

Color BASIC
Color BASIC

DOS



O computador e a unidade de disco devem estar desligados quando você conectar a interface de disco, caso contrário o computador e a interface de disco poderão ser danificados.

PORQUE O DISCO É RÁPIDO

O termo exato para o disco, seria: "disco flexível", mas neste manu
al nós o chamaremos de disco. Ele serve para guardar informações e
é bem superior à fita cassete que é a outra alternativa de armaze
namento de dados.

O disco foi especialmente projetado para arquivar suas informa
ções, de maneira que o computador possa recuperá-las imediatamente
para você. Isto significa que guardar e recuperar informações, que
leva muito tempo em fita cassete, pode ser feito rapidamente e de
forma eficiente.

COMENTÁRIOS SOBRE ESTE MANUAL

Este manual ensina como ler e gravar informações num disco e foi escrito para atender a três grupos distintos de pessoas.

O primeiro grupo, formado por aqueles que aprenderem a programar lendo os manuais COLOR BASIC e COLOR BASIC EXTENDIDO, encontrarão nas seções I e II deste manual, uma experiência nova e encantadora e se forem ambiciosos, também apreciarão a seção III.

Os que nunca programaram e gostariam de usar programas aplicativos escritos por outras pessoas, formam o segundo grupo. Leiam o capítulo 1 "PARA COMEÇAR".

Se tiverem interesse e quiserem aproveitar todas as vantagens do COLOR DOS, continuem lendo a seção I, "O DISCO". Não é necessário conhecer nada sobre programação para entendê-la.

Porém, se você não pertence a nenhum desses dois grupos, provavelmente já sabe como programar com um sistema de disco. Leia o capítulo 1 para aprender a ligar o sistema e vá diretamente ao "RESUMO DO BASIC", no fim deste manual. Todos os assuntos estão ali resumidos com as respectivas referências, permitindo encontrar, com facilidade, as páginas onde tudo é abordado detalhadamente.

0 DISCO

Um disco é semelhante a um sistema de arquivamento. Tudo nele é organizado e por isto fa
cil de ser usado.

Nesta seção mostraremos como o computador organiza os programas e os dados no disco e como
você pode tirar proveito disto.

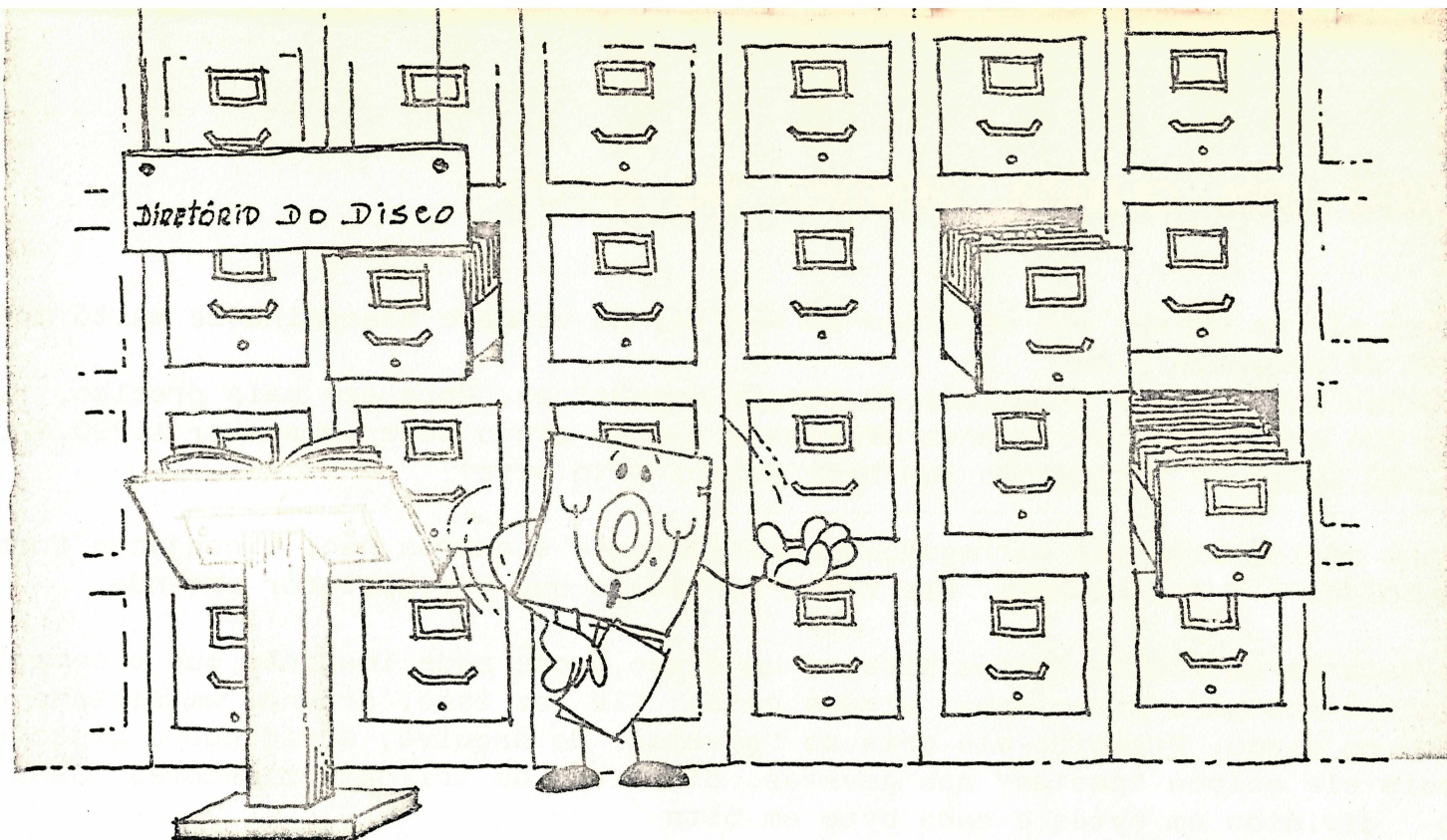
Você está convidado a ler esta seção e verá que nada é preciso saber sobre computador para
entendê-la.

SEÇÃO

I

0A932

CAPÍTULO
2



CONHEÇA O SEU DISCO

DANDO UMA OLHADA ^DENTRO^ DO DISCO

Embora ele se pareça com um disco de música, na verdade assemelha-se muito mais a um conjunto de pequenos imãs.

Um disco pode conter um milhão de cargas magnéticas. Para ser mais preciso, 1.290.240. É isto que queremos dizer quando afirmamos que um disco pode armazenar 1.290.240 bits ou 161.280 bytes de informação (um byte possui oito bits)

Alguns destes bits são carregados magnéticamente e outros não. O conjunto formado por estas cargas é o que importa. Ele forma um código que o computador entende.

Com mais de um milhão destes bits em um disco, você pode imaginar que o computador precisa organizá-los para poder achar alguma coisa. Ele faz isso, criando um sistema de arquivamento em disco. Primeiro ele cria as "gavetas" do arquivo, dividindo o disco em trilhas. Depois ele coloca "pastas" nas gavetas, dividindo as trilhas em setores. Cada setor é então dividido em bytes e cada byte em bits

NOTA: Para ser preciso, existem 35 trilhas num disco.
18 setores por trilha, 256 bytes em cada setor
e 8 bits em cada byte.

Depois de criar este sistema de arquivamento, o computador coloca um diretório no disco. Nele o computador indica onde tudo está guardado. Sempre que ele quiser encontrar um programa ou um arquivo, ele usa o diretório para descobrir a trilha e o setor em que está guardado e pode, então, ir diretamente ao local de armazenamento.

Este método de arquivamento é que torna o sistema de disco tão poderoso, permitindo-lhe encontrar rapidamente qualquer coisa que tenha sido gravada no disco.

O trabalho de criar este sistema de arquivamento no disco, chama-se "formatar o disco". A última tarefa que você fez no capítulo 1 foi inserir um disco "não formatado" na unidade de disco. Antes de usar este disco é preciso "formatá-lo" em trilhas e setores.

COMO FORMATAR UM DISCO

Como se formata um disco? Bem ... peça ao computador para fazê-lo. se seguiu todas as instruções do capítulo 1, você já ligou o sistema e colocou um disco não formatado dentro da unidade. Verifique se a porta da unidade está bem fechada.

Agora digite qualquer letra e pressione a tecla (ENTER) de maneira que

OK

seja a última linha de sua tela. (OK significa que o computador está pronto para fazer alguma coisa). Agora digite o que você deseja que ele faça. Digite:

DSKINIØ

e pressione a tecla (ENTER). O computador poderá imprimir ?SN ERROR. Caso isto ocorra, não se preocupe. Este erro significa que o comando foi digitado incorretamente. Digite novamente.

Sempre que alguma coisa sair errada, o computador informa a ocorrência através de uma mensagem de erro. Desta forma é possível corrigir imediatamente o erro. Se você obtiver outra mensagem de erro além de SN, veja a sua descrição no apêndice G. Neste apêndice estão relacionadas todas as mensagens de erro com as respectivas explicações.

Depois de digitar DSKINIØ (ENTER), a lâmpada vermelha da unidade de disco acende-se e começa a fazer barulho.

Cerca de 40 segundos depois o computador imprime OK. Isto quer dizer que a formatação está encerrada. A partir deste momento, você pode armazenar suas informações no disco. Lembre-se que não é possível armazenar dados num disco não formatado. Sempre que adquirir um disco não formatado, é preciso formatá-lo antes de usá-lo. Uma forma rápida de ver se um disco já foi formatado é verificar o seu diretório. (veja "COMO VERIFICAR O DIRETÓRIO", ao final deste capítulo). Se ocorrer uma mensagem de erro durante a formatação, ela não será efetuada.

NOTA: Reformatar um disco não causa danos ao disco. Esta é uma forma comum de apagar todo o seu conteúdo.

Se você possuir mais de uma unidade, pode formatar um disco numa outra unidade. Basta substituir o zero pelo número da unidade no comando DSKIN. Por exemplo, DSKINI 1 formata um disco na unidade 1.

COMO COLOCAR UM ARQUIVO NO DISCO

Um arquivo pode conter qualquer tipo de informação. Um programa, uma lista de endereços, cheques... O primeiro arquivo a ser criado por você, conterá um programa Basic por ser este o tipo de arquivo mais fácil de se criar.

Se você ainda não sabe programar em Basic, digite este programa como exemplo. Digite cada linha exatamente como está mostrado abaixo. Pressione a tecla **(ENTER)** ao final de cada uma delas. Digite:

```
10 PRINT "ARQUIVE ESTE PROGRAMA NO DISCO" (ENTER)
20 PRINT "E VOCE NUNCA O PERDERA" (ENTER)
```

Você já terminou? Agora que o programa está na "memória" do computador, você pode guardá-lo num disco. Para fazê-lo, nós o chamaremos de arquivo e lhe daremos o nome "SIMPLES/PRO" (todos os arquivos têm um nome). Para guardá-lo, digite:

```
SAVE "SIMPLES/PRO" (ENTER)
```

Depois que você pressionar a tecla **(ENTER)**, a unidade rangerá e a luz vermelha se acenderá. Neste momento o computador estará:

- . procurando um lugar no disco para guardar o programa "SIMPLES/PRO"
- . informando ao diretório do disco onde o programa "SIMPLES/PRO" será guardado
- . guardando o programa "SIMPLES/PRO" no disco

NOTA: O computador armazena o programa "SIMPLES/PRO" da mesma forma que ele guarda todas as outras coisas, sob forma de um código de cargas magnéticas.

Neste ponto, é importante preveni-lo de não retirar o disco da unidade enquanto a luz vermelha estiver acesa. Isto confunde o computador e pode danificar o conteúdo, não só do arquivo que você está utilizando, como também de outros que já tenham sido gravados no disco.

Quando o computador terminar de gravar o programa "SIMPLES/PRO", ele imprimirá a mensagem OK na tela.

NOTA: Note a diferença: "SAVE" guarda um programa no disco, "CSAVE" guarda um programa na fita cassete.

MEMÓRIA VS ARMAZENAMENTO EM DISCO

Para aqueles que estão mexendo em computadores pela primeira vez, nós gostaríamos de explicar um pouco o que é a "memória" do computador. Se você já sabe o que é, pule para o próximo título, "COMO CARREGAR UM ARQUIVO DO DISCO".

Sempre que digitar uma linha de programa BASIC e pressionar a tecla (ENTER), o computador, automaticamente, coloca-a na memória. Uma vez na memória, é possível fazer várias coisas com esta linha. Por exemplo, digite:

RUM (ENTER)

O computador imprime:

ARQUIVE ESTE PROGRAMA NO DISCO
E VOCE NUNCA O PERDERA.

Para listar o programa no vídeo, digite:

LIST (ENTER)

A memória é o local onde o computador grava tudo o que você lhe disser. Sempre que colocar uma informação na sua memória, o computador pode imprimi-la, reorganizá-la, combiná-la e fazer muitas outras coisas.

Mais tarde, provavelmente, você vai querer colocar outras coisas na memória, como por exemplo, a lista de seus amigos.

Para fazê-lo, será preciso escrever ou comprar um programa feito especialmente para esta aplicação. Este aplicativo permitirá que o computador coloque na memória as informações que você desejar.

É bom lembrar que ao desligarmos o computador, o conteúdo de sua memória é perdido. Depois que a memória for apagada, não é mais possível recuperá-la. A única forma de manter uma cópia do que foi digitado na memória, é guardá-la num disco ou numa fita cassete.

COMO CARREGAR UM ARQUIVO DO DISCO

Digite NEW (ENTER) para apagar todo o conteúdo da memória. Para ter certeza que tudo foi apagado, digite um dos seguintes comandos:

RUN (ENTER)
LIST (ENTER)

Embora NEW tenha apagado o programa "SIMPLES/PRO" da memória, ele permanece gravado no disco. Você pode trazê-lo de volta à memória a qualquer momento, simplesmente carregando-o do disco. Para fazê-lo digite:

LOAD "SIMPLES/PRO" (ENTER)

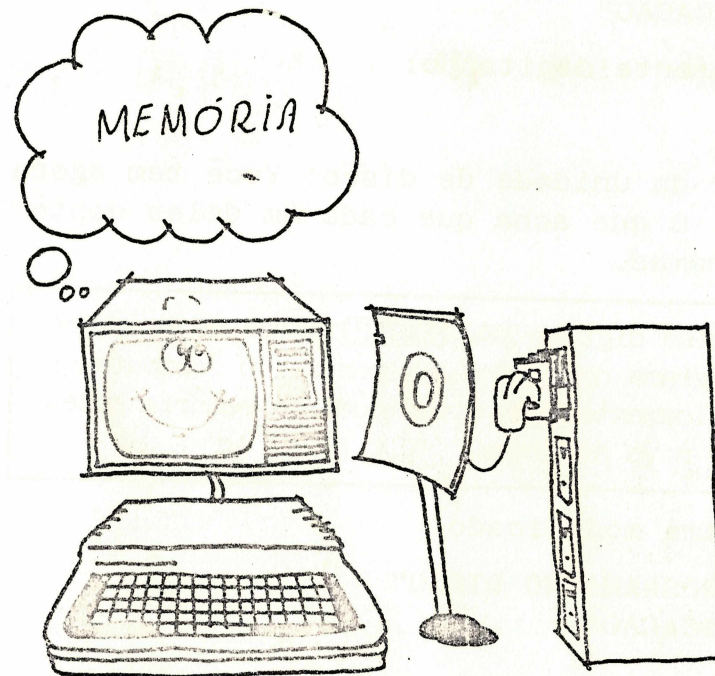
Novamente, a unidade de disco rangerá um pouco e a luz vermelha se acenderá. Neste momento o computador estará:

- . lendo o diretório para descobrir onde está guardado o programa "SIMPLES/PRO".
- . indo ao local onde está o programa "SIMPLES/PRO" e lendo o seu conteúdo.
- . colocando o programa "SIMPLES/PRO" na memória.

Agora você pode digitar um dos seguintes comandos para ver se o programa "SIMPLES/PRO" realmente se encontra na memória.

LIST (ENTER)

RUN (ENTER)



ALGO MAIS SOBRE MEMÓRIA VS ARMAZENAMENTO EM DISCO

Se você ainda está um pouco confuso quanto ao conceito de memória e de disco, faça o seguinte exercício. Você acaba de carregar o programa chamado "SIMPLES/PRO" na memória, certo? Modifique-o, digitando:

```
20 PRINT "COM ESTA MODIFICACAO" (ENTER)
```

Liste o programa novamente para ver se o computador registrou a modificação da linha 20.

```
10 PRINT "ARQUIVE ESTE PROGRAMA NO DISCO"
```

```
20 PRINT "COM ESTA MODIFICACAO"
```

Guarde-o num arquivo diferente digitando: ,

```
SAVE "MUDANCA" (ENTER)
```

Ouçá o zumbido e o ranger da unidade de disco! Você tem agora dois arquivos no disco: "SIMPLES/PRO" e MUDANÇA". O que acha que cada um deles contém? Tente carregar (LOAD) e listar (LIST) ambos os programas.

NOTA: Você não precisa digitar NEW (ENTER) antes de carregar (LOAD) um programa novo para a memória. O computador apaga automaticamente todo o conteúdo da memória antes de carregar o novo programa.

"MUDANÇA" contém o programa modificado:

```
10 PRINT "ARQUIVE ESTE PROGRAMA NO DISCO"
```

```
20 PRINT "COM ESTA MODIFICACAO"
```

Entretanto, "SIMPLES/PRO" ainda contém o programa antigo:

```
10 PRINT "ARQUIVE ESTE PROGRAMA NO DISCO"
```

```
20 PRINT " E VOCE NUNCA O PERDERA"
```

A única forma de modificar o conteúdo de um arquivo em disco é... bem, você mesmo pode dar a resposta. Como é possível fazer o arquivo "SIMPLES/PRO" conter:

```
10 PRINT "ARQUIVO MODIFICADO"
```


Resposta:

Digite:

NEW (ENTER)

10 PRINT "ARQUIVO MODIFICADO"

SAVE "SIMPLES/PRO" (ENTER)

NOME DE ARQUIVOS

Você já usou um nome de arquivo:

"SIMPLES/PRO"

Se você fez o exercício "memória VS armazenamento em disco", já usou um segundo nome de arquivo:

"MUDANCA"

Foi dado ao nome "SIMPLES" uma extensão "PRO". Você deve dar um nome a todas as coisas que for guardar no disco. A extensão fica a seu critério, ela não é obrigatória:

Quais os nomes que você pode dar a um arquivo? Todos que você quiser, desde que siga as regras abaixo:

1. O nome não pode ter mais que oito caracteres.
2. Dando uma extensão ao nome, ela não pode ter mais que três caracteres.
3. Tem que existir uma barra (/) ou um ponto (.) entre o nome e a extensão.

Está bem claro, Bom.

NOTA: Você pode usar qualquer caracter no nome do arquivo, exceto dois pontos(:) e zero(0). Você só pode usar a barra(/) e o ponto(.) para separar o nome da extensão.

NOME DE ARQUIVO QUANDO VOCÊ TEM MAIS DE UMA UNIDADE DE DISCO

Se você possui mais que uma unidade, pode acrescentar o número da unidade ao nome do arquivo. (Lembre-se: você numerou todas as suas unidades no capítulo 1).

Por exemplo:

```
LOAD "SIMPLES/PRO:1"
```

Carrega o programa "SIMPLES/PRO" do disco na unidade número 1.

```
SAVE "MUDANCA : 1"
```

Grava o programa "MUDANCA" no disco na unidade 1. Se você não indicar o número da unidade, o computador assume que é para usar a unidade Ø.

VERIFICANDO O DIRETÓRIO

Como já foi dito anteriormente, o disco possui um diretório que o computador usa para encontrar os arquivos que estão gravados nele. Se o computador pode usar o diretório, voçê também pode. Digite:

```
DIR (ENTER)
```

O computador lista no vídeo as informações de todos os arquivos que foram gravados no disco. Se até o momento você só gravou os arquivos "SIMPLES/PRO" e "MUDANCA", o computador imprimirá o seguinte:

SIMPLES	PRO	Ø	B	1
MUDANCA	BAS	Ø	B	1

A primeira e a segunda coluna indicam o nome do arquivo. A primeira é o nome, enquanto que a segunda é a extensão. Note que mesmo não tendo especificado uma extensão para o programa "MUDANCA", o computador criou a extensão "BAS".

O computador prefere que todos os nomes de arquivos possuam uma extensão. Se você não der uma extensão quando da gravação do arquivo, ele especificará, automaticamente, uma das seguintes extensões:

"BAS", se for um programa Basic

"DAT", se for um dado (tais como nomes, números, etc.)

"BIN", se for um programa em linguagem de máquina

NOTA: Um programa em linguagem de máquina é um programa altamente técnico que fala diretamente com o computador.

As três colunas seguintes contem informações de interesse para programadores técnicos. Você está interessado também? Então continue lendo...

A terceira coluna relaciona os tipos de arquivos que são:

Ø programa BASIC

1 dados criados por um programa BASIC

2 dados criados por um programa em linguagem de máquina

3 programa fonte criado por um editor de assembler

NOTA: O editor de assembler é um programa utilizado para ajudá-lo no desenvolvimento de programas em linguagens de máquinas.

A quarta coluna indica em que formato foi gravado o arquivo.

