pinheiro Elias



EDITORA CIÊNCIA MODERNA

SISTEMAS OPERACIONAIS DO MSX E SUAS FERRAMENTAS

AUTORES: SERGIO GUY PINHEIRO ELIAS

PAULO ROBERTO PINHEIRO ELIAS

EDITORA CIÊNCIA MODERNA LTDA

SISTEMAS OPERACIONAIS DO MSX E SUAS FERRAMENTAS.

Sergio Guy Pinheiro Elias
Paulo Roberto Pinheiro Elias.

Editor: Paulo André P. Marques

Revisão: Carlos Augusto L. Almeida

Composição e Diagramação: ECM.

Capa: Renato Martins

FICHA CATALOGRÁFICA

Elias, Sergio Guy Pinheiro e Elias, Paulo Roberto Pinheiro.

SISTEMAS OPERACIONAIS DO MSX E SUAS FERRAMENTAS - Sergio Guy Pinheiro Elias e Paulo Roberto Pinheiro Elias - Rio de Janeiro: EDITORA CIÊNCIA MODERNA LTDA. - 1989.

1-Sistemas Operacionais do MSX(Programa de Computador).I. Título

CDD.001.642

EDITORA CIÊNCIA MODERNA LTDA.
Av. Rio Branco, 156 sala 713
20043 Rio de Janeiro - RJ.
TEL. 262.3111 / 533.0300

PREFÁCIO

Um microcomputador sem um Sistema Operacional de Disco é como um carro sem gasolina: se você fizer força para empurrá-lo, poderá até ir longe com ele, mas o percurso será penoso e desgastante.

O objetivo deste livro é introduzir o leitor ao potencial dos Sistemas de Disco usados no MSX, dando destaque ao MSX-DOS e às suas ferramentas, e terminando com uma descrição do BASIC DE DISCO. Fizemos um esforço enorme para que o livro seja legível pelo usuário sem nenhum conhecimento nesta área e que, posteriormente, sentirá necessidade de aprofundar seus conhecimentos e sua utilização do computador com o disk drive. Em muitos trechos desta obra, são feitas comparações com a máquina padrão MSX (capaz de ser autônoma apenas com um gravador cassete), de modo que o leitor possa sentir as diferenças entre as duas configurações (com e sem drive) e possa se adaptar melhor ao novo Sistema Operacional adotado.

No passado recente, bons livros sobre drives foram editados em número expressivo, alguns dos quais serviram de referência ao nosso trabalho. Tiramos grande proveito em sermos o último da lista, já que isto nos permitiu fazer uma análise mais tranquila da evolução do MSX no mercado, longe do "oba-oba" que marcou o lançamento dos drives, em fins de 1986 e início de 1987. O resultado foi que tivemos chance de produzir um manuscrito mais voltado ao que presumimos ser a realidade do usuário, com relação aos softwares que vêm sendo produzidos por dedicação e interesse de programadores e software houses, que querem ver no MSX uma máquina para aplicações profissionais: a informação mais básica deve ser do conhecimento daqueles que pensam em usufruir do potencial que programas mais sofisticados lhes oferecem.

Infelizmente, é triste de se constatar um grande desestímulo empresarial nos setores que impulsionaram o MSX no início e que hoje praticamente abandonaram o padrão. A MICROSOL TECNOLOGIA, por exemplo, empresa que lançou o primeiro drive, abandonou um projeto já em desenvolvimento, sobre o SOLX-DOS, Versão 2.0, o qual teria constado deste livro, se tal não acontecesse.

A posição da MICROSOL é compreensível, em função da pirataria desenfreada que seus produtos têm sofrido. De fato, não interessa ao empresário investir tempo e dinheiro em projetos complicados, sem que haja um retorno justo. Aliás, a pirataria já está tão institucionalizada, que alguns termos como "clone", "padrão", etc., já são usados como eufemismo para caracterizar o produto pirateado.

Em vista disso, e em função de um iminente desaparecimento dos micros de 8 Bits, é de se admirar que ainda haja tanta gente envolvida com o MSX, ainda por cima, pessoas sérias e competentes. O número de programas-ferramentas vem aumentando de forma impressionante nos últimos meses. Os programadores, malgrado as poucas divulgações técnicas do MSX, têm virado o micro de cabeça para baixo e, com isso, alcançando objetivos até pouco tempo atrás inconcebíveis. Um exemplo muito claro disto é o programa FAST!COPY, produzido pela PAULISOFT, capaz de copiar fisicamente um disquete de face dupla de 5 1/4" em 35 segundos.

Em nosso livro, tivemos que selecionar nestes ultimos meses os utilitários que serviriam de referência para exemplificar as chamadas "ferramentas do DOS". Começamos pela MICROSOL, com inteira justiça, que produziu dois utilitários muito bons, dos quais se fala muito pouco. Depois, são explorados os MSX-DOS TOOLS, da ASCII japonesa, distribuídos por algumas software houses, sem qualquer tipo de suporte, por se tratarem de programas sem representação no Brasil (pelo menos, que seja do nosso conhecimento). Decidimos incluí-los por representarem corretamente o conceito de comandos externos do DOS.

As estrelas do show, entretanto, são dois multi-utilitários poderosos, dignos representantes da produção nacional e que, por isso mesmo, mereceram Capítulos separados: o HELLO, de Eduardo Barbosa, e o BKP DISCO, de Júlio Velloso, distribuídos pelas software houses NEMESIS e PAULISOFT.

A apresentação destes programas, além de mostrar o que há de melhor no momento, ainda nos serviu para poder explorar, com o leitor, os conceitos e aplicações mais importantes dos Sistemas de Disco do MSX.

O livro começa pelos princípios e fundamentos que o leitor deverá estudar, antes de partir para a utilização plena do seu Sistema Operacional.

O MSX-DOS foi a base da qual partimos para estudar os comandos do DOS, tendo sido incluídos também os comandos do SOLX-DOS. Depois de apresentados, em vários Capítulos, os programas utilitários, passamos a destacar o BASIC DE DISCO do MSX, sendo este o único Capítulo no qual partimos da premissa de que o leitor já deverá ter obtido uma noção, ainda que pequena, de programação no BASIC convencional da máquina.

Propositalmente, deixamos de lado os comandos do CP/M, pelos seguintes motivos: o MSX-DOS foi projetado pela MICROSOFT para ser compatível com o CP/M Versão 2.2. Esta compatibilidade é satisfatória para rodar os vários aplicativos originários do ambiente CP/M, como o dBASE, SuperCalc 2, WORDSTAR e outros. O CP/M pos-

sui uma infinidade de livros dedicados a ele, inclusive para o MSX. A maioria dos fabricantes de controladoras adotou o "padrão" da MICROSOL, no qual o CP/M não foi incluído.

Outra limitação proposital foi a ausência quase absoluta dos drives de 3 1/2", muito embora a maioria do conteúdo deste livro seja altamente aplicável para este Sistema. E o motivo para isto é muito simples: não temos este drive e nem acesso a ninguém que pudesse nos ceder um, para os devidos testes.

Finalmente, esperamos que nossos leitores não tenham dificuldade em compreender o texto e possam se beneficiar deste livro por muito tempo. Sabemos que os Sistemas de Disco são o futuro da informática. Dentro em breve, o CD-ROM, com seus 600 Megabytes de memória, se transformará num CD gravável. Até lá, temos a esperança de que o nosso aprendizado de hoje não se torne obsoleto amanhã.

OS AUTORES.

AGRADECIMENTOS

Escrever um livro é uma tarefa das mais difíceis. Muitas pessoas colaboram, às vezes anonimamente, para que o livro saia com uma melhor qualidade. De nossa parte, queremos agradecer às seguintes pessoas, que nos ajudaram a produzir esta obra:

EDUARDO ALBERTO BARBOSA
engenheiro, programador e escritor, por ter-nos cedido muitas informações importantes e a sua solidariedade na redação deste livro.

JULIO RENATO SOARES VELLOSO
autor do BKP DISCO, por ter estudado conosco vários aspectos do Sistema Operacional, que ajudaram a enriquecer vários trechos do manuscrito.

MARCELO VALLE FRANCO
proprietário da NEMESIS INFORMÁTICA LTDA, pelo apoio no fornecimento de softwares que ajudaram na confecção do livro.

OSCAR JULIO BURD
Gerente de Informática e Telefonia da GRADIENTE ELETRÔNICA, pela remessa de informações que nos ajudaram, na fase final do livro, a corrigir o texto e acrescentar novos dados sobre o padrão MSX.

JULIO CESAR SANTOS DE FREITAS
Supervisor da GRADIENTE ELETRÔNICA, filial RIO, por ter assumido problemas insolúveis de manutenção no equipamento de um dos autores, o que possibilitou que este livro chegasse a termo.

VALDELIRIO PEREIRA SOARES FILHO
Diretor Comercial da MICROSOL TECNOLOGIA, pelo envio do material sobre MSX-DOS, que nos ajudou a entender muita coisa sobre a arquitetura interna do MSX.

DEDICATÓRIA

Da única herança verdadeira que podemos deixar aos nossos filhos e afilhados, um dos bens de maior valor é o exemplo da determinação, dedicação e honestidade para alcançar os nossos objetivos.

Este livro é dedicado, com estes propósitos, aos filhos que Deus nos deu e aos filhos dos quais somos padrinhos.

Às crianças:

MARCELO, TATIANA e GUSTAVO GHASTINE ELIAS
(por parte de Sergio Guy Pinheiro Elias)

PATRICIA, GUSTAVO e BERNARDO KREMER DINIZ GONÇALVES
(por parte de Paulo Roberto Pinheiro Elias)

Com todo o nosso carinho.

AVISOS GERAIS AOS LEITORES

Os comentários sobre os programas apresentados restringem-se às seguintes Versões:

MSX-DOS - Versão 1.03 (COMMAND Versão 1.11)
Obs.: a GRADIENTE ELETRÔNICA é atualmente a detentora da marca "MSX-DOS" e, em breve, deverá lançar nova Versão, cuja numeração desconhecemos.

SOLX-DOS - Versão 1.2.

HELLO e BKP DISCO - Versão 1.0
Obs.: novas Versões de ambos os programas estavam em preparação, quando este livro ficou pronto.

BASIC DE DISCO - Versão 1.0.

Os demais programas têm suas Versões especificadas no corpo do texto.

Os autores não se responsabilizam por discordâncias entre o texto do livro e eventuais modificações feitas nos programas comentados, já que seus programadores, como legítimos proprietários, têm o direito de proceder a quaisquer alterações, sem aviso prévio aos usuários.

O equipamento para avaliação do funcionamento dos programas comentados constou de: microcomputador EXPERT XP-800, Versão 1.1; interface controladora de drives CDX-2 e placa de 80 colunas VMX-80, ambas da MICROSOL TECNOLOGIA.

ACONSELHAMOS QUE SEJA LIDO E ESTUDADO O PRIMEIRO CAPÍTULO, SOBRE AS "NOÇÕES BÁSICAS" DOS SISTEMAS DE DISCO, ANTES DA LEITURA DE QUALQUER OUTRA PARTE DO LIVRO.

SUMÁRIO:

CAPÍTULO 2
COMANDOS DO MSX-DOS E SEUS COMPATÍVEIS.

CAPÍTULO 3
UTILITÁRIOS E FERRAMENTAS DO DOS - primeira PARTE: UTILITÁRIOS SOLX-DOS E MSX-DOS TOOLS

CAPÍTULO 5
UTILITÁRIOS E FERRAMENTAS DO DOS - terceira PARTE:
O SISTEMA MODULAR "BKP DISCO", VERSÃO 1.0.

CAPÍTULO 6
O BASIC DE DISCO DO MSX

CAPíTULO 1

SISTEMAS OPERACIONAIS DE DISCO: NOÇõES BÁSICAS

Introdução

Os microcomputadores são basicamente equipamentos para processar dados. Para que isto ocorra como se deseja, é necessário programá-los.

Ao fazer isso, o programador é obrigado a instruí-lo através de comandos dispostos dentro de uma certa seqüência.

Pela sua própria natureza, os computadores são capazes somente de entender a LINGUAGEM DE MÁQUINA, que é a maneira como os primeiros computadores eram programados.

Esta, por sua vez, corresponde à codificação de instruções estipuladas por seqüências de zeros (0) e uns (1), que, assim, melhor representam os estados elétricos de desligado (0 Volt) e ligado (1 Volt). Isto tornava a programação lenta e principalmente complicada.

Com a evolução dos computadores, apareceram as linguagens cujos comandos estavam mais próximos do usuário, as chamadas LINGUAGENS DE ALTO NÍVEL. Como os computadores só estão capacitados a entender a linguagem de máquina (de BAIXO NÍVEL), programas interpretadores ou compiladores tiveram que ser implementados para "traduzir" os comandos da linguagem de ALTO NÍVEL.

Assim, os interpretadores possibilitaram o uso de linguagens mais sofisticadas e dedicadas a uma determinada finalidade.

Com a entrada em cena dos interpretadores, chegamos ao conceito de que existem programas que são necessários para fazer o computador funcionar de modo a

receber outros programas.

O conjunto de programas capaz de fazer o computador operar é denominado de

SOFTWARE BÁSICO.

O que é o DOS ?

Baseados neste conceito, podemos dizer que o DOS é um SOFTWARE BÁSICO, pois permite ao usuário a realização de tarefas específicas, servindo de "intérprete" dos comandos por ele introduzidos no microcomputador.

Para sermos mais justos, diríamos que o DOS é não só um intérprete, mas também um GERENCIADOR do funcionamento do sistema de disco acoplado ao micro.

A denominação D.O.S. refere-se a

DISK OPERATING SYSTEM

(ou SISTEMA OPERACIONAL DE DISCO).

O DOS é um SISTEMA OPERACIONAL bastante sofisticado e eclético, pois permite:

1 - carregar no micro interpretadores de linguagens de alto nível (por ex.: Pascal).

2 - fazer rodar programas aplicativos de qualidade profissional, como Editores de Texto, Bancos de Dados,Planilhas Eletrônicas, etc.

3 - abrir portas de comunicação com o computador, unidades de disco e impressora, propiciando o GERENCIAMENTO de seus recursos.

Além disso, o Sistema Operacional DOS garante a compatibilidade de gravação e leitura de arquivos em disco entre todos os microcomputadores sob seu funcionamento.

O MSX-DOS, base para todo DOS utilizado no Padrão MSX, compõe-se de um programa em dois blocos:

MSXDOS.SYS

COMMAND.COM

O primeiro arquivo contém o programa principal enquanto que o segundo conterá a relação de comandos do Sistema. Ambos, principalmente o segundo, são permanentemente atualizados pela empresa que os criou, dando origem aos novos "releases" (novas Versões).

Isto faz com que novos comandos possam ser criados e os atuais aperfeiçoados, tornando este software cada vez mais versátil.

Carregamento do DOS:

O DOS exige uma configuração de memória RAM mínima de 64 Kbytes, o que não constitui problema em termos dos micros MSX nacionais. Exige também, pelo menos, UMA unidade de disco instalada.

Para que possa operar, o DOS deve ser carregado na memória RAM do computador, quando, então, assume o completo controle do seu funcionamento.

A entrada em operação do DOS modifica o status de memória livre do micro, liberando maior quantidade do que o habitual, o que permite, por seu turno, o carregamento de programas que exigem maior área de memória para funcionar.

Ao assumir o gerenciamento das operações de entrada e saída do micro, o DOS possibilita rodar programas que se beneficiam de seus recursos.

Costuma-se dizer que estes programas rodam sob o DOS ou "em AMBIENTE (operacional) DOS".

Ambiente DOS corresponde, portanto, às modificações das operações de entrada e saída, criadas pela presença do Sistema Operacional DOS.

Isto significa na prática ter um "novo" computador em operação.

Para o carregamento do DOS é necessário dar partida no computador, com o disquete contendo o programa alojado no **drive A.**

N.B.: existem duas maneiras de se dar partida no micro:

1 - partida a frio ("COLD BOOT"): é conseguida no ato de ligar ou religar o computador.

2 - partida a quente ("WARM BOOT"): é conseguida provocando-se um "RESET" (REINICIALIZAÇÃO) no micro, estando este já em funcionamento.

EXPERT 1.0: devem ser digitadas as seguintes instruções em BASIC:
DEFUSR=0 <RETURN>
A=USR(0) <RETURN>

EXPERT 1.1: pressionar simultaneamente as teclas <CONTROL> + <SHIFT> + <STOP>

HOTBIT: pressionar o botão vermelho lateral.

Observação: Os termos "RESETAR", "RESTARTAR", "REINICIALIZAR" e a expressão "dar um BOOT no Sistema" têm todos o mesmo significado, que é o ato de dar partida no micro. "RESETAR" vem do verbo "TO RESET" (recolocar em operação). "RESTARTAR" vem de "TO RESTART" (recomeçar), enquanto que "BOOT" se refere a uma forma de expressão inglesa, aproximadamente traduzida como "dar um pontapé inicial". Quando o computador é ligado, a rotina CHKRAM (abreviação de "CHECK RAM"), residente no BIOS (Basic Input/Output System) da ROM, é acionada. Esta rotina verifica ("checa") se a RAM está liberada para uso. Ao seu final, informa ao usuário o número de Bytes livres. O endereço de execução desta rotina é &H0000 (ou &H0). A qualquer momento, as instruções DEFUSR=0 e A=USR(0), anteriormente citadas, definem e executam, respectivamente, esta rotina, provocando assim um RESET no sistema.

Quando a partida é dada, o Sistema Operacional do micro procura no disquete o programa do DOS. Se este não for encontrado, será acionado o BASIC de DISCO.

Importante:

Por questões de padronização, todas as interfaces controladoras de acionadores de disco (disk drives) gravam dois nomes de DOS na rotina de partida, quando o disquete é formatado. O padrão MSX determina que

UM DELES SERÁ SEMPRE O MSX-DOS !

Os nomes constantes do diretório do disco são comparados com os da formatação. O DOS será carregado somente se houver coincidência com um dos dois.

Interfaces de diferentes fabricantes nomeiam o segundo DOS pelo título do seu próprio Sistema Operacional.

Na prática, quando se grava um determinado Sistema Operacional num disquete formatado pela interface de outro fabricante, não se consegue carregá-lo.

No entanto, ao mudar o nome do DOS para MSXDOS.SYS, a partida terá sucesso. Veja mais detalhes sobre este assunto no Tópico sobre Rotina de Partida, deste Capítulo e no Apêndice 1.

Entrada em operação do DOS:

Quando o DOS entra no ar, aparecem as mensagens do seu fabricante. No caso do MSX-DOS, será impresso algo deste tipo na tela:

```
MSX-DOS version 1.03
Copyright 1984 by Microsoft

Command version 1.11
Current date is Fri 25-04-1986
Enter new date:
```

A segunda mensagem, relativa ao arquivo de comandos (COMMAND.COM), não aparecerá se este não estiver no mesmo disquete. Isto geralmente acontece quando o usuário faz uma cópia incompleta do MSX-DOS. Quando tal fato ocorre, o MSX-DOS emite a mensagem de erro:

Insert DOS disk in default drive
and strike any key when ready

Traduzindo:
Insira disco do DOS no drive default
e tecle algo quando pronto.

A data "atual" ("Current date") enunciada pelo DOS é lida na interface controladora do drive. Como os micros MSX nacionais até o momento não dispõem de relógio interno, esta data "atual" será sempre repetida.

O leitor deve se acostumar a digitar a nova data, oportunidade esta que se apresenta quando o DOS entra no ar ("Enter new date: ").

Isto possibilitará registrar no diretório do disco a data em que as operações de gravação foram feitas.

O formato desta nova "data atual" é igual ao mostrado pela EPROM da interface:

dd-mm-aa
ou
dd/mm/aa

onde: **dd - dois dígitos para o dia**
mm - dois dígitos para o mês
aa - dois últimos dígitos do ano.

Serão aceitas as seguintes digitações: apenas um dígito para dd e mm, e quatro dígitos para aa (ex.: 1988).

Datas fora deste formato serão consideradas inválidas e a sua entrada será recusada.

Teclando-se <RETURN> ao invés de nova data, a anterior será mantida.

A linha de comando do DOS:

Ao entrar em operação, o DOS habilita o usuário a digitar um comando, exibindo um "PROMPT", que é o SINAL de que o computador está "PRONTO" a receber instruções.

O prompt do MSX-DOS será necessariamente da seguinte forma:

nome do drive -----> A> <------ sinal indicador da linha de comando(" > ")

O drive A, nomeado na linha de comando, é portanto o drive ASSUMIDO como INICIAL (drive DEFAULT).

Todas as operações a serem efetuadas pelo DOS referir-se-ão a este drive, a não ser que o usuário mencione outro !

Depois de iniciada a operação do DOS, poder-se-á mudar o drive assumido, mesmo que não haja nenhum outro drive fisicamente instalado.

Isto só é possível porque as interfaces controladoras de dois drives assumem que as operações possam ser feitas em um drive VIRTUAL, isto é, não existente fisicamente. Em outras palavras, a controladora SIMULA a existência de outro drive, possibilitando assim que certas operações do DOS, que pressupõem um segundo drive, possam ser realizadas sem problemas.

Importante:

Para qualquer efeito, a NOMEAÇÃO do drive pelo usuário deverá ser feita pela LETRA correspondente ao mesmo, seguida do sinal de dois pontos (A:, B:, etc.).

Assim, para mudar o drive default inicial para B, deve-se comandar:

A>B: <RETURN>
    O prompt na linha de comando mudará para:
B>

Isto indica que o drive CORRENTE é o B e as operações do DOS serão a ele referenciadas. O drive corrente poderá ser novamente o A, se assim for desejado.

Configuração das Unidades de Disco:

É fundamental, para o bom desempenho de qualquer Sistema Operacional, que as unidades de disco (disk drives) estejam instaladas no Sistema eletricamente configuradas como A e B.

A configuração poderá ser feita pelo usuário ou por um técnico especializado. O acesso a ela está, geralmente, no chamado "impresso-mãe" ou "placa-mãe" ("motherboard"), o circuito impresso principal do drive.

O procedimento em si é muito simples:

1 - Retira-se a tampa do gabinete onde o drive está parafusado. Todo o cuidado deve ser tomado para não danificar o aparelho !

2 - O impresso-mãe é facilmente reconhecível, por ser o maior circuito impresso existente no drive.

Obs.: "circuito impresso" é uma placa de fenolite ou fibra de vidro, onde são soldados os componentes do aparelho (capacitores, resistores, e outros). A placa tem o desenho do circuito e dos componentes impressos sobre a superfície superior.

Quando o impresso-mãe possui indicações mais detalhadas, a configuração do drive aparece com a inscrição "DRIVE SELECT".

O NOME do drive, neste caso, acompanha a terminologia padrão, substituindo-se as letras por NÚMEROS. A correspondência entre ambos é a seguinte:

0 - drive A
1 - drive B
} usados para os dois primeiros drives

2 - drive C
3 - drive D
} usados para a expansão do sistema

3 - Verifica-se a posição do estrape, que serve para selecionar o drive e muda-se se necessário. O conector é retirável puxando-o CUIDADOSAMENTE com a ponta dos dedos. O leitor deverá reparar que, existindo dois estrapes próximos um do outro, somente aquele que estiver na posição correspondente à numeração acima poderá ser retirado.

4 - Repõe-se imediatamente a tampa verificando se os cabos estão firme e corretamente instalados.

Instalação de um ou mais drives

A maioria das interfaces controladoras atualmente em uso permitem utilizar de um a dois drives de 5 1/4" ou 3 1/2", para disquetes de face dupla ou simples. Teoricamente, com dois drives conectados nos slots externos A e B do MSX, seria possível a operação de quatro drives simultaneamente, nomeados por A, B, C e D.

Isoladamente, as controladoras permitem o uso de apenas um drive para todas as operações. Este será necessariamente configurado eletricamente como drive A.

Quando, através de um comando, for necessário usar o drive B, o Sistema Operacional fará uma PAUSA na operação corrente e solicitará a INSERÇÃO DE DISQUETE COMO DRIVE B. Terminada esta operação, o drive CORRENTE será o B. Assim, se for necessário voltar ao A, o Sistema solicitará a INSERÇÃO DE DISQUETE COMO DRIVE A.

Em um ou outro caso, uma das unidades de disco é tratada como DRIVE LÓGICO.

Portanto, DRIVE LÓGICO é um drive SIMULADO pela controladora para indicar ao usuário, através de mensagens, a direção das operações que se utilizam de mais de um drive.

Dessa forma, é possível usar apenas um drive como se fosse A e B, alternadamente, muito embora o drive fisicamente instalado deverá ser o drive A.

No caso de dois drives fisicamente instalados, não haverá mais a simulação do drive lógico, a não ser que o usuário desligue a alimentação elétrica do drive B. O drive A não poderá ser desligado sob nenhuma hipótese.

Observações importantes:

Alguns programas em BASIC e LINGUAGEM DE MÁQUINA, para serem carregados na memória do micro, exigem a anulação dos drives lógicos (veja outros detalhes mais adiante).

Teclando-se <control> na partida do micro, somente os drives físicos se tornam

operantes. Esta tecla deve ser mantida pressionada até que os drives sejam inspecionados pela ROM. O computador emite um BEEP quando esta inspeção termina. Os drives físicos piscam os seus LEDS e as suas cabeças de leitura e gravação são estacionadas na trilha zero. Havendo necessidade de anular os drives lógicos, as respectivas alimentações elétricas devem ser desligadas. É altamente recomendável manter fontes e/ou interruptores separados para cada drive, caso contrário, esta tarefa será impossível.

N.B.: A ANULAÇÃO DE UM DRIVE LÓGICO IMPEDIRÁ TODAS AS OPERAÇÕES REFERENTES A ELE, SENDO O MESMO TRATADO PELA CONTROLADORA COMO "DRIVE INVÁLIDO".

A GRANDE MAIORIA DOS PROGRAMAS QUE RODAM EM AMBIENTE DOS, COM EXCEÇõES PROPOSITAIS, DISPENSAM A ANULAÇÃO DO DRIVE LÓGICO.

As conexões entre os drives e a interface deverão também merecer a atenção do leitor, no caso de serem instalados dois drives.

O cabeamento utilizado pelos fabricantes de interfaces controladoras possui dois terminais numa das extremidades, para a conexão dos drives, sendo INDIFERENTE ligá-los a um ou outro drive. É IMPERIOSO, entretanto, que cada drive esteja configurado diferentemente do outro, caso contrário, haverá funcionamento errôneo da controladora. Eis porque o leitor foi alertado, no item anterior, para verificar a configuração elétrica das suas unidades de disco ANTES de colocá-las em operação.

Apenas por convenção, no caso de existirem 2 drives, a unidade A ficará montada em CIMA da unidade B, ou à ESQUERDA da unidade B, se os drives forem montados em pé. Isto facilita a sua identificação imediata no momento do uso.

A instalação de dois drives torna as operações de cópia mais confortáveis e seguras, pela desnecessidade de trocar disquetes como drives A e B sucessivamente. Quando o usuário possui apenas um drive, deve proteger o disco fonte, para evitar que, num erro de trocas, os arquivos originais sejam perdidos.

Por outro lado, com dois drives físicos, a disponibilidade de memória periférica (fora do computador) permite a utilização mais eficiente de programas que processam seus arquivamentos criando arquivos temporários no disco. Neste caso, a limitação do tamanho do arquivo será diretamente proporcional ao espaço disponível nos dis-

quetes onde ele está sendo gravado. Os projetistas de softwares profissionais preferem este tipo de design, para contornar eventuais limitações de memória do próprio computador.

Limitações de memória no BASIC DE DISCO:

Como foi comentado anteriormente, a anulação do drive B, seja ele lógico ou físico, é determinada pela necessidade de contornar limitações no espaço de memória RAM do MSX, no ambiente do BASIC DE DISCO.

Praticamente todos os microcomputadores têm, em um ponto ou em outro, limitações deste tipo, a despeito de estarem implementados os chamados "expansores de memória".

O microprocessador Z80-A, por exemplo, só pode acessar diretamente 64 KBytes de memória de cada vez. Os construtores do padrão MSX colocaram 4 Slots Primários de 64 KB em paralelo para tentar contornar, por chaveamento interno via software, esta limitação de gerenciamento de memória da CPU Z80-A. Infelizmente, porém, no máximo 32 KBytes podem ser gerenciados pelo Interpretador BASIC, residente na ROM do MSX. Esta limitação de memória é agravada pela necessidade de se alocar partes da RAM para armazenar variáveis dos diversos sistemas, uma vez que estes sejam conectados. Desta forma, a memória livre para o usuário escrever ou depositar programas será determinada pela configuração da máquina que ele utiliza e será avisada no vídeo, na partida no computador.

Deve-se reparar que, quando se instala a interface controladora de discos, a memória livre cai ainda mais, porque outra parte da RAM ficará reservada para a área de trabalho da controladora.

Na prática, isto significa que programas em BASIC cujo tamanho em Bytes exceda a memória livre, só poderão ser parcialmente carregados. Se for forçado, o computador emite uma mensagem de erro: OUT OF MEMORY, avisando que a memória livre acabou.

Por outro lado, programas em linguagem de máquina cujo endereçamento final ultrapasse a área reservada ao funcionamento dos drives (após o último endereço disponível-HIMEM), além de não serem carregados, provocarão uma interrupção no funcionamento do computador.

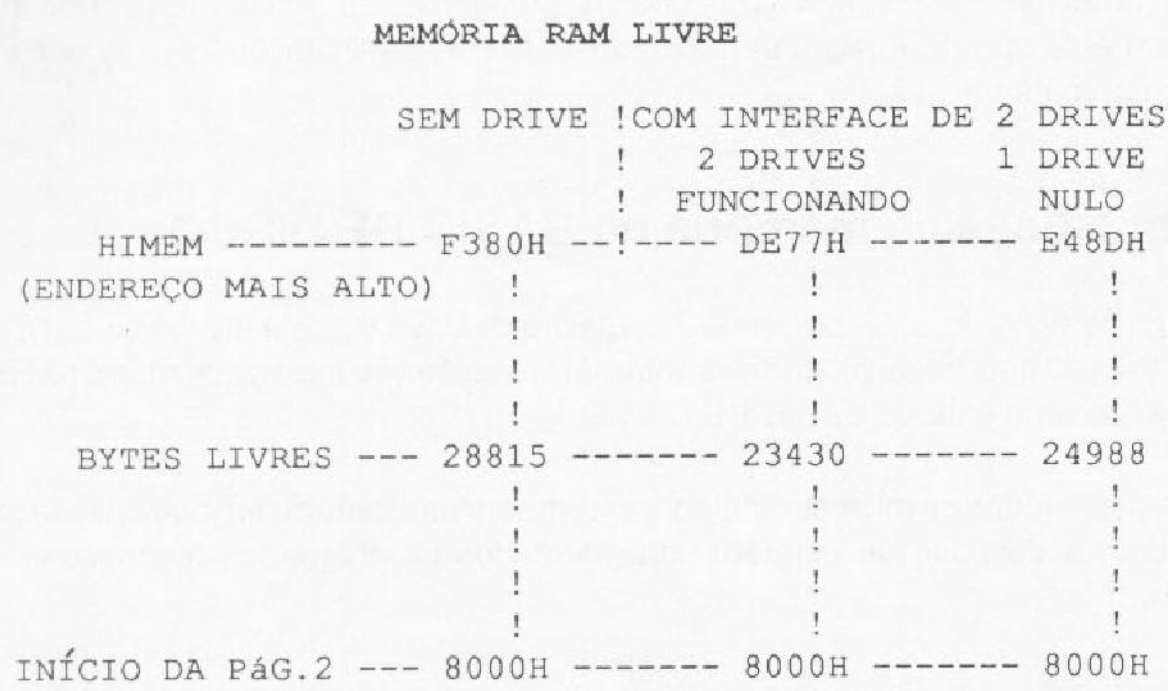


Figura 1.1 - Espaço disponível para o usuário carregar programas na RAM: programas em BASIC são carregados a partir de &H8001, enquanto que programas em linguagem de máquina podem ocupar outras posições, desde que não ultrapassem o topo da MEMÓRIA DISPONÍVEL (HI-MEM - "HIghest available MEMory").
* Interface CDX-2 MICROSOL/EXPERT 1.1

Programas gravados em fita cassete que se encaixem nas características acima, não poderão ser copiados em disco, visto que o seu carregamento e funcionamento é virtualmente impossível. Algumas providências terão que ser tomadas pelo programador, para que estes programas possam ser modificados (adaptados) para uma "versão de disco". Uma série de artifícios para conseguir tal intento pode ser implementada, como por exemplo, relocar o programa em linguagem de máquina, tirar todos os espaços desnecessários no programa em BASIC e assim por diante. Essas providências não estão, entretanto, ao alcance de todos os usuários, visto que são complicadas e trabalhosas, algumas delas nem sempre dando o resultado que se poderia esperar.

NOTA:

Na Figura 1.1, o leitor poderá ter uma idéia do espaço disponível na memória (livre) em Bytes e do último endereço disponível para carregamento dos programas em linguagem de máquina. Outras interfaces diferentes daquela mencionada na Figura poderão apresentar valores diferentes dos mencionados.

A anulação do drive B, através da tecla <control>, implica na alocação da área de trabalho reservada a este drive na RAM do computador para o carregamento de programas, como mencionado anteriormente. Por este motivo, não é possível gerenciar as operações que envolvem este drive. Por exemplo: copiar um programa do A para o B. Neste caso, o programa em questão só poderia ser copiado dentro do mesmo drive, com outro nome.

N.B.: As restrições acima mencionadas referem-se exclusivamente a programas que carregam e rodam na área reservada às operações com o BASIC DE DISCO.

As limitações de memória em ambiente DOS são bem menos críticas. Isto ocorre porque o padrão MSX prevê a reserva de 64 KBytes (!) de RAM para o Sistema de Disco operar. Como o DOS ocupa apenas cerca de 8 KBytes desta área, o restante fica livre para receber programas ou para o gerenciamento das operações realizadas pelo DOS.

Enquanto no BASIC DE DISCO o máximo conseguido (nas condições estipuladas ao rodapé da Figura 1.1) era de 24988 bytes livres, com o DOS este valor poderá atingir a cerca de 54790 Bytes, SEM ANULAÇÃO do drive B.

Assim, por exemplo, a área de "buffer" de entrada e saída, num processo de cópia de arquivos de um drive para outro, é infinitamente maior que a mesma área no BASIC DE DISCO. Praticamente todo o espaço livre poderá ser alocado. Resultado: cópia mais rápida e com maior volume de Bytes copiados de cada vez. Para quem possui apenas um drive para fazer cópias, isto significa menos trocas de disquetes como A e B.

O leitor poderá mensurar o que afirmamos acima, ao fazer cópias de vários arquivos simultaneamente. Basta somar os tamanhos em Bytes de cada arquivo, fornecidos pelo próprio DOS (veja o comando DIR) e conferir quantos Bytes são lidos pelo Sistema Operacional antes de passá-los para o outro drive.

Por outro lado, os Sistemas Operacionais de Disco permitem aos programadores a estruturação de um programa em várias partes. Apenas uma delas é carregada no micro, ficando assim RESIDENTE na máquina. As outras partes, ou "blocos", ficam no disquete: são os arquivos NÃO- RESIDENTES, que podem ser trocados posteriormente.

Em ambiente DOS, o bloco principal do programa pode ser parcialmente substituído pelos blocos NÃO-RESIDENTES. Estes últimos são mantidos no disco e por se

sobreporem ao programa RESIDENTE, são chamados arquivos de "OVERLAY".

Sendo a maioria dos programas em ambiente DOS escritos em ASSEMBLER ou COMPILADOS (traduzidos) de uma linguagem de alto nível, a quantidade de memória utilizada pelos mesmos é relativamente mais reduzida, o que por seu turno permite a elaboração de programas bem mais sofisticados num espaço maior de memória disponível. Eis aí porque o usuário dispõe de Aplicativos poderosos rodando sob o DOS. No caso específico do MSX-DOS, a sua compatibilidade com o CP/M 2.2 permite que o micro aceite programas originalmente criados para esse Sistema, com pouca ou nenhuma adaptação.

Um espaço maior na memória RAM, aliado à qualidade no desempenho das operações básicas com o gerenciamento do Sistema de Disco, tornam o MSX um micro com características mais adequadas para os trabalhos de nível profissional, sendo até bastante aceitável o seu emprego em tarefas desta natureza, caso as limitações inerentes ao computador (velocidade, memória) não sejam fatores de impedimento do seu uso.

O que é o BASIC DE DISCO ?

Um microcomputador MSX standard vem da fábrica capacitado apenas a gravar e ler em fitas cassete, ou ler um programa em cartucho.

A grande limitação operacional das fitas cassete é a de só poder ler ou gravar SEQÜENCIALMENTE.

Fora este aspecto, a confiabilidade de gravação e leitura de arquivos é muito baixa, principalmente quando os dados são enviados ao gravador em maior quantidade (por exemplo, acima de 2400 Bauds). A isto se soma o fato do processo em si ser muito lento e arriscado (o usuário pode, acidentalmente, gravar um arquivo em cima do outro) desaconselhando assim o seu uso em aplicativos profissionais.

Quando o usuário compra um MSX e acopla a ele somente um gravador cassete, todas as instruções de gravação e leitura de dados ou programas serão referidas a este periférico. Neste conjunto estão: SAVE, LOAD, RUN, BSAVE, BLOAD e OPEN.

O interpretador BASIC, entretanto possui uma lista de comandos, à qual o usuário tem acesso quando conecta uma interface controladora de discos adequada.

Estes novos comandos possibilitam o uso de disk drives e são por isso chamados de BASIC DE DISCO ou DISK BASIC.

Interação entre o Interpretador BASIC e a controladora:

Embora estejam previstos os comandos em BASIC necessários ao funcionamento dos drives, as ROTINAS referentes aos mesmos não se encontram no Interpretador.

Essas rotinas são programas em linguagem de máquina, que se utilizam do BIOS (Basic Input/Output System ou "Sistema Básico de Entrada e Saída") para gerenciar os drives.

Se o usuário comandar ou usar um programa com estas instruções, sem a interface de discos instalada, será emitida uma mensagem de erro, geralmente do tipo:

ILLEGAL FUNCTION CALL

CHAMADA ILEGAL DE FUNÇÃO

Para acionar o BASIC DE DISCO corretamente, é necessário que a interface controladora de discos contenha essas rotinas. Caso contrário, o usuário só poderá operar seu Sistema de Disco através do DOS.

Assim, podemos dizer que o BASIC DE DISCO está previsto na ROM do MSX, mas não implementado. Poderíamos dizer também que ele só se torna disponível quando a interface controladora de discos, contendo suas rotinas, é conectada a um dos slots externos do micro.

Versatilidade do BASIC DE DISCO

A utilização do BASIC DE DISCO aumenta enormemente a capacidade operacional do computador. Através dele, o programador tem acesso aos recursos do BIOS de disco.

Esses recursos são necessários para o gerenciamento de arquivos gravados no disco. Devido ao seu formato circular, os disquetes permitem a varredura ao longo do seu diâmetro, pela cabeça de leitura e gravação, ao contrário das fitas cassete, que são lidas por uma cabeça estacionária, de forma necessariamente seqüencial.

Os gravadores cassete domésticos são adaptados para o uso com o computador. Para que ele pudesse ter a mesma versatilidade do acionador de disco, seria necessário uma modificação radical no design do sistema de transporte de fita, a fim de aumentar a velocidade de gravação e leitura e permitir a edição de bytes na fita.

Com o uso dos disquetes, essa versatilidade é alcançada com facilidade. Sendo a varredura da cabeça feita longitudinalmente, em fração de segundo toda a superfície do disco pode ser examinada para a coleção ou edição de dados, em lugares definidos pelo usuário, através do Sistema Operacional.

Isto, na prática, significa que o ACESSO a informações gravadas no disco é factível, sem que haja necessidade de ler TODOS os dados que ANTECEDEM os desejados.

Assim, o usuário pode criar e manipular, usando recursos do BASIC DE DISCO, arquivos de ACESSO RANDÔMICO OU ALEATÓRIO, nos quais a flexibilidade de coleta ou armazenamento de dados é bem maior.

Não obstante os recursos citados acima, o formato de arquivamento SEQÜENCIAL, na forma convencional, poderá, a critério do programador, ser também implementado. Mesmo neste caso, a leitura de dados é significativamente mais veloz e insuperavelmente mais confiável do que nas fitas cassete.

O BASIC DE DISCO permite ainda, através de comandos próprios, a operação da maioria dos recursos importantes do MSX-DOS, tais como: gravação e leitura de arquivos, cópia, renomeação e apagamento de um ou mais arquivos, de forma selecionada, etc.

Além disso, TODOS os outros comandos constantes do BASIC convencional serão aceitos sem restrições. Apenas o usuário deverá prestar atenção ao fato de que os comandos referentes às funções de gravação e leitura de programas referir-se-ão preferencialmente ao Sistema de Disco. Havendo necessidade de se gravar ou ler programas e arquivos em fita cassete, este periférico deverá ser especificado na linha do comando:

Em DISCO:

SAVE <"[DRIVE:] programa">[,A] - grava um programa em BASIC, opcionalmente no formato ASCII, se a cláusula A for mencionada.

LOAD <"[DRIVE:] programa">[,R] - carrega um programa em BASIC, independentemente se foi gravado em ASCII ou no formato convencional em binário condensado. A cláusula R, se incluída, fará o programa rodar após ser carregado.

RUN <"[DRIVE:] programa">[,R] - carrega e roda um programa em BASIC, independentemente se foi gravado em ASCII ou não. Zera as variáveis e fecha os arquivos abertos. Quando R é usado, os arquivos não são fechados.

MERGE <"[DRIVE:] programa" ou "arquivo"> - carrega um programa em BASIC ou arquivo, gravados no formato ASCII sem destruir o conteúdo de um programa preexistente, desde que a numeração das linhas não seja coincidente.

BSAVE <"[DRIVE:] programa">,<endereço inicial>,<endereço final>,[<endereço de execução>] [,S] -grava em programa em linguagem de máquina, previamente alojado nos endereçamentos especificados. A clausula S salva uma tela gráfica direto da RAM de vídeo (VRAM).

BLOAD <"[DRIVE:] programa">[,R] [,S] - carrega um programa em linguagem de máquina. Carrega e roda se R for incluído no comando. Carrega tela gráfica se S for mencionado no lugar de R.

OPEN <"[DRIVE:] arquivo"> - abre um arquivo para gravação ou leitura de dados, em formato ASCII. Se for RANDÔMICO, inclui-se também o tamanho do arquivo.

Em fita cassete:

CSAVE <"programa"> - grava um programa em BASIC.

CLOAD [<"programa">] - carrega um programa em BASIC.

SAVE <"CAS:programa"> - grava um programa em BASIC, no formato ASCII.

LOAD <'CAS:[<programa]">[,R] - carrega um programa em BASIC previamente gravado no formato ASCII. Roda o programa se a cláusula R for incluída. Se o nome do programa não for citado, será carregado o primeiro programa encontrado na fita.

RUN [<número da linha do programa>] - roda um programa em BASIC previamente carregado na memória do micro, através dos comandos CLOAD e LOAD "CAS: ". Se o número da linha não for citado, o comando zera todas as variáveis, fecha arquivos abertos e roda o programa desde seu início.

RUN <"CAS:[<programa]">[,R] - carrega e executa um programa em BASIC gravado no formato ASCII. Se o nome do programa não for citado, será carregado e rodado o primeiro programa encontrado na fita. R mantém arquivos abertos.

BSAVE <"CAS:programa",>,<endereço inicial>,<endereço final>,<endereço de execução>[,S] - grava um programa em linguagem de máquina, nos moldes do modo correspondente em disco.

BLOAD <"CAS:[<programa>]">[,R] [,S] - carrega e roda (R) um programa em linguagem de máquina. A cláusula S permite o carregamento de tela gráfica previamente salva da VRAM. Sem o nome do programa, o comando carrega o primeiro que for encontrado na fita.

OPEN <"CAS:arquivo"> - idem ao mesmo comando em disco, só podendo ser aberto no formato SEQÜENCIAL.

Comparando-se os comandos acima, observa-se que o periférico "gravador cassete" é especificado na linha de comando pela letra "C" e pelo prefixo "CAS:".

Deve-se observar também que certos comandos, como SAVE e LOAD, têm sintaxes diferentes, quando empregados para o acionamento de um ou outro periférico.

Para facilitar a compreensão daqueles leitores que são usuários recentes do disk drive, depois de um certo período manuseando fitas cassete, tabelamos os comandos acima, mostrando a correspondência entre eles:

Use a TABELA a seguir para passar programas de fita para disco, ou para rodar e salvar programas em fita, quando a interface de disco estiver instalada e ativa.

Note que o comando RUN no BASIC DE DISCO não tem uma correspondência exata nas operações com fita: no Sistema de Disco, RUN "programa" carrega e roda automaticamente o programa em BASIC, esteja ele gravado no formato binário condensado (gravação normal) ou no formato ASCII; em fita, RUN "CAS:" (ou LOAD "CAS:",R) só é capaz de carregar e rodar automaticamente programas gravados previamente no formato ASCII, através do comando SAVE "CAS:".

Por outro lado, há uma duplicidade de comandos que executam a mesma função. Por exemplo: RUN "CAS:" e LOAD"CAS:",R carregam e rodam programas em BASIC gravados na fita em formato ASCII, sendo indiferente usar uma ou outra instrução, quando o que se deseja é pura e simplesmente carregar e rodar arquivos, sem se importar com variáveis ou arquivos abertos.

TABELA COMPARATIVA DE COMANDOS EM BASIC PARA GRAVAÇAO E LEITURA DE PROGRAMAS E ARQUIVOS

PARA USO COM DISCO:	PARA USO COM FITA CASSETE:
LOAD "[DRIVE:] programa"	CLOAD ["programa"] LOAD"CAS:[<programa>]"(ASCII)
SAVE "[DRIVE:] programa"	CSAVE "programa"
SAVE "[DRIVE:]programa",A (ASCII)	SAVE"CAS:programa" (ASCII)
RUN "[DRIVE:] programa"	RUN"CAS:[<programa>]"(ASCII) LOAD"CAS:[<programa>]",R (ASCII)
RUN [número da linha]	RUN [número da linha]
BLOAD"[DRIVE:]prog." [,R]	BLOAD "CAS:[<programa>]" [,R]
BSAVE"[DRIVE:]prog.",etc.	BSAVE "CAS:programa", etc.
MERGE "[DRIVE:]programa"	MERGE "CAS:programa"
OPEN "[DRIVE:] arquivo"	OPEN "CAS:arquivo"

Note também que, no BASIC DE DISCO, a nomeação do programa ou arquivo NÃO É OPCIONAL ! Isso ocorre porque o acesso aos dados no disco não é seqüencial e sim randômico, podendo o Sistema Operacional achar, assim, qualquer coisa gravada. Se o nome não for mencionado, o Sistema não saberá o que procurar !!!

Lembre-se sempre deste detalhe, se for programar em BASIC. Não se esqueça de prever no seu programa a possibilidade de invalidar a entrada, quando ao pedir o nome do arquivo, o usuário teclar <return>. Por exemplo:

```
10 INPUT"DIGITE O NOME DO ARQUIVO";A$
20 IF A$=" " THEN PRINT"ENTRADA INVÁLIDA":GOTO 10
```

Alguns programas aplicativos em disco permitem que o usuário leia e grave arquivos em fita com um comando único. Neste caso, o usuário deverá respeitar a sintaxe das instruções específicas dos dois periféricos, o que geralmente significa incluir o prefixo "CAS:" na nomeação do arquivo em fita.

Organização de trabalho nos disquetes

Como o leitor possivelmente já deve estar percebendo, a implantação de um Sistema Operacional de Disco, seja ele através do BASIC DE DISCO, seja através do MSX-DOS e seus compatíveis, requer um alto grau de organização e confiabilidade, o que é, em grande parte, responsabilidade do próprio Sistema Operacional.

Para que se possa entender isso claramente, sem ser obrigado a fazer um Curso de Doutorado em Informática, é necessário lançar mão de alguns dados relativos à maneira como as informações são gravadas em um disquete.

Começaremos por afirmar que, estando na EPROM da Interface Controladora de Discos todas as rotinas (ou programas) necessárias para acessar o Sistema de Entrada e Saída (BIOS) do computador, esta é, em última análise, o CORAÇÃO DO SISTEMA OPERACIONAL DE DISCO.

As informações que servem de guia ao Sistema Operacional são gravadas por ele próprio no disquete. Através delas, o Sistema poderá localizar dados gravados pelo usuário em qualquer posição do disco.

Quando os disquetes saem da fábrica, não há nada gravado sobre a sua superfície. Se eles fossem utilizados assim, seria impossível precisar em que trecho da superfície estariam sendo gravados ou lidos os programas. Para evitar que tal aconteça, são feitas marcações, de uma FORMA organizada, pelo Sistema Operacional.

A marcação corresponde a uma série de TRILHAS MAGNÉTICAS CONCÊNTRICAS, divididas cada uma em vários SETORES.

Uma parte desses SETORES é reservada ao Sistema Operacional, para gravar e ler informações necessárias ao seu gerenciamento. Sobre elas falaremos mais adiante.

Nos outros SETORES são gravados os arquivos e programas do usuário. O que isto significa em termos de memória dependerá do tipo de disquete usado.

O PROCESSO DE MARCAÇÃO DE TRILHAS E SETORES NO DISCO É DENOMINADO FORMATAÇÃO.

A FORMATAÇÃO é peculiar de cada Sistema Operacional, o que significa, na prática, que um disquete formatado por um Sistema de Disco só poderá ser lido por outro Sistema Compatível.

Por especificação da MICROSOFT, a formatação de discos no padrão MSX é totalmente compatível com o MS-DOS, do IBM-PC, com exceção da área destinada à ROTINA DE PARTIDA, cujo programa é exclusivo do MSX.

Os SETORES reservados ao Sistema Operacional, os quais contêm informações preciosas demais para serem alteradas pelo usuário, são chamados de SETORES PRIVADOS ou PRIVATIVOS DO SISTEMA OPERACIONAL.

