

# TK 2000/II

sistema operacional em disco

INDICE:

1. INTRODUÇÃO

2.

# INTERFACE DO CONTROLADOR DE DISCO

3. PROPOSTA DE INTERFACIA

3.1. Descrição da interface

3.2. Diagrama de blocos

3.3. Descrição dos dados

3.4. Descrição dos procedimentos

3.5. Descrição dos erros

INTERFACE DO  
CONTROLADOR  
DE DISCO

## ÍNDICE:

<b>1. INTRODUÇÃO</b> .....	1
1.1 O Disquete.....	1
1.2. A Organização do Disquete .....	1
1.3. O TKDOS 3.3 .....	2
1.4. Instalação da interface de disco .....	2
1.5. Cuidados com os Drives e disquetes .....	4
<b>2. OPERAÇÃO INICIAL</b> .....	6
2.1. Introduzindo o TKDOS 3.3.....	6
2.2. "BOOT" do TKDOS 3.3.....	6
2.3. Inicializando novos disquetes .....	6
2.4. Gravando e lendo programas pelo TKDOS .....	8
2.5. O comando CATALOG .....	10
2.5.1. Nomes de arquivos.....	10
2.6. Outros comandos do TKDOS .....	10
2.6.1. DELETE.....	10
2.6.2. LOCK.....	11
2.6.3. UNLOCK.....	11
2.6.4. RENAME.....	12
2.6.5. VERIFY .....	12
2.7. Utilização de comandos do TKDOS dentro de programas .....	12
<b>3. ARQUIVOS SEQUENCIAIS</b> .....	14
3.1. Ativação de arquivos sequenciais.....	14
3.2. Desativação de arquivos sequenciais .....	15
3.3. Gravando arquivos sequenciais .....	15
3.4. Lendo arquivos sequenciais .....	17
3.5. Adicionando dados em arquivos seqüenciais .....	19
3.5.1. O comando APPEND .....	19
3.5.2. O comando POSITION.....	20

<b>4. ARQUIVOS DE ACESSO RANDÔMICO.....</b>	<b>21</b>
4.1. Ativação de arquivos de acesso randômico .....	21
4.2. Lendo e escrevendo em arquivos de acesso randômico.....	22
<b>5. COMANDOS AUXILIARES DO DOS.....</b>	<b>24</b>
5.1. O comando EXEC.....	24
5.2. Comando MAXFILES.....	25
5.3. Comandos MON e NOMON .....	26
5.4. Comando TRACE.....	26
<b>6. ARQUIVOS EM LINGUAGEM DE MAQUINA.....</b>	<b>28</b>
6.1. BSAVE .....	28
6.2. BLOAD .....	28
6.3. BRUN.....	29
<b>APÊNDICE A — MENSAGENS DO TKDOS.....</b>	<b>30</b>
<b>APÊNDICE B — SUMÁRIO DOS PROCEDIMENTOS DO TKDOS.....</b>	<b>34</b>
<b>APÊNDICE C — DESCRIÇÃO DOS PROGRAMAS UTILITÁRIOS.....</b>	<b>39</b>

# 1. INTRODUÇÃO

O Sistema Controlador de Disco possibilita o armazenamento e acesso a informações e programas numa velocidade superior aos métodos convencionais, ampliando assim a performance do sistema.

## 1.1. O Disquete

Com a necessidade da criação de um sistema de armazenamento de dados preciso e rápido e também graças aos sucessos contínuos da tecnologia, chegou-se à criação do disco flexível. Este disco é revestido de material magnético e requer o uso de unidades especialmente desenvolvidas para acessar informações nele contidas, os disk drives ou unidades de discos.

## 1.2. A Organização do Disquete

Para se tornar possível o armazenamento de informações no disco é necessário que, antes de operá-lo, haja uma organização do disco. Desta forma o computador, ao acessar o disquete, para leitura ou para gravação de informações, pode se referenciar dentro do disco da mesma forma que num armário com divisões, prateleiras etc.

Esta organização é alcançada com a formatação do disco, que define o disquete com 35 trilhas concêntricas numeradas de 0 a 34 a partir da borda externa em direção ao centro, cada uma delas dividida em 16 setores numerados de 0 a 15, conforme mostra a figura 1.

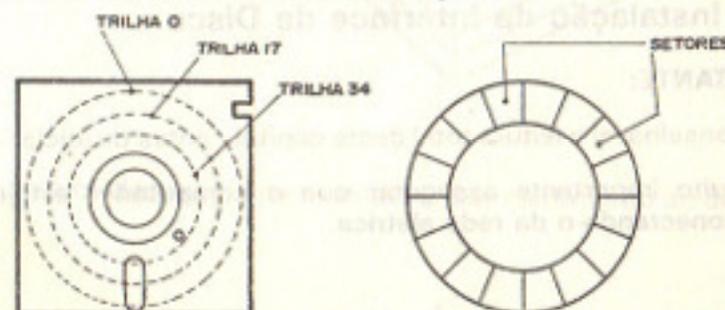


Figura 1

### 1.3. O TKDOS 3.3

Acompanhando a interface de disco, é fornecido um disquete denominado MESTRE. Este disquete contém programas especiais que permitem ao computador usar o disk drive, além de alguns programas utilitários.

O programa central, o mais importante no disco mestre, é denominado TKDOS 3.3 (DOS equivale a DISK OPERATIONS SYSTEM). O TKDOS é o programa que formaliza as interações entre o computador e o disk drive. Uma vez introduzido na memória, ele habilitará uma série de novos comandos, que se constituirão em ferramentas importantes no manuseio dos disquetes.

Na borda do disquete MESTRE existe um selo de proteção. ESTE SELO NÃO DEVERÁ SER RETIRADO, pois cobre uma cavidade especial que, quando tampada, impede que se grave no disquete, protegendo-o assim contra danos. Caso seja necessário gravar, é aconselhável adquirir outro disquete e seguir as instruções contidas neste manual.

selo de proteção

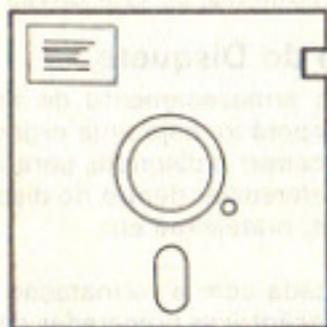


Figura 2

### 1.4. Instalação da Interface de Disco

**IMPORTANTE:**

- 1) É aconselhável a leitura total deste capítulo antes de iniciar a instalação.
- 2) É muito importante assegurar que o computador esteja desligado, desconectando-o da rede elétrica.

O material necessário para a instalação da interface de disco é:

- a. O disk drive, com o respectivo cabo conector.
- b. A interface de disco do TK-2000
- c. O disquete MESTRE
- d. Este manual

Inicialmente ligue o cabo do disk drive à interface conforme a figura 3.

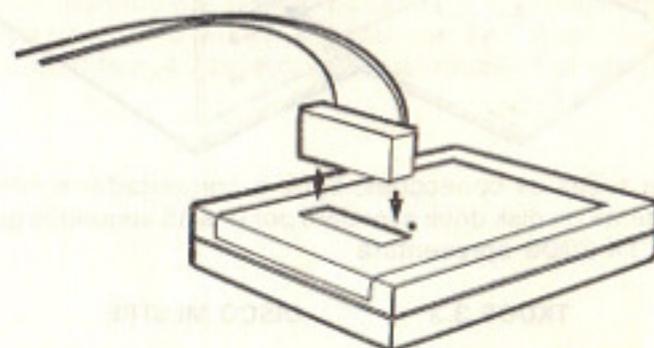


figura 3

Ligue, então a interface de disco ao computador conforme a figura 4.

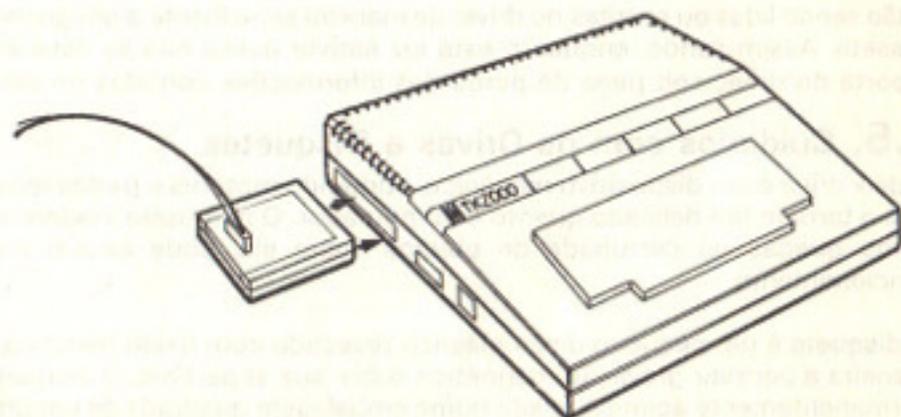


figura 4

A seguir, levante cuidadosamente a tampa do compartimento do disk drive

e introduza o disco MESTRE no seu interior (verificando que, o lado da etiqueta deve estar para cima).

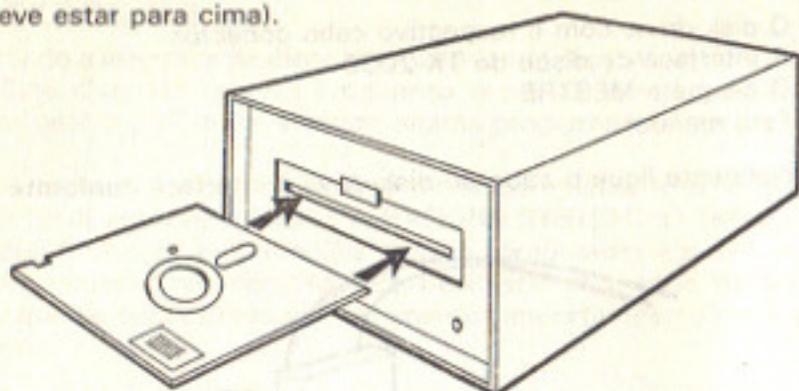


figura 5

Após verificar todas as conexões, ligue o computador e note que a luz vermelha existente no disk drive acenderá por uns 15 segundos quando então a tela de seu TK-2000 apresentará:

TKDOS 3.3

DISCO MESTRE

A operação de ligar o computador e carregar o TKDOS 3.3 na memória é denominada "boot" (leia-se "but"). Este assunto será explicado mais adiante.