A, ASCII

B, Binário

O significado destas palavras será visto no capítulo 10.

A quinta coluna mostra quantos "grânulos" cada arquivo ocupa. "SIMPLES/PRO" e "MUDANCA/BAS" consumiram um "grânulo" cada. (o computador usa "grânulos" para indicar a capacidade de armazenamento de um disco. Um disco possui 68 "grânulos").

Se você inseriu discos formatados em outras unidades, também é possível verificar seus diretórios. Por exemplo DIR 1 (ENTER) lista na tecla o diretório do disco que se encontra na unidade 1.

Você está impressionado? Ficarão mais ainda quando descobrir com que velocidade é possível gravar e recuperar programas grandes. Mas antes que você fique muito envolvido, por favor, leia o próximo capítulo. Ele garantirá que sua experiência com o COLOR DOS está sendo aperfeçoada de forma agradável e segura.

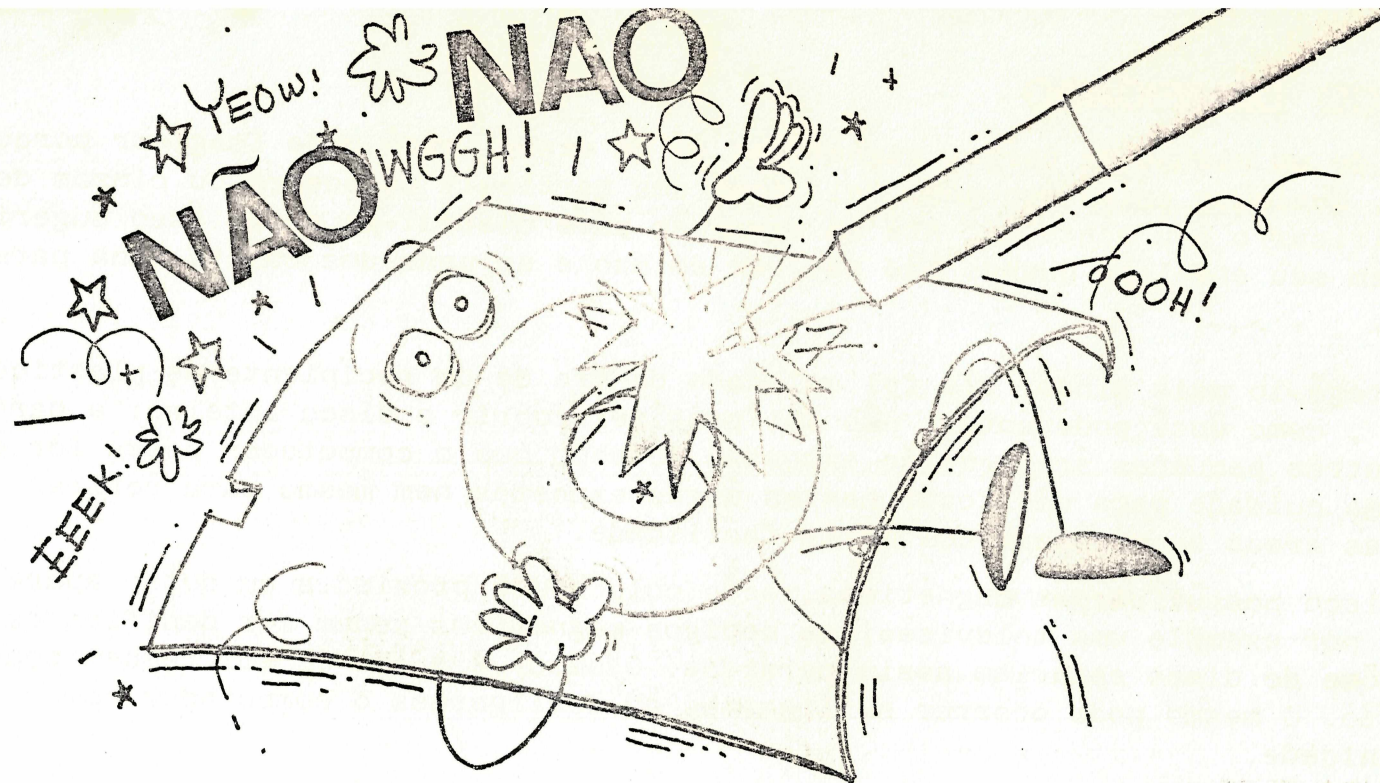
NOTA: Para parar a listagem do diretório, pressione as teclas (SHIFT) @ simultaneamente e logo em seguida (BREAK).

VERIFICAÇÃO DE APRENDIZADO

1. Por que não é possível guardar coisas num disco não formatado?
2. O que é diretório do disco?
3. O que é um arquivo em disco?
4. Qual é a diferença entre o que está na memória e o que está no disco?
5. Como você muda o conteúdo de um arquivo em disco?

OBS.: As respostas estão no apêndice B.

CAPITULO
3



UM DISCO DANIFICADO

UM DISCO DANIFICADO

Com mais de um milhão de cargas magnéticas em um disco, você pode imaginar porque ele é tão delicado. Qualquer coisa pequena, tal como uma partícula de poeira ou cinzas de cigarro, pode danificar o seu conteúdo. Até um arranhão pode destruí-lo e por isso sugerimos manter o disco em seu envelope quando não estiver em uso e só usar uma caneta fina para etiquetá-lo.

Para protegê-lo mais ainda, ele foi colocado dentro de um recipiente de plástico preto. Entretanto, como você pode notar, não foi possível cobrir o disco inteiro: a parte do meio e duas outras pequenas áreas estão expostas de modo que o computador possa ler e escrever nele. Tome cuidado para não tocar nestas áreas expostas, nem mesmo para retirar a poeira, pois estas áreas se arranham com muita facilidade.

Como o disco possui cargas magnéticas, se o colocarmos próximo a um outro aparelho magnético, como por exemplo uma televisão, os códigos magnéticos podem ser danificados. Todas as informações do disco estariam assim perdidas. O calor e a luz do sol também podem provocar tal efeito. O mesmo pode ocorrer se ligarmos ou desligarmos o computador com o disco dentro da unidade.

Mais uma coisa... Se você estiver executando um programa e precisar trocar de disco, recomendamos que seja digitado o seguinte comando:

UNLOAD (ENTER)

antes de fazer a troca. Desta forma o computador coloca as informações de fechamento no disco apropriado. Se Você não fizer isso, o computador pode colocar estas informações num disco errado e danificar o conteúdo dos dois discos.

NOTA: Para os programadores BASIC: Todos os arquivos abertos devem ser fechados antes da troca de discos. O comando UNLOAD fecha todos os arquivos abertos.

CÓPIA DE SEGURANÇA

Tudo isto pode parecer um pouco complicado para você, mesmo sendo uma pessoa cuidadosa. Por esta razão foi criado o comando BACKUP, que permite a confecção de cópias dos discos.

Sugerimos que você faça regularmente cópias de segurança dos discos que contêm informações importantes. Desta forma não precisará se preocupar quanto à possibilidade de perdê-las.

Como o disco pode torna-se defeituoso devido ao uso, é aconselhável fazer cópias de discos antigos para novos. Assim sendo, quando o computador começar a apresentar problemas para ler ou gravar num disco, você poderá usar a sua cópia.

Você quer fazer uma cópia de segurança? Pegue os dois discos a serem utilizados:

1. O disco fonte é aquele que você quer duplicar. Pode usar qualquer disco que possua arquivos. Se você está começando agora, pegue o disco que usou no capítulo 2.
2. O disco destino é aquele para onde será feita a cópia. Use um disco vazio ou, se você já estiver usando o sistema de disco há mais tempo, qualquer disco que contenha arquivos inúteis.

NOTA: Todo o conteúdo do disco destino será apagado.
O disco destino será preenchido com as informa
ções contidas no disco fonte.

Se o disco destino estiver vazio, é preciso primeiro "formatá-lo". Você ainda se lembra como fazê-lo? Coloque o disco destino dentro da unidade, feche a porta e digite:

DSKINIØ (ENTER)

Faça agora a cópia de segurança. Este procedimento depende do número de unidades de disco que você possui.

CÓPIAS DE SEGURANÇA COM UMA UNIDADE DE DISCO

Se você possuir uma única unidade de disco, o computador levará cerca de cinco minutos para fazer uma cópia de segurança. Coloque o disco fonte na unidade e feche a porta. Digite DIR (ENTER) para ver quais os arquivos que serão copiados.

Comece agora o procedimento de cópia. Digite:

BACKUP Ø (ENTER)

Depois de ler uma parte do disco fonte fazendo barulho, o computador imprime a seguinte mensagem na tela:

INSERT DESTINATION DISKETTE AND PRESS (ENTER)
(insira o disco destino e pressione (ENTER))

Retire o disco fonte e insira o disco destino. Feche a porta e pressione (ENTER). Você ou virá de novo um barulho enquanto o computador "escreve" no disco destino. Depois disso ele imprime no vídeo:

INSERT SOURCE DISKETTE AND PRESS (ENTER)
(insira o disco fonte e pressione (ENTER))

Esta rotina de cópia será executada várias vezes até que todos os dados contidos no disco fonte, sejam copiados para o disco destino. Durante este processo de cópia, verifique se está introduzindo o disco certo e se ele foi corretamente colocado na unidade. Quando a cópia terminar, o computador imprimirá a mensagem OK na tela:

Para ter certeza que a cópia de segurança foi executada com sucesso, insira o disco destino na unidade e digite DIR (ENTER).

COPIAS DE SEGURANÇA COM VÁRIAS UNIDADES DE DISCO

Se você possuir mais de uma unidade de disco, é bem mais fácil fazer cópias de segurança. Cada cópia levará cerca de dois minutos. Coloque o disco fonte na unidade 0 e o disco destino na unidade 1 (O capítulo 1 mostra como identificar as unidades de disco) Digite então:

BACKUP 0 TO 1 (ENTER)

Você ouvirá um barulho enquanto o computador transfere o conteúdo do disco fonte na unidade 0 para o disco destino na unidade 1. Ao final da cópia é impresso na tela a mensagem OK. Para ter certeza que a cópia de segurança foi executada com sucesso, basta digitar DIR 1 (ENTER).

Você pode usar unidades de discos diferentes se quiser. Por exemplo:

BACKUP 1 TO 0 (ENTER).

copia o conteúdo do disco da unidade 1 para o disco da unidade 0.

SE SURGIREM PROBLEMAS DURANTE A CÓPIA

Se ocorrer uma mensagem de erro durante uma cópia, será provavelmente porque você introduziu o disco incorretamente na unidade de disco ou há algo errado com o disco. No final deste capítulo falaremos sobre as mensagens de erro, para ajudá-lo a descobrir qual o problema. Se estiver usando um disco defeituoso, será preciso fazer a cópia com outro disco.

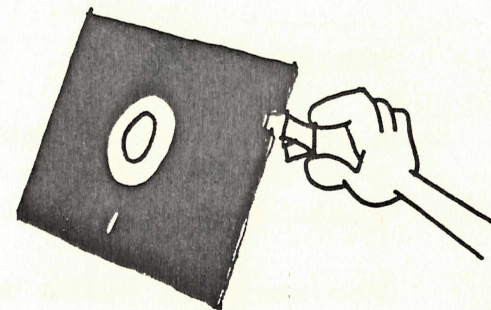
Depois de determinar o problema, aperte o botão de RESET para sair do programa de cópia (BACKUP) e comece novamente o procedimento de cópia.

NOTA: O botão RESET fica na parte de trás do computador, do lado direito de quem olha para ele.

COMO PROTEGER UM DISCO

É possível proteger um disco contra gravações. Vamos supor que você tenha um disco com alguma informações importantes, como por exemplo, um programa complexo que você não pretende modificar. Você planeja ler o conteúdo do disco, diariamente, carregando este programa para a memória mas não pretende "escrever" (gravar informações) nele.

Se colocarmos uma pequena etiqueta no entalhe do disco, isto possibilitará ao computador ler informações do disco mas impedirá a gravação de dados. Qualquer tipo de fita gomada serve. Junto com a caixa de discos deve ter vindo uma cartela cheia de pequenas etiquetas.



COMO RECUPERAR UM DISCO

Já foi mencionado anteriormente que um disco não dura para sempre. Antes de jogar fora um disco antigo, tente recuperá-lo. Você pode conseguir recuperá-lo "formatando-o", como se fosse um disco virgem.

Embora você consiga recuperar o disco, não será possível salvar o seu conteúdo. "Reformatando" o disco, você apagará tudo que ele contém. Entretanto, isso evitará que você gaste dinheiro comprando um novo disco.

Se ocorrer um erro de **IO** enquanto tenta reformatar o disco (veja "mensagem de erro" ao final deste capítulo), isto quer dizer que o disco provavelmente já alcançou o seu limite de utilização. Se você possuir um apagador (bulk eraser), tente limpar o disco e reformatá-lo. Caso contrário, jogue-o fora e use outro disco.

NOTA: Se você tiver mais de uma unidade de disco, é possível copiar alguns dos arquivos que estão no disco defeituoso para outro disco. Esta operação é feita com o comando COPY que será apresentado no próximo capítulo.

VERIFICANDO UM DISCO

O computador "escreve" dados no disco a uma velocidade muito rápida. Quase sempre isto é feito sem erro.

Algumas vezes você vai querer ter certeza que não ocorreram falhas durante a gravação. Isto pode ser feito ligando o comando de verificação. Para fazê-lo, digite:

VERIFY ON (ENTER)

A partir deste momento, o computador irá informá-lo sempre que algo for escrito no disco e se ocorrer alguma falha na gravação. O único problema é que desta forma, o computador gasta duas vezes mais tempo de gravar.

Por exemplo, vamos admitir que você faça agora uma cópia de segurança (BACKUP). O computador levará duas vezes mais tempo para fazê-lo, mas em contrapartida, ele lhe informará sobre qualquer falha durante a cópia.

O comando VERIFY permanece "ligado" até que você o desligue. Para fazer isto, digite:

VERIFY OFF (ENTER)

QUANDO AS COISAS NÃO DÃO CERTO

O computador sabe que ninguém é perfeito. Quando você cometer um erro ele procurará informá-lo imediatamente, indicando o tipo de erro que você cometeu.

Você provavelmente já teve um "SN ERROR". Se você ainda não teve este tipo de erro, digite DIIR (ENTER) escrevendo propositalmente o comando DIR de forma errada.

"SN" significa erro de sintaxe. É a maneira do computador lhe dizer que "DIIR" não faz sentido para ele. Esta palavra não está em seu vocabulário. Um erro "SN" geralmente significa que você cometeu um erro de digitação.

A seguir são apresentadas outras mensagens de erro que podem ocorrer em sistemas de disco.

AE - Você está tentando mudar o nome de um arquivo usando um nome que já existe. Para modificar o nome de um arquivo, usa-se o comando RENAME. Este comando será visto no próximo capítulo.

DF - O disco em que você está tentando armazenar o seu arquivo está cheio. Use outro disco.

DN - Você está usando um número de unidade de disco superior a 3. Este erro também ocorre se você não especificar um número de unidade de disco quando usar os comandos DSKINI e BACKUP. Se você possuir uma única unidade de disco, indique o número 0 nestes comandos (DSKINI0 e BACKUP0).

FN - Você está usando um formato incorreto para dar nome ao seu arquivo. No capítulo 2 você encontrará as regras para criar nomes corretos.

FS - Há algo de errado com o seu arquivo em disco. Veja a mensagem de erro IO para saber o que fazer neste caso.

- /Ø - Tecnicamente isso significa que você pediu ao computador para dividir um número por zero. O que é impossível, entretanto você também pode obter este erro se não escrever o nome de um arquivo entre aspas.
- IO - O computador está encontrando problemas para gravar ou ler informações do disco.
- (1) Certifique-se de que o disco esteja colocado corretamente dentro da unidade de disco e que a porta esteja fechada.
 - (2) Se este erro persistir, deve haver algo de errado com o seu disco. Tente inseri-lo novamente, depois tente usar outro disco ou reformatá-lo. (Lembre-se que reformatar um disco implica em perder o seu conteúdo).
 - (3) Se você ainda continuar tendo este erro, o computador ou a unidade de disco deve estar com problemas. É necessário entrar em contato com um representante técnico.
- NE - O computador não está conseguindo achar o arquivo que você quer. Verifique o diretório do disco para ter certeza de que ele está lá. Se você possuir mais de uma unidade de disco, pode ter esquecido de colocar ao lado do nome o número da unidade de em que se encontra o arquivo. Se está usando os comandos COPY, KILL ou RENAME (descritos no próximo capítulo), poderá ter esquecido de indicar a extensão do nome.
- TM - Tecnicamente este erro é provocado por um programa que mistura cadeias de caracteres com números. Entretanto, você também pode obter este erro se você não escrever o nome de um arquivo entre aspas.
- VF - Você obterá este erro quando o comando VERIFY estiver ligado e ocorrer uma gravação defeituosa. O computador está informando que há uma falha no que ele gravou. Veja as instruções do **IO** para saber o que fazer neste caso.
- WP - Você está tentando armazenar informações num disco que está protegido contra gravação. Tire a etiqueta do entalhe de proteção ou use um outro disco que não esteja protegido. Se o seu disco não estiver protegido, então ocorreu um erro de entrada ou saída. Veja as instruções do erro **IO** para saber o que fazer neste caso.

Todos os outros erros que poderão surgir, são erros do programa que você está usando. Se não foi você quem escreveu o programa e apareceu um destes erros, será preciso entrar em contato com a pessoa que o fez. Se foi você que o escreveu, veja o apêndice G. Nele encontrará uma explicação para todas as mensagens de erro.

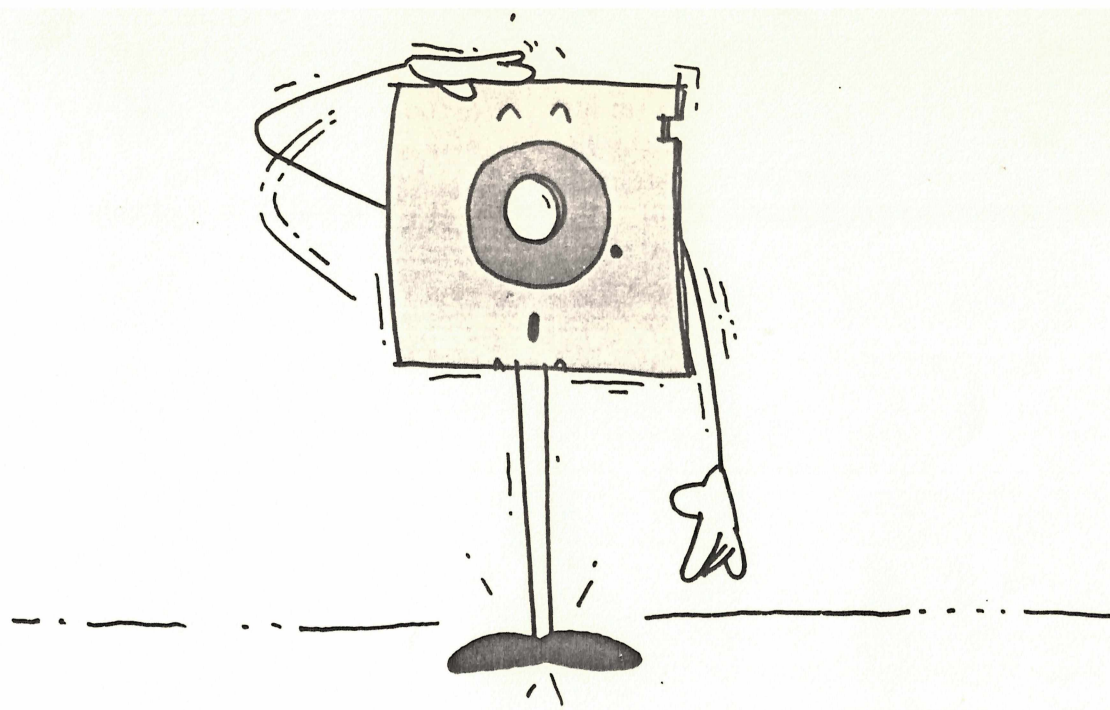
Cuidar de seu disco pode parecer um pouco incômodo a princípio. Você gastou muito tempo em sua vida protegendo seus papéis e agora está lidando com um instrumento diferente.

Daqui a pouco, contudo, proteger os discos contra a poeira e os aparelhos magnéticos parecerá tão natural quanto proteger seus papéis de uma ventania. E assim que você se habituar a manter o disco em bom estado, nunca mais vai voltar aos lápis e papéis novamente (assim esperamos).

VERIFICAÇÃO DE APRENDIZADO

1. Por que não se deve ligar ou desligar o computador enquanto o disco está dentro da unidade de disco?
2. Que tipo de caneta deve-se usar para identificar o disco?
3. O que são mensagens de erro?
4. O que significa proteger um disco contra gravações? Como fazê-lo?
5. Como se faz cópias de segurança de um disco?

CAPÍTULO
4



VOCÊ É O PATRÃO



VOCE É O PATRÃO

VOCÊ É O PATRÃO

Graças ao sistema de arquivamento em disco, você pode pedir ao computador que faça muitas coisas úteis para você. Pode por exemplo alterar o nome de um arquivo. Caso já tenha retirado o disco formatado da unidade, insira-o novamente.