Os outros SETORES, utilizados para o arquivamento de dados ou programas do usuário, são de acesso não-privativo, e por isso chamados de SETORES PÚBLICOS. Por força da formatação utilizada no padrão MSX, cada TRILHA é dividida em 9 SETORES, numerados de 0 a 8 (veja também o Apêndice 2, para mais detalhes).

Via de regra, os disquetes de 5 1/4" são formatados com 40 trilhas em cada lado, enquanto que os de 3 1/2" requerem 80 trilhas de cada lado.

IMPORTANTE:

NUNCA FORMATE UM DISCO DE FACE SIMPLES COMO SE FOSSE UM DE DUPLA, POIS O SEGUNDO LADO NÃO FOI TESTADO PELO FABRICANTE.

Cada SETOR é capaz de comportar 512 Bytes de informação. Assim sendo, os disquetes de 5 1/4" de face simples e dupla poderão receber:

Face Simples: 512 x 9 x 40 = 184320 Bytes = 180 KBytes*

Face Dupla: 512 x 9 x 40 x 2 = 368640 Bytes = 360 KBytes

* 1 KByte = 1024 Bytes.

Esses valores de 180 e 360 KBytes não estão totalmente disponíveis para o usuário, pois, como foi dito antes, uma parte é alocada para uso exclusivo do Sistema. Para saber a memória real disponível, devemos contabilizar o quanto é gasto para os Setores Privados:

Num disquete de face simples, a formatação reserva os setores 0 a 8 da trilha 0, perfazendo um total de nove setores para esta finalidade. O número de Bytes gasto teórico será:

512 x 9 = 4608 Bytes = 4,5 KBytes

Portanto, o total disponível aos Setores Públicos será necessariamente:

184320 - 4608 = 179712 Bytes = 175,5 KBytes

Analogamente, num disquete face dupla, os setores 0 a 8 do Lado 0 e os setores 0 a 2 do Lado 1, são alocados para esse fim, perfazendo um total de doze setores. Isto dará:

512 x 12 = 6144 Bytes = 6 Kbytes

Repetindo o mesmo cálculo, verificamos o valor real de Bytes disponíveis:

368640 - 6144 = 362496 Bytes = 354 KBytes

Esses valores de 175,5 e 354 KBytes são, portanto a MEMÓRIA AUXILIAR real que o usuário dispõe em cada disquete de face simples e dupla, a qual servirá ao armazenamento de programas e arquivos.

Ao contrário da MEMÓRIA RESIDENTE no computador, que se apaga quando o desligamos, a MEMÓRIA AUXILIAR é do tipo NÃO-VOLÁTIL, isto é, permanece gravada pelo tempo que desejarmos. Essa MEMÓRIA AUXILIAR pode ser considerada, para fins práticos, como uma EXTENSÃO da MEMÓRIA RESIDENTE do computador.

Para que são alocados os espaços de memória de 4,5 (face simples) e 6 KBytes (face dupla) dos Setores Privados?

Para armazenar três tipos de informação:

1 - A Rotina de Partida (BOOTSTRAP)

2 - A Tabela de Alocação de Arquivos (FAT)

3 - O Diretório.

Ao compreender o que são e para que servem estas três informações, iremos entender de maneira mais simples como funciona o Sistema Operacional de Disco.

A Rotina de Partida

É um programa gravado pela interface controladora no setor 0 da trilha 0, portanto, bem no início do disco, que possibilita carregar e rodar o Sistema Operacional DOS assim que o usuário dá partida no computador.

Este programa contém uma série de informações sobre a formatação (tipo) do disco (veja o Apêndice 1) e as instruções necessárias para carregar e rodar o DOS.

Quando a partida é dada, a ROM da interface de discos assume o controle e processa uma série de investigações, para verificar se existe algum cartucho conectado ao MSX. Se houver, o BASIC DE DISCO será configurado para rodar o programa nele existente. Caso contrário, a controladora providencia a leitura da ROTINA DE PARTIDA e o seu armazenamento em &HC000, até &HC0FF. Se o primeiro Byte da ROTINA não for EBH ou E9H, ou ainda, se ocorrer alguma falha de leitura por qualquer motivo, o BASIC DE DISCO será ativado. A controladora inspeciona a RAM para verificar se a memória disponível é de 64 KBytes, antes de carregar o DOS. Se isto não for verdade ou se nenhum dos programas DOS citados na ROTINA existir no disco, a preferência será dada novamente ao BASIC.

Embora este seja o padrão elaborado pela MICROSOFT, nem todas as controladoras o seguem à risca. A CD-X2 da MICROSOL, por exemplo, só ativa o BASIC DE DISCO depois que um disquete sem DOS é lido.

A Tabela de Alocação de Arquivos

É a Tabela que informa ao Sistema os locais exatos onde os arquivos constantes do Diretório estão gravados. Esta Tabela é conhecida como FAT, abreviação de FILE ALLOCATION TABLE.

As informações que estão tabeladas referem-se aos "CLUSTERS" ("aglomerados"), que são monopolizados para a alocação dos arquivos.

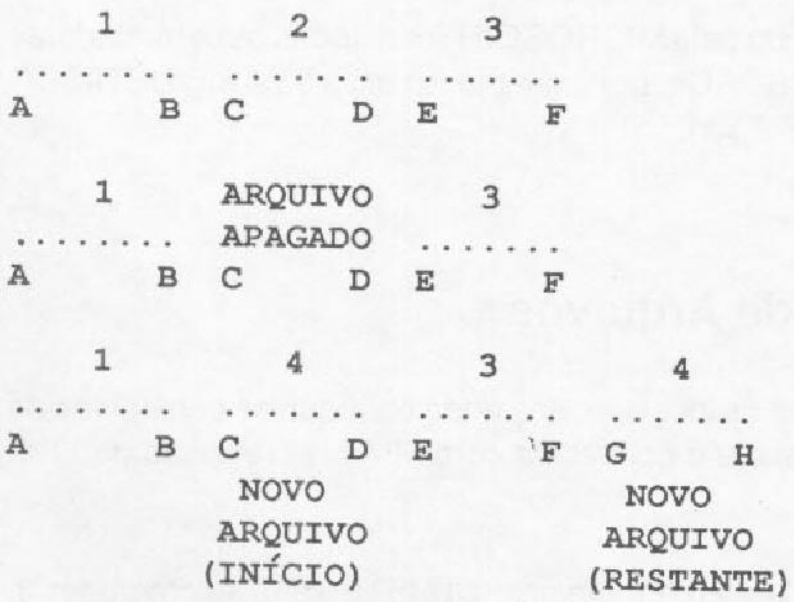
O CLUSTER é o espaço MÍNIMO em Bytes, usado para a gravação de um arquivo. No disquete de face simples, equivale a um setor (512 Bytes) e no de face dupla, a dois setores (1024 Bytes).

O Sistema Operacional aloca um determinado arquivo, de acordo com o número de CLUSTERS que ele irá ocupar, a partir de um CLUSTER inicial definido pelo Diretório. O número do setor correspondente a este aglomerado diz ao Sistema onde o arquivo começará a ser lido ou gravado.

Na FAT, estão tabelados os CLUSTERS de cada programa ou arquivo. Como a ocupação do disco é seqüencial, cada nova gravação começará sempre no primeiro CLUSTER disponível. À medida em que os CLUSTERS vão sendo alocados, sempre seqüencialmente, a FAT vai sendo construída. Quando o tamanho de um arquivo for maior do que a seqüência de CLUSTERS inicialmente alocada, o restante do arquivo será gravado a partir do primeiro CLUSTER disponível seguinte.

O exemplo mais comum desta circunstância ocorre quando o usuário grava e apaga arquivos sucessivamente. Suponhamos que três arquivos, 1, 2 e 3, estejam gravados num disco e que o arquivo 2 tenha sido apagado. A próxima gravação se dará no lugar do primeiro CLUSTER desocupado, que corresponde ao arquivo 2, que foi apagado. Se o novo arquivo (4) for maior do que o 2, somente uma parte dele ocupará os CLUSTERS do arquivo 2. O restante será alocado no primeiro CLUSTER disponível após o arquivo 3.

Esquematicamente, a representação seria a seguinte:



No exemplo acima, A,B,C,D,E,F,G e H são setores do disco. O arquivo 1 ocupa os setores A e B e os outros assim por diante. Quando o arquivo 2 foi apagado, os setores C e D ficaram desocupados. Assim, sendo o arquivo 4 maior que o 2, uma parte dele ocupará os setores anteriormente alocados ao 2 e o restante, os próximos setores vagos (G e H).

Por aí se vê a importância da FAT, pois é ela quem informa em que setores estão localizados os programas e arquivos constantes do disco. Quando a FAT é danificada, o Sistema Operacional recorre a uma cópia de segurança existente no próprio disco. Há casos onde o disquete poderá ter o seu mapeamento de tal forma alterado, que novas leituras do arquivo se tornam inviáveis.

Alguns utilitários recuperam disquetes procurando CLUSTERS perdidos e reagrupando-os, de forma a serem novamente acessados como arquivos.

Na maioria das vezes em que um disquete apresenta o chamado "erro de leitura", o leitor deve prestar atenção para saber se o erro ocorre com apenas um arquivo ou com todo o disco. Às vezes, é suficiente RECOPIAR apenas os arquivos com problemas de leitura. Se o erro persistir, é bastante possível que tenha havido problemas com a formatação (veja também a pág. 124, no Cap. 5).

CUIDADO:

INTERFACES DEFEITUOSAS FORMATAM OS DISQUETES COM IRREGULARIDADES, CAUSANDO PROBLEMAS NAS OPERAÇÕES DE GRAVAÇÃO E CóPIA DE ARQUIVOS.

Uma interface e/ou cabos de drive intermitentes dificultam o gerenciamento das operações de entrada e saída para o disco. Outros pontos a considerar são: a alimentação elétrica do equipamento, a existência de desalinhamento das cabeças dos drives, a integridade física dos disquetes utilizados, etc. O usuário deve se prevenir, instalando o equipamento em local arejado, mas sem poeira ou poluição ambiental. Se for necessário, deve ligar o sistema a um estabilizador de rede, dotado de filtro de linha, de boa qualidade.

O Diretório:

Dos Setores Privativos do Sistema, o Diretório é o mais conhecido, porque dele partem informações que o usuário pode acessar através do comando DIR, dos programas DOS e CP/M.

O DIRETÓRIO, como o nome diz, é uma LISTA ou CATÁLOGO de informações, a respeito dos arquivos e programas constantes no disquete, as quais veremos adiante.

O Diretório é preenchido cada vez que um arquivo é gravado ou copiado no disco. São ao todo 32 Bytes de dados para cada arquivo, os quais, quando visualizados através de um programa que possibilite ler o conteúdo do disco, se apresentam assim distribuídos:

```
NO NO NO NO NO NO NO NO
EX EX EX AT 00 00 00 00
00 00 00 00 00 00 HO HO
DA DA CI CI TA TA TA TA
```

Dissecando estas linhas, verificamos os seguintes dados:

NO NO NO NO NO NO NO NO -- 8 Bytes para o NOme do arquivo.

EX EX EX AT 00 00 00 00 -- 3 Bytes para a EXtensão do NOme do arquivo e, ao lado, 1 Byte para o ATributo; os Bytes restantes são zero.

00 00 00 00 00 00 HO HO -- 6 Bytes zero e 2 Bytes para a anotação da HOra em que o arquivo foi gravado.

DA DA CI CI TA TA TA TA -- 2 Bytes para a marcação da DAta em que o arquivo foi gravado, 2 Bytes para a indicação do Cluster inicial ao Sistema Operacional e 4 Bytes para a gravação do TAmanho do arquivo.

Observações:

Quando o NOme do arquivo for menor do que oito caracteres (1 Byte = 1 caractere), os Bytes restantes serão considerados espaço em branco (20H ou 32 na Tabela ASCII). O mesmo se aplica à EXtensão.

O Byte relativo ao ATributo se apresentará necessariamente zerado, devido ao fato de que até a presente versão do MSX-DOS não existe a previsão, através de comando próprio, de sua utilização.

O estabelecimento de um ATributo tem duas finalidades básicas:

1ª - ATributo de "ler-somente": impede que uma gravação possa ser feita num arquivo (proteção contra escrita), mas permite que este arquivo seja lido.

2ª - ATributo de "arquivamento": seleciona os arquivos de um disquete que serão ou não copiados, dependendo da maneira como o ATributo é estabelecido.

O ATributo pode ser, ainda, usado para outras finalidades, como por exemplo, para colocar um rótulo ("label") ou "volume" no disco. Neste caso, o NOme do volume constará do Diretório e ocupará 11 caracteres (8 do NOme e mais 3 da EXtensão), mas não será listado como arquivo, devido à mudança no Byte do ATributo. Quando o Sistema prevê a leitura do Volume do Disco (caso do padrão Microsoft para os micros IBM-PC compatíveis), este será anunciado antes do conteúdo do Diretório e pode ser acessado tanto no comando DIR, quanto no comando VOL (do MS-DOS).

Os 2 Bytes referentes à HOra em que o programa é gravado, ao contrário do Byte do ATributo, podem ser enunciados quando o usuário solicita a listagem do Diretório. A gravação da HOra, entretanto, só poderá ser conseguida através de um programa adicionado ao Sistema Operacional ou de um relógio interno no micro. No MSX, a primeira opção é impraticável, e a segunda não é fornecida ao usuário. Comandando DIR para listar arquivos gravados em um IBM-PC, a hora das gravações poderá ser visualizada.

Em resumo: ATributo e HOra não são aproveitados até a presente versão do MSX-DOS, mas quanto ao segundo, o sistema poderá lê-lo, caso conste do Diretório. Os Bytes do ATributo deverão permanecer zerados, se o usuário quiser ver o arquivo listado com o comando DIR.

Já os Bytes pertencentes à informação do Cluster Inicial fornecem ao Sistema a indicação do primeiro setor na FAT alocado ao arquivo em questão. De posse deste dado, o Sistema Operacional pesquisará na Tabela todos os outros setores alocados ao arquivo.

E, finalmente, quanto ao TAmanho do arquivo, apenas 2 Bytes são usados quando o arquivo é um programa, mas os arquivos de dados podem alcançar mais de uma centena de KBytes, quando então os Bytes restantes serão também usados.

Limitações físicas do Diretório:

Como vimos, 32 Bytes são gastos pelo Sistema Operacional para anotar informações no Diretório a respeito de um determinado arquivo. Como são reservados 4 e 7 setores do disco (face simples e dupla, respectivamente) para o Diretório, poderemos estimar o máximo de arquivos que podem ser gravados num disquete de 5 1/4":

Face Simples: 512 x 4 / 32 = 64 arquivos

Face Dupla: 512 x 7 / 32 = 112 arquivos

Aparentemente, estes valores são um pouco exagerados, pois, na maioria das vezes, os programas que gravamos ou copiamos têm um tamanho em Bytes superior a 10000. É bastante provável que os Setores Públicos sejam ocupados ANTES que o Diretório fique cheio.

Por outro lado, quando um disquete é utilizado para arquivar blocos de um número baixo em Bytes, é possível que a situação se reverta. Supondo, por exemplo, que resolvemos arquivar cartas digitadas em um Processador de Textos, com cerca de 2500 caracteres, ocupando idêntico espaço no disco, o MÁXIMO de que as cartas arquivadas poderiam dispor seria de:

2500 x 64 = 160000 Bytes (Face Simples)

2500 x 112 = 280000 Bytes (Face Dupla)

O restante (20224 Bytes, Face Simples e 82496 Bytes, Face Dupla), não poderia ser ocupado, pois não há mais espaço vago no Diretório. Quando o usuário tentar gravar

mais um arquivo, será alertado pelo Sistema com a mensagem: DIRETÓRIO CHEIO.

Arquivos deletados no Diretório:

Todos os arquivos que aparecem na listagem impressa pelo computador, quando o usuário digita DIR na linha de comando do DOS, são considerados ARQUIVOS PRESENTES. Quando um arquivo é apagado (deletado) E NENHUM OUTRO É COLOCADO EM SEU LUGAR, este arquivo permanecerá FISICAMENTE no disco e o seu nome mantido no Diretório.

Entretanto, o comando DIR é incapaz de acessar os nomes dos ARQUIVOS DELETADOS e estes são, por isso, considerados ARQUIVOS AUSENTES.

O que impede o acesso aos nomes dos arquivos deletados é a substituição, pelo Sistema Operacional, do primeiro caractere do NOME do arquivo, pelo caractere 229 da Tabela ASCII, representado pelo seu valor em hexadecimal &HE5 (ou E5H). Se outro caractere, dentro daqueles permitidos na nomeação de arquivos, for novamente colocado em seu lugar, ele será novamente acessado pelo comando DIR do DOS e pelo FILES do BASIC DE DISCO, mas isto pouco significará, pois o Sistema Operacional se baseia pela FAT, que, neste caso, estará zerada.

Para recuperar um arquivo deletado, os utilitários com este fim reagrupam, manual ou automaticamente, os setores anteriormente alocados para o arquivo e regravam estes setores na forma de um novo arquivo, em outra parte do disco. A parte do DIRETÓRIO contendo a marca de deleção (E5H), será utilizada para dados de outra operação de gravação (leia a pág. 146 do Cap. 5).

ARQUIVOS DELETADOS PODEM SER RECUPERADOS, DESDE QUE OS SETORES DO DISCO POR ELE OCUPADOS NÃO TENHAM SIDO REAPROVEITADOS PARA GRAVAR OUTRO ARQUIVO

A recuperação de um arquivo deletado é uma tarefa complicada e, às vezes, quase impossível. É recomendável evitar ao máximo a deleção desnecessária de arquivos, pois arrependimentos tardios podem acabar custando muito caro.

Regras para a nomeação de arquivos:

Como acabamos de ver no tópico sobre Diretório, um total de 11 Bytes são reservados para o nome dos arquivos e suas respectivas extensões.

Existem regras bem definidas, que discorreremos a seguir, para a nomeação dos arquivos e das extensões, as quais, se não forem seguidas à risca, poderão causar sérios embaraços aos usuários.

O leitor deve se recordar, caso já tenha usado anteriormente o gravador cassete como meio de armazenamento de seus programas, que a nomeação dos arquivos segue duas regras básicas:

1 - até seis caracteres podem ser usados no nome do programa.
2 - o usuário deve optar se usará letras maiúsculas ou minúsculas.

A nomeação de programas ou arquivos em disco é bem mais flexível e versátil, porém, deve ser feita com mais critérios:

Em primeiro lugar, é INDIFERENTE digitar letras maiúsculas ou minúsculas para nomear um arquivo, pois elas serão convertidas em maiúsculas de qualquer maneira, quando forem gravadas no Diretório do disco !

Em segundo lugar, o NOME do arquivo (oito caracteres) deve ser separado da extensão (três caracteres) por um PONTO:

NOME DO ARQUIVO.EXTENSÃO DO NOME

Em terceiro lugar (e não menos importante), nenhum caractere passível de ser interpretado pelo Sistema Operacional poderá ser usado, seja no nome, seja na extensão.

OS CARACTERES PROIBIDOS SÃO:

\ , . / ; : + = ? * " [ ]

É muito fácil entender o porquê disso: basta ver, como exemplo, que os nomes dos arquivos são separados de suas extensões por pontos. Portanto, não é possível usar o ponto como nome do arquivo, pois este é o sinal que o DOS usa para saber que terminou no caractere precedente o nome em questão. O mesmo ocorre com todos os outros caracteres. Os seus significados ficarão claros à medida em que os co-

mandos do DOS forem descritos.

Nomes que se refiram a dispositivos externos ao DOS, tais como PRN , NUL, AUX, CON, etc., também não podem ser usados. Estes se referem, em geral, ao destino ou alvo de um comando em particular. Por exemplo, PRN leva para a impressora conectada à saída paralela, o resultado da operação solicitada na linha de comando; NUL se refere dispositivo externo NULO, etc.

Significado das EXTENSÕES:

Outro aspecto importante a ser considerado nos nomes dos arquivos são as suas extensões.

EXTENSÃO, como o nome já diz, é um prolongamento do nome de um arquivo, cujo objetivo é o de informar ao Sistema Operacional ou ao usuário, do que se trata o arquivo em questão.

Como apenas três letras são permitidas, elas têm por obrigação representar mnemônicamente o seu significado. Isto é válido para ambos os casos (Sistema e usuário) citados acima.

Para o MSX-DOS, a extensão indica a natureza dos arquivos. Quando se dá, como comando, o nome de um arquivo, o DOS analisa a extensão, para saber o que fazer com ele.

Algo parecido ocorre quando se dá partida no micro e o Sistema de Disco procura arquivos com extensões .SYS ou .SIS, de acordo com a Rotina de Partida do disco.

Exemplos de extensões interpretadas pelo DOS:

COM - abreviação de COMmand (COMando), indica que o arquivo é um programa objeto COMpilado de uma linguagem de programação (por exemplo: Assembly, linguagem C, Pascal, etc.), para ser carregado e rodado pelo DOS. na forma de um comando.

EXE - abreviação de EXEcutable (EXEcutável), indica que o arquivo é um programa em linguagem de máquina e está pronto para rodar (programa EXEcutável).

BAT - abreviação de BATch ("LOTE"), indica que o arquivo é uma lista de comandos do DOS, ou seja, um "lote" de comandos, que, ao rodar, serão colocados automaticamente na linha de comando e executados.

SYS(SIS) - abreviação de SYStem (SIStema), indica que o arquivo é o bloco principal do Sistema Operacional, sendo este o tipo de arquivo procurado quando o usuário dá partida no computador.

Estas extensões são de uso obrigatório para caracterizar a natureza do arquivo. Desta forma, não poderão ser usadas indiscriminadamente.

Se arquivos que necessitam estas extensões não forem nomeados corretamente, o DOS poderá perder o controle ao tentar executá-los, podendo até obrigar o usuário a reinicializar o computador.

O uso de extensões fora daquelas citadas acima, privativas do DOS, fornece uma idéia clara do tipo de arquivo com o qual o usuário está lidando, desde que empregadas de forma padronizada e correta.

As extensões não interpretadas pelos Sistemas Operacionais ou por programas específicos, não provocam o impedimento do carregamento ou a interrupção do funcionamento do micro. Estas extensões poderão ser criadas e aquelas de uso já consagrado deverão ser conhecidas, para um imediato reconhecimento do programa examinado, facilitando assim o seu carregamento e/ou a sua execução.

Exemplos de extensões de uso não restritivo:

BIN - abreviação de BINary (BINário), indicando que o arquivo é um programa em linguagem de máquina, necessitando do comando em BASIC BLOAD"arquivo",R para ser carregado e rodado.

BAS - abreviação de BASic, indicando que o arquivo é um programa em BASIC, necessitando do comando RUN ou LOADarquivo,R para ser carregado e rodado.

OVR - abreviação de OVeRlay ("COBERTURA"), indicando que o arquivo é parte de um programa que fica residente no micro e que poderá ler partes dele para cobrir ou substituir partes residentes.

TXT - abreviação de TeXT (TeXTo), indicando que o arquivo está gravado no formato de TEXTO (ASCII), sendo possivelmente resultante de um Processador de Textos ou arquivo de dados de outro aplicativo, convertido para este formato de gravação.

ASC - abreviação de ASCII, indicando que o arquivo é um programa em BASIC gravado no formato ASCII.

BAK - abreviação de BAcKup, indicando que o arquivo é uma CÓPIA DE SEGURANÇA de outro arquivo, possivelmente com o MESMO NOME, mas sem esta extensão.

USANDO AS EXTENSÕES COM INTELIGÊNCIA, O USUÁRIO MANTERÁ SEMPRE UM PERFEITO CONTROLE SOBRE TODO O MATERIAL GRAVADO EM SEUS DISQUETES.

As extensões podem também ser usadas pelos programas aplicativos, para identificar quais os arquivos por ele criados.

O dBASE, por exemplo, usa a extensão .DBF, para saber se um arquivo no disco é uma Banco de Dados, enquanto que o SuperCalc usa uma extensão .CAL, para identificar as suas planilhas.

Quando extensões distintas são usadas por um programa aplicativo, o usuário pode usar o mesmo nome em arquivos diferentes.

Diferença entre PROGRAMA e ARQUIVO:

Até aqui, usou-se indistintamente os termos PROGRAMA e ARQUIVO para referenciar uma gravação em disquete.

Na realidade, o ARQUIVO é uma coletânea de dados diversos, CRIADOS pelos PROGRAMAS APLICATIVOS e gravados no disquete.

O termo ARQUIVO se refere ao fato de que os dados coletados são ARMAZENADOS desta forma no disco. Se estes dados forem um conjunto de INSTRUÇÕES, então o arquivo é um PROGRAMA !

Não obstante, o termo ARQUIVO substitui, na prática, ARQUIVOS VERDADEIROS e PROGRAMAS. O Sistema Operacional, para SIMPLIFICAR todas as operações, trata o que está gravado no disquete como arquivo.

O número de arquivos é contabilizado pelo DOS nas operações de cópia e é mostrado ao usuário quando este solicita o Diretório.

Outras informações gravadas nos disquetes:

Além de arquivos e programas, é possível gravar informações ou DADOS, na forma de Bytes, em Setores preestabelecidos do disco. Da mesma forma, as informações ou dados gravados podem ser lidos, desde que se saiba previamente quais os setores alocados à gravação.

Tanto a gravação quanto a leitura destes DADOS, só é factível através de programas especificamente construídos para esta finalidade.

No Diretório, como foi visto, não há previsão sobre a localização de dados isolados, de maneira que, através dele, não é possível saber se existem informações que não sejam arquivos.

Isto poderá acarretar problemas, pois quando é feita uma gravação de DADOS em setores onde previamente existia um arquivo, este será danificado pela presença dos novos Bytes.

Portanto, quando se gravam DADOS em um disquete desta forma, é conveniente saber "a priori" se os setores a serem utilizados pelo programa estão desocupados.

Cópia de disquetes:

Depois da formatação de disquetes, imediatamente à sua compra, o que o usuário deve aprender a fazer é uma cópia de reserva dos mesmos, depois de gravados.

Isto, antes de mais nada, é uma atitude de bom senso, pois ninguém é imune à perda de arquivos ou de disquetes inteiros. Fazendo assim, os programas perdidos podem ser recuperados.

A CÓPIA DE UM ARQUIVO OU DISQUETE É DENOMINADA DE "BACKUP" OU "CÓPIA DE SEGURANÇA".

Existem basicamente duas formas de se proceder a cópia de disquetes:

1 - CÓPIA SOMENTE DOS ARQUIVOS;

2 - CÓPIA INTEGRAL OU PARCIAL DE SETORES DO DISQUETE.

Na primeira hipótese, o Sistema Operacional se encarregará de ler nos Setores Privados as informações necessárias e copiará todos os arquivos exatamente na ordem em que eles aparecem no diretório. Não importará se os arquivos estiverem gravados no disco fonte em setores separados, pois, no disco destino, os setores serão alocados seqüencialmente, como se aquela fosse uma gravação "NOVA" (esta seria uma maneira elegante de "arrumar" um disquete muito mexido).

Na segunda hipótese, a cópia integral será feita a partir do primeiro setor da trilha 0, até o último setor do disco fonte, que serão LIDOS (processo de cópia ou leitura) e, depois, gravados (processo de escrita) no disco destino. Neste caso, pouco importa se os Bytes contidos nos setores são arquivos, programas ou dados: todos serão tratados como se fossem Bytes gravados em setores. Informações como aquelas constantes nos Setores Privados (Rotina de Partida, FAT e Diretório), passarão para o disco destino, que será um "espelho" do disco fonte.

Por este motivo, este processo é denominado apropriadamente de CÓPIA FÍSICA DO DISQUETE.

A cópia física poderá ser PARCIAL, nos casos onde, através de um programa utilitário copiador, são selecionados no disco fonte apenas os setores efetivamente alocados para os arquivos nele residentes. Dependendo do copiador, pode-se escolher quais arquivos serão repassados ao disco destino. Caso todos os arquivos sejam copiados, o disco destino conterá o equivalente a uma cópia física integral, menos os setores relativos à Rotina de Partida e os outros setores vagos.

Em muitos momentos da vida do usuário, a Cópia Física Integral será de grande utilidade. Por exemplo:

Caso nº 1:

O usuário dispõe de um computador IBM-PC compatível e deseja copiar disquetes do MSX, mantendo intacta a formatação original (que possibilita, por ex.,dar Partida no MSX-DOS).

Como as formatações, para fins de leitura de Dados,são compatíveis, pode-se usar este recurso, que também faz parte do MS-DOS, e o disco obtido rodará no MSX sem nenhum problema.

Caso nº 2:

O usuário precisa de um disquete rigorosamente igual ao disco fonte, porque o mesmo contém DADOS imprescindíveis ao funcionamento de algum programa.

Como o Sistema normalmente não vê no Diretório DADOS gravados em setores isolados, a única forma de copiá-los é pela cópia física.

Observação importante:

Normalmente, os comandos de cópia física operam com as formatações normais do Sistema. Em alguns casos, são implementadas "trancas" na gravação, através de um processo especial, com a finalidade de "proteger" os programas ou o disquete contra cópias, impedindo assim a ação desses comandos.

Na maioria das vezes, um ou mais "erros de leitura" são mencionados durante a operação, quando um disco "protegido" está sendo copiado fisicamente.

A única forma de contornar problemas desta natureza é usar um utilitário capaz de "destrancar" o disquete protegido. Se esta operação for bem sucedida, é bem possível que a cópia contenha exatamente a mesma "tranca" do disco original, o que quer dizer que o disco fonte foi apenas fisicamente copiado e não destrancado.

O leitor poderá ver mais detalhes sobre como fazer um ou outro tipo de cópia nas páginas seguintes deste livro, quando tratarmos sobre os comandos do DOS relativos a ambos os processos.

A estrutura física dos disquetes flexíveis de 5 1/4":

Ao examinar pela primeira vez um disquete que se acabou de comprar, o usuário sem experiência é induzido a prestar atenção em alguns detalhes da sua estrutura, possivelmente aqueles que o fabricante menciona, como por exemplo, o entalhe lateral, que permite proteger um disco contra escrita (gravação).

Mas existem outros pontos que deveriam merecer a atenção do leitor, quando o que se deseja é aprender mais conscientemente sobre o que se está manipulando e, assim, utilizar mais racionalmente a coleção de disquetes que se vai fazendo.

O disco flexível é assim chamado devido à natureza do material com que os mesmos são fabricados, o que não quer dizer que eles podem ser dobrados sem que ocorra algum tipo de dano à sua integridade física.

Olhando-se o disquete de perto, observam-se dois rasgos importantes sobre a superfície do invólucro:

1 - a janela de leitura e gravação: facilmente reconhecível por se postar longitudinalmente e por ser o rasgo de maior tamanho. Sobre ele se posicionarão as cabeças de leitura e gravação do drive, para ter acesso ao disquete.

2 - o furo correspondente ao furo de índice: próximo ao centro do disco e, coincidente, com um furo menor no próprio disquete, que é o furo de índice propriamente dito.

A seguir, são dadas explicações sobre estes 2 itens:

A abertura de leitura e gravação no invólucro torna o disquete um objeto vulnerável, altamente danificável, por permitir acesso à superfície magnética do disco. Nenhum tipo de detrito, seja sólido ou líquido, deverá passar por ali, sob pena de não só estragar o disco, como até mesmo as cabeças leitoras do drive. Por este motivo, é recomendável guardar os disquetes imediatamente após o uso.

Tudo aquilo que aprendemos no passado sobre fitas magnéticas, é válido para os disquetes: quanto maior for a quantidade de material químico magnetizável e quanto maior for o polimento da superfície, melhor será a qualidade do disquete. O primeiro desses itens refere-se à capacidade de retenção da magnetização imposta pela cabeça de gravação e à qualidade em resposta de freqüência do sinal gravado. Assim, um disquete de DUPLA DENSIDADE será necessariamente melhor do que um de DENSIDADE SIMPLES, podendo armazenar maior quantidade de Bytes com mais segurança. Quanto ao segundo, o polimento, ele é necessário para diminuir o atrito entre a cabeça e a superfície do disco e, assim, aumentar a longevidade (duração) da mesma.

Quando um disquete é dito como de FACE SIMPLES, isto significa que o fabricante não testou a qualidade do material magnético pulverizado sobre a segunda face, não havendo, portanto, garantias de que as gravações feitas ali sairão perfeitas. É geralmente dito ao usuário que a utilização da segunda face poderá resultar em dano à

cabeça de gravação. Pode-se desconfiar um pouco da severidade deste aviso, porque num drive de dupla face, as 2 cabeças (que tocam ambas as faces) movimentam-se simultâneamente, o que tornaria o uso de um disquete face simples proibitivo.

Os cuidados que normalmente são dispensados à fita magnética, devem ser exercitados com os disquetes. A poluição ambiental, como por exemplo, fumaça de cigarros e congêneres, solventes dos mais diversos tipos, desde a água até os ditos solventes orgânicos, como os álcoois, etc., são extremamente danosos ao disco, pela possibilidade de retirada do material magnético de sua superfície. Nisto inclui-se a umidade (vapor d'água), sendo aconselhável estocar os discos em ambiente seco.

Da mesma forma, não se pode tocar na superfície magnética do disco, sob pena de deixar lá a gordura e outras substâncias que estão nos dedos.

Quanto ao furo do índice, a sua utilidade é enorme para o Sistema Operacional: quando o disquete é inserido dentro do drive, o local do furo coincide com um sistema de leitura ótico simples, constando de uma célula foto-elétrica e um sensor de luz posicionados em baixo e em cima do disquete, respectivamente. Ao ser rodado, o disco exporá o furo de índice a cada volta, momento no qual o sensor "lê" a passagem da luz pelo mesmo.

Isto permite que a passagem de luz seja interpretada como um "aviso" para o Sistema. No caso, isto significará que estará passando pela cabeça de leitura e gravação a parte do disquete onde estão gravados todos os inícios de trilha, o que corresponde, como visto, ao Setor "Zero" de cada trilha.

Este processo de identificação de início de trilha é chamado de setorização por software (no caso, o Sistema Operacional) e os disquetes que contêm os furos de índice de "soft-sectored".

Um outro detalhe físico importante dos disquetes, está num pequeno anel, geralmente de cor branca ou preta, colocado na borda interna do disco, chamado de ANEL DE REFORÇO ou ANEL CENTRAL ("hub ring"). Este anel, quando a trava externa do drive é abaixada, não só impede que a borda interna do disco se deforme com o tempo (o que inutilizaria o disquete), como também facilita o arraste do disco dentro do invólucro, ao permitir um contato mais firme da tampa giratória superior, que pressiona o disco contra a base do motor. Quando você adquire disquetes, é interessante certificar-se da presença deste anel, pois é ele quem prolonga sobremaneira a vida útil dos mesmos.

Cuidados na manipulação e armazenamento de disquetes:

Devido à fragilidade dos invólucros que envolvem os disquetes, além dos outros fatores anteriormente mencionados, uma série de atenções e cuidados devem ser exercidos na manipulação e, principalmente, no armazenamento deles.

As condições ideais de armazenamento são:

*** Ambiente Seco:**
A variação de umidade tolerável por um disquete está em torno de 20 a 80%.

*** Temperatura Amena:**
NUNCA exponha o disquete a fontes de calor, pois, pela sua própria natureza, ele poderá ficar irremediavelmente deformado. A faixa aceitável de temperatura pode oscilar entre 10 a 52 C.

*** Armazenamento em Pé:**
Nenhum tipo de peso pode ser colocado em cima dos disquetes e nem se deve "grampeá-los" com qualquer tipo de "clips", sob pena de deformar o invólucro.

*** Afastamento de Campos Magnéticos:**
Uma série de artefatos elétricos e eletrônicos desenvolvem um campo magnético suficientemente forte para apagar uma gravação, seja em disco, seja em fita: motores, transformadores, alto-falantes, telas de TV ou de Monitores, etc.

Além disso, o manuseio de disquetes deve ser cuidadoso, principalmente ao alojá-lo no drive. Para escrever sobre alguma etiqueta no invólucro, só é permitido o uso de canetas de ponta porosa, já que com elas se pode escrever sem pressionar o disco.

PARA MANTER SUA COLEÇÃO DE DISQUETES EM ORDEM

1 - CUIDE SEMPRE DE SEUS DISCOS, SEGUINDO AS RECOMENDAÇÕES DESCRITAS ANTERIORMENTE.

2 - FAÇA BACKUPS DE TODOS OS SEUS DISCOS.

3 - EVITE DEIXAR DISQUETES DENTRO DOS DRIVES.

4 - MANTENHA UMA LISTAGEM DOS ARQUIVOS DE CADA DISCO E OS MESMOS SEMPRE ETIQUETADOS E/OU NUMERADOS.

Obtenção da listagem dos arquivos na impressora:

Para listar seus arquivos na impressora, rode seu Sistema Operacional e, na linha do prompt, tecle ^P (mantendo<control> pressionada e teclando uma única vez a letra "P"). Ligue a impressora, coloque os disquetes no(s) drive(s) e comande:

A>DIR/W <return>

ou:

A>DIR/W B: <return>

Terminada a listagem, tecle ^ N (<control > + N). Fazendo isto, você estará desativando a saída do DOS para a impressora.

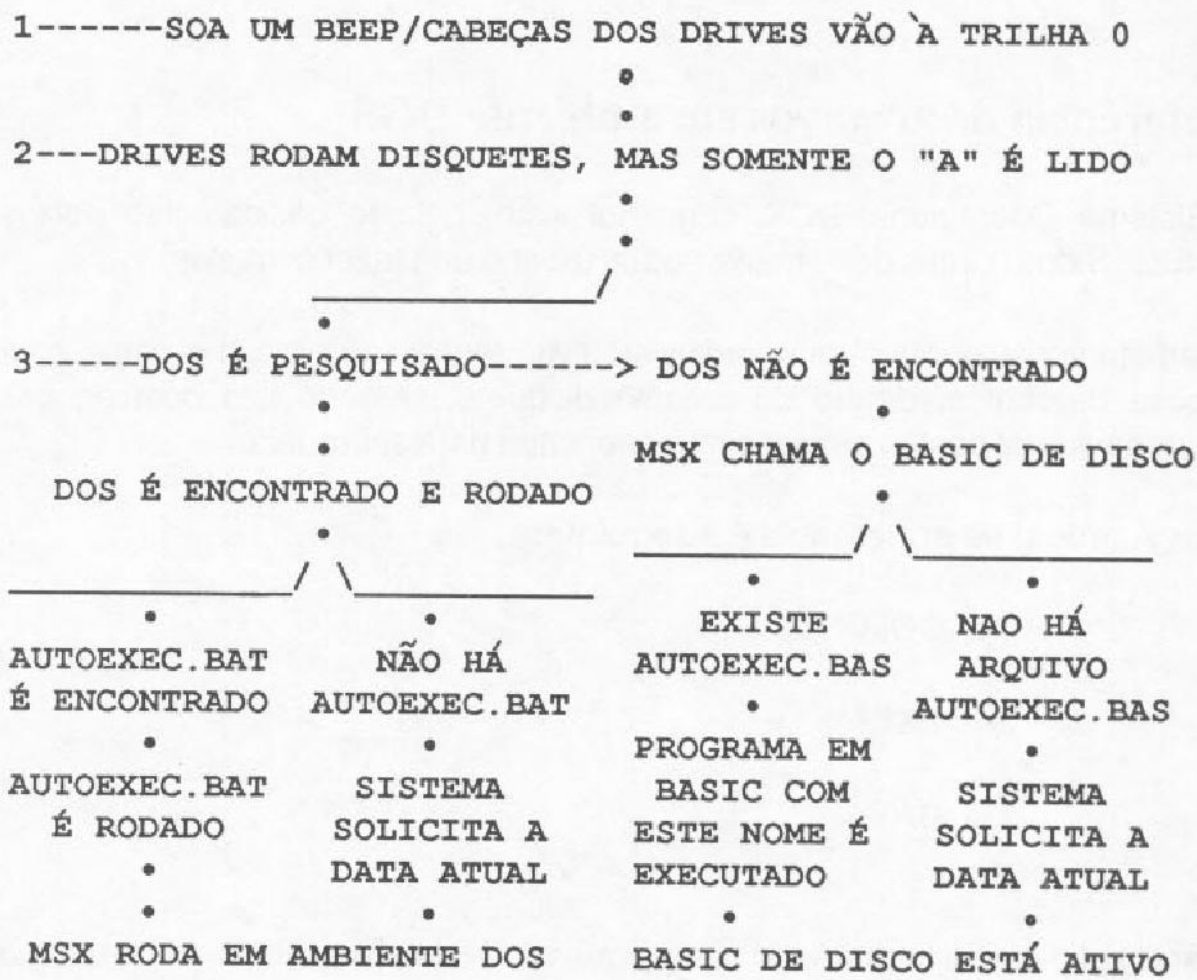
Estude o significado dos comandos do DOS (incluindo o acima citado) no próximo Capítulo.

Inicialização dos Sistemas Operacionais de Disco:

Antes de passarmos aos Capítulos seguintes, onde veremos os comandos e recursos dos Sistemas Operacionais de Disco, vamos observar como o MSX inicializa suas operações com o disk drive.

Quando o usuário dá partida no computador, a sua rotina habitual de inicialização (já previamente comentada) verifica a RAM, para conferir a memória livre e realizar outras operações, entre elas a vistoria dos slots primários externos, com o objetivo de verificar se há, em algum deles, um ou mais cartuchos conectados. Em caso positivo, o programa existente neste(s) cartucho(s) é rodado, a começar pelo slot A.

Desde o início da partida, até a fase em que a controladora de discos inicia o carregamento do DOS, a seguinte seqüência deverá acontecer:



Os arquivos AUTOEXEC.BAT e AUTOEXEC.BAS são arquivos que, uma vez identificados, são carregados e rodados automaticamente na partida do computador MSX. O primeiro deles se refere a um arquivo de comandos do DOS, pois sua extensão é .BAT, enquanto que o segundo nada mais é do que um programa em BASIC qualquer, que foi nomeado assim para que seu carregamento ocorra na partida do micro.

Note que o carregamento do DOS prevalece sob o carregamento de qualquer outro programa. Assim, num disco contendo este Sistema Operacional, só será possível a

auto-execução de um arquivo AUTOEXEC.BAT, mesmo que um outro chamado AUTOEXEC.BAS esteja presente.

Entretanto, se o DOS não estiver gravado no disquete, o BASIC DE DISCO prevalecerá, provocando deste modo a execução do AUTOEXEC.BAS, mesmo que um arquivo AUTOEXEC.BAT esteja presente.

Se nenhum dos dois tipos de arquivo de auto-execução estiver presente no disco, então ambos os Sistemas poderão ser ativados da forma habitual.

Preferência de arquivos em ambiente DOS:

O Sistema Operacional DOS, como foi visto, guia-se basicamente pelas EXTENSÕES dos nomes dos arquivos para saber o que fazer com eles.

É perfeitamente possível que vários arquivos tenham um mesmo nome com extensões diferentes, dentro do mesmo disquete. Quando isto ocorrer, o DOS selecionará qual destes arquivos terá prioridade para ser rodado.

A ordem de preferência é a seguinte:

1º - .COM

2º - .EXE

3º - .BAT

A explicação para isto reside no fato de que o DOS interpreta o nome de um arquivo como se fosse um comando. No caso deste nome não existir como um comando embutido no programa COMMAND.COM, ele o pesquisará no diretório do disco, para saber se existe um programa .COM com o nome do comando, o qual é tratado, neste caso, como um comando EXTERNO.

A existência de comandos EXTERNOS, em número estabelecido pelo usuário, aumenta a capacidade operacional do DOS, mas impõe a sua presença no disquete com o DOS, para que os mesmos sejam acionados corretamente.

Como, normalmente, ao digitar um nome de um arquivo na linha do prompt do DOS, não se digita a extensão, não é aconselhável manter no mesmo disco arquivos em

lote com o mesmo nome de um programa, porque o DOS sempre acionará o programa primeiro, seja ele .COM ou .EXE.

Por exemplo: num disquete, existe um programa dBASE, com o nome de DBASE.COM. O usuário não deve criar um arquivo de comandos com o nome DBASE.BAT, pois este, ao se digitar A>DBASE, jamais será rodado. No caso disto ser estritamente necessário, é preferível nomear o arquivo de comandos como AUTOEXEC.BAT e, assim, obrigar o DOS a rodar primeiro o arquivo em lote e depois o dBASE.

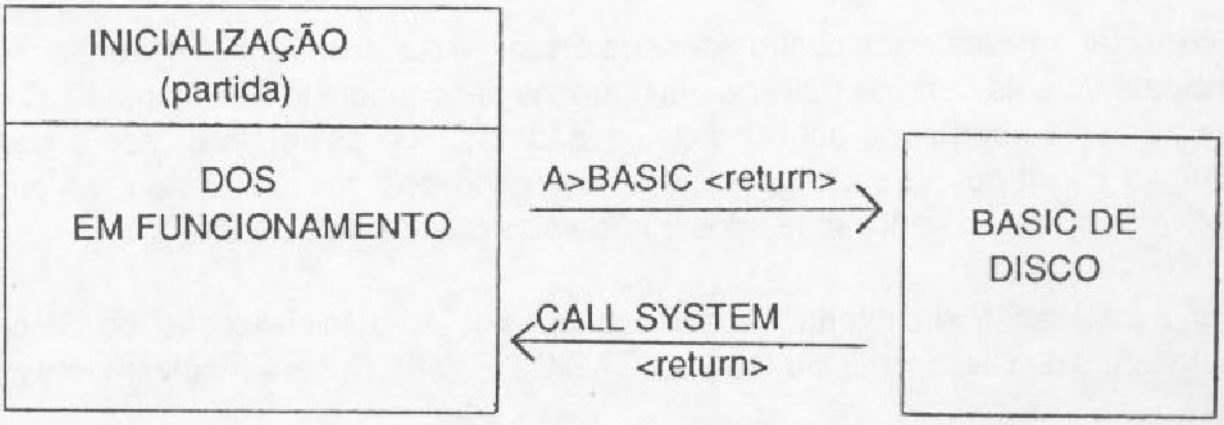
Nota:
A criação dos arquivos BATCH, de grande importância prática, será tema de uma discussão especial mais adiante, neste livro.

Interação entre o DOS e o BASIC DE DISCO:

Como foi visto no esquema anterior, onde mostramos a rota de inicialização dos Sistemas Operacionais de Disco, não há necessidade de se dar partida no computador com o disquete contendo o Sistema Operacional, já que o BASIC DE DISCO se torna ativo quando a interface que contém as suas rotinas é instalada.

Entretanto, o Padrão MSX prevê no MSX-DOS um comando capaz de ativar o BASIC DE DISCO, que será estudado no próximo Capítulo: o comando BASIC. Tal fato confere ao MSX uma grande flexibilidade de operação, facultando ao usuário trabalhar com ambos os sistemas, seja isoladamente, seja, em alguns casos, até em conjunto, sem que haja necessidade de dar nova partida no micro.

Isto pode ser conseguido, se a partida no micro for dada com o disquete contendo o DOS, provocando o seu carregamento para a memória do micro:



Deve-se notar que não é possível fazer a operação oposta: inicializar o BASIC DE DISCO e entrar no DOS. Em condições normais de utilização, o micro responde a esta solicitação com uma mensagem de erro.

A passagem de uma para outra configuração de memória poderá ser efetuada sem nenhum prejuízo ao funcionamento do computador. Apenas deve-se providenciar a colocação do disquete contendo o mesmo DOS utilizado para a partida no drive A, pois, ao retornar para ele, o programa propriamente dito terá que ser recarregado.

Versões diferentes de DOS e DOS de outros fabricantes terão grande chance de não serem aceitos pelo MSX na recarga. Neste caso, o computador emitirá mensagem de erro, que variará de acordo com o fabricante do DOS.

A mudança de status no funcionamento do MSX só é possível entre o DOS e o BASIC DE DISCO. Embora seja factível carregar e rodar um Sistema Operacional CP/M (com a devida interface), uma vez que este assuma o controle do micro, não será permitido passar para o BASIC.

Como o MSX-DOS é compatível com o CP/M 2.2, permitindo rodar os programas deste último em ambiente DOS, adquire-se óbvia vantagem em se operar com o DOS.

Pelo exposto acima e ainda pelo fato de que o CP/M não faz parte da máquina MSX Padrão, deixamos de apresentar neste livro os comandos relativos a este Sistema Operacional.

Nota: CP/M significa Control Program for Microprocessors e é Marca Registrada da Digital Research.

Observações finais sobre a inicialização:

Ao conectar uma interface controladora de discos, o slot preferencial deverá ser o A, principalmente no caso de mais de um cartucho ser conectado ao computador. Embora não seja obrigatório este tipo de ligação, fazendo assim, forçamos o gerenciamento inicial do Sistema Operacional de Disco pelo micro, ou seja, os outros periféricos só serão verificados após e não antes da interface de disco.

Com a colocação da interface de disco no slot A, a inicialização do Sistema Operacional de Disco, seja ele DOS ou BASIC DE DISCO, será relativamente mais

rápida, pois o computador dará vez à interface de discos para atuar em primeiro lugar e assumir o controle do computador. Tal fato não impedirá a varredura de outros slots (primário, como o B, ou secundários, como nos expansores de slot), permitindo que o computador reconheça, por exemplo, se o usuário conectou algum programa em cartucho ou um periférico adicional.

Se o cartucho colocado no outro slot for um programa, este será rodado, inibindo a inicialização dos Sistemas Operacionais de Disco (os drives não serão rodados). Se for um periférico (por exemplo, 80 colunas), este será igualmente ativado, mas o controle voltará à interface de discos.

O usuário pode desativar (ou inibir) o controle exercido pela controladora, mantendo pressionada a tecla shift até que a partida do computador seja efetivada. Ao fazer isso, o BASIC normal da máquina padrão MSX é ativado, impedindo assim qualquer operação de E/S com os drives. Para retornar ao status de memória com o sistema de discos, deve-se "resetar" o computador. Note que não há necessidade de se desativar a interface de discos para realizar as operações de E/S com fitas cassete, pois os comandos habituais do BASIC continuam operantes.