A lâmpada existente no disk drive indica quando o computador está acessando o disco. Logo, no momento em que ela está acesa, informações estão sendo lidas ou escritas no drive; de maneira semelhante a um gravador cassete. Assim sendo, enquanto esta luz estiver acesa não se deve abrir a porta do drive, sob pena de perda das informações contidas no disco.

## 1.5. Cuidados com os Drives e Disquetes

O disk drive é um dispositivo mecânico, contendo motores e partes móveis que o tornam tão delicado quanto o computador. O manuseio inadequado, como quedas ou derrubada de objetos sobre ele, pode causar mau funcionamento.

O disquete é um pequeno disco plástico revestido com óxido metálico, de maneira a permitir gravação magnética sobre sua superfície. O disquete é permanentemente acondicionado numa embalagem quadrada de cor preta, que lhe oferece proteção, ajudando a mantê-lo limpo e facilitando o movimento giratório do disco. Esta embalagem nunca deverá ser aberta.

Deve-se evitar tocar na parte exposta do disquete, pois mesmo a impressão

digital poderá apagar, ou até destruir a superfície do disquete. Assim, mantenha sempre seus disquetes acondicionados em suas embalagens originais, prolongando a vida de seus dados e programas.

Muito embora sendo o disquete as vezes chamado de "floppy-disk" (floppy = flexível), é completamente desaconselhável dobrá-los, pois certamente haverá danos ao disco. Deve-se evitar também, variações bruscas de temperatura, respeitando seu limite mínimo de 10 graus, e máximo de 50 graus. Deve ser mantido em ambientes secos, e livres de impurezas. Outra recomendação muito importante é evitar escrever sobre os mesmos. As etiquetas de identificação devem ser atualizadas com canetas de ponta macia.

# 2. OPERAÇÃO INICIAL

## 2.1. Introduzindo o TKDOS 3.3

Como já foi explicado no capítulo anterior, o ato de introduzir o TKDOS, ou qualquer sistema operacional, no computador, é denominado "boot". Este processo nada mais é do que carregar numa área pré-determinada da memória do TK-2000, o sistema operacional, que permitirá a utilização de novos comandos de controle para operação com disquetes.

O TKDOS será carregado numa área de memória que inclui a região destinada a segunda página de vídeo (MP). Sendo assim, uma vez com o TKDOS em operação esta área não deverá ser usada.

## 2.2. "BOOT" do TKDOS

Para efetuar o "boot", desligue o computador, instale a interface de disco (conforme explicado no capítulo anterior), insira o disquete MESTRE e feche cuidadosamente o drive. Ligue o computador. A luz indicadora "EM USO" de seu drive acenderá e o TKDOS será carregado. Caso o computador já esteja ligado e se queira carregar o TKDOS, teclie o comando:

```
>DSK      (RETURN)
```

## 2.3. Inicializando novos disquetes

Para poder-se utilizar um novo disquete é necessário que ele seja inicializado. Apresentaremos abaixo a sequência de inicialização de um disquete, com um programa introdutório, que poderá ser alterado de acordo com a necessidade do usuário:

1. Carregue o TKDOS no computador como já foi exemplificado.

2. Retire o disquete MESTRE, e introduza o disquete a ser inicializado.

3. Digite NEW (RETURN), entrando em seguida o programa introdutório abaixo:

```
>5  REM INICIALIZACAO DO PRIMEIRO DISQUETE  
>10 HOME : REM LIMPA A TELA  
>20 PRINT "DISQUETE NUMERO 1"  
>30 PRINT "OPERADOR: (escreva seu nome) "  
>40 END
```

O propósito do programa acima é identificar rapidamente o disquete após ser introduzido.

4. A seguir escreva:

```
>INIT HELLO
```

Após digitar RETURN, o disk drive irá funcionar por cerca de dois minutos, para formatar o disquete e gravar o sistema operacional. Neste comando também pode ser utilizado o parâmetro volume. Isto quer dizer que poderemos acrescentar no disco uma identificação que, em caso de necessidade, poderá diferenciá-lo de outros discos com arquivos semelhantes. Este parâmetro poderá variar de 1 a 254.

Por exemplo, poder-se-ia identificar o seu novo disco desta maneira:

```
>INIT HELLO,V01
```

Assim, toda vez que for pedido o diretório do disco, o seu número de identificação constará no topo da lista. Caso este parâmetro não seja incluído no comando INIT, o volume será assumido como "254".

5. Finalmente para testar o novo disquete, existem duas alternativas:

a) Desligar e ligar o computador com o disquete dentro do drive; ou,

b) Digitar DSK e em seguida RETURN.

A seguir aparecerá, no alto da tela, a mensagem:

DISQUETE NUMERO 1  
OPERADOR: "seu nome"

## 2.4. Gravando e Lendo Programas pelo TKDOS

Chame o TKDOS utilizando:

>DSK (RETURN)

em seguida digite

>NEW (RETURN)

este comando apagará então o programa HELLO que deverá estar na memória. Escreva o seguinte programa:

```
>5 REM PROGRAMA CONTADOR
>10 FOR I = 0 TO 9
>20 PRINT I
>30 NEXT I
>40 END
```

Ao executar o programa, o resultado será este:

```
>RUN
0
1
2
3
4
5
6
7
8
9
```

Para guardar este programa no disquete (que será chamado de CONTADOR), basta digitar:

>SAVE CONTADOR

Após pressionar RETURN, o DISK DRIVE irá funcionar por alguns segundos e, quando a luz de "EM USO" apagar e o cursor aparecer novamente, a operação estará terminada. Caso se digite:

SAVEA

ou

SAVET nome

o programa será transmitido à saída de cassete, conforme consta no Manual de Operação do TK-2000.

Para carregar o programa novamente do disquete para o computador, bastará seguir as instruções abaixo:

### 1. NEW (RETURN)

Para apagar o programa original que ainda consta na memória.

### 2. LOAD CONTADOR (RETURN)

O DISK DRIVE irá funcionar por alguns segundos. Assim que ele parar, liste o programa (comando LIST). O seu programa CONTADOR, estará novamente na memória do computador. Para executá-lo, bastará digitar:

>RUN (RETURN)

o resultado já apresentado se repetirá.

**OBS:** Caso, no momento de carregar o programa, fosse digitado um nome que não consta do arquivo do disquete, o computador emitiria uma mensagem de erro:

NAO HA ARQUIVO

assim sendo, deve-se conferir o nome do programa, e digitar o comando LOAD, com o nome correto.

Outro modo de executar o programa CONTADOR é, após ter apagado o programa da memória (comando NEW), digitar:

>RUN CONTADOR (RETURN)

o computador se encarregará de procurar o programa no disquete, carregá-lo e, em seguida, executá-lo. Este modo é mais direto que o anterior. Verifique os dois métodos para sentir a diferença.

## 2.5. O Comando CATALOG

O comando CATALOG lista na tela os nomes dos arquivos existentes no disquete, também chamado diretório.

Para cada arquivo do disquete, o CATALOG lista a classe, a quantidade de setores que ocupa no disco e também se o arquivo está travado ou não. Ao digitar:

```
>CATALOG      (RETURN)
```

a tela do TK-2000 apresentará:

```
VOLUME 254  
A 001 HELLO  
A 001 CONTADOR  
>
```

o número 254 é o parâmetro volume, que no caso não foi utilizado. A letra "A" na coluna da esquerda indica o tipo de programa que foi armazenado. No caso, indica que se trata de um programa em BASIC; caso aparecesse a letra "B" indicaria que se trata de um arquivo binário;

Caso o disco possua mais de 18 arquivos, a instrução CATALOG apresentará os primeiros 18 e, para poder prosseguir com a listagem, bastará pressionar qualquer tecla (menos RESET).

### 2.5.1. Nomes de arquivos

Ao se dar um nome para um programa a fim de armazená-lo em disquete, devem ser respeitados um limite máximo de 30 caracteres para o nome. A primeira posição deverá ser preenchida sempre por uma letra e as outras, por qualquer caractere, menos vírgula.

## 2.6. Outros Comandos do TKDOS

### 2.6.1 DELETE

O comando DELETE remove arquivos do disquete.

Inicialmente poderíamos fazer uma pequena experiência com o programa

CONTADOR. Para não perdê-lo, podemos carregá-lo, e a seguir gravar com outro nome (JOAO, por exemplo), assim, serão dois programas iguais, com nomes diferentes e, ocupando áreas distintas do disquete.

Após realizar esta operação, confira através de CATALOG a existência deste terceiro programa e, a seguir, digite:

```
>DELETE JOAO      (RETURN)
```

Após o RETURN, o drive funcionará por alguns instantes. Após isso, pode-se conferir pelo CATALOG que o programa JOAO não consta mais na lista de programas do disquete.

### 2.6.2. LOCK

Alguns programas ou arquivos de dados em disquete devem ser mantidos permanentemente. Para isso, o TKDOS possui uma técnica de proteção chamada "locking" (travamento). Travar um arquivo evita que ele seja acidentalmente eliminado ou que algo seja escrito sobre ele. Para travar um arquivo, basta digitar LOCK, seguido do nome do arquivo, por exemplo:

```
>LOCK HELLO      (RETURN)
```

se tentar apagar um programa travado, o computador dará a seguinte mensagem de erro

```
<TRAVADO>
```

ao se dar o CATALOG aparecerá a seguinte resposta:

```
VOLUME 254  
*A 001 HELLO  
A 001 CONTADOR
```

sendo que o asterisco existente no lado esquerdo do tipo de arquivo indicará que este arquivo está travado.

### 2.6.3. UNLOCK

Quando se deseja regravar ou apagar um arquivo travado, a única forma de se remover o LOCK é com o comando UNLOCK, por exemplo:

>UNLOCK HELLO (RETURN)

Pode-se verificar, através do CATALOG, que o asterisco correspondente ao travamento desapareceu.