NOTA: Caso não se lembre se o disco foi formatado, verifique o seu diretório digitando DIR (ENTER) (ou DIR Ø ou DIR 1 se você possuir mais de uma unidade).

Digite as seguintes linhas para colocar um arquivo no disco:

```
1Ø PRINT "ISTO E UM ARQUIVO" (ENTER)
SAVE "ORIGINAL/NOM" (ENTER)
```

Verifique o diretório para ter certeza que o programa foi armazenado com o nome "ORIGINAL/NOM" ... Agora modifique este nome. Digite:

```
RENAME "ORIGINAL/NOM" TO "NOVO/NOM" (ENTER)
```

Você está ouvindo a unidade de disco trabalhar? Verifique o diretório novamente. Caso tenha gostado, carregue (LOAD) e liste (LIST) o programa "NOVO/NOM". O nome do programa foi simplesmente trocado. Todo o resto permanece igual.

O comando RENAME é fácil de usar, mas há uma coisa que é preciso saber. Crie um arquivo dando-lhe um nome sem extensão e tente alterar o seu nome. Digite:

```
1Ø PRINT "ARQUIVO NUMERO DOIS" (ENTER)
SAVE "ARQUIVO1" (ENTER)
```

```
RENAME " ARQUIVO1" TO "ARQUIVO2" (ENTER)
```

O computador indica que ocorreu o erro NE. Isto significa que ele não conseguiu encontrar o arquivo.

Quando você for trocar o nome de um arquivo, é preciso digitar o nome completo deste arquivo para que o computador possa encontrá-lo. Isto inclui o nome e a extensão.

Como já foi dito no capítulo 2, sempre que você gravar um arquivo no disco, o computador verificará se ele possui uma extensão. Se você não fornecer uma extensão o computador o fará.

Você pode listar o diretório para descobrir a extensão do programa "ARQUIVO1" . Depois disso tente mudar o nome do programa, digitando:

```
RENAME "ARQUIVO1/BAS" TO "ARQUIVO2/BAS" (ENTER)
```

Se você está mudando o nome de um programa, certifique-se de que o novo nome possua uma extensão. Em outras palavras, não digite `RENAME "ARQUIVO1/BAS" TO "ARQUIVO2"` (ENTER). O computador mudará o nome do programa mas entretanto o nome "ARQUIVO2" não terá uma extensão. Isto causará um problema quando tentar carregá-lo (`LOAD "ARQUIVO2"`) para a memória, porque todos os arquivos que desejar carregar (`LOAD`), precisam ter uma extensão.

Isto pode parecer conflitante com o que foi dito acima. Você gravou o programa "ARQUIVO1" (`SAVE "ARQUIVO1"`) sem fornecer uma extensão. O computador neste caso fornece uma extensão antes de gravar o programa. O comando `RENAME` funciona de outra maneira. O computador não designará automaticamente uma extensão para o programa cujo nome você mudou.

NOTA: Só existe uma forma de carregar o programa "ARQUIVO2" sem extensão. Isto se faz indicando que não existe extensão digitando `LOAD "ARQUIVO2/"` (ENTER). Isto não é correto e por esta razão sugerimos que sempre que você usar o comando `RENAME`, coloque uma extensão nos nomes.

COMO USAR VÁRIAS UNIDADES DE DISCO

Você pode mudar o nome de um arquivo que está numa outra unidade de disco, simplesmente digitando o número apropriado da unidade. introduza um disco formatado na unidade 1 (se ele ainda não foi introduzido). Armazene um arquivo dentro dele:

```
10 PRINT "CONTAS" (ENTER)
SAVE "VELHACON/DAT:1" (ENTER)
```


e troque o seu nome digitando:

RENAME "VELHACON/DAT:1" TO "NOVACON/DAT:1" (ENTER)

NOTA: Caso você queira colocar este programa num outro disco, não pode usar o comando RENAME. É preciso usar o comando COPY.

ACABOU O ESPAÇO NO DISCO?

Mais cedo ou mais tarde você vai querer saber quanto espaço sobra no seu disco. Digite:

PRINT FREE (Ø) (ENTER)

O computador imprimirá o número de "grânulos que ainda estão livres (FREE) no seu disco.

Existem 68 grânulos ao todo. Se o computador lhe disser que só resta um grânulo livre, é melhor tomar uma das seguintes alternativas: passe a usar um novo disco ou elimine (KILL) alguns dos arquivos.

Eliminar (KILL) um arquivo do disco consiste em fazer exatamente o que o nome indica. Por exemplo, se você colocou o programa "MUDANCA" no seu disco, como indicado no capítulo 2, digite:

KILL "MUDANCA/BAS" (ENTER)

Verifique o diretório e o tamanho do espaço livre no seu disco. O programa "MUDANCA/BAS" não se encontra mais no disco e o espaço que ele ocupava está livre para ser usado por no vos arquivos:

Repare que foi preciso incluir a extensão "BAS" do nome "MUDANCA" para eliminá-lo. O computador exige que você digite o nome completo do arquivo com uma precaução extra. Ele não quer eliminar um arquivo que não era para ser destruído.

NOTA: Você quer ser detalhista? O arquivo eliminado continua no disco, entretanto o computador não sabe que ele está lá porque o comando "KILL" apaga do diretório, todas as referências ao arquivo. Por esta razão não é mais possível recuperar os dados gravados e o computador consegue gravar novos arquivos naquele lugar do disco.

COMO USAR VÁRIAS UNIDADES DE DISCO

Os comandos FREE e KILL podem ser usados com outras unidades de disco da mesma forma que o comando RENAME. Basta digitar o número da unidade. Exemplos:

PRINT FREE (1) (ENTER)

Informa o tamanho do espaço livre no disco da unidade 1.

KILL "NOVACON/DAT:1" (ENTER)

Elimina o programa "NOVACON/DAT" do disco da unidade 1.

COMANDOS ESPECIAIS PARA CONFIGURAÇÕES COM VÁRIAS UNIDADES

A seguir falaremos de dois comandos que você só pode usar se possuir várias unidades de disco. Caso só tenha uma, vá diretamente para a verificação de aprendizado ao final deste capítulo.

O primeiro comando copia um arquivo em disco. Você já deve ter um programa gravado no disco da unidade 0, chamado "NOVO/NOM". Faça uma cópia dele. Digite:

COPY "NOVO/NOM:0" TO "NOVO/NOM:1" (ENTER)

Se você quiser, é possível modificar o nome do arquivo com o comando COPY. Por exemplo, COPY "NOVO/NOM:1" TO "OUTRO/NOM:0" (ENTER). Copia o programa "NOVO/NOM" que está no disco da unidade 1 para o programa "OUTRO/NOM" que está no disco da unidade 0.

O segundo comando muda o número da unidade de disco que o computador utiliza, se você não fornecer um número. Se digitar, por exemplo, SAVE "QUALQUER/EX" (ENTER), o computador assume que você quer usar a unidade 0. Ele grava então este programa no disco da unidade 0.

Para mudar esta pré suposição, digite:

DRIVE 1 (ENTER)

Este comando faz com que o computador assuma que é para utilizar a unidade 1 a menos que lhe seja indicado para usar outra.

Depois que se alterar o número da unidade de disco que o computador deve assumir quando nada é dito, ele responde de forma diferente ao comando SAVE "QUALQUER/EX" (ENTER). Neste caso, o computador vai gravar o programa "QUALQUER/EX" no disco da unidade 1. Você agora precisará digitar SAVE "QUALQUER/EX:0" (ENTER) para gravá-lo no disco da unidade 0.

VERIFICAÇÃO DE APRENDIZADO

1. Como se altera o nome de um arquivo? Por que é necessário especificar a extensão do nome deste arquivo?
2. Como deve você proceder quando o espaço livre no disco estiver acabando?
3. Se você possuir mais de uma unidade de disco e não indicar o número da unidade a ser utilizada, qual o computador usará? Como se muda isso?

Parabéns. Você é agora um autêntico operador de computador com unidades de disco. Deve ter agora uma boa compreensão de como funciona o sistema de disco e como tirar vantagens dele.

O primeiro passo para a instalação do sistema é a verificação da configuração do hardware. É necessário garantir que o computador atenda aos requisitos mínimos para a execução do sistema. Isso inclui a verificação da capacidade de memória, do espaço em disco e da velocidade do processador.

Para obter mais informações, consulte o manual de instalação do sistema.

DRIVE 1 (CD-ROM)

Este capítulo descreve como instalar o sistema a partir do CD-ROM. É necessário ter um drive de CD-ROM disponível no computador. O processo de instalação é simples e pode ser concluído em poucos minutos.

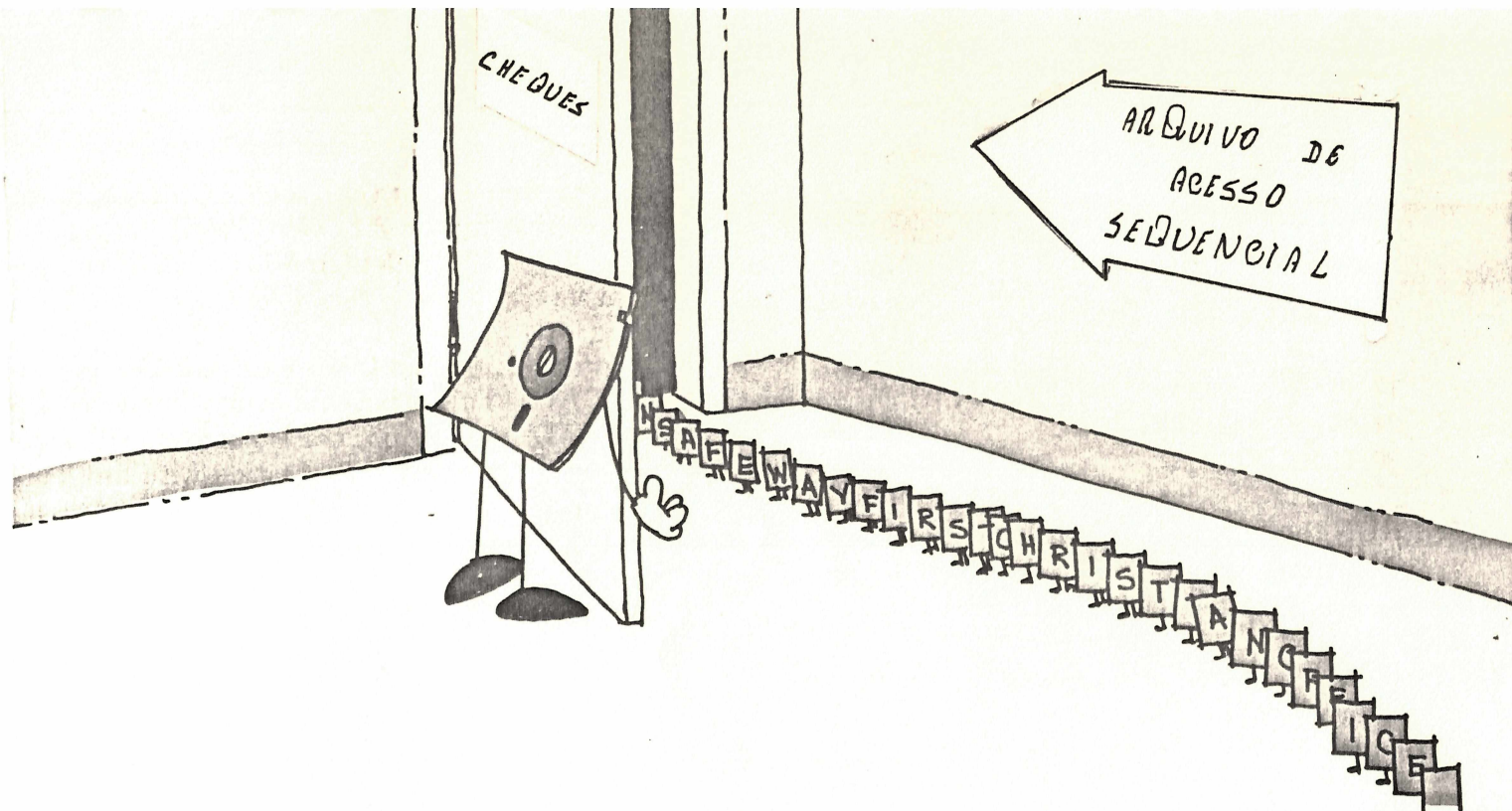
Depois de ter instalado o sistema, é necessário configurar o hardware. Isso inclui a verificação da configuração da placa de vídeo, do teclado e do mouse. O sistema também requer a instalação de drivers para os dispositivos de hardware. O manual de instalação fornece informações detalhadas sobre como fazer isso.

VERIFICAÇÃO DE APRENDIZADO

1. Como se alterar o nome de um arquivo? For the answer, see the manual.
2. Como se obter o endereço de um arquivo? For the answer, see the manual.
3. Se você quiser saber se um arquivo de dados é novo ou antigo, como fazer isso? For the answer, see the manual.

Este manual contém informações importantes sobre a instalação e o uso do sistema. É importante ler este manual cuidadosamente antes de prosseguir com a instalação. O manual também contém informações sobre como obter suporte técnico e como atualizar o sistema.

CAPITULO
5



UMA COISA DE CADA VEZ

(ARQUIVO DE ACESSO SEQUENCIAL)

12-00000
1000
1000000

3
4
5
6
7
8
9
0
1
2

UMA COISA DE CADA VEZ

(ARQUIVO DE ACESSO ESPECIAL)

SEÇÃO

II

SECRETARIA DE BENS

SEÇÃO

II

O PROGRAMA DE DISCO

Armazenar um programa BASIC é muito fácil, basta usar o comando SAVE, mas armazenar dados é um pouco mais trabalhoso, você precisa fazer um programa.

Alguns de vocês poderão preferir comprar um programa pronto. Entretanto, se quiser dominar melhor o computador e investir mais tempo, você vai gostar de fazer os seus próprios programas.

Nesta seção mostraremos como escrever um programa BASIC que grava dados num arquivo em disco. Vamos supor que você já conheça a linguagem BASIC. Caso não a conheça, leia o manual COLOR BASIC.

PROGRAMA DE DISCO

Alguns de nossos produtos possuem o sistema de
armazenamento de dados em disco, sendo assim, os
dados ficam armazenados em um disco rígido, sendo
possível acessar os dados em qualquer momento.

Alguns de nossos produtos possuem o sistema de
armazenamento de dados em disco, sendo assim, os
dados ficam armazenados em um disco rígido, sendo
possível acessar os dados em qualquer momento.

Alguns de nossos produtos possuem o sistema de
armazenamento de dados em disco, sendo assim, os
dados ficam armazenados em um disco rígido, sendo
possível acessar os dados em qualquer momento.

Alguns de nossos produtos possuem o sistema de
armazenamento de dados em disco, sendo assim, os
dados ficam armazenados em um disco rígido, sendo
possível acessar os dados em qualquer momento.

UMA COISA DE CADA VEZ

(ARQUIVO DE ACESSO SEQUENCIAL)

A fita cassete é fácil de usar e só existe uma maneira de ler ou gravar dados nela. Já o disco é mais complexo, pois existem várias formas de arquivar dados nele.

Neste capítulo e no seguinte, mostraremos como escrever um programa que grava dados num arquivo sequencial: o tipo de arquivo mais fácil de se criar. É muito parecido com um arquivo em fita cassete e no capítulo 7 mostraremos o arquivo de acesso direto, que é uma forma alternativa de armazenamento de dados em disco.

Ao mostrarmos como gravar informações no disco usaremos, frequentemente, as palavras arquivo e diretório. Estes conceitos foram explicados no capítulo 2, mas vamos relembra-los agora:

Tudo o que você armazenar num disco, precisa estar dentro de um arquivo e este tem que possuir um nome. O computador guardará a posição do arquivo no diretório do disco e se você quiser, por exemplo, armazenar o nome de seus amigos, você pode colocá-los num arquivo em disco chamado "AMIGOS". O diretório deste disco conterà, então, a posição em que foi gravado este arquivo.

Há certamente uma boa razão para tudo isso. Usando o sistema de arquivamento, o computador é capaz de achar imediatamente qualquer arquivo no disco.

COMO CRIAR UM ARQUIVO EM DISCO

Vamos supor que você queira guardar seus cheques no disco:

CHEQUES
DR. PEDRO
MERCADO
ESCOLA
ESCRITORIO

Vamos começar com um pequeno programa que grava o primeiro cheque ("DR. PEDRO") no disco. Insira um disco "formatado" na unidade.

(Se você possuir mais de uma unidade, coloque o disco na unidade Ø)

NOTA: No capítulo 2 mostramos como formatar um disco. (Digite DIR ENTER caso não se lembre se o disco foi formatado). No capítulo 1 explicamos os números das unidades de disco.

Digite então:

```
1Ø OPEN "O", # 1, "CHEQUES/DAT"  
2Ø WRITE # 1, "DR. PEDRO"  
3Ø CLOSE # 1
```

Execute o programa digitando RUN ENTER. Você ouvirá o motor da unidade de disco e verá a lâmpada vermelha se acender e assim o computador estará trabalhando, fazendo várias tarefas.

Primeiro ele abre (OPEN) a comunicação com o disco, de forma que você possa mandar seus cheques para o disco. Depois disso ele procura um lugar vazio para guardar os cheques e anota a posição deste arquivo no diretório.

Tudo isso ocorre a linha 1Ø. Note o significado de "O", # 1 e "CHEQUE/DAT":

1. # 1 é uma área especial na memória chamada # 1.
O "buffer" se comunica com a unidade de disco. A linha 1Ø abre (OPEN) este "buffer".
(Se você usou fita cassete, você deve se lembrar que o buffer # -1 se comunica com o gravador).
2. "O" é a letra "O" e não o número zero. Ela significa saída (OUTPUT) e informa ao computador que o buffer # 1 vai mandar dados para o disco.
3. "CHEQUES/DAT" é o nome do arquivo. O diretório do disco guarda este nome e a localização inicial e final deste arquivo.

Na linha 2Ø, o computador envia as palavras "DR. PEDRO" para o buffer # 1 que as escreve no disco.

Na linha 30 o computador fecha (CLOSE) a comunicação com o buffer # 1. Fazendo isso, ele:

- envia todos os dados que permaneciam no buffer # 1 para o arquivo em disco.
- anota no diretório onde termina o arquivo "CHEQUES/DAT"

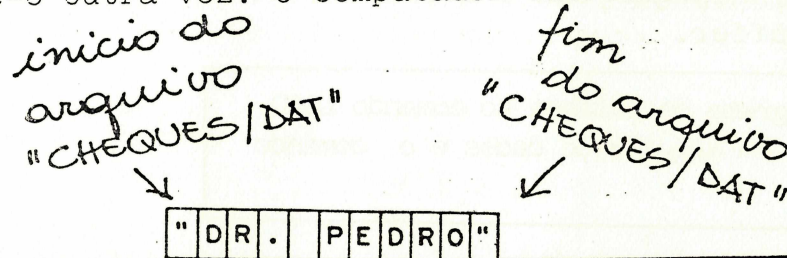
NOTA: O "buffer" armazena, temporariamente, os dados que o computador vai ler ou gravar no disco em blocos de 249 caracteres (bytes). Como o buffer # 1 só contém 9 caracteres ("DR PEDRO"), eles não são enviados ao disco sem que se feche o arquivo antes.

É muito importante fechar a comunicação com o buffer # 1. Por que? Bem, vamos deixar o buffer # 1 aberto. Elimine a linha 30 e execute (RUN) o programa várias vezes.

O programa parece funcionar da mesma forma cada vez que você o executa. Isto porque sempre que você executar (RUN) ou carregar (LOAD) um programa, o computador fecha (CLOSE) automaticamente a comunicação com os "buffers" deixados abertos.

Vamos supor agora que você colocou um disco na unidade e executou (RUN) ou carregou (LOAD) um programa. O computador fechará, automaticamente, a comunicação com o buffer # 1. Fazendo isso, ele envia as últimas informações contidas no buffer para o novo disco (pensando que ainda é o velho). Isso provavelmente truncará o conteúdo de ambos os discos.

Agora que você está prevenido da importância da linha 30, introduza-a novamente no seu programa e execute-o outra vez. O computador escreverá no disco o seguinte:



NOTA: Você gostou da forma como representamos o disco? O Arquivo "CHEQUES/DAT" é formado pelas palavras "DR. PEDRO. O diretório do disco anota o começo e o final deste arquivo.

Pode verificar se o computador executou esta tarefa, listando o diretório do disco. Você ainda se lembra como fazê-lo? (Digite DIR ENTER).

Pelo fato deste programa enviar seus dados para o arquivo em disco, nós o chamaremos de programa de saída (OUTPUT).