CAPÍTULO 2

COMANDOS DO MSX-DOS E SEUS COMPATÍVEIS

Introdução

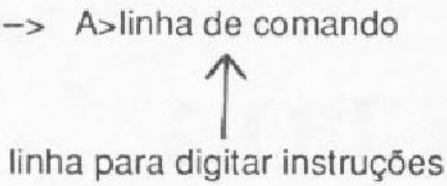
Se você chegou até aqui lendo o Capítulo 1, não terá provavelmente diticuldades em entender a nomenclatura das palavras usadas daqui para a frente. Independentemente disso, procuraremos repetir, sempre que necessário, alguns conceitos, com o objetivo de tentar facilitar ao máximo a compreensão do texto.

O MSX-DOS será discutido e apresentado em primeiro plano, servindo de referência para os demais Sistemas que mantêm com ele equivalência ou compatibilidade. Sempre que for cabível, faremos analogia com o SOLX-DOS, da MICROSOL TECNOLOGIA, e, em alguns casos, citaremos com exclusividade os comandos deste último.

Chamamos a atenção do leitor para o fato de que os comandos serão descritos em ordem alfabética. Os comandos e argumentos devem ser separados por delimitadores: espaço em branco (usado neste livro), vírgula, ponto-e-vírgula, sinal de igual e tecla tab. Exemplo:
DEL,B:CAP1-1.TXT ou DEL B:CAP1-1.TXT.

Edição da linha de comando

Como foi visto no Capítulo anterior, quando o DOS assume o controle operacional do computador, ele exibe um ***"prompt"*** indicativo do drive corrente e da linha de comando, na qual o usuário terá direito a digitar suas instruções:



É bem possível que os comandos digitados envolvam instruções longas, sendo que, no caso de algum erro cometido, toda a linha será invalidada. Não há necessidade de digitar tudo outra vez, uma vez que se saiba como EDITAR a linha recém-introduzida.

Antes de apresentarmos os comandos de edição, teremos que deixar bem claro que uma instrução dada pelo usuário ao DOS só é interpretada quando se tecla

<RETURN>

Depois que o <return> for teclado, duas coisas acontecerão:

1 - o comando será interpretado e, se estiver errado, será emitida uma mensagem de erro.

2 - a linha digitada será armazenada na memória do computador (buffer do teclado), até que a tecla <return> seja acionada com outro comando ou até que um comando de edição apropriado seja digitado, o que forçará a retirada da instrução da memória.

Como se vê, por este segundo item, cada vez que uma instrução é fornecida para o DOS, é possível reproduzi-la, chamando-a da memória do micro. Se a instrução em questão estiver incorreta, teremos meios, através das chaves de edição do DOS, de alterá-la de modo a corrigir apenas as partes erradas e, assim, poupar um novo trabalho de digitação.

Para tornar mais fácil a compreensão dos comandos de edição, convencionaremos o seguinte:

LINHA NOVA - será a linha digitada A SEGUIR uma outra linha (que estará na memória).

Obs.: poderá ser também a primeira linha digitada no início das operações com o DOS.

LINHA ANTERIOR - será a linha armazenada na memória, como a ÚLTIMA instrução fornecida pelo usuário ao DOS.

COMANDOS OU CHAVES DE EDIÇÃO

Tecla HOME

Atua de 2 modos:

* Com o CURSOR posicionado no INÍCIO da linha de comando, ela APAGARÁ da memória a LINHA ANTERIOR, imprimirá o sinal de "@" (arroba) e levará o CURSOR para a linha seguinte, onde se poderá digitar nova instrução. Caso <return> seja teclado, o CURSOR voltará a um novo prompt.

* Com o CURSOR posicionado no FIM da linha digitada, ela possibilitará EDITAR esta linha, levando-a antes para a memória do micro, para o caso da mesma ser parcialmente copiada por outros comandos. Também neste caso, o caractere "@" será impresso ao final da linha invalidada.

Observação:

Teclando-se <control> + K (^ K), obtém-se o mesmo efeito, já que esta teclagem duplica a ação da tecla < home>. Outras teclas de controle serão descritas nas chaves de edição seguintes.

Tecla ESC (ESCAPE) ou SETA PARA CIMA (SPC)

Invalidam a NOVA LINHA, mantendo a LINHA ANTERIOR e fazendo o CURSOR retornar ao INÍCIO da linha digitada.

Observação: teclando-se ^U (<control> +U) obtém-se o mesmo efeito.

SETA PARA A DIREITA (SPD)

Cada vez que é teclada, COPIA um caractere da LINHA ANTERIOR (guardada na memória) para a NOVA LINHA em digitação. Se a memória estiver "limpa", a tecla não terá ação e o CURSOR ficará parado.

SETA PARA A ESQUERDA (SPE), BACK SPACE (BS) ou ^H

Cada vez que são tecladas apagam um caractere da LINHA NOVA , à medida em que o CURSOR retrocede uma posição.

SETA PARA BAIXO (SPB)

COPIA integralmente a LINHA ANTERIOR (guardada na memória) para a LINHA NOVA a ser digitada. É útil quando se deseja REPETIR um comando anterior mais de uma vez.

Tecla SELECT + uma tecla de um caractere da LINHA ANTERIOR

COPIA todos os caracteres da LINHA ANTERIOR para a LINHA NOVA, até a primeira incidência do caractere teclado após SELECT, posicionando o CURSOR em cima deste. É útil quando se deseja trocar apenas um caractere de cada vez, podendo ser usada mais de uma vez na mesma linha a ser editada.

Tecla CLS (CLEAR SCREEN) + um caractere da LINHA ANTERIOR

APAGA todos os caracteres entre a posição do CURSOR e a primeira incidência do caractere teclado após CLS, exclusive este, e mantendo o CURSOR posicionado sobre ele. Pode ser usada mais de uma vez na mesma linha a ser editada.

Tecla INSERT

LIGA/DESLIGA o modo de INSERÇÃO: quando acionada a primeira vez, INSERE os caracteres digitados a seguir, empurrando os caracteres da LINHA ANTERIOR para a esquerda. Ao ser teclada novamente, desliga o modo de inserção de novos caracteres.

N.B.: ao contrário do que acontece nas operações com o BASIC, o cursor NÃO ficará reduzido à metade para avisar que o modo de inserção está ligado.

Tecla DELETE

APAGA o caractere da LINHA ANTERIOR que estiver posicionado SOBRE o CURSOR.

^C (<control> +C) ou <control>+<stop>

ANULAM o comando atual e fazem o CURSOR retornar à linha do prompt.

Como EDITAR uma linha de comando

Uma vez apresentados os Comandos ou Chaves de EDIÇÃO, vamos exemplificar como os mesmos podem ser usados para modificar uma linha recém-digitada, a qual foi introduzida com erro por um usuário.

O leitor poderá simular estes exemplos no seu computador, seguindo as seqüências listadas a seguir e acompanhando o efeito das teclas.

Exemplo 1:

A>DOR B:AUTOEXEC.BAS

O DOS responderá:
Bad command or file name.
Comando ou arquivo inexistente.

Explicação: o comando é DIR e não DOR.

Edição (teclar na seqüência abaixo):
- <SELECT> O ou SETA PARA A DIREITA (teclar uma vez)
- I
- SETA PARA BAIXO

Exemplo 2:

A>TYPESTI DBASEMSG.TXT
Ao terminar de digitar, tecle <HOME> ao invés de <RETURN> para anular a digitação, pois o comando era TYPE e não TYPESTI.

Edição 1:
- SETA PARA A DIREITA (teclar sucessivamente até o cursor se posicionar na letra E)
- <CLS> I
- <DELETE>
- SETA PARA BAIXO

Edição 2:

-<SELECT> S
-<DELETE> (teclar três vezes)
- SETA PARA BAIXO

Exemplo 3:

A>DIR B:CAP35-20.XT
TECLE <HOME>: a extensão está errada.

Edição:

- <SELECT> X
- <INSERT> T e <INSERT>
- SETA PARA BAIXO

Exemplo 4:

A>COPY B:AUTOECEMATI.BAT A:
TECLE <HOME>: o nome do arquivo está errado.

Edição:

-<SELECT> C (teclar duas vezes, para que o cursor pare no segundo C)
- <DELETE>
- <INSERT > X e <INSERT>
- SETA PARA A DIREITA
- <CLS> . (teclar o ponto)
- <INSERT> C e <INSERT>
- SETA PARA BAIXO

Os exemplos acima referem-se a situações relativamente banais, mas o leitor poderá se defrontar com digitações extensas e trabalhosas, como é o caso, por exemplo, das cópias com concatenações de arquivos.

O conhecimento e principalmente a prática do uso dos comandos de edição em muito facilitará a tarefa de corrigir digitações erradas. O que no início poderá parecer penoso (conhecer o funcionamento das chaves de edição e depois memorizá-las), é posteriormente compensado nas situações em que o usuário precisa realizar operações com o DOS, nas quais a mudança entre uma instrução e outra é de apenas um ou poucos caracteres.

Observações importantes:

O leitor deve ter notado que algumas teclas de caracteres, associados à tecla <CONTROL> , duplicam as funções de algumas chaves de edição. Tal fato decorre da compatibilidade do MSX com computadores de arquitetura interna mais antiga, os quais não possuíam teclas especiais, como por exemplo, as SETAS para a movimentação do CURSOR.

Algumas dessas teclas são de grande importância no funcionamento do MSX-DOS e seus compatíveis, sendo, portanto, imprescindível o seu conhecimento:

<CONTROL> + C ou <CONTROL> + <STOP>
Além de anular comandos, como mencionado acima, interrompem a execução dos arquivos de comandos do DOS em lote (arquivos "batch").

<CONTROL > + S
Interrompe as listagens decorrentes de operações do DOS, como por exemplo, de Diretórios longos ou de leituras de arquivos gravados em formato de texto (ASCII).

NOTA: após <CONTROL> +S interromper a listagem, acionando-se qualquer tecla a listagem recomeçará.

< CONTROL> + P

Habilita a SAÍDA para a impressora, se a mesma estiver ligada e em linha na saída paralela do micro. Caso contrário, <CONTROL> +P não surtirá efeito.

<CONTROL> + N
Desliga a SAÍDA para a impressora, ativada por <CONTROL> + P.

NOTA: ao acionar <CONTROL> +P, TODOS os caracteres que o MSX-DOS imprime na tela do monitor aparecerão na impressora. Este recurso, combinado com alguns comandos do DOS, que serão citados posteriormente, traz para a impressora listagens de grande interesse para o usuário.

N.B.: os códigos que o MSX-DOS envia à impressora para a impressão dos caracteres podem não ser análogos aos códigos constantes, para estes mesmos caracteres, na Tabela ASCII da impressora, o que pode resultar em ausência de impressão ou impressão de um caractere diferente do original.

<CONTROL> + M
Teclado após o comando digitado, equivale a teclar <RETURN>, para a entrada da linha. É perfeitamente dispensável o seu emprego, visto que o uso de <RETURN> é mais óbvio e direto para o usuário.

Limitações da linha de comando

O usuário poderá digitar até 128 caracteres na linha de comando. Quando este limite for alcançado, soará um BEEP para avisar o fim da linha e o CURSOR ficará imóvel.

Raras vezes este limite constituirá um problema, mas existem casos, como por exemplo nas operações de concatenação de arquivos, onde a linha digitada ultrapassará os 128 caracteres.

Mesmo nestes casos, existem maneiras de contornar este limite, como será visto mais adiante.

Descrição dos comandos do MSX-DOS

Nas páginas seguintes estarão listados todos os comandos do MSX-DOS e alguns comandos exclusivos do SOLX-DOS.

A listagem será feita com os comandos seqüenciados em ordem alfabética, para facilitar futuras consultas.

Cada comando será classificado, como é habitual nestes casos, de acordo com o TIPO, que poderá ser INTERNO ou EXTERNO.

COMANDO INTERNO - é um comando embutido no programa MSX-DOS, no bloco COMMAND.COM.

COMANDO EXTERNO - é um PROGRAMA que acompanha o MSX-DOS com uma extensão .COM e que, ao terminar de rodar, retorna ao DOS.

Os comandos serão descritos com a seguinte sistemática:

SINTAXE: enuncia a maneira como o comando é escrito.
[e] - indicam argumentos opcionais
< e > - indicam dados obrigatórios.

TIPO: indica comando interno ou externo.

SINONÍMIA: quando cabível, lista comando equivalente.

FUNÇÃO: explica o objetivo do comando.

DESCRIÇÃO: discorre detalhadamente sobre a maneira como o comando funciona e como ele poderá ser aplicado.

EXEMPLOS: quando cabível, cita uma ou mais maneiras de digitar o comando, estando incluídos no item DESCRIÇÃO.

Comando MSX-DOS: BASIC.

SINTAXE: BASIC [<drive>:<nome de um programa em BASIC>].

TIPO: INTERNO.

SINONÍMIA: não há.

FUNÇÃO: permite reconfigurar a memória do computador de modo que o mesmo

possa ser operado sob o controle do BASIC DE DISCO.

DESCRIÇÃO:

O comando BASIC ativa o BASIC DE DISCO e permite assim que as operações de entrada e saída possam ser acessadas através do mesmo.

Como as rotinas do BASIC DE DISCO, necessárias ao Interpretador BASIC, estão na Interface Controladora de Disco, quando a mesma não contiver estas rotinas, o BASIC DE DISCO não será ativado.

Se o nome de um programa em BASIC for mencionado na linha de comando, este será carregado do disco e rodado. Caso o programa citado não esteja no drive especificado (ou no drive corrente, quando não há especificação de drive), será emitida mensagem de erro de "arquivo não achado" ("file not found").

Algumas vezes, na passagem para o BASIC com o nome de um programa, este tem a sua execução abortada por algum motivo qualquer, o que pode ser visto pela impressão na tela da mensagem "Break". Quando isto acontece, é suficiente comandar RUN + <return> para reiniciar o programa.

Ao passar para o BASIC DE DISCO, o DOS deixa na memória do micro uma rotina capaz de chamar o Sistema Operacional de volta, como se fosse uma nova partida.

Esta rotina pode ser ativada através do comando CALL do BASIC, seguido do nome da rotina:

CALL SYSTEM ou _SYSTEM

Comando SOLX-DOS: COPIED.

SINTAXE: COPIED <drive de origem > <drive destino>.

TIPO: INTERNO.

SINONÍMIA: não há.

FUNÇÃO: permite copiar um disco integralmente, inclusive dados relativos à formatação.

DESCRIÇÃO:

COPIED significa COPIE Disco, nome derivado do termo COPIE, uma adaptação de COPY para o Português. Este comando realiza uma cópia física, setor a setor, do disco fonte para um disco destino, o qual não precisa estar previamente formatado.

É preciso uma certa atenção na sintaxe deste comando, que difere radicalmente do comando COPY (ou COPIE), descrito a seguir. O COPIED requer a especificação do DRIVE ou dos DRIVES onde a cópia será feita, enquanto que o COPY referir-se-á a DISQUETES como A: e B:. Sendo assim, é possível nomear apenas um drive na linha deste comando para realizar a cópia física.

Se o usuário possui apenas um drive no sistema, o comando será necessariamente redigido da seguinte forma:

A>COPIED A: A:

Com apenas um drive, a cópia física é bastante trabalhosa, devido ao grande número de vezes que o usuário é solicitado a trocar os disquetes como FONTE e DESTINO, sucessivamente. Isto se deve ao fato de que o número de setores lido por vez é relativamente reduzido.

No processo de cópia física, a leitura é um pouco mais rápida que a gravação (escrita), mas globalmente isto faz pouca diferença, sendo tudo muito lento. Assim sendo, leva-se grande vantagem na utilização de dois drives para este tipo de cópia.

Comando MSX-DOS: COPY.

SINTAXE: COPY [<origem>:] [<arquivo>] [<destino>:] [<arquivo>] [/V] [/A] [/B].

TIPO: INTERNO.

SINONÍMIA: COPIE ... [/L] (SOLX-DOS).

FUNÇÃO: permite copiar dados de um dispositivo para outro, com a opção de nomeação e renomeação de arquivos e verificação da cópia efetuada.

DESCRIÇÃO:

Na maioria das vezes em que este comando é empregado, ele serve para copiar arquivos de um drive para outro. Entretanto, o COPY é capaz de transferir dados de outros dispositivos diferentes dos disk drives, bastando para isso saber de antemão se isso é possível. Por exemplo, se for necessário enviar um texto à impressora, especifica-se como origem o drive no qual o arquivo-texto está gravado e o seu nome, e como destino a impressora (PRN: ou LST:):

A>COPY B:CARTA.TXT PRN

A>COPY MANUAL.DOC LST

Quando se deseja criar um arquivo "batch", cita-se como dispositivo de origem o console (CON:), isto é, o conjunto teclado + vídeo, e como destino o nome do arquivo que se deseja criar:

A>COPY CON B:AUTOXEC.BAT

Desta forma, todos os caracteres digitados aparecerão no vídeo, indo automaticamente para um buffer de memória. Assim que o usuário avisar ao sistema que o arquivo terminou, o que é conseguido teclando-se ^ Z (<control> + Z) e <return>, o arquivo será imediatamente gravado no drive destino.

Pode-se escrever uma carta ou qualquer outro documento, usando-se este mesmo artifício, bastando que se preste atenção à extensão empregada na nomeação do arquivo. No caso de um arquivo em lote, a extensão será necessariamente .BAT, enquanto que os arquivos de texto facultativamente terão extensão .TXT ou .DOC. Por exemplo:

A>COPY CON CARTA.DOC

A grande desvantagem no uso do comando COPY para processar textos é a limitação de recursos de edição, motivo pelo qual é preferível empregar um aplicativo deste tipo, inclusive para criar arquivos "batch". As chaves de edição descritas no início deste Capítulo não podem ser utilizadas. Além disso, a extensão das linhas digitadas está restrita aos mesmos 128 caracteres, que são também o limite da linha de comando do DOS. Para escrever um documento desta forma, seria necessário teclar <return> a cada limite de linha. Por outro lado, a grande vantagem deste recurso do COPY está no fato de não haver necessidade de se abandonar o DOS para criar tais arquivos.

É possível também conectar o teclado com a impressora (PRN: ou LST:), porém, o texto deve ser redigido antes do envio, e este último só ocorrerá após a sinalização de fim arquivo dada pelo usuário: <control> + Z + <return> . Neste caso, as sintaxes do comando poderiam ser:

A>COPY CON PRN
A>COPY CON LST

Observações:

O dispositivo LST: refere-se ao "LIST" adotado pelo CP/M. A Versão 1.1 inicial do SOLX-DOS não possuía este recurso, tendo o mesmo sido implementado na Versão 1.2, para permitir que aqueles programas originários do CP/M que adotam esta saída para a impressora pudessem rodar no MSX em ambiente SOLX-DOS sem problemas.
Caso não haja correspondência de caracteres entre micro e impressora, a listagem nesta última sairá com ausência ou troca de caracteres impressos. Em ambiente DOS, prevalece a Tabela MSX, em detrimento de qualquer filtro (ABNT ou ABICOMP) instalado no computador.

Pode-se ainda usar o COPY para listar um arquivo de texto no vídeo, tendo como destino o console. Neste caso, não há problemas de compatibilidade de caracteres entre o micro e o terminal de vídeo. Por ex.:

A>COPY B:VIDEO.TXT CON

Note que, na nomeação dos dispositivos, é dispensável a digitação de dois pontos junto com o seu nome. Note ainda que qualquer arquivo poderá ser copiado por este sistema, mas no caso específico de listagens para a impressora ou para o vídeo, só terá sentido copiar arquivos de texto ou todos aqueles gravados neste formato (ASCII), já que os outros arquivos, uma vez "traduzidos" para caracteres, exibirão notações desconexas e ininteligíveis.

Em seu uso mais abrangente, o comando COPY facilita a confecção de "backups" de arquivos e programas mais importantes do usuário. Para tornar mais flexível e versátil esta tarefa, são previstos dois recursos de grande importância, os quais podem ser citados como cláusulas na linha de comando:

1 - Caracteres-chaves ou "coringas":
São aqueles que possibilitam a substituição de caracteres dos nomes dos arquivos,

de maneira a facilitar o seu reconhecimento pelo sistema operacional, no caso de indicações de vários arquivos simultaneamente para a cópia. Os caracteres-chaves são dois:

* - substitui todos os caracteres indistintamente;

? - substitui um caractere de cada vez, podendo ser usado tantas vezes quantas forem necessárias.

Exemplos:

A>COPY *.* B:
copia todos os arquivos do disco.

A>COPY WS*.* B:
copia todos os arquivos cujos nomes se iniciem por WS e possuam qualquer extensão.

A>COPY WS.* B:
copia todos os arquivos de nome WS e com qualquer extensão.

A>COPY B:*.TXT
copia todos os arquivos com extensão .TXT existentes no drive B, para o drive corrente (A).

A>COPY CAP?-?.TXT B:
copia todos os arquivos nomeados por CAP - e com extensão .TXT (CAP1 -1.TXT, CAP1-2.TXT, etc.).

2 - Comando de verificação da cópia: /V (/L - SOLXDOS)

É o comando que permite que os arquivos a serem copiados do disco fonte sejam comparados com os mesmos arquivos gravados no disco destino, com o objetivo de verificar se a cópia foi feita corretamente. Neste caso, a cópia será mais lenta.

Exemplo:

A>COPY A:COMUNICA.DBF B:/V

Importante:

Se for detectado erro de gravação no drive destino, solicite, antes de recomeçar, a listagem do diretório do disco (comando DIR), para observar se o disco destino já está cheio. Caso contrário, verifique a integridade da formatação deste disco através de um utilitário apropriado ou, se for o caso, reformate o disco. Antes de reformatar, tente a gravação novamente, pois são muitas as variantes pelas quais uma gravação pode não ocorrer como devia, sendo possível que, numa segunda tentativa, o Sistema Operacional tenha sucesso.

É fundamental especificar o drive destino, quando o mesmo não for aquele mencionado na linha do "prompt", ou quando não constar a renomeação do arquivo.
Se o drive destino não for mencionado, o MSX-DOS emitirá uma mensagem de erro:

FILE CANNOT BE COPIED ONTO ITSELF.

Esta mensagem indica que o arquivo não pode ser gravado nele mesmo.

CUIDADO:

O SOLX-DOS PERMITE QUE A GRAVAÇÃO OCORRA SEM A MENÇÃO DO DRIVE OU DISPOSITIVO DESTINO, PERMITINDO ASSIM QUE A CÓPIA OCORRA EM CIMA DO ARQUIVO ANTERIOR. SE OCORRER ERRO DE GRAVAÇÃO, O ARQUIVO ORIGINAL ESTARÁ PERDIDO. SE O NÚMERO DE BYTES DOS ARQUIVOS A SEREM COPIADOS ULTRAPASSAR O ESPAÇO DE MEMÓRIA ALOCADO AO BUFFER DE GRAVAÇÃO, TODOS OS ARQUIVOS INDICADOS SERÃO DANIFICADOS.

Uma cópia de arquivos pode ser realizada no mesmo disco, desde que se renomeie o arquivo na linha de comando. A renomeação pode ser opcionalmente efetuada no disco destino, caso haja algum interesse do usuário nisso:

A>COPY WSMSX.BAT B:AUTOEXEC.BAT

Havendo interesse em copiar apenas os arquivos do disco fonte, o comando COPY deverá ser utilizado. Caso contrário, é obrigatório empregar a cópia física do disco fonte (ver comando COPIED do SOLX-DOS).

Quando um determinado disquete é muito utilizado para regravações sucessivas de arquivos, ele deve ser copiado com o comando "COPY *.*" para outro disco e reformatado para novas utilizações. Fazendo isso, se rearruma a disposição de arquivos

dispersos ao longo do disco, como foi explicado no Capítulo 1.

CÓPIA POR CONCATENAÇÃO*

(* não funciona no SOLX-DOS ! Veja utilitário COPIARQ)

Da mesma forma que, como vimos, é possível passar vários arquivos de um disquete para outro de maneira simultânea, é também permitido copiar vários arquivos para um só arquivo, o que se denomina de CÓPIA POR CONCATENAÇÃO, em virtude do fato de que os arquivos citados na linha de comando são SOMADOS ou ANEXADOS um ao outro. O termo CONCATENAÇÃO refere-se à forma organizada como a cópia ocorre.

A concatenação de arquivos pode ser feita de duas formas, a primeira delas sendo a mais singela, e funciona da seguinte forma: Suponhamos que se tenham num disquete vários arquivos com extensão .TXT (arquivos de texto). Quando o usuário indica ao Sistema Operacional que o resultado da cópia é apenas um nome de arquivo, o Sistema entende que a cópia ocorrerá por concatenação.

Por exemplo:

A>COPY *.TXT SOMA.TXT

No caso acima, todos os arquivos com extensão .TXT serão somados e aparecerão no arquivo SOMA.TXT, o qual poderá ou não ser gravado no mesmo disco, desde que se indique o local da gravação (por ex.: B:SOMA.TXT).

É preciso tomar um certo cuidado com este tipo de concatenação. Repare que, na linha de comando, foi feita uma referência ambígua aos arquivos .TXT pela colocação do asterisco, o qual, como sabemos, indica ao Sistema Operacional TODOS os nomes de arquivo com aquela extensão. É bem possível que haja dentro desta lista algum arquivo com a denominação dada ao arquivo destino. Se for este o caso, duas coisas podem acontecer:

1 - Quando o arquivo de origem, cujo nome seja coincidente com o do arquivo destino, for o primeiro da lista, a ele será somado o conteúdo dos arquivos seguintes. Isto significa que ele deixará de existir na forma antiga, pois o Sistema Operacional gravará o arquivo destino em cima dele.

2 - Quando o arquivo coincidente estiver posicionado do segundo lugar em diante, o Sistema Operacional, ao tentar copiá-lo, verificará que o conteúdo original não mais existe, visto que a gravação do primeiro arquivo já foi feita em cima deste. Sendo assim, é emitida uma mensagem de erro, avisando ao usuário que o arquivo anteriormente citado na linha de comando não mais existe: CONTENT OF DESTINATION LOST BEFORE COPY (Conteúdo do arquivo destino perdido ANTES da cópia). Isto significa que, ao tentar copiar um arquivo preexistente, o Sistema verificou que o seu conteúdo não era mais o mesmo.

Note-se que, em ambos os casos, o arquivo de origem cujo nome coincidiu com o destino foi ou alterado, como no primeiro caso, ou totalmente destruído, como no segundo exemplo.

A única forma de evitar que tal fato ocorra é verificar inicialmente o diretório, para evitar que referências ambíguas possam acarretar este tipo de erro. Em caso de coincidência; deve-se trocar a extensão ou o nome do arquivo destino. A pura e simples troca da extensão já evitará este problema. Por exemplo:

A>COPY *.TXT SOMA.DOC

A segunda forma, mais ortodoxa e mais segura de se concatenarem arquivos se faz pela menção do nome de cada arquivo que se deseja anexar, seguido do sinal de "+", indicativo da concatenação. Por exemplo:

A>COPY CAP1-1.TXT+CAP1-2.TXT+CAP1-3.TXT CAP1.TXT

Note que: não há necessidade de digitar um espaço em branco entre os nomes dos arquivos de origem, o que é bom para economizar espaço na linha de comando; qualquer arquivo nesta linha pode opcionalmente ser mencionado JUNTO com a indicação do DRIVE que o contém. Por exemplo:

A>COPY COPY1-1.TXT+B:CAP1-2.TXT+CAP1-3.TXT B:CAP1-TXT

Vê-se por aí que esta forma de concatenar arquivos é amplamente vantajosa, embora mais trabalhosa que a primeira. Mesmo desta maneira, o usuário não deve mencionar como destino o nome de um dos arquivos de origem, porque, da mesma forma como no caso anterior, o Sistema Operacional compara sempre os nomes dos arquivos de origem com o nome do arquivo destino e fará sempre a gravação neste último, caso ele já exista previamente.

Em qualquer circunstância onde houver coincidência entre um arquivo de entrada (ou de origem) com o arquivo de saída (ou destino), o arquivo de entrada será OMITIDO do processo de concatenação.

Na anexação de arquivos, pode acontecer que a lista de nomes suplante os 128 caracteres permitidos na linha de comando do DOS. Quando isto acontece, a única saída será fazer a operação de concatenação em mais de uma etapa, criando entre elas um arquivo destino temporário, o qual será anexado aos demais na etapa seguinte:

A>COPY CARTA1.TXT+...CARTA5.TXT CARTAS.TXT

A>COPY CARTAS.TXT+CARTA6.TXT... CARTAS-F.TXT

Outro detalhe importante na concatenação, somente aplicável se os arquivos forem nomeados individualmente, refere-se aos argumentos /A e /B, inseridos como cláusulas na linha de comando, além de /V, já citado.

Cada um destes argumentos possui um significado para o Sistema Operacional, dependendo da maneira como seja empregado:

/A - Indica arquivo gravado no formato de texto (A vem de Ascii). Quando digitado como cláusula de um arquivo de origem, faz com que o Sistema Operacional trate o arquivo desta maneira, forçando-o a copiá- lo até encontrar a primeira marca de FIM DE ARQUIVO (END OF FILE - <control> + Z ou &H1A (26) na Tabela de Códigos ASCII). Depois de encontrada esta marca, o restante do arquivo, se existente, não será copiado. Por outro lado, quando indicado como cláusula do arquivo destino, determina que a marca de FIM DE ARQUIVO seja gravada ao seu término. ESTA CONDIÇÃO É DEFAULT NO MSX-DOS, NÃO PRECISANDO SER DIGITADA !

/B - Indica arquivo gravado no formato Binário. Quando indicado como cláusula em um arquivo de origem, força a gravação de todo o arquivo, incluindo as marcas de FIM DE ARQUIVO, mesmo que haja mais de uma dentro do mesmo arquivo. Quando se constitui em cláusula no arquivo destino, impede que a marca de FIM DE ARQUIVO possa ser gravada ao seu final.

Via de regra, não será necessário incluir as cláusulas anteriormente mencionadas no processo de anexação, sendo preferível, na dúvida de como empregá-las, deixar que o próprio Sistema Operacional se encarregue desta tarefa.

Arquivos em BASIC podem ser MERGEADOS (fundidos) por processo de concatenação diretamente no disco, mas deverão estar também gravados no formato ASCII, para que a fusão ocorra sem problemas. O usuário deve prestar atenção para ver se há coincidência de linhas entre os arquivos concatenados e acertá-las antes de efetuar a concatenação. Isto poupará o impedimento do correto funcionamento do programa mergeado. Consulte o comando SAVE, do BASIC DE DISCO, para saber como gravar um programa em BASIC no formato ASCII.

Comando MSX-DOS: DATE.

SINTAXE: DATE [dia-mês-ano]
DATE [dia/mês/ano].

SINONÍMIA: DATA (SOLX-DOS).

FUNÇÃO: Permitir ao usuário a renovação da última data a ele apresentada pelo Sistema Operacional.

DESCRIÇÃO:

Quando o DOS entra no ar e assume o controle operacional do computador, a primeira coisa que aparece na tela é a requisição, pelo Sistema Operacional ao usuário, da digitação da data.

A marcação da data no início das operações é importante para o registro da mesma no diretório dos disquetes, quando se efetua qualquer operação de gravação. Através deste recurso, pode-se saber qual a última vez em que os arquivos listados foram gravados ou atualizados com o uso do comando DIR. Neste particular, há uma diferença básica entre o MSX-DOS e o SOLX-DOS. No MSX-DOS, nas operações que envolvem gravação de arquivos por CÓPIA, a data registrada no diretório será a data do arquivo copiado e não a data atual, o que aliás é bastante razoável, já que isto permite saber em que data o arquivo em questão foi gravado do micro para o disco, ou atualizado (presumindo-se que nesta ocasião a data tenha sido digitada pelo operador). Com o SOLX-DOS, todas as datas de gravação, inclusive por CÓPIAS, serão necessariamente a data digitada pelo usuário no início da sessão com o DOS.

Se o usuário teclar <return> , ao invés de digitar a data quando solicitado, a data anterior, chamada, como explicamos no Capítulo 1, de "data atual" ou "current date", será mantida.

Por outro lado, quando algum arquivo de nome "AUTOEXEC.BAT" é encontrado no diretório, o DOS não passa pela requisição da data, impedindo o usuário de atualizá-la. Sendo assim, o único recurso será incluir no arquivo "AUTOEXEC" o comando DATE como o PRIMEIRO DA LISTA. Chamamos a atenção do leitor para que evite, neste caso e em outros parecidos, o emprego da SINONÍMIA em português dos comandos dos DOS, já que estes comandos são reconhecidos somente pelo SOLX-DOS, enquanto que aqueles digitados na sintaxe inglesa serão reconhecidos por todos os DOS indistintamente.

O MSX-DOS aceita a digitação da data no chamado formato europeu, com a colocação de dia, mês e ano, diferente da notação inglesa, que exige mês, dia e ano. O comando DATE permite que a data seja digitada diretamente, junto com o comando, facilitando assim a tarefa do usuário:

A>DATE 19/01/89

O comando DATE sem argumentos força a exibição da "data atual" ("current date"). Fazendo assim, o usuário consegue saber do Sistema qual a indicação da última data e se ela está ou não atualizada.

É indiferente digitar "-" ou "/" na separação de dia, mês e ano. Para os dois primeiros, é aceitável digitar apenas um número, enquanto que para o último, até quatro números poderão ser indicados.

Comando MSX-DOS: DEL.

SINTAXE: DEL [<drive:>] <nome dos arquivos>

TIPO: INTERNO.

SINONÍMIA: ERASE
APAGUE (SOLX-DOS).

FUNÇÃO: permite apagar um ou mais arquivos especificados na linha de comando.

DESCRIÇÃO:

O comando DEL apaga todos os arquivos citados pelo usuário. Como foi explicado

no Capítulo 1, a ação inicial do "apagamento" consiste na substituição do primeiro caractere dos nomes dos programas apagados pelo caractere de número 229 (&HE5) da Tabela ASCII, não havendo, portanto, a eliminação física do programa no disquete. Como a exibição dos nomes dos arquivos na listagem do diretório (comando DIR) é impedida, os arquivos deletados são considerados "ausentes" do disco, muito embora isto não seja a expressão da verdade, a não ser que, a seguir, o usuário grave algum outro arquivo no disco, o qual irá necessariamente ocupar o lugar do primeiro arquivo deletado.

Isto teoricamente dá chance ao usuário de recuperar o arquivo deletado antes que nova gravação seja feita. Esta recuperação só é possível ser feita com o auxílio de programas utilitários especificamente desenhados para esta finalidade. O potencial de cada programa varia muito e em disquetes muito mexidos, com gravações e regravações sucessivas, esta tarefa é quase que impossível. Por este motivo, o comando DEL é bastante perigoso quando usado indiscriminadamente. Além disso, usando caracteres-chaves, como * e ?, vários arquivos podem ser deletados de uma só vez. Apagando-se todo o disco (DEL *.*), ele se torna IRRECUPERÁVEL ! O DOS manda um sinal de alerta para evitar que um desastre aconteça, através da mensagem:

ARE YOU SURE (Y/N)? (MSX-DOS)
APAGA TUDO (S/N)? (SOLX-DOS).

Comando MSX-DOS: DIR.

SINTAXE: DIR [<drive:>] [<arquivo(s)>] [/P] [/W].

TIPO: INTERNO.

SINONÍMIA: não há.

FUNÇÃO: permite listar parcialmente informações constantes no diretório dos discos: nome dos arquivos, tamanho em Bytes, data e hora da gravação.

DESCRIÇÃO:

O comando DIR é um dos mais importantes do DOS e deveria ser o primeiro comando aprendido e assimilado pelos usuários iniciantes. Através dele pode-se conhecer o conteúdo total ou parcial da lista de arquivos constantes em um disco em exame. Tal procedimento deve ser PRIORITÁRIO antes da execução de qualquer

outro comando dado pelo usuário, para evitar desastres do tipo: apagar arquivos indevidamente, renomear um arquivo com o nome de outro arquivo previamente existente no mesmo disco, etc.

O DIR do MSX-DOS é um comando bastante eclético em relação ao seu congênere do CP/M. Não só o número de informações listadas é maior, como é possível, através de argumentação apropriada, listar o conteúdo dos arquivos de forma idêntica ao CP/M.

O comando DIR pode ser empregado de diversas formas. O usuário pode solicitar toda a lista de arquivos de uma só vez, partilhada em telas diferentes, ou ainda no formato de colunas. Pode-se pedir a listagem parcial de apenas um ou alguns arquivos, com referências ambíguas aos mesmos, utilizando-se dos caracteres-chaves mencionados no comando COPY.

Usando-se os caracteres "*" e "?", obtém-se listagens PARCIAIS, a não ser que seja digitada a opção "*.*", o que é ABSOLUTAMENTE DISPENSÁVEL (!), pelo fato da mesma estar IMPLÍCITA no comando. Através do emprego destes caracteres, é possível saber sobre a existência de determinados arquivos no disco de maneira bem mais facil, já que não é necessário vistoriar toda a lista de arquivos, o que, no caso específico de discos muito cheios, pode ser uma mão-de-obra considerável. Por exemplo:

A>DIR *.BIN (ou DIR .BIN)
verifica todos os arquivos binários gravados com a extensão .BIN no diretório.

A>DIR *.SYS (ou DIR .SYS)
verifica se foi gravado no disco algum Sistema Operacional com esta extensão.

A>DIR D*.* (ou DIR D*)
verifica se existe algum arquivo que comece com a letra D no disco.

A>DIR PROVA.* (ou DIR PROVA)
verifica se no disco existe algum arquivo de nome PROVA, com qualquer extensão.

Note que existem algumas analogias entre várias sintaxes empregadas, como mostrado acima, o que torna não obrigatório o emprego, em alguns casos, dos caracteres-chaves. Em todos os exemplos anteriores era exibida uma lista dos arquivos questionados, mas somente destes, enquanto que DIR sem argumentos listará obrigatoriamente TODOS os arquivos do disco, equivalendo portanto a DIR *.*.

O FORMATO DE EXIBIÇÃO também pode ser escolhido pelo usuário, o que é altamente conveniente em alguns casos. Dois argumentos estão disponíveis como cláusulas opcionais na linha de comando e que servem para esta finalidade: /P e /W. Ambos os argumentos podem ser usados isoladamente ou em conjunto, sendo que, aquele que for indicado primeiro, terá prioridade na formatação.

/P - Promove a listagem dos arquivos no formato de uma página de texto (Page=Página), com todas as informações que o comando permite. Isto equivale ao preenchimento da tela do computador com os nomes dos arquivos, até que uma tecla qualquer seja pressionada, quando então a listagem continuará, até que outra tela seja preenchida ou até que termine a listagem dos arquivos. Cada uma das listagens de tela pode ser parcialmente interrompida quando o usuário tecla <control> + S e depois reassumida, pressionando-se qualquer tecla.

/W - Promove a listagem dos arquivos no formato de janela (Window=janela), em número de colunas diretamente proporcional ao espaço disponível na tela. Neste caso, a exibição das informações sobre os arquivos fica restrita somente ao nome e extensão de cada um, sem o número de Bytes, data e hora das respectivas gravações.

É importante assinalar que o número de colunas varia de acordo com o DOS e de acordo com a largura (comando MODE) determinada pelo usuário. Em tela de 40 colunas, o número máximo exibido pelo MSX-DOS é de duas colunas, enquanto que o SOLX-DOS permite até três colunas. Conectando-se uma Placa de Vídeo de 80 colunas em um dos slots externos do MSX, o número máximo de colunas passa a ser de seis colunas, a despeito do DOS utilizado.

O formato de janela é utilizado pelo CP/M.

Em ambos os casos, o comando DIR fornece, como default, o número de Bytes livres de cada disco. No caso do SOLX-DOS, será fornecido também o número de Bytes consumidos pelo usuário com a gravação de arquivos.

Exemplos de linhas de comandos

A>DIR/W (Note que *.* está implícito no comando)

A>DIR/P (idem)

A>DIR/W/P
Neste caso, a listagem ocorrerá na forma de janelas,com interrupções a cada tela cheia. A argumentação contrária (/P/W) não é operacionalmente possível.

A>DIR *.BAS/W
A>DIR/P B:F1
} Note que é indiferente colocar o argumento /W antes ou depois da nomeação ambígua dos arquivos a serem listados.

A>DIR/W B: P*.*
FILE NOT FOUND
} Não é possível a digitação de espaço em branco após a indicação do drive, pois o DOS não achará o arquivo.

Na execução do comando DIR, teclando-se si-multaneamente <control> + S, a exibição da listagem é interrompida até que o usuário pressione qualquer tecla. Se for teclado <control> + C ou <control> + <stop>, a exibição da listagem é abortada.

Comando MSX-DOS: FORMAT

SINTAXE: FORMAT (MSX-DOS)
FORMAT [<drive:opção do Menu de Formatação>]
(SOLX-DOS).
FORMATE [<drive:opção do Menu de Formatação>]
(SOLX-DOS).

TIPO: INTERNO.

SINONÍMIA: FORMATE (SOLX-DOS).

FUNÇÃO: permite formatar um disco, levando em consideração o seu tipo e o drive utilizado.

DESCRIÇÃO:

O comando FORMAT marca trilhas magnéticas no disco, conforme explicado no Capítulo 1, de acordo com o padrão de formatação do computador, no caso, MS-DOS compatível.

A rotina de formatação e o acesso a ela estão gravados na interface de disco utilizada pelo usuário, de tal forma que se pode ter acesso a ela através do DOS ou do BASIC DE DISCO.

Na rotina de formatação, é exibido um MENU, no qual constam os tipos de formatação utitilizados pelos discos flexíveis de 5 1/4" e de 3 1/2", comumente conectados ao MSX. Os dizeres do Menu podem variar de um fabricante para outro, porém, as opções são sistematicamente iguais. O Menu exibido a seguir é aquele constante da controladora CDX-2 da Microsol Tecnologia:

1 - 40 trilhas simples face
2 - 40 trilhas dupla face
3 - 80 trilhas simples face
4 - 80 trilhas dupla face

As opções 1 e 2 servem para a formatação dos disquetes de 5 1/4", enquanto que as opções 3 e 4 são para os drives de 3 1/2".

Muitas vezes, o usuário é tentado a formatar o disquete de 5 1/4" com 80 trilhas. Fazendo isso, mesmo que a formatação pareça ter tido sucesso, serão observados erros de leitura posteriormente, pois o drive de 360 KBytes não foi projetado para tal tipo de formatação.

O MSX-DOS aceita apenas o comando FORMAT como meio de acessar o Menu de Formatação, porém, o SOLX-DOS permite que o usuário possa especificar diretamente o drive cujo disquete será formatado, como também a opção de formatação do MENU acima. Assim, se o usuário comandar:

A>FORMAT B:2

será formatado o disquete alojado no drive B, com 40 trilhas em face dupla.

CUIDADO:

ANTES DE FORMATAR UM DISCO JÁ USADO, VERIFIQUE SE NÃO EXISTE NELE NENHUM PROGRAMA IMPORTANTE, POIS AO FORMATAR UM DISCO, TODOS OS ARQUIVOS NELE EXISTENTES SERÃO TOTALMENTE APAGADOS !

IMPORTANTE:

A FORMATAÇÃO EXIGE UMA CORRETA ALIMENTAÇÃO ELÉTRICA E ALINHAMENTO DOS DRIVES. SE O SEU SISTEMA OPERACIONAL NÃO CONSEGUIR FORMATAR, VERIFIQUE A INTEGRIDADE DO EQUIPAMENTO, COMEÇANDO PELAS FONTES DE ALIMENTAÇÃO E CABOS DE CONEXÃO ENTRE DRIVE E INTERFACE.

O comando FORMAT, normalmente sem argumentos, solicita do usuário a seleção do drive e a opção de formatação, como anteriormente indicado. A seguir, pede que alguma tecla seja pressionada para se efetivar a operação. O usuário poderá abortar a formatação teclando <control> + C ou <control> + <stop>. O Sistema Operacional deverá emitir mensagem confirmando que ocorreu a interrupção solicitada. NA DÚVIDA, DESLIGUE O COMPUTADOR OU RETIRE O DISQUETE DO DRIVE !!!

CUIDADO:

NO CASO DE VOCÊ ESTAR OPERANDO COM O SOLX-DOS E COMANDAR A FORMATAÇÃO COM ARGUMENTOS NA LINHA DE COMANDOS, ESTA SE INICIARÁ SEM A NECESSÁRIA PAUSA, COMO ACIMA DESCRITO, BASTANDO APENAS TECLAR <RETURN>.

Comando MSX-DOS: MODE.

SINTAXE: MODE <Parâmetro>

TIPO: INTERNO.

SINONÍMIA: MODO <Parâmetro> (SOLX-DOS).

FUNÇÃO: Permite alterar o número de colunas no terminal de vídeo do MSX.

DESCRIÇÃO:

Fornecendo o número de colunas como parâmetro, o usuário consegue, através do comando MODE, modificar a largura da tela do terminal de vídeo, o que é útil em certos casos onde, na falta de um monitor, é ligada uma TV comum ao MSX, a qual possui limitações na visualização do conteúdo da tela.

O comando MODE opera aceitando um valor numérico como determinante do espaço máximo que será utilizado pelo terminal do MSX.

Além disso, valores abaixo ou acima de um determinado limiar determinam a seleção do tipo de screen de texto de forma automática.

Números fora daqueles permitidos pelo DOS retornarão ao usuário como erro, sendo emitida a mensagem de:

INVALID PARAMETER (MSX-DOS)
PARÂMETRO ILEGAL (SOLX-DOS).

Neste ponto, há uma diferença fundamental entre o MSX-DOS e o SOLX-DOS na aceitação destes valores:

MSX-DOS:

Valor mínimo - 0
Valor máximo - 40
Limiar de transição: 33 e acima - screen 0
32 e abaixo - screen 1

SOLX-DOS:

Valor mínimo - 32
Valor máximo - 80
Limiar de transição: 33 e acima: screen 0
32 - screen 1

Conectando-se uma placa de vídeo de 80 colunas em um dos slots do MSX, ou rodando-se o DOS mum MSX da série 2.0, é possível obter POR DEFAULT a listagem em 80 colunas, mesmo que seja utilizado o MSX-DOS. Entretanto, ao se usar este último, não se poderá comandar o MODE para 40 colunas, sob pena de não mais retornar a listagem ao formato original de 80 colunas, já que este parâmetro não é aceito por este DOS.

Operando sem placa de 80 colunas, o comando MODE provoca o apagamento da tela, com posicionamento do prompt nas coordenadas 0,0 (<home>). Quando a placa de 80 colunas VMX-80 da Microsol Tecnologia é instalada, a operação do MODE com ela determina o não apagamento da tela, mas mantendo o deslocamento do prompt para < home>, o que faz com que ela fique totalmente embaralhada.

Apesar disso, o usuário poderá digitar normalmente todos os comandos, sem qualquer prejuízo do funcionamento do DOS. Isto só é possível porque as instruções para o Sistema Operacional são recebidas diretamente do teclado e não lidas da tela

do computador, como acontece na digitação de programas em BASIC na tela do MSX. Procedendo assim, é muitas vezes difícil a legibilidade do resultado das operações comandadas. A solução, nestes casos, será teclar <return> sucessivamente, tantas vezes quantas forem necessárias, até que o prompt se desloque para uma região da tela abaixo de onde as linhas a seguir estejam em branco.

Trabalhando nas screens de texto normais do micro, o efeito acima descrito não será observado, mesmo com a placa de 80 colunas ligada a um dos slots.

Comando MSX-DOS: PAUSE.

SINTAXE: PAUSE [<comentário>]

TIPO: INTERNO.

SINONÍMIA: PAUSA [<comentário>] (SOLX-DOS).

FUNÇÃO: permite interromper um arquivo de comandos em lote (batch), facultando ao usuário a continuação do mesmo.

DESCRIÇÃO:

Este comando se encaixa no grupo daqueles que só têm utilidade para o usuário quando empregado na programação de um arquivo "batch". Em muitos momentos, a continuidade da execução do arquivo em lote deve ser propositalmente interrompida (mas não abortada), com o objetivo de permitir ao usuário a leitura de um aviso de qualquer natureza, como por exemplo:

A>PAUSE Coloque disquete no drive B.

Imediatamente após à execução da pausa, o Sistema Operacional emitirá uma mensagem do tipo:

STRIKE A KEY WHEN READY (MSX-DOS)
TECLE ALGO QUANDO PRONTO (SOLX-DOS).

Assim, ao teclar qualquer tecla, o comando seguinte será executado. Nesta oportunidade, o usuário pode interromper as execuções dos comandos, teclando <control> + C ou <control> + <stop>. Sendo este o caso, o DOS solicitará a confirmação

do término:

TERMINATE BATCH FILE (Y/N)? (MSX-DOS)
Encerrar arquivo BAT (S/N)? (SOLX-DOS)

O posicionamento do comando PAUSE nos arquivos em lote será melhor discutido posteriormente no tópico sobre este tipo de arquivo, mais adiante.

Comando MSX-DOS: REM

SINTAXE: REM [<comentário>]

TIPO: INTERNO.

SINONÍMIA: / [<comentários>]

FUNÇÃO: Permite colocar numa linha de programa do arquivo em lote (batch) uma mensagem, a qual será impressa na tela junto com o comando.

DESCRIÇÃO:

O comando REM também é daqueles que só têm utilidade no arquivo de comandos. Seu comportamento é semelhante à instrução PRINT do BASIC, exceto que não é possível omitir o nome do comando da linha do prompt (no MS-DOS, isto é factível pela existência de um comando interruptor do "eco", o "ECHO OFF"). Sendo o comentário opcional, o REM poderá ser utilizado SEM ELE, com o objetivo de espaçar linhas do arquivo batch, de modo a melhorar a legibilidade do mesmo.

Somente a vírgula, o espaço em branco e o tabulador poderão ser usados como separadores de palavras no comentário impresso pelo REM. Os exemplos de utilização do REM e da / serão apresentados no tópico sobre arquivos "Batch", na página 106 do próximo Capítulo.

Comando MSX-DOS: REN

SINTAXE: REN [<drive:>] <nome ANTIGO> <nome NOVO>.

TIPO: INTERNO.

SINONÍMIA: RENAME (com a mesma SINTAXE).

FUNÇÃO: Permite renomear um ou mais arquivos no disquete do drive especificado ou, na ausência deste, no disquete do drive corrente.