#### 2.6.4. RENAME

Para se trocar um nome de qualquer arquivo existente num disquete, utiliza-se o comando RENAME através do formato:

```
RENAME "velhonome";"novonome"
```

O novo nome deverá ser diferente dos que já existem no disco. Um RENAME para um nome já existente provocará duplicidade, e conseqüentemente problemas em posteriores acessos.

O comando RENAME deverá ser executado somente em arquivos destravados.

#### 2.6.5. VERIFY

O comando VERIFY serve para verificar se um determinado arquivo está íntegro. Se quisermos verificar o programa CONTADOR, basta digitar:

```
>VERIFY CONTADOR
```

caso o programa CONTADOR esteja íntegro, não haverá resposta do TKDOS, caso contrário, ocorrerá uma mensagem de erro:

I/O ERRO

É interessante observar que o comando VERIFY não verifica a lógica do arquivo armazenado, e sim, se ele é consistente com o formato usado pelo sistema operacional.

### 2.7. Utilização de Comandos do TKDOS dentro de Programas

O uso dos comandos TKDOS dentro de programas é bastante simples. Basta dar um PRINT do comando desejado, precedido por um caracter de controle (CTRL-D ou CHR\$(4)).

Para exemplificar, se quisermos elaborar um programa que forneça um "CATALOG" dos arquivos em disco, basta digitar:

```
>10 HOME  
>20 D$ = " " : REM ENTRE ASPAS FOI DIGITADO UM CONTROL-D  
>30 PRINT D$"CATALOG"
```

É importante observar que:

1. Os comandos TKDOS devem sempre estar entre aspas.
2. Sempre que definirmos o CONTROL-D, é interessante que se documente esta ocorrência com um REM.
3. Cada comando PRINT deve conter apenas um comando TKDOS.
4. A instrução PRINT que contenha comando TKDOS, nunca deverá terminar com ponto e vírgula (;), nem com vírgula (,).
5. Os seguintes comandos TKDOS só podem ser usados dentro de programas BASIC, ou seja, não são aceitos em modo direto:

```
OPEN  
APPEND  
READ  
WRITE  
POSITION
```

6. Os seguintes comandos TKDOS podem ser usados tanto dentro de programas BASIC como em modo direto:

BLOAD	BRUN	BSAVE	CATALOG
CLOSE	DELETE	EXEC	INIT
LOAD	LOCK	MON	NOMON
RENAME	RUN	SAVE	UNLOCK

7. O comando INIT deve ser usado apenas em modo direto.

# 3. ARQUIVOS SEQUENCIAIS

O TKDOS permite manipular, além de programas, arquivos que contenham informações como: listas de preços, tabelas, etc. Este tipo de arquivo é chamado de arquivo tipo TEXTO. Ao requerer CATALOG, esta classe de arquivo é caracterizada pela letra "T" que precede o nome do arquivo.

O TK-2000 pode operar com dois tipos de arquivo em disco: os seqüenciais e os de acesso randômico (ou aleatórios). Cada um destes arquivos é composto de grupos de dados, também chamados registros.

Os arquivos seqüenciais são caracterizados pela forma de acesso na qual para gravar ou ler um registro, é necessário que se leia todos os anteriores.

Os arquivos randômicos permitem o acesso a um registro diretamente, sem a necessidade de se passar pelos outros registros.

## 3.1. Ativação de Arquivos Seqüenciais

Para manipular um arquivo seqüencial, você deverá utilizar o comando OPEN. Seu formato é:

```
>OPEN nome-do-arquivo
```

O comando OPEN só pode ser utilizado dentro de um programa BASIC, devendo estar incluído dentro de uma declaração PRINT, precedido pelo caracter CONTROL-D (CHR\$(4)).

O exemplo abaixo ativa um arquivo com o nome LIVRO.

```
>2000 D$ = CHR$(4) : REM CONTROL-D
>300 PRINT D$ "OPEN LIVRO"
>400 END
```

Ao rodar o programa acima pode-se verificar, através do comando CATALOG, que o arquivo "LIVRO" foi incluído na lista do diretório.

## 3.2. Desativação de Arquivos Seqüenciais

O programa do item anterior contém um perigo potencial de destruição do arquivo já que este não foi "fechado" após seu uso. Para efetuar este fechamento, usa-se o comando CLOSE. Este comando pode ser dado de duas formas:

```
a) >390 PRINT CHR$(4) "CLOSE"
```

Assim irá desativar todos os arquivos que estejam abertos até este momento (é permitido ativar mais de um arquivo de cada vez) ou,

```
b) >390 PRINT CHR$(4) "CLOSE LIVRO"
```

Neste modo irá desativar especificamente o arquivo indicado, sem alterar os outros.

## 3.3. Gravando Arquivos Seqüenciais

O comando para gravação de dados em arquivos é WRITE.

O comando WRITE sempre deverá estar incluído num comando PRINT, precedido pelo caractere de controle CHR\$(4) (CONTROL-D).

Após a execução do comando WRITE, todas as mensagens que ocorrerem na tela de vídeo (PRINT) serão gravadas no arquivo.

Como exemplo, inclua as seguintes linhas em nosso programa:

```
>305 PRINT D$"WRITE LIVRO"
>310 PRINT "PRIMEIRA FRASE DO ARQUIVO"
```

assim ao listarmos (LIST), teremos:

```
>200 D$ = CHR$(4) : REM CONTROL-D
>300 PRINT D$"OPEN LIVRO"
>305 PRINT D$"WRITE LIVRO"
>310 PRINT "PRIMEIRA FRASE DO ARQUIVO"
>390 PRINT D$"CLOSE LIVRO"
>400 END
```

Note que o prefixo CONTROL-D só foi usado em comandos TKDOS, e nunca quando se está gravando no arquivo.

Quando se grava um arquivo seqüencial, se já existir um arquivo com o mesmo nome no disco, a nova gravação ocorrerá "por cima" do arquivo anterior, o que pode gerar efeitos desagradáveis.

Como solução para esta situação deve ser usado o comando DELETE, que removerá o arquivo anterior do disco. Este comando deverá estar incorporado num comando PRINT, precedido de CONTROL-D.

Esta solução (o comando DELETE) só deve ser aplicada quando tivermos certeza da existência do arquivo no disquete, pois, caso contrário, o computador emitirá a mensagem de erro: NAO HA ARQUIVO.

Para resolver este problema, é aconselhável garantir a existência do arquivo através de um comando OPEN a mais no início do programa. Com esta nova opção, nosso programa obterá o seguinte aspecto:

```
>200 D$ = CHR$(4) : REM CONTROL-D
>210 PRINT D$"OPEN LIVRO"
>220 PRINT D$"DELETE LIVRO"
>300 PRINT D$"OPEN LIVRO"
>305 PRINT D$"WRITE LIVRO"
>310 PRINT "PRIMEIRA FRASE DO ARQUIVO"
>390 PRINT D$"CLOSE LIVRO"
>400 END
```

Pode-se utilizar variáveis para nomes de arquivo. Neste caso, o próprio programa nos perguntará o nome, como mostra o exemplo abaixo:

```
>200 CHR$(4) : REM CONTROL-D
>205 INPUT "NOME DO ARQUIVO = ";A$
>210 PRINT D$"OPEN "; A$
>220 PRINT D$"DELETE "; A$
>300 PRINT D$"OPEN "; A$
>305 PRINT D$"WRITE "; A$
>310 PRINT "PRIMEIRA FRASE DO ARQUIVO"
>390 PRINT D$"CLOSE "; A$
>400 END
```

Pode-se eliminar o efeito do comando WRITE, sem desativar (CLOSE) o arquivo. No lugar desejado acrescenta-se um comando TKDOS neutro, como o próprio CONTROL-D. No nosso exemplo, se antes de encerrar o arquivo

quiséssemos que aparecesse algo na tela poderíamos incrementar as seguintes linhas:

```
>320 PRINT D$
>330 PRINT "VOCE QUER IMPRIMIR OUTRA FRASE NO
ARQUIVO?" : INPUT P$
>340 IF P$ <> "S" THEN 390
>350 PRINT D$"WRITE ";A$
>360 PRINT "SEGUNDA FRASE DO ARQUIVO"
```

Deste modo, a pergunta da linha 330 não seria incluída no arquivo (apareceria somente na tela); e, para que a segunda frase seja gravada no disco temos que executar um novo comando WRITE.

**OBSERVAÇÃO:** Existe a possibilidade de guardarmos números em arquivos seqüenciais. Para isto devemos utilizar o comando PRINT, utilizando a seguir uma variável numérica, ou o número diretamente, observando que, apenas um valor deve ser escrito em cada declaração PRINT. Caso contrário, a inclusão de vírgulas, ou ponto e vírgula, gerará resultados diferentes do desejado.

### 3.4. Lendo Arquivos Seqüenciais

Como já foi explicado anteriormente, a finalidade dos arquivos é a de armazenar dados para uso futuro. No item anterior foi visto como escrever em arquivos. Agora, o comando READ será usado para ler dados de um arquivo.

Fazendo uma analogia com o comando do item anterior, o formato do comando READ é o mesmo que do WRITE, ou seja, deve estar incluído num comando PRINT, precedido de CONTROL-D.

O comando PRINT simples, após o comando WRITE, transfere dados ao arquivo, agora o comando INPUT após o comando READ, irá ler dados do arquivo e associá-los a variáveis em memória.

```

> 500 D$ = CHR$(4) : REM CONTROL-D
> 510 INPUT "NOME DO ARQUIVO A SER LIDO: ";E$
> 520 PRINT D$"OPEN ";E$
> 530 PRINT D$"READ ";E$
> 540 INPUT A$
> 550 PRINT A$
> 560 PRINT D$
> 570 INPUT "VOCE QUER LER MAIS DADOS ? "; Q$
> 580 IF Q$ = "S" THEN 530
> 590 END

```

Através do programa acima note que:

1. Para que a variável Q\$ na linha 570 não fosse associada a nenhum dado do arquivo, tivemos de anular o comando READ na linha 530. Para continuar a ler o conteúdo no arquivo, teremos de retornar a linha 530 com um novo comando READ.
2. Caso quiséssemos ler mais dados do que os existentes no arquivo, apareceria a mensagem FIM ARQUIVO, seguido da mensagem BREAK EM 540, parando o programa.
3. No fim do programa não constou a instrução CLOSE E\$. No caso de apenas leitura de um arquivo não é necessário que o desativemos após a leitura, pois neste caso nada foi gravado e nenhuma informação necessita ser atualizada no disquete.