COMO LER UM ARQUIVO DO DISCO

Para ler os dados que estão no disco para a memória, é preciso fazer um programa de entrada (INPUT). Apague o programa de saída que está na memória digitando NEW ENTER. Depois disto, digite e execute o seguinte programa de entrada:

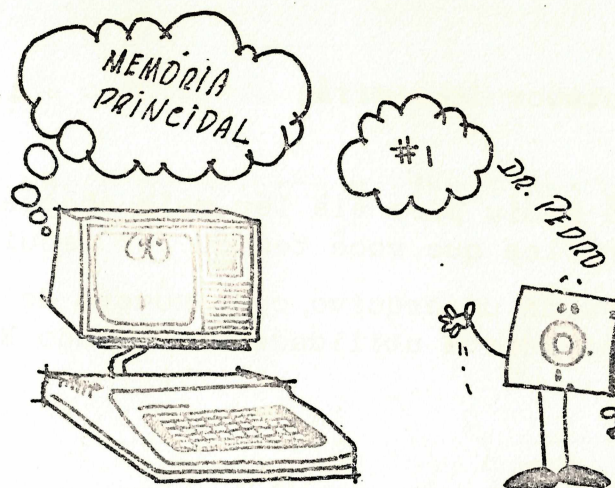
```
100 OPEN "I", # 1, "CHEQUES/DAT"  
110 INPUT # 1, A$  
120 PRINT A$  
130 CLOSE # 1
```

A linha 100 abre novamente a comunicação com o buffer # 1. Desta vez ela é aberta (OPEN) para entrada ("I" - INPUT). O computador busca no diretório a posição do arquivo "CHEQUES/DAT".

Na linha 110, o computador lê (INPUT), o primeiro item de dado do arquivo "CHEQUES/DAT", dando-lhe o nome A\$. A linha 120 imprime (PRINT) A\$.

Finalmente a linha 130 fecha a comunicação com o buffer # 1. Ao fazer isso, o computador lê os dados remanescentes no buffer.

NOTA: Você pode comparar um programa de entrada ao comando LOAD.
Um programa de entrada lê um arquivo de dados e o comando
LOAD lê um programa.



UM CHEQUE DE CADA VEZ

Até o momento já usamos um programa de entrada e outro de saída. Vamos agora combiná-los num único programa. Digite:

```
10 OPEN "O", # 1, "CHEQUES/DAT"
20 WRITE # 1, "DR. PEDRO"
30 CLOSE # 1
100 OPEN "I", # 1, "CHEQUES/DAT"
110 INPUT # 1, A$
120 PRINT A$
130 CLOSE # 1
```

Acrescente as linhas abaixo e execute (RUN) o programa:

```
25 WRITE # 1, "MERCADO"
115 INPUT # 1, B$
120 PRINT A$, B$
```

As linhas de 10 a 30 gravam dois cheques no arquivo:

*início do
arquivo
"CHEQUES/DAT"*

*fim
do arquivo
"CHEQUES/DAT"*

"DR. PEDRO" "MERCADO"

As linhas de 100 a 130 lêem os cheques para a memória. Tente gravar mais que dois cheques. Mude as linhas 115 e 120:

```
115 PRINT A$
120 GOTO 110
```


e execute (RUN) o programa ... O computador imprimirá:

?IE ERROR IN 11Ø

O computador está informando que você pediu para ele ler mais cheques do que existem no arquivo. Tecnicamente o erro IE significa que você tentou ler depois do fim do arquivo.

Este erro cria dificuldades na leitura de um arquivo, cujo número de registros não conhecemos. Mostramos este erro para você entender a utilidade do comando EOF. Digite:

```
1Ø5 IF EOF (1) = -1 THEN 13Ø
12Ø GOTO 1Ø5
```

e execute o programa. O comando EOF verifica se foi atingido o fim do buffer #1 (o número entre parênteses). Se EOF (1) for igual a 1 você atingiu o fim do buffer e se EOF (1) for igual a 0, ainda existem dados a serem lidos.

Acrescentando a linha 1Ø5 ao programa, o computador verifica se foi alcançado o fim do arquivo. A linha 13Ø fecha a comunicação com este arquivo.

DETALHES

Até agora foi fácil manipular o arquivo "CHEQUES/DAT", mas ele não apresenta muita utilidade. Você provavelmente gostaria de acrescentar mais detalhes:

CHEQUES		
A ORDEM DE	VALOR	DESPESA
DR. PEDRO	45.50	MEDICO
MERCADO	22.30	COMIDA
ESCOLA	27.90	MENSALIDADE
ESCRITORIO	13.40	CONTAS

Altere as linhas 25 e 115 e acrescente estas novas linhas ao programa:

```
25 WRITE # 1, 45.50
27 WRITE # 1, "MEDICO"
110 INPUT # 1, A$,B,C$
115 PRINT A$,B,C$
```

Liste (LIST) o programa. Esta é a forma como ele aparecerá no vídeo:

```
10 OPEN "O", # 1, "CHEQUE/DAT"
20 WRITE # 1, "DR. PEDRO"
25 WRITE # 1, 45.50
27 WRITE # 1, "MEDICO"
30 CLOSE # 1
100 OPEN "I", # 1, "CHEQUES/DAT"
105 IF EOF (1) = -1 THEN 130
110 INPUT # 1, A$, B, C$
115 PRINT A$, B, C$
120 GOTO 105
130 CLOSE # 1
```

Execute-o

UM PROGRAMA BOM E SEGURO

Como faria para guardar uma lista completa de cheques? Se continuar aumentando lentamente o número de linhas, este programa logo se tornará intolerável.

A seguir apresentaremos um programa que pergunta quais os dados que você deseja arquivar. Vamos gravá-los no disco e os leremos para a memória.

Apague (NEW) a memória e digite:


```

5 CLS
10 OPEN "O", # 1, "CHEQUES/DAT"
20 INPUT "CHEQUE PAGO A ORDEM DE : "; A$
30 IF A$ = "" THEN 80
40 INPUT "VALOR : CR$" ; B
50 INPUT "DESPESA : " ; C$
60 WRITE # 1, A$, B, C$
70 GOTO 20
80 CLOSE # 1
90 CLS
100 PRINT "SEUS CHEQUES FORAM ARMAZENADOS"
105 PRINT "NO DISCO"
110 PRINT "PRESSIONE <ENTER> PARA LER OS"
115 INPUT "CHEQUES DO DISCO" ; A$
120 OPEN "I", # 1, "CHEQUES/DAT"
130 IF EOF(1) = -1 THEN 170
140 INPUT # 1, A$, B, C$
150 PRINT A$; B; C$
160 GOTO 130
170 CLOSE # 1

```

Execute (RUN) o programa. Forneça os cheques e quando terminar, pressione a tecla ENTER
em resposta à pergunta CHEQUES PAGOS A ORDEM DE:?

```

CHEQUE PAGO A ORDEM DE :? BANCO NOVO ENTER
VALOR : Cr$? 247.50 ENTER
DESPESAS :? CARRO ENTER
CHEQUE PAGO A ORDEM DE :? ENTER

```

```

SEUS CHEQUES FORAM ARMAZENADOS
NO DISCO
PRESSIONE< ENTER>PARA LER OS
CHEQUES DO DISCO ENTER

BANCO NOVO 247.50 CARRO

```

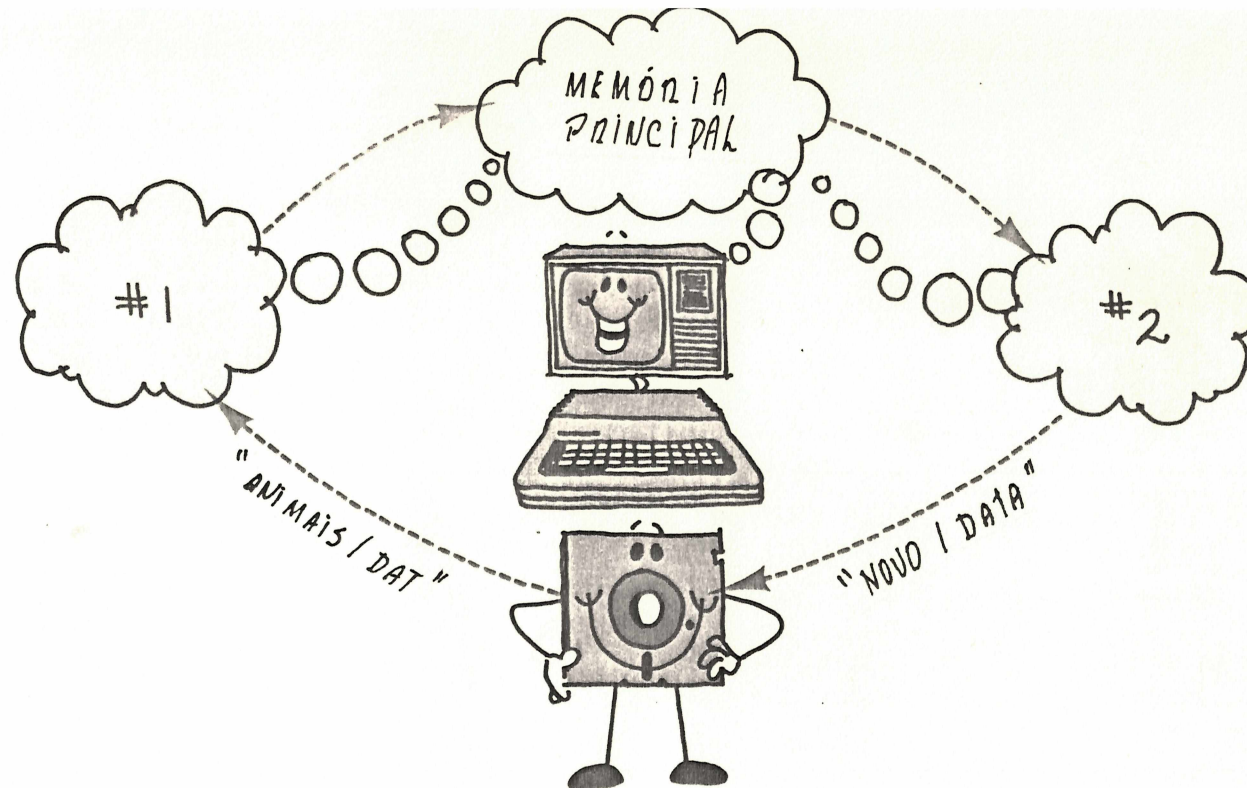

EXERCÍCIO DE PROGRAMAÇÃO Nº 5.1

Escreva um programa que só imprima os cheques gastos com o carro.

As respostas aos exercícios de programação encontram-se no APÊNDICE A.

VERIFICAÇÃO DE APRENDIZADO

1. O que é buffer #1?
2. Por que é preciso abrir (OPEN) um arquivo em disco?
3. Por que é preciso fechar (CLOSE) um arquivo em disco?
4. Qual é a diferença entre um arquivo aberto para entrada ou saída de dados?



COMO ALTERAR DADOS

(COMO ATUALIZAR UM ARQUIVO DE ACESSO SEQUENCIAL)



COMO ALTERAR DADOS

(COMO ATUALIZAR UM ARQUIVO DE ACESSO SEQUENCIAL)

COMO ALTERAR DADOS

(COMO ATUALIZAR UM ARQUIVO DE ACESSO SEQUENCIAL)

Tudo que você colocar ou retirar de um disco passa por uma área especial de memória, chamada "buffer". Quando mostramos como gravar dados em uma fita cassete no manual COLOR BASIC, não falamos sobre os "buffers". Isso não foi preciso porque só existe um buffer que se comunica com o gravador cassete (buffer #1).

Com o COLOR DOS você pode usar até 15 buffers. Isso significa que é possível ter até 15 áreas na memória comunicando-se com 15 arquivos diferentes ao mesmo tempo.

A razão pela qual abordamos agora este assunto, é porque queremos mostrar como modificar informações gravadas em um arquivo. Para fazer isso é muito útil usar dois "buffers".

NOTA: No capítulo 10 mostraremos como melhorar os programas usando vários "buffers".

Digite o programa abaixo:

```
10 OPEN "0", #1, "ANIMAIS/DAT"
20 WRITE #1, "CAVALO"
30 WRITE #1, "VACA"
40 CLOSE #1
```

Execute-o (RUN). Agora vamos supor que você deseje trocar "VACA" por "GIRAFA. Primeiro é preciso ler os dados para a memória com um programa de entrada. Apague a memória e digite NEW (ENTER):

```
10 OPEN "I", #1, "ANIMAIS/DAT"
20 IF EOF = -1 THEN 110
30 INPUT 1, A$
40 CLS = PRINT @ 106, "ITEN" : " A$;
100 GOTO 20
110 CLOSE #1
```


Depois disso é preciso acrescentar novas linhas de programação, que permitirão alterar um dos itens e gravar a mudança no arquivo. Digite:

```
50 PRINT @ 451, "PRESSIONE <ENTER> SE NAO HOUVER MUDANCA";  
60 PRINT @ 263 "MUDANCA :";  
70 INPUT X$  
80 IF X$ = "" THEN X$ = A$  
90 WRITE # 1, X$
```

Execute (RUN) o programa. Quando o computador chegar à linha 90, ele imprimirá:

?FM ERROR IN 90

Liste (LIST) o programa. A linha 10 abre o buffer # 1 para entrada (INPUT) de dados. Ali
na 90, entretanto, está tentando transmitir (OUTPUT) dados para o buffer # 1. O computa
não pode transmitir (OUTPUT) dados para um buffer aberto para entrada (INPUT).

Neste momento é que o buffer adicional se faz necessário. Para gravar modificações no dis
co, deve-se abrir um outro buffer para saída (OUTPUT).

Acrescente as linhas abaixo:

```
15 OPEN "O", # 2 "NOVO/DAT"  
90 WRITE # 2, X$  
120 CLOSE # 2
```

Execute (RUN) o programa. Troque "VACA" por "GIRAFA". O programa deverá aparecer agora da seguinte maneira no vídeo:


```

10 OPEN "I", # 1, "ANIMAIS/DAT"
15 OPEN "O", # 2, "NOVO/DAT"
20 IF EOF(1) = -1 THEN 110
30 INPUT # 1, A$
40 CLS : PRINT @ 106, "ITEM : " A$;
50 PRINT @ 451, "PRESSIONE<ENTER>SE NAO HOUVER MUDANCA";
60 PRINT @ 263, "MUDANCA .:";
70 INPUT X$
80 IF X$ = "" THEN X$ = A$
90 WRITE # 2, X$
100 GOTO 20
110 CLOSE # 1
120 CLOSE # 2

```

A linha 10 abre o buffer #1 para entrada (INPUT) de dados no arquivo. "ANIMAIS/DAT". A linha 15 abre o buffer #2 para saída (OUTPUT) de dados para o arquivo em disco "NOVO/DAT".

A linha 30 lê (INPUT) A\$ do buffer #1. A linha 70 permite que você forneça (INPUT) X\$ que substituirá A\$.

A linha 90 transmite (OUT PUT) X\$ para o buffer # 2 que por sua vez grava-o no arquivo em disco "NOVO/DAT".

A linha 110 fecha (CLOSE) a comunicação com o buffer #1 e a linha 120 fecha (CLOSE) a comunicação com o buffer # 2.

Você possui agora, dois arquivos "ANIMAIS/DAT", contendo os dados antigos e "NOVO/DAT" contendo os novos dados. Acrescente as linhas abaixo e execute o programa novamente:

```

130 KILL "ANIMAIS /DAT"
140 RENAME "NOVOS/DAT" TO "ANIMAIS/DAT"

```

O antigo arquivo "ANIMAIS/DAT" foi eliminado do disco e o arquivo "NOVO/DAT" mudou de nome chamando-se agora "ANIMAIS/DAT". Para certificar-se de que o arquivo está atualizado, salve (SAVE) este programa no disco, apague (NEW) a memória e digite e execute o programa abaixo:


```

10 OPEN "I", # 1, "ANIMAIS/DAT"
20 IF EOF (1) = - 1 THEN 60
30 INPUT # 1, A$
40 PRINT A$
50 GOTO 20
60 CLOSE # 1

```

Entendeu tudo? Então, tente fazer os exercícios a seguir:

EXERCÍCIO DE PROGRAMAÇÃO 6.1

Escreva um programa que permita acrescentar outros nomes de animais no arquivo "ANIMAIS/DAT".

Observação: Estes nomes deverão ser colocados ao final do arquivo:

EXERCÍCIO DE PROGRAMAÇÃO 6.2

Escreva um programa que permita eliminar nomes de animais do arquivo "ANIMAIS/DAT".

O próximo exercício será escrever um programa de caderneta de endereços. Começaremos com as linhas a seguir, que permitem cadastrar os nomes, endereços e números de telefone de seus amigos:

```

80 OPEN "O", # 1, "AMIGOS/DAT"
90 GOSUB 430
100 IF N$ = "" THEN CLOSE # 1: END
110 WRITE # 1, N$, E$, T$
120 GOTO 90
430 CLS: PRINT "PRESSIONE <ENTER> QUANDO TERMINAR" : PRINT
440 INPUT "NOME DO AMIGO : "; N$
450 IF N$ = "" THEN 480
460 INPUT "ENDEREÇO : "; E$
470 INPUT "NUMERO DO TELEFONE : "; T$
480 RETURN

```


Agora termine-o resolvendo o exercício de programação 6.3. Pode parecer difícil, mas acreditamos que você já tenha condições de fazê-lo. Lembre-se: ninguém está observando. Se não conseguir, veja a resposta no APÊNDICE A.

EXERCÍCIOS DE PROGRAMAÇÃO Nº 6.3

Escreva um programa em que você possa:

1. Ver os nomes, endereços e números de telefone de seus amigos.
2. Mudar o endereço de um dos amigos.
3. Acrescentar novos amigos.
4. Eliminar amigos.

Tudo isso funciona bem com poucos dados, mas como seria com um arquivo grande !!! Como seria se você tivesse 500 amigos no seu arquivo "AMIGOS/DAT" e quisesse mudar o endereço do amigo número 453?

O método ainda seria o mesmo. Você precisaria ler todos os 500 amigos do arquivo e então gravá-los, novamente, em outro. Todo esse trabalho só para mudar um endereço ... Deve existir uma forma mais fácil de se fazer isso, não acha?

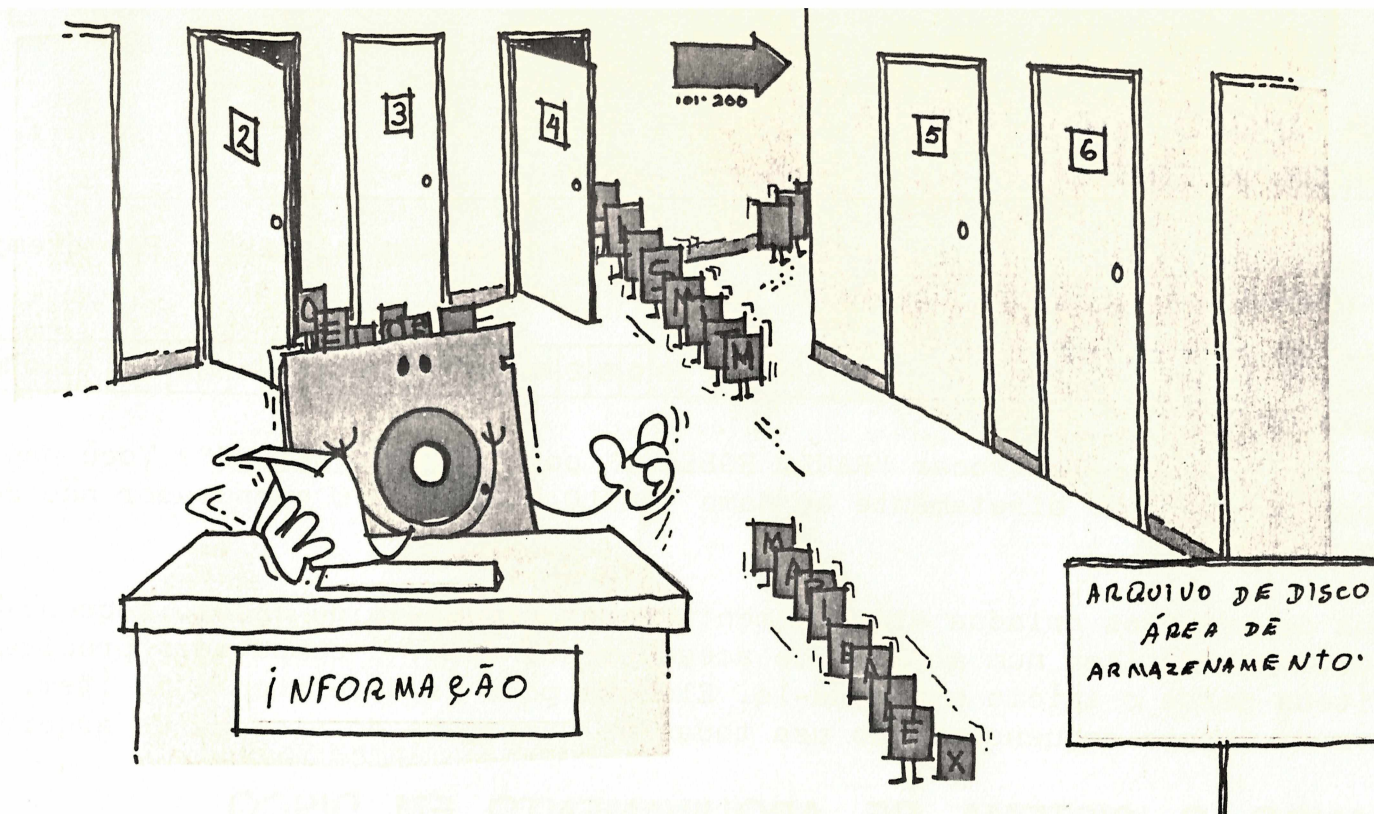
A forma mais fácil chama-se "programar usando arquivos de acesso direto". É mais fácil e mais rápido atualizar este tipo de arquivo, mas em muitos casos ele ocupa mais espaço no disco.