DESCRIÇÃO:

O REN é um importante recurso dos Sistemas Operacionais de disco, possuindo SINTAXES próprias para o DOS e para o BASIC DE DISCO. Ao deixar o usuário mudar o nome de um arquivo, o REN evita que o mesmo tenha que voltar à memória do computador para ser regravado com outro nome.

Mais de um arquivo do diretório poderá ser renomeado simultaneamente, com o auxílio dos caracteres-chaves "*" e "?", os quais podem ser empregados tanto no item relativo ao nome antigo, quanto no do novo nome. Por exemplo:

A>REN B:*.ARQ *.DOC

A>REN CAP2-?.DOC CAP2-?.TXT

É preciso uma certa cautela ao usar * e ? na renomeação. Quando os caracteres-chaves são indicados no lugar do NOVO nome dos arquivos, TODOS os caracteres dos nomes originais correspondentes às posições de * e ? NÃO SÃO MODIFICADOS.

Assim, os seguintes comandos não teriam efeito na renomeação:

A>REN *.BIN *.*

A>REN SAMPLE?.DOC *.DOC

O Sistema Operacional, ao reconhecer a coincidência, rejeitará a renomeação e emitirá a seguinte mensagem de erro:

RENAME ERROR (MSX-DOS)
ERRO AO RENOMEAR ARQUIVO (SOLX-DOS).

Esta mesma mensagem aparecerá quando ocorrer uma indicação de um nome antigo de arquivo que não se encontra no diretório, já que não é possível renomear um arquivo inexistente.

Quando o usuário indicar, no NOVO nome do arquivo, um DRIVE diferente daquele onde a operação estiver sendo feita, o DOS ignorará a indicação e fará a renomeação no drive corrente ou no especificado antes do antigo nome, isto porque não é também possível renomear arquivos ATRAVÉS de unidades de disco diferentes.

Comando SOLX-DOS: SAVE.

SINTAXE: SAVE <número de blocos> [drive:] <nome do arquivo>

TIPO: INTERNO.

SINONÍMIA: SALVE (com a mesma SINTAXE).

FUNÇÃO: Permite salvar uma região de memória do computador, a partir de &H0100, no disquete.

DESCRIÇÃO:

O comando SAVE tem função utilitária. Quando um programa feito pelo usuário estiver contido na área da memória destinada aos programas do DOS, isto é, entre &H0100 e &HFC4B, ele poderá ser gravado no disco na forma de um arquivo.

O programa deve ser dividido em blocos de 256 Bytes, e o número obtido, colocado na linha de comando. Para tanto, basta determinar o comprimento em Bytes do programa e dividir o resultado por &H0100 (ou 256, se a conta for feita em decimal).

Deve-se levar em conta que o início de armazenamento de qualquer programa em ambiente DOS será necessariamente &H0100. Assim, deve-se saber em qual endereçamento termina o programa a ser salvo, pois, calculando-se a diferença entre término e início, determina-se o seu comprimento em Bytes. Por exemplo:

Término do programa: &H5400
Comprimento: &H5400 - &H0100 = &H5300 (21248 Bytes)

Número de Blocos:
&H5300/&H0100 = &H53 (Nota: 53H=83D)
21248 / 256 = 83

O comando, neste caso, seria:

A>SAVE 83 PROG.COM

Erros de sintaxe retornarão como "Parâmetro ilegal". O comando deve ser executado antes que seja alterada a área de programas (TPA).

Comando MSX-DOS: TIME

SINTAXE: TIME [<hh:mm:ss>]

TIPO: INTERNO.

SINONÍMIA: não há.

FUNÇÃO: Permite atualizar o horário dos relógios internos nos computadores que o possuem, de modo que ele fique registrado no diretório, nas operações de gravação.

DESCRIÇÃO:

O comando TIME funciona de maneira semelhante ao DATE, já descrito anteriormente. TIME sem argumento retorna com o valor anterior do relógio interno ou, na inexistência dele, com:

12:00:00.00a (MSX-DOS)
00:00:00.00 (SOLX-DOS).

Os últimos dois zeros de cada horário referem-se a centésimos de segundo, os quais não são ajustáveis com o comando, até as presentes Versões dos dois DOS.

O horário das gravações deverá ser sempre exibido na tela através do comando DIR (ou ainda DIR/P), quando o seu valor no diretório for diferente de ZERO, e independentemente do fato de que o computador tenha ou não relógio interno.

Nos casos onde é possível a atualização do horário, é suficiente digitar somente a hora, ou somente a hora e os minutos. A digitação obedece os seguintes parâmetros:

hh - qualquer número entre 0 e 23; qualquer número entre 0 e 12, seguido de A (ou AM-parte da manhã) ou de P (ou PM - parte da tarde) (Ajusta a hora).

mm - qualquer número entre 0 e 59 (Ajusta os minutos).

ss - idem (Ajusta os segundos).

Note que os números de hora, minuto e segundo são separados por ":", enquanto que os centésimos de segundo são separados por ".", sendo esta a digitação correta na SINTAXE do comando. Horários fora deste formato, ou com números fora dos especificados, terão a mensagem:

INVALID TIME (MSX-DOS)
HORA INVÁLIDA (SOLX-DOS).

Teclando-se <return>, os valores anteriores são mantidos.

Quando o micro não tiver relógio interno, o DOS aceitará a digitação, porém esta não terá efeito.

Comando MSX-DOS: TYPE

SINTAXE: TYPE <nome do arquivo>

TIPO: INTERNO.

SINONÍMIA: LIST ou LISTE <nome do arquivo> (SOLX-DOS).

FUNÇÃO: Permite imprimir na tela o conteúdo de um arquivo gravado no formato ASCII.

DESCRIÇÃO:

O comando TYPE é útil para a observação e leitura de um arquivo cujo formato de gravação está na forma de caracteres da Tabela ASCII, o chamado FORMATO DE TEXTO ou FORMATO ASCII. O TYPE não permite, entretanto, que sejam feitas modificações de qualquer espécie nos arquivos inspecionados.

O comando TYPE funciona corretamente em telas de 80 colunas, o que facilita em muito a visualização da formatação de arquivos produzidos pelos Processadores de Texto e, PRINCIPALMENTE, se os arquivos-textos produzidos por um dado aplicativo gravam o arquivo com ou sem troca de caracteres.*

*** Observação:**
Muitas vezes passa despercebido para o usuário o fato de que a maioria dos Processadores de Textos, na hora da gravação dos arquivos, troca os caracteres que foram digitados por outros, utilizados pelo próprio programa, geralmente para codificar suas próprias formatações. Este fato, por si só, dificulta a migração (transporte) do arquivo-texto para outros computadores ou Processadores de Texto. Um dos raros Editores de Texto que não fazem isso é o MSX-WORD na Versão 3.0, e, na área dos Editores para programas-fonte, o MED (visto mais adiante), parente mais novo do SCED.

Outra aplicação muito útil do TYPE é a listagem de arquivos batch(.BAT), permitindo inclusive a exibição de caracteres acentuados, mesmo se isso não acontece quando o arquivo de comandos é executado.

O TYPE aceita a indicação de referências ambíguas a arquivos de disco. Com o uso de caracteres-chaves "*" e "?". Como porém somente um arquivo pode ser listado, o DOS procurará o PRIMEIRO do diretório que corresponder à referência e o apresentará na tela.

A listagem do arquivo solicitado preencherá a tela e rolará sem interrupção (efeito de "scrolling" contínuo), até que o arquivo termine. O usuário, entretanto, poderá interromper o "scrolling" teclando <control> + S e reiniciá-lo teclando qualquer tecla. Poderá também abortar a listagem, teclando <control>+<stop> ou <control> + C.

O modo de operação do comando TYPE é ligeiramente diferente no MSX-DOS e no SOLX-DOS. Enquanto no primeiro todo o arquivo é lido e DEPOIS EXIBIDO, no segundo, a exibição do arquivo é feita quase que simultaneamente à sua leitura do disco. Assim, com o SOLX-DOS, ao teclar <control> + S, o drive apagará até que o usuário mande continuar a leitura.

A saída do comando TYPE é exclusivamente para o terminal de vídeo do MSX. Contudo, teclando-se <control> + P, todas as operações do DOS terão ECO na impressora, já que este comando ativa esta saída. Assim, é possível listar um arquivo-texto em ambas as saídas, sendo que <control> + P pode ser teclado antes ou após o comando ser digitado na linha do prompt. Após a listagem, teclando-se <control> + N, a saída para a impressora é desativada.

Lembre-se: nem sempre a impressora é totalmente compatível com o computador, de tal forma que a listagem poderá aparecer truncada, com ausência ou troca de caracteres. Da mesma forma, arquivos fora do formato de texto serão exibidos de forma descontrolada e com caracteres desconexos na tela. É comum, nestes casos,

o envio de caracteres interpretados como códigos de controle do terminal de vídeo, tais como: bell (ou Beep), clear screen ou form feed (no vídeo, apaga a tela e na impressora, provoca a mudança de página), carriage return, line feed e outros.

Veja, portanto, que uma das muitas funções que o comando TYPE poderá trazer ao usuário, é descobrir se um determinado arquivo está gravado ou não no formato de texto.

O TYPE, digitado sem o nome de um arquivo ou com o nome de um arquivo não constante no diretório do disco, acarretará do Sistema Operacional uma mensagem de erro:

FILE NOT FOUND (MSX-DOS)
ARQUIVO NÃO ENCONTRADO (SOLX-DOS).

Comando MSX-DOS: VERIFY.

SINTAXE: VERIFY <ON> (habilita)
VERIFY <OFF> (desabilita)

TIPO: INTERNO.

SINONÍMIA: VERIFY [ON] [OFF] [/L] [/D]
VERIFICA [ON] [OFF] [/L] [/D]
(ambos do SOLX-DOS: /L-liga; /D-desliga).

FUNÇÃO: Permite definir o interruptor de verificação da gravação de arquivos nas operações de cópia.

DESCRIÇÃO:

O comando VERIFY produz uma mudança permanente no status de verificação de gravações por cópia, solicitadas pelo usuário. A habilitação da verificação, de outra forma, só poderá ser feita extemporaneamente (no momento da gravação), utilizando-se o argumento /V na SINTAXE do comando COPY:

A>COPY *.BAS B:/V

ON/OFF ou /L e /D, estes últimos exclusivos do SOLX-DOS, são na realidade parâmetros do comando e empregados na sua SINTAXE. Caso um deles seja

omitido, o DOS emitirá a seguinte mensagem de erro:

INVALID PARAMETER (MSX-DOS)
PARÂMETRO ILEGAL (SOLX-DOS).

Uma vez habilitada a verificação pelos argumentos ON ou /L, este status só poderá ser modificado por força da execução de um programa, ou por comando direto do usuário, em sentido oposto, usando OFF ou /D como parâmetros.

A verificação é feita setor a setor, o que torna a cópia um pouco mais lenta. Embora dificilmente o DOS erre ao copiar e, quando acontece algo do tipo "erro de escrita" (erro de gravação no disco destino), o usuário seja avisado, as vantagens da verificação são óbvias, porque ninguém gosta de se sentir inseguro quanto ao resultado das operações de cópia.

É preciso, na hipótese da ocorrência e constatação de erro na gravação, tentar avaliar a fonte do problema:

Quando este reside em defeitos na formatação do disco destino, a pura e simples troca por outro corretamente formatado será suficiente. Infelizmente, problemas de gravação por cópia podem ir desde o próprio DOS até o estado físico dos drives e fontes de alimentação. Proceda com cautela: troque primeiro o disco e, depois, o DOS (mude, por exemplo, do MSX-DOS para o SOLX-DOS) e, depois então, verifique o seu equipamento (geralmente, se o drive é de boa qualidade, o problema estará na interface controladora ou na fonte de alimentação).

A verificação de cópias é um bom recurso dos Sistemas Operacionais e pode ser ativada também pelo BASIC DE DISCO, ou empregada em programas copiadores de boa qualidade, em alguns casos, até opcionalmente, já que muitos usuários preferem sacrificar a segurança em prol da rapidez e também da confiabilidade na qualidade do Sistema Operacional em uso.

CAPÍTULO 3

UTILITÁRIOS E FERRAMENTAS DO DOS: PRIMEIRA PARTE

UTILITÁRIOS SOLX-DOS E MSX-DOS TOOLS

Introdução

O leitor foi alertado, nos Capítulos anteriores, para um fato importante: de que existem comandos INTERNOS (constantes do programa DOS propriamente dito) e EXTERNOS (constantes de programas independentes, que podem ser carregados como comandos e, uma vez terminadas as suas funções, devolvem o controle ao DOS), identificados de forma muito própria por este Sistema Operacional.

Essencialmente, os comandos INTERNOS são limitados EM NÚMERO, com o objetivo aparente de ocupar o mínimo espaço de memória RAM possível e EM FUNÇÃO, restringindo o DOS a realizar autonomamente apenas as operações fundamentais de entrada e saída. Estes dois fatores têm importância estratégica, sob o ponto de vista operacional:

Quanto mais memória livre disponível, maior será a área a ser ocupada pelo programa que funcionará neste ambiente, facilitando todas as operações que dela dependam. Na prática, isto significa que, na medida em que estas operações possam ser feitas DENTRO do computador, a velocidade de processamento será significativamente maior, pois as manobras de entrada e saída de dados serão contornadas. De forma similar,quanto menos arquivos de OVERLAY necessitarem ser lidos do disco, menos tempo será perdido para a execução de determinadas tarefas.

Por outro lado, o DOS "per se" deverá manter as condições corretas de funcionamento dos programas nos ambientes por ele criados, devendo possuir as rotinas necessárias para que isto aconteça.

A limitação em número e versatilidade dos comandos INTERNOS tornaria o DOS um programa igualmente limitado, não fosse a possibilidade de se adotar programações (ou rotinas) EXTERNAS capazes de realizar tarefas específicas, muitas vezes complementares àquelas do próprio DOS.

Assim, o Sistema Operacional DOS pode ser expandido ou atualizado, sem que, na realidade, a programação original de comandos tenha mudado.

Em contrapartida, toda vez que estes comandos forem utilizados, eles deverão estar necessariamente contidos em um disquete acessível pelo operador, caso contrário, será emitida uma mensagem de erro do tipo:

BAD COMMAND OR FILE NAME (MSX-DOS)
COMANDO OU ARQUIVO INEXISTENTE (SOLX-DOS).

Este último detalhe implica na existência de espaço de memória periférica em quantidade suficiente para armazenar todos os programas utilizados como comando externo. Neste ponto, leva enorme vantagem quem possui dispositivos de armazenamento com maior capacidade, como os "WINCHESTERS" (discos rígidos com dezenas de Megabytes de memória) ou mesmo como os expansores de memória do tipo "PSEUDO-DRIVE", podendo este último estar ao alcance dos MSX, no momento.

A maioria dos sistemas de disco em uso, entretanto, compreende apenas um e, no máximo, dois drives de 5 1/4", o que pode ser insuficiente para tornar exeqüíveis operações com comandos externos em número generoso.Sendo assim, o usuário deverá selecionar os programas do DOS de maior uso e utilidade, e mantê-los gravados juntos no disco que contém o Sistema Operacional em atividade.

Quando o usuário compra um drive, normalmente recebe o disquete contendo o Sistema e alguns utilitários. Os softwares deste tipo, desenvolvidos por terceiros, poderão ser adicionados e completar esta coleção. No caso de existirem arquivos "batch" num disco, com comandos externos incluídos, os arquivos a eles referentes deverão também constar do mesmo disco, preferencialmente, ou de outro disco indicado pelo arquivo "batch".

O objetivo deste Capítulo é apresentar os principais utilitários que o usuário poderá empregar para melhorar o desempenho do seu Sistema Operacional, todos eles na forma de COMANDOS EXTERNOS.

AVISO AOS LEITORES

Não é possível apresentar todos os utilitários existentes no mercado e nem prever a existência de novos comandos do DOS. Sendo assim, decidimos selecionar aqueles programas que poderão dar pelo menos uma idéia clara ao usuário de dois itens muito importantes: a)- de como dispor de softwares existentes no momento; b)- quais os critérios que levarão o leitor a exercer seu próprio julgamento sobre que utilitários serão convenientes a si mesmo.
Mesmo dentro dos tópicos apresentados a seguir, foram feitas seleções de títulos, pelos mesmos motivos declarados acima.

Todos os utilitários que trabalham como comandos EXTERNOS são como que "FERRAMENTAS DO DOS", pois ajudam ao usuário a trabalhar com arquivos de diversas procedências, através do Sistema Operacional. Na primeira Parte constante deste Capítulo, serão apresentados os Utilitários SOLX-DOS, os MSX-DOS TOOLS da ASCII CORP. e ,para finalizar, a criação de ARQUIVOS DE PROCESSAMENTO DE COMANDOS EM LOTE, os chamados ARQUIVOS "BATCH".

Na descrição dos utilitários, será seguida uma sistemática semelhante àquela empregada na listagem dos comandos do DOS, exceto pelo item "SINONÍMIA", que, neste caso, perde a razão de ser.

Lembre-se que no enunciado da SINTAXE, dois sinais gráficos acompanham o comando:

[e] - indicam cláusulas ou argumentos opcionais.
< e > - indicam dados que o usuário é obrigado a digitar no argumento utilizado.

Utilitários SOLX-DOS

São programas fornecidos pela MICROSOL, que acompanham o SOLX-DOS. Dos quatro arquivos constantes no disco de Sistema, apenas dois serão descritos a seguir. Os outros dois referem-se a arquivos em lote (.BAT), que o leitor poderá inspecionar através do comando TYPE do DOS, não cabendo aqui maiores comentários.

Utilitário SOLX-DOS: CONVSOL, Versão 2.0.

SINTAXE: CONVSOL.

TIPO: EXTERNO.

FUNÇÃO: permite ler discos formatados em padrões CP/M semelhantes ao SOL/M ou ao SISTEMA 700, e gravar os arquivos lidos em disquetes com formatação MS-DOS compatível (adotada pelo MSX).

DESCRIÇÃO

O programa CONVSOL foi desenvolvido com a finalidade de permitir ao usuário a transferência de arquivos originalmente gravados em CP/M, para disquetes formatados no MSX e vice-versa.

As formatações em CP/M lidas pelo CONVSOL são aquelas compatíveis com a do padrão SOL/M (usada pelo CP-500) e a do S-700 (Sistema 700), o que significa discos em CP/M formatados com 40 trilhas (padrão SOL/M) ou 35 trilhas (padrões SOL/M e S-700).

Na Versão 1.1 do CONVSOL há uma mensagem em destaque na tela de abertura, avisando que "DISCOS PADRÃO APPLE NÃO SÃO ACEITOS". Nesta Versão (e na 1.0), o programa só conseguia ler discos formatados em FACE SIMPLES, e a conversão só se dava no sentido CP/M->DOS, o que foi corrigido na Versão 2.0, que processa a conversão de discos formatados em face dupla também e em ambos os sentidos. Por este motivo, somente a Versão 2.0 é descrita a seguir:

Digitando-se CONVSOL na linha de comando do DOS, o programa é carregado, exibindo uma tela de abertura com um Menu inicial, onde constam as seguintes opções:

0 .. SOL/M 40T SF
1 .. SOL/M 40T DF
2 .. S-700 35T SF
3 .. S-700 35T DF
4 .. F I M

Sua opção:

Ao escolher uma opção de conversão, aparecerá um segundo Menu, com as opções

de sentido da conversão, ou seja, a partir de que formatação será feita a leitura e em qual formatação será feita a gravação. Exemplo:

0 .. MENU 1
1 .. CONVERSÃO CP/M - SOLXDOS
2 .. CONVERSÃO SOLXDOS - CP/M

Sua opção:

Após a digitação da segunda opção, o programa solicita o nome do arquivo a ser convertido, com a pergunta:
Nome do arquivo fonte ?

Neste ponto, o usuário tem um importante recurso: o da obtenção do DIRETÓRIO do disco fonte, bastando para isso digitar DIR e teclar <return> no lugar do nome do arquivo. Além disso, como o CONVSOL informa ao Sistema Operacional como ler o disco fonte, é possível ler o diretório do disco fonte seja qual for o drive em uso (por exemplo: B:DIR <return>).

O CONVSOL permite que sejam adotadas referências ambíguas na operação de conversão, da mesma forma que estas referências são utilizadas no comando COPY do DOS. Por exemplo:

Nome do arquivo fonte ? A:*.*

Nome do arquivo fonte ? B:DB*.*

A seguir, o usuário é solicitado a digitar o nome do arquivo destino, só que, neste caso, isso só será preciso caso haja necessidade de RENOMEAÇÃO do arquivo fonte. Mais importante do que a renomeação, na maioria das vezes desnecessária, É A INDICAÇÃO DO DRIVE DESTINO ! Por exemplo

Nome do arquivo destino: B:

Tal fato permite que o usuário possa especificar o MESMO DRIVE como fonte e destino, da mesma forma como no comando COPIED do SOLX-DOS. Se isto for feito, o CONVSOL pedirá a troca de disquetes CP/M ou DOS, de acordo com a natureza da operação realizada.

Se tudo correr bem, o programa emite mensagem para aviso de ...
Fim de transferência.

Teclando-se <return> no lugar do nome do arquivo fonte, o programa volta ao Menu anterior, permitindo a escolha de uma nova opção ou a volta ao Menu inicial, e daí a outras opções, incluindo o fim do programa, como mencionado.

Observações:

De posse de um disco CP/M para conversão, pode acontecer de não se saber antecipadamente se o mesmo foi gravado em face simples ou dupla, e se foi formatado com 35 ou 40 trilhas. Como não há no CONVSOL um algoritmo capaz de proceder a esta identificação, o usuário só tem como opção partir para o método da tentativa e erro.

A primeira coisa que se deve fazer é proteger o disco fonte contra gravação, para evitar que, caso algo saia errado, não hajam danos ao mesmo.

Em seguida, deve-se observar atentamente se, com a opção de conversão escolhida, a leitura do disco fonte é feita sem dificuldades. Caso esta indicação não apareça claramente durante a leitura, deve-se de qualquer maneira testar a integridade dos arquivos convertidos.

Se estes arquivos forem programas, eles deverão rodar sem nenhum problema, a não ser, evidentemente, que haja algum fator contra, como por exemplo, ausência de terminal de vídeo de 80 colunas, falta de instalação neste terminal ou em áreas chaves do programa, rotinas de acesso ao BIOS do DOS diferentes daquelas previstas pelo Padrão MSX, etc.

O item INSTALAÇÃO deverá ser checado pelo usuário. Dependendo do programa, a simples conexão de uma placa de 80 colunas será suficiente para rodá-lo sem necessidade de outros tipos de adaptação. Em outros casos, deve-se recorrer ao Manual de Instalação e ao programa instalador, quando ambos estão disponíveis.

A migração de programas em CP/M para ambiente DOS trouxe ao usuário a oportunidade de operar com aplicativos profissionais de alto nível. Várias software houses interessaram-se em não só proceder as adaptações (e melhorias) que o MSX exige no desempenho destes programas, como até confeccionar Versões contendo novos comandos, visando assim aproveitar todo o potencial da máquina.

Observação final: teclando-se <control> + C ou <control> + <stop>, é possível abortar o funcionamento do CONVSOL, voltando-se ao prompt do DOS. Na hipótese de funcionamento errôneo nas operações de leitura subseqüentes, é recomendável resetar o computador.

Utilitário SOLX-DOS: COPIARQ Versão 1.2.

SINTAXE: COPIARQ [<linha de comando>] (ver DESCRIÇãO).

TIPO: EXTERNO.

FUNÇÃO: Permite efetuar operações de cópia, usando como dispositivos os drives e a impressora.

DESCRIÇÃO

O leitor foi avisado, durante a descrição do comando COPY do DOS, que as operações de concatenação de arquivos não eram possíveis através do SOLX-DOS. Para contornar este inconveniente, este programa foi desenvolvido e não só é capaz de processar as concatenações, como também funciona como um copiador sofisticado, incluindo a saída formatada para uma impressora.

A SINTAXE na linha de comando dependerá do tipo da operação a ser realizada pelo usuário. O acesso a uma única operação poderá ser feito diretamente na linha do prompt do DOS. Por exemplo:

A>COPIARQ *.BAS B:(P)

No caso de se desejar mais de uma operação de cópia, é altamente recomendável carregar primeiro o programa, através do comando:

A>COPIARQ <return>

Fazendo isso, o COPIARQ ficará totalmente residente no micro e passará a interpretar as linhas de comando digitadas no seu próprio prompt: um ponto (".").

Caso o usuário deseje abandonar as operações a qualquer momento, será suficiente teclar <return> com o prompt do COPIARQ vazio (ou ainda os costumeiros <control> + C ou <control> + <stop>).

O prompt do COPIARQ aparece logo após as mensagens do fabricante. As SINTAXES listadas a seguir serão divididas de acordo com a sua função:

Operações disco a disco

SINTAXE GERAL
[<drive>:]<arquivo-fonte> [PARA] [,][<drive>:][<arquivo-destino>] ([P],[E],[V],[M],[N],[I])

O COPIARQ faz cópias das mais simples até as concatenações, com a opção de colocação de argumentos na linha de comando, sob a forma de parâmetros de cópia, representados pelas letras P,E,V,M,N e I. A citação de um ou mais arquivos para cópia se faz de forma semelhante àquela utilizada no comando COPY, exceto que o COPIARQ aceita também as cláusulas PARA ou ",".

Assim, para uma cópia de um ou mais arquivos, três SINTAXES serão admissíveis:

.*.BIN B:
.*.BIN PARA B:
.*.BIN,B:

A adição de parâmetros como argumentos irá depender do interesse do usuário e da sua aplicação. Um ou mais argumentos poderão ser colocados na linha de comando entre parênteses, com a separação opcional entre eles por uma vírgula.

O significado de cada parâmetro é o seguinte:

P - Solicita do usuário a permissão para copiar cada um dos arquivos solicitados. É muito útil quando se deseja inspecionar a listagem dos arquivos lidos, obtida pela menção de uma referência ambígua na linha de comando. A cada arquivo listado, o COPIARQ emite uma mensagem, com a solicitação:

DBASE.COM Copia? (s/n/f)

As opções s/n/f referem-se à resposta que o usuário é obrigado a dar: "s" permite que a cópia seja feita; "n" impede a cópia, passando a listagem ao arquivo seguinte com a mesma pergunta, e "f" obriga o programa a abandonar a operação corrente de cópia. Exemplo de comando:

. .*.* B:(P)

E - Exime o COPIARQ da criação de arquivos tempórarios, durante a operação de cópia: normalmente, a cada arquivo que passa para o disco destino, o COPIARQ cria um arquivo de nome RASCUNHO.SOL, onde o arquivo fonte é gravado; terminada a operação e se tudo correu bem, este arquivo é renomeado com o nome do arquivo fonte. Quando existir, no disco destino, um arquivo com o mesmo nome do arquivo fonte, a gravação se realizará em cima deste. O parâmetro "E" impede que a verificação seja feita pelo método acima citado. Este fato contra-indica a sua aplicação, correndo-se o risco de danos ao arquivo destino.

Este parâmetro não pode ser usado na CONCATENAÇÃO de arquivos. Se o usuário assim o fizer, será exibida uma mensagem de erro:
O parâmetro "E" é incompatível com merge.

V - Verifica a existência, no disco destino, de arquivos do disco fonte solicitados para a gravação. Se houver coincidência, a gravação não se realizará e será emitida uma mensagem de aviso ao usuário:

Arquivo destino existe

IMPORTANTE: este parâmetro só funcionará quando for feita uma REFERÊNCIA AMBÍGUA aos arquivos do disco fonte a serem copiados. Neste caso, toda vez que for detectada uma coincidência, o COPIARQ passará à gravação do arquivo seguinte na lista. Exemplo:

.*.TXT B:(V)
CAP1-1.TXT
Arquivo destino existe
CAP1-2.TXT
CAP1-3.TXT etc.

OBSERVAÇÃO: a ausência, não sabemos por que cargas d'água, do comando DIR durante o funcionamento do COPIARQ, torna o parâmetro "V" um argumento obrigatório na maioria das operações de cópia, a não ser quando a intenção do usuário é fazer "backups" atualizadas do disco fonte (arquivos de texto, por exemplo).

M - Processa a troca, durante a operação de cópia, de letras maiúsculas do arquivo fonte por letras minúsculas no arquivo destino. Este parâmetro só tem utilidade no caso de um pré-processamento de arquivos texto. A saída poderá ser também direcionada para a impressora, bastando para isso a mudança na sintaxe do comando.

N - Idem ao anterior, mas com sentido inverso, ou seja, troca de minúsculas para maiúsculas.

I - Permite a cópia de arquivos do disco fonte para o disco destino, usando um só drive, especificado pelo usuário. O COPIARQ se encarregará de pedir ao operador a troca de disquetes. Por exemplo:

```
.*.BAS A:(I)
MONITOR.BAS
Ponha o disco fonte e tecle enter:
Ponha o disco destino e tecle enter:
etc.
```

Concatenação de arquivos

Como foi visto no comando COPY do DOS, é possível fazer a concatenação de arquivos de duas formas, mas somente aquela realizada com o auxílio do sinal de "+" poderá ser utilizada através do COPIARQ.

Se o usuário tentar mergear arquivos através da cópia por referência ambígua, como na sintaxe do comando COPY, será emitida uma mensagem de erro. Por exemplo:

```
.*.DOC B:SOMA.DOC
Origem ambígua e destino especificado ???
```

Todas as regras que mencionamos para a concatenação de arquivos, no Capítulo anterior, deverão ser observadas. Os parâmetros de gravação do COPIARQ acima citados poderão ser utilizados, bem como a especificação dos drives onde os arquivos estão contidos.

Quando o nome do arquivo destino não for mencionado na linha de comando, o primeiro nome da lista será adotado. Neste ponto, o usuário deve prestar a devida atenção, para evitar que este arquivo seja gravado no disco que contém o primeiro arquivo da lista, já que, isto acontecendo, este deixará de existir em detrimento do arquivo destino.

A SINTAXE para a concatenação é a mesma do comando COPY, admitidas as variações do COPIARQ. Exemplo:

```
.TEMA1.TXT+B:TEMA2.TXT+TEMA3.TXT B:TEMAS.TXT
```

Cópias de arquivos para a impressora

Se a impressora utilizada pelo usuário é perfeitamente compatível com o MSX em caracteres acentuados (somente na Tabela original ou com um filtro de impressão em ambiente DOS), este recurso do COPIARQ poderá ser bastante útil, pois a saída para a impressora pode ser formatada com o auxílio de parâmetros apropriados, para os quais podem ser estipulados valores adequados.

A SINTAXE, neste caso, prevê como destino o dispositivo IMP:, exclusivo do programa, não sendo admitidos os dispositivos PRN:, LST: e quaisquer outros com comunicação com impressoras. Exemplos:

.CREDITO.DOC IMP:
.CREDITO.DOC PARA IMP:
.CREDITO.DOC,IMP:

Os parâmetros de impressão são aqueles listados a seguir:

F [<nnn>]

Define o tamanho da folha impressa, calculado em função do número de entrelinhas. <nnn> pode ir de ZERO a 255.

Quando <nnn> for ZERO, todos os demais parâmetros de impressão serão ignorados, impedindo a formatação da página. Isto significa dizer que nenhum código de controle será enviado à impressora pelo COPIARQ.

Quando <nnn> for diferente de ZERO, ele estipulará quantas linhas serão impressas em cada folha, após o que o COPIARQ envia um código de FORM FEED (alimentação de formulário - CHR$(12) na Tabela ASCII).

Se <nnn> não for especificado, será assumido o valor default para os formulários A4, que é de 55 entrelinhas.

T [<nnn>]

Define a tabulação horizontal, calculada em função do número de colunas a partir da margem esquerda. <nnn> pode variar de ZERO a 256; no entanto, quando o valor ZERO for indicado, a tabulação assumirá <nnn> igual a 256 colunas.

Quando<nnn> for omitido, será assumido o valor default para os formulários A4, de 8 colunas.

Observação:

A tabulação é conseguida pelo acionamento da impressora a posições tabuladas, pelo envio do código de controle CHR$(9) da Tabela ASCII. Dependendo da impressora, poderá ser necessário uma pré-programação, para que a mesma possa obedecer a este código.

L<"texto para o cabeçalho">

Permite que o usuário imprima um cabeçalho a cada topo de formulário, incluindo o primeiro (não é possível desativar a impressão na primeira folha, como acontece em alguns Processadores de Texto). O texto deverá ser indicado entre aspas. Não há indicação de limite de caracteres e nem maneira de centrar o cabeçalho no formulário.

R<"texto para o rodapé [numeração das páginas]">

Permite que o usuário digite um texto para cada rodapé de formulário, com opção de numerar as páginas de maneira formatada.

A formatação e a indicação da numeração é conseguida com o emprego de cerquilhas ("#") dentro das aspas e ao lado do texto. O número de cerquilhas definirá a formatação da paginação. Quando o número de algarismos impresso na página for menor que o número de cerquilhas, a paginação será justificada à direita, com preenchimento de zeros à esquerda do número. Por ex.: R"Página: ###", será impresso: Página 001, etc.

As mesmas restrições de impressão do cabeçalho são igualmente percebidas, como a ausência de centragem do texto.

Parâmetros M e N

São empregados da mesma forma que na cópia de arquivos, trocando maiúsculas para minúsculas e vice-versa.

Exemplos de SINTAXES:

.PROVA1.TXT IMP:(F65L"primeira Prova Parcelada"R"Página: ##")

.CONVITE.TXT IMP:(F20R"DATA: 08/02/89")

.B:W3-2.DOC,IMP:(F50L"TERCEIRO TRABALHO-SEGUNDA PARTE"R"###")

Depois de teclado <return> , o COPIARQ emitirá uma mensagem confirmando o envio do texto para a impressora:

Saída para dispositivo

O usuário deverá preparar a impressora previamente, deixando-a ligada e em linha com o micro.

Observações gerais e mensagens de erro

As linhas de comando digitadas pelo usuário poderão ser repetidas integralmente, teclando-se "R" ou "r" + <return> , no prompt do programa.

Qualquer linha de comando digitada fora dos parâmetros do COPIARQ, poderá ter duas interpretações:se houve erro de digitação, o programa emite mensagem de "Erro de sintaxe"; se parâmetros forem usados indevidamente, o COPIARQ avisará ao usuário e pedirá a confirmação da continuação:

Parâmetro(s) incompatível(is) para disco, continua ?
Parâmetro(s) incompatível(is) com impressão, continua ?

Em outros casos, poderá haver indicação, pelo programa, de parâmetros errados ou inexistentes.

Note que não há necessidade de se mencionar o drive CORRENTE nas operações com o COPIARQ, o que o leitor poderá observar em TODOS os exemplos que fornecemos.Todo o cuidado deve ser tomado neste sentido, para evitar que este drive seja trocado. Na dúvida, é recomendável mencionar os drives fonte e destino em cada linha de comando.

As demais mensagens de erro são bem parecidas com aquelas normalmente emitidas pelo DOS, como por ex., Disco (ou diretório) cheio, Erro de leitura, Arquivo não existe, etc.

Ferramentas do MSX-DOS: MSX-DOS TOOLS.

Um série de programas, na forma de comandos externos, são comercializados pela ASCII CORP., uma empresa japonesa que licencia a marca MSX. Como todos os programas são utilitários, eles são, por justo motivo, denominados de MSX-DOS TOOLS, fazendo aí, aparentemente, uma analogia com o PC TOOLS.

Os TOOLS (ferramentas) do MSX são vendidos por várias software houses domésticas, da mesma forma que os jogos, isto é, sem o devido licenciamento, o que, aliás, é ruim para o consumidor, devido à ausência de um Manual e de suporte técnico. Por outro lado, não há, pelo menos que seja do nosso conhecimento, um representante desta empresa no Brasil, ficando difícil um contato entre o usuário e o fabricante.

Apesar disso tudo, decidimos incluir neste livro a descrição de alguns desses TOOLS, pelo simples fato de que eles estão perfeitamente encaixados no conceito de comandos externos do DOS, podendo até, em certos casos, serem usados na confecção de arquivos de comandos. Além disso, e talvez como aspecto mais importante, os TOOLS do MSX têm nome e função (com pequena variação de sintaxe) idênticos a vários comandos do MS-DOS, como por exemplo: CHKDSK, DISKCOPY, CLS, MORE, etc. Por isso, dependendo do grau de compatibilidade entre eles, é perfeitamente factível rodar um arquivo de comandos feito no MSX em qualquer PC, facilitando assim o intercâmbio de programas.

Os MSX-DOS TOOLS possuem tela de ajuda que o usuário poderá aprender a acessar digitando HELP na linha de comando do DOS, ou então HELP <comando>, ou ainda <comando>/H. As telas não são, a nosso juízo, bem escritas, chegando a confundir o ajudado com termos do tipo "standard input" (entrada padrão) ou "standard output" (saída padrão), colocados lá sem uma definição concreta. É nesse ponto que faz falta um bom manual ou serviço de apoio ao usuário.

Como o nosso objetivo não é fazer um manual, ficaremos restritcs aos comentários essenciais, e apenas de alguns dos TOOLS descritos a seguir.

A lista dos TOOLS fornecida é muito grande, mas poderia ser dividida, por nosso arbítrio, de acordo com as funções a que eles se destinam:

Ferramentas de programação

BSAVE - transforma um arquivo contendo códigos de programação em linguagem de máquina dos microprocessadores INTEL 8080 e ZILOG Z-80, num arquivo em linguagem de máquina carregável e executável pelo comando BLOAD do BASIC.

DUMP e PATCH - Ambos fornecem a listagem de arquivos em disco, no formato hexadecimal e ASCII. O DUMP só permite a visualização do arquivo, enquanto que o PATCH permite a EDIÇÃO.

LIST - permite listar um programa em BASIC, gravado no disco em formato binário.

M80, CREF80, L80 e LIB80 - são quatro programas que operam em conjunto, com o objetivo de montar ("assemblar"), ligar ("linkar") e recorrer a bibliotecas de rotinas em assembler. São de uso restrito aos programadores em códigos de máquina e podem já estar defasados em relação a assembladores mais poderosos. O Manual do M-80 é extenso, indigesto e difícil de ser obtido pelo público. O M-80 é útil para compilar programas em ASSEMBLY, para rodar em ambiente DOS ou, através do programa BSAVE (acima), para rodar no BASIC DE DISCO.

Ferramentas de manipulação de Texto

HEAD, BODY e TAIL-são três utilitários para listagem de arquivos de texto, de aplicações muito semelhantes e de relativa pouca utilidade. A diferença entre eles reside na contagem das linhas do arquivo de texto: HEAD e TAIL contam as linhas a partir de início e fim, respectivamente, enquanto que BODY permite a especificação de início e fim de contagem de linhas. Nenhum destes utilitários altera o arquivo gravado, apenas mostrando a sua saída na tela ("standard output") ou na impressora, se esta saída for ativada por <control> + P.

UNIQ - remove e conta linhas duplicadas de um arquivo de texto, o qual, entretanto, permanece intacto.

TR e EXPAND - ambos os programas devem vasculhar os arquivos de texto, o primeiro, para trocar cadeias de caracteres (strings) e o segundo, para trocar tabulações existentes por espaço em branco e vice-versa. Não há indícios nas sintaxes de especificação de um arquivo de saída. Nas averiguações que fizemos sobre o funcionamento destes programas, constatamos que ambos não funcionam nem na "saída padrão" (tela do computador), por motivos que desconhecemos.

SORT - funciona de forma semelhante ao seu congênere do MS-DOS, exceto que não permite a criação de um arquivo de saída, fazendo com que este utilitário tenha pouca aplicação. Seu objetivo é "sortar" (termo do "computês", que significa ORDENAR um arquivo de texto de forma crescente (A-Z) ou reversa, tendo como parâmetros a tabela ASCII ou o alfabeto. Neste ponto, este programa é mais versátil que o do PC, pois este ignora se os caracteres são maiúsculos ou minúsculos, enquanto que o do MSX permite a opção pelos códigos ASCII, onde os caracteres maiúsculos têm valor menor do que os minúsculos.

MED - MICROSOFT EDITOR - este é um utilitário que poderia ser considerado um programa aplicativo, já que permite a edição "full screen" de arquivos de texto. Sua importância no nosso contexto é grande, pois através dele poderemos criar arquivos de comandos em lote com muita facilidade. Por este motivo, daremos a ele um tratamento especial, mais adiante.

VIEW - permite visualizar um arquivo de texto, página a página, usando os recursos do teclado.

Ferramentas que operam como comandos do DOS

BEEP - emite o código de controle CHR$(7) à saída padrão (tela), sendo aplicável em arquivos batch, para avisar ao usuário sobre o término de alguma operação, por exemplo.

CLS - emite o código de controle CHR$(12) à saída padrão (com opção de redirecionamento à impressora).

ECHO - permite a impressão de mensagem em arquivos batch de forma idêntica ao seu congênere do MS-DOS, exceto que ele opera de forma incompleta: enquanto no PC o ECHO pode desligar a impressão de comandos do DOS no prompt, o que é útil para se fazer telas de apresentação, no MSX, o ECHO é um mero similar do comando REM, não tendo assim muita valia, já que este último é INTERNO.

KEY - permite as funções do comando KEY ON/OFF e as respectivas redefinições das PFs, sendo por isso muito útil.

MORE - duplica com vantagens a operação do comando TYPE, já que permite a exibição controlável do arquivo-texto por página; é igual ao MORE do MS-DOS.

LS e MENU - ambos exibem o diretório de um disco, sendo que o LS classifica a listagem de acordo com os critérios impostos pelo usuário e o MENU, além de listar pelos ditames do usuário, permite várias operações com os arquivos.

SLEEP - ativa um temporizador, podendo ser usado em arquivos batch com esta finalidade.

CHKDSK - verifica num disquete a existência de clusters perdidos, com o objetivo de recuperá-los; deve ser usado com precaução, como comentaremos mais adiante, neste Capítulo. Possui um análogo no MS-DOS, com funcionamento semelhante.

DISKCOPY - executa a cópia física de disquetes em um ou dois drives, além de paralela ou isoladamente verificar o resultado da operação; neste último ponto, o programa duplica a função de comparador entre dois disquetes, tornando-se equivalente ao DISKCOPY e ao DISKCOMP do MS-DOS, respectivamente, neste último caso, com a omissão da operação de cópia.

Ferramentas sortidas

BIO - calcula e imprime o biorritmo de acordo com as especificações dadas pelo usuário.

CALC - permite ao usuário a realização de operações matemáticas na linha de comando do DOS.

CAL - imprime um calendário, nas especificações prescritas pelo usuário.

Observações:

Os MSX-DOS TOOLS da ASCII se auto-adaptam às condições de tela do usuário, um recurso dos mais interessantes. Os programas são capazes de reconhecer se a saída do terminal do MSX é feita com o auxílio de uma placa de 80 colunas, aumentando a capacidade de funcionamento e modificando a disposição do texto no vídeo.

Em 80 colunas, certos TOOLS melhoram muito o seu desempenho. O Editor de Textos MED, por exemplo, aumenta o tamanho da régua que "mede" a coluna do cursor. Já DUMP e PATCH passam a operar com maior números de Bytes por tela, enquanto que BIO e CAL fornecem, respectivamente, um gráfico aumentado e um calendário melhor desenhado. Como bonificação, o usuário poderá dispor dessas 80 colunas também na impressora, bastando para isso ativar o "eco" de caracteres do DOS através da saída para a impressora (<control> + P).

Todos os TOOLS que aceitam a não especificação de arquivos na entrada, passam alegadamente a trabalhar com a "entrada padrão" ("standard input"), só que, NESTE CASO, a entrada padrão não se refere a comandos do teclado, mas sim a REFERÊNCIAS AMBÍGUAS na nomeação dos arquivos. Quando isto é feito, geralmente o programa especifica, no início de cada operação, o drive e o arquivo a que ela se refere. Algumas vezes, é possível omitir esta indicação.

A seguir, particularizaremos alguns detalhes importantes a respeito dos TOOLS.

PATCH e DUMP

Estes utilitários lêem o arquivo do disco e o dispõem em linhas ao longo da tela, em três colunas distintas: a primeira com a numeração dos Bytes em hexadecimal, a segunda, com os Bytes propriamente ditos, em hexadecimal e a terceira, com os mesmos Bytes, traduzidos para caracteres ASCII (códigos abaixo de 32 são representados por "." nesta coluna).

Como a contagem dos Bytes se inicia em 0000, o usuário, para poder identificar corretamente os valores desejados, deve descontar o endereçamento inicial do programa. Por exemplo: todo programa em ambiente DOS começa a ser carregado em &H0100. Assim, esse será o Byte 0000 dos programas. Um endereçamento &H0247 estará no Byte &H0147, pois &H0247 - &H0100 = &H0147.

No PATCH, somente valores em hexadecimal poderão ser modificados. Não há perigo em usar este programa, desde que se saia dele teclando<esc> + Q (ou q) +<return> , para evitar gravar o arquivo de entrada com modificações erradas.

NUNCA faça modificações no arquivo original: trabalhe sempre com uma cópia de segurança (backup).

MED - MICROSOFT FULL SCREEN EDITOR

Este é um programa sobre o qual vamos fazer algumas observações bastante relevantes:

O MED é uma ferramenta das mais poderosas para o programador, pois permite a redação de arquivos de texto como programas-fonte de qualquer natureza, incluindo aquele que nos interessa mais de perto: os arquivos de comandos do DOS.

O MED é bastante aparentado ao SCED, possuindo muitos comandos equivalentes e outros bem diferentes em sintaxe, porém com a mesma finalidade. Nenhum dos dois Editores é adequado para a confecção de arquivos-texto fora daqueles acima mencionados, pelo simples fato de não possuírem justificação da margem direita.

A ausência deste alinhamento é coerente com o fato de que este tipo de Editor se presta exclusivamente à redação de arquivos não-documento, podendo no máximo

remendar arquivos-texto PREVIAMENTE FORMATADOS por outro Editor. Pelo mesmo motivo, não há também recursos de impressão, fato este que frustará aqueles usuários que insistam em utilizá-lo como Processador de Textos equivalente aos dedicados à formatação de textos para impressão.

Os textos a serem digitados com o MED de forma correta são linhas de programa. O usuário poderá escrever cada linha sem qualquer interrupção até que a mesma termine, neste caso, respeitando os limites da linguagem de programação utilizada. A cada linha digitada, deve-se teclar <return> para iniciar uma nova linha de programa. O cursor, coerentemente, se posicionará na primeira coluna à esquerda. O digitador deve se guiar sempre pela régua disposta na linha inferior.

O usuário poderá digitar direto na linha de comando do DOS o nome do arquivo que deseja criar:

A>MED B:AUTOEXEC.BAT

Opcionalmente, poderá solicitar o início do programa sem especificar qualquer nome de arquivo. Na tela de abertura, é oferecida novamente esta oportunidade, que o usuário ainda assim poderá declinar, pois, na saída, o MED solicitará um nome para o arquivo, sem o que a gravação não se realizará. Todo nome de arquivo digitado será inicialmente pesquisado no diretório do disco indicado. Se este arquivo for achado, ele será lido, caso contrário, o MED emitirá mensagem de "arquivo novo", abrindo-o para o usuário.

A tela de auxílio do MED

Logo na tela de abertura, o usuário é avisado a teclar F6 para pedir ajuda. Assim o fazendo, o arquivo MED.HLP contido no disco é exibido na tela, mostrando várias telas de auxílio, mas que, na realidade, são um contínuo desfilar de comandos do programa, tal qual o seu primo SCED.

Passar estas telas para uma impressora é uma iniciativa recomendável e pode ser conseguido através do comando COPY ou TYPE (com a saída da impressora ativada por <control> + P) ou por qualquer Processador de Textos capaz de ler o arquivo MED.HLP.

A primeira tela de auxílio é particularmente desorganizada e totalmente indiscriminada, levando o leitor a confundir os comandos lá dispostos. Fazendo uma faxina nesta tela, podemos encontrar o seguinte:

1 - Comandos de rolamento ("scroll")

a - scroll lateral: é conseguido por ^A (uma coluna para a esquerda) e ^S (uma coluna para a direita).

Nota: a régua de medição das colunas acompanha o rolamento, como seria de se esperar.

b - scroll vertical: é conseguido por ^U (uma linha para cima) e ^D (uma linha para baixo).

c - scroll de tela: é conseguido por ^W (uma página para cima) e ^Z (uma página para baixo).

2 - Deslocamento do cursor

a - por palavra: é conseguido por ^B (palavra anterior) e ^F (palavra posterior).

b - por linha: é conseguido por ^C (início da linha) e ^V (fim da linha).

c - por tela: é conseguida por ^T (fim da tela) e ^K ou home (início da tela).

3 - Manipulação de linhas

a - divisão: ^N (divide uma linha em duas, a partir da posição do cursor.

b - ligação: ^J (liga uma linha, a partir da posição do cursor, com a linha seguinte).

c - eliminação: ^Y (deleta uma linha).

d - inserção: ^O (insere uma linha em branco).

Observação: O "fim" de uma linha é encontrado após o sinal de <carriage return>, que não se vê na tela. Passeando com o cursor, usando o controle de seta para baixo, pode-se observar quando este sinal é encontrado pelo MED.

Outra indicação confusa é aquela que se dispõe na borda inferior da tela, quando se tecla <esc> para acionar um comando de forma indireta. Se o leitor prestar atenção, verificará que esta linha refere-se ao status do arquivo e do posicionamento do cursor:

[drive:nome] nº linha/nº total de linhas nº coluna Bytes
Exemplo:
[A:TESTE.TXT] 5/200 6 1500

A marcação de blocos de texto

Esta marcação é facilmente identificada pelo usuário pela disposição de dois sinais de "@" contínuos, que são imediatamente apagados quando o bloco é manipulado.Passeando-se com o cursor livremente pelo texto e teclando-se <return>

nas posições desejadas, ou ainda, indicando os números das linhas, é possível marcar início e fim de bloco

O MED trabalha com um arquivo temporário no disco, de nome TEMP.$$$, para usá-lo como "buffer" de um bloco marcado. Este arquivo é identificado e carregado na tela do Editor toda vez que se tecla ^P ("Paste Block") na posição do cursor.