Para muitos programadores, a ocorrência de uma mensagem de erro seguida de uma interrupção de programa seria uma coisa muito desagradável, para isso existem duas maneiras de evitar a mensagem FIM ARQUIVO que aparece na nota 2:

- a) Pelo uso do comando ONERR GOTO, explicado em detalhes no capítulo 11, do Manual de Operação do TK-2000 COLOR e no APÊNDICE B do mesmo manual.
- b) Se introduzirmos no final do arquivo uma palavra chave indicando fim. O programa ao ler esta palavra deverá identificá-la, e anular o comando READ, continuando então o processamento sem nenhuma mensagem de erro.

Uma forma de indicar fim de arquivo é inserir um caracter neutro, exemplo: CHR\$(0), usando um comando PRINT antes de fechar o arquivo.

**OBSERVAÇÃO:** Da mesma forma que utilizamos a declaração INPUT, para ler dados do arquivo, poderemos também utilizar a declaração GET. Porém, neste caso os dados seriam lidos um a um, caractere por caractere. Assim, com 8 instruções GET, seria possível ler a palavra PRIMEIRA (primeira palavra do arquivo definido no nosso primeiro programa). O único inconveniente do uso do comando GET, é que sempre que for usado, o primeiro caractere do próximo comando PRINT será ignorado. Assim, a única solução é que sempre no comando PRINT, seguinte ao GET, comece com um caractere de controle neutro ao BASIC do TK-2000 (CHR\$(1), por exemplo), evitando assim perdas de dados.

### 3.5. Adicionando Dados em Arquivos Sequenciais

#### 3.5.1. O Comando APPEND

Ao se trabalhar com um arquivo seqüencial, após este ser desativado, é perdido o controle do bloco em que o último dado foi gravado. Lembre-se também que, como foi visto no item em cima dos anteriores, estragando o arquivo anterior.

Assim sendo, sempre que quisermos escrever num arquivo já existente, devemos primeiramente localizar o final dele de modo ser possível aumentá-lo (incrementá-lo) com novos dados.

O comando APPEND, exerce finalidade muito semelhante ao comando OPEN, com a diferença que o segundo ativa o arquivo localizando o apontador de arquivo no início deste, enquanto o primeiro localiza o apontador no primeiro espaço não usado.

Outra diferença importante é que o comando APPEND necessita de um arquivo que já existe, caso contrário emitirá a mensagem NAO HA ARQUIVO.

O formato do comando APPEND é o mesmo que o do comando OPEN ou seja, deve ser incluído em um comando PRINT, além de estar precedido do caractere de controle CONTROL-D.

### 3.5.2. O Comando POSITION

Em muitas ocasiões, teremos a necessidade de alterar algum dado em qualquer lugar do arquivo; assim, necessitamos de um controle do apontador de arquivo para poder acessar um campo específico, tanto para escrita como para leitura.

O comando POSITION só avança o apontador do número de campos partindo da sua atual posição. Os campos são delimitados pelo caractere "carriage return" (ou CHR\$(13)). Caso não consiga identificar tantos campos quanto solicitado no comando, será emitida a mensagem de erro FIM ARQUIVO.

Um exemplo deste comando seria:

```
>500 PRINT D$ "POSITION ARQUIVO, R10"
```

O número que segue a R deve estar entre 0 e 32767.

No exemplo acima o apontador seria deslocado 10 registros a partir da posição atual contando 10 carriage return. Se usássemos R0, o apontador não se moveria.

Note que:

1. O comando de localização do apontador deve ser sempre utilizado antes do que qualquer comando de leitura, ou de escrita; caso contrário este comando os cancelaria por se tratar de um comando TKDOS (vide os itens anteriores)
2. Antes de utilizarmos o comando POSITION devemos ativá-lo através do comando OPEN que localizará o apontador no início, deixando inicialmente a área toda do arquivo à disposição do POSITION. Note também que se o arquivo fosse ativado com um comando APPEND, o comando POSITION perderia a validade em alguns casos já que este não movimenta o apontador para trás.

## 4. ARQUIVOS DE ACESSO RANDÔMICO (ALEATÓRIO)

Os arquivos de acesso randômico tem um comportamento muito mais flexível que os de acesso seqüencial. A principal diferença é que neste caso não há necessidade de acessar o arquivo inteiro para alcançar qualquer ponto em seu interior como no seqüencial. Neste tipo de arquivo podemos acessar qualquer campo do arquivo independente de sua posição.

O arquivo de acesso randômico, assim como o arquivo seqüencial, é composto por vários grupos de dados também chamados registros.

O arquivo de acesso randômico, inicialmente é formado apenas por um registro, mas quando o aumento de informações ultrapassa o comprimento definido do registro automaticamente é formado um outro registro cuja identificação se dará por um valor inteiro maior em uma unidade que o registro anterior. Assim o primeiro registro do arquivo receberá o número 0, o segundo receberá o número um, o terceiro receberá dois, etc...

Deverá ser observado que, os arquivos do tipo randômico tendem a ocupar mais espaço no disco, do que um arquivo seqüencial.

### 4.1. Ativação de Arquivos de Acesso Randômico

Para se ativar um arquivo de acesso randômico, o procedimento é semelhante ao do arquivo seqüencial, com a diferença que neste caso, temos que obrigatoriamente definir o comprimento de cada registro através do parâmetro L.

```
>10 LET D$ = CHR$(4) : REM CONTROL-D  
>20 PRINT D$ "OPEN ARQ1,L20"
```

O parâmetro L deve variar entre 1 e 32767. É importante que este parâmetro seja bem calculado, pois se for inserido algum registro (incluindo vírgulas

e o carriage return) de tamanho maior que o especificado, ocorrerão conseqüências imprevisíveis no armazenamento das informações.

Se, a ler um arquivo, você especificar um parâmetro (L) diferente daquele que foi utilizado para gravá-lo, o TKDOS irá utilizá-lo cegamente para calcular as posições dos registros e os dados lidos não corresponderão aos originalmente gravados.

Para fechar um arquivo de acesso randômico usa-se o comando CLOSE do mesmo modo que no sistema de arquivos seqüenciais.

## 4.2. Lendo e Escrevendo em Arquivos de Acesso Randômico

Os arquivos de acesso randômico operam de modo semelhante aos de acesso seqüencial, somente que neste caso o parâmetro registro deve, obrigatoriamente, acompanhar os comandos READ e WRITE. Este parâmetro (R), como já foi dito, representa o registro que desejamos acessar.

```
>30 PRINT D$ "READ ARQ1, R3"
```

ou

```
>30 PRINT D$ "WRITE ARQ1, R3"
```

O último parâmetro a ser descrito é o BYTE (B). Ele pode acompanhar os comandos READ, WRITE E POSITION (logo, é também utilizável em arquivos seqüenciais).

O parâmetro BYTE move o apontador para um byte específico em um registro já selecionado. No exemplo abaixo será fácil determinar sua utilidade:

```
READ ARQ1, R13, B7
```

Neste comando, a leitura iniciará no registro 13 a partir do byte 7.

Este comando, dentro de um mesmo registro, pode movimentar o apontador para frente como também para trás.

A seguir, é apresentado um pequeno exemplo de utilização de arquivos de acesso randômicos:

```
>10 REM CRIA LISTA DE NOMES
>20 D$ = CHR$(4) : REM CONTROL-D
>30 INPUT "NOME: ",NO$
>40 INPUT "END.: ";EN$ : REM COM OS DADOS
>50 INPUT "TEL.: ";TE$
>60 PRINT D$"OPEN LISTA, L150"
>70 PRINT D$"WRITE LISTA, R0"
>80 PRINT NO$ : PRINT EN$ : PRINT TE$
>90 PRINT D$"CLOSE LISTA"
```

Para ler os dados guardados no arquivo — lista, basta seguir o programa abaixo:

```
>10 REM LE OS DADOS DA LISTA
>20 D$ = CHR$(4) : REM CONTROL-D
>30 PRINT D$"OPEN LISTA, L150"
>40 PRINT D$"READ LISTA, R0"
>50 INPUT NO$, EN$, TE$
>55 PRINT : REM PULA UMA LINHA
>60 PRINT NO$
>70 PRINT EN$
>80 PRINT TE$
>90 PRINT D$"CLOSE LISTA"
```

Se quisermos montar uma lista com vários nomes, endereços e telefones associados entre si (como um caderno de telefones), basta associar-se a cada registro uma pessoa com seu endereço e telefone.

# 5. COMANDOS AUXILIARES TKDOS 3.3

## 5.1. Comando EXEC

O poderoso comando EXEC é muito semelhante ao comando RUN, e é relacionado a arquivos seqüenciais que, ao invés de simples informações ou dados, contém comandos ou linhas de programa BASIC, inseridas no arquivo. O EXEC permite que um pseudo-programa contido no arquivo texto seja executado, em um nível de controle hierarquicamente superior ao de um programa executado por RUN.

A seqüência abaixo explicará melhor este comando:

Supondo que você queira fazer um programa que execute as seguintes funções:

- Listar o programa TESTE.
- Paralizar a execução por alguns segundos.
- Fornecer o CATALOG do disco.
- Rodar o programa.

Inicialmente devemos criar e salvar o programa TESTE. Este programa poderia ser assim:

```
>NEW
>10 REM PROGRAMA TESTE
>30 HOME
>40 FOR I = 1 TO 100
>50 K = I * 2
>60 PRINT I,K
>70 NEXT I
>80 FOR L = 1 TO 1000 : NEXT L
>90 END
>SAVE TESTE
```

Quando a luz do disk drive se apagar, o programa TESTE já estará gravado no disco.

O seguinte programa, ao qual chamamos de CRIA AUTOMATO, criará o arquivo EXEC.