A escolha é sua. Abordaremos o acesso direto no próximo capítulo.

NOTA: Até agora só mostramos pequenos programas. Existem muitas maneiras de aperfeiçoá-los. Veja os "programas simples" no apêndice C, para ter idéias.

VERIFICAÇÃO DE APRENDIZADO

1. Por que não é possível ler (INPUT) e gravar (OUTPUT) dados ao mesmo tempo num "buffer"?
2. É possível ler (INPUT) dados de um arquivo aberto para gravação ("O" - OUTPUT)?



UMA VISÃO MAIS DIRETA

(ARQUIVO DE ACESSO DIRETO)

ARQUIVO DE ACESSO DIRETO

início do
arquivo

"	"
---	---

* NOME/DAT" →

fin do
arquivo
"NORIS/DAT"

Todos os arquivos criados até o momento foram "arquivos de acesso sequencial". Para achar um determinado item num arquivo de acesso sequencial, o computador precisa pesquisar todos os itens desde o início até achá-lo. Ele não pode ir diretamente ao item. Resumindo, um arquivo de acesso sequencial não usa todas as vantagens do sistema de arquivamento em disco.

No capítulo 2 mostraremos que ao "formatarmos" um disco, criamos o sistema de arquivamen
to. Na nossa analogia, as gavetas do arquivo são as trilhas do disco e as pastas, os seto
res. Você pode usar as trilhas e os setores para achar rapidamente qualquer informação num
disco.

A seguir mostraremos como será este novo tipo de arquivo:

unício do
arquivo
"NOMES/DAT"

registro l

[illegible]

registro 2

[illegible]

registro 3

Fim do arquivo
← "NOMES.DAT"

Como todos os registros têm o mesmo comprimento (o tamanho de um setor), o computador po
de ir diretamente a "PAULO ROBERTO". Tudo que ele precisa fazer é contar até o segundo re
gistro.

Nós chamamos isso de "arquivo de acesso direto". Por acesso direto, queremos dizer que po de-se acessar qualquer registro no arquivo sem ter que passar pelos demais.

O arquivo de acesso direto possui um problema: cada registro tem o comprimento de um setor-256 bytes. Como o byte pode guardar um caracter de dado, cada registro tem capacidade para 256 caracteres.

A nossa representação do registro não está correta. Se mostrássemos todos os espaços vazios de cada registro, eles teriam que ser quase dez vezes maior. Nós simplesmente não temos espaço suficiente para representá-los numa página.

Se você for um iniciante, todo este espaço vazio não vai incomodá-lo. Um disco vazio suporta até 612 registros de 256 bytes cada. Mais tarde, quando você dominar melhor a programação, vai querer armazenar mais registros num arquivo em disco. Nesta ocasião vá ao capítulo 9 onde mostraremos como criar registros menores.

COMO COLOCAR UM REGISTRO NO DISCO

Chega de teoria! Vamos colocar um registro num arquivo. Como será um arquivo de acesso direto, não precisamos começar pelo primeiro registro, começaremos pelo segundo.

Apague a memória e digite:

```
10 OPEN "D", # 1, "NOMES/DAT"  
20 WRITE #1, "PAULO ROBERTO"  
30 PUT #1, 2  
40 CLOSE #1
```

O programa parece-lhe familiar... exceto a palavra PUT na linha 30 e o "D" na linha 10. Veremos mais tarde esta novidades.

Vamos agora acrescentar algumas linhas para que o computador possa ler este registro para a memória principal. Digite:

```
34 GET #1, 2  
36 INPUT #1, A$  
38 PRINT A$
```

Repare que na linha 34 surgiu uma nova palavra:—GET. Hmmmm... você tem alguma idéia? Vamos dar uma olhada em todo o programa:

```
10 OPEN "D", #1, "NOMES/DAT"  
20 WRITE #1, "PAULO ROBERTO"  
30 PUT #1, 2  
34 GET #1, 2  
36 INPUT #1, A$  
38 PRINT  
40 CLOSE #1
```

Execute-o... Você ouvirá um barulho vindo da unidade de disco. O computador está escrevendo "PAULO ROBERTO" no arquivo em disco e em seguida lendo-o de volta para a memória. Veja como isto é feito:

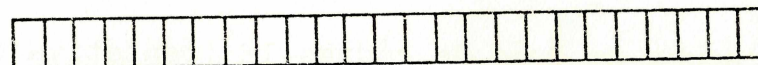
A linha 10 abre o buffer #1 que se comunica com o arquivo chamado "NOMES/DAT". Como dissemos nos dois últimos capítulos, o buffer #1 é uma das áreas de comunicação com o disco.

A comunicação foi aberta (OPEN) para "D". "D" significa direto — acesso direto. Neste tipo de arquivo não precisa indicar se está abrindo para leitura ou gravação. A letra "D" significa ambos.

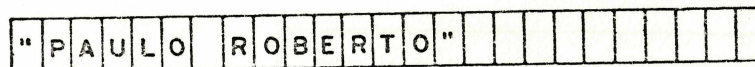
A linha 20 escreve "PAULO ROBERTO" no buffer #1. Como o arquivo "NOMES/DAT" foi aberto para acesso direto, "PAULO ROBERTO" permanecerá no buffer #1, até que o programa o envie para outro lugar.

A linha 30 faz isso. Ela coloca (PUT) o conteúdo do buffer #1 no segundo registro do arquivo.

*início do
arquivo
"NOMES/DAT"*



registro 1



registro 2

*fim do
arquivo
"NOMES/DAT"*

Agora "PAULO ROBERTO" não se encontra mais no buffer #1. Ele está no segundo registro do arquivo.

A linha 34 pega (GET) o segundo registro e coloca-o de volta no buffer #1. Agora "PAULO ROBERTO" está no arquivo e no buffer #1.

A linha 36 lê (INPUT) o registro do buffer #1 para a memória principal, dando-lhe o nome A\$. Agora "PAULO ROBERTO" está no arquivo e na memória principal. Ele não se encontra mais no buffer #1.

Como "PAULO ROBERTO" está na memória principal, a linha 38 pode imprimi-lo no vídeo.

NOTA: Nos programas em que eram usados arquivos de acesso sequencial (capítulos 5 e 6), você não precisou usar os comandos PUT e GET. O computador fazia isso automaticamente. O comando OPEN indicava se o buffer devia ser aberto para gravar (PUT) dados no arquivo ou para ler (GET) dados do arquivo.

Repare que nossa representação só mostrou dois registros do arquivo. Tente ler (GET) o terceiro registro. Digite:

34 GET #1, 3

e execute o programa ... O computador acusa um erro de código IE (tentativa de ler após o fim do arquivo). Isso ocorre porque o último registro gravado no arquivo é o de número 2. Por isso, o registro número 2 é o fim do arquivo.

NOTA: Se este erro não ocorreu, é porque já existe no seu disco, um arquivo chamado "NOMES/DAT" com três ou mais registros.

Para gravar (PUT) mais registros no arquivo, acrescente as linhas abaixo e execute nova
mente o programa:

31 WRITE # 1, "ELISABETE GOMES"

32 PUT # 1, 3

Agora o arquivo "NOMES/DAT" terá estes três registros: **registro 1**

INICIO DO ARQUIVO
"NOMES/DAT"

[illegible][illegible]

registro 2

"	E	L	I	S	A	B	E	T	E		G	O	M	E	S	"							
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	--	--	--	--	--	--	--

registro 3

FIM DO ARQUIVO
"NOMES/ DAT."

[illegible]

registro 1

"	P	A	U	L	O
R	O	B	E	R	T
O	"				

registro 2

"	E	L	I	S	A	B	E	T	E	G	O	M	E	S	"
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

registro 3

FIM DO ARQUIVO
"NOMES/DAT"

EXERCÍCIO DE PROGRAMAÇÃO 7.1

Altere as linhas 32 e 34 para que o computador grave "MARTA DA SILVA" no registro número 4.

LIDANDO COM LIXO

Você ainda não colocou nada no registro 1. Peça ao computador para ler o registro número 1 e veja o que ocorre. Digite a linha abaixo e execute o programa:

34 GET # 1, 1

Como o computador ainda não colocou nada no registro 1, o registro 1 contém um lixo qualquer que já estava lá.

Quando você pedir ao computador para pegar (GET) e ler (INPUT) o registro 1, ele vai pegar o lixo ou acusar um erro de código OS (fora do espaço da cadeia de caracteres). O erro OS significa simplesmente que o comprimento do lixo excedeu a 200 caracteres (bytes).

Os registros vazios contêm lixo até que você os preencha com alguma coisa, por isso é interessante colocar algum tipo de dado antes. Apague a memória e digite o programa a seguir:


```

10 OPEN "D", # 1, "NOMES/DAT"
20 FOR X = 1 TO 8
30 WRITE # 1, "VAZIO"
40 PUT # 1, X
50 NEXT X
60 CLOSE # 1

```

Execute-o. Este programa cria um arquivo chamado "NOMES/DAT" com 8 registros. Cada registro contém a palavra "VAZIO":

início do
arquivo
"NOMES/DAT"
✓

[illegible]

registro 1

[illegible]

registro 2

registro 3

registro 4

registro 5

registro 6

registro 7

registro 8

fim 
do arquivo
"NOMES/DAT"

Agora apague a memória e digite o seguinte programa:

```
1Ø OPEN "D", #1 "NOMES/DAT"
2Ø INPUT "REGISTRO N. (1-8)"; R
3Ø IF R > 8 THEN 2Ø
4Ø IF R < 1 THEN 13Ø
5Ø GET #1, R
6Ø INPUT #1, A$
7Ø PRINT A$ "-- E O NOME NO REGISTRO" R
8Ø INPUT "DIGITE UM NOVO NOME E PRESSIONE <ENTER>"; A$
9Ø IF A$= "" THEN 2Ø
10Ø WRITE #1, A$
11Ø PUT #1, R
12Ø GOTO 2Ø
13Ø CLOSE #1
```

Execute-o. Veja como todos os seus registros contêm a palavra "VAZIO". Agora você pode al
terar os dados quantas vezes quiser. (Para terminar o programa, forneça Ø na pergunta RE
GISTRO N. (1-8) ?

COMO LER TODOS OS REGISTROS DE UM ARQUIVO

Agora você já pode pedir ao computador que imprima todos os registros do arquivo "NOME/
DAT" com as respectivas posições no arquivo. Salve o programa anterior, apague a memória e
digite e execute o programa abaixo:

```
1Ø OPEN "D", #1, "NOMES/DAT"
2Ø R = 1
3Ø GET #1, R
4Ø INPUT #1, A$
5Ø PRINT A$ "-- ESTA NO REGISTRO" R
6Ø IF R = 8 THEN 9Ø
7Ø R = R + 1
8Ø GOTO 3Ø
9Ø CLOSE #1
```


8000 miles

Existirão ocasiões em que você não saberá o número exato de registros no arquivo. Altere a linha 60 e execute o programa:

"C. Var." novelas 6 1130 1arboz Lev de aln

REGISTRO

[illegible]

registro 1

VERIFICAÇÃO DE APRENDIZADO

1. O que são registros? Por que é preciso usá-los para acessar da dos diretamente?
2. O que são campos?
3. Qual é a diferença entre um arquivo de acesso sequencial e um acesso direto?
4. Por que é mais rápido atualizar um arquivo de acesso direto?

O PROGRAMA DE DISCO MAIS REQUINTADO

Depois de escrever vários programas que usam disco, você pode querer torná-los mais efi
cientes. Talvez até queira colocar mais dados no disco. Pode querer também economizar me
mória ou usar mais buffers.

Agora convidamos todas as pessoas ambiciosas a lerem esta seção. O assunto aqui aborda
do é mais avançado e técnico. Depois de lerem esta seção, terão todas as informações ne
cessárias para escrever programas mais requintados.

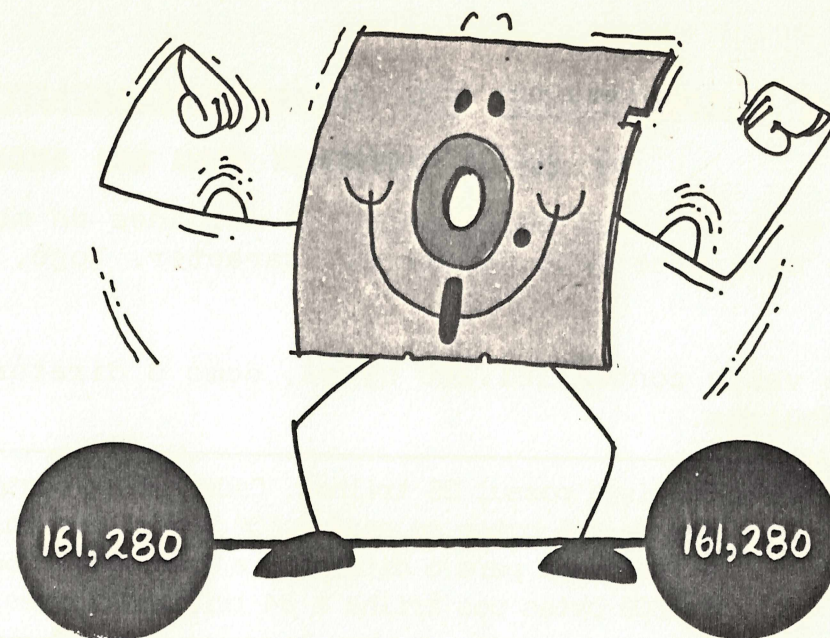
SEÇÃO

III

SEÇÃO

III

CAPITULO 8



QUAL A CAPACIDADE DE ARMAZENAMENTO DE UM DISCO ?

(O QUE O COMPUTADOR GRAVA NUM DISCO)

QUAL A CAPACIDADE DE ARMAZENAMENTO DE UM DISCO?

(O QUE O COMPUTADOR GRAVA NUM DISCO)

O disco está dividido em milhares de unidades de mesmo comprimento. Cada unidade é um "byte" e cada byte pode conter um caracter. Logo, a palavra MANUAL ocupa seis bytes no disco.

Um disco vazio contém 161.280 bytes, como o diretório ocupa 4.608, sobram 156.672 bytes para os arquivos.

NOTA: Um disco possui 35 trilhas. Cada trilha contém 18 setores de 256 bytes ou seja 4.608 bytes. Uma das trilhas é reservada para o diretório e isso deixa 156.672 bytes (4.608 bytes pos trilha x 34 trilhas) livres.

Será que isso significa que você pode usar todos os 156.672 bytes para armazenar dados? Provavelmente. Existem dois fatores que determinarão isso.

O primeiro fator depende da forma como o computador aloca espaço para os arquivos em disco. Ele cria um arquivo em múltiplos de 2.304 bytes. Cada grupo (cluster) de 2.304 bytes é chamado de grânulo.

Por causa disso, todos os seus arquivos em disco conterão múltiplos de 2.304 bytes. Se o seu arquivo ocupar 2.305 bytes, por exemplo, o computador reservará 2 grânulos para ele, ou sejam 4.608 bytes (2.304 x 2).

O computador aloca espaço desta maneira porque é a forma mais eficiente de se criar um arquivo. É muito difícil modificar este método. Só técnicos teriam interesse em fazê-lo. (Leia o capítulo 11, "Informações Técnicas" para maiores esclarecimentos).

O segundo fator que afeta o volume de dados que é possível colocar num disco, é o seu programa. Alguns programas são muito eficientes enquanto que outros colocam muitos espaços vazios nos arquivos.

Nos capítulos 9 e 10 vamos comparar outros tipos de programas. Todos gravarão os mesmos dados — 5, "TEL"—16 e "PAPEL"—num arquivo em disco chamado "ESCRIT/DAT". A quantidade de espaços vazios que cada programa deixará no arquivo "ESCRIT/DAT", varia bastante.

O programa 1 usa o comando WRITE para colocar estes dados no disco. Digite e execute:

PROGRAMA 1

24 bytes

```
1Ø OPEN "0", #1, "ESCRIT/DAT"  
2Ø WRITE # 1, 5, "CANETA"  
3Ø WRITE # 1, -16, "PAPEL"  
4Ø CLOSE # 1
```

Existe uma maneira mais fácil de verificar se as linhas 2Ø e 3Ø gravaram dados no disco. Digite estas duas linhas exatamente como elas estão indicadas acima mas tirando # 1 de cada uma delas. Isso evitará que o computador escreva os dados no disco, (via,buffer #1). O computador os imprimirá no vídeo. Digite:

WRITE 5, "CANETA" (ENTER)

WRITE -16, "PAPEL" (ENTER)

Veja com atenção o que o computador imprimiu. Todos os espaços em branco e a pontuação con tam.

Repare como o computador imprime as duas cadeias de caracteres (CANETA E PAPEL). Ele colo ca duas aspas entre cada palavra e escreve os números (5 e -16) de forma diferente. Se o número for negativo, o computador coloca um sinal menos à sua frente e se for positivo, simplesmente coloca um branco antes do número.

Existem dois caracteres que foram digitados, mas que o computador não imprimiu no vídeo. São os dois caracteres (ENTER) que você digitou ao final de cada linha. Em vez de imprí los, o computador pulou para a próxima linha.

5, "CANETA"

OK

-16 "PAPEL"

OK

Quando o computador no disco, ele realmente escreve cada caracter (ENTER) exatamente como você digitou. A ilustração a seguir mostra o que o programa 1 escreve no disco. (Usamos as teriscos para representarmos o caracter (ENTER)).

*início do
arquivo
"ESCRIT/DAT" →*

5	.	"	C	A	N	E	T	A	"	*	-	1	6	.	"	P	A	P	E	L	"	*
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

*← fim do
arquivo
"ESCRIT/DAT"*

NOTA: Você quer ser mais precioso? O que o computador realmente escreve no disco são códigos binários. Cada caracter tem um código ASCII (veja o apêndice D) que o computador converte para um número binário.

Conte os caracteres. Cada (ENTER) (representado por um asterisco), vírgula e duas aspas, vale um caracter. Não esqueça o espaço em branco antes do 5. Você deve ter 24 caracteres. O programa 2 coloca 24 bytes no arquivo "ESCRIT/DAT".

Como o computador aloca espaço em grânulos, o arquivo "ESCRIT/DAT" vai ocupar 1 grânulo ou seja 2.304 bytes. Entretanto, para compararmos este programa com os demais, só consideraremos os 24 bytes que o programa 1 realmente gravou no arquivo "ESCRIT/DAT".

COMO O COMPUTADOR LÊ DADOS NO DISCO

Para ler dados do arquivo "ESCRIT/DAT", digite e execute o programa abaixo (apague a memória antes):

PROGRAMA - INPUT

```
1Ø CLS
2Ø OPEN "I", # 1, "ESCRIT/DAT"
3Ø IF EOF (1) = -1 THEN 8Ø
4Ø INPUT #1, A, B$
5Ø PRINT: PRINT "ITEM DE DADO:" A
6Ø PRINT "ITEM DE DADO: " B$
7Ø GOTO 3Ø
8Ø CLOSE #1
```

Ele leu os dados, mas não leu as aspas, as vírgulas e os espaços em branco que estavam entre dados

Para ver realmente o que o programa 1 escreveu no disco, você pode usar o programa "LINE INPUT". Primeiro salve (SAVE) o programa "INPUT" que está na memória. (Você vai usá-lo mais tarde).

Agora altere o programa "INPUT", transformando-o no programa "LINE INPUT". Elimine a linha 5Ø e altere as linhas 4Ø e 6Ø. Digite:

```
4Ø LINE INPUT #1, L$
5Ø
6Ø PRINT "LINHA DE DADO: " L$
```

e execute este novo programa... A linha 4Ø lê uma linha inteira do arquivo em vez de um único item de dado. Esta linha inclui tudo, até o caracter (ENTER) - letras, números, caracteres especiais, enfim, tudo.

No arquivo "ESCRIT/DAT", a primeira linha contém

5.	"	C	A	N	E	T	A	"
----	---	---	---	---	---	---	---	---

. A linha 4Ø atribuiu-lhe o nome L\$ e a linha 6Ø imprime L\$ no vídeo.

Depois disso o programa lê e imprime

-	1	5.	"	P	A	P	E	L	"
---	---	----	---	---	---	---	---	---	---

 que é a segunda e última linha do arquivo.

Este programa pode ser facilmente modificado para contar o número de bytes que existem no arquivo. Acrescente estas linhas ao programa e execute-o:

```
25 PRINT "ESTE ARQUIVO CONTEM:"
27 PRINT: PRINT: PRINT: PRINT
57 M$ = L$ + "*"
60 PRINT M$;
65 L = LEN (M$) + L
90 PRINT @ 394, L "BYTES"
```

A linha 57 acrescenta um asterisco a cada linha que representa o caracter ENTER.
A linha 65 determina o número total de caracteres (bytes) em cada linha.

A seguir apresentamos o programa "LINE INPUT" por inteiro:

PROGRAMA - LINE INPUT

```
10 CLS
20 OPEN "I", #1, "ESCRIT/DAT
25 PRINT "ESTE ARQUIVO CONTEM: "
27 PRINT: PRINT: PRINT: PRINT
30 IF EOF (1) = -1 THEN 80
40 LINE INPUT #1, L$
57 M$ = L$ + "*"
60 PRINT M$;
65 L = LEN(M$) + L
70 GOTO 30
80 CLOSE # 1
90 PRINT @ 394, L "BYTES"
```

Salve-o (SAVE). Ele será útil para compararmos o que os programas 2,3 e 4 gravam no arquivo em disco.