Cuidados com a manipulação de arquivos

Todo cuidado é pouco com certas operações de manipulação de arquivo. O comando UPDATE, por exemplo, que ATUALIZA o arquivo no qual se está trabalhando, grava o arquivo atual em digitação em cima do antigo arquivo do disco SEM CRIAR UMA CÓPIA DE SEGURANÇA (backup).Por causa disso, o MED solicita a confirmação do usuário para realizar a operação.

O leitor pode ter certeza de que quando o MED solicita uma confirmação do comando, é porque o resultado da operação pode ser desastroso para o usuário. Por isso, manda o bom senso que você trabalhe com cópias de seus arquivos antes de se familiarizar com todos os comandos.

Uma vez terminada a digitação, a saída do MED é através do comando indireto QUIT, quando se tem a possibilidade de salvar ou não o arquivo.

Comandos de pesquisa e troca

Este é, talvez, um dos pontos altos do MED, gerenciado através de duas PFs: F3 para a pesquisa e F4 para a troca. Teclando-se <esc> para acionar comandos indiretos, o usuário pode optar pela pesquisa ou troca, de duas formas: uma, determinando a operação de maneira fracionada, segundo um número fixo de vezes ou um a um, e a outra, deixando que a operação se realize automaticamente, isto é, sem a interferência do usuário. Exemplo:

<esc>+?+F3 - pesquisa as strings uma por uma;
<esc>+5+F3 - pesquisa a string cinco vezes;
<esc>+*+F4 - troca todas as strings do texto.

A pesquisa e a troca podem ser também acionadas por comando direto, teclando-se F3 e F4, só que, assim, o usuário terá direito a apenas uma operação.

A operação de pesquisa e troca pode ser repetida teclando-se ^Q diretamente. O mais importante na maneira como o MED efetua a pesquisa e a troca é o respeito à string digitada pelo usuário, inclusive os espaços em branco, deixados até mesmo antes de qualquer outro caractere. Assim, as seguintes strings seriam, em princípio, diferentes para o Editor:

" PALAVRA" "PALAVRA" "PALAVRA "

Além disso, o MED é capaz de pesquisar e trocar substrings (cadeias de caracteres dentro de outras). Por causa disso, deve-se tomar cuidado com a discriminação do alvo ("target") da pesquisa e troca, quando a cadeia de caracteres indicada for pouco específica. Por exemplo: digitando-se apenas uma ou duas letras ("a", "de", etc.). Neste caso, todas as palavras que contiverem a letra "a" ou a sílaba "de" serão pesquisadas.

Assim, numa primeira pesquisa, é interessante comandar a operação com o auxílio do ponto de interrogação, como acima demonstrado. Fazendo isto, a operação de pesquisa se realizará mais lentamente e a de troca, com solicitação de permissão. O número de cadeias pesquisadas aparece na parte de baixo da tela do Editor.

Importante:

Quando há interesse em se saber o número de strings pesquisadas, deve-se prestar atenção ao número indicativo mencionado anteriormente. Note que, quando a operação é feita sem argumentos ("?", "*" ou número), esta indicação só aparecerá no fim da operação e de forma muito fugaz, às vezes não dando tempo de ser lida.

A indicação da string a ser pesquisada deve também ser feita com cuidado. O usuário não deve digitar aspas, a não ser que elas estejam incluídas na pesquisa. Espaços em branco serão considerados, devendo o usuário teclar <barra de espaço> nas posições desejadas. Quando o espaço em branco estiver no fim da string, o cursor servirá de referência para o número de espaços a serem considerados. Só depois o usuário poderá teclar <return> para executar o comando.

Por default, a pesquisa só poderá ser feita A PARTIR DA POSIÇÃO DO CURSOR ATÉ O FIM DO TEXTO. A última ocorrência pesquisada fará o cursor ali estacionar, caso contrário, o MED emitirá mensagem avisando não ter achado a string solicitada, na borda inferior da tela.

A troca de strings é feita com muita rapidez, fato este que pode atrapalhar o usuário inexperiente ou não familiarizado com este comando. Use a opção "?" como recomendado, para evitar destruir desnecessariamente seu texto. Teclando <control> + C ou <control>+<stop>, o MED sai desta ou de qualquer outra operação.

Salvando um disquete com o CHKDSK ("CHECK DISK")

Este utilitário procura clusters perdidos na FAT do disquete, aparentando trabalhar corretamente com disquetes de 3 1/2" e não com os de 5 1/4".

A maioria dos disquetes passíveis de serem recuperados pelo CHKDSK são aqueles onde ocorreram erros de gravação, sendo pouco provável que este programa tenha sucesso com erros de leitura ocasionados por destruição de setores do disco. Para este último caso, existem utilitários bem melhores e mais poderosos.

Os clusters perdidos são transformados em arquivos, por autorização do usuário, ou em espaço vazio reutilizável. Aí corre-se um risco de apagar arquivos, ou de criar um número enorme de arquivos inúteis, como se fosse um enorme jarro quebrado em caquinhos, quase impossíveis de serem colados.

Em função disso, deve-se usar o CHKDSK com a devida cautela. A sua melhor utilização é a simples constatação da integridade do disco, através do argumento "/m" na linha de comando, ou então sem utilizar nenhum argumento.

DISKCOPY: um bom copiador e comparador eficiente

Como foi mencionado anteriormente, o comando DISKCOPY duplica as funções de copiador (exatamente como seu análogo no MS-DOS) e como verificador da semelhança física entre dois disquetes (análogo, neste caso, ao DISKCOMP do PC).

O DISKCOPY copia integralmente o disco fonte através do processo de cópia física, descrito no Capítulo 1. Usando o argumento "/C" na linha de comando, as operações de cópia e verificação são feitas simultaneamente, o que é de todo interessante para aumentar a confiabilidade no processo.

Usando-se somente o argumento "/V", o programa efetua apenas a comparação. Note que esta função é complementar à primeira. Quando o usuário tentar comparar dois discos com os mesmos programas, mas formatados por controladoras de fabricantes diferentes, o CHKDSK emitirá mensagem constatando a ocorrência de

erro.

Como toda operação de cópia física, a deste programa é relativamente lenta. Assim, ela será mais rápida e confortável quando o usuário possuir dois drives.

Disquetes com trancas produzidas por alteração de formatação não são copiáveis pelo CHKDSK. Neste caso, ele acusará erro de leitura no disco fonte.

Arquivos de comandos do DOS ("BATCH")

Freqüentemente utilizada na informática, a denominação "batch" serve para designar os arquivos que se referem a um LOTE de comandos do DOS, cujo processamento se dará seqüencialmente.

O objetivo primário de um arquivo de comandos em lote é o de automatizar a execução de certas instruções sistematicamente repetidas pelo usuário. Na prática, isto significa transformar uma série de comandos em apenas um (!), referente ao nome do arquivo.

Quando se aciona um arquivo "batch", todos os comandos contidos nas linhas do programa são automaticamente colocados na linha do prompt do DOS, como se estivessem sendo digitados naquele momento.

Vai depender do interesse e da imaginação do usuário a aplicação deste tipo de arquivo, a qual, de uma maneira geral, consiste em: auto-execução de programas aplicativos, copiadores, atualizadores de arquivos de dados, etc.

O arquivos "batch" é também chamado de "arquivo de lote", uma tradução literal de "batch file", ou de arquivo de PROCESSAMENTO EM LOTE, já que todos os comandos são processados um a um, como descrito.

Como construir um arquivo "batch"

O arquivo "batch" nada mais é do que um arquivo de texto no qual os comandos são escritos em linhas separadas, da mesma forma como muitos programas-fonte de linguagens de alto nível. Nenhuma linha deve ser numerada, e sim escrita como se os comandos ali contidos estivessem sendo digitados do teclado.

O caminho mais fácil e prático para se digitar um arquivo "batch" é, portanto, através de um Processador de Textos, pelo simples fato de se poder EDITAR sem problemas qualquer parte do arquivo digitado.

Porém, nem todos os Editores de Texto são apropriados para montar um arquivo de comandos. Editores que formatam o arquivo de saída não podem ser usados, exceto aqueles que possuem uma opção de entrada onde o usuário pode definir o formato de gravação para "arquivo não-documento".

Opcionalmente, pode-se usar o comando COPY do próprio DOS, como indicado na sua descrição no Capítulo anterior, neste caso, ordenando ao DOS que mande "copiar" o teclado (CON).

Em ambos os casos, o nome de um arquivo de comandos deverá ter necessariamente a extensão .BAT, pois só assim o DOS o reconhecerá como tal. Se for necessário que o arquivo entre em execução imediatamente após a partida do computador, então, é obrigatório que o seu nome seja AUTOEXEC.BAT.

Digitando o "batch" com um Processador de Textos, é interessante escolher aqueles aplicativos que trabalham em ambiente DOS, pois assim ficará mais fácil recorrigir erros de digitação ou alterar o arquivo de acordo com as necessidades. O Editor MED da MICROSOFT, descrito neste Capítulo, é uma excelente opção.

Cada instrução deve ser digitada teclando-se <return> logo após, como se fosse um comando direto dado da forma habitual. O usuário não deve se esquecer que o "buffer" de teclado do DOS aceita somente até 128 caracteres, sendo, portanto, importante que o Editor de Textos possua algum recurso de medição de colunas.

Terminada a digitação dos comandos, não será necessário teclar ^Z, como acontecia no comando COPY, pois o sinal de "fim de arquivo" é sempre gravado pelo próprio Processador de Textos. Exemplo comparativo:

Comando COPY

```
A>COPY CON AUTOEXEC.BAT
DATE  <RETURN>
DIR/W  <RETURN>
COPY INÍCIO.TXT PRN <RETURN>
^ Z <RETURN>
```

Processador de Textos:

```
A>MED AUTOEXEC.BAT
DATE  <RETURN>
DIR/W  <RETURN>
COPY INÍCIO.TXT PRN <RETURN>
```

Parâmetros de substituição

Tanto o MS-DOS quanto o MSX-DOS permitem a introdução, no arquivo de comandos, de parâmetros substituíveis, com o objetivo de tornar a execução dos arquivos "batch" mais flexíveis.

O usuário tem direito a até dez substituições por arquivo, codificadas pelo sinal de "%" seguido do número da substituição, que vai de ZERO a 9 (%0, %1, %2, etc.). A numeração é importante para avisar ao DOS a seqüência de substituição dos parâmetros.

Os parâmetros de substituição são fictícios para o Sistema Operacional. Isto significa que, caso as substituições não sejam determinadas, o DOS será incapaz de executar corretamente os comandos por elas atingidos.

A determinação das substituições é feita pelo usuário diretamente na linha de comando, na ordem em que elas aparecem no arquivo e separadas por um espaço em branco. Por exemplo, se num arquivo de comandos COPY forem empregados os parâmetros %1 e %2 para designar os drives, o arquivo deverá ser acionado necessariamente da seguinte forma:

A>COPIADOR A: B:

Se, no caso acima, o programador incluir os dois pontos dentro do programa, eles não serão digitados na linha do comando, ficando assim:

A>COPIADOR A B

Digamos que cada linha do arquivo tenha sido digitada da seguinte maneira:

COPY %1:TTY.TXT %2

Se a indicação de substituição for A e B, respectivamente para os parâmetros %1 e %2, a linha do arquivo ficará assim, quando executada:

COPY A:TTY.TXT B:

O programador deve prever e listar, em cada arquivo "batch", os possíveis parâmetros de substituição, através de anotações incorporadas no início do arquivo (comando REM ou /).

Através do comando TYPE, o usuário poderá ver antecipadamente quais são estes parâmetros e substituí-los adequadamente.

Importante:

O parâmetro %0, quando é utilizado, será sempre substituído pelo nome da unidade de drive (se indicada) e pelo nome do arquivo "batch" (obrigatoriamente citado na linha de comando).

Que comandos podem ser utilizados num arquivo "batch" ?

Em princípio, qualquer comando INTERNO ou EXTERNO poderá ser utilizado no arquivo de comandos. Lembrando que quando um PROGRAMA é carregável pelo DOS, isto é, feito pela interpretação da extensão .COM, ele também poderá constar do "batch".

A inclusão de comandos EXTERNOS deve prever a sua existência no disquete alojado na unidade de drive citada nas linhas de comando do arquivo de comandos. As restrições a esta inclusão foram discutidas no início deste Capítulo.

IMPORTANTE

Dependendo da analogia dos comandos digitados no arquivo em lote, ele poderá ser rodado também em um micro tipo IBM-PC, sem nenhum problema. O programador deve verificar a sintaxe desses comandos, nos dois Sistemas Operacionais, que possa produzir igual efeito.

Se os comandos forem digitados com as suas respectivas "versões" em português, o arquivo só poderá ser rodado em ambiente SOLX-DOS, fato este que restringe muito o seu uso. RECOMENDAMOS VEEMENTEMENTE QUE VOCÊ NÃO ADOTE ESTE PROCEDIMENTO.

Algumas aplicações de arquivos de comandos

A seguir, iremos listar alguns arquivos de comandos que você poderá digitar ou aproveitar parcialmente em seus próprios "batches".

1 - Montando um copiador

Um recurso muito útil quando se tem vários programas em um só disquete que necessitam ser copiados eventualmente para um novo disco de trabalho, é montar copiadores específicos para os programas desejados.

Vamos supor que o usuário tenha feito uma cópia de segurança de dois aplicativos importantes: dBASE e WORDSTAR. Deste disquete serão feitas as cópias para os disquetes desejados. Como nem todos os arquivos do disco original são necessários no disco de trabalho, o copiador se encarregará de selecionar os arquivos que interessam a cada caso. A estrutura de cada copiador poderá ficar assim:

Para o dBASE

```
REM  Utilitário para
REM   copiar o dBASE
REM  do drive A para B
REM
REM  Insira disco fonte no
REM  drive A. Para cancelar,
REM  tecle <control>  + C.
REM  Para continuar...
PAUSE
VERIFY ON
COPY A:SOLXDOS.SIS B:
COPY A:DBASE.COM B:
COPY A:DBASEOVR.COM B:
COPY A:DBASEMSG.TXT B:
REM **********************
REM    FIM DA CÓPIA.
REM **********************
```

Para o WORDSTAR

```
/ Programa copiador
/ para o WORDSTAR.

/ Utiliza parâmetros
/ substituíveis %1 e %2.
```

```
/
/ Digite a indicação dos
/ drives na linha de
/ comando. Por exemplo:
/ A>COPIAWS A B
/
/ Tecle <control> + <stop>
/ para cancelar ou
PAUSE
VERIFY ON
COPY %1:MSXDOS.SYS %2:
COPY %1:COMMAND.COM %2:
COPY %1:WS*.* %2:
COPY %1:MAILMRGE.OVR %2:
COPY %1:SPELSTAR.OVR %2:
/ FIM DA CÓPIA.
```

Observe que as duas listagens foram redigidas de forma diferente, embora o objetivo dos programas seja absolutamente o mesmo. É desnecessário e diríamos até pouco recomendável digitar palavras com caracteres acentuados nas linhas contendo "REMARKS" ("anotações"). Pode acontecer de não ser possível enviar estes caracteres para o vídeo, provocando "buracos" no texto.

Note que o comando PAUSE é colocado estrategicamente após o aviso de cancelamento da execução do arquivo. Isto obrigará o DOS a emitir a mensagem usual de aguardo da digitação de alguma tecla.

Na primeira listagem, todos os arquivos a serem copiados foram citados sem referência ambígua. Este procedimento deve ser adotado sempre que no disquete apareçam arquivos que não possam ser copiados, com iniciais semelhantes aos arquivos de interesse. Já na segunda listagem, exemplificou-se com a referência ambígua, supondo que não existam outros arquivos com iniciais WS.

As referências ambíguas devem sempre ser evitadas num "batch" copiador, pois isto anulará a hipótese de se copiarem arquivos por engano, mesmo que o disco fonte tenha o seu diretório modificado.

2 - Rodando um programa automaticamente

Muitas vezes, há interesse em se entregar ao usuário um "pacote" fechado, contendo um determinado programa, que deverá ser rodado assim que o computador for ligado. Para atingir este objetivo, entrega-se um disquete com um aviso pedindo-se que o mesmo seja alojado no drive A antes de se dar partida no micro.

Em outras circunstâncias, o usuário tão freqüentemente se utiliza de um dado Aplicativo, que o acionamento do mesmo na partida do micro se torna monótono.

Em ambos os casos, a solução é montar um arquivo "batch" de execução automática. A única peculiaridade deste arquivo é o fato de que o seu nome, AUTOEXEC, é pesquisado e interpretado pelo Sistema Operacional na partida, como foi descrito no fim do Capítulo 1.

Este recurso é tão importante para o Sistema de discos que o BASIC, ele próprio, possui o seu AUTOEXEC: o AUTOEXEC.BAS, este último, absolutamente independente do DOS.

Vamos aproveitar uma carona no exemplo anterior e redigir um "batch" para cada um dos aplicativos:

dBASE

```
DATE
DBASE
```

WORDSTAR

```
DATE
WS
```

Se o micro possuísse relógio interno, poderia ser também incluído no arquivo o comando TIME.

Lembre-se de um detalhe importante: SÓ PODERÁ HAVER UM ARQUIVO AUTOEXEC.BAT POR DISCO. O resto, como se viu, é de uma simplicidade "Franciscana". E, embora não seja obrigatória a inclusão dos comandos DATE ou TIME, eles são necessários para registrar as operações de arquivamento no diretório do

disquete, com a data e, se possível, com a hora em que elas foram efetuadas.

3 - Executando um comando repetidamente

Quem viu "Tempos Modernos", de Charles Chaplin (o genial Carlitos), deve se lembrar da cena em que ele fica completamente atordoado de tanto apertar porcas dentro daquela fábrica, e acaba por sair na rua apertando tudo que se assemelha a uma porca.

Realmente, as tarefas massacrantes não só em nada contribuem ao trabalho e à saúde do indivíduo, como também o levam a cometer enganos. Talvez um dos benefícios que os computadores trouxeram ao nosso sistema de vida, mas que é pouco percebido pela maioria da população, é a execução de tarefas desgastantes de forma repetitiva e confiável, no lugar do ser humano.

O mesmo princípio pode ser aplicado ao programador, quando do uso prático do computador pessoal. O programa abaixo, por exemplo, visa repetir uma operação tantas vezes quanto desejado. No caso, escolhemos a formatação de disquetes no drive B, supondo que o programa vá ser utilizado quando o usuário acaba de comprar uma caixa nova de disquetes e precisará comandar a mesma operação dez vezes.

Note que o programa só pode ser rodado em ambiente SOLX-DOS, da maneira como foi escrito. Com uma pequena modificação, poderá servir para um micro IBM-PC.

Durante a execução do programa, haverá um inevitável troca-troca de disquetes. O disquete contendo o programa deverá ficar alojado no drive A, se o usuário possuir dois drives. Com apenas um drive, o programa também poderá rodar, mas, após o Sistema Operacional proceder à leitura do "batch", ele solicitará a troca de disquetes no momento da formatação e depois no retorno à execução da mesma operação, quando, então, o disquete contendo o programa deverá ser recolocado de volta. Sugerimos que este disquete seja protegido contra gravação, para evitar que ele seja formatado por engano na troca de discos.

Quando for digitá-lo, nomeie o programa como "DISKFORM.BAT". A listagem é a seguinte:

```
REM PROGRAMA FORMATADOR
REM    PARA DISQUETES
REM 40 TRILHAS DUPLA FACE
REM
```

```
REM  COLOQUE DISQUETE NO
REM  DRIVE B. PARE COM
REM  <control> + C OU...
PAUSE
FORMAT B:2
DISKFORM
```

Fazendo as devidas modificações, o programa poderá formatar com outras opções do Menu de Formatação (ver comando FORMAT, no Capítulo anterior). Além disso, o programa poderá rodar num IBM-PC, modificando a linha com a instrução FORMAT B:2 para FORMAT B: (aqui poderão ser incluídos argumentos do MS-DOS, se necessário). Você poderá formatar disquetes no PC e usá-los no MSX, mas não poderá utilizá-los para dar "boot" no DOS (leia sobre a Rotina de Partida no Capítulo 1 e no Apêndice 1).

Repare que, propositalmente, a última instrução deste arquivo "batch" é o nome do próprio programa, fazendo com que ele seja novamente lido e executado. Ao entrar na execução seguinte, a instrução PAUSE fará a necessária interrupção no programa, para permitir que o usuário troque o disquete no drive B ou, então, pare definitivamente as formatações. Caso você grave o programa com outro nome, não se esqueça de modificar também esta última linha.

CAPÍTULO 4

UTILITÁRIOS E FERRAMENTAS DO DOS

SEGUNDA PARTE:

O SISTEMA MULTIUTILITÁRIO "HELLO"

Introdução

Conforme foi descrito no Capítulo 1, o Sistema Operacional DOS, depois de carregado na memória do computador, assume o gerenciamento de todas as operações de entrada e saída de forma autônoma. Para isso, ele possui o seu próprio BIOS (Sistema Básico de Entrada e Saída), conjugado ao Sistema Operacional Básico (BDOS).

Quando um comando é dado pelo usuário, o DOS interpreta a ordem e, caso não tenha havido nenhum erro de sintaxe, ele ativará a rotina correspondente do BIOS, o qual, por seu turno, acionará, através de comandos em linguagem de máquina próprios, programas gravados na ROM da controladora de discos, para que a operação desejada se realize.

Nada impede, entretanto, que um programa carregado pelo DOS possa executar as mesmas tarefas que ele, e até mais algumas, sem se valer das rotinas do BDOS ou aproveitando-as apenas parcialmente. Neste caso, este programa teria que possuir rotinas capazes de substituir rotinas correspondentes no Sistema de Entrada e Saída (BIOS).

Como as operações do DOS são basicamente de natureza utilitária, um programa único, capaz de substituir e/ou acrescentar funções semelhantes, poderia ser classificado como um SISTEMA MULTIUTILITÁRIO. E é neste conceito que se encaixa o programa HELLO, descrito neste Capítulo.

Em princípio, quais as vantagens de se desenhar rotinas diferentes para executar as

mesmas tarefas que o DOS ?

É bem possível que as rotinas internas do DOS não sejam as que melhor exploram o potencial e a versatilidade do computador. Muitos itens podem ser melhorados, o que justifica, inclusive, o aparecimento de Versões subseqüentes às atuais.

O não aparecimento de novas Versões, bem como a necessidade de implementar recursos mais ecléticos para o usuário, levam aqueles programadores mais arrojados e experientes a construir um Sistema do tipo do HELLO.

Na Versão 1.0 atual do HELLO, é possível realizar a maior parte das tarefas do DOS com maior eficiência e versatilidade, ao lado de algumas operações exclusivas, como, por exemplo, a pesquisa e edição de arquivos ou de setores do disco com recurso de saída para a impressora, de modo a fornecer ao usuário um "retrato" da tela do computador, onde as informações mais importantes estão sendo exibidas.

Carregamento e entrada em operação do HELLO

O programa HELLO é fornecido em um disquete na forma de um pacote. Por isso, o seu acionamento será automático se este disquete for alojado no drive A, antes de se ligar o computador. Um arquivo AUTOEXEC.BAT se encarregará desta tarefa.

Não é obrigatório, entretanto, carregar o programa desta forma. Se o usuário acionar o computador com um disco de Sistema (preferencialmente o MSX-DOS, por ser este o DOS contido no disquete do HELLO), o carregamento poderá ser conseguido da forma habitual, digitando-se HELLO na linha de comando do DOS. Este último procedimento é mais interessante, para se conseguir registrar no diretório do disco a data das operações de gravação, porventura realizadas durante a sessão com o HELLO.

A cópia do HELLO fornecida ao usuário é personalizada e a partir dela não é permitida a cópia de segurança ("backup"), em virtude da existência de várias "trancas" no disquete original. Em função disto, é altamente recomendável proteger o disquete recebido contra gravação, para evitar que algum acidente estrague o programa original, já que a maior parte dele está gravada fora do alcance do diretório, em setores localizados apenas pelo "loader" (carregador) do HELLO (este, por sinal, é o único programa que se encontra listado no diretório com este nome).

Antes que o programa entre no ar, será feita uma verificação da formatação do disco e do rótulo onde o nome do usuário ou do proprietário está gravado. Após este reco-

nhecimento, o programa será vasculhado no disco e carregado.

Importante: caso o Sistema Operacional não esteja plenamente operante ou o disquete tenha sofrido danos, o HELLO não será carregado. Um defeito na fonte de alimentação dos drives ou um problema qualquer na interface controladora será suficiente para que isto ocorra.

Mas, se tudo correr bem, será exibida inicialmente uma tela de apresentação do programa, contendo a Versão e o nome do usuário. O HELLO inspeciona a existência de drives físicos e lógicos antes de liberar o programa para comandos do usuário. Aqueles que possuem mais de um drive no Sistema, poderão notar que os LEDs se acendem momentaneamente durante esta inspeção. Não há necessidade de que os drives extras contenham discos neste momento.

Teclando-se a barra de espaço, o programa sai da tela de apresentação e passa ao MENU PRINCIPAL. Todo o HELLO funciona na forma de MENUS incluídos em JANELAS, que se abrem à medida em que as operações vão sendo solicitadas. Uma barra em vídeo reverso estará necessariamente disposta na primeira opção do Menu Principal. Usando as setas de controle do cursor, para cima e para baixo, alcançam-se as outras opções. Uma vez atingido o "FIM" de qualquer janela, não haverá retorno automático da barra ao início das opções, sendo necessário teclar a seta para cima para fazê-la voltar. O HELLO trabalha em tela gráfica de 40 colunas. Havendo uma placa de 80 colunas, ela será automaticamente desativada.

Cada opção do Menu Principal forçará a abertura de uma segunda janela, onde constará um submenu contendo a opção desejada, a mudança de drive para as operações subseqüentes e o retorno ao Menu Principal, para escolha de outras opções.

Um detalhe importante: o HELLO ficará totalmente residente no micro e o disquete que o contiver poderá ser retirado do drive para o alojamento de outros disquetes a serem manuseados. Recomendamos que o disquete contendo o HELLO fique à mão, para uma eventual necessidade de recarregar o programa.

A seguir, descreveremos as diversas seções e recursos do HELLO, sob o título das opções do Menu Principal:

DIRETÓRIO

Não é por acaso que a opção de Diretório figura em primeiro lugar na lista do Menu Principal. Conforme comentamos no Capítulo 2 sobre os comandos do DOS, esta é hierarquicamente a instrução mais importante para o usuário, em vista da mesma permitir o reconhecimento do conteúdo do disquete manuseado.

Ao teclar a barra de <espaço> duas vezes para acessar o Diretório do disco, este é lido e exibido numa janela lateral, à esquerda do vídeo. A parte de cima da janela contém o "nome" do disquete que está sendo lido e a parte de baixo, a lista parcial de arquivos do Diretório.

Não há exibição de outros dados além daqueles acima mencionados. Se o usuário quiser obter outras informações a respeito dos arquivos, terá que recorrer à opção MAPA DO DISCO, mais abaixo.

O "nome" do disquete apresentado não é equivalente ao VOLUME do disco que se obtém pelo MS-DOS, de modo que se o usuário comandar DIR num micro IBM-PC compatível, para ler este disquete, o MS-DOS acusará que o disco não tem volume. A explicação para isso é muito simples: no MS-DOS, o volume (ou "label") do disco está gravado no diretório, com um Byte de atributo 28H, o que faz com que ele seja lido não como arquivo, mas como "volume do disco", através dos comandos VOL e DIR do MS-DOS. Como o MSX-DOS não tem previsão para o uso de atributos até a presente Versão, não é possível gravar ou ler o "volume" do disco. Uma saída é usar os Bytes iniciais da ROTINA DE PARTIDA na trilha zero, onde está armazenada a string com a marca do fabricante da interface ou outro dado semelhante. Apesar da trilha ZERO fazer parte dos setores privativos do Sistema Operacional, terreno este perigoso ao acesso do usuário, não há nenhum problema em alterar estes primeiros Bytes, o que pode ser feito via software, como é o caso do HELLO e de outros programas que se utilizam deste artifício. Obviamente, toda vez que se desejar ler o "nome" do disco, o usuário será obrigado a recorrer ao mesmo programa.

Nem sempre é possível exibir todos os nomes de arquivos do Diretório de uma só vez, devido à exigüidade do tamanho da janela. Para continuar a leitura da listagem, o usuário deve teclar a seta para a direita sucessivamente, até achar um arquivo desejado e, depois, a seta para a esquerda, para voltar às janelas anteriores. Quando as janelas contendo a lista de programas terminarem, os comandos de seta não mais terão ação.

Achando um arquivo que seja de interesse, o usuário deve teclar (com cuidado !) a barra de <espaço> novamente, para que seja aberta uma nova janela contendo um

segundo submenu, com as opções de manipulação possíveis do arquivo selecionado. Estas opções são as seguintes:

1 - Executar

O arquivo selecionado com <espaço> poderá ser rodado a partir do HELLO, mas somente aqueles que possuem extensão .COM. Usando esta opção com arquivos de outra natureza, o HELLO ignorará a ordem de execução e sairá automaticamente da opção.

2 - Renomear

Entrando nesta opção, o HELLO abre uma janela, onde o cursor estará posicionado para a digitação do novo nome do arquivo. Teclando-se <return> sem nenhum nome, é possível sair da opção. Não se deve digitar o nome do arquivo seguido de sua extensão e sim fazêlo por etapas: primeiro o nome, depois <return> e finalmente a extensão e <return> .Uma vez teclado<return> após a digitação do nome, o cursor não mais voltará ao início da digitação. Exemplo:

|ENSAIO . | <return>

|ENSAIO .DOC| <return>

3 - Apagar

Esta opção é de acionamento simples e direto, o que nos leva a recomendar a devida atenção ao leitor, pois se trata do apagamento de um arquivo. Sobre este tema já falamos anteriormente, quando apresentamos o comando DEL do MSX-DOS.

4 - Exibir

Da mesma forma como no comando TYPE do DOS, só é de interesse exibir arquivos de texto, pelos mesmos motivos previamente comentados na descrição daquele comando. Neste caso, em particular, o acionamento de <control> + S não interrompe a exibição do arquivo, como no TYPE. Para conseguir isso, o usuário deve teclar <stop> para interromper a exibição e <espaço> para continuá-la. Teclando-se <esc> pode-se SAIR da opção e voltar ao HELLO. Se o usuário teclar <control> + C ou <control> + <stop>, o HELLO será abortado e, para funcionar de novo, terá que ser recarregado !

5 - Copiar

Esta opção funciona de forma inteligente, usando a lógica do programa na seleção dos drives fonte e destino. Quando o usuário solicitou a listagem do Diretório, ele o fez com o disquete inserido no drive desejado. Como a função de cópia deriva desta listagem, o HELLO depreende que o disco fonte está alojado na unidade de disco selecionada. Sendo assim, o disco destino estará necessariamente no drive não selecionado. O HELLO entretanto, inspeciona previamente o sistema do usuário para verificar quantas e quais são as unidades de disco instaladas. Em função desta inspeção, será emitida uma mensagem de orientação do usuário para a operação de cópia. Digamos que se tenha solicitado o Diretório do disquete no drive A e seja selecionado um programa para a cópia. Se o usuário tiver um drive B no Sistema, o HELLO solicitará a colocação de disquete no drive B e procederá a cópia automaticamente para este drive. Havendo um só drive no sistema, o programa solicitará a troca de disquetes como fonte e destino, até que a cópia termine. Não há possibilidade de RENOMEAÇÃO do arquivo destino, o que torna a cópia de um arquivo no mesmo disquete uma operação impossível. Note que não é possível também a seleção de mais de um arquivo por cópia, sendo vedada a utilização de referências ambíguas. Havendo necessidade de se copiar mais de um arquivo simultaneamente, o usuário só poderá recorrer à cópia física integral do disco fonte, pela opção BACKUP do Menu Principal.

6 - Label

Como foi explicado anteriormente, o "Label" a que se refere esta opção corresponde a uma string fornecida pelo usuário, gravada no início da trilha zero do disquete, não podendo, assim, ser reconhecida pelos comandos do MS-DOS DIR e VOL. Como a gravação do "label" é normalmente oferecida pelo HELLO ao usuário após a formatação, esta opção visa a mesma gravação em discos não formatados pelo programa. Ao entrar nesta opção, o usuário sópoderá gravar o "Label", enquanto que a leitura é forçosamente executa da antes da listagem do Diretório.

EDITAR

Este é um dos mais importantes recursos do HELLO, o qual dificilmente será encontrado pelo usuário nos Sistemas Operacionais convencionais. Ele cumpre o papel de permitir a leitura e a modificação de Bytes nos arquivos, ou até em setores isolados do disco em exame.

O usuário poderá, através deste Editor, realizar as operações de depuração ("debug") de programas-objeto em linguagem de máquina, sem a necessidade de utilizar um programa especificamente para este fim.

A Edição é feita diretamente nos setores do disco, de tal forma que não há possibilidade de modificações no tamanho original do arquivo, como ocorre com outros tipos de editores que trazem totalmente o arquivo para a memória do micro e depois gravam o resultado da edição.

O Editor do HELLO trabalha também conjugado a um pesquisador de strings (acessado mais adiante, no Menu Principal), o que torna a sua operação bastante flexível. O usuário pode optar se deseja trabalhar via pesquisador (descrito posteriormente), ou se diretamente no Editor, na hipótese de que o setor do disco ou o arquivo em exame já sejam previamente conhecidos.

Ao usuário é permitido trabalhar com o disco, lendo e obtendo na tela e na impressora as informações que deseja, sem ser obrigado a gravar qualquer modificação efetuada.

Uma vez acionado, o Editor exibe uma tela de apresentação dos dados relativos às operações do momento: na parte superior, são exibidos o drive onde o disco examinado está alojado, o nome do arquivo e o setor que está sendo vistoriado. Todos os comandos à disposição do usuário estão dispostos em diversas teclas, que descreveremos a seguir.

<control> + R e <control> + A
Como cada setor do disco é exibido de forma paginada, as teclas <control> R e A permitem a troca de telas com as diferentes páginas dos setores. Acionando <control> + R, se retrocede uma página, enquanto que teclando-se <control > + A, passa-se à página seguinte. Por causa deste formato de exibição, o deslocamento do cursor se limitará à última linha da tela atual. A movimentação por <control> + R e A facilita a pesquisa de Bytes, quando a mesma é feita de forma manual. Dentro da mesma tela, pode-se mover o cursor através das SETAS LATERAIS do MSX.

<control> + X
Define um seletor que muda a posição do cursor, para facilitar a tarefa de edição. O seletor possui as duas opções de exibição dos Bytes do arquivo na tela: HEXADECIMAL (no lado esquerdo da tela) e ASCII (numa parte menor, localizada à direita). Ao mover o seletor, o cursor se localizará na posição correspondente da tela. O usuário só poderá editar Bytes de acordo com a opção do seletor. Cada Byte editado será automaticamente mudado nas respectivas posições da tela. Note-se que

valores em hexadecimal abaixo de 32 (&H20) não têm correspondentes em suas posições em ASCII. Isto se explica pelo fato de queos códigos ASCII desta região correspondem a códigos de controle e não a caracteres de texto. Embora o MSX duplique estas posições com caracteres gráficos, a impres são destes na tela não tem sentido para fins de edição. Por causa disso, em seus lugares, o HELLO coloca pontos na tela, disposição esta, aliás, comum a outros programas desta natureza. Na maioria das vezes, o usuário deverá entrar com valores em hexadecimal, para editar um setor ou arquivo, mas conta com enorme vantagem em poder editar usando caracteres ASCII naqueles locais onde aparecem trechos de TEXTO.

<tab> ou <control> + I

Ambas as formas de teclagem ativam a saída do HELLO para a impressora, dando ao usuário a oportunidade de obter um "HARDCOPY" da tela atual, isto é, uma transposição integral (Byte a Byte) da imagem que aparece no vídeo. A utilidade deste recurso é enorme para fins de documentação de algum arquivo de interesse. A impressora conectada ao computador deve ser capaz de trabalhar em padrão gráfico compatível com o EPSON (caso da maioria das máquinas conjugadas ao MSX). A imagem obtida na impressora é equivalente à tela do computador, mas invertida em relação aos Bytes do vídeo, fornecendo uma impressão da imagem "normal" na sua disposição no formulário.

<control> + G

Aciona comando de gravação do arquivo ou setor que está sendo editado, provocando a sua alteração permanente no disco, devendo, por isso, ser usado com muita cautela: trabalhe sempre com cópias de segurança !

<esc> ou <control> + [

Este comando permite sair das operações de edição SEM GRAVAR AS MODIFICAÇÕES FEITAS. É, portanto, importante que o usuário as use para evitar danificar um arquivo ou setor importante do disco, ou então, para garantir apenas a visualização dos trechos pretendidos.

Ao entrar na função de EDITAR, o HELLO abre um submenu, onde constam as opções de edição de um arquivo ou um setor do disco, mencionadas anteriormente. Na opção "Editar um arquivo", o programa exibe inicialmente o Diretório do disco onde a edição será feita. Teclando-se <espaço>, entra-se na listagem do Diretório e, através das setas de movimentação do cursor, escolhe-se o arquivo desejado, tal como na função DIRETÓRIO, previamente descrita. Teclando-se <espaço> nova-

mente, o HELLO procederá à leitura do arquivo no disco e posicionará a primeira tela do primeiro setor do mesmo para edição. Esta tela, coerentemente, mostrará na parte superior o drive onde a leitura foi feita, o nome do arquivo lido e o setor em exibição.

Usando-se a opção "Editar um setor", o HELLO abre uma janela, onde solicita do usuário a indicação do setor do disco a ser lido. Inicialmente, é exibido o setor zero. Se for este o setor desejado, tecla-se <espaço>, para permitir que o programa leia o setor e o exiba na tela. Se não for este, teclando-se a seta para a direita, promove-se um incremento do valor inicial (zero) de dez unidades, e assim sucessivamente, até que se ache o setor desejado. Através da seta para a esquerda, pode-se retroceder aos valores anteriores, em decrementos também de dez unidades. Após a seleção e a leitura do setor, a tela de edição conterá, na sua parte superior, a indicação do drive, à esquerda da tela e o número do setor, no canto direito.

EXAMINAR

O HELLO fornece um recurso semelhante ao comando CHKDSK do MS-DOS e do MSX-DOS TOOLS, que procedem à inspeção do disco, exceto que os métodos de correção são diferentes. Enquanto nestes últimos há uma procura de clusters perdidos, a função EXAMINAR do HELLO verifica a integridade física de cada setor, sendo bem mais útil na maioria dos casos onde o erro de entrada e saída (neste caso, erro de leitura) é ocasionado por danos à formatação do disquete, o que é muito comum, principalmente quando o funcionamento da controladora é irregular e pouco confiável.

Por outro lado, é bem possível que disquetes contendo erros de gravação, mas com a sua formatação intacta, sejam examinados e passem no teste, não havendo assim qualquer possibilidade de correção.

Ao EXAMINAR um disquete, o HELLO exibe uma janela indicando o número de cada setor examinado e, logo abaixo, o último setor onde foi detectado um erro. A cada erro encontrado, o HELLO oferece ao usuário a opção de mandar corrigi-lo. Se for este o caso, o programa procede à recomposição do setor e, se tudo correr bem, continua a examinar o disquete.

Ao EXAMINAR um setor, o HELLO verifica se os Bytes denominados "CRC", gravados pelo Sistema Operacional no início de cada setor, correspondem à soma dos Bytes contidos no setor examinado. Se esta correspondência não existir, o setor é identificado como errado e, se o usuário permitir, será perfeitamente recuperado.

Dependendo da severidade do erro encontrado, pode acontecer que o HELLO não consiga recuperar o setor na primeira tentativa e, então, irá abrir uma janela auxiliar, onde solicitará ao usuário a permissão de tentar novamente ou desistir da recuperação. Se houver desistência, o programa continuará a examinar os outros setores.

N.B. No caso de tentativas sistematicamente mal sucedidas, o usuário pode desistir de recuperar o setor e tentar reformatar o disquete. Se esta operação não for possível, é bastante provável que o disquete tenha sido danificado fisicamente de forma irremediável, pois tanto o módulo FORMATADOR do HELLO, bem como o EXAMINADOR, são bastante potentes. A opção FORMATAR (a seguir) é capaz de verificar a integridade física do disco e com isso recuperar discos com pequenos arranhões. Antes de reformatar o disco, deve-se salvar os arquivos que o Sistema ainda consegue ler.

FORMATAR

Esta opção duplica a mesma função disponível no DOS, porém, com a vantagem de ser bem mais rápida e confiável, além de permitir a colocação de um "Label" de identificação no disco recém-formatado.

Ao entrar na opção de FORMATAR, o HELLO exibe uma nova janela, contendo as opções de formatação e uma opção de saída, caso o usuário desista da operação, voltando ao menu anterior. As setas do cursor e a barra de <espaço> mais uma vez servem para facilitar o comando da operação.

Uma vez acionada a formatação, uma nova janela é aberta, mostrando ao usuário as trilhas sendo formatadas e o lado correspondente às mesmas. Para se ter uma noção da rapidez deste processo, ele é realizado cerca de 25 segundos a menos, num disco de 40 trilhas e face dupla, do que a mesma formatação através da respectiva rotina na interface controladora CD-X2, fornecida pela MICROSOL.

Depois de formatado o disco, uma outra janela se oferece para a digitação do nome do disco. Se o usuário teclar <return> , esta região do disco, próxima do início da trilha zero, ficará em branco (20H ou 32).

RESTAURAR

Este é um dos recursos importantes contidos no programa, mas deve ser usado com bastante cautela e critério. O objetivo dele é muito simples e extremamente bem idealizado: através da opção RESTAURAR, o HELLO cria uma cópia integral da trilha 0 na trilha 40, contendo os setores privados do disco, conforme explicado no Capítulo 1. Isto permite ao usuário, nos casos onde ocorra alteração do diretório por qualquer motivo, a sua recomposição pela leitura e subseqüente gravação do conteúdo da trilha 40 novamente na trilha 0, processo este classificado como "RESTAURAR", pelo HELLO.

A utilização da trilha 40 é lícita, devido ao fato de que a numeração, num disquete de 40 trilhas, vai de 0 a 39. No caso, a trilha 40 corresponderia, portanto, à 41ª trilha, não alocada para a gravação de arquivos.

Entrando na opção de RESTAURAR, o HELLO exibe uma janela com duas subopções: "SALVAR A TRILHA 0" e "RECUPERAR A TRILHA 0". Na primeira delas, acionada pela barra de <espaço>, será promovida a gravação da última versão do diretório do disco, enquanto que na segunda, também acionada por <espaço>, será feita a operação inversa, como anteriormente descrito.

O objetivo fundamental deste recurso do HELLO é o de preservar, por cópia, a integridade do Diretório do disco, o qual, conforme foi explicado no Capítulo 1, tem enorme importância para o Sistema Operacional, por conter informações vitais como, por exemplo, o cluster onde se inicia a alocação de cada arquivo constante da lista. Nos casos onde, por erro do Sistema, ocorre algum dano ao Diretório, ele pode ser RESTAURADO, permitindo assim ao usuário acessar novamente os arquivos lá constantes. Em outros casos, o usuário poderá alterar o Diretório por algum motivo intencional (por exemplo, mandando organizá-lo em ordem alfabética) e, depois, recuperar a antiga ordenação por esta opção.

Importante

Para não ter problemas futuros na opção RESTAURAR, o usuário deve prestar atenção à atualização do Diretório do disco. Toda vez que for acrescentado um novo arquivo ou deletado um arquivo anteriormente constante no Diretório, deve-se SALVAR a versão atualizada para a trilha 40. Se isto não for feito, quando for efetuada a próxima RECUPERAÇÃO, os dados lidos da trilha 40 não terão consistência com a disposição atual dos arquivos e a sua leitura será impossível, inutilizando o disco.

Queremos também chamar a atenção do leitor para o fato de que mexer no diretório de um disquete é uma operação bastante perigosa, pois uma vez danificado, sua recuperação é praticamente inviável. Por este motivo, o HELLO permite ao usuário SALVAR o conteúdo atual da trilha zero, a qual, como sabemos, inclui toda a base de dados importante para o Sistema de disco operar: FAT, ROTINA DE PARTIDA e o DIRETÓRIO. O usuário deve tornar esta a sua primeira opção e com isso GARANTIR a possibilidade de RECUPERAR o disco alterado. Para tornar intuitivo este procedimento, o HELLO coloca internacionalmente esta opção como default, ao entrar na janela com os comandos de restauração.

MAPA

Pelos meios convencionais disponíveis nos Sistemas Operacionais de Disco, como por exemplo o DOS, dificilmente podem-se obter mais informações sobre os arquivos, além daquelas impressas pelo comando DIR, visto no Capítulo 2. Freqüentemente, interessa ao usuário ter uma visão topográfica do disquete, de maneira a poder instantaneamente avaliar o real "status" do disco.

Conforme foi explicado no Capítulo introdutório deste livro, a ocupação de um disquete é sempre feita na seqüência dos setores disponíveis, de tal forma que se um disquete for recebendo arquivos sem nenhuma deleção, todos os setores serão preenchidos em ordem crescente. À medida, porém, que um ou mais arquivos sejam deletados, o Sistema tende a ocupar setores isolados para um mesmo arquivo, o que pode tornar o disco relativamente "desorganizado". Embora, na maioria das vezes, dificilmente o Sistema Operacional de Disco erre ao carregar um arquivo, ele perderá um certo tempo para localizar setores não seqüenciados e dispersos ao longo da superfície do disco. O ideal, portanto, seria uma ocupação do disco sem interrupções, para que a carga de programas se fizesse sempre mais rápida e mais confiável. O mapeamento do disco serve para dar uma idéia do grau de "desorganização" do disquete em exame e, com isso, tentar fazer um juízo se é aconselhável a "rearrumação" dos arquivos dispersos no disco.

Observação:

Existem programas utilitários na linha IBM-PC capazes de rearrumar um disquete muito reutilizado. Como não existe aparentemente programa semelhante na linha MSX, sugerimos no Capítulo 2 o uso do comando COPY *.*, passando a outro disquete os arquivos um a um, de forma organizada, pelo próprio DOS.

Através da opção MAPA, o usuário recebe de volta não só o mapeamento do disco, como também muitas informações a respeito da sua formatação. Estas informações estão dispostas na tela inicial da opção e sobre elas teceremos alguns comentários:

A formatação adotada pelo MSX prevê a estruturação e utilização de duas Tabelas de Alocação de Arquivos (FAT), motivo pelo qual a tela inicial enuncia o número de setores por FAT. Esta situação é válida tanto para os disquetes de face simples, quanto para os de dupla. Na realidade, a primeira dessas FAT é a principal e a segunda, uma cópia de segurança dela, a qual o Sistema Operacional recorre em casos de perda de dados na primeira. Em todos os disquetes de 5 1/4", o número de setores por FAT será sempre de dois.

Da mesma forma, o início do Diretório nos disquetes de 5 1/4" começará sempre no setor 5, independente da formatação, mas terminará no setor 11, se for de dupla face e no setor 8, se for de face simples.

Uma informação curiosa obtida nesta tela é aquela referente ao "número de setores ocultos", que normalmente indicará zero. Estes setores são aqueles não identificáveis pelo Sistema Operacional, podendo ser utilizados em algum tipo de "tranca" no disco.

Após a tela inicial com os dados acima mencionados, teclando-se a seta de movimentação do cursor para a direita, obtém-se o primeiro MAPA propriamente dito do disco. Note-se que ele dá uma idéia global da ocupação do disco, com todos os setores alocados para a gravação dos arquivos. Note-se também que, na parte superior da tela, constam informações sobre o disco e sobre a numeração das trilhas 0 a 39, com os setores representados por PONTOS.

Para ler o MAPA corretamente, de maneira a poder identificar os diversos setores, deve-se reparar que cada trilha é representada por números dispostos na abcissa (eixo dos X - na horizontal), enquanto que os pontos que representam os setores estão colocados em linha vertical (eixo dos Y - ordenada). Assim, por exemplo, o primeiro ponto, logo abaixo do zero, será correspondente ao setor zero desta trilha. Logo abaixo, estarão os setores (pontos) subseqüentes, até o número 18, perfazendo o total de nove setores por trilha, para os lados 0 e 1 de um disquete face dupla.

Observação:
Os setores são numerados seqüencialmente. Sua posição, entretanto, pode ser determinada de forma relativa, seja em relação ao disco como um todo ou, então, em relação a uma trilha e lado do disco em particular. Por exemplo: o setor zero da trilha 1 no lado 0 é também o setor número 18 do disco. Já o setor zero da mesma trilha 1, porém no lado 1, será o de número 26, e assim por diante.

Os setores ocupados por arquivos são mostrados na tela em VÍDEO REVERSO, de modo a permitir a sua fácil identificação no MAPA.

Na primeira tela do MAPA, a linha superior indica o drive sendo lido, a seguir, no meio da linha, o número de Bytes usados e, à direita, o número de Bytes livres.

CUIDADO: Se você fizer esta avaliação com o disquete do HELLO, não se iluda com os 12288 Bytes usados e os 350208 Bytes livres. Estes dados se baseiam no Diretório, mas o programa HELLO propriamente dito é muito mais extenso do que mostra o Diretório e ocupa setores isolados dentro do disquete, numa posição somente localizada pelo seu carregador.

Teclando-se a seta para a direita novamente, pode-se MAPEAR cada arquivo individualmente. O nome do arquivo em tela é indicado na linha superior e, à sua direita, o número de Bytes que ele possui.

IMPORTANTE teclando-se<espaço>, obtém-se um "hardcopy" das telas exibidas. Se a impressora não estiver ligada, o computador será travado.

Para voltar a qualquer tela do MAPA anterior, pode-se usar a seta para a esquerda. Chegando à última tela ou à inicial, as setas não mais terão ação. Se você voltar à tela inicial (aquela com as indicações da formatação), poderá sair da opção MAPA teclando <esc>.

ORDENAR

Muitas vezes, em Diretórios muito extensos, o usuário tem interesse em ORDENAR a listagem, com o objetivo de facilitar a leitura e procura de um determinado arquivo.