Programa AUTOMATO:

```
>NEW
>10 REM PROGRAMA CRIA AUTOMATO
>20 D$ = CHR$(4) : REM CONTROL-D
>30 PRINT D$ "OPEN AUTOMATO"
>40 PRINT D$ "WRITE AUTOMATO"
>45 PRINT "NEW"
>50 PRINT "LOAD TESTE"
>55 PRINT "HOME"
>60 PRINT "LIST"
>70 PRINT "100 FOR I = 1 TO 10000"
>80 PRINT "120 NEXT I : END"
>90 PRINT "RUN 100"
>100 PRINT "CATALOG"
>105 PRINT "RUN"
>110 PRINT D$ "CLOSE AUTOMATO"
>SAVE CRIA AUTOMATO
```

Seqüência de instruções que serão usadas pelo EXEC

Digitando-se então:

```
>RUN CRIA AUTOMATO
```

o arquivo texto AUTOMATO será criado em seu disquete. Para fazer funcionar o comando EXEC, basta digitar:

```
>EXEC AUTOMATO
```

e veja a seqüência.

## 5.2. Maxfiles

No modo normal, o TK-2000 ativa até 3 arquivos simultaneamente. Muitas vezes esta quantidade não é suficiente para as nossas necessidades. Para contornar este problema existe o comando MAXFILES, que permite alterar este número para até 16 arquivos simultâneos. O formato deste comando é:

```
PRINT D$ "MAXFILES 8"
```

deste modo, agora serão permitidos que até 8 arquivos sejam ativados de uma só vez.

Convém avisar que este comando quando necessário, deve sempre estar logo no início do programa, caso contrário, muitas funções lógicas do BASIC poderão não funcionar adequadamente.

### 5.3. MON e NOMON

MON e NOMON são importantes ferramentas do TKDOS, que permitirão depurar programas, isto é, acompanhar seu comportamento enquanto estes estão "rodando"; maneira muito utilizada para se encontrar erros de lógica em programas.

O comando MON permite que sejam monitoradas uma série de informações. Ele pode ser desativado pelo comando NOMON, restabelecendo-se a operação normal do computador. Para este comando existem três parâmetros diferentes e independentes que podem ser aplicados:

- C = monitora todos os comandos de disco (OPEN, READ, etc..)
- I = monitora os comandos de entrada do disco (ler um arquivo - READ, GET)
- O = monitora os comandos de saída do disco (escrever no arquivo - WRITE)

Estes parâmetros são exclusivos dos comandos MON e NOMON.

O comando MON pode também ser desativado pela instrução DSK (ativa o disk drive, carregando novamente o TKDOS) e pelo comando RESET. Por exemplo:

```
>MON C,I,O
```

### 5.4. TRACE

O comando TRACE, é uma ferramenta muito útil na depuração de programas BASIC. Quando um programa contém comandos TKDOS e o TRACE é ativado; os comandos TKDOS no interior do programa não funcionarão, já que TRACE imprime o número da linha antes do comando TKDOS. Uma solução para este problema é inserir um carriage return (CHR\$(13)) antes do CONTROL-D, na definição de D\$. Por exemplo:

10 D\$ = CHR\$(13) + CHR\$(4)

Assim, a maioria dos comandos TKDOS poderão operar normalmente com o TRACE ativado. A exceção corresponde a leitura de disco através de INPUT sendo que neste caso o programa se deterá aguardando uma entrada pelo teclado.

# 6. ARQUIVOS EM LINGUAGEM DE MÁQUINA

O TK-2000 permite além do que já foi apresentado, que se possa armazenar em disquete regiões de memória definidas. Contida nesta parte da memória podem existir programas em linguagem de máquina, telas de alta resolução ou qualquer outra informação.

## 6.1. BSAVE

Este comando armazena no disco um arquivo de nome "k", situado em uma faixa de memória e seu formato é:

**BSAVE k, Aa, Lj**

onde o parâmetro A indica o endereço inicial da porção da memória a ser armazenada no disco; este endereço, se precedido do sinal dólar (\$) significa o endereço em hexadecimal, caso contrário, em decimal.

O parâmetro L indica o comprimento em bytes, do arquivo a ser armazenado. Da mesma forma que o parâmetro A, se o valor numérico do comprimento vier precedido do sinal (\$), então indicará que o parâmetro está apresentado em hexadecimal.

Exemplo:

**>BSAVE VIDEO1, A\$2000,L\$2000**

## 6.2. BLOAD

O comando BLOAD carrega o conteúdo de um arquivo do disco para a memória de seu TK-2000. Os programas BASIC não serão afetados com o carregamento, a menos que os dados sejam carregados na área do BASIC do TK-2000.

O formato do comando BLOAD é:

**BLOAD k,[Aa]**

Note que no caso de BLOAD, não é necessário acrescentar nenhum parâmetro como ocorreu em BSAVE. Isto ocorre porque o computador irá reconhecer o arquivo e, então, carregá-lo exatamente no mesmo endereço em que está quando foi armazenado no disco. Caso se utilize o parâmetro A (endereço) opcionalmente no comando BLOAD, o arquivo então será carregado a partir do endereço especificado por A. Exemplo:

**>BLOAD UTIL, A\$300**

## 6.3. BRUN

O comando BRUN segue exatamente o comportamento de BLOAD, explicado no item anterior, somente que, após ter carregado o programa em linguagem de máquina este faz com que o computador inicie sua execução.

O formato do comando BRUN é:

**BRUN k**

Note que o formato de BRUN obedecerá as mesmas regras de BLOAD, inclusive quanto ao parâmetro opcional A, o qual, se utilizado, carregaria o programa no endereço especificado provocando um salto (JMP), para o endereço inicial do bloco.

## APÊNDICE A — MENSAGENS DO TKDOS

Ao ocorrer um erro reconhecido pelo TKDOS, se o processamento for desviado por um comando ONERR GOTO, o código de erro poderá ser obtido através de PEEK(222).

### LINGUAGEM INEXISTENTE

Código = 1

O TKDOS sempre emitirá esta mensagem de erro quando for tentado processar um programa originalmente produzido em outra linguagem que não o APPLESOFT BASIC.

### INDICE ERRO

Código = 2 ou 3

Esta mensagem de erro é emitida quando um dos parâmetros do TKDOS ou um próprio comando TKDOS apresentar valores fora da faixa permitida.

Por exemplo:

```
>10 PRINT CHR$(4) "OPEN ARQUIVO, V344"
```

sendo que o maior valor permitido para o parâmetro Volume é 254.

### DISCO PROTEGIDO

Código = 4

Esta mensagem ocorrerá sempre que for tentado gravar em um disco, com o selo de proteção no lugar. A única saída, neste caso, é de remover o selo e repetir a operação.

### FIM ARQUIVO

Código = 5

Geralmente esta situação ocorre quando, na leitura de um arquivo, chega-se ao final deste, e o processador tenta obter mais informação.

### NAO HA ARQUIVO

Código = 6

Ocorre quando é tentado acessar algum arquivo inexistente no disco.

### VOLUME ILEGAL

Código = 7

Ocorre quando é acessado um arquivo-texto utilizando-se o parâmetro V (Volume) e este parâmetro não confere com o valor registrado no momento da gravação deste arquivo.

### I/O ERRO

Código = 8

Ocorre nas seguintes situações:

1. O disquete não está colocado no disk drive ou a porta deste não está fechada.
2. O disquete não está devidamente formatado.
3. Existe alguma irregularidade na gravação do programa no disquete decorrida ou por dano existente no disco ou por irregularidade ocorrida no processo de gravação/leitura.

### <LOTADO>

Código = 9

Ocorre quando for tentado gravar algum programa ou arquivo em um disco que não contenha espaço disponível para receber esta informação adicional. Neste caso a melhor saída seria procurar outro disco com maior área disponível ou deletar algum arquivo sem importância contido no disco aumentando assim a área livre.

O programa utilitário TKFID que já vem contido no disco MESTRE permite avaliar a área livre em um disco evitando que erros como este possam ocorrer.

### <TRAVADO>

Código = 10

Ocorre quando é tentado escrever em um arquivo ou alterar um programa que foi protegido através da instrução LOCK. Neste caso, a única saída é de gravar esta informação com outro nome ou senão utilizar a instrução UNLOCK sobre o arquivo ou programa permitindo que haja esta nova alteração.

#### SINTAXE #ERRO

Código = 11

Ocorre ao existir um erro na sintaxe de algum comando TKDOS, por exemplo:

```
> 15 PRINT CHR$(4) "OPEM ARQUIVO"
```

#### SEM BUFF. DISPONIVEL

Código = 12

Ocorre quando é tentado abrir um número maior de arquivos simultaneamente do que o permitido. A única saída para este caso é utilizar o comando MAXFILES para expandir a capacidade do sistema.

#### ARQ. INCOMPATIVEL

Código = 13

Ocorre quando é tentado carregar um programa utilizando-se um comando incompatível com a classe de arquivo. Por exemplo, se tentarmos carregar o programa em BASIC "TESTE" com o comando destinado a leitura de programa em linguagem de máquina.

```
>BLOAD TESTE <RETURN>
```

#### EXCEDE MEMORIA

Código = 14

Acontece ao se carregar um programa que necessite uma memória superior à disponível no seu TK-2000. Geralmente ocorre quando a capacidade disponível foi alterada através do comando HIMEM:

#### COMANDO ILEGAL

Código = 15

Este erro será indicado ao se tentar utilizar um comando TKDOS APPEND, OPEN, POSITION, READ ou WRITE através do modo direto. Estes comandos TKDOS devem ser usados somente através do comando PRINT nas linhas do programa.

## APÊNDICE B — SUMÁRIO DOS PROCEDIMENTOS DO DOS

### CARREGANDO O DOS

Quando no modo BASIC, digite DSK, seguido por RETURN.

### INICIALIZANDO O DISQUETE

Para inicializar um disquete virgem basta:

- Carregar o TKDOS 3.3
- Colocar o disquete virgem no disk drive
- Digitar o programa no qual se deseja apresentar o disquete, por exemplo:

```
5 REM APRESENTA O DISCO E SEU CATALOGO
10 D$ = CHR$(4) : REM CONTROL - D
20 HOME
30 PRINT "DISQUETE NUMERO 1"
40 PRINT D$ "CATALOG"
```

- Considerando que você tenha denominado o programa acima pelo nome ALO, digite então o comando:

```
>INIT ALO
```

- Após a luz do disk drive se apagar, o seu disquete já estará inicializado, agora o melhor passo seria retirar o disquete do drive e identificá-lo com uma etiqueta.