PRINT _ PARA FAZER ALTERAÇÕES

Até o momento nós só utilizamos o comando WRITE para gravar dados num arquivo em disco. Se você já usou outros tipos de BASIC, pode estar acostumado a usar o comando PRINT em vez do WRITE.

O COLOR DOS também permite que se faça isso. No entanto, o comando PRINT é bem mais difícil de ser usado. Se você não está acostumado a usá-lo, não se preocupe em aprender, pule diretamente para o Programa 4.

... você ainda esta conosco? Então elimine (KILL) o velho arquivo "ESCRIT/DAT", digitando:

KILL "ESCRIT/DAT" (ENTER)

Agora apague a memória e digite e execute o programa 2. Se quiser execute também os programas "INPUT" e "LINE INPUT".

Aqui está o Programa 2:

```
PROGRAMA      2
( 45 bytes)

1Ø OPEN "O", #1, "ESCRIT/DAT"
2Ø PRINT #1, 5, "CANETA"
3Ø PRINT #1, -16, "PAPEL"
4Ø CLOSE #1
```

As linhas 2Ø e 3Ø transferem (PRINT) os seus dados para o buffer #1. Para ver o que o programa 2 grava, digite:

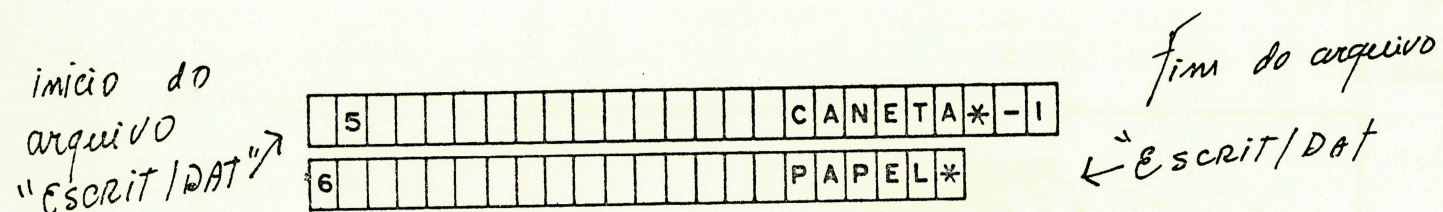
PRINT 5, "CANETA" (ENTER)
PRINT -16, "PAPEL" (ENTER)

Analisemos agora os espaços em branco. Vamos começar pelo primeiro — aquele que está antes do 5. Este espaço em branco significa que ele fez a mesma coisa que o comando WRITE. 5 é um número positivo.

Vejamos agora os outros espaços em branco... Sempre que o computador imprime (PRINT) um número, coloca um branco depois dele. Isto explica o primeiro espaço em branco depois do 5 e do - 16.

Como explicar os outros espaços? Lembra-se o que a vírgula faz num comando PRINT (Manual COLOR BASIC)? A vírgula faz com que o computador imprima os dados em colunas, inserindo brancos entre estas colunas.

O computador grava (PRINT) todos estes espaços em branco no seu arquivo em disco:



Conte todos os caracteres. O programa 2 grava 45 bytes no arquivo "ESCRIT/DAT".

NOTA: Se você não entendeu o que a vírgula faz num comando PRINT, digite algumas linhas usando o comando PRINT e vírgulas entre os dados:

```
PRINT 1, 2, 3, 4, 5, 6, 7, (ENTER)
```

```
PRINT "CAVALO", "VACA", "COELHO" (ENTER)
```


Você pode achar que todos estes espaços em branco inseridos pelo comando PRINT, dentro de seu arquivo, é uma perda de espaço. A forma de eliminar este desperdício é usar ponto e vírgula. Pode consultar mais uma vez o Manual COLOR BASIC para ver que o ponto e vírgula usa do no comando PRINT, comprime dos dados. Digite:

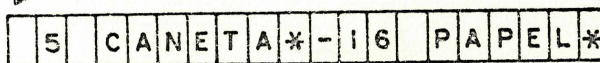
```
PRINT 5; "CANETA" (ENTER)
PRINT -16 "PAPEL" (ENTER)
```

Você pode comprimir os seus dados no disco da mesma forma. Limpe a memória e elimine (KILL) o velho arquivo "ESCRIT/DAT". Depois disso digite e execute o Programa abaixo:

```
1Ø OPEN "O", #1, "ESCRIT/DAT"
2Ø PRINT #1, 5; "CANETA"
3Ø PRINT #1, -16; "PAPEL"
4Ø CLOSE #1
```

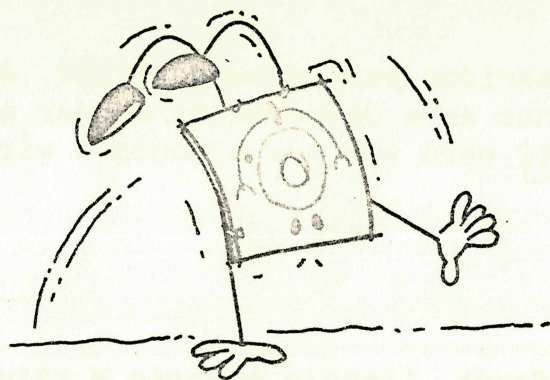
Isto é que o programa 3 grava (PRINT) no seu disco. (Use o programa "LINE INPUT" para verificar se é verdade):

início do arquivo
"ESCRIT/DAT"



fim do arquivo
"ESCRIT/DAT"

É bem mais eficiente não acha? Só ocupou 2Ø bytes. Só existem três espaços em branco neste arquivo. Um antes do 5 (para indicar que é um número positivo) e dois após o 5 e o -16, (para indicar que são números). Não há espaços em branco entre as cadeias de caracteres.



A PARTE MAIS DIFÍCIL

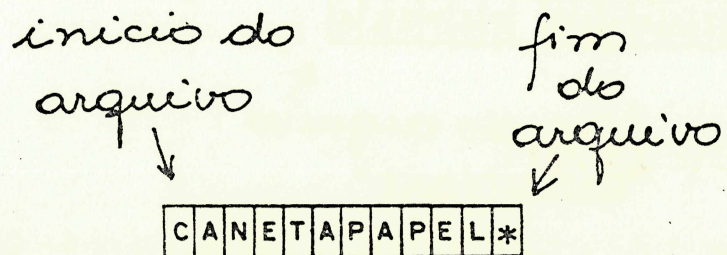
Existem tipos de impressões usando o comando PRINT, que são difíceis de entender. (Você foi avisado, não foi?) Digite:

```
PRINT "CANETA"; "PAPEL" (ENTER)
```

```
PRINT "MARTA, ANNA" (ENTER)
```

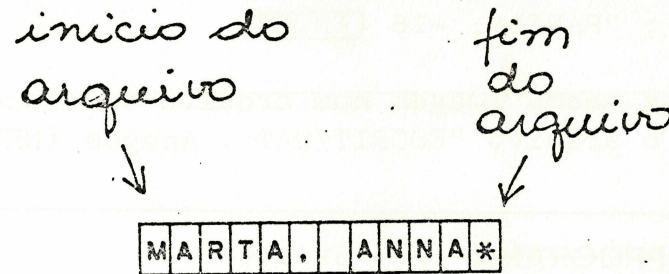
```
PRINT "CANETA", 5
```

A linha PRINT #1, "CANETA"; "PAPEL" (dentro de um programa) gravaria o seguinte num arquivo em disco:



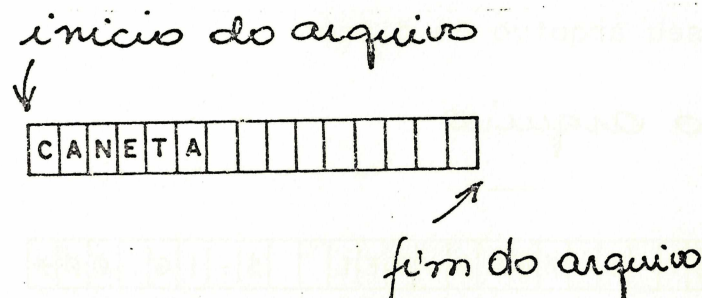
O computador leria para a memória CANETAPAPEL como um único item. A razão é que não existe nenhum delimitador (vírgula, duas aspas ou espaço), separando CANETA e PAPEL.

A linha PRINT #1, "MARTA, ANNA" (dentro de um programa) gravaria o seguinte num arquivo em disco:



O computador leria para a memória MARTA, ANNA, como sendo dois itens MARTA e ANNA. (Razão: o computador interpreta a vírgula como um delimitador).

A linha PRINT #1, "CANETA", 5 (dentro de um programa), gravaria o seguinte num arquivo em disco:



O computador leria

C	A	N	E	T	A							5
---	---	---	---	---	---	--	--	--	--	--	--	---

 (com todos os espaços em branco) para a memória, como um único item. (Razão: Embora o computador interprete espaços em branco como delimitadores, ele não o fará neste caso porque os espaços em branco estão antes de um número e após uma cadeia de caracteres).

UM ARQUIVO EM DISCO INTERESSANTE

PRINT USING é outro comando que pode substituir o WRITE. (Este comando foi apresentado no Manual COLOR BASIC EXTENDIDO). Digite:

PRINT USING "%\$ + ##. ## ; "CANETA", 5 (ENTER)

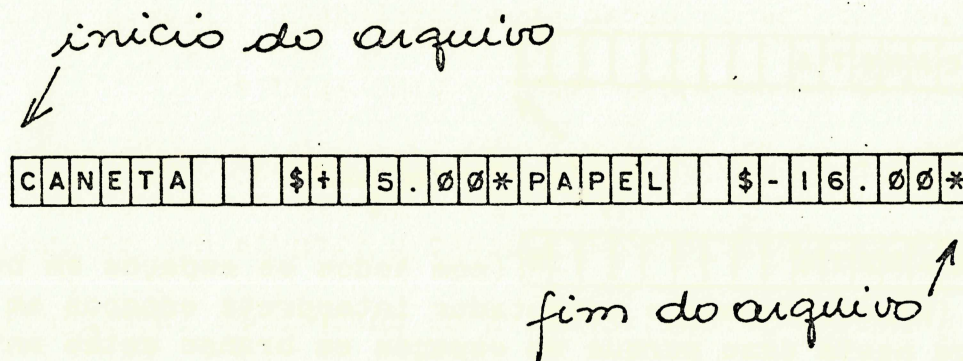
PRINT USING "%\$ + ##. ## ; "PAPEL", -16 (ENTER)

Você pode fazer o computador imprimir a mesma imagem num arquivo em disco, escrevendo o programa abaixo. Primeiro elimine (KILL) o arquivo "ESCRIT/DAT". Apague (NEW) a memória e digite e execute o programa:

PROGRAMA 4 (32 bytes)

```
1Ø OPEN "O", #1, "ESCRIT/DAT"
2Ø PRINT #1, USING "%$ + ##. ## "; "CANETA", 5
3Ø PRINT #1, USING "%$ + ##. ## "; "PAPEL", -16
4Ø CLOSE #1
```

Este programa imprime o seguinte no seu arquivo em disco:



NOTA: Existem seis espaços em branco entre os caracteres % nas linhas 2Ø e 3Ø. Contando os dois caracteres %, esta cadeia de caracteres (para imprimir caneta e papel), possui oito bytes.

Agora que os dados estão numa forma interessante de impressão, você pode ler e imprimí-los usando um programa "INPUT". Apague a memória e digite e execute o programa abaixo:


```
1Ø OPEN "I", #1, "ESCRIT/DAT"  
2Ø IF EOF (1) = -1 THEN 6Ø  
3Ø LINE INPUT #1, A$  
4Ø PRINT A$  
5Ø GOTO 2Ø  
6Ø CLOSE #1
```

Todos os arquivos criados neste capítulo foram de acesso sequencial. No próximo compararemos outros quatro programas, que gravarão as mesmas informações em arquivos de acesso direto.

VERIFICAÇÃO DE APRENDIZADO

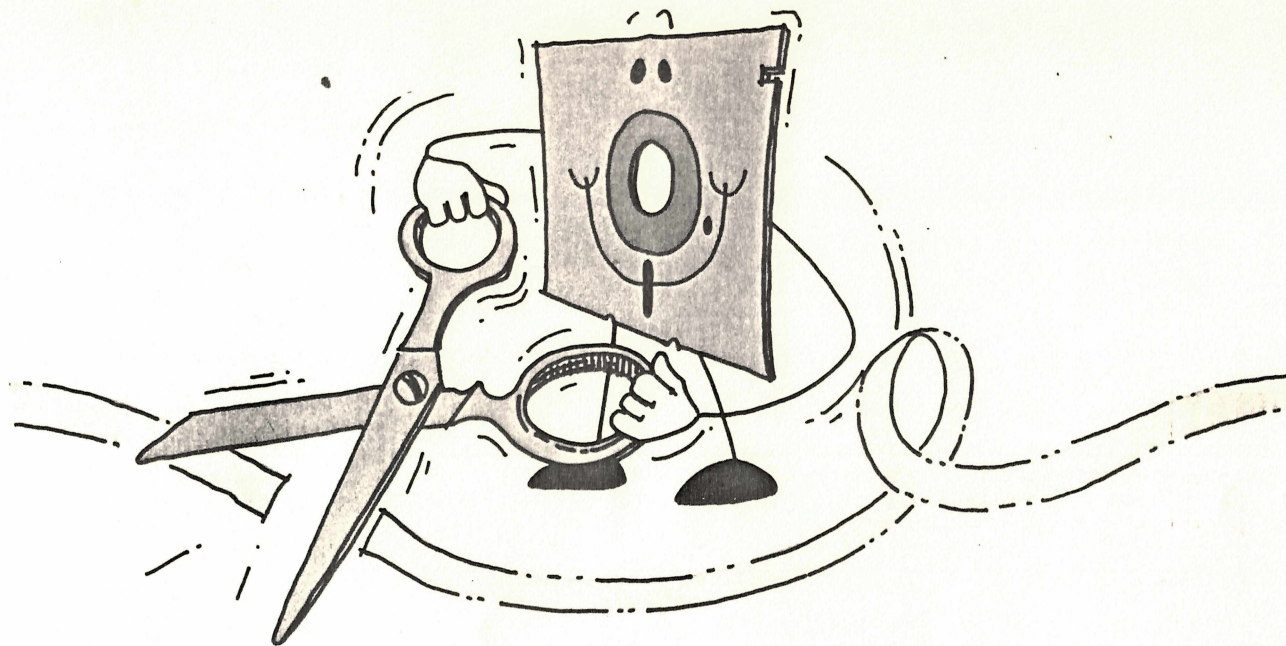
1. Qual o tamanho mínimo de um arquivo em disco? Por que não é possível criar um arquivo menor?
2. Como o computador grava (WRITE) números num arquivo em disco?
3. Como ele escreve as cadeias de caracteres (strings)?
4. Qual a diferença entre os comandos INPUT e LINE INPUT?
5. O que faz o computador quando encontra uma vírgula num comando PRINT?
6. O que faz o computador quando encontra ponto e vírgula num comando PRINT?
7. Como o computador imprime (PRINT) cadeias de caracteres (strings)?

25. 1000 10
26. 1000 10
27. 1000 10
28. 1000 10
29. 1000 10
30. 1000 10

VERIFICAÇÃO DE APRENDIZADO

1. Qual a função básica de um arquivo de dados? (1 ponto)
2. Como se chama o arquivo de dados que contém os dados de um sistema? (1 ponto)
3. Qual a função básica de um arquivo de dados? (1 ponto)
4. Qual a diferença entre os comandos INPUT e LINE INPUT? (1 ponto)
5. O que faz o comando PRINT? (1 ponto)
6. O que faz o comando PRINT#? (1 ponto)
7. Como se chama o arquivo de dados que contém os dados de um sistema? (1 ponto)

CAPÍTULO 9



ELIMINANDO O EXCESSO NOS ARQUIVOS DE ACESSO DIRETO

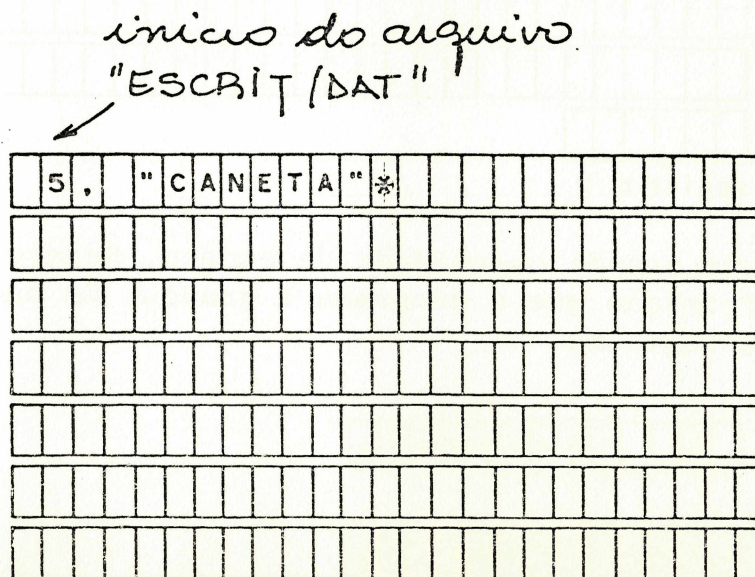
(COMO CRIAR UM ARQUIVO DE ACESSO DIRETO)



STENO GRAFIA DE MOVIMENTOS E POSICIONES
DEBIDO A LA FALTA DE ESPACIO EN LA PAGINA

Os arquivos de acesso direto frequentemente contêm muitos espaços vazios. O nosso primeiro exemplo é semelhante ao programa 1 do capítulo 8. No entanto, por ser um arquivo de acesso direto, ele alocará 512 bytes para o arquivo "ESCRIT/DAT":

Um programa de acesso direto coloca os dados dentro de registros. Cada registro tem 256 bytes. O programa 5 coloca dois registros no arquivo "ESCRIT/DAT". Logo este arquivo consumirá 2 x 256 ou 512 bytes.



registro l

registro 2

fim do arquivo
"ESCRIT.DAT"

Isto, obviamente, desperdiça uma grande quantidade de espaço. Repare que o computador grava em cada registro a mesma informação que o programa 1 gravou. Em outras palavras, são 12 bytes no primeiro e no segundo registro.

-	1	6	,		"	P	A	P	E	L	"	*
---	---	---	---	--	---	---	---	---	---	---	---	---

NOTA: Poderíamos ter usado os comandos PRINT e PRINT USING em vez de WRITE. O computador teria então colocado os seus dados em cada registro, usando os formatos correspondentes a PRINT e PRINT USING.

COMO ELIMINAR O EXCESSO

O programa 6 é idêntico ao programa 5, exceto número 12 que foi acrescentado ao final da linha 10. Isso indica ao computador que cada registro deve ter um comprimento de 12 bytes:

PROGRAMA 6 (21 bytes)

```
10 OPEN "D", #1, "ESCRIT/DAT", 12
20 WRITE #1, 5, "CANETA"
30 PUT #1, 1
40 WRITE #1, -16, "PAPEL"
50 PUT #1, 2
60 CLOSE #1
```

Isso realmente reduz o tamanho do arquivo:

*início do
arquivo
"ESCRIT/DAT"*

5	.	"	C	A	N	E	T	A	"	*
---	---	---	---	---	---	---	---	---	---	---

registro 1

-	1	6	.	"	P	A	P	E	L	"	*
---	---	---	---	---	---	---	---	---	---	---	---

registro 2

*fim do arquivo
"ESCRIT/DAT"*

Num arquivo de acesso direto, todos os registros têm o mesmo tamanho. (Isso foi explicado no capítulo 7). Se você não indicar o tamanho do registro, o computador assume 256 bytes.

Neste programa foram criados registros de 12 bytes por ser este o tamanho do maior registro. Se quiser, digite e execute o programa. (Certifique-se de ter apagado (NEW) a memória e eliminado (KILL) o velho arquivo "ESCRIT/DAT", primeiro). Depois de executar o programa 6, use o programa abaixo para ler o arquivo:

PROGRAMA DE LEITURA DIRETA

```
1Ø OPEN "D", #1, "ESCRIT/DAT", 12
2Ø R = R + 1
3Ø GET #1, R
4Ø INPUT #1, A, B$
5Ø PRINT "REGISTRO" R ":" A; B$
6Ø IF LOF(1) < > R THEN 2Ø
7Ø CLOSE #1
```

NOTA: Não é possível usar o "Programa LINE INPUT" para determinar quantos bytes este arquivo consome, porque o comando LINE INPUT não pode recuperar os espaços em branco que segue o caracter ENTER.

EFICIÊNCIA, EFICIÊNCIA...

Podemos obter mais eficiência ainda. No próximo programa o arquivo só ocupará 17 bytes. Apague a memória, elimine o velho arquivo "ESCRIT/DAT" e digite e execute o programa 7.