Este é o objetivo da opção ORDENAR, do HELLO. Ela dá chance ao usuário de conseguir esta nova organização do Diretório, a partir de dois critérios, ambos atingíveis pela janela que se abre quando se entra nesta opção:

a - pela EXTENSÃO do arquivo;
b - pelo NOME do arquivo.

O usuário é alertado mais uma vez a não tocar no Diretório sem antes fazer uma cópia da trilha zero, pela opção RESTAURAR, descrita anteriormente.

A opção default da janela de ORDENAÇÃO segue o critério do uso da extensão, pois esta classificação é mais utilizada pelos usuários. Se o leitor se acostumar a gravar arquivos digitando a extensão, lucrará mais em usar este tipo de ordenação, caso contrário, será aconselhável ordenar pelo nome do arquivo.

Um outro caso onde a ordenação pelo nome se torna interessante, é quando se tem um grande número de arquivos com nome muito parecido, digamos, por exemplo, um disquete com todos os capítulos de um manuscrito, distintos entre si apenas por números: CAP1-1.TXT, CAP1-2.TXT, CAP1-3.TXT e assim por diante.

É importante assinalar que a ORDENAÇÃO pura e simples do Diretório não implica na reorganização do disco, devendo esta ser feita pelos métodos já descritos anteriormente. Depois de ORDENAR o disco, pode-se recompor o Diretório pela opção RESTAURAR, também já descrita, DESDE QUE NENHUMA ALTERAÇÃO TENHA SIDO FEITA NO DISQUETE CUJO DIRETÓRIO FOI ORDENADO !

PESQUISAR

Esta função do HELLO trabalha em conjunto com a opção EDITAR, anteriormente descrita, de modo que se pode pesquisar uma string (cadeia de caracteres) em algum setor ou arquivo do disco e depois, opcionalmente, proceder à EDIÇÃO dos Bytes desejados.

Trata-se, assim, de um recurso de grande utilidade, por permitir executar as duas tarefas simultaneamente, coerente com as necessidades do usuário de PESQUISAR trechos de um programa, para poder EDITÁ-LO.

Ao entrar na opção de PESQUISAR, o HELLO abre uma janela contendo as duas opções básicas:

a - PESQUISA num arquivo;
b - PESQUISA no disco.

A PESQUISA é feita em função de uma string digitada pelo usuário, de até 19 caracteres, que é automaticamente guardada em um buffer de memória para posterior utilização. O usuário pode valer-se das CHAVES DE EDIÇÃO DO DOS, descritas no Capítulo 2, para EDITAR a cadeia de caracteres digitada, ou para chamar caracteres guardados na memória. O HELLO interpreta literalmente a string digitada, levando em consideração se os caracteres são digitados em maiúsculo ou minúsculo. Caracteres acentuados não aparecem na tela, mas são PESQUISADOS de forma correta.

Quando se faz opção pela PESQUISA num arquivo, o Diretório será lido e listado em sua posição habitual na tela e, através da barra de <espaço> e das setas do cursor, pode-se selecionar o arquivo desejado, como nas vezes anteriores. Neste caso, a PESQUISA começará a ser feita a partir do setor no qual se inicia o arquivo.

Quando a string é encontrada, o HELLO mostra em qual setor a PESQUISA teve sucesso e então oferece ao usuário a chance de EDITAR o setor. Note-se que nem sempre o primeiro setor exibido conterá a string, mas usando os comandos próprios do EDITOR, neste caso <control> + A, avançam-se as páginas até que a cadeia de caracteres seja encontrada.

Usando a opção de PESQUISA no disco, o HELLO abre uma nova janela, solicitando ao usuário que escolha o setor desejado, ao mesmo tempo em que exibe o setor zero como default de escolha. Se o usuário não sabe que setor do disco irá escolher, poderá teclar <espaço> com a escolha do setor em zero e o HELLO começará a PESQUISA a partir do início do disco. Como no caso anterior, assim que a string for encontrada, o setor onde ela se encontra será anunciado junto com a opção de EDIÇÃO.

Se a PESQUISA feita não tiver sucesso, será exibida uma mensagem pelo programa declarando não tê-la encontrado.

HARDWARE

Este talvez seja um dos grandes atrativos do HELLO: o teste do equipamento, acessado através da opção HARDWARE.

Nota:
As palavras HARDWARE e SOFTWARE não são privativas da informática, mas quando usadas neste sentido se referem, respectivamente, ao equipamento ("setup") e aos programas em geral.

Para usar esta função do HELLO, o usuário precisa estar alerta sobre certos detalhes do funcionamento dos drives, dos disquetes e do computador. Ao descrever cada um dos testes separadamente, procuraremos orientar o leitor sobre o que observar e o que avaliar.

Ao entrar na opção HARDWARE, uma segunda janela será mostrada com os subítens que podem ser escolhidos para avaliação, que detalhamos a seguir.

VELOCIDADE

Colocando um disquete dentro do drive e executando esta função, toda vez que o furo de índice (ver descrição no Capítulo 1) passa sobre a célula fotoelétrica que o detecta, o programa conta uma volta e, deste modo, calcula e exibe o número de rotações por minuto (R.P.M.) do drive, em uma escala semelhante à de um galvanômetro, com poucas divisões entre os valores nominais apresentados.

A ausência de um maior número de divisões na escala ou da sua substituição por uma escala digital (somente com números) torna a leitura da velocidade um pouco imprecisa, mas, na prática, isto não tem grande importância, em função de dois aspectos: o primeiro se refere à razoável tolerância no ajuste da velocidade do motor do drive, sendo aceitável uma faixa de valores relativamente generosa; o segundo, referente ao ajuste do drive propriamente dito, feito através de um componente elétrico de pouca precisão.

A velocidade ideal de um disk drive de 5 1/4" é de 300 rpm, mas é aceitável que o motor do drive gire entre 296,5 e 304,5. Velocidades abaixo ou acima desta faixa acarretam uma perda real de sincronismo de leitura e gravação, pelo fato de que o Sistema Operacional grava Bytes de sincronismo no início de cada trilha, os quais permitem que o drive ajuste a cadência do motor de acordo com a leitura do disco. Isto é feito através de um servo-mecanismo implementado dentro do drive, que acelera ou retarda a velocidade pré-ajustada de acordo com o disquete utilizado.

Pequenas oscilações na leitura da velocidade podem ocorrer em função principalmente da dificuldade de arraste do disquete, normalmente por causa das características físicas do invólucro do mesmo.

Neste sentido, é de todo interessante utilizar para esta avaliação somente disquetes de procedência confiável. Na dúvida, o usuário poderá experimentar vários disquetes de fabricantes diferentes até achar aquele que apresentar maior velocidade possível.

Sendo necessário, o ajuste da velocidade pode ser conseguido desmontando-se o drive e tendo acesso ao "trimpot" que faz este ajuste, geralmente colocado bem próximo ao motor. O "trimpot" é um componente semelhante a um potenciômetro, mas de tamanho miniatura e, no lugar do eixo, encontra-se uma fenda, onde se introduz uma chave adequada para girar o cursor do "trimpot".

Embora seja bastante simples, este ajuste só deve ser feito por pessoas habilitadas. Se você se aventurar a fazê-lo, eis algumas dicas: não force a fenda do "trimpot" para baixo, para evitar danificá-lo; nunca use uma chave de fenda inadequada (procure a de tamanho certo); gire o "trimpot" bem devagar, observando a tela do computador: quando a escala mostrar o valor de 300 rpm em cima da marca, considere o ajuste satisfatório. Caso haja mais de um drive no Sistema, compare as duas velocidades, mas não se obrigue a ajustá-los rigorosamente iguais.

Terminados os ajustes ou a leitura, tecle <esc> para sair da operação. Verifique todas as conexões elétricas do drive antes de remontá-lo.

ALINHAMENTO

O posicionamento das cabeças num drive de face dupla deve ser mantido rigorosamente alinhado, sob pena de não o se conseguir realizar nenhuma operação corretamente.

Um drive de boa procedência, quando vem de fábrica, é alinhado e lacrado nos parafusos onde este ajuste é feito, de tal forma que dificilmente haverá problemas posteriores de alinhamento. Na realidade, este deve ser o último item a ser conferido, pois até mesmo as fontes de alimentação têm enorme influência no desempenho dos drives.

Esta avaliação deve ser feita com o disquete original do HELLO, na hipótese de que ele tenha sido gravado num drive corretamente alinhado pela software house que o forneceu.

Pode-se também usar qualquer disquete, o qual se saiba antecipadamente ter sido formatado em um drive confiável. Na dúvida, o usuário poderá utilizar disquetes de várias procedências e repetir o teste no mesmo drive. Caso não seja acusado nenhum erro, é pouco provável que o alinhamento esteja incorreto, muito embora sem um disquete de referência que possa servir de padrão, nada se pode afirmar concretamente.

O teste é feito pelo HELLO através da leitura de diversos pontos do disco, uma cabeça de cada vez. A tela do programa mostra um diagrama animado para simular o deslocamento das cabeças e a indicação de qual das duas está realizando a leitura num dado momento. O observador não deve estranhar a representação neste gráfico de uma cabeça ligeiramente deslocada em relação à outra, pois isto representa a realidade física dos drives: se as duas cabeças fossem colocadas precisamente uma

em cima da outra, seus campos magnéticos se anulariam mutuamente.

Terminado o teste, o programa emite um relatório, onde constarão as avaliações da leitura de cada cabeça, ou lado do disquete examinado.

LEITURA E GRAVAÇÃO

Este teste é previsto para ser feito com o disquete original do HELLO. Para tanto, deve ser retirado o "tab" (selo) de proteção contra gravação (se existente), sob pena de receber de volta uma mensagem do programa avisando que o disquete está protegido, sem a possibilidade de sair desta operação.

Para fazer a avaliação de leitura e gravação, o programa grava e depois lê 512 Bytes na trilha 20, usando os setores 360 e 369. Nenhum disquete contendo programas nestes setores poderá ser utilizado, pois, conforme foi explicado no Capítulo 1, o Sistema Operacional é incapaz de identificar sozinho dados gravados em setores isolados, já que os mesmos não figuram no Diretório do disco e, assim, qualquer programa ou arquivo ali contido seria destruído.

Para diagnosticar a qualidade no desempenho do drive, o HELLO considera que uma margem de erro abaixo de 5% é indicativa de um drive em bom estado de funcionamento, enquanto que erros acima de 10% indicam drives com problemas de gravação e leitura, que necessitam de correção. Neste caso, o usuário deve fazer o teste de alinhamento inicialmente e, depois, a verificação do resto do sistema de disco instalado (fontes, cabos, interface).

Se o usuário não quiser usar o disquete contendo o HELLO para fazer este teste, sugerimos o emprego de um disquete recém-formatado, sem programas, o qual poderá depois ser reformatado, apenas como garantia de sua integridade, já que, na prática, isto não tem grande importância (o Sistema Operacional irá gravar programas nestes setores, de qualquer maneira).

LIMPAR

Quem já usou alguma vez na vida um gravador de fita magnética (rolo ou cassete), sabe que, com o tempo, as cabeças sujam, pelo contato prolongado com o material magnético (pó dos óxidos que o compõem) espalhado sobre a superfície da fita. Sabe também que se as cabeças não forem limpas e desmagnetizadas periodicamente, o

processo de gravação e leitura das fitas se deteriorará.

Nos drives de 5 1/4", as cabeças de leitura e gravação não tocam na superfície do disquete, de maneira que raramente elas se sujarão com o material magnético ali depositado. Eventualmente, entretanto, algum material ou pequeno detrito que se desprenda do disco poderá sujar as arestas das cabeças, quando, então, uma operação de limpeza poderá se fazer necessária, para evitar maiores males.

Quando esta operação é feita pelo HELLO, o que o programa faz é movimentar as cabeças do drive para frente e para trás por dez vezes. Este movimento, por si só, já é suficiente para "limpar" as arestas das cabeças, que tendem a sujar da forma como mencionamos.

Teoricamente, o usuário deveria usar um disquete especial de limpeza, descrito como não-abrasivo e, portanto, inofensivo às cabeças. Pessoalmente, achamos que tais produtos não são inócuos e duvidamos da sua eficiência na limpeza, já que, com as fitas magnéticas, seus equivalentes demonstraram claramente o que afirmamos. O melhor método empregado para os gravadores de fita podem também ser aqui empregados, SE realmente as cabeças do drive necessitarem de limpeza: o emprego de álcool isopropílico 100% puro, passado com delicadeza sobre a superfície das cabeças com o auxílio de um cotonete e, depois, bem secado com outro cotonete.

Outro procedimento é usar um disquete de boa qualidade no lugar do disquete de limpeza. A sujeira eventualmente largada pelas arestas das cabeças pode ser absorvida sem problemas pela camada de algodão interna, que fica entre o disquete e o invólucro, cujo objetivo é proteger o disquete contra arranhões causados pelo atrito do seu arraste.

SISTEMA

O Sistema Operacional depende das condições de funcionamento do computador, sendo um dos pontos críticos a memória utilizada para armazenamento de programas, alocamento do próprio Sistema e de suas áreas de trabalho.

O HELLO inspeciona o equipamento do usuário nos pontos de memória RAM necessárias ao funcionamento do DOS (64 KBytes) e, de tabela, inspeciona também a memória de vídeo (VRAM de 16 Kbytes), necessária, por sua vez, ao Terminal adotado pelo Padrão MSX, o VT-52.

Uma vez acionado o teste, ele não pode ser interrompido. A memória é vasculhada ponto a ponto, com a contagem das posições mostradas na tela: 65535 na RAM e 16383 na VRAM, no total. Se tudo correr bem, estes serão os números finais mostrados à guisa de relatório.

Se, ao contrário, for detectado um problema elétrico em algum CHIP de memória, o HELLO interromperá a inspeção e abrirá uma janela no canto superior direito da tela, exi-bindo a posição de memória onde o erro foi constatado.

Além da memória, o HELLO verifica a presença de drives físicos e lógicos do Sistema de discos. O interessante aqui é que o programa não pesquisa a área de variáveis do Sistema para verificar se existem dois drives, mas realmente procura a existência física das unidades, através do acionamento da leitura de cada uma das células foto elétricas de detecção de setor. O usuário nem precisa ter qualquer disquete dentro dos drives. Pode-se observar esta avaliação pelo acendimento do LED de cada drive físico. Ao final, são mostrados os números do Sistema instalado. Só aí o usuário poderá sair desta opção, teclando barra de <espaço>.

LIMPAR

Esta opção é, na realidade, um COMANDO do programa com a finalidade de LIMPAR a tela, com a volta ao Menu Principal. O objetivo deste comando é o de DESPOLUIR a tela das inúmeras janelas que a vão ocupando, sem desaparecer após operações subseqüentes.

O recurso de limpeza é útil, basicamente em função de dois aspectos: o primeiro, relativo à dificuldade de visualização, pelo usuário, do resultado das várias operações realizadas pelo programa, a despeito do fato de que sempre a última janela a aparecer fica em primeiro plano; o segundo, pela necessidade de "clarear" a área previamente à impressão, em papel (formulário), de alguma tela específica do programa.

Assim sendo, a opção-comando LIMPAR deve ser acionada todas as vezes em que se perceba que existe a necessidade de depurar a visualização da tela do programa, com isto diminuindo o grau de distração do usuário e permitindo-lhe concentrar-se no que está efetivamente fazendo.

VERSÃO

Durante o carregamento do HELLO, aparece sempre uma tela inicial contendo a VERSÃO do programa, com o nome do seu proprietário. Esta opção permite que essa tela apareça novamente no vídeo, recurso este perfeitamente dispensável, já que raramente o usuário se interessará em resgatar este tipo de informação.

BACKUP

Além da cópia de arquivos, descrita na opção DIRETÓRIO, o HELLO possui um excelente copiador para a confecção de BACKUPs do disco original. A cópia, neste caso, é física, sendo reconhecida a formatação do disco fonte a partir do drive definido pelo usuário. O programa pressupõe que o disco destino esteja alojado no drive não selecionado.

A partir do reconhecimento da formatação do disco fonte, o disco destino será formatado previamente à cópia, caso a formatação não seja coincidente. Assim, o programa assegura-se que a cópia de segurança seja rigorosamente igual ao original. Quando a formatação do disco destino é necessária, uma tela inicial é aberta, exibindo a operação de formatação.

Um mapa da cópia é mostrado na tela enquanto a mesma é realizada, permitindo assim que se veja a leitura e a gravação de cada um dos lados. O programa indica a trilha e o processo separadamente. As trilhas são indicadas no mapa inicialmente por pontos. Quando a leitura é feita, um "x" substitui o ponto e a indicação "LENDO" aparecerá numa barra, no canto inferior direito do vídeo. Quando a gravação da trilha está se processando, uma cerquilha ("#") aparecerá no lugar do "x", junto com a indicação "GRAVANDO". Os dois lados do disco são indicados no mapa, de tal forma que o usuário facilmente percebe quando a cópia pertence a um disquete de face simples (somente o lado 0 é lido) ou a um face dupla.

A opção BACKUP só pode ser feita com dois drives físicos no Sistema. Se o usuário tentar fazê-la com somente uma unidade, o programa avisa que não pode prosseguir a operação.

A cópia de segurança, uma vez deflagrada, não poderá ser interrompida. Ao seu final, teclando-se <espaço> volta-se ao Menu Principal.

RETORNAR

Terminando a sessão com o HELLO, pode-se voltar ao DOS pela opção RETORNAR. Quando o disquete do programa não estiver alojado no drive A no momento do retorno, é provável que apareça uma mensagem do Sistema Operacional pedindo a inserção de disquete com arquivo "batch". Neste caso, existem duas opções: uma, alojar de volta o disquete do HELLO e provocar o encerramento do "batch" pelo DOS; a outra, mais simples e direta, é teclar <control> + C (ou <control>+<stop>) e abortar a execução do "batch".

Em ambos os casos, o prompt do DOS reaparecerá. Como os caracteres das telas do HELLO são redefinidos, eles assim permanecerão, a não ser que o usuário resete o micro ou reative a placa de 80 colunas, quando a mesma estiver instalada no computador.

CAPÍTULO 5

UTILITÁRIOS E FERRAMENTAS DO DOS TERCEIRA PARTE:

O SISTEMA MODULAR "BKP DISCO", VERSÃO 1.0.

Introdução

O BKP DISCO é um programa utilitário cujo objetivo é permitir o acesso ao gerenciamento e controle da coleção de disquetes do usuário. Uma série de doze funções dispostas no lado esquerdo da tela principal do programa, por onde corre uma barra em vídeo reverso, controlada pelas setas do cursor para cima e para baixo, que seleciona a opção acionada por <return>.

As primeiras onze opções da tela principal correspondem a um arquivo separado, dedicado à execução da tarefa solicitada. Todos os arquivos estão dispostos no disco do BKP na forma de "overlays" e, portanto, identificados pela extensão .OVR no diretório. Cada vez que se tecla <return> , o arquivo correspondente à posição da barra é lido e as operações por ele realizadas aparecem na tela principal, numa janela à direita. Por este motivo, o usuário é aconselhado a manter o disco mestre do programa no drive A e, ao retirá-lo por qualquer motivo, deixá-lo próximo, a fim de realojá-lo sempre que necessário.

O nome "BKP", dado ao programa, refere-se ao mnemônico da palavra "BACKUP", originalmente empregada no utilitário copiador de fitas cassete, construído pelo mesmo autor. O BKP DISCO se assemelha ao BKP em cassete no fato de que ele também foi construído para ajudar a salvaguarda de programas do usuário.

As características do BKP DISCO, em termos de programação, em muito diferem do seu congênere anterior, levando vantagem plena do ambiente de disco para o qual ele foi pretendido.

Instalação e carregamento do BKP DISCO

Em vista das diferentes tabelas de caracteres utilizadas nos micros MSX nacionais, o BKP DISCO permite que o usuário possa instalar o programa de acordo com a tabela do seu micro. Dois arquivos extras são fornecidos no disco mestre para esta finalidade: HOTSET.ALF e EXPSET.ALF. O primeiro deles contém as tabelas existentes no HOTBIT 1.1 e 1.2 e no EXPERT 1.1, enquanto que o segundo instala a tabela originalmente adotada no EXPERT 1.0.

O procedimento da instalação é muito simples: o usuário seleciona qual dos arquivos acima deseja instalar; depois, é só copiar este arquivo com o nome de BKP.ALF, no mesmo disco onde está o programa. Exemplo:

```
A>COPY HOTSET.ALF BKP.ALF
```

Após a instalação, o disco mestre deverá ser protegido contra gravação, antes de se utilizar o programa. Com uma configuração normal de saída de vídeo no micro, para carregá-lo, digita-se na linha do prompt do DOS o nome BKP, seguido de<return>. Se houver uma placa de 80 colunas em um dos slots, o usuário deverá acionar um pequeno arquivo "batch" existente no disco mestre, de nome BKP80, o qual se encarregará de desativar esta interface de vídeo e, assim, permitir o correto funcionamento do programa.

Importante: na Versão do BKP DISCO por nós utilizada, só há uma previsão de desativação de placa de 80 colunas, no caso, a VMX-80. Nas Versões subseqüentes, uma rotina de carregamento verifica mais de um tipo de placa de 80 colunas, o que possivelmente deverá resolver uma incompatibilidade entre o programa e o equipamento do usuário. Não obstante, nada impede que a placa de 80 colunas venha a ser desativada por comando do teclado, ou que o programa seja adaptado pelo revendedor.

Uma vez carregado, após alguns segundos aparecerá a tela principal do programa, com a barra branca na primeira opção. A seguir, descreveremos sumária e objetivamente estas opções, para orientação do leitor.

Arquiva disco

A primeira função do BKP DISCO funciona em conjunto com outras funções do menu de comandos exibido na tela principal. Ao entender o que é e como o arquivamento de discos se processa, o usuário poderá escolher as outras opções de acordo com

a conveniência das operações posteriores.

O módulo ARQUIVA DISCO permite a criação de um arquivo, de nome BKP.ARQ, o qual pode ou não constar do disco mestre, cujo objetivo é guardar o conteúdo do diretório dos discos da coleção do usuário.

O BKP.ARQ é um arquivo seqüencial, que se organiza da seguinte maneira: a cada disco arquivado é dado um nome de até quinze caracteres. Este nome é muito importante, pois ele será a referência da qual o BKP se valerá para efetuar outras operações do programa. Após cada nome, são gravados todos os nomes constantes do diretório do disco. O usuário tem direito a arquivar até 40 KBytes de informação neste arquivo.

O disco cujo diretório será arquivado deverá ser alojado no drive B. Se este não existir, o BKP DISCO se encarregará de pedir as necessárias trocas de disco. Uma vez iniciado o processo (teclando-se <return> com a barra de vídeo reverso na opção), o programa arquivará continuadamente todos os discos do usuário, fazendo uma pausa entre cada arquivamento e permitindo que o usuário possa descontinuar a operação.

Antes de cada arquivamento, será solicitado o nome do disco. Esse nome não será gravado no disco arquivado, de forma que o usuário não terá nele qualquer indicação neste sentido, a não ser que este nome seja anotado no rótulo do disco. Como se podem usar até quinze caracteres, é de todo interessante usar nomes que identifiquem o mais próximo possível o CONTEÚDO do disco.

Todos os arquivamentos serão guardados inicialmente num buffer de memória. Quando ocorrer a primeira desistência de novo arquivamento, o BKP DISCO fará imediatamente a gravação. Neste momento, é importante que se tenha alojado no drive A, previamente, o disco onde o arquivo BKP.ARQ será gravado.

Se antes ou depois desta gravação (num arquivamento posterior), for dado a um disco o mesmo nome de um disco anterior, o conteúdo do primeiro disco será substituído pelo atual. Assim, é necessário prestar a devida atenção para não incorrer neste erro, a não ser que haja uma intenção clara de corrigir ou atualizar o diretório de um determinado disco previamente arquivado. Neste caso, devido à organização do arquivo, este novo conteúdo ficará sempre no seu final, o que, na prática, não tem a menor importância.

Observação: na entrada de nomes neste módulo, as chaves de edição do DOS (Capítulo 2) poderão ser utilizadas. Teclando-se <esc>, abortam-se as operações de arquivamento a qualquer momento.

Apaga arquivo

O objetivo deste módulo é permitir ao usuário retirar as informações referentes ao arquivamento de um determinado disco, ou até mesmo a eliminação de todo o arquivo BKP.ARQ.

Ao acionar este recurso, deve-se ter o disco mestre do BKP no drive A e, depois, neste mesmo drive, o disco contendo o arquivo BKP.ARQ.

Na parte inferior da tela do BKP DISCO, é perguntado ao usuário se ele deseja o apagamento total. Respondendo "N", é oferecida a oportunidade de digitação do nome do arquivo a ser eliminado. O arquivo BKP.ARQ é lido e armazenado na memória do micro, aumentando a velocidade de pesquisa do nome digitado pelo usuário. Caso este não exista, será emitida uma mensagem de aviso e o programa fará uma pausa. Teclando <esc> e depois <return> , pode-se reiniciar a operação.

No caso de ser solicitado o apagamento integral dos arquivos ou de haver apenas um disco arquivado, o arquivo BKP.ARQ será DELETADO do disco.

Observação: da mesma forma como no módulo anterior, as chaves de edição do DOS poderão ser utilizadas para facilitar a digitação de dados. Teclando-se <esc>, pode-se abandonar a operação a qualquer momento.

Procura arquivo

Este é o módulo de pesquisa e acionamento de alguns programas constantes nas listagens dos diretórios arquivados no BKP.ARQ. Sua função é, portanto, complementar à existência das duas primeiras funções do BKP DISCO.

Ao ser acionado, o PROCURA ARQUIVO solicitará o nome de um arquivo ao usuário. É essencial que seja alojado o disco contendo o arquivo BKP.ARQ no drive A, para que a pesquisa possa ser realizada sem problemas.

Referências ambíguas para o nome do arquivo a ser pesquisado, com o auxílio dos caracteres-chaves "*" e "?", são aceitas, permitindo assim a procura de mais de um

arquivo simultaneamente. Se a referência * ou *.* for digitada, todo o arquivo BKP.ARQ será listado.

Os arquivos serão listados à direita da tela principal. Quando mais de um arquivo for listado, o usuário ficará limitado à leitura dos títulos. O uso das teclas de controle do cursor, neste caso, são úteis:

SETA PARA BAIXO: avança a listagem atual, correspondente ao arquivamento de um disco, fazendo a tela rolar para cima.

SETA PARA CIMA: volta a listagem atual para seu início, fazendo a tela rolar em sentido oposto à tecla anterior.

SETA PARA A DIREITA: troca a listagem atual para a do próximo disco arquivado, caso o nome do arquivo pesquisado esteja também contido neste último.

SETA PARA A ESQUERDA: faz retornar a listagem anterior, após se ter avançado para o diretório do disco arquivado seguinte.

IMPORTANTE: teclando-se <insert> durante a exibição das listagens, pode-se obter uma cópia na impessora dos conteúdos dos discos arquivados, com seus respectivos nomes.

Assim, para listar todo o seu arquivo BKP. ARQ, deve-se fornecer a referência "*" ou "*.*" para o nome do arquivo, preparar a impressora com um formulário contínuo (de preferência) e depois teclar <insert>.

A pesquisa de arquivos poderá ser feita até que um determinado arquivo seja encontrado. Se houver interesse em rodar este arquivo, o usuário deverá teclar <esc> e depois <return>, indicando especificamente este nome para o BKP DISCO. O programa automaticamente perguntará se há interesse em carregar e rodar o arquivo. Note que apenas os arquivos .COM, binários e em BASIC poderão rodar.

Respondendo "S" a esta pergunta, o BKP DISCO pedirá a inserção do disco, cujo nome corresponde ao do arquivo pesquisado. Este disco só poderá ser inserido no drive B, o que, circunstancialmente, poderá acarretar problemas, principalmente naqueles programas que rodam com overlays.

Observação: caso o arquivo pesquisado não seja encontrado, será emitida uma mensagem de aviso e o programa fará uma pausa; teclando-se <esc> e depois <return>, pode-se reiniciar a operação de pesquisa. Teclando-se <esc> em qualquer

momento, interrompe-se o funcionamento deste módulo.

Mostra diretório

Esta seção funciona de forma simples e objetiva: ela lista o conteúdo do diretório de qualquer disco do usuário ou, ainda, examinando o arquivo BKP.ARQ, os nomes dos discos dados pelo usuário.

A exibição das listagens é sempre feita na tela principal, do seu lado direito. As setas de controle do cursor auxiliam o usuário na leitura do conteúdo completo dos arquivos. Teclando seta para baixo, a listagem rolará para o aparecimento dos arquivos seguintes. Numa situação inversa, teclando-se seta para cima, volta-se a listagem aos primeiros nomes de arquivos.

Neste módulo, não há qualquer possibilidade de se acionar ou ler os arquivos listados. Teclando-se <esc>, como habitualmente é feito, pode-se sair da listagem atual e até abandonar a operação. Ao teclar novamente <return>, pode-se ler o diretório de outro disco.

Ordena diretório

Este é um terreno perigoso, sobre o qual o leitor já foi anteriormente avisado para pisar com cautela, pois mexer no diretório de um disquete pode acabar custando caro ao usuário.

Por medida de segurança, o BKP DISCO realiza esta operação em duas etapas: na primeira, ele lê o diretório que se deseja ordenar e processa a ordenação na memória do computador; na segunda fase, o programa grava o diretório ordenado, quando, então, o usuário reza para nada sair errado, como por exemplo, faltar luz, emperrar o drive, etc.

Ao teclar <return> e acionar o módulo, um menu de opções de ordenação aparece à direita da tela. Três critérios podem ser adotados: um, tendo como referência o NOME dos arquivos; o segundo, levando-se em consideração a EXTENSÃO dos nomes dos arquivos, enquanto que o último critério soma o NOME e a EXTENSÃO.

Normalmente, só haverá interesse em se processar este tipo de ordenação usando o NOME como parâmetro, mas nem todo disquete é beneficiado com esta ou qualquer ordenação, pois muitas vezes interessa manter a SEQÜÊNCIA com que os arquivos

são gravados, ou então manter agrupados arquivos de um mesmo programa, que utilizam nomes completamente diferentes.

Depois de escolhida a opção, o BKP DISCO pede que você coloque o disco cujo diretório será ordenado. Teclando algo, a operação se iniciará. PRESTE ATENÇÃO ÀS MENSAGENS DO PROGRAMA, na parte inferior da tela. NÃO RETIRE O DISCO até que tudo termine, o que é indicado pela presença da pergunta sobre o desejo de nova ordenação. Não tecle "S" sem antes tirar o disco previamente ordenado, pois, caso contrário, recomeçará tudo de novo com o mesmo disco.

Se você possui um disco importante, não o use neste ou em qualquer ordenador de diretório sem fazer uma cópia de segurança do mesmo. Na dúvida, tecle <esc> e saia desta opção, para usar o módulo copiador do BKP DISCO, descrito mais adiante.

Examina arquivo

Este é um recurso dos mais úteis oferecidos pelo BKP DISCO, embora com uma performance limitada pelas condições do Sistema Operacional de Disco do MSX.

Na listagem do diretório, obtida pelo comando DIR, o usuário se acostuma a ver alguns dados a respeito dos arquivos, mas somente daqueles constantes dos 32 Bytes anteriormente descritos (ver Capítulo 1).

No entanto, freqüentemente interessa conhecer as características do arquivo que possam servir de guia para orientação do usuário. Estas informações, na sua quase totalidade, são apresentadas neste módulo.

Ao ser acionada a opção, aparecerá automaticamente uma listagem do diretório do disco examinado. Esta listagem pode ser vasculhada da forma habitual, com o auxílio das setas do cursor, como anteriormente feito. Uma vez achado o nome do arquivo desejado, tecla-se <return> para obter a permissão de digitar este nome.

Depois disto feito, o BKP DISCO examinará o arquivo indicado. As informações a respeito deste arquivo aparecerão à direita da tela, e serão limitadas ao que o programa poderá apurar sobre ele.

O máximo de informações que poderão ser colhidas será:

Tipo: refere-se à natureza do arquivo (binário, BASIC, etc.)
EI: refere-se ao endereço inicial onde o programa se alojará na memória do micro.

EF: o mesmo que EI, só que para o endereço final.
EE: quando cabível, mostra o endereço de execução.

O usuário irá notar que nem sempre será possível exibir todas estas informações, ao mesmo tempo em que algumas delas se repetem para um mesmo TIPO de arquivo. Isto ocorre porque nem todos os arquivos do disco possuem um "header" (cabeçalho) com indicações padrões do TIPO do arquivo, fazendo com que o BKP se guie pelas EXTENSÕES dos nomes dos arquivos. Quando esta estiver ausente, será impossível identificar o arquivo, prejudicando as outras informações que dependem desta identificação.

Por exemplo: um arquivo de comandos do DOS terá necessariamente a extensão .COM e o seu endereço inicial só poderá ser &H0100.

Outros arquivos, como aqueles carregados e rodados pelo BLOAD do BASIC DE DISCO, apresentarão no seu cabeçalho as indicações dos endereços inicial, final e de execução, pois os mesmos servem de guia ao Sistema Operacional para carregá-los e rodá-los automaticamente. Sendo assim, fica fácil para o BKP DISCO obtê-las e listá-las para o usuário.

Restaura arquivo

Este módulo compreende uma série de ferramentas destinadas à recuperação de arquivos indevidamente deletados pelo usuário.

A recuperação não é feita de forma automática, a partir da indicação do nome do arquivo, como acontece com outros utilitários. Em contrapartida, uma vez conhecendo bem o arquivo a ser resgatado, as chances de que isto aconteça são maiores, relativamente, com os meios fornecidos pelo BKP DISCO.

O usuário que se aventurar pela selva de dados gravados no disquete deve conhecer bem o terreno onde pisa, sob pena de cair em armadilhas de difícil saída ou de se embrenhar em trilhas que não levam a canto nenhum. O leitor deve saber bem os conceitos amplamente descritos no Capítulo 1, sobre a organização de trabalho e alocação de setores no disco, para poder se nortear durante a manipulação do disquete contendo o arquivo deletado. Se você não se recorda destes conceitos ou não leu o primeiro Capítulo, é convidado a fazê-lo antes de prosseguir adiante.

É importante que o usuário faça uma cópia física do disquete a ser trabalhado e se familiarize com os comandos deste módulo, antes de qualquer outra iniciativa.

O grau de dificuldade na recuperação dos arquivos deletados envolve os seguintes aspectos: o arquivo mais fácil de ser restaurado é aquele cujo tamanho em Bytes é igual ou menor que um cluster. Neste caso, a pesquisa do arquivo no disco será facilitada pela sua menção no diretório do disco, que indica o cluster inicial (neste caso, o único). Em qualquer outra circunstância, haverá necessidade de se pesquisar os clusters alocados ao arquivo a ser recuperado e ordenar a sua regravação pelo BKP DISCO. O usuário deve observar no diretório o cluster inicial dos arquivos deletados (reconhecidos na listagem pelo Byte E5H, antes do nome) e vasculhar o disco para ver e marcar todos os setores alocados ao arquivo. A dificuldade aí será diretamente proporcional ao grau de "desorganização" do disco.

A "desorganização" de um disco tende a ser maior, conforme mais arquivos forem sendo deletados e novos arquivos gravados a seguir. Como foi dito no Capítulo 1, o Sistema Operacional vai ocupando seqüencialmente os setores do disco. Quando um arquivo é deletado, os setores alocados a ele se tornam vagos para a gravação de outros arquivos. O próximo arquivo gravado ocupará necessariamente este espaço, mesmo que existam outros arquivos dispostos em setores posteriores a este. Quando mais de um arquivo é deletado, todos os setores a eles alocados se tornam disponíveis. O próximo arquivo que for gravado neste disco, ocupará os setores anteriormente alocados ao primeiro arquivo deletado encontrado no diretório, o que, em muitos casos, pode corresponder ao arquivo deletado mais antigo.

Se o usuário for gravando arquivos sem deletar nenhum, o Sistema Operacional ocupará os clusters em perfeita seqüência. Se, num dado momento, um arquivo for deletado no meio desta seqüência, o mapa do disco mostrará um verdadeiro "buraco", correspondendo aos setores zerados na FAT. A recuperação, nestas circunstâncias, fica mais fácil pelo fato de se poder delimitar o início e o fim do arquivo deletado, pela simples inspeção do mapa do disco, obtido por este ou qualquer software utilitário que cumpra a mesma função.

Se, ao contrário, forem feitas gravações e deleções sucessivas, é bastante provável que os arquivos fiquem partilhados em setores isolados, tornando a recuperação uma tarefa difícil e, às vezes, praticamente impossível. Nestes casos, é necessário uma boa dose de conhecimento do arquivo que se deseja retaurar, o que depende em grande parte da natureza do arquivo. Um arquivo de texto, por exemplo, é mais fácil de ser identificado pela grande maioria dos usuários, devido à sua legibilidade. Já os arquivos gravados em linguagem de máquina, provavelmente só serão reconhecidos por aqueles que conhecem não só as características da linguagem, mas também os programas em questão, o que nem sempre é fácil, mesmo para usuários muito experimentados. Para resgatar um arquivo deletado em discos desta natureza, é preciso muita paciência e, por que não dizer, um pouquinho de sorte.

O processo básico utilizado no BKP DISCO para a restauração de arquivos, consiste na seleção de clusters do disco, feita pelo usuário, e o seu armazenamento num "buffer" de memória criado pelo programa. Na etapa final, estes clusters são gravados no disco na forma de um novo arquivo, com um nome dado pelo usuário.

Em princípio, existem duas formas de abordagem na seleção dos clusters: a primeira, levando em consideração todos os setores disponíveis para gravação, segundo o mapa do disco; a segunda, pela eliminação de todos os setores atualmente ocupados pelos arquivos presentes no diretório (não deletados).

Em ambos os casos, o usuário, partindo do cluster inicial, deve estabelecer a seqüência de clusters a ser armazenada no buffer. Para isto, ele conta com vários recursos acionados pelas teclas de função, sendo necessário conhecer bem cada um destes comandos antes de trabalhar na procura do arquivo, motivo pelo qual eles são descritos a seguir:

F1 - Apaga o buffer de memória contendo os clusters marcados pelo usuário. A quantidade de setores e o tamanho do arquivo em Bytes são automaticamente zerados.

F2 - Marca o cluster examinado, em exibição na tela do BKP DISCO, para inserção no buffer. A marcação pode ser feita em qualquer ordem, pois o programa se encarrega de colocar todos os clusters na seqüência numérica crescente, na qual eles aparecem no disco. Quando um cluster é marcado uma segunda vez, a sua indicação é apagada no buffer, sendo esta a forma de editar o seu conteúdo.

F3 - Seleciona clusters para inserção no buffer, a partir de uma indicação do usuário: optando pela entrada (E), o cluster inicial do arquivo deletado indicado é pesquisado no diretório e colocado no buffer; optando por tudo (T), todos os clusters em disponibilidade dentro da FAT são inseridos no buffer, a partir do cluster inicial do arquivo deletado (quando este for indicado) ou a partir do primeiro cluster livre encontrado.

F4 - Permite a saída do modo de exame de clusters marcados, acionado por F7. <esc> faz o mesmo efeito.

F5 - Grava, na forma de um arquivo, os clusters marcados por F2 e guardados no buffer, de acordo com o nome indicado pelo usuário. Este é o comando recuperador por excelência, pois, se tudo correu bem, traz de volta o arquivo deletado, tornando-o um arquivo presente no diretório.

F6 - Apaga clusters da forma definida pelo usuário: pelo mapa (opção M), todos os clusters pertencentes a arquivos presentes, constantes na FAT, serão eliminados; pelo buffer (opção B), todos os clusters anteriormente marcados serão eliminados.

Observação: a eliminação, no caso, não retira do diretório os nomes anteriormente constantes, dando a falsa impressão de que os arquivos continuam gravados no disco.

IMPORTANTE: Esta função não poderá, sob hipótese alguma, ser excutada no disquete original. Usando a opção M, é possível separar os setores alocados anteriormente pelos arquivos não deletados, que agora ficarão "apagados"(na tela, aparecerá o caractere &H40 ("@") no lugar dos Bytes dos arquivos). Isto facilita a pesquisa dos clusters a serem marcados, a partir do cluster inicial lido por F3 no diretório. Usando a opção B, é preciso, antes, verificar se o conteúdo do buffer não tem mais serventia, pois, se estes clusters pretencerem a um arquivo a ser recuperado, os dados referentes a ele estarão irremediavelmente perdidos. A opção B será útil apenas nos casos onde houver interesse em eliminar os clusters de um arquivo já restaurado, para fazer a pesquisa dos clusters de outro arquivo deletado.

F7 - Lista a seqüência de clusters marcados pelo usuário, ou pelas funções do programa (F3).

F8 - Monta um diretório com nomes fictícios de arquivos, seguindo uma codificação própria: NOMEnnnn.RES, onde nnnn se refere a uma numeração colocada pelo programa. Somente deverá ser usado em discos cujo diretório houver sido danificado. Em discos com o diretório em ordem, trará problemas ao mesmo. A nomeação dos arquivos se baseará pela FAT, pela data atual e pelo respectivo tamanho.

F9 - Exibe o primeiro cluster marcado ou o último que foi examinado.

F10 - Modifica o tamanho do programa restaurado. O usuário deve informar o novo tamanho do arquivo.

Ao acionar a opção RESTAURA ARQUIVO, aparecerá inicialmente uma tela com o primeiro cluster do disco examinado. O usuário deverá estar familiarizado com as notações dos números em hexadecimal, pois todas as informações referentes a clusters, trilhas, etc., são dadas desta maneira, na Versão atual do programa.

As seguintes teclas auxiliam o usuário no exame dos clusters dispostos nestas telas:

SETA PARA A DIREITA: avança um setor do cluster atual; se este setor for o último, avança para o cluster seguinte.

SETA PARA A ESQUERDA: recua um setor do cluster atual; se este setor for o inicial, recua para o cluster anterior.

SETA PARA BAIXO: rola a tela atual para a leitura das linhas seguintes.

SETA PARA CIMA: rola a tela atual para a leitura das linhas anteriores.

SHIFT + SETA PARA BAIXO: avança até o último cluster do disco.

SHIFT + SETA PARA CIMA: retorna até o primeiro cluster do disco.

SHIFT + SETA PARA DIREITA: avança uma trilha.

SHIFT + SETA PARA A ESQUERDA: recua uma trilha.

<select> - carrega um setor escolhido pelo usuário: na primeira opção, pode-se entrar com o cluster (&H0 a &H167) e com o lado (&H1 ou &H2); dentro da segunda opção (alcançada pressionado-se <tab>, pode-se escolher a trilha (&H0 a &H27) e o cluster (&H0 a &H8).

<home> - retorna à tela o último setor escolhido através da tecla .

<cls> - abandona o exame do disco atual e processa a leitura de um novo disco; o mesmo efeito pode ser conseguido teclando-se <esc> e depois <return> para reiniciar as operações de restauração de arquivos.

<tab> - define a exibição do setor, para contagem a partir do início do disco ou a partir da trilha atual.

<esc> - sai de qualquer operação, podendo voltar à tela principal.

Exemplos de restauração de um arquivo

Vamos supor que num determinado disco tenham sido gravados quatro arquivos e deletado apenas um deles. Ao examinar o mapa deste disco (conseguido no BKP na função EDITA DISCO), percebe-se que nada ainda foi gravado posteriormente.

Teclando F3, podemos colocar no buffer todos os clusters zerados na FAT, a partir do nome por nós indicado, menos a letra inicial (se o arquivo se chamava DESENHO.TXT", a digitação será "ESENHO.TXT". Isto fica intuitivo na tela do BKP DISCO, pois o programa coloca um sinal de "_" após a solicitação do nome. Para armazenar todos os clusters, tecla-se a opção T(udo).

Se o arquivo deletado estiver em seqüência no meio dos outros arquivos, o que se pode observar pelo mapa do disco (função EDITA DISCO), será suficiente teclar F5, para que se proceda à leitura dos setores armazenados no buffer, e depois se grave o "novo" arquivo neste ou em outro disco (o BKP fará para isso a necessária pausa).

Se mais de um arquivo tiver sido deletado neste mesmo disco, será interessante pedir, na função F3, o armazenamento da entrada no buffer e, posteriormente, marcar manualmente os clusters, até que o arquivo esteja completo. Depois de gravá-lo com F5, usa-se F6 para eliminar estes clusters do disco. Limpa-se o buffer (F1) e pede-se novamente a entrada do cluster inicial por F3 ou, caso este seja o último arquivo deletado, o armazenamento de todos os clusters disponíveis, caindo assim no caso anterior.

Discos muito mexidos, mas com apenas um arquivo deletado, beneficiam-se do apagamento por F6 dos clusters anteriormente usados pelos arquivos presentes (apagamento pelo (M)apa). Depois, deve-se pesquisar e marcar os clusters com F2, gravando-os depois com F5.

Edita disco

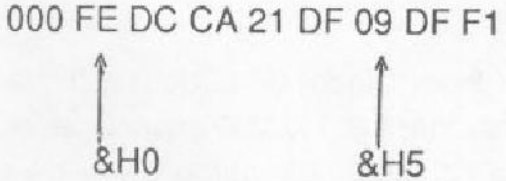
O módulo EDITA DISCO do BKP funciona como um processador de dados no disco, permitindo modificar os Bytes de cada setor, copiar setores em locais especificados pelo usuário, apagar clusters, confeccionar o mapa do disco, examinar arquivos e muitas outras funções.

Por todos estes motivos, o EDITA DISCO se constitui numa valiosa ferramenta de trabalho, principalmente para os usuários mais experimentados, para quem certos recursos são notoriamente dirigidos.

O módulo funciona em dois modos distintos de operação: o primeiro, permitindo examinar, com vários recursos, o conteúdo de um disco; o segundo, deixando o usuário editar o conteúdo do disco. Neste segundo modo, é sempre conveniente lembrar o leitor que se utilizar do programa que use somente uma cópia de segurança do disco, de modo a evitar danos irremediáveis ao mesmo.

Ambos os modos de operação estão ativos simultaneamente, devendo o usuário tomar uma certa cautela ao acionar os diversos comandos do módulo, de modo a evitar que qualquer ação se faça indevidamente. Um conhecimento prévio deste comando é, portanto, imprescindível. Independente disto, o próprio BKP DISCO interroga o usuário nos comandos modificadores mais "perigosos", pedindo a necessária confirmação. Se você, leitor, é do tipo afoito, recomendamos que nunca confirme a execução de um comando, quando você não sabe o resultado advindo dele. Lembre-se que estas perguntas não são colocadas pelos programadores por acaso !

Ao entrar em execução, o módulo EDITA DISCO examina o disco desejado, colocando inicialmente na tela do computador um display do primeiro setor do primeiro cluster lido no disco. Todas as informações, nesta Versão, estão escritas em hexadecimal, fato este que complica um pouco a leitura para aqueles que não estão familiarizados com esta notação. Na descrição dos Bytes do setor, contidas no centro da tela, os três primeiros números à esquerda correspondem a um endereço contado a partir do início do setor, indo de &H0 a &H1FF, completando assim os 512 Bytes de um setor. Os Bytes correspondentes a cada endereço estão dispostos nas linhas a seguir, de modo que o endereço inicial da linha seguinte deverá ser o endereço imediatamente posterior ao do último Byte da linha anterior. Se o usuário quiser saber o Byte de um endereço específico, deverá se guiar pelo endereço da linha mais próximo, e contar os Bytes da linha até chegar ao valor desejado. Por exemplo: desejando saber o conteúdo de &H5, procura-se a linha onde está o endereço 000 (&H0) e contam-se 6 Bytes, chegando-se ao valor procurado:



Como se vê, o primeiro Byte da linha corresponde ao endereço descrito na linha. No canto direito da tela, acontece a mesma coisa, só que cada Byte aparece traduzido em caracteres da Tabela ASCII, cuja finalidade, para o usuário, é facilitar a iden-

tificação de trechos de texto dentro dos arquivos, principalmente daqueles que não são arquivos-texto por natureza. Um detalhe importante, neste particular, é a exibição restrita de valores da Tabela ASCII, fazendo com que apareçam pontos (".") no lugar dos Bytes que não têm tradução dentro do programa.

As seguintes teclas podem ser usadas para a identificação e edição dos diversos setores:

SETA PARA BAIXO - rola a tela para cima, de modo a permitir a exibição das linhas seguintes do setor.

SETA PARA CIMA - rola a tela em sentido contrário ao anterior.

SETA PARA DIREITA - avança para o setor seguinte.

SETA PARA ESQUERDA - retrocede ao setor anterior.

SHIFT + SETA PARA BAIXO - pula para o último cluster.

SHIFT + SETA PARA CIMA - volta ao primeiro cluster.

SHIFT + SETA PARA DIREITA - avança para a trilha seguinte.

SHIFT + SETA PARA ESQUERDA - volta para a trilha anterior.

<control> + SETA PARA BAIXO - avança um setor, dentro do exame de um arquivo.

<control> + SETA PARA CIMA - retorna um setor, nas mesmas condições.

<control> + SETA PARA DIREITA - avança para o arquivo seguinte.

<control> + SETA PARA ESQUERDA - volta para o arquivo anterior.

<control> + <shift> + SETA PARA BAIXO: avança para o final do arquivo.

<control> + <shitf> + SETA PARA CIMA: retorna ao início do arquivo.

<control> + <shift> + SETA PARA DIREITA: avança para o último arquivo do disco.

<control> + <shift> + SETA PARA ESQUERDA: retorna para o primeiro arquivo do disco.

<home> - lê novamente no disco o setor em exame, objetivando reeditá-lo, anulando o trabalho anterior.

<select> - carrega um setor escolhido pelo usuário,através de duas opções: 1ª)- entrando com o cluster (&H0 a &H167) e com o lado (&H1/&H2); 2ª)- entrando com a trilha (&H0 a &H27) e com o cluster (&H0 a &H8). As opções são acessadas pela tecla <tab>.

<cls> - carrega novamente o último setor escolhido por <select>.

<return> - entra no modo de modificação: apertando esta tecla pela primeira vez, permite a edição dos Bytes em hexadecimal (valores entre &H0 e &HF); teclando uma segunda vez, deixa que a edição seja feita pela digitação de caracteres ASCII diretamente, sendo útil para a modificação de trechos de texto em programas em linguagem de máquina.