### UTILIZANDO COMANDOS DO TKDOS DENTRO DE UM PROGRAMA

Os comandos TKDOS podem ser executados dentro de programas em BASIC, através do comando PRINT, acompanhados de um CONTROL-D. Pode-se também, definir o caractere CONTROL-D, já no início de programa, através da declaração:

```
D$ = CHR$(4)
```

Os comandos abaixo, obrigatoriamente, deverão ser usados no modo de execução indireta (dentro de um programa), ou seja, dentro de um comando PRINT e precedido do caractere CONTROL-D:

```
OPEN
READ
APPEND
WRITE
POSITION
```

### MUDANDO O PROGRAMA APRESENTADOR DO DISQUETE

Caso se queira mudar o programa apresentador de um determinado disquete, basta escrever ou carregar o novo programa apresentador e a seguir, armazená-lo com o mesmo nome do programa que foi utilizado anteriormente.

### CRIANDO E ACESSANDO ARQUIVOS SEQUENCIAIS

Na criação ou ativação de um arquivo seqüencial, o comando OPEN deve sempre ocorrer antes do comando WRITE; após o WRITE, toda instrução PRINT enviará todos os caracteres ao disquete. O comando WRITE só será cancelado, através de uma instrução INPUT, ou através de uma outra instrução TKDOS como, por exemplo, uma simples instrução neutra.

```
PRINT CHR$(4)
```

Após um arquivo ser escrito, este deverá ser desativado com o comando CLOSE, sob o risco de perder todos os dados contidos no arquivo. A seguir, é apresentado um exemplo de ativação e escrita de um arquivo seqüencial:

```
>10 REM PROGRAMA PRIMEIRO
>20 D$ = CHR$ : REM DEFININDO O CONTROL-D
>30 REM CERTIFICANDO QUE O ARQUIVO ESTEJA APAGADO
>40 PRINT D$ "OPEN PRIMEIRO"
>50 PRINT D$ "DELETE PRIMEIRO"
>60 REM INAUGURANDO UM NOVO ARQUIVO
>70 PRINT D$ "OPEN PRIMEIRO"
>80 PRINT D$ "WRITE PRIMEIRO"
>90 PRINT "O TRANSPORTE DE DADOS"
>100 PRINT "VAI COMECAR"
>110 FOR B = 0 TO 100 STEP 2
>120 PRINT B
>130 NEXT
>140 REM FECHANDO O ARQUIVO
>150 PRINT D$ "CLOSE PRIMEIRO"
```

Ao ativar-se um arquivo que já existe, os dados após o comando WRITE, irão cobrir os dados já existentes. Para que os dados comecem a entrar diretamente ao final do arquivo, basta abrir o programa diretamente com o comando APPEND no lugar de OPEN.

A seguir, será acessado o programa exemplo "PRIMEIRO", definido acima. Caso se queira ver o que está sendo lido no disquete, pode-se utilizar o comando:

```
> MON I
> 10 REM ACESSANDO O PROGRAMA PRIMEIRO
> 15 DIM P(60)
> 20 D$ = CHR$(4)
> 30 REM ABRINDO O PROGRAMA PRIMEIRO
> 40 PRINT D$"OPEN PRIMEIRO"
> 50 PRINT D$"READ PRIMEIRO"
> 60 REM LENDO INICIALMENTE OS REGISTROS DE CADEIAS
> 70 INPUT R1$, R2$, R3$
> 80 FOR I = 1 TO 51
> 90 INPUT P (I)
> 100 NEXT
> 110 REM FECHANDO O ARQUIVO PRIMEIRO
> 120 PRINT D$"CLOSE PRIMEIRO"
```

O comando READ deverá sempre estar precedido pelo comando de abertura do arquivo, e só é cancelado após encontrar o caractere CONTROL-D dentro de uma declaração PRINT, existindo ou não, comando DOS no comando.

### ACRESCENTANDO DADOS A UM ARQUIVO SEQÜENCIAL

Para acrescentar dados em um arquivo seqüencial, basta ao invés do comando OPEN, ativar o arquivo com o comando APPEND que, deste modo transportará o apontador de arquivo ao final deste.

Com o apontador de arquivo no final os dados acrescentados através de comando WRITE irão se acumulando ao arquivo. A seguir é apresentado um exemplo desta operação.

```
> 10 REM INCREMENTANDO DADOS NO ARQUIVO PRIMEIRO
> 20 D$ = CHR$(4) : REM CONTROL-D
> 30 REM ABRINDO O ARQUIVO, E LEVANDO O APONTADOR PARA O FINAL
> 40 PRINT D$"APPEND PRIMEIRO"
> 50 PRINT D$"WRITE PRIMEIRO"
> 60 PRINT "FIM DA PARTE 1"
> 70 REM FECHANDO O ARQUIVO
> 80 PRINT D$"CLOSE PRIMEIRO"
```

### UTILIZANDO UM ARQUIVO SEQÜENCIAL COMO UM PROGRAMA BASIC

Ao ser executado, o programa seguinte cria um arquivo-texto de nome AUTOMATICO, que conterà os comandos

```
LIST
RUN ALO
CATALOG
```

```
> 10 REM PROGRAMA CRIADOR DO AUTOMATICO
> 20 D$ = CHR$(4)
> 30 PRINT D$"OPEN AUTOMATICO"
> 40 PRINT D$"WRITE AUTOMATICO"
> 50 PRINT "LIST"
> 60 PRINT "RUN ALO"
> 70 PRINT "CATALOG"
> 80 REM FECHANDO O ARQUIVO AUTOMATICO
> 90 PRINT D$"CLOSE AUTOMATICO"
> RUN
```

Programa destinado ao EXEC

Ao funcionar o programa CRIADOR DO AUTOMATICO, o arquivo-texto AUTOMATICO é criado no disco. Para que o conteúdo deste arquivo seja executado, basta digitar:

```
> EXEC AUTOMATICO
```

e veja os resultados.

### CRIANDO E ACESSANDO ARQUIVOS DE ACESSO RANDÔMICO

No exemplo abaixo, é criado um arquivo de acesso aleatório de nome SEGUNDO, sendo que seus registros serão limitados a 50 caracteres cada. Na seqüência, o programa grava nos registros 3 a 20, o string "REGISTRO NUMERO", seguido pelo número do registro. Finalmente, o programa alterará os registros 15 e 16 para conter somente o string "MICRODIGITAL".

```

>10 REM ARQUIVO SEGUNDO
>20 D$ = CHR$(4) : REM CONTROL-D
>30 REM APAGA O ARQUIVO SEGUNDO, SE PORVENTURA EXISTIR
>40 PRINT D$"OPEN SEGUNDO"
>50 PRINT D$"DELETE SEGUNDO"
>60 PRINT D$"OPEN SEGUNDO,L50"
>70 FOR J = 3 TO 20
>80 PRINT D$"WRITE SEGUNDO, R" ; J
>90 PRINT "REGISTRO NUMERO" ; J
>100 NEXT
>110 FOR J = 15 TO 16
>120 PRINT D$"WRITE SEGUNDO,R" ; J
>130 PRINT "MICRODIGITAL"
>140 NEXT
>150 PRINT D$"CLOSE SEGUNDO"

```

Note que, nas linhas 80 e 120 do programa acima, foi utilizado o ponto e vírgula para unir o indicador do parâmetro R com o índice representado por J. Esta é uma prática muito usual para esta espécie de arquivo.

O programa a seguir irá verificar o arquivo formado pelo programa acima do terceiro ao vigésimo registro. Quando este verificar que os cinco primeiros caracteres do arquivo forem MICRO, surgirá uma mensagem "HOUE UMA ALTERACAO NO REGISTO NUMERO s DO ARQUIVO", sendo "s" o registro alterado, após a mensagem a procura continuará.

```

>10 REM ACESSANDO O ARQUIVO SEGUNDO
>20 D$ = CHR$(4) : REM CONTROL-D
>30 PRINT D$"OPEN SEGUNDO, L50"
>40 FOR K = 3 TO 20
>50 PRINT D$"READ SEGUNDO, R" ; K
>60 INPUT T$
>70 IF LEFT$(T$,5) = "MICRO" THEN PRINT "HOUE UMA
ALTERACAO NO REGISTO NUMERO ";K;" DO ARQUIVO"
>80 NEXT K
>90 PRINT D$"CLOSE SEGUNDO"

```

## APÊNDICE C — DESCRIÇÃO DOS PROGRAMAS UTILITÁRIOS

### TKFID

O TKFID é um utilitário para uso geral em disco.

Para executá-lo digite: "BRUN TKFID"

#### 1 — Funções:

O TKFID possui as seguintes funções:

- <1> — Cópia — Copia arquivos
- <2> — Catalog — Cataloga todos os arquivos do disquete
- <3> — Espaço — Fornece o número de bytes ocupados e os disponíveis no disquete.
- <4> — Destrava — Destrava o arquivo
- <5> — Trava — Trava o arquivo
- <6> — Verifica — Verifica a integridade dos arquivos.
- <7> — Fim — Retorna ao BASIC

#### 2 — Nomes de arquivos:

Para fornecer nomes de arquivos você pode entrar o nome completo dos mesmos ou ainda utilizar a eficiente rotina de busca disponível neste programa.

O símbolo "=" serve para informar ao programa, durante a seleção, que qualquer caractere naquela posição é aceitável. Veja no exemplo abaixo algumas configurações possíveis.

EXEMPLO:

=XX= — Serão relacionados todos os arquivos que contenham o string "X" em qualquer posição de seu nome.

XX= — serão relacionados todos os arquivos que tenham o seu nome iniciado por XX.

=XX — serão relacionados todos os arquivos que tenham o seu nome finalizado por XX.

3 — **Controle Manual:** Ao ser perguntado sobre o controle manual, responda:

— "S" — assume controle manual em que após cada nome de arquivo deve ser digitado "S" para processar ou "N" para cancelar.

— "N" — assume que todos os arquivos selecionados serão processados.