PROGRAMA 7 (17 bytes)

```

10 OPEN "D", #1, "ESCRIT/DAT", 9
20 FIELD #1, 3 AS A$, 6 AS B$
30 LSET A$ = "5"
40 LSET B$ = "CANETA"
50 PUT #1, 1
60 LSET A$ = "-16"
70 LSET B$ = "PAPEL"
80 PUT #1, 2
90 CLOSE #1

```

Neste programa surgiram duas novas palavras que comentaremos mais tarde. Vejamos primeiro o que o programa faz. Salve-o e leia os dados gravados com o programa abaixo:

PROGRAMA DE LEITURA POR CAMPO

```

10 OPEN "D", #1, "ESCRIT/DAT", 9
20 FIELD #1, 3 AS A$, 6 AS B$
30 R = R + 1
40 GET #1, R
50 PRINT "REGISTRO " R ":" A$, B$
60 IF LOF (1) < > R THEN 30
70 CLOSE #1

```

Usando os comandos FIELD e LSET, seu programa vai funcionar da mesma forma que qualquer programa de acesso direto. A diferença está no que FIELD e LSET colocam em cada registro:

*início do
arquivo
"ESCRIT/DAT"* →

5			C	A	N	E	T	A
---	--	--	---	---	---	---	---	---

registro 1

-	1	6	P	A	P	E	L
---	---	---	---	---	---	---	---

registro 2

← *fim do arquivo
"ESCRIT/DAT"*

Só o essencial foi armazenado. Veja como funciona o programa 7 ...

A linha 20 divide cada registro em dois campos. O primeiro campo chama-se A\$ e o segundo B\$. Estes dois campos terão sempre o mesmo tamanho nos registros. A\$ terá 3 bytes e B\$ 6 bytes.

Agora que já explicamos o que faz o comando FIELD, podemos colocar dados em cada campo. A linha 30 atribuiu (LSET) 5 ao campo A\$. Como o caracter 5 só ocupa 1 byte e existem 3 bytes no campo A\$, vão sobrar 2 espaços vazios ao final do 5.

Repare que foi preciso converter o número 5 para um "string", colocando-o entre aspas. Você não pode usar o comando LSET com números, é preciso convertê-los para string.

A linha 40 atribui (LSET) CANETA ao campo B\$. Neste caso todos os bytes de B\$ são ocupados.

A linha 50 grava A\$ e B\$ no registro 1. Este processo se repete para o registro 2 (-16 e PAPEL).

Vamos analisar agora o "Programa de leitura por campo". Repare que foi usado o comando FIELD. Tente executá-lo sem a linha 20 e veja o que ocorre ...

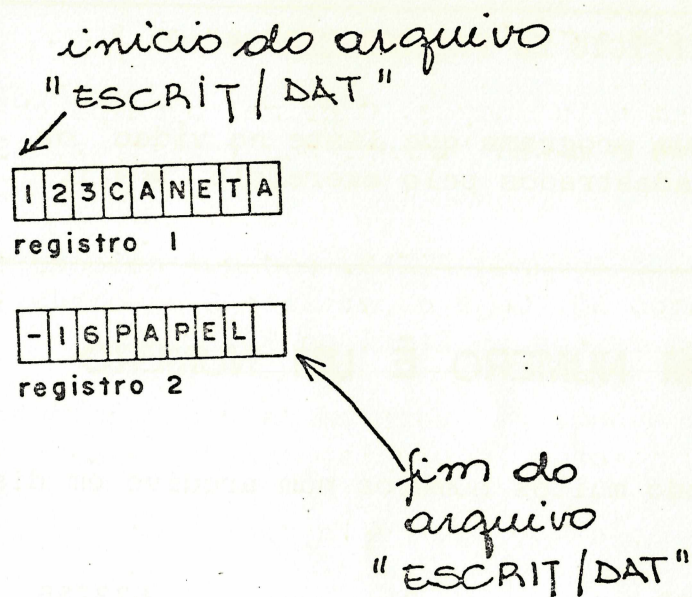
Sem o comando FIELD, o computador não sabe onde estão os dois campos A\$ e B\$. Sempre que você ler dados divididos em campos, use o comando FIELD no programa.

Você seria capaz de imaginar o que o computador faria se tentasse atribuir (LSET) uma cadeia de caracteres longa (STRING) - "123456789" por exemplo - a um dos campos? Carregue (LOAD) o programa 7, altere a linha 30 e execute-o. (Salve primeiro o "programa por campo" com a linha 20 corrigida)..:

```
30 LSET A$ = "123456789"
```

Agora carregue e execute o "programa de leitura por campo".

Como A\$ só tem 3 bytes, o computador só guardou os 3 primeiros bytes de "123456789" e truncou os demais caracteres.



Voltaremos a falar neste assunto mais tarde ... Antes de fazermos o próximo programa, tente escrever o seu próprio programa usando os comandos FIELD e LSET:

EXERCÍCIO DE PROGRAMAÇÃO #9.1

Escreva um programa que use um arquivo de acesso direto para gravar os endereços de seus amigos. Crie o registro com 47 bytes de comprimento e os seguintes campos:

- | | | |
|-------------|---|----------|
| 1. nome | - | 15 bytes |
| 2. endereço | - | 15 bytes |
| 3. cidade | - | 10 bytes |
| 4. uf | - | 2 bytes |
| 5. cep | - | 5 bytes |

EXERCÍCIO DE PROGRAMAÇÃO #9.2

Escreva um programa que liste no vídeo os dados cadastrados pelo exercício #9.1.

UM NÚMERO É UM NÚMERO

Vamos supor que você esteja gravando muitos números num arquivo em disco. Cada número pode ter um tamanho diferente:

-5.237632

31

673285

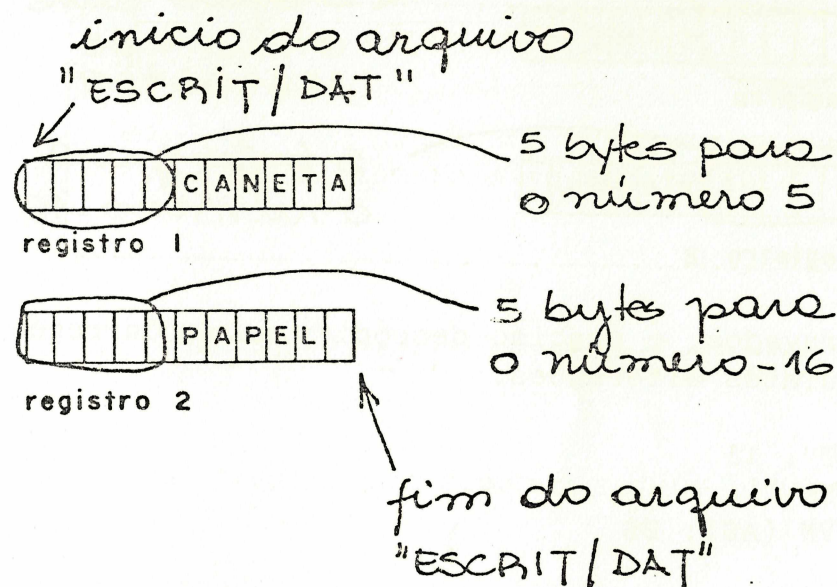
É muito importante que o computador não trunque nenhum desses valores.

O comando MKN\$ resolve este problema:

PROGRAMA 8 (20 bytes)

```
10 OPEN "D", #1, "ESCRIT/DAT", 11
20 FIELD #1, 5 AS A$, 5 AS B$
30 LSET A$ = MKN$ (5)
40 LSET B$ = "CANETA"
50 PUT #1, 1
60 LSET A$ = MKN$ (-16)
70 LSET B$ = "PAPEL"
80 PUT #1, 2
90 CLOSE #1
```

As diferenças entre este programa e o de nº 7 estão nas linhas 10, 20, 30 e 60. Este grava as seguintes informações no arquivo em disco:



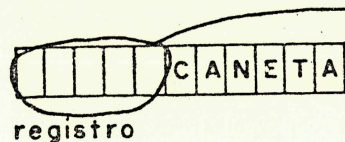
O comando NKN\$ converte um número num código string e, independente do seu tamanho, o comando NKN\$ sempre o converterá para um string de 5 bytes.

Por exemplo, altere a linha 30 para atribuir a A\$ um número com mais de 5 dígitos:

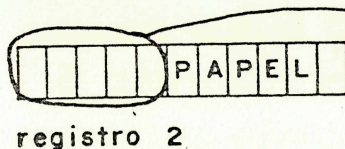
```
30 LSET A$ = NKN$ (123456789)
```

Apague a memória, elimine o arquivo "ESCRIT/DAT" e digite e execute o programa. Ele gravará as seguintes informações no arquivo em disco:

início do
arquivo
"ESCRIT/DAT" →



5 bytes para
o número
123456789



5 bytes para
o número -16

← fim do arquivo
"ESCRIT/DAT"

Para poder usar os números gravados, é preciso descodificá-los. Carregue o programa de letura por campo e faça as seguintes alterações:

```
1Ø OPEN "D", #1, "ESCRIT/DAT", 11
2Ø FIELD #1, 5 AS A$, 6 AS B$
3Ø PRINT "REGISTRO" R ":", CVN (A$); B$
```

e execute-o ... o comando CVN (na linha 5Ø) descodifica A\$ para o número que ele representa.

NOTA: O computador só interpreta os 9 primeiros dígitos de um número. Ele arredonda o restante.

EXERCÍCIO DE PROGRAMAÇÃO #9.3

Escreva um programa que grave, num arquivo de acesso direto, o nome e o número de habitantes de vários países. Crie o registro com 15 bytes e os seguintes campos:

1. País - 10 bytes
2. nº habitantes - 5 bytes

EXERCÍCIO DE PROGRAMAÇÃO #9.4

Escreva um programa que liste no vídeo os dados cadastrados pelo exercício #9.3.

VERIFICAÇÃO DE APRENDIZADO

1. Caso você não indique o tamanho do registro, quantos bytes conterà cada um deles?
2. Por que é preciso usar o comando FIELD junto com o comando LSET?
3. Quantos bytes ocupará um número codificado pelo comando MKN\$?

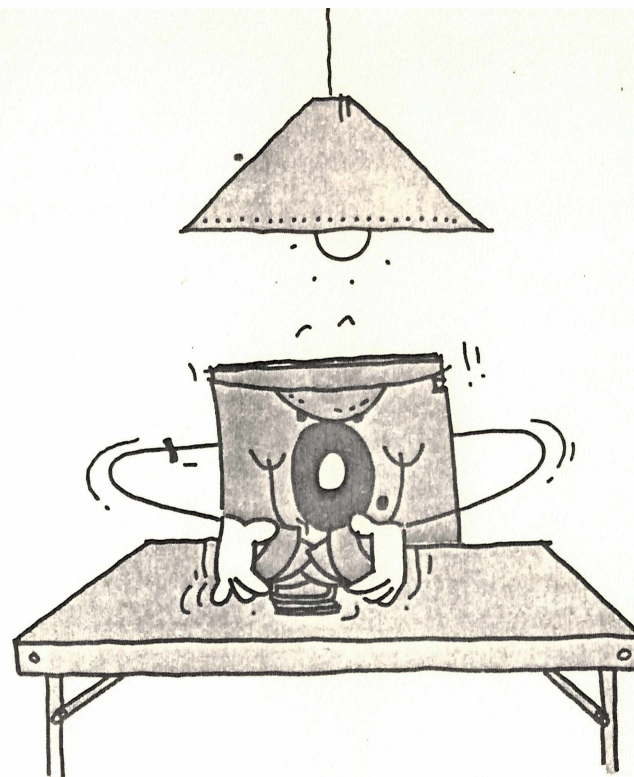
VERIFICAÇÃO DE APRENDIZADO

1. Caso você não indique o formato de resposta, quando o texto estiver em branco?

2. Por que é necessário usar o comando `trim` junto com o comando `trim`?

3. Quando o texto estiver em branco, qual comando deve ser usado?

CAPÍTULO 10



JUNTANDO ARQUIVO EM DISCO

(COMO JUNTAR PROGRAMAS, COMO USAR VÁRIOS ^BUFFERS^)



THE CHINESE DESK AND CHAIR

(A STUDY OF THE CHINESE DESK AND CHAIR)

JUNTANDO ARQUIVO EM DISCO

Você vai querer aproveitar ao máximo seus arquivos em disco porque neles é fácil gravar e recuperar informações. Neste capítulo falaremos de algumas maneiras especiais de usá-los.

COMO JUNTAR ARQUIVO EM DISCO

Com este primeiro método, você pode construir um programa a partir de outros já gravados no disco. Pode juntar qualquer programa gravado no disco com um que já se encontra na memória.

Digite e salve os dois programas abaixo:

```
10 REN CONVERSÃO DE IDADE EM MESES
20 N = N * 12
30 A$ = STR$(N) + " MESES"
SAVE "MESES", A (ENTER)
```

```
10 REN CONVERSÃO DE IDADE EM SEMANAS
20 N = N * 52
30 A$ = STR$(N) + " SEMANAS"
SAVE "SEMANAS", A (ENTER)
```

Certifique-se de ter digitado a letra A quando você salvou os programas. Nós lhe explicaremos o porquê mais tarde... Apague a memória e digite o seguinte programa:

```
5 INPUT "DIGITE A SUA IDADE"; N
40 PRINT "VOCÊ VIVEU" A$
```

Combine-o agora, (usando o comando MERGE), com um dos programas que você acaba de salvar. Digite:

MERGE "MESES" (ENTER)

Liste o programa ... O computador juntou o programa "MESES" com o programa que estava na memória. Repare que os números das linhas são as mesmas dos programas individuais.

A seguir apresentamos o programa que se encontra na memória, no momento:

```
5 INPUT "DIGITE A SUA IDADE"; N
10 REN CONVERSÃO DE IDADE EM MESES
20 N = N * 12
30 A$ = STR$(N) + "MESES"
40 PRINT "VOCÊ VIVEU APROXIMADAMENTE" A$
```

Junte o programa "SEMANAS" com o programa acima, digitando MERGE "SEMANAS" (ENTER). Depois disso liste o novo programa.

Repare que as linhas 10, 20 e 30 do programa que se encontrava na memória, foram substituídas pelas linhas 10, 20 e 30 do programa "SEMANAS".

Os números das linhas indicam ao computador como juntar os dois programas. Sempre que houver um conflito de linhas (duas linhas com o mesmo número), prevalece a linha do programa que está no disco.

Vamos agora explicar, tecnicamente, o que ocorre. O computador normalmente escreve no disco, códigos ASCII para cada caracter de dado. Por exemplo: ele escreve a palavra DE com dois códigos - O código ASCII para "D" (68) e o código ASCII para "E" (69). (Os códigos ASCII estão relacionados no apêndice D).

No entanto, quando ele salva o programa, escreve as palavras BASIC de outra forma e para economizar espaço, compacta cada palavra BASIC em um código de um byte binário.

Não é possível juntar arquivos que contenham estes códigos binários. Esta é a razão pela qual você digitou a letra "A" na gravação dos dois programas acima. Esta letra "A" indica ao computador para escrever cada palavra BASIC em código ASCII, ao invés de código binário.

Listando o diretório é possível verificar se os dados dentro dos arquivos estão em códigos ASCII ou binários. Se houver uma letra "A" na quarta coluna, todos os códigos são ASCII. Uma letra "B" indica que algumas (ou todas) palavras estão em código binário.

NOTA: Tente digitar MERGE "MESES, R (ENTER). A letra "R" faz com que o computador execute o programa depois de executar o comando MERGE.

COMO USAR MAIS ÁREA PARA ^BUFFER^

Quando se liga o computador com o COLOR DOS, ele cria 2 áreas de buffer para comunicação com o disco. Você pode usá-los ou não, para leitura ou gravação de dados.

Até agora nos só usamos os buffers #1 e #2. Mas, como já foi dito antes, podemos usar até 15 áreas de buffer.

Para usar mais de dois buffers, é necessário antes de mais nada reservar espaço na memória para eles. Para fazê-lo, use a palavra FILES. FILES 3 por exemplo reserva 3 buffers.

Usando vários buffers é possível simplificar os programas. Vamos supor, por exemplo, que você possua uma escola de computadores. Para organizá-la cadastrou todos os estudantes num arquivo chamado "MICRO/ESC". Apague a memória e digite e execute o programa abaixo:

```
1Ø OPEN "O", #1, "MICRO/ESC"
2Ø FOR X = 1 TO 6
3Ø READ A$
4Ø PRINT #1, A$
5Ø NEXT X
6Ø CLOSE #1
7Ø DATA MARIA, CARLOS, PAULO
8Ø DATA CYNTHIA, ELISABETE, PEDRO
```


Agora pode fazer um outro programa para indicar se o estudante está na turma de BASIC ou de PASCAL. Apague a memória e digite o seguinte programa:

```
10 FILES 3
20 OPEN "O", #1, "BASIC/TUR"
30 OPEN "O", #2, "PASCAL/TUR"
40 OPEN "I", #3, "MICRO/ESC"
50 IF EOF(3) = -1 THEN 120
60 INPUT #3, ES$
70 PRINT: PRINT ES$
80 INPUT "(1) BASIC OU (2) PASCAL"; R
90 IF R > 2 OR R < 0 THEN 80
100 WRITE #R, ES$
110 GOTO 50
120 CLOSE #1: CLOSE #2: CLOSE #3
```

Execute-o. Após indicar o curso de cada aluno, é possível emitir a relação de alunos com o programa abaixo. Apague a memória e digite e execute-o:

```
10 CLS
20 PRINT "TURMA DE BASIC": PRINT
30 OPEN "I", #1, "BASIC/TUR"
40 IF EOF(1) = -1 THEN 80
50 INPUT #1, A$
60 PRINT A$
70 GOTO 40
80 CLOSE #1
```

NOTA: Substitua na linha 20 "TURMA DE BASIC" por "TURMA DE PASCAL" e na linha 30 "BASIC TUR" por "PASCAL/TUR" para obter a relação de alunos do curso de pascal.

No programa que colocava cada aluno na sua turma, usamos três buffers. Por esta razão, foi possível comunicar-se com três arquivos em disco ao mesmo tempo.

A linha 10 reserva área de memória para estes três buffers. Nas linhas 20, 30 e 40 abrimos estes três buffers. Depois disso, a linha 60 lê um aluno do arquivo "MICRO/ESC" para o buffer #3.

A linha 100 transfere o nome do aluno para o buffer #1 ("BASIC/TUR") ou para o buffer #2 ("PASCAL/TUR").

Quando todos os alunos do buffer #3 ("MICRO/ESC") tiverem sido lidos, a linha 50 manda o computador para a linha 120 que fecha os três buffers.

COMO AUMENTAR O TAMANHO DE UM BUFFER

Ainda existe um assunto a ser abordado relativo ao comando FILES. Apague a memória e digite e execute o programa a seguir:

```
10 CLEAR 400
20 FILES 1, 400
30 A$ = "NORMALMENTE NAO SERIA POSSIVEL ESCREVER TODAS ESTAS FRASES NUM DISCO DE .UMA SO
    VEZ."
40 B$ = "ISSO PORQUE SEM USAR FILES, VOCE SO TEM 256 BYTES DE ESPAÇO PARA OS BUFFERS."
50 C$ = "NESTE PROGRAMA RESERVAMOS 400 BITES DE ESPAÇO PARA ELES."
60 D$ = "DESTA FORMA VOCE PODE MANDAR TODAS ESTAS FRASES PARA O BUFFER AO MESMO TEMPO."
70 E$ = "E O BUFFER AS GRAVARA NO DISCO DE UMA SO VEZ."
80 OPEN "O", #1; "PALAVRA/DAT"
90 WRITE #1, A$, B$, C$, D$, E$
100 CLOSE #1
```

Se quiser ler do disco este parágrafo, acrescente as linhas abaixo ao programa anterior e execute-o novamente:

```
200 OPEN "I", #1, "PALAVRA/DAT"
210 INPUT #1, A$, B$, C$, D$, E$
220 CLS
230 PRINT A$; B$; C$; D$; E$
240 CLOSE #1
```

NOTA: Você pode dar qualquer tamanho a um buffer.

VERIFICAÇÃO DE APRENDIZADO

1. Como deve ser salvo (SAVE) um programa para podermos usar o comando MERGE?
2. Quando juntamos dois programas que possuem uma mesma linha , qual prevalece?
3. Quantos buffers são reservados pelo COLOR.DOS quando o computador é ligado?
4. Qual é o tamanho da área reservada para os buffers se nada for indicado?
5. O que significa FILES, 3000?