<insert> - grava o setor atual no disco, concretizando qualquer modificação efetuada. O comando solicita confirmação. NA DÚVIDA, NÃO CONFIRME !

<tab> - alterna o display do cluster: no primeiro modo, conta os clusters a partir do início do disco; no segundo, conta os clusters a partir do início da trilha atual. O primeiro modo é utilizado na função acionada por F7 e o segundo por F2.

<esc> - sai do modo de edição, podendo voltar à função anterior ou à tela principal do programa.

Para tornar mais versátil e eclética a edição pretendida, algumas facilidades foram incluídas neste módulo, acionadas pela teclas de função:

F1 - Copia um setor do disco escolhido pelo usuário para outro local do disco. A operação é feita em duas etapas: na primeira, os Bytes do setor são armazenados em um buffer de memória e na segunda, o programa pede autorização para copiar o setor oupara abortar a operação, através das opções (C)opia ou (A)borta.

Observação: Para efetuar esta operação com segurança, proceda da seguinte forma: tecle F1 ao encontrar o setor que você deseja copiar; desloque-se para o

outro setor que será substituído pelo primeiro e tecle novamente F1. Ao aparecerem as opções (C)opia ou (A)borta, tecle "C".

F2 - Mostra na tela o MAPA do disco. Inicialmente, o BKP DISCO lê as informações necessárias para exibir o mapa, após o que, apresenta uma visão gráfica geral do disco, com todos os clusters preenchidos. As coordenadas do mapa são as seguintes: as linhas representam as trilhas, enquanto que as colunas mostram os clusters de cada trilha. Ambas as numerações estão em notação hexadecimal, nesta Versão. Ao apertar qualquer tecla, o programa passa para o mapa dos arquivos, na ordem em que eles aparecem no diretório do disco. As seguintes teclas são redefinidas para facilitar esta inspeção:

SETA PARA DIREITA: avança para o mapa do próximo arquivo.

SETA PARA ESQUERDA: retorna para o mapa do arquivo anterior.

SETA PARA BAIXO: avança para o último arquivo do disco.

SETA PARA CIMA: retorna para o primeiro arquivo do disco.

<insert> - passa para o mapa do arquivo seguinte, sem apagar os clusters do mapa do arquivo anterior.

<delete> - apaga os clusters relativos ao mapa do arquivo anterior.

<select> - permite que o usuário escolha o mapa de um arquivo, através da indicação do seu nome.

A função de exibição do mapa é muito útil para a avaliação da ocupação do disco, principalmente antes de entrar no módulo de restauração de arquivos, comentado anteriormente. Os clusters ocupados são exibidos na forma de pequenos quadrados preenchidos, enquanto que os clusters vazios são representados por estas mesmas figuras sem preenchimento, como seria de se esperar. O contraste entre eles fornece uma visão pictórica do disco.

F3 - Pesquisa uma seqüência de Bytes (string ou cadeia de caracteres). Opera de modo simples: uma vez indicada a string, o BKP efetua a procura a partir do setor atual.

F4 - Sai do modo de exame de arquivos (acionado por F9), para voltar ao último setor exibido na tela do programa, que corresponde ao último setor carregado.

F5 - renomeia o arquivo atual em exame (F9).

F6 - Quando este módulo é acionado, o disco é automaticamente lido. Esta função permite que um novo disco seja lido sem necessidade de sair e voltar novamente ao módulo. Para acioná-la, coloque o novo disco no drive corrente e tecle F6. Tenha certeza de que nenhuma operação com o disco anterior ficou pendente, pois os dados relativos a ele desaparecerão.

F7 - Fornece informações sobre o arquivo atual em exame, lidas do diretório do disco e da FAT. Estando nesta função, o usuário pode continuar examinando outros arquivos, usando para isto as teclas de função:

SETA PARA DIREITA - avança para ler as informações do próximo arquivo do diretório.

SETA PARA ESQUERDA - retorna para ler as informações do arquivo anterior ao atual.

SETA PARA BAIXO - avança para o último arquivo do disco.

SETA PARA CIMA - retorna para ler dados sobre o primeiro arquivo do disco.

<select> - lê dados sobre um arquivo cujo nome é indicado pelo usuário.

NOTA: Se os dados sobre os arquivos forem excessivos (caso de arquivos que alocam grande quantidade de clusters), o BKP DISCO preencherá a tela e aguardará que seja teclado <return>, para continuar a listagem. Para voltar a ler o início deve-se teclar <home>.

F8 - Pesquisa no disco um arquivo indicado pelo usuário. Quando este é encontrado, o seu primeiro setor é exibido automaticamente.

F9 - Entra no modo de exame de arquivos. Acionada pela primeira vez, F9 começa a ler o disco a partir do primeiro arquivo do diretório. Acionada outras vezes, F9 lerá sempre o último arquivo examinado pela função.

F10- Leva para a impressora o conteúdo do setor atual. A impressora utilizada deve ser de 80 colunas ou ajustada para tal. O usuário pode optar pela impressão de todos os dados (opção (D)isplay) ou somente pelos caracteres de texto (ASCII), encontrados no setor (opção (T)exto). Na primeira opção,ao contrário da tela de vídeo do BKP, que exibe somente 8 Bytes de cada vez, são impressos 16 Bytes por linha,

estando os endereços de início de linha modificados de acordo.

A leitura do mapa do disco e dos arquivos, junto com a opção de listagem dos clusters, tornam este módulo útil na avaliação da situação dos discos, quando se parte para a recuperação de arquivos deletados.

Copia Disco

Este é um dos módulos mais interessantes do BKP DISCO, pois permite a cópia física do disco fonte, de forma parcial ou total, a critério do usuário. Com este recurso, é possível salvar partes específicas de disquetes danificados por qualquer motivo.

Se, nos discos com problemas, o usuário se der o trabalho de verificar a situação de cada arquivo, usando os recursos do módulo anterior, poderá anotar os locais exatos onde o disco está com defeito (ver também "Mostra Situação") e, depois, copiar para outro disquete somente os trechos que o BKP é capaz de ler.

O módulo COPIA DISCO possui duas opções básicas para o usuário:

a - CÓPIA NORMAL

Nesta opção, pode-se determinar ao BKP DISCO quais os setores a serem copiados, através da seleção dos clusters e do lado dos discos. Ao entrar na opção, o programa solicita ao usuário que informe se o disco é face Simples ou Dupla. O objetivo desta pergunta é determinar, através de sua resposta, os valores default dos clusters inicial e final, bem como dos lados do disco, que auxiliarão o usuário, em dúvida na escolha da cópia INTEGRAL do disco fonte, bastando, para isso, teclar <return> para responder as perguntas relativas a estes itens. Na realidade, um disco de 40 trilhas, seja ele face dupla ou simples, apresentará necessariamente valores extremos de &H0 até &H167, perfazendo um total de &H168 (360) clusters. Isto se dá porque, num disquete face dupla, o número de Bytes por cluster é o dobro do utilizado num disquete de face simples (Capítulo 1):

Face dupla - 368640/1024 = 360 (&H168) clusters
Face simples - 184320/512 = 360 (&H168) clusters.

Teclando "D" ou "S", o que o programa faz é ajustar o default para valores extremos de &H2 (face Dupla) ou &H1 (F (Face Simples), para os lados do disco fonte. Valores abaixo ou acima dos extremos acima mencionados, serão considerados ILEGAIS e

a sua entrada será recusada pelo programa. O usuário pode determinar se a cópia será feita com um ou dois drives, bastando para isso, no primeiro caso, definir o mesmo drive para fonte e destino, respectivamente. O BKP se encarregará de solicitar a troca de disquetes para a leitura e gravação, alternadamente.

b - CÓPIA ESPECIAL

No caso desta opção, o processo será feito de forma automática e transparente para o usuário, o que é conseguido através da inspeção que o programa faz do disco fonte, para avaliar neste quais os clusters efetivamente ocupados por programas. Somente os setores referentes a estes clusters serão copiados. O resultado prático disto se reflete numa poupança de tempo significativa, naqueles casos onde o disco fonte possui um número pequeno de arquivos gravados ou com diminuta ocupação da área disponível para gravação. Considerando-se que as cópias físicas são relativamente lentas (leia a NOTA abaixo), e com grande número de trocas de discos (caso de um único drive), a opção de CÓPIA ESPECIAL é bastante vantajosa, pela objetividade com que a operação é feita.

NOTA: A velocidade real de qualquer copiador é determinada pelo algoritmo de cópia projetado pelo programador. No caso das cópias físicas, o processo tende a ser mais lento, em função do número de setores lidos de cada vez, mas em princípio, nada se pode afirmar a este respeito, pois outros fatores devem ser levados em consideração, como por exemplo, a quantidade de memória disponível alocada para a operação. Sendo assim, faz-se necessário cronometrar a velocidade de cópia, quando se deseja comparar a performance de vários utilitários copiadores.

O item velocidade, embora um aspecto importente em qualquer circunstância, no caso do BKP DISCO, é compensado pela versatilidade do seu copiador e pela integração deste com os outros módulos.

COPIA ARQUIVO

Este módulo é complementar ao anterior, com o objetivo de trabalhar basicamente em cima do arquivo BKP.ARQ criado através do programa e utilizá-lo como referência nas operações de cópia ou, opcionalmente, em cima do diretório de um disco qualquer. No primeiro caso, o BKP DISCO pesquisará em qual disco está contido **o** ou **os** arquivos que se desejam copiar. Este recurso é muito importante, nos casos onde a coleção de discos é grande e o usuário dedica discos especificamente a

categorias de programas, onde o número destes impede que se possa fazer uma relação, na etiqueta do disco, de cada um dos nomes. Normalmente, dá-se um título geral para o disco (por exemplo: "APLICATIVOS E UTILITÁRIOS Nº 2") e, quando se tem uma impressora, imprime-se a listagem do disco, como sugerimos no Capítulo 1. Quando esta listagem não está disponível, o meio mais confiável é trabalhar com o arquivo BKP.ARQ e deixar que o BKP DISCO pesquise em qual dos discos da coleção podemos achar o arquivo desejado, ao invés de ler todos os diretórios de disco para chegar ao mesmo objetivo.

Quando o módulo COPIA ARQUIVO é ativado, aparecem estas duas opções, as quais, quando acionadas, atuam interativamente com o usuário, através de menus e mensagens adequadas a cada caso.

Cópia por um diretório

Entrando nesta opção, pode-se selecionar um disquete qualquer para a cópia ou apagar arquivos deste mesmo disquete. Em qualquer dessas duas operações, o BKP DISCO lê e exibe o diretório à direita da tela, para seleção dos arquivos a serem copiados ou deletados. Dois sinais gráficos (e) se posicionam no nome do primeiro arquivo da lista. Ao mover as setas do cursor, este sinais se deslocam para outros nomes. Quando um arquivo é marcado, os sinais gráficos e permanecem junto ao nome do arquivo. As marcações são feitas com o auxílio de algumas teclas redefinidas pelo programa para ajudar nesta tarefa:

SETA PARA BAIXO: desce os sinais e para os nomes seguintes da listagem.

SETA PARA CIMA: sobe os sinais e para os nomes anteriores da listagem.

<select> - marca um arquivo para a operação de cópia ou deleção; se o arquivo já tiver sido marcado antes, a tecla <select> apaga a marcação.

<insert> - marca todos os arquivos de uma só vez, poupando, assim, um trabalho manual desnecessário.

<delete> - apaga todas as marcações de uma só vez, sendo útil para se refazer o trabalho de marcação.

<home> - inverte a marcação de marcados para desmarcados e vice-versa, sendo útil quando, depois de feita uma marcação trabalhosa, se tecla acidentalmente <delete>.

<return> - encerra a fase de marcações e inicia a gravação ou o apagamento, selecionados previamente pelo usuário.

<esc> - aborta qualquer operação deste módulo

As deleções de arquivo são feitas imediatamente após se teclar <return> , sem pedir confirmação, partindo da hipótese de que o usuário tinha certeza do apagamento dos arquivos, quando se dispôs a gastar tempo marcando os arquivos. Se o usuário desistir da operação, portanto, deverá teclar <esc>, ao invés de <return> . Para arrependimentos tardios, deve-se partir imediatamente para o módulo RESTAURA ARQUIVO e tentar resgatar o engano cometido.

No processo de cópia propriamente dito, o BKP pede a indicação dos drives fonte e destino, caso haja mais de um drive disponível no sistema. Teclando <return>, os defaults são A para a fonte e B para o destino. Após aviso de inserção de discos, o programa inicia a cópia lendo os arquivos marcados, listando-os na tela à direita e exibindo mensagens de orientação ao usuário, na parte inferior da tela. Após o primeiro arquivo gravado, o BKP exibe o valor, em KBytes, da memória livre disponível no disco destino. A cada arquivo lido, o programa faz um cálculo, onde avalia, com base no tamanho do arquivo e na memória livre, se ainda existe espaço suficiente para a gravação. Em caso negativo, exibe uma mensagem ao usuário para a troca do disco destino por outro, no qual o mesmo cálculo será feito. Com este controle, a possibilidade de ocorrer erro de escrita por falta de espaço no disco é impossível. Além disso, o usuário se livra da tarefa de ficar, ele mesmo, contando Bytes livres para saber se consegue copiar os arquivos que deseja, o que nem sempre dá certo.

Cópia por um arquivo

Entrando nesta opção, o BKP DISCO se guiará pelo arquivo BKP.ARQ, para orientar o usuário na inserção dos discos fonte no drive A. Inicialmente, o programa lê o BKP.ARQ para, através de sua listagem, permitir ao usuário a marcação dos arquivos para cópia. O nome dos discos, nesta etapa, não será exibido, nem o usuário deve se preocupar com isso, e sim com a marcação propriamente dita dos arquivos que desejar.

Além das setas do cursor anteriormente usadas, o BKP provê mais duas, com o objetivo de facilitar a pesquisa de um nome de arquivo, em listagens muito longas:

SETA PARA DIREITA: avança para o conteúdo do próximo disco arquivado no BKP.ARQ.

SETA PARA ESQUERDA: recua para o conteúdo do disco anterior arquivado no BKP.ARQ.

Depois da marcação feita, e teclando-se <return> para iniciar a operação, o BKP DISCO emitirá mensagens para o usuário, orientando-o a colocar os discos contendo os arquivos para a cópia. O procedimento posterior é idêntico ao descrito na opção de cópia anteriormente descrita.

Mostra situação

Este módulo é também uma ferramenta bastante útil para a avaliação do STATUS de um disco, bem como da sua integridade operacional. O módulo é previsto para auxiliar os comandos EDITA DISCO e RESTAURA ARQUIVO, mas o seu emprego é desejável em qualquer circunstância, onde o usuário precise obter informações precisas sobre a incidência dos chamados erros de e/s (I/O ERROR, ou erro de entrada e saída).

Existem três opções básicas de inspeção: a do diretório do disco, a do arquivo BKP.ARQ e a da integridade global do próprio disco:

Situação do diretório

Nesta opção, o BKP DISCO examina o diretório do disco solicitado e dá ao usuário as seguintes informações:

FACE (dupla ou simples)

ARQUIVOS (lista o número de arquivos PRESENTES)
APAGADOS (lista o número de arquivos DELETADOS)

BYTES USADOS
BYTES LIVRES
> ambos os valores em Kbytes

Situação do arquivo (BKP.ARQ)

O disco contendo este arquivo deverá ser alojado no drive A. Depois da sua leitura, o BKP informa:
ARQUIVO: BKP.ARQ

ARQUIVOS (lista o total de nomes de arquivos constante)
DISCOS (lista o total de discos que foram arquivados)

BYTES USADOS (lista o número de KBytes que já foram ocupados no BKP.ARQ, sendo o máximo permitido de 40 KBytes; valores menores do que 1 KByte serão listados como ZERO)

BYTES LIVRES (lista o número de KBytes disponível; se o número de BYTES LIVRES for menor do que 1 KByte, o valor indicado será o máximo: 40 KBytes).

Situação do disco

O programa examina e exibe simultaneamente, à direita do vídeo, todas as trilhas, clusters e lados do disco desejado. O usuário pode optar, pressionando a tecla tab, se quer ver a contagem dos clusters a partir do início do disco ou a partir do início de cada trilha. O exame demora um certo tempo, mas se o usuário quiser interrompê-lo, basta teclar <esc>.

A cada setor defeituoso encontrado, o programa emitirá uma mensagem de alerta ao usuário e incrementará o contador de "Número de Setores Defeituosos" (NSD), colocado na parte de baixo do vídeo.

Ao final, será emitido um relatório sobre o exame executado, onde constam o número de setores bons e os defeituosos, além do número de setores examinados.

Esta última opção serve para ajudar o usuário na identificação e, possivelmente, na correção, via módulo EDITA DISCO, de setores que provocam problemas de leitura no disco. Devido à forma ortodoxa com que isto é feito, só mesmo os usuários mais experimentados terão chance de algum sucesso.

Usando o EDITA DISCO, tecle <select> para tentar ler os setores identificados como defeituosos. Se houver sucesso nas leituras, tente regravá-los teclando <insert> e torça para tudo dar certo.

Volta ao DOS

Este comando aciona a saída do BKP DISCO e o retorno ao Sistema Operacional. Como a placa de 80 colunas é desativada pelo BKP, a volta ao DOS, nesta Versão, será necessariamente na screen 0 normal do computador.

CAPÍTULO 6

O BASIC DE DISCO DO MSX

Introdução

No Capítulo 1, fizemos uma descrição do BASIC DE DISCO e da maneira como ele é acessado pelo Interpretador BASIC, residente na ROM do MSX. Diferentes Versões do BASIC DE DISCO podem ser implementadas num computador MSX "standard", bastando que se troque a interface controladora de discos, ligada num dos slots externos do micro, ou a sua EPROM.

O objetivo principal do BASIC DE DISCO é complementar ao BASIC residente na ROM do computador, de forma que se possa conseguir a construção e o gerenciamento de arquivos em disco. Praticamente qualquer programa originalmente escrito em BASIC poderá ser adaptado para operar em disco, com exceção daqueles que ocupam uma área livre de memória agora restrita às operações dos drives, tal como foi amplamente discutido no início deste livro.

Todos os comandos de arquivamento do BASIC convencional da máquina MSX, com exceção de CSAVE e CLOAD, que são exclusivos da fita cassete, referem-se à saída para o disco, em condições default. Na prática, isto significa que quando estes comandos são escritos num programa, ou dados diretamente do teclado, sem que ne nhum argumento seja acrescentado a eles, o computador executará a operação solicitada usando o disco como meio de entrada e saída.

No entanto, é somente quando a interface de disco está conectada, que a adição da discriminação do periférico a ser utilizado é estritamente necessária. Para evitar que o programa intencionado venha a ter um funcionamento errado, o programador deve se acostumar a especificar, dentro dos comandos, o periférico a que se destina, ou, no mínimo, fornecer meios aos usuários de proceder a esta especificação, quando necessário, seja pela digitação, do periférico por ele próprio usuário, seja através de um menu de opções, fazendo com que o programa se encarregue desta tarefa.

Outra especificação importante é a do drive para a qual a entrada e a saída são dirigidas. Quando um programa não inclui esta possibilidade, ele ficará limitado ao drive corrente, impondo ao usuário, principalmente aquele que possui mais de um drive, uma séria restrição no processo de arquivamento de dados.

O BASIC DE DISCO fornece os principais recursos do DOS na forma de comandos, como por exemplo, a obtençào do diretório do disco, a memória livre disponível, a cópia de um drive para outro, etc. Entretanto, certos comandos do DOS têm equivalência com uma analogia restrita e outros comandos não têm equivalência alguma. Por exemplo: no DOS, o comando DIR fornece uma boa parte das informações contidas no diretório do disco, mas o comando FILES, do BASIC DE DISCO, lista apenas os nomes dos arquivos. Usando o nome do drive seguido de dois pontos, o usuário do DOS pode mudar o drive corrente, enquanto que no BASIC DE DISCO, para alcançar o mesmo objetivo, é necessário FAZER UM PROGRAMA, já que inexiste um comando com este recurso. A confecção de uma rotina com tais finalidades envolve a manipulação de variáveis do sistema, o que nem todos os programadores estão habilitados a fazer. Neste ponto, a linguagem BASIC falha na sua missão de, como linguagem de alto nível, facilitar a programação pelo usuário que não se interessa nos detalhes operacionais da máquina que ele utiliza.

Também nós ficaremos restritos aos comandos descritos do BASIC DE DISCO, exemplificando com linhas de programa, mas sem nos envolvermos profundamente com técnicas de programação, devendo o leitor consultar, para este fim, uma literatura mais específica.

A listagem dos comandos e funções do BASIC DE DISCO será apresentada em ordem alfabética, com enunciado das SINTAXES e DESCRIÇÃO das respectivas operações. Os comandos do BASIC contidos na máquina padrão não serão aqui repetidos, com exceção daqueles que dizem respeito às operações de entrada e saída.

Comando do MSX DISK BASIC: BLOAD.

SINTAXE: BLOAD "[<drive>:]<nome do arquivo>" [,R] [,S] [,deslocamento].

FUNÇÃO: Ler e carregar um programa em linguagem de máquina gravado no disco.

DESCRIÇÃO:

Quando o comando BLOAD é dado, o Sistema Operacional lê no cabeçalho

("header") do arquivo citado os endereços inicial, final e de execução do programa. Os dois primeiros endereços servem para localizar o programa na memória do micro, enquanto que o terceiro é utilizado para rodá-lo, quando o argumento ",R" é dado.

A cláusula ",S" obriga o carregamento do arquivo na VRAM (RAM de Vídeo), a qual se dispõe de forma independente da RAM do computador. Um total de 16 KBytes estão disponíveis para esta finalidade, dentro dos endereços 0H a 4000H (ou entre 0 e 16383). Arquivos gravados, cujo conteúdo em Bytes for oriundo da VRAM, deverão ser carregados obrigatoriamente com esta cláusula.

O usuário só poderá usar uma das duas cláusulas citadas acima.

O item <deslocamento> só poderá ser utilizado quando o programa for relocável. Nas condições normais de funcionamento deste comando, o deslocamento não é mencionado, assumindo-se que é igual a zero.

Comando do MSX DISK BASIC: BSAVE.

SINTAXE: BSAVE "[<drive>:]<nome do arquivo>",<endereço inicial>,<endereço final> [,<endereço de execução>] [,S].

FUNÇÃO: Salvar o conteúdo da memória especificada nos endereçamentos para o disco.

DESCRIÇÃO:

Este comando é utilizado para salvar um programa em linguagem de máquina, alojado na memória do computador. O endereço de execução muitas vezes é idêntico ao endereço inicial quando então a sua menção é facultativa. Por isso, se o usuário comandar uma gravação sem mencionar este endereçamento, esta condição será assumida.

O conteúdo da memória de vídeo (VRAM) também pode ser salva, se for mencionada a cláusula ",S". Neste caso, o arquivo só poderá ser lido pelo comando BLOAD, acrescido da mesma cláusula. Telas gráficas construídas pelo usuário podem ser salvas, definindo-se o tamanho do arquivo pelo tamanho da memória da VRAM. Por exemplo:

BSAVE "GRAF3.SCR",&H0,&H4000,S <return> ou

BSAVE "GRAF3.SCR",0,16383,S <return>

Procure adotar, neste caso, a extensão .SCR, uma abreviação convencional de "SCReen" (tela), para que você possa saber depois que se trata de memória VRAM, e que a cláusula ",S" deve ser adotada no comando BLOAD.

Comando do MSX DISK BASIC: CLOSE.

SINTAXE: CLOSE [#<número dos arquivos entre vírgulas>].

FUNÇÃO: Fechar os arquivos cujos números sejam mencionados na linha do comando.

DESCRIÇÃO:

O comando CLOSE fecha os arquivos nomeados na linha de comando. Esta nomeação é aquela que foi feita através do comando OPEN, com a identificação dos arquivos por números. Por exemplo:

CLOSE #1 <return>

CLOSE #2,#3 <return>

Se nenhum arquivo for indicado, TODOS os arquivos abertos serão fechados.

Os seguintes comandos do MSX BASIC fecham arquivos abertos automaticamente: CLEAR, END, MAXFILES, NEW e RUN (sem a cláusula ",R").

O comando CLOSE é de uso obrigatório na manipulação de arquivos seqüenciais: depois que a entrada ou saída de dados terminar, o arquivo deve ser imediatamente fechado, para não prejudicar outras operações do computador. Se o arquivo não for fechado, será emitida uma mensagem de erro.

Comando do MSX DISK BASIC: COPY.

SINTAXE: COPY "[<drive>:]<nome do arquivo fonte>" TO "[<drive>:][<nome do arquivo destino]".

FUNÇÃO: Copiar arquivos de um drive para outro e, opcionalmente, dentro do mesmo drive.

DESCRIÇÃO:

Embora análogo ao mesmo comando do DOS, as funções do COPY estão limitadas às operações de cópia de arquivos de um drive para outro e, assim mesmo, com uma performance bastante limitada em termos de rapidez e do número de vezes que o usuário se obriga a trocar disquetes, quando possui um só drive para fazer a cópia.

O leitor deve reparar que a SINTAXE deste comando é bem diferente do seu congênere do DOS, incluindo o nome do drive e dos arquivos entre aspas. Por exemplo:

COPY "A:FUTEBOL.TXT" TO "B:" <return> ou

COPY "A:FUTEBOL.TXT" TO "B:PARTIDA.TXT" <return>

As regras para a menção dos arquivos, entretanto, são exatamente as mesmas do DOS: o usuário pode optar pela cópia de vários arquivos simultaneamente, com o uso de caracteres-chaves "*" e "?". Por exemplo:

COPY "*.BAS" TO "B:" <return>

A especificação do drive corrente não será necessária, tal como foi mostrado no exemplo acima. Por outro lado, quando o drive destino não for mencionado, a cópia será feita no mesmo drive.

Sob nenhuma hipótese um arquivo poderá ser copiado para o mesmo drive com o mesmo nome, pois o Sistema Operacional se confundirá e danificará o arquivo a ser copiado. Quando isto ocorre, uma mensagem de erro é emitida.

Por outro lado, seguindo-se a SINTAXE correta, a cópia de um arquivo para o mesmo disco, com nome diferente, será bem mais rápida do que a cópia para outro drive. Por exemplo:

COPY "A:TESTE.DOC" TO "A:RELATO.DOC" <return>

Funções do MSX DISK BASIC: CVI, CVS e CVD.

SINTAXES:
<variável numérica inteira> = CVI (<variável string>)
<variável numérica simples> = CVS (<variável string>)

<variável numérica dupla> = CVD (<variável string>)

FUNÇÃO: Converter variáveis string de arquivos randômicos, de volta nas respectivas variáveis numéricas.

DESCRIÇÃO:

Os arquivos randômicos (ou aleatórios) só podem conter variáveis string. Quando o arquivo está prestes a ser escrito (gravado) no disco, todas a variáveis numéricas têm que ser convertidas em strings (ver também as funções MKI$, MKS$ e MKD$, que fazem estas conversões). No processo inverso, à medida em que o arquivo vai sendo lido do disco, as variáveis string são convertidas de volta nas respectivas variáveis numéricas.

A notação descrita na SINTAXE destas funções faz exatamente isso. Para o leitor que não está familiarizado com o BASIC, esclarecemos que a variável escrita à esquerda é aquela que recebe o conteúdo da variável mencionada à direita. Por este motivo, as funções de conversão CVI, CVS e CVD são colocadas neste lado da equação.

CVI (ConVert to Integer) efetua a conversão de uma variável string de 2 Bytes do arquivo randômico, numa variável numérica inteira. Já CVS (ConVert to Single), faz a mesma coisa com uma variável string de 4 Bytes para outra variável numérica de precisão simples, enquanto que CVD (ConVert to Double) converte uma variável string de 8 Bytes em uma variável numérica de precisão dupla.

Exemplos:

```
A=CVI(A$)
B=CVS(B$)
C=CVD(C$)
```

Observação: O tamanho em Bytes relacionado às variáveis string é definido pela instrução FIELD.

Função do MSX DISK BASIC: DSKF.

SINTAXE: DSKF(<número do drive>).

FUNÇÃO: Fornecer a quantidade em Bytes livres para gravação em um disquete.

DESCRIÇÃO:

A função DSKF retorna com um valor em KBytes, o qual, sozinho, não é de grande valia, se não levarmos em consideração a formatação do disco que está sendo lido.

O leitor deve se lembrar, se leu antes o Capítulo1, que um disquete de face dupla aioca dois setores, ou 1024 Bytes, para um cluster, enquanto que num disquete de face simples, um cluster ocupa apenas um setor, ou 512 Bytes.

Quando se usa o DSKF, deve-se multiplicar o valor obtido na função pelo tamanho em Bytes do cluster do disco lido. Assim, as seguintes expressões deverão ser usadas para a obtenção da memória livre:

Disquete de Face Dupla: DSKF(número do drive) * 1024

Disquete de Face Simples: DSKF(número do drive) * 512

Podem-se obter facilmente estes valores com comandos diretos do teclado, pedindo-se ao BASIC para imprimi-los no vídeo:

PRINT DSKF(0) * 1024 <return>

A indicação do drive, neste caso, é feita por números, segundo a seguinte convenção:

0 - drive corrente (drive default)
1 - drive A
2 - drive B
3 - drive C
4 - drive D
5 - drive E
6 - drive F

Os usuários que possuem apenas um drive poderão empregar somente o número 0 como indicação, mas os usuários que têm dois drives deverão dar preferência aos valores 1 e 2, para evitar erros de programação.

A seguir, iremos apresentar a função DSKI$ e o comando DSKO$, relacionando um ao outro. O motivo para isto reside no fato de que ambos operam funções diametralmente opostas. Enquanto a função DSKI$ lê dados contidos no setores de um disco, para armazená-los na memória do micro, o comando DSKO$ retira dados da memória, para escrevê-los nos setores do disquete desejados. Usados conveniente-

mente, DSKI$ e DSKO$ podem transferir dados de um drive para outro, como iremos mostrar mum programa-exemplo, logo a seguir.

Função do MSX DISK BASIC: DSKI$.

SINTAXES: <variável>=DSKI$(<número do drive>,<número do setor) PRINT DSKI$(<número do drive>,<número do setor).

FUNÇÃO: Ler um setor do disco contido no drive especificado e armazená-lo na memória do MSX.

DESCRIÇÃO:

Ao usar a função DSKI$, 512 Bytes pertencentes ao setor de um disquete são armazenados numa região da RAM, cujo endereço inicial está apontado pelo conteúdo dos endereçamentos &HF351 e &HF352. Para saber este endereço, pode-se comandar:

```
PRINT HEX$(PEEK(&HF351)+PEEK(&HF352)*256) <return>
EB95
Ok
```

No exemplo acima, o endereço &HEB95 contém o primeiro Byte do setor lido. As 511 posições de memória seguintes conterão o restante do setor, ficando este entre &HEB95 e &HED94.

Quando se deseja ler o conteúdo de um setor através da função DSKI$, não se pode efetuar nenhum comando do DISK BASIC (como por exemplo: FILES, OPEN, CLOSE, etc.), sob pena de destruir a área de memória onde estão contidos os Bytes do setor lido por DSKI$. Será necessário "dumpear" a área da RAM em questão, através de comandos PEEK e PRINT. Por exemplo:

```
10 FOR E=&HEB95 TO &HED94
20 A$=CHR$(PEEK(E))
30 IF A$>31 THEN PRINT A$
40 NEXT E
```

A numeração dos drives é a mesma da função DSKF, descrita anteriormente (0 para o default, 1 para o A, 2 para o B, etc.). A indicação do número do setor para leitura deve ser feita com a máxima atenção, pois a função DSKI$ não verifica se o setor é válido ou não. Os setores válidos vão de 0 a 719, para discos Face Dupla e 0 a 359, para Face Simples.

Comando do MSX DISK BASIC: DSKO$.

SINTAXE: DSKO$ <número do drive>,<número do setor>.

FUNÇÃO: Escrever (gravar) dados num setor de um disquete.

DESCRIÇÃO:

Quando se deseja escrever dados no setor de um disquete, indica-se ao BASIC DE DISCO a área de memória onde os dados estão contidos e emprega-se o comando DSKO$. O endereço inicial deverá estar na forma LSB-MSB, nos endereçamentos &HF351 e &HF352. Usando o exemplo anterior, com o endereço &HEB95, escreveríamos:

```
20 POKE &HF351,&H95:POKE &HF352,&HEB
```

Os 512 Bytes da RAM contidos a partir deste endereço serão escritos no disco, mas, da mesma forma como no caso anterior, nenhum comando do MSX DISK BASIC poderá ser acionado antes, para evitar que a área da memória utilizada seja destruída.

Uma vez informado o endereço inicial a partir do qual a cópia da memória será feita, a instrução DSKO$, com os parâmetros especificados, fará a gravação no disco. Por exemplo: DSKO$ 2,40 irá gravar no setor 40 do disquete do drive B.

O leitor deverá estar alerta para um fato que apontamos no Capítulo 1, sobre a gravação de dados no disquete, e que é verdadeiro para este comando: o Sistema Operacional não grava informações sobre a escrita de dados no disco, nem verifica se os setores que receberão estes dados através de DSKO$ estão alocados na FAT para algum arquivo. Sendo assim, se os devidos cuidados não forem tomados, corre-se o risco de escrever dados em cima de um arquivo ou programa, danificando-o. Da mesma forma, ao gravar um arquivo, pode-se substituir os dados escritos por DSKO$ pelos Bytes provenientes do mesmo.

Na prática, isto significa que DSKO$ só tem aplicação nos trabalhos de edição dos setores do disco, usado em conjunto com a função DSKI$.

O comando DSKO$ também não verifica se o setor onde a gravação será feita é válido ou não, fato que pode acarretar um erro irrecuperável de entrada e saída (erro de E/S). Os setores válidos são de 0 a 359, para discos Face Simples e 0 a 719, para Face Dupla, e os números dos drives acompanham o padrão descrito na página 171.

Deve-se reparar que, tanto DSKO$ quanto DSKI$, têm a área de buffer de memória indicada pelo mesmo processo, exceto que, quando se usa a função DSKI$, é o próprio DISK BASIC quem armazena o endereço inicial do buffer, ao passo que no comando DSKO$, o endereço deve ser indicado pelo usuário, "pokeando" esta indicação. Se a função DSKI$ for usada e, imediatamente após, o comando DSKO$, sem que seja feita qualquer indicação em &HF351 e &HF352, o último valor depositado ali pelo DISK BASIC será utilizado. Assim, é possível efetuar uma cópia física de um setor de um disquete para outro, sendo nisto que se baseia o copiador listado a seguir:

```
10 REM Programa para cópia física de disquetes.
20 REM Concepção: Alexandre da Costa Medeiros.
30 REM Programação: P.R.P.Elias e S.G.P.Elias(Março/89).
40 REM Utilização somente com dois drives físicos.
50 LOCATE,,0:WIDTH 40
60 CLS:PRINT"Copiador de disquetes de A para B":LOCATE,3
:PRINT"Insira disquetes nos drives A e B":LOCATE,6:PRINT"Tecle<1> -face Simples
ou<2> -face Dupla"
70 F$=INPUT$(1):IF F$="1" THEN N=359 ELSE IF F$="2" THEN N=719
80 IF F$<"1"OR F$>"2" THEN 70
90 LOCATE,9:PRINT"Pressione<espaço> para copiar"
100 A$=INPUT$(1):IF A$<>CHR$(32) THEN 100
110 FOR S=0 TO N
120 PRINT DSKI$(1,S):LOCATE3,12:PRINT"LENDO O SETOR:";S
130 DSKO$2,S:LOCATE,14:PRINT"GRAVANDO O SETOR:";S
140 NEXT S
150 LOCATE,18:PRINT"FIM DA CÓPIA !"
160 END
```

Se você possui dois drives, digite o programa acima para fazer backups de discos de forma rápida e eficiente. Se desejar, faça modificações na sua estrutura, com base no que foi apresentado anteriormente, como por exemplo, redefinir os drives origem e destino.

O programa copiador foi escrito para ser usado em telas de 40 colunas. Se uma placa de 80 colunas estiver conectada ao micro, será conveniente retirar a instrução WIDTH 40 da linha 50, já que não é possível reconfigurar a largura da tela de volta para 80 colunas, porque o comando WIDTH 80 não é aceito pelo Interpretador BASIC (seria necessário resetar o micro). Na execução do programa, o usuário é instruído a inserir disquetes nos drives A e B, e teclar 1 ou 2, para disquetes face simples ou dupla, respectivamente. Uma crítica à entrada deste dado é feita na linha 80, para impedir que valores errados de setores sejam adotados na cópia. O usuário deve prestar atenção, para colocar o disco fonte no drive A somente, caso contrário, seus arquivos serão irremediavelmente danificados.

Função do MSX DISK BASIC: EOF.

SINTAXE: EOF(<número do arquivo>).

FUNÇÃO: Testar se o sinal de fim de arquivo (<control> + Z ou 26 (1AH) na Tabela ASCII) é encontrado num arquivo seqüencial.

DESCRIÇÃO:

Quando um arquivo seqüencial é lido (ver mais adiante o comando OPEN), o sinal de fim de arquivo não aborta automaticamente a leitura e a entrada de dados (comandos INPUT# e LINE INPUT#). Quando se sabe antecipadamente quantos dados serão lidos do disco, é fácil controlar a entrada dos mesmos pelos comandos citados, limitando-se a leitura ao número desses dados. Quando não, é necessário projetar um algoritmo (método) de leitura em que o número de entrada de dados prossiga até que o primeiro sinal de fim de arquivo ("End Of File"), isto é, o Byte 1AH (26), seja encontrado no disco. A função EOF, referida ao número do arquivo aberto, propicia o teste de leitura do Byte 1AH. Quando este é encontrado, EOF retorna com um valor de -1, caso contrário, o valor retornado será zero, correspondendo às situações de verdadeiro e falso, respectivamente.

Se o teste de EOF não for incluído no algoritmo de leitura do arquivo, esta continuará além do seu fim. Neste momento, o Sistema Operacional detectará um erro de leitura após o fim do arquivo: INPUT PAST END (erro número 55, na tabela do MSX).

Exemplo de leitura controlada pelo conhecimento prévio do número de dados a serem lidos:

```
10 OPEN "TESTE.ARQ" FOR INPUT AS #1
20 FOR I=1 TO 40
30 INPUT#1,C$(I)
40 INPUT#1,N$(I)
50 NEXT I
60 CLOSE #1
70 RETURN
```

Exemplo de leitura onde o número de dados a serem lidos é controlado pelo teste de EOF:

```
10 OPEN "TESTE.ARQ" FOR INPUT AS #1
20 C=0
30 IF EOF(1) THEN 70  (ou: 30 IF EOF(1)=-1 THEN 70)
40 INPUT#1,C$(C)
50 INPUT#1,N$(C)
60 C=C+1:GOTO 30
70 CLOSE #1
80 RETURN
```

No algoritmo acima, estabeleceu-se um contador pela variável "C", que é zerada na linha 20. Depois de cada leitura (linhas 40 e 50), o contador é incrementado de uma unidade na linha 60, ordenando-se que o programa volte à linha 30, onde novo teste de EOF é executado. Note que o contador "C" espelha o número real de registros lidos, podendo ser usado para esta determinação.

Note também que, em ambos os casos, as sub-rotinas exemplificadas executam um "loop" (laço); a primeira, através de FOR e NEXT, e a segunda, através de GOTO. As variáveis são indexadas com o auxílio do próprio laço, facilitando a leitura e entrada de dados.

Comando do MSX DISK BASIC: FIELD.

SINTAXE: FIELD [#]<número do arquivo,argura do campo AS <variável string>, etc.(lista de variáveis).

FUNÇÃO: Alocar espaço no buffer de um arquivo randômico para armazenar variáveis string.

DESCRIÇÃO:

FIELD serve para formatar o buffer de memória destinado a um arquivo randômico, antes que as variáveis do arquivo sejam armazenadas. Através de FIELD, determina-se quantos Bytes ou posições devem ser reservados para cada variável, o que serve para preparar o buffer de modo a poder receber dados através do comando GET (leitura) e enviá-los através do comando PUT (escrita).

A lista de variáveis pode ser enunciada com apenas uma instrução FIELD, separando cada variável com vírgulas.

Para compreender melhor o significado do comando FIELD, é preciso lembrar que um registro de arquivo é composto de um conjunto de campos ("FIELDS"), onde serão digitados os dados do registro. Fazendo uma analogia com um arquivo de fichas, pode-se dizer que cada ficha corresponde a um registro, na qual dados como "nome", "endereço, "telefone", etc., correspondem aos campos, neste caso, ocupando um espaço restrito na ficha, imposto por quem a desenhou.

Exemplificando com duas variáveis A$ e B$, de comprimentos 5 e 10 posições, devemos escrever:

FIELD #1,5 AS A$,10 AS B$

É importante mencionar que FIELD apenas prepara o buffer, mas não coloca nem retira de lá nenhum dado ! Para que os dados sejam colocados no buffer criado por FIELD, devem-se usar comandos específicos dos arquivos randômicos, os quais são capazes de ler e posicionar os dados corretamente no buffer (comandos GET, LSET e RSET) e de retirá-los da posição certa para escrita no disco (comando PUT).

Se comandos LET ou INPUT forem utilizados com variáveis definidas por FIELD, o Sistema Operacional perderá seus dados, que, neste caso, serão colocados na área normal de variáveis string da RAM.

Um aspecto fundamental a ser considerado é a relação entre o número de Bytes (ou posições) definidos por FIELD e o tamanho total em Bytes de um registro, definidos por LEN:

Se o total de Bytes em um registro for SUPERIOR ao somatório de Bytes dos campos delimitados por FIELD, haverá desperdício de espaço no arquivo e, conseqüentemente, de memória no disco. Se, ao contrário, o total de Bytes definido por LEN for INFERIOR ao necessário para alocar os Bytes requisitados por FIELD, ocorrerá um

erro por falta de espaço no buffer, ficando este sobrecarregado de dados. O erro emitido será FIELD OVERFLOW, de número 50 na tabela do MSX.

Sendo assim, deve-se fazer uma estimativa, a mais correta possível, do espaço que deverá ser usado para cada campo e definida por FIELD. Se o número de posições utilizado pelo usuário for inferior ao alocado na memória, as posições que sobrarem serão preenchidas por espaço em branco (número 32 ou 20H da tabela ASCII).

Comandos do MSX DISK BASIC: FILES/LFILES.

SINTAXE: FILES ["<drive:><especificação de arquivos>"]. LFILES (idem).

FUNÇÃO: Ler o diretório do disquete inserido no drive especificado e imprimir no vídeo (FILES) ou na impressora (LFILES) os nomes dos arquivos.

DESCRIÇÃO:

Quando FILES é comandado sem argumentos, todos os arquivos do drive corrente são exibidos no vídeo.

Para o usuário com apenas um drive, não há necessidade de especificar o drive no comando FILES. Mas, se o usuário comandar FILES "B:", o Sistema Operacional, tal como no DOS, solicitará a colocação de disquete como drive B.

A especificação de arquivos também segue as mesmas características do DOS, podendo-se usar os caracteres-chaves "*" e "?" (ver comando DIR do MSX-DOS). Por exemplo:

FILES "*.BAS" <return>
FILES "B:CARTA?.DOC" <return>

A exibição dos arquivos no vídeo é sempre feita no formato de janela (equivalente ao DIR/W). O número de colunas dependerá da largura (WIDTH) da tela e da existência de uma placa de 80 colunas instalada no micro.

Já no comando LFILES, a listagem na impressora ocorrerá com a impressão do nome de um arquivo por linha.

Comando do MSX DISK BASIC: FORMAT.

SINTAXE: CALL FORMAT
_ FORMAT.

FUNÇÃO: Chamar a rotina de formatação contida na interface controladora de discos.

DESCRIÇÃO:

Este comando possibilita formatar um disquete com a mesma rotina de formatação usada pelo DOS.. Ao chamar esta rotina, o usuário poderá optar pelo drive e pelo tipo de formatação que melhor lhe convier. A formatação foi amplamente discutida no Capítulo 1 e no comando FORMAT do MSX DOS. o Leitor deverá se reportar a estes trechos do livro, caso persistam dúvidas a respeito.

Comando do MSX DISK BASIC: GET.

SINTAXE: GET[#]<número do arquivo>[,<número do registro>].

FUNÇÃO: Ler um registro de um arquivo randômico e colocá-lo no buffer apropriado, para uso do sistema.

DESCRIÇÃO:

Ao ser lido um registro do arquivo especificado por GET, todos os seus dados são colocados no buffer criado por FIELD, descrito anteriormente. Através deste buffer, é possível, então, coletar os dados do registro usando os comandos convencionais do BASIC, inclusive INPUT# e LINE INPUT#.

O número do arquivo é referido na elaboração do programa e a sua colocação é obrigatória. Entretanto, o número do registro poderá ser omitido, caso o comando GET já tenha sido executado pelo menos uma vez. Neste caso, GET lerá o registro seguinte ao mencionado no último comando GET executado. O maior número que poderá ser indicado no comando GET é 4.294.967.295.

Exemplo de programação com GET:

```
10 OPEN "NOTAS.DAT" AS #2 LEN=10
20 FIELD #2,5 AS N1$,5 AS N2$
```

```
30 FOR I=1 TO 30
40 GET#2,I
50 PRINT"1ª NOTA:";N1$,"2ª NOTA:";N2$
60 NEXT I
70 CLOSE#2
```

Comando do MSX DISK BASIC: INPUT#.

SINTAXE: INPUT#<número do arquivo>,<lista de variáveis>.

FUNÇÃO: Ler dados de um arquivo seqüencial e armazená-los em variáveis definidas no programa.

DESCRIÇÃO:

Após um arquivo seqüencial ser aberto para leiitura de dados (ver mais adiante o comando OPEN), o uso de INPUT# processa a entrada desses dados, como se os mesmos estivessem vindo pelo teclado. Assim, é importante que a lista de variáveis, estipulada no comando, esteja rigorosamente de acordo com as variáveis gravadas no disco, nos aspectos de tipo (numérica, string, etc.) e da ordem com que elas são escritas no disco. Para evitar erros, o programador deve se reportar a esta lista, quando ela foi gravada no arquivo.

A operação de leitura do INPUT# é feita da seguinte forma: as variáveis numéricas são reconhecidas pela leitura do primeiro caractere encontrado, que não seja<espaço> em branco (32 ou 20H), carriage <return> (12 ou 0DH) ou line feed (10 ou 0AH). O término desta variável é determinado pela leitura do próximo caractere que corresponder a um <espaço> em branco, carriage <return>, line feed, ou vírgula (44 ou 2CH). As variáveis string (alfanuméricas) são identificadas pelo primeiro caractere correspondente às aspas (36 ou 24H) e a sua leitura se processará até que seja encontrado novamente o caractere aspas (por este motivo, as aspas são definidas pelo BASIC como CARACTERES DELIMITADORES de variáveis string).

Observações: Nos casos onde a variável string não é gravada com delimitadores, o seu início é também definido pelo primeiro caractere encontrado, diferente de <espaço>, <return> ou line feed. O término da string, neste caso, é determinado pelo próximo aparecimento dos caracteres mencionados, de vírgula ou, ainda, após a leitura de 255 caracteres. O sinal de EOF (<control> + Z) determina o encerramento da entrada de dados, para qualquer tipo de variável.

Exemplo:

```
10 OPEN "VALORES.DAT" FOR INPUT AS #1
20 INPUT#1,A$,V1
30 PRINT A$
40 IF EOF(1) THEN CLOSE:END ELSE GOTO 20
```

Função do MSX DISK BASIC: INPUT$.

SINTAXE: INPUT$(<número de caracteres>[,[#]<número do arquivo>])

FUNÇÃO: Ler a entrada de dados do console ou de um arquivo, de acordo com o número de caracteres estabelecido.

DESCRIÇÃO:

INPUT$ é uma função de leitura de entrada de dados, que pode ser usada com duas finalidades distintas. Quando a função é escrita sem a discriminação de um arquivo, o BASIC assume que a entrada de dados a ser lida será a do terminal (no caso, o teclado). Nesta circunstância, os caracteres digitados pelo usuário não serão "ecoados" no vídeo. Mas o usuário pode contornar este efeito facilmente, pois a função, uma vez terminada a leitura, permite que os dados fiquem à disposição para a execução de qualquer outra instrução. Por exemplo:

```
10 A$=INPUT$(1)
20 PRINT A$
```

No exemplo acima, armazenou-se a leitura em uma variável (A$), logo depois impressa no vídeo com o comando PRINT.

No caso de um arquivo ser mencionado, a sua indicação é feita através do número com que foi aberto para leitura (ver comando OPEN). O número de caracteres, neste caso, deverá corresponder à string gravada que será lida. Se este dado não estiver disponível, pode-se estipular um número arbitrário de caracteres para leitura e repetir a operação até que todo o arquivo seja lido. No exemplo a seguir, usa-se este artifício com o objetivo de ler um arquivo gravado no formato ASCII (por exemplo, um arquivo de texto) e imprimi-lo no vídeo, para que possa ser lido pelo usuário, simulando o comando TYPE do MSX-DOS. O algoritmo adotado é bastante simples: INPUT$ lê um caractere de cada vez APÓS ter sido testado se o arquivo chegou ao fim. O

programa retorna a este teste toda a vez que outro caractere for lido, até que a leitura termine. Para interromper e reiniciar a leitura, tecla-se <stop>.

```
10 ON ERROR GOTO 90
20 CLS:LINE INPUT"INDIQUE O ARQUIVO A SER LIDO";A$
30 IF A$="" THEN 20
40 OPEN A$ FOR INPUT AS #1
50 IF EOF(1) THEN CLOSE#1:END
60 C$=INPUT$(1,#1)
70 PRINT C$;
80 GOTO 50
90 IF ERL=40 THEN PRINT"ARQUIVO NÃO EXISTE":FOR T=1 TO 300:NEXT
T:RESUME 20
```

Ao estudar o funcionamento da função INPUT$, verifica-se que a instrução <return> é automaticamente executada após todos os caracteres serem lidos.

Comando do MSX DISK BASIC: KILL.