## CHAIN

Este utilitário permite o encadeamento de programas sem a perda das variáveis em memória, possibilitando a um novo programa carregado utilizar estas variáveis.

Para encadear, insira as seguintes linhas no programa que efetuará o encadeamento:

```
>xxx PRINT CHR$(4) "BLOAD CHAIN,A520"  
>xxx CALL 520"nome do programa a ser carregado"
```

Observações:

- 1 — Insira as linhas uma após a outra.
- 2 — Não deixe "brancos" entre o "CALL 520" e as aspas.
- 3 — Copie o programa "CHAIN" do discc MESTRE para seu disquete de programas, através do utilitário TKFID.

## CRIA TEXTO

Este utilitário cria arquivos-texto seqüenciais.

Para executá-lo, digite:

## >RUN CRIA TEXTO (RETURN)

Os registros serão inseridos através de "linhas", sendo que cada uma representa um registro. Cada "linha" pode conter um máximo de 239 caracteres, sendo que após o seu preenchimento deve-se pressionar a tecla RETURN. Após pressionar RETURN duas vezes, será pedido o nome do arquivo a ser gravado.

## LE TEXTO

Este utilitário serve para recuperar arquivos-texto gravados pelo programa CRIA-TEXTO.

Para executá-lo, digite:

## >RUN LE TEXTO (RETURN)

Em seguida, introduza o nome do arquivo-texto desejado, pressionando-se em seguida a tecla RETURN.

## MINIASM

Este utilitário permite montar programas em linguagem de máquina através dos mneumônicos 6502. Digite "BRUN MINIASM" para carregar o programa.

Para iniciar sua operação digite:

```
>TK2000 RETURN em BASIC ou,  
03F8G RETURN no modo anterior.
```

O modo de operação é idêntico ao descrito no manual técnico para o mini-assembler da ROM.

Observe que não deve ser usado o comando ASS quando esteja instalada a interface de disco.

## COPIA

Utilitário copiador de disquetes. Digite "RUN COPIA" para executar o programa. Verifique previamente se o disquete que receberá a cópia não

possue arquivos ou informações que não possam ser apagados, pois o disco será novamente inicializado.

## RENUMERADOR

Utilitário que realiza renumeração de linhas e/ou MERGE de programas BASIC.

Digite "RUN RENUMERADOR INSTRUcoes" para instruções, ou "RUN RENUMERADOR" para executar o programa.

## DEMO CORES

Este utilitário gera efeitos visuais coloridos, em alta resolução para demonstração dos recursos gráficos do TK-2000 COLOR.

Para executar o programa, use o comando:

>RUN DEMO CORES

Os gráficos são gerados aleatoriamente, oferecendo um padrão de cores bastante diversificado.

O espaçamento entre as linhas impressas no vídeo (passo), também é randômico, sendo especificado abaixo da figura gerada.

A tecla espaço interrompe a execução do programa.

## OTHELLO

Instruções:

Este é um jogo de estratégia que se desenvolve num tabuleiro de 8x8 posições cujo objetivo é possuir o maior número de peças que o computador.

Para executar o programa digite "RUN OTHELLO"

Para ganhar as peças adversárias, você deve aprisioná-las em uma das três posições (horizontal, vertical ou diagonal) entre duas peças suas.

A tecla R, ao ser pressionada, reinicia a partida e a tecla P interrompe o jogo informando o vencedor.

## LISTA TELEFÔNICA

Instruções:

Este programa demonstra o uso do computador em arquivo pessoal de telefones.

Para executá-lo digite:

>RUN LISTA TELEFONICA <RETURN>

O programa apresenta 7 opções:

- 1 - Consulta de registros
- 2 - Listagem dos registros
- 3 - Entrada de novos registros
- 4 - Exclusão de registros
- 5 - Alteração de registros
- 6 - Impressora ligada/desligada
- 7 - Final do programa

## EDITOR

O Editor é um programa escrito em linguagem de máquina, que auxilia nos trabalhos de edição, normalmente em programação BASIC.

Para carregá-lo, com o TKDOS ativo, digite "BRUN EDITOR", e RETURN. Aguarde até aparecer novamente o cursor.

Agora você poderá editar programas BASIC ou outros comandos digitados diretamente pelo teclado.

O processo adotado para a edição, neste programa, é do tipo TELA, o qual exige que o texto a ser editado esteja constando na tela do televisor (ou monitor de vídeo) no momento da edição.

A edição é efetuada em duas etapas. A primeira delas terá como objetivo alcançar o ponto, na tela, onde se encontra o texto a ser editado. Para isso o cursor deverá ser conduzido usando as teclas de movimentação neutra, explicadas mais adiante.

A segunda etapa, com o cursor posicionado, o texto deverá ser "digitado" utilizando-se das teclas de movimentação horizontal (→) e (←), ao mesmo tempo em que se corrige o texto existente.

As teclas de movimentação neutra são:

(CONTROL-Z) — habilita as teclas seguintes a trabalhar no modo de movimentação neutra;

- A — leva o cursor para direita, só 1 posição;
- B — leva o cursor para esquerda, só 1 posição;
- C — leva o cursor para baixo, só 1 posição;
- D — leva o cursor para cima, só 1 posição;
  
- I — leva o cursor para cima, com repetição;
- J — leva o cursor para a esquerda, com repetição;
- K — leva o cursor para direita, com repetição;
- M — leva o cursor para baixo, com repetição.

Após o CONTROL-Z, a digitação de qualquer tecla que não as indicadas acima, anulará o modo de movimentação neutra, portanto se for necessário outra movimentação, deverá ser novamente aplicado um CONTROL-Z.

O processo de movimentação do cursor é efetuado com a tecla →, ao "passar por cima" de um carácter na tela, equivalerá a "redigitar" o dito carácter, enquanto que a tecla ← "apaga" o carácter.

Usaremos daqui por diante as expressões "mova o cursor", querendo indicar o uso das teclas de movimentação neutra, "redigite" para usar a tecla →, e "apague" para usar a tecla ←. Todo processo de edição é finalizado pressionando-se RETURN.

Exemplo: Digite textualmente o seguinte programa:

```
>10 PRINT "MEU NOME E";  
>20 PRINT "JOAP"  
>
```

Percebendo dois erros de sintaxe, um em cada instrução, você irá corrigi-los.

Usando inicialmente CONTROL-Z, "mova" o cursor até o 1 da linha 10. "Redigite" a linha até o cursor ficar em cima da letra M, digite N, e continue "redigitando" até o final da linha, ultrapassando o ponto e vírgula. Pressione RETURN.

Efetue um LIST e você verá que a linha 10 foi corrigida. Agora, mova o cursor até ficar em cima do 2 da linha 20, e redigite até o carácter "P", digite no lugar o carácter "O", e redigite até o fim da linha. Pressione RETURN e efetue um novo LIST para conferir.

Observe que a tecla de redigitação permite copiar textos que aparecem na tela. Suponhamos que você queira incrementar a seguinte linha:

```
30 PRINT "MEU NOME E JOAO"
```

Em vez de digitar esta linha diretamente, usaremos as funções de edição.

Efetue um LIST, digite 30, acione CONTROL-Z e mova o cursor até a linha 10 no início da palavra PRINT, redigite até as aspas finais. Acione CONTROL-Z e mova o cursor até a letra J da linha 20, redigite até o fim da linha e pressione RETURN. Liste agora o programa.

Para inserir um texto dentro de uma linha de programa deve-se proceder como no exemplo adiante. Nele iremos inserir TAB(10), redigite até após a palavra PRINT, "mova" o cursor para cima 1 posição, digite "TAB(10)"; mova o cursor novamente para linha 30 para a mesma posição de onde o cursor foi deslocado. Dai, redigite até o fim da linha. Pressione RETURN e confira o programa.

Existem ainda, outras funções auxiliares habilitadas também por CONTROL-Z usando as seguintes teclas:

E — Apaga a linha, desde onde se encontra o cursor até o final da mesma

F — Apaga a tela desde onde a linha em que o cursor se encontra até o final.

Use CONTROL-T <RETURN> ao invés de HOME para limpar a tela sempre que o programa EDITOR estiver ativado.

## IMP01

Software para impressora. Para carregá-lo do disco MESTRE, digite "BRUN IMP01".

O modo de utilização é descrito no manual que acompanha o kit de impressora.

## MÓDULO DE REFERÊNCIA — TKDOS

Neste cartão, os comandos TKDOS estão agrupados em 5 categorias:

— Comandos próprios do TKDOS:

INIT	LOAD	DELETE	VERIFY	MAXFILES
CATALOG	RUN	LOCK	MON	SAVE
RENAME	UNLOCK	NOMON		

— Comandos de acesso:

DSK

— Comandos de arquivos-texto sequenciais:

OPEN	READ	APPEND	EXEC
CLOSE	WRITE	POSITION	

— Comandos de arquivos-texto aleatórios:

OPEN	CLOSE	READ	WRITE
------	-------	------	-------

— Comandos de arquivos em linguagem de máquina.

BLOAD	BRUN	BSAVE
-------	------	-------

### NOTAÇÃO E SINTAXE

Os parâmetros serão representados por letras maiúsculas, normalmente acompanhadas por um número (representado por letras minúsculas), que oferece informações adicionais para executar um comando. Os parâmetros múltiplos podem aparecer em qualquer ordem, mas devem estar separados um do outro por vírgulas. Os parâmetros apresentados entre colchetes [como estes] são opcionais.

Um nome-de-arquivo (apresentado aqui por uma letra X) devem estar imediatamente após uma palavra de controle. Estes nomes devem obrigatoriamente começar com uma letra; somente os 30 primeiros caracteres serão reconhecidos como nome-de-arquivo. Uma vírgula deverá separar o nome dos seus parâmetros.

### Comandos TKDOS

Parâmetro	Apresentação	Min	Max
MAXFILES	MAXFILES n	n=1	n=16

Se um comando omitir o parâmetro Volume ou utilizar V0, o número do volume do disquete é ignorado. Um comando que utilizar o parâmetro Volume Vv não irá executar a menos que o número de volume seja v.

### Comandos próprios do TKDOS

#### INIT X [,Vv]

Inicializa um disquete virgem para o formato de disco auxiliar. Assume como programa apresentador o programa de nome X e número de volume v (se especificado).