SINTAXE: KILL "[<drive>:]<lista de arquivos>".

FUNÇÃO: Deletar um ou mais arquivos no disquete inserido no drive corrente ou no especificado.

DESCRIÇÃO:

O comando KILL funciona de forma idêntica ao seu equivalente do MSX-DOS, o DEL (ou ERASE), aceitando, inclusive, indicações ambíguas de nomes de arquivos, com o uso de caracteres-chaves "*" e "?".

O usuário deve tomar cuidado com a instrução KILL "*.*", pois, ao contrário do DOS, o BASIC DE DISCO não pede confirmação ao usuário após se pressionar <return> !

Qualquer tentativa de deletar um arquivo ainda aberto, provocará a detecção de erro e conseqüente impressão da mensagem FILE ALREADY OPEN (erro número 54).

Comando do MSX DISK BASIC: LINE INPUT #.

SINTAXE: LINE INPUT#<número do arquivo>,<variável string>.

FUNÇÃO: Ler uma string de até 254 caracteres, sem delimitadores, de um arquivo seqüencial, armazenando-a na variável indicada.

DESCRIÇÃO:

LINE INPUT# lê os caracteres da string, até que um sinal de carriage<return> (0DH) seja encontrado. Quando isto ocorre, o comando "pula" a seqüência carriage return-line feed (0DH-0AH), apontando para a próxima string a ser lida pelo comando LINE INPUT# seguinte. Caso não haja outro comando, o anterior será encerrado.

Portanto, o comando será útil para a leitura de um arquivo de dados, no qual os campos tenham sido gravados com "quebra" entre os campos, sinalizada pelos Bytes mencionados acima. Exemplo:

```
10 OPEN "ARQUIVO.DAT" FOR INPUT AS #1
20 IF EOF(1) THEN CLOSE:END
30 LINE INPUT#1,A$
40 PRINT A$
50 GOTO 20
```

Comando do MSX DISK BASIC: LOAD.

SINTAXE: LOAD "[<drive>:]<nome do arquivo>"[,R].

FUNÇÃO: Carregar e, opcionalmente, rodar um programa em BASIC, gravado em formato binário ou ASCII.

DESCRIÇÃO:

O comando LOAD não distingue em que formato esta gravado o programa. Se a cláusula "R" for incluída, o programa será automaticamente rodado. Exemplo:

```
LOAD"B:BANCO3.BAS",R <return>
```

Função do MSX DISK BASIC: LOC.

SINTAXE: LOC (<número do arquivo>).

FUNÇÃO: LOC é uma função que retorna o número do último registro lido ou escrito de arquivos randômicos, ou o número de Bytes lidos ou escritos, se o arquivo for seqüencial.

DESCRIÇÃO:

Após o comando OPEN ser executado para abrir um arquivo randômico, LOC retorna com o número do último registro lido ou escrito (operação de entrada e saída). Entretanto, se nenhuma operação de E/S for feita, LOC retorna com zero.

Nos arquivos seqüenciais, a situação é um pouco diferente, pois os arquivos são divididos em blocos de 256 Bytes. Quando o início de um bloco é lido, há um incremento de 256 Bytes no valor de LOC. Na leitura do último bloco, entretanto, LOC retorna com o número de Bytes total do arquivo, equivalendo à função LOF (adiante). Quando o arquivo é aberto para leitura, LOC assume um valor de 256 Bytes. Para escrita, se nenhum dado for gravado, LOC retornará com zero.

Trecho de um programa com LOC:

```
40 OPEN "FICHAS.DAT" FOR INPUT AS #1
50 IF EOF(1) THEN CLOSE#1
60 A$=INPUT$(1,#1)
70 PRINT"Bytes lidos:";LOC(1);"Blocos:";LOC(1)/256
80 GOTO 50
```

Função DO MSX DISK BASIC: LOF ("LENGHT OF FILE").

SINTAXE: LOF (<número do arquivo>).

FUNÇÃO: Indicar o tamanho em Bytes de um arquivo randômico ou seqüencial.

DESCRIÇÃO:

LOF é uma função que retorna o número em Bytes de um arquivo, podendo ser usada de várias formas: com o comando PRINT, auxilia na obtenção do tamanho do arquivo; com a instrução FOR...NEXT, auxilia a varredura do arquivo, sem a neces-

sidade de se saber previamente o seu tamanho; dividindo-se o seu valor pelo tamanho em Bytes de um registro (arquivo randômico), pode-se saber quantos registros o arquivo tem. Exemplos:

```
10 FOR I=1 TO LOF(1)
20 IF EOF(1) THEN 40 ELSE A$=INPUT$(1,#1):PRINT A$;
30 NEXT I
40 CLOSE #1
```

```
10 OPEN "CLASSE-2" AS #1 LEN=240
20 PRINT "Tamanho do arquivo:";LOF(1)" Bytes"
30 PRINT "Número de registros:";LOF(1)/240
40 CLOSE #1
```

Observação: Com a função LOF, é possível saber o tamanho em Bytes de qualquer arquivo no disco, informação esta obtida normalmente com o comando DIR do MSX-DOS. Para conseguir este dado, deve-se abrir o arquivo examinado e solicitar a impressão dos dados obtidos pela função. Exemplo em programação por modo indireto (linhas de programa) e por modo direto (comandos digitados do teclado e enviados ao Interpretador BASIC por <return>):

Modo indireto:

```
10 OPEN "WS.COM" FOR INPUT AS #1
20 PRINT"Tamanho do arquivo:";LOF(1)
30 CLOSE #1
```

Modo direto:

```
OPEN "WS.COM" FOR INPUT AS #1:PRINT"Tamanho:";LOF(1):CLOSE#1
<return>
```

Existe uma grande vantagem em se usar a programação por modo direto, neste caso, principalmente se não quisermos alterar algum programa em BASIC que esteja na memória do micro, no momento da sua execução.

Comandos do MSX DISK BASIC: LSET/RSET

SINTAXES: LSET <variável string>=<expressão da string>
RSET<variável string>=<expressão da string>

FUNÇÃO: Armazenar dados da memória em um buffer de arquivos randômicos

(registro de saída) definido pelo comando FIELD, através do alinhamento à esquerda (LSET) ou à direita (RSET).

DESCRIÇÃO:

LSET ("LEFT SET") e RSET ("RIGHT SET") têm funcionamento semelhante, exceto que o alinhamento se faz em sentidos diferentes:

```
10 FIELD #1,14 AS A$,14 AS B$
20 Z$="EXAME":X$="FINAL"
30 LSET A$=Z$
40 RSET B$=X$
```

Resultado:

```
LSET: EXAMEbbbbbbbbb      RSET: bbbbbbbbbFINAL
                  1                        1
      12345678901234           12345678901234
```

Note que as posições não ocupadas são preenchidas por espaços em branco (20H ou 32), representados acima por "b". O alinhamento é denominado de justificação à esquerda ou à direita.

Importante: a string alinhada não poderá ser maior do que a largura do campo definido por FIELD, pois se assim for, os comandos LSET e RSET truncarão o seu conteúdo, isto é, deixarão de fora os caracteres excedentes.

Variáveis numéricas podem ser justificadas, desde que isto se faça através das funções de conversão apropriadas (ver MKI$, MKS$ e MKD$).

Exemplo:

```
10 N=50
20 LSET C$=MKI$(N)
```

Outro uso possível de LSET e RSET para variáveis não definidas por FIELD, é conseguido através da função SPACE$ do BASIC. A sintaxe de SPACE$ permite que seja definida uma string de caracteres em branco com um tamanho predefinido entre parênteses. A justificação é feita com o uso desta string, criada por SPACE$.

Exemplo:

```
10 B$="MARIA"
20 A$=SPACE$(8)
30 LSET A$=B$
```

Resultado:

```
MARIAbbb
12345678
```

Este recurso pode ser utilizado para formatar a saída de variáveis a serem impressas.

Observação: A justificação de variáveis é prática adotada em programação, principalmente na confecção dos Bancos de Dados. Por convenção, as variáveis numéricas são alinhadas à direita, enquanto que as strings (alfanuméricas) são alinhadas à esquerda.

Comando do MSX DISK BASIC: MAXFILES.

SINTAXE: MAXFILES= <número de arquivos>.

FUNÇÃO: Especificar o número de arquivos que serão abertos pelo comando OPEN dentro de um programa.

DESCRIÇÃO:

Ao elaborar um programa, o comando MAXFILES indica ao BASIC quantos arquivos serão abertos pelo usuário. Embora a faixa aceitável na expressão <número de arquivos> seja de 0 a 15, somente seis arquivos poderão ser abertos simultaneamente, mesmo que a indicação do comando seja maior do que este valor. A razão para isto ser assim não é do nosso conhecimento.

Quando o usuário trabalha com apenas um arquivo aberto, não há necessidade de se especificar este valor, pois ele é assumido como default. Entretanto, se o usuário especificar que MAXFILES é igual a zero, isto significará para o Sistema Operacional que nenhum arquivo poderá ser manipulado pelo comando OPEN. Não obstante, os comandos SAVE e LOAD continuam operantes.

Comando do MSX DISK BASIC: MERGE.

SINTAXE: MERGE [<drive>:]<nome do arquivo>

FUNÇÃO: Adendar (ou "mergear") um programa em BASIC, gravado no formato ASCII, a outro programa alojado na memória do computador.

DESCRIÇÃO:

Qualquer programa em BASIC poderá ser gravado em formato de texto (formato ASCII) pela instrução SAVE, com a inclusão da cláusula ",A". Isto permite que a leitura do disco simule a digitação das linhas do programa via teclado, o que é muito útil quando se quer fundir um programa gravado no disco com outro na memória do micro.

Deve-se ter em mente que o programa indicado por MERGE será o último a entrar na memória do computador. Portanto, se houver coincidência na numeração das linhas entre os programas, as linhas do programa no disco terão prioridade sobre qualquer outra coincidente.

Assim, quando se desejar "mergear" uma rotina gravada em disco a outros programas, deve-se renumerá-la para valores de linhas os mais altos possíveis, o que diminui o risco da substituição que mencionamos.

Somente arquivos gravados em formato ASCII poderão ser "mergeados". A execução deste comando com outros tipos de arquivos acarretará detecção e mensagem do erro número 61: BAD FILE MODE.

Funções do MSX DISK BASIC: MKI$, MKS$ e MKD$.

SINTAXES: MKI$(<expressão de valor inteiro>)
MKS$(<expressão de precisão simples>)
MKD$(<expressão de precisão dupla>).

FUNÇÃO: Converter expressões numéricas em suas respectivas strings.

DESCRIÇÃO:

Para armazenar qualquer dado numérico em um buffer de arquivo randômico, é necessário convertê-lo em uma string, cujo número de Bytes corresponda ao tipo da expressão convertida:

MKI$ - produz uma string de 2 Bytes, a partir de um valor numérico inteiro.

MKS$ - produz uma string de 4 Bytes, a partir de um valor numérico de precisão simples.

MKD$ - produz uma string de 8 Bytes, a partir de um valor numérico de precisão dupla.

O valor em Bytes de cada string deverá ser indicado na formulação do buffer pelo comando FIELD. As strings convertidas por MKI$ (MaKe Integer), MKS$ (MaKe Single) e MKD$ (MaKe Double) retornarão a expressões numéricas através das funções CVI$, CVS$ e CVD$.

Comando do MSX DISK BASIC: NAME.

SINTAXE: NAME "[<drive>:]<nome antigo>" AS "[<drive>:]<novo nome>".

FUNÇÃO: Renomear um arquivo no disco do drive indicado.

DESCRIÇÃO:

Para renomear um arquivo, as indicações de drive devem ser as mesmas. Quando os drives não são mencionados, é assumido o drive corrente.

Comando do MSX DISK BASIC: OPEN.

SINTAXE: OPEN "[<drive>:]<nome do arquivo>" [FOR <modo> AS [#]<número do arquivo>] [LEN=número de Bytes].

FUNÇÃO: Abrir um arquivo do disco para entrada (leitura) ou saída (escrita ou gravação).

DESCRIÇÃO:

O comando OPEN deve ser escrito de acordo com o tipo de arquivo que será aberto. Se o arquivo for do tipo SEQÜENCIAL, as seguintes SINTAXES são possíveis:

OPEN "drive:arquivo" FOR INPUT AS #<número>- leitura
FOR OUTPUT - escrita
FOR APPEND - escrita no fim (em apenso).

Não especificando o modo, será aberto um arquivo RANDÔMICO (default). Neste caso, a SINTAXE será:

OPEN "drive:arquivo" AS #<número>[LEN=<bytes>].

Note-se que, neste caso, o tamanho (LEN) não é obrigatoriamente citado no comando, muito embora isto seja desejável. Na ausência desta citação, o BASIC assume que o arquivo terá 256 Bytes em cada registro. Este é o valor máximo que um arquivo de qualquer tipo poderá ter em cada registro. A diferença aqui é que os arquivos SEQÜENCIAIS são automaticamente gravados, com o espaço efetivamente ocupado pelas suas variáveis, enquanto que nos RANDÔMICOS, LEN restringe e economiza espaço no disco. O valor mínimo que pode ser adotado para LEN é 1.

Somente valores inteiros podem ser citados como tamanho (LEN) dos registros. Os arquivos devem ser obrigatoriamente numerados e este número servirá de referência para a leitura ou escrita, em todos os comandos que incluam esta citação (INPUT#, PRINT#, etc.).

O máximo de arquivos abertos ao mesmo tempo para leitura é de seis, mesmo que MAXFILES especifique um valor maior. Para escrita, somente um deverá estar aberto.

Se um arquivo for aberto para leitura, mas não for achado no disco do drive indicado, será emitida uma mensagem de erro de arquivo não encontrado (FILE NOT FOUND, de número 53). Em contrapartida, se um arquivo aberto para escrita existir no disco destino, ele será automaticamente substituído pelo novo arquivo.

Comandos do MSX DISK BASIC: PRINT#/PRINT# USING

SINTAXE: PRINT# <número do arquivo>,[USING"formato";]<lista de variáveis>.

FUNÇÃO: Escrever dados num arquivo seqüencial.

DESCRIÇÃO:

O comando PRINT# escreve uma ou mais variáveis, ou expressões, no disco, de forma semelhante ao conhecido comando PRINT, usado para o terminal de vídeo.

Por este motivo, devem ser implementados delimitadores adequados para evitar truncar as várias expressões contidas num só comando PRINT#. Valores ou expressões numéricas devem ser delimitados por ";" e os valores alfanuméricos por vírgulas ou por strings de espaços em branco (" "). Por exemplo:

```
PRINT#1,A;X;D

PRINT#1,A$,B$;" ";C$
```

Como se pode observar, usa-se a mesma técnica do comando PRINT, pois os campos de impressão, neste caso, são os mesmos.

Delimitando-se as strings com vírgulas, pode-se posteriormente, armazenar seus conteúdos em variáveis separadas durante o processo de leitura. Para escrever vírgulas no corpo do texto, solicita-se a "impressão" de CHR$(34).

A mesma coisa ocorre com PRINT# USING, podendo-se usar as mesmas técnicas de formatação do BASIC convencional. Por exemplo:

```
PRINT#2,USING"####";N
```

PRINT# só pode ser usado quando o arquivo for aberto com o modo FOR OUTPUT. Se algum outro dispositivo diferente do disk drive for mencionado, a saída de PRINT# será direcionada para ele, exatamente com a mesma SINTAXE.

Comando do MSX DISK BASIC: PUT.

SINTAXE: PUT[#]<número do arquivo>[,<número do registro>].

FUNÇÃO: Escrever dados do buffer de um arquivo randômico no arquivo aberto pelo comando OPEN.

DESCRIÇÃO:

Depois que os dados entrados via teclado são armazenados no buffer criado para

um arquivo randômico, pelos comandos LSET/RSET, eles ficarão disponíveis para a gravação no disco, através do comando PUT.

É essencial que o comando PUT esteja associado ao número do arquivo. Se o número do registro não for especificado, o Sistema Operacional assumirá que a gravação será feita no registro, imediatamente após o último comando PUT executado. Os números de registro admitidos pelo BASIC DE DISCO vão de 1 até 4.294.967.295.

Exemplo de programação:

```
10 OPEN "ESCRITA" AS #1, LEN=10
20 FIELD #1,10 AS ES$
30 FOR I=1 TO 30
40 INPUT "Rubrica (máximo: 10 car.)";R$
50 LSET ES$=R$
60 PUT#1,I
70 NEXT I
80 CLOSE #1
```

Comando do MSX DISK BASIC: RSET (ver LSET/RSET, pág.183)

Comando do MSX DISK BASIC: RUN.

SINTAXE: RUN "[<drive>:]<nome do arquivo>"[,R].

FUNÇÃO: Carregar e rodar um programa em BASIC gravado no disco do drive especificado.

DESCRIÇÃO:

Ao ser executado, o comando RUN fecha todos os arquivos abertos por OPEN, o que equivale ao comando CLOSE sem argumentos, ao mesmo tempo em que apaga da memória o programa atual, se existente, o que, por sua vez, equivale ao comando NEW.

RUN é um comando muito útil para substituir o programa atual pelo programa do disco, sem prejuízo no funcionamento do computador. Digamos, por exemplo, que se tenha um programa muito longo em BASIC, para rodar no ambiente do BASIC DE

DISCO. Muitas vezes, isto leva a saturar a memória do computador, de tal forma que o programa não roda da maneira como desejamos. Isto sem falar no fato de que há uma limitação de memória livre quando a interface de discos é conectada, o que impede a carga de todas as linhas dos programas cujo tamanho em Bytes exceda esta memória. Para contornar todos estes problemas de forma elegante, deve-se partilhar o programa principal em rotinas separadas e salvá-las no disco como programas separados, que serão chamados por RUN à medida em que forem sendo necessárias.

Se a situação acima descrita envolver um arquivo aberto pelo programa principal, então, deve-se usar a cláusula ",R" no comando RUN, para evitar que ele seja fechado.

Exemplo:

```
10 LINE INPUT "Digite a opção:";A$
20 IF A$="1" THEN GOSUB 640
30 IF A$="2" THEN RUN "SUBR-1"
40 IF A$="3" THEN RUN "SUBR-2",R
```

Repare que no caso anteriormente descrito, o comando RUN substitui totalmente o programa contido na memória. O programador deve prever o retorno ao programa principal nas rotinas escritas à parte, usando a mesma estratégia. Para não substituir completamente o programa na memória, seria necessário usar o comando MERGE, mas, neste caso, a execução do programa atual seria automaticamente encerrada, obrigando o usuário a usar novamente o comando RUN, mas, desta vez, na forma de um comando DIRETO DO TECLADO. Se RUN for usado sem argumentos, o programa rodará do início, mas RUN linha ou GOTO linha rodam o programa a partir do ponto especificado.

Observação: O comando RUN não faz distinção entre programas gravados em formato binário ou ASCII, mas o MERGE só carrega arquivos em ASCII. Seu equivalente na máquina MSX padrão, neste último caso, é RUN "CAS:[<arquivo>.]".

Comando do MSX DISK BASIC: SAVE.

SINTAXE: SAVE "[<drive>:]<nome do arquivo>"[,A].

FUNÇÃO: Salvar um programa em BASIC no disco do drive especificado, em formato binário (default) ou em formato ASCII, se a cláusula ",A" for usada.

DESCRIÇÃO:

Depois de digitar ou carregar um programa em BASIC na memória do micro, pode-se salvá-lo no disquete desejado, através do comando SAVE.

Normalmente, o programa é gravado em formato binário condensado, mas se a cláusula ",A" for adicionada ao comando, o programa passará para o disco no formato de texto, através da gravação de códigos da Tabela ASCII. É neste último formato que o programa pode ser "mergeado" a outro, ou então lido e rodado em outro computador que possua um interpretador BASIC que aceite os comandos do MSX, e com formatação de discos compatível.

Comando do MSX DISK BASIC: SYSTEM.

SINTAXES: CALL SYSTEM
_ SYSTEM.

FUNÇÃO: Sair do BASIC DE DISCO e retornar ao ambiente DOS.

DESCRIÇÃO:

Quando se aciona o comando BASIC no DOS, o MSX passa a operar com o BASIC DE DISCO. O usuário tem opção de retornar ao ambiente DOS pelo comando CALL SYSTEM. Se o fizer, deve-se certificar que não haja nada na memória do micro que necessite ser salvo em memória periférica, pois este comando destrói todos os dados e linhas de programa porventura existentes no momento de sua execução.

O comando CALL SYSTEM só funciona se a ativação do BASIC DE DISCO tiver sido feita pelo comando BASIC do DOS.

Função do MSX DISK BASIC: VARPTR.

SINTAXE: VARPTR (#<número do arquivo>).

FUNÇÃO: Retorna com o endereço do início do buffer do arquivo, indicado pelo número.

DESCRIÇÃO:

O Sistema Operacional de disco reserva duas áreas para armazenamento de dados dos arquivos abertos pelo usuário, uma para anotações de informações pertinentes ao arquivo e outra, apontada pela função VARPTR (VARiable PoinTeR), que armazena os dados a serem lidos ou escritos.

VARPTR indica o endereço de memória que corresponde ao início deste buffer. Quando o número retornado é negativo, pode-se somar a ele 65536 (64 KBytes), que corresponde ao endereço real em notação decimal.

Comando do MSX DISK BASIC: VERIFY.

SINTAXES: CALL VERIFY ON (_ VERIFY ON)
CALL VERIFY OFF (_ VERIFY OFF).

FUNÇÃO: Ativar (ON) ou desativar (OFF) a rotina de verificação de gravação.

DESCRIÇÃO:

Este comando permite acessar a rotina de verificação de uma gravação feita num disquete. O default do computador é OFF.

A rotina de verificação torna a escrita mais lenta, porém mais confiável, porque todos os dados gravados são conferidos com os originais. Quando algum erro é detectado, o Sistema Operacional emite uma mensagem de erro de entrada e saída: DISK I/O ERROR, de número 69.

Arquivos SEQÜENCIAIS e RANDÔMICOS

Até agora, uma série de comandos e funções do MSX DISK BASIC foi apresentada, muitos dos quais são utilizados para o arquivamento de dados no disco. Este recurso é um dos mais interessantes e importantes que os computadores oferecem e, embora se possa objetar que a linguagem não é a ideal para se construirem arquivos, em função principalmente da velocidade de processamento, o fato é que o MSX DISK BASIC oferece uma gama enorme de recursos a qualquer usuário que queira construir um Banco de Dados, usando um disk drive como periférico, com uma relativa facilidade de construção do programa gerenciador.

Antes de partir para a elaboração deste programa, o usuário pode optar pela forma de armazenamento que mais lhe convém, mas, para isto, é preciso ter uma noção operacional das formas de armazenamento passíveis de serem executadas através do BASIC DE DISCO.

As formas de armazenamento a que nos referimos são as dos arquivos SEQÜENCIAL e RANDÔMICO. A primeira delas, possivelmente, já será familiar aos usuários da máquina padrão do MSX, aquela à qual se conecta apenas um gravador cassete, portanto, sem as rotinas do BASIC DE DISCO. Os termos SEQÜENCIAL e RANDÔMICO se referem à maneira como os dados são lidos ou escritos no disco, depois que o arquivo é gravado pela primeira vez.

Num arquivo SEQÜENCIAL, como o próprio nome já diz, cada vez que o arquivo é aberto, é obrigatória a leitura ou gravação de todos os dados indistintamente, na seqüência em que eles aparecem no arquivo. Já no arquivo RANDÔMICO, é possível ler ou gravar apenas um registro de cada vez, em qualquer posição do arquivo, daí o termo "RANDÔMICO", que vem de "RANDOM", que quer dizer "ALEATÓRIO". Estes arquivos também são chamados de DIRETOS, por permitirem o acesso aos dados desta maneira.

Antes de optar por um ou outro tipo de arquivo, o usuário deve pesar na balança quais as vantagens e desvantagens de cada um deles. No arquivo SEQÜENCIAL, a grande desvantagem é a obrigatoriedade da leitura completa do arquivo, para que se possa ler um registro. A implicação principal deste método é a ocupação de memória RAM do computador, limitando fisicamente o tamanho do arquivo ao espaço nela disponível. Em contrapartida, depois que o arquivo é transferido para a memória do micro, o PROCESSAMENTO de busca de dados é o mais veloz possível, porque, agora, o computador só depende dele. Este é o motivo, aliás, pelo qual se procura incrementar a memória RAM dos computadores, através de placas de expansão. Por outro lado, os arquivos RANDÔMICOS podem ser bem maiores que os SEQÜENCIAIS, já que dependem da memória periférica (espaço disponível no disquete) e não da RAM. A grande desvantagem, no caso, é ter que acessar o disco toda vez que a leitura ou gravação de um dado for necessária, perdendo com isso um tempo enorme. Esta perda de tempo é proporcional à lentidão do sistema de disco que o usuário acoplou ao computador e depende de uma série de fatores, entre os quais se inclui a qualidade do acionador (disk drive) conectado à controladora. Portanto, como se vê, ganha-se de um lado, mas perde-se do outro.

O leitor deve ter em mente que quanto menos operações de entrada e saída (I/O), mais rápido será o processamento de dados do programa gerenciador. Se o volume de dados não é muito grande, adquire-se óbvia vantagem com o uso do arquivamen-

to SEQÜENCIAL. O único transtorno, no caso, é a exigência de um programa gerenciador mais elaborado, no qual é necessário armazenar os dados do disco em variáveis indexadas, para que se possam manipulá-los.

Em arquivos muito longos, a forma SEQÜENCIAL pode ser mais vantajosa operacionalmente, se o número de alterações no arquivo for normalmente grande, caso contrário, os arquivos RANDÔMICOS devem ser escolhidos, já que as modificações são feitas diretamente no registro gravado no disco. Neste tipo de arquivamento, não há tantas restrições ao tamanho do arquivo, porque ele depende mais da memória periférica, porém, o critério na elaboração da ficha (registro) deve ser mais consciente, para evitar desperdício de espaço no disco (LEN superestimado) ou falta de espaço para digitar dados (LEN subestimado). Este problema é mais difícil de ocorrer nos arquivos SEQÜENCIAIS, porque nestes, o Sistema Operacional aloca no disco apenas o espaço reservado às variáveis de memória do arquivo na RAM.

Entre os comandos do DISK BASIC que podem ser usados para ambos os arquivos estão:

SEQÜENCIAL - OPEN, CLOSE, INPUT#, LINE INPUT#, LOC, LOF, MAXFILES, PRINT# e PRINT# USING.

RANDÔMICO - OPEN, CLOSE, PUT, GET, CVI, CVS, CVD, MKI, MKS, MKD, LSET/RSET, FIELD, MAXFILES, LOC e LOF.

Note que, no SEQÜENCIAL, para escrever usa-se PRINT# e para ler, INPUT#. Já no RANDÔMICO, usa-se PUT e GET, respectivamente. No primeiro caso, uma vez de posse das variáveis, pode-se escrever diretamente no disco. No arquivo RANDÔMICO, há necessidade de se criar antes um buffer especial, pelo comando FIELD e, depois, alinhar os dados das variáveis à esquerda ou à direita, durante a sua transferência para o buffer. Só depois será possível escrever os dados no disco.

Detecção de erros no BASIC DE DISCO

Durante o desenrolar de um programa que realiza operações de entrada e saída, é possível que sejam detectados erros de qualquer natureza, o que pode prejudicar irremediavelmente o seu funcionamento.

Nos tempos dos nossos avós, dizia-se que "é preferível prevenir do que remediar", um santo ditado popular, altamente aplicável nestas circunstâncias: ao desenvolver um programa com leitura e gravação no disco, o programador deve partir do princípio

que o erro pode sempre ocorrer. Por exemplo, ao abrir um arquivo que não se encontra no disco, a mensagem FILE NOT FOUND (ARQUIVO NÃO ENCONTRADO) é na realidade UM ERRO (!) e não uma orientação ao usuário. Se for prevista uma rotina de tratamento de erro adequada, ele será detectado SEM INTERRUPÇÃO DO PROGRAMA e será facultado ao usuário dizer o que fazer com ele. O importante aqui é não deixar o programa parar de rodar.

As rotinas de detecção de erro existem com este objetivo: impedir a paralização do programa, para evitar prejuízos causados ao usuário e permitir que o erro seja tratado convenientemente.

Quando ocorrer um erro, ele será detectado independente da vontade do usuário. Assim, só resta ao programador indicar ao BASIC o que fazer quando isto acontecer. Na redação do programa, isto corresponde à declaração ON ERROR GOTO, escrita nas primeiras linhas. Na linha indicada por esta declaração, o programador redige o tratamento do erro. Para fazer isto, dispõem-se de duas variáveis poderosas: ERR e ERL. A primeira armazena o número de código do erro detectado e a segunda, a linha onde o erro ocorreu. Manobrando-se essas variáveis com critério, através de rotinas IF...THEN...ELSE, consegue-se um eficiente tratamento de erro. O passo final é determinar em que linha o programa deve continuar. Para isto, é obrigatório o uso do comando RESUME, que significa "REASSUMA" ou "CONTINUE" na linha tal. Este comando é bastante flexível: RESUME NEXT manda continuar na linha seguinte àquela onde o erro ocorreu; RESUME 0 manda reassumir na mesma linha onde o erro ocorreu e RESUME ha indica um lugar específico do programa.

É de suma importância que a rotina de tratamento de erros preveja todos os casos possíveis, o que nem sempre é fácil. Mas, ainda assim, existem meios de se contornar este problema, como demonstrado a seguir.

Inicialmente, vamos colocar um caso simples: a detecção de um arquivo não encontrado. Note que este tipo de erro só deverá ocorrer quando o programa executar uma operação de leitura no disco. Isto restringe a detecção de erros aos comandos pertinentes a esta operação, como FILES, OPEN, LOAD, MERGE e KILL. O programa poderia ser escrito assim:

```
10 ON ERROR GOTO 500
20 REM MENU PRINCIPAL
                |
90 FILES
100 OPEN A$ FOR INPUT AS #1
                |
```

```
500 IF ERL=90 THEN PRINT"DISCO VAZIO":RESUME 20
600 IF ERL=100 THEN PRINT"ARQUIVO NÃO ENCONTRADO":RESUME 20
```

Repare que a detecção de FILE NOT FOUND é factível na linha 90, mas não é a única possibilidade da linha 100, o que faria esta rotina ficar falha. Para corrigi-la, podemos fazer uma triagem de erros diferentes de FILE NOT FOUND, no início da rotina:

```
500 IF ERR<>53 THEN PRINT "OCORREU UM ERRO Nº ";ERR:RESUME 20
510 IF ERL=90 THEN PRINT"DISCO VAZIO":RESUME 20
520 IF ERL=100 THEN PRINT"ARQUIVO N  3O ENCONTRADO":RESUME20
```

Melhorou bastante, mas falta ainda dar tempo ao usuário de ler as mensagens que serão exibidas. Podemos alterar a rotina retirando a instrução RESUME 20, que está repetida, e colocando-a em separado com outra instrução que possa fazer uma pausa no programa.

```
500 IF ERR<>53 THEN PRINT "OCORREU UM ERRO Nº ";ERR
510 IF ERL=90 THEN PRINT"DISCO VAZIO"
520 IF ERL=100 THEN PRINT"ARQUIVO NÃO ENCONTRADO"
530 PRINT"TECLE ALGO":A$=INPUT$(1):RESUME 20
```

As rotinas de detecção de erro podem adquirir uma elaboração bastante sofisticada, dependendo da imaginação do programador e da complexidade do programa.

É essencial a quem programa, e muitas vezes a quem é usuário, o conhecimento dos códigos de erro da tabela interna do MSX. Por isso, torna-se conveniente ter sempre à mão a listagem dos códigos com o respectivo significado, na forma de uma "cartão de referência", ou através do manual do computador. Neste último, geralmente são listados os códigos da máquina MSX padrão, onde não constam os erros do BASIC DE DISCO. Por este motivo, passamos a listá-los no original em inglês, com a tradução entre parênteses, seguida de uma breve explicação.

Nota: a lista completa dos códigos de erro, incluindo os listados-abaixo, estão contidos em nosso livro"Tudo sobre o MSX-WORD das Versões 1.6 a 3.0", publicado pela CIÊNCIA MODERNA COMPUTAÇÃO.

LISTA DE ERROS DO MSX DISK BASIC:

50 - FIELD OVERFLOW (sobrecarga em FIELD)
Uma declaração FIELD é tentada para alocar mais Bytes do que foi especificado no tamanho do registro, em um arquivo randômico.

51 - INTERNAL ERROR (erro interno)
Ocorreu um erro irrecuperável interno no BASIC.

52 - BAD FILE NUMBER (número de arquivo incorreto)
Um comando do BASIC DE DISCO refere-se a um número de arquivo não aberto ou fora das especificações da instrução MAXFILES.

53 - FILE NOT FOUND (arquivo não encontrado)
Um comando LOAD, OPEN, MERGE ou KILL não encontrou o arquivo especificado ou o comando FILES encontrou um disco vazio.

54 - FILE ALREADY OPEN (arquivo já aberto)
Um comando OPEN ou KILL foi tentado com o arquivo já aberto antes.

55 - INPUT PAST END (INPUT após fim-de-arquivo)
Um comando INPUT foi tentado após o arquivo ter sido completamente lido, ou para um arquivo vazio. A função EOF deve ser usada para evitar este erro.

56 - BAD FILE NAME (nome de arquivo incorreto)
Nomeação ilegal de um arquivo em disco.

57 - DIRECT STATEMENT IN FILE (declaração direta no arquivo)
Uma declaração direta foi encontrada num arquivo gravado no formato ASCII, o que provocou a interrupção da leitura.

58 - SEQUENTIAL I/O ONLY (E/S seqüencial somente)
Um comando GET ou PUT está sendo usado num arquivo seqüencial.

59 - FILE NOT OPEN (arquivo não aberto)
Um comando de leitura ou escrita está sendo tentado em um arquivo não aberto.

60 - BAD ALLOCATION TABLE (erro na FAT)
O disquete não está formatado, ou está com problemas de leitura na FAT.

61 - BAD FILE MODE (acesso incorreto)
Os comandos GET e PUT foram usados em um arquivo seqüencial, ou tentativa de usar LOAD com um arquivo randômico, ou então, uma tentativa de abrir um arquivo em um modo ilegal.

62 - BAD DRIVE NAME (nome de drive inválido)
Foi indicado um drive não existente, lógica ou fisicamente.

63 - SECTOR ERROR (erro no setor)
Foi detectado um erro de leitura ou escrita num setor do disquete, ou então, foi tentada uma operação num setor inexistente.

64 - FILE STILL OPEN (arquivo ainda aberto)
Tentativa de continuar a execução do programa sem ter fechado o arquivo antes.

65 - FILE ALREADY EXISTS (arquivo já existe)
Tentativa de renomear um arquivo com nome de outro já existente no disco.

66 - DISK FULL (disco cheio)
O disquete não comporta a gravação de outro arquivo.

67 - TOO MANY FILES (diretório cheio)
Não é possível gravar outro arquivo no disco, por falta de espaço no diretório.

68 - DISK WRITE PROTECTED (disco protegido)
Tentativa de escrever dados num disco protegido.

69 - DISK I/O ERROR (erro de E/S no disco)
Ocorrência de erro irrecuperável na leitura ou gravação do disco.

70 - DISK OFFLINE (ausência de disco no drive)
Este erro depende da interface na qual o drive está ligado.

71 - RENAME ACROSS DISK (renomeação em drive errado)
Foi tentada a renomeação de um arquivo com a citação dos nomes em drives diferentes.

APÊNDICE 1

A ROTINA DE PARTIDA

Conforme foi explicado no Capítulo 1, a ROTINA DE PARTIDA é um programa gravado no início da trilha zero, quando o disquete é formatado. Neste programa, uma série de informações sobre a formatação do disquete estão gravadas junto com a rotina de pesquisa e carregamento do Sistema Operacional DOS. Abaixo, está listada a ROTINA DE PARTIDA, escrita num disco face dupla pela interface controladora CD-X2, da MICROSOL:

```
000 EB FE 90 4D 53 58 2D 30 32 20 20 00 02 02 01 00
010 02 70 00 D0 02 FD 02 00 09 00 02 00 00 00 D0 ED
020 53 6B C0 32 F3 C0 36 68 23 36 C0 31 1F F5 11 CD
030 C0 D5 0E 0F CD 7D F3 B7 CA 4B C0 D1 11 A7 C0 0E
040 0F CD 7D F3 3C 28 2E 11 A7 C0 D5 11 00 01 0E 1A
050 CD 7D F3 21 01 00 22 B5 C0 22 DB C0 D1 21 00 3F
060 0E 27 CD 7D F3 C3 00 01 58 C0 CD 00 00 79 E6 FE
070 FE 02 C2 6A C0 3A F3 C0 A7 CA 22 40 11 8B C0 0E
080 09 CD 7D F3 0E 07 CD 7D F3 18 A0 45 72 72 6F 20
090 6E 61 20 63 61 72 67 61 0D 0A 54 65 63 6C 65 20
0A0 61 6C 67 6F 0D 0A 24 00 4D 53 58 44 4F 53 20 20
0B0 53 59 53 00 00 00 00 00 00 00 00 00 00 00 00 00
0C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 53 4F
0D0 4C 58 44 4F 53 20 53 49 53 00 00 00 00 00 00 00
0E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Nesta listagem estão descritos 256 Bytes em notação hexadecimal, correspondentes aos que são armazenados entre &HC000 e &HC0FF, durante a partida. Para vê-lo no vídeo ou na impressora, deve-se usar qualquer um dos utilitários descritos neste livro ou equivalentes, capaz de ler e listar um setor do disco.

Para identificar algum trecho daqueles que serão mencionados a seguir, basta

reparar na numeração das linhas, que corresponde ao primeiro dos 16 Bytes da linha. Ex.: o Byte 003 está na quarta coluna da primeira linha.

Para ler corretamente uma informação disposta na notação parte baixa ("low" ou LSB - Least Significant Byte) e parte alta ("high" ou MSB - Most Significant Byte), efetua-se o seguinte cálculo: número ou endereço = LSB + 256*MSB

Na listagem, a parte baixa é sempre escrita antes da parte alta. Por exemplo, para calcular o número de entradas no DIRETÓRIO, procuram-se os Bytes na posição 011 e 012, e verifica-se que eles são 70 e 00. O cálculo no computador será:

```
PRINT &H70 + &H00*512 <return>
112
```

INFORMAÇÕES CONTIDAS NA ROTINA DE PARTIDA:

INFORMAÇÃO	ENDEREÇO
Instrução de partida no DOS (WBOOT)	000
Nome do fabricante em ASCII	003
Bytes por setor	00B (baixa)
Bytes por setor	00C (alta)
Número de Setores por cluster	00D
Número de setores reservados	00E (baixa)
Número de setores reservados	00F (alta)
Número de FAT's	010
Número de entradas no diretório	011 (baixa)
Número de entradas no diretório	012 (alta)
Número total de setores	013 (baixa)
Número total se setores	014 (alta)
Byte de idéntificação do tipo do disco	015
Número de setores por FAT	016 (baixa)
Número de setores por FAT	017 (alta)
Setores por trilha	018 (baixa)
Setores por trilha	019 (alta)
Número de faces	01A (baixa)
Número de faces	01B (alta)
Número de setores ocultos	01C (baixa)
Número de setores ocultos	01D (alta)

Observações importantes sobre a ROTINA DE PARTIDA:

A string que contém o nome MSXDOS.SYS está entre os Bytes 0A8 e 0B2, enquanto que SOLXDOS.SIS fica entre os Bytes 0CE e 0D8. No início da ROTINA, os Bytes EBH ou E9H, quando presentes, provocam um desvio para o endereço contido nos dois Bytes seguintes, dando início à partida do DOS. Se esta instrução não estiver presente na rotina, o BASIC DE DISCO será ativado.

As informações listadas na tabela anterior, a partir do Byte 00B até o fim (01D), pertencem ao BLOCO DE PARÂMETROS DO DRIVE ("Drive Parameter Block" - DPB), referentes às Versões 2.0 e acima, do MS-DOS.

A ROTINA DE PARTIDA no MSX é bem menor do que aquela utilizada nos micros do tipo IBM-PC. Por este motivo, uma boa parte do setor 0 ficará desocupada, quando o disco for formatado por uma controladora padrão MSX.

Além disso, o usuário do MSX poderá ler um disco formatado no IBM-PC, mas não poderá utilizá-lo para dar partida no MSX-DOS e vice-versa. Por outro lado, a cópia de arquivos pode ser realizada sem problemas por qualquer um dos computadores e, através do recurso da cópia física, é possível replicar a ROTINA DE PARTIDA, de modo a tornar a partida do DOS factível.

APÊNDICE 2

FORMATAÇÃO DE DISCOS NO MSX (MS-DOS COMPATÍVEL)

Quando a MICROSOFT idealizou o PADRÃO MSX, teve o bom senso de adotar nele o tipo de formatação em uso pelo MS-DOS, o que permite que ambos os computadores leiam sem problemas os disquetes gravados pelo outro.

Na prática, isto significou a possibilidade de "migrar" arquivos do MSX para o IBM-PC e vice-versa. Além disso, dependendo do tipo de programa gravado no disco, estes poderão também rodar sem nenhum constrangimento. Por exemplo: programa em BASIC gravado no formato ASCII, com as devidas adaptações de comandos; arquivos "batch", com as mesmas precauções; programas em dBASE II sem nenhuma adaptação, se rodado com seu equivalente no IBM-PC, e assim por diante.

Abaixo, estão listadas as características, para conhecimento do leitor, dos formatos de disco adotados para o MS-DOS em drives de 3", 3 1/2" e 5 1/4":

Byte de identificação do tipo	F8H	F9H	FAH	FBH	FCH	FDH	FEH	FFH
Entradas de Diretório	112	112	112	112	64	112	64	112
Setores por FAT	2	3	1	2	2	2	1	1
Setores por trilha	9	9	8	8	9	9	8	8
Número de lados	1	2	1	2	1	2	1	2
No.de trilhas/lado	80	80	80	80	40	40	40	40
Bytes por setor	512	512	512	512	512	512	512	512
Número de FAT's	2	2	2	2	2	2	2	2
No.de setores/cluster	2	2	2	2	1	2	1	2

Observações sobre a tabela anterior:

Os discos formatados com oito setores por trilha, praticamente não são mais utilizados, embora o MS-DOS possua uma opção de formatação (/8) para este tipo. As controladoras de 5 1/4" do MSX não usam esta formatação.

Também indo para o cemitério de HARDWARE, estão os drives de face simples e suas respectivas controladoras.

Os drives de 5 1/4" são identificados na tabela pela formatação de 40 trilhas e os de 3 1/2", pela de 80.

O Byte de identificação do tipo de disco está contido na ROTINA DE PARTIDA (veja o APÊNDICE 1).

APÊNDICE 3

ALGUNS ENDEREÇOS ÚTEIS DA RAM

Toda vez que um disquete é lido por um drive, certas informações a respeito deste disquete são armazenadas na memória do micro. As áreas utilizadas pelo Sistema dependerão do número de drives físicos instalados e efetivamente usados.

Para um Sistema com apenas um drive, duas situações são possíveis de ocorrer: a primeira, sem teclar <control> na partida, onde um drive físico estará disponível, ao lado de um drive lógico (drive simulado), perfazendo um total de dois drives; a segunda, teclando-se <control> na partida, somente um drive estará apto para o usuário, neste caso, o único drive físico existente.

Quando dois drives são usados no Sistema, as informações na RAM são armazenadas em duas áreas distintas, mas se apenas um drive estiver disponível, somente a parte da memória destinada ao drive B será utilizada:

INFORMAÇÕES	DRIVE A	DRIVE B (*)
Bytes/setor	&HF197 (baixa) &HF198 (alta)	&HF1AC (baixa) &HF1AD (alta)
Setores/cluster	&HF19C	&HF1B1
Número de FAT's	&HF19F	&HF1B4
Entradas no diretório	&HF1A0 (baixa)	&HF1B5 (baixa)
Identificação do tipo de disco	&HF196	&HF1AB
Início do diretório	&HF1A6	&HF1BB

(*) ou drive A, teclando <control> na partida.

Os dados indicados na tabela anterior são os mesmos que o Sistema escreve na ROTINA DE PARTIDA, mas, curiosamente, as informações desta última não são aparentemente utilizadas para qualquer tipo de operação de entrada e saída, servindo apenas para identificar o tipo de disquete formatado.

Outros endereços da RAM são manipulados pelo Sistema, para conter algumas informações que podem ser úteis ao programador, se souber usá-las com inteligência:

&HF347
Indica o número de drives físicos reconhecidos pelo Sistema de Discos, quando o usuário pressiona <control> na partida, ou o número de drives disponíveis, se <control> não for pressionada. Assim, retornará com 01H, se o usuário possui um drive e teclou <control> , ou 02H, em caso contrário. Quando um programa é idealizado para operar com um drive lógico anulado, o teste deste Byte servirá para saber se o usuário teclou <control> na partida.

&HF348
Indica em que slot está conectada a interface de discos. É útil para se poder acessar instruções em linguagem de máquina das rotinas de entrada e saída contidas na ROM da controladora.

&HF341, &HF342, &HF343, &HF344
Indicam em que slots estão as páginas 0, 1, 2 e 3 da RAM do MSX, respectivamente. O Byte retornado dependerá do fabricante da máquina. No EXPERT, por exemplo, o valor 02H será obtido.

&HF346
Indica se o DOS foi carregado antes de se acionar o BASIC DE DISCO, quando o Byte armazenado for diferente de zero. Normalmente, quando o DOS é inicializado na partida, o Byte deste endereço é FFH, caso contrário, deverá ser 00H, indicando que o BASIC DE DISCO foi ativado sem o DOS. Se o usuário ligou o computador sem ter carregado o DOS, poderá fazê-lo posteriormente, "pokeando" neste endereço um valor qualquer diferente de zero e chamando o DOS logo a seguir, com o comando CALL SYSTEM.

Os endereçamentos mostrados neste APÊNDICE, são alguns daqueles que puderam ser coletados para exemplo de armazenamento de dados de interesse, quando o computador opera com o Sistema de Discos.

As informações aqui listadas foram prestadas gentilmente por Eduardo Barbosa e parcialmente contidas na bibliografia citada a seguir, para o Capítulo 1.

REFERÊNCIAS BIBLIOGRÁFICAS:

CAPÍTULO 1:
MSX-DOS CP/M-80 Technical Information. Microsoft Corporation, 1983.

Hoffman, P. MSX: Guia do Usuário. 1ª edição em português. McGraw-Hill, São Paulo, 1986.

Avalon Software. O Livro Vermelho do MSX. 1ª edição em português. McGraw-Hill, São Paulo, 1988.

Barbosa, E. A. Dicas, Macetes e Programas em Assembly Para MSX DISK DRIVE. Ciência Moderna Computação, Rio de Janeiro, 1988.

CAPÍTULOS 2 e 3:
Microsoft MS-DOS: Referência do usuário do MS-DOS, Versão 3.2. Microsoft Corporation, 1986.

CDX-2: Guia do Usuário. Microsol Tecnologia, 1986.

Oliveira, R. S., Silva Jr., R. P. Sistema de Disco Para MSX: SOLX-DOS e BASIC DE DISCO. Editora Aleph, São Paulo, 1987.

CAPÍTULO 4:
Barbosa, E. A. Funcionamento do HELLO. Comunicação Pessoal, 1989.

CAPÍTULO 5:
Velloso, J. R. S. Funcionamento do BKP DISCO. Comunicação Pessoal, 1989.

CAPÍTULO 6:
Oliveira, R. S., Silva Jr., R. P. Sistema de Disco Para MSX: SOLX-DOS e BASIC DE DISCO. Editora Aleph, São Paulo, 1987.

Casari, N. MSX Com Disk Drive. McGraw-Hill, São Paulo, 1987.

Impresso na
GRAPHOS
Rio - (021) 223-1272
com filmes fornecidos.

SISTEMAS OPERACIONAIS DO MSX E SUAS FERRAMENTAS:

É UMA OBRA QUE PROCURA MOSTRAR AO LEITOR O POTENCIAL DE USO DOS SISTEMAS OPERACIONAIS DE DISCO DO MSX. SÃO APRESENTADAS DE FORMA DIDÁTICA AS PRINCIPAIS CARACTERÍSTICAS TÉCNICAS DOS SISTEMAS MSX-DOS, SOLX-DOS E BASIC DE DISCO, ALÉM DE SEREM DESCRITAS COM MINÚCIAS AS BASES UTILIZADAS NA FORMATAÇÃO DOS DISQUETES UTILIZADOS NO MSX.

O LEITOR PODERÁ APLICAR TODO O APRENDIZADO CONTIDO NESTE LIVRO PARA INICIAR-SE NO MS-DOS, O SISTEMA OPERACIONAL DO IBM-PC. COMO BÔNUS DESTE APRENDIZADO, SÃO APRESENTADOS E DISCUTIDOS OS MELHORES PROGRAMAS-FERRAMENTA FEITOS PARA O MSX NO BRASIL.

Sobre os autores:

SERGIO GUY PINHEIRO ELIAS é profissional da área de INFORMÁTICA com larga experiência de trabalho em computadores de médio e grande porte, atualmente aplicando estes conhecimentos no domínio de sistemas empregados em microcomputadores.

PAULO ROBERTO PINHEIRO ELIAS é Mestre em Ciência (M.Sc.) e Professor de Bioquímica Médica no Departamento de Patologia da Faculdade de Medicina da UFRJ. Começou a estudar microinformática no Núcleo de Computação Eletrônica da UFRJ.

DOS MESMOS AUTORES, A **CIÊNCIA MODERNA** PUBLICOU DUAS IMPORTANTES OBRAS QUE NÃO DEVEM FALTAR EM SUA BIBLIOTECA:

dBASE II PLUS MSX SEM MISTÉRIOS
O primeiro livro que trata de forma abrangente e didática a construção de um Banco de Dados gerenciado pela única versão do dBASE dedicado ao MSX.

TUDO SOBRE O MSX WORD DAS VERSÕES 1.6 A 3.0
A única publicação que ensina o leitor a explorar todos os recursos deste processador de Texto, incluindo os procedimentos de adaptação do programa ao equipamento do usuário.

EDITORA CIÊNCIA MODERNA