#### CATALOG

Apresenta o número de volume e todos os arquivos e programas contidos no disquete com seus respectivos tipos e comprimentos (número de setores utilizados). O símbolo \* indica um arquivo travado.

TIPO	DESCRIÇÃO
I	INTEGER BASIC (não roda no TK-2000)
A	Programa BASIC tipo APPLESOFT
T	Arquivo-texto
B	Programa em linguagem de máquina

#### SAVE X [,Vv]

Grava o programa BASIC contido na memória no disquete com o nome X. Escreve sobre qualquer arquivo anterior que seja do mesmo tipo e com o mesmo nome sem fornecer qualquer aviso.

#### LOAD X [,Vv]

Carrega para a memória um programa BASIC contido no disco, depois de limpá-la do programa anterior.

#### RUN X [,Vv]

Carrega o programa BASIC com o nome X executando o mesmo a seguir.

#### RENAME X,Y [,Vv]

Troca o nome do arquivo X para Y.

**DELETE X [,Vv]**

Apaga o arquivo X do disquete.

**LOCK X [,Vv]**

Trava o arquivo X contra uma mudança ou apagamento accidental. Este travamento pode ser observado no CATALOG através do \*.

**UNLOCK X [,Vv]**

Destrava o arquivo X (caso este já estivesse travado) permitindo mudança ou deleção do mesmo.

**VERIFY X [,Vv]**

Checa o arquivo X em sua consistência interna. Se X não possuir nenhum erro, não será fornecido nenhuma mensagem.

**MON [,C] [,I] [,O]**

Causa a apresentação dos comandos de disco (C), instruções de INPUT no disco (I), e instruções de OUTPUT para o disco (O). Se nenhum destes parâmetros acompanhar o comando, este será ignorado.

**NOMON [,C] [,I] [,O]**

Cancela a apresentação dos comandos de disco (C), instruções de INPUT no disco (I), e instruções de OUTPUT para o disco (O). Se nenhum destes parâmetros acompanhar o comando, este será ignorado.

**MAXFILES n**

Reserva n buffers de arquivo para a entrada e saída de disco (normalmente são reservados apenas 3 buffers de arquivo). Este comando deve ser usado sempre antes de carregar ou rodar um programa.

**COMANDO DE ACESSO****DSK**

Acessa o disk drive introduzindo o TKDOS no sistema. Caso o TKDOS já estava contido na memória, os programa BASIC contido nela será apagado entrando em seu lugar o programa apresentador do novo disco.

**COMANDOS DE ARQUIVOS-TEXTO SEQUENCIAIS****OPEN X [,Vv]**

Abre ou cria um arquivo seqüencial com o nome X, atribuindo a ele um buffer,

e preparando para receber os comandos WRITE ou READ para o início do arquivo.

**CLOSE [X]**

Completa o comando WRITE X, se necessário, e desatribui o buffer reservado para o arquivo-texto X. Sem o nome do arquivo, este comando irá fechar todos os arquivos que eventualmente estejam abertos nesta situação (exceto ou arquivo EXEC).

**WRITE X [,Bb]**

Permite que a instrução PRINT envie caracteres para o arquivo-texto seqüencial X. A escrita inicia-se no início do arquivo ou (se especificado) pelo byte b. Este comando é cancelado por qualquer comando TKDOS posterior a ele.

**READ X [,Bb]**

Permite que a instrução INPUT leia caracteres do arquivo-texto X. A leitura inicia-se no início do arquivo ou (se especificado) pelo byte b. A leitura inclui um registro por vez, sendo que o registro é delimitado pelo caracter RETURN. Este comando é cancelado por qualquer comando TKDOS posterior a ele.

**APPEND X [,Vv]**

Abre um arquivo-texto X já existente, este comando é similar ao OPEN, mas difere que este prepara o arquivo para receber o comando WRITE diretamente no final deste de modo a acrescentar mais informação.

**POSITION X, Rp**

Em um arquivo-texto aberto através do comando OPEN, permite deslocar o indicador de arquivo até uma região indicada pelo parâmetro p.

**EXEC X [,Rp] [,Vv]**

Executa sucessivamente os campos de um arquivo-texto seqüencial X interpretando cada campo deste arquivo como uma linha de um programa BASIC. O parâmetro p serve apenas para indicar de qual campo (linha) se deseja iniciar a execução.

**COMANDOS DE ARQUIVO-TEXTO ALEATÓRIOS****OPEN X, Lj [,Vv]**

Abre ou cria um arquivo de acesso randômico de nome X, atribuindo um buffer para este arquivo, e definindo um comprimento de campo de j bytes. Prepara para receber um comando WRITE ou READ do início do campo 0.

O mesmo comprimento de campo **deve** ser atribuído a cada vez que se abre o mesmo arquivo.

#### **CLOSE [X] [,Vv]**

Finaliza o comando WRITE X, se necessário e desatribui o buffer reservado para o arquivo texto X. Se o nome do arquivo, o comando CLOSE fechará todos os arquivos abertos neste instante.

#### **WRITE X [,Rr] [,Bb]**

Permite que a instrução PRINT escreva informação no arquivo X. Quando não é apresentado nenhum parâmetro, o comando inicia diretamente na posição do apontador de arquivo. Com apenas o parâmetro Rr, o comando inicia no byte 0 do campo r. Com o parâmetro Bb, o comando inicia no byte b da posição do apontador de arquivo (campo r, caso este esteja especificado). Este comando é cancelado com qualquer comando TKDOS posterior a ele.

#### **READ X [,Rr] [,Bb]**

Permite que a instrução INPUT leia os caracteres do arquivo-texto de acesso aleatório X. A identificação da posição do início de leitura tem o mesmo comportamento do comando WRITE, descrito acima.

### **COMANDOS DE ARQUIVOS BINÁRIOS**

#### **BSAVE X, Aa, Lj [,Vv]**

Guarda em disquete, sob o nome de X, o conteúdo dos j bytes de memória começando pelo endereço a.

#### **BLOAD X [,Aa] [,Vv]**

Carrega o arquivo binário X no mesmo endereço de memória no qual ele foi originalmente salvo ou (se especificado) no endereço a.

#### **BRUN X [,Aa] [,Vv]**

Carrega o arquivo binário X, e executa a instrução JMP para o endereço inicial deste arquivo.

Permite que a instrução INPUT leia caracteres do arquivo-texto X. A leitura inicia-se no início do arquivo ou (se especificado) pelo byte b. A leitura inclui um registro por vez, sendo que o registro é delimitado pelo caracter RETURN. Este comando é cancelado por qualquer comando TKDOS posterior a ele.

#### **APPEND X [,Vv]**

Abre um arquivo-texto X já existente, este comando é similar ao OPEN, mas difere que este prepara o arquivo para receber o comando WRITE diretamente no final deste de modo a acrescentar mais informação.

#### **POSITION X, Rp**

Em um arquivo-texto aberto através do comando OPEN, permite deslocar o indicador de arquivo até uma região indicada pelo parâmetro p.

#### **EXEC X [,Rp] [,Vv]**

Executa sucessivamente os campos de um arquivo-texto seqüencial X interpretando cada campo deste arquivo como uma linha de um programa BASIC. O parâmetro p serve apenas para indicar de qual campo (linha) se deseja iniciar a execução.

### **COMANDOS DE ARQUIVO-TEXTO ALEATÓRIOS**

#### **OPEN X, Lj [,Vv]**

Abre ou cria um arquivo de acesso randômico de nome X, atribuindo um buffer para este arquivo, e definindo um comprimento de campo de j bytes. Prepara para receber um comando WRITE ou READ do início do campo 0. O mesmo comprimento de campo **deve** ser atribuído a cada vez que se abre o mesmo arquivo.

#### **CLOSE [X] [,Vv]**

Finaliza o comando WRITE X, se necessário e desatribui o buffer reservado para o arquivo texto X. Se o nome do arquivo, o comando CLOSE fechará todos os arquivos abertos neste instante.

#### **WRITE X [,Rr] [,Bb]**

Permite que a instrução PRINT escreva informação no arquivo X. Quando não é apresentado nenhum parâmetro, o comando inicia diretamente na posição do apontador de arquivo. Com apenas o parâmetro Rr, o comando inicia no byte 0 do campo r. Com o parâmetro Bb, o comando inicia no byte b da posição do apontador de arquivo (campo r, caso este esteja especificado). Este comando é cancelado com qualquer comando TKDOS posterior a ele.

### READ X [,Rr] [,Bb]

Permite que a instrução INPUT leia os caracteres do arquivo-texto de acesso aleatório X. A identificação da posição do início de leitura tem o mesmo comportamento do comando WRITE, descrito acima.

### COMANDOS DE ARQUIVOS BINÁRIOS

#### BSAVE X, Aa, Lj [,Vv]

Guarda em disquete, sob o nome de X, o conteúdo dos j bytes de memória começando pelo endereço a.

#### BLOAD X [,Aa] [,Vv]

Carrega o arquivo binário X no mesmo endereço de memória no qual ele foi originalmente salvo ou (se especificado) no endereço a.

#### BRUN X [,Aa] [,Vv]

Carrega o arquivo binário X, e executa a instrução JMP para o endereço inicial deste arquivo.

CONTENTS OF ANNUAL BIRDS

1891-1892

THESE BIRDS WERE TAKEN AT THE STATIONERS' HALL, LONDON, IN THE MONTHS OF JANUARY AND FEBRUARY, 1891-1892. THE SPECIES ARE LISTED IN THE ORDER IN WHICH THEY WERE TAKEN.

PLATE I [A4] [V4]

THESE BIRDS WERE TAKEN AT THE STATIONERS' HALL, LONDON, IN THE MONTHS OF JANUARY AND FEBRUARY, 1891-1892. THE SPECIES ARE LISTED IN THE ORDER IN WHICH THEY WERE TAKEN.

PLATE II [A4] [V4]

THESE BIRDS WERE TAKEN AT THE STATIONERS' HALL, LONDON, IN THE MONTHS OF JANUARY AND FEBRUARY, 1891-1892. THE SPECIES ARE LISTED IN THE ORDER IN WHICH THEY WERE TAKEN.