APÊNDICES

APÉNDICES

APÊNDICE A

EXERCÍCIO DE PROGRAMAÇÃO # 5.1

```
10 PRINT: PRINT "CHEQUES GASTOS COM O CARRO"
20 OPEN "I", #1, "CEHQUES"
30 IF EOF(1) = -1 THEN 100
40 INPUT #1, A$, B, C$
50 IF C$ = "CARRO" THEN 70
60 GOTO 90
70 PRINT: PRINT "CHEQUE PAGO A ORDEM DE : "; A$
80 PRINT "VALOR : CR$"; B
90 GOTO 30
100 CLOSE #1
```

EXERCÍCIO DE PROGRAMAÇÃO # 6.1

```
10 OPEN "I", #1, "ANIMAIS /DAT"
20 OPEN "O", #2, "NOVO/DAT"
30 IF EOF(1) = -1 THEN 70
40 INPUT #1, A$
50 WRITE #2, A$
60 GOTO 30
70 CLOSE #1: CLS
80 PRINT "DIGITE OS NOMES DOS ANIMAIS QUE"
85 PRINT "VOCE QUER ACRESCENTAR."
90 PRINT "PRESSIONE <ENTER> QUANDO VOCE"
95 PRINT "TERMINAR"
100 INPUT A$
110 IF A$ = "" THEN 140
120 WRITE #2, A$
130 GOTO 80
140 CLOSE #2
150 KILL "ANIMAIS/DAT"
160 RENAME "NOVO/DAT" TO "ANIMAIS/DAT"
```

EXERCÍCIO DE PROGRAMAÇÃO # 6.2

```

10 OPEN "I", #1, "ANIMAIS/DAT"
20 OPEN "O", #2, "NOVO/DAT"
30 IF EOF(1) = -1 THEN 100
40 INPUT #1, A$
50 PRINT: PRINT A$
60 INPUT "VOCE QUER ELIMINAR ESTE"; R$
70 IF R$ = "SIM" THEN 90
80 WRITE #2, A$
90 GOTO 30
100 CLOSE #1
110 CLOSE #2
120 KILL "ANIMAIS/DAT"
130 RENAME "NOVO/DAT" TO "ANIMAIS/DAT"

```

EXERCÍCIO DE PROGRAMAÇÃO # 6.3

```

10 CLS: PRINT "VOCE DESEJA - -"
20 PRINT "(1) CRIAR UM NOVO ARQUIVO"
30 PRINT "(2) VIZUALIZAR O ARQUIVO"
40 PRINT "(3) FIM DE PROCESSAMENTO"
50 INPUT Q 1
60 ON Q1 GOTO 80, 130, 420
70 GOTO 10
80 OPEN "O", #1, "AMIGOS/DAT"
90 GOSUB 430
100 IF N$ = "" THEN CLOSE #1: GOTO 10
110 WRITE #1, N$, E$, T$
120 GOTO 90
130 OPEN "I", #1, "AMIGOS/DAT"
140 OPEN "O", #2, "TEMPOR/DAT"
150 CLS: INPUT "VOCE QUER ALTERAR DADOS"; Q2$
160 IF EOF(1) = -1 THEN 320

```



```

170 INPUT #1, N$, E$, T$
180 PRINT: PRINT "NOME : " N$
190 PRINT "ENDereco : " E$
200 PRINT "TELEFONE : " T$
210 IF Q2$ = "NAO" THEN 300
220 PRINT: PRINT "VOCE DESEJA --"
230 PRINT "(1) ALTERAR O ENDEREÇO"
240 PRINT "(2) ELIMINAR O AMIGO"
250 PRINT "(3) LER O PROXIMO AMIGO"
260 INPUT N
270 ON N GOTO 290, 160, 300
280 GOTO 230
290 INPUT "FORNECA O NOVO ENDEREÇO"; E$
300 WRITE, #2, N$, E$, T$
310 GOTO 160
320 PRINT: PRINT "VOCE QUER ACRESCENTAR UM"
325 INPUT "NOVO AMIGO"; Q3$
330 IF Q3$ = "NAO" THEN 380
340 GOSUB 430
350 IF N$ = "" THEN 380
360 WRITE #2, N$, E$, T$
370 GOTO 340
380 CLOSE #1, #2
390 KILL "AMIGOS/DAT"
400 RENAME "TEMPOR/DAT" TO "AMIGOS/DAT"
410 GOTO 10
420 END
430 CLS
435 PRINT "PRESSIONE <ENTER> PARA TERMINAR"
440 PRINT
445 INPUT "NOME DO AMIGO: "; N$
450 IF N$ = "" THEN 480
460 INPUT "ENDEREÇO :"; E$
470 INPUT "NUMERO DO TELEFONE :"; T$
480 RETURN

```

EXERCÍCIO DE PROGRAMAÇÃO # 7.1

```

10 OPEN "D", #1, "NOMES/DAT"
20 WRITE #1, "ANA MARIA"
30 PUT #1, 2
31 WRITE #1, "MARTA DA SILVA"
32 PUT #1, 4
34 GET #1, 4
36 INPUT #1, A$
38 PRINT A$
40 CLOSE #1

```

EXERCÍCIO DE PROGRAMAÇÃO # 7.2

Esta alteração provoca um erro de código FD - arquivo de dado errado - na linha 36. O primeiro campo no registro 2 é "NAVIO", que é uma cadeia de caracteres (STRING). A linha 36 lê este campo para dentro de N que é uma variável numérica.

EXERCÍCIO DE PROGRAMAÇÃO # 7.3

```

10 OPEN "D", #1, "NOMES/DAT"
20 GOTO 70
30 FOR X= 1 TO 8
40 PRINT: PRINT "REGISTRO" X
50 GOSUB 180
60 NEXT X
70 INPUT "QUAL E O REGISTRO (1-8)"; X
80 IF X > 8 THEN 170
90 IF X < 1 THEN END
100 GET #1, X
110 INPUT #1, N$, E$, C$, UF$, CEP$
120 PRINT: PRINT "REGISTRO" X
130 PRINT N$, E$, C$, UF$, CEP$
140 INPUT "VOCE QUER ALTERAR ALGO"; R$
150 IF R$ = "SIM" THEN GOSUB 180

```



```

16Ø GOTO 7Ø
17Ø CLOSE #1: END
18Ø INPUT "NOME :"; N$
19Ø INPUT "ENDEREÇO :"; E$
20Ø INPUT "CIDADE :"; C$
21Ø INPUT "ESTADO :"; UF$
22Ø INPUT "CEP" :"; CEP$
23Ø WRITE # 2, N$, E$, C$, UF$, CEP$
24Ø PUT #1, X
25Ø RETURN

```

EXERCÍCIO DE PROGRAMAÇÃO # 9.1

```

1Ø OPEN "D", #1, "ENDCO/DAT", 47
2Ø FIELD #1, 15 AS N$, 15 AS E$, 1Ø AS C$, 2 AS U$, 5 AS P$
3Ø R = R + 1
4Ø CLS
5Ø INPUT "NOME      :"; N1$
6Ø INPUT "ENDEREÇO  :"; E1$
7Ø INPUT "CIDADE    :"; C1$
8Ø INPUT "UF        :"; U1$
9Ø INPUT "CEP       :"; P1$
10Ø REM
11Ø LSET N$ = N1$
12Ø LSET E$ = E1$
13Ø LSET C$ = C1$
14Ø LSET U$ = U1$
15Ø LIST P$ = P1$
16Ø REM
17Ø PUT #1, R
18Ø PRINT
19Ø INPUT "MAIS DADOS (S/N)"; R$
20Ø IF R$ = "S" THEN 3Ø
21Ø CLOSE #1

```

EXERCÍCIO DE PROGRAMAÇÃO # 9.2

```

10 OPEN "D", #1, "ENDCO/DAT", 47
20 FIELD #1, 15 AS N$, 15 AS E$, 10 AS C$. 2 AS U$, 5 AS P$
30 R = R + 1
40 CLS
50 GET #1, R
60 PRINT N$
70 PRINT E$
80 PRINT C$ ", " U$
90 PRINT P$
100 PRINT
110 IF LOF(1)=R THEN 150
120 PRINT "PRESSIONE <ENTER> PARA IMPRIMIR"
130 INPUT "O PROXIMO NOME"; R$
140 GOTO 30
150 CLOSE #1

```

EXERCÍCIO DE PROGRAMAÇÃO # 9.3

```

10 OPEN "D", #1, "PAIS", 15
20 FIELD #1, 10 AS PAIS $, 5 AS HABI$
30 R = R + 1
40 CLS
50 INPUT "PAIS          ":";P$
60 INPUT "N HABITANTES ":";H
70 LSET PAIS$ = P$
80 LSET HABI$ = NKN$(H)
90 PUT #1, R
100 PRINT
110 INPUT "MAIS DADOS(S/N)"; R$
120 IF R$ = "S" THEN 30
130 CLOSE #1

```


EXERCÍCIO DE PROGRAMAÇÃO # 9.4

```
1Ø OPEN "D", #1, "PAIS", 15
2Ø FIELD #1, 1Ø AS PAIS$, 5 AS HABI$
3Ø R = R + 1
4Ø GET #1, R
5Ø PRINT PAIS$, CVN(HABI$)
6Ø IF LOF(1) < > R THEN 3Ø
7Ø CLOSE #1
```

CRP 3

CAP 4

1. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

2. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

3. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

4. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

5. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

6. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

7. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

8. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

9. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

10. O sistema de controle de acesso ao computador é composto por um conjunto de programas que controlam o acesso dos usuários ao sistema. Este sistema é responsável por garantir a segurança e a integridade dos dados armazenados no computador.

CAPÍTULO 5

1. O buffer #1 é uma área de armazenamento temporário que fica na memória do computador. É usada na transferência de dados entre a memória e o disco.
2. Um arquivo em disco precisa ser aberto (OPEN) antes que qualquer informação seja transmitida entre a memória e o disco.
3. Um arquivo em disco precisa ser fechado (CLOSE) para que as últimas informações contidas no seu buffer sejam transferidas para o disco. Todos os arquivos de um disco devem ser fechados antes que ele seja retirado da unidade de disco.
4. Um arquivo aberto para entrada permite que sejam transmitidas informações do arquivo em disco para a memória do computador. Um arquivo aberto para saída permite que sejam transmitidas informações da memória para o arquivo.

CAPÍTULO 6

1. Quando você abre um "arquivo de acesso sequencial", você só pode abri-lo para entrada ("I" - INPUT) ou saída ("O" - OUTPUT). Não é possível abri-lo para as duas alternativas ao mesmo tempo. Você não pode gravar (OUTPUT) num arquivo aberto para entrada (INPUT) nem ler (INPUT) num arquivo aberto para gravação (OUTPUT).
2. Não. O arquivo precisa primeiro ser fechado (CLOSE) e então aberto para leitura.

CAPÍTULO 7

1. Os registros de um arquivo são divisões que possuem o mesmo tamanho. Neles podemos guardar dados. Como todos têm o mesmo tamanho, o computador pode usar um número para acessá-los diretamente.
2. Os campos são subdivisões de um registro.
3. Num arquivo de acesso sequencial, as únicas duas posições que o computador conhece são o começo e o fim do arquivo. Num arquivo de acesso direto, ele pode determinar onde está cada um dos registros (usando o tamanho do registro).

4. Como os registros têm sua posição conhecida, o computador pode apagá-los sem precisar passar pelos registros precedentes, como seria necessário num arquivo de acesso sequencial.

CAPÍTULO 8

1. O menor tamanho de um arquivo em disco é 2.304 bytes (um grânulo). O computador aloca espaço no disco em grânulos, por isso um arquivo não pode ter menos que um grânulo.
2. O computador grava (WRITE) primeiro o sinal do número (um sinal menos ser for negativo ou um espaço em branco ser for positivo). Depois disso ele escreve o número seguido de um espaço em branco.
3. Uma cadeia de caracteres (string) é gravada entre duas aspas.
4. O comando INPUT lê um item de cada vez enquanto que LINE INPUT lê tudo até encontrar um caracter ENTER.
5. A vírgula faz o computador saltar para a próxima coluna de impressão antes de imprimir um novo dado.
6. O ponto e vírgula faz o computador imprimir os dados um ao lado do outro.
7. Uma cadeia de caracteres (string) é impressa (PRINT) simplesmente como ele é. Não são adicionadas duas aspas no início ou no fim.

CAPÍTULO 9

1. O computador aloca 256 bytes para cada registro.
2. Porque os dados precisam ser formatados em campos (FIELD) de tamanho constante para se usar o comando LSET. O tamanho dos campos é fornecido no comando FIELD.
3. O comando NKN\$ converte um número num string de 5 bytes.

CAPÍTULO 10

1. É preciso colocar a letra "A" no fim do comando SAVE se pretendemos usar futuramente es te programa com o comando MERGE.
2. Prevalece a linha do programa que está no disco.
3. O COLOR DOS reserva dois buffers quando ligamos o computador.
4. O COLOR DOS reserva 256 bytes para buffers quando ligamos o computador.
5. FILES 3, 30000 manda o computador criar 3 buffers com uma área total de 3000 bytes.

PROGRAMA SIMPLES 1 - CONTROLE DE CHEQUES

Este programa cria um arquivo em disco contendo todos os movimentos bancários (depósitos e saques) que você realizou durante um ano. Os dados podem ser analisados por mês ou por ano, para usar a impressora, basta trocar as instruções PRINT por PRINT -2. (Para maiores esclarecimento, leia o capítulo 21 do manual COLOR BASIC.

```

10 REM *** PROGRAMA "CHCRIA/BAS" ***
20 REM
30 REM ESTE PROGRAMA ARQUIVA TODOS OS DEPOSITOS E SAQUES EFE
40 REM TUADOS NA SUA CONTA BANCARIA. VOCE PODE ARQUIVAR TAMB
50 REM EM. PARA CADA CHEQUE EMITIDO, O CODIGO DA DESPESA A Q
60 REM UE ELE SE REFERE.
70 REM O PROGRAMA USA UM ARQUIVO DE ACESSO DIRETO E CADA REG
80 REM ISTRO DE 40 BYTES ESTA DIVIDIDO NOS SEGUINTE CAMPOS:
100 REM      -08 BYTES PARA A DATA (DD/MM/AA)
110 REM      -04 BYTES PARA O NUMERO DO CHEQUE OU DEPOSITO
115 REM      -20 BYTES PARA O NOME DO DESTINATARIO DO CHEQUE
120 REM      -03 BYTES PARA O CODIGO DE DESPESA OU RECEITA
125 REM      -05 BYTES PARA O VALOR DO CHEQUE OU DEPOSITO
130 REM
140 CLEAR 1000
150 DIM CH$(50)
160 CLS
170 PRINT @ 102, "*** SELETOR ***"
180 PRINT @ 161, "1 .. ACRESCENTAR CHEQUES "
190 PRINT @ 193, "2 .. IMPRIMIR O HISTORICO "
200 PRINT @ 225, "3 .. FIM DE PROCESSAMENTO "
230 PRINT @ 289, "INDIQUE A OPCAO (1, 2 OU 3)"
240 REM
250 OPCAO$ = INKEY$
260 IF OPCAO$ = "" THEN 250
265 IF OPCAO$ < "1" OR OPCAO$ > "3" THEN 250
270 ON VAL(OPCAO$) GOSUB 310,700,1560
280 GOTO 160
290 REM
300 REM
310 REM ROTINA PARA LER A DATA
320 REM
330 OPEN "D",#1,"CHEQUES/DAT",40
340 FIELD#1,8 AS DATO$,4 AS NCH$, 20 AS DEST$,3 AS COD$,5 AS MONT$
350 REG = LOF(1)
360 REG = REG + 1
370 CLS
380 PRINT @64, "CHEQUE OU DEPOSITO (C/D)?"
390 OPCAO$ = INKEY$
400 IF OPCAO$ = "D" THEN 430
410 IF OPCAO$ = "C" THEN 490
420 GOTO 390
430 INPUT "DATA DEPOSITO (DD/MM/AA)";D$
440 INPUT "NUMERO DEPOSITO (NNNN) ";C$
450 P$=""
460 INPUT "CODIGO DE RECEITA (CCC) ";A$
470 INPUT "VALOR DEPOSITADO ";V
480 GOTO 550
485 REM
490 INPUT "DATA CHEQUE (DD/MM/AA) ";D$
500 INPUT "NUMERO DO CHEQUE (NNNN)";C$
510 INPUT "NOME DO DESTINATARIO ";P$
520 INPUT "CODIGO DE DESPESA (CCC)";A$

```

```

530 INPUT "VALOR SACADO ";V
540 V= -V
550 LSET DATO$=D$
560 LSET NCH$ =C$
570 LSET DEST$=P$
580 LSET COD$ =A$
590 LSET MONT$=MKN$(V)
600 PUT #1,REG
610 PRINT @ 352, "MAIS DADOS (S/N)?"
620 OPCAO$ = INKEY$
630 IF OPCAO$ = "N" THEN 660
640 IF OPCAO$ = "S" THEN 360
650 GOTO 620
660 CLOSE #1
670 RETURN
680 REM
690 REM
700 REM ESTA ROTINA PROCESSA OS DADOS E EMITE O RELATORIO
710 REM
720 OPEN "D",#1,"CHEQUES/DAT",40
730 FIELD#1,8 AS DATO$,4 AS NCH$,20 AS DEST$,3 AS COD$,5 AS MONT$
740 CLS
750 PRINT @ 160, "VOCE QUER UMA LISTAGEM MENSAL QUANUAL (M/A) ?";
760 A$ = INKEY$
770 IF A$<>"M" AND A$<>"A" THEN 760
780 IF A$="M" THEN PRINT@256,"QUAL E' O MES (MM) ";:INPUT MS$
790 BAL = 0
800 FOR REG=1 TO LOF(1)
810 GET #1,REG
820 BAL = BAL + CVN(MONT$)
830 IF A$="M" AND MID$(DATO$,4,2)<>MS$ THEN 1030
840 CLS
850 IF DEST$="" THEN 930
860 PRINT @ 63, " ";
865 PRINT "DATA DO CHEQUE ";:DATO$
870 PRINT "NUMERO DO CHEQUE ";:NCH$
880 PRINT "NOME DESTINATARIO ";:DEST$
890 PRINT "CODIGO DA DESPESA ";:COD$
900 PRINT "VALOR DO CHEQUE ";:
905 PRINT USING "#####.##";-CVN(MONT$)
910 PRINT "SALDO ATUAL ";:
915 PRINT USING "#####.##";BAL
920 GOTO 980
925 REM
930 PRINT @ 63, " ";
935 PRINT "DATA DO DEPOSITO ";:DATO$
940 PRINT "NUMERO DEPOSITO ";:NCH$
950 PRINT "CODIGO DEPOSITO ";:COD$
960 PRINT "VALOR DO DEPOSITO ";:
965 PRINT USING "#####.##";CVN(MONT$)
970 PRINT "SALDO ATUAL ";:
975 PRINT USING "#####.##";BAL
980 PRINT @ 392, "PRESSIONE <ENTER>"

```



```

984 PRINT @ 424, "PARA CONTINUAR OU"
988 PRINT @ 456, "<E> PARA ENCERRAR";
990 OPCAO$ = INKEY$
1000 IF OPCAO$ = CHR$(13) THEN 1030
1010 IF OPCAO$ = "E" THEN 1040
1020 GOTO 990
1030 NEXT REG
1040 CLOSE #1
1050 RETURN
1540 REM
1550 REM
1560 REM  ROTINA DE FIM DE PROGRAMA
1570 REM
1580 CLOSE : CLS : NEW : END

```

PROGRAMA SIMPLES 2 - ORDENADO OS CHEQUES

Este programa classifica o arquivo criado pelo programa CONTROLE DE CHEQUES (1) em ordem crescente de conta. Se você quiser saber quanto gastou com médicos, educação etc... este programa o ajudará muito.

```

10 REM *** PROGRAMA "CRIACAO/BAS" ***
20 REM
30 REM O OBJETIVO DESTES PROGRAMAS E' CRIAR UMA LISTA DE NOMES
40 REM E TELEFONES. OS NOMES SERAO ARMAZENADOS NO DISCO EM O
50 REM RDEM ALFABETICA. OS NOMES E TELEFONES SAO PRIMEIRO LI
60 REM DOS PARA A VARIÁVEL DADOS$(I). DEPOIS COLOCADOS EM OR
70 REM DEM ALFABETICA E POR ULTIMO GRAVADOS NUM ARQUIVO EM D
80 REM ISCO CHAMADO "NOMES/DAT". O ARQUIVO TEM 32 BYTES DE C
90 REM OMPRIMENTO. TODOS ESTES BYTES ESTAO ALOCADOS A VARIÁV
100 REM EL INFO$. VOCE PODE ACRESCENTAR NOVOS DADOS AO ARQUIV
105 REM VO A QUALQUER MOMENTO E ESTES DADOS SEMPRE ESTARAO E
110 REM M ORDEM ALFABETICA. ESTE PROGRAMA PODE SER USADO JUN
120 REM TO COM OS PROGRAMAS "PESQUISA/BAS" E "ATUALIZA/BAS".
130 REM
140 CLEAR 1050
150 DIM DADOS$(30)
160 OPEN "D",#1,"NOMES/DAT",32
170 FIELD #1,32 AS INFO$
180 REM
190 REM PRIMEIRO VERIFICAMOS SE JA' EXISTE ALGUM REGISTRO GR

200 REM AVADO NO ARQUIVO "NOMES/DAT".
210 REM
220 IF LOF(1) = 0 THEN I=1 : GO TO 310
230 FOR I=1 TO LOF(1)
240 GET #1,I
250 DADOS$(I) = INFO$
260 NEXT I
270 REM
280 REM OS NOVOS NOMES SAO LIDOS E ARMAZENADOS NA VARIÁVEL D
290 REM ADOS$.
300 REM
310 CLS
320 PRINT @ 64
330 INPUT "NOME ";N$
340 INPUT "INICIAIS ";I$
350 INPUT "DDD ";D$
360 INPUT "TELEFONE ";T$
370 REM
380 DADOS$(I)=LEFT$(N$+" "+I$+"
",22)+D$+T$
390 PRINT@288,"MAIS DADOS (S/N)?"
400 OPCAO$ = INKEY$

```



```

410 IF OPCAO$ = "S" THEN I=I+1 : GOTO 310
420 IF OPCAO$ = "N" THEN 470
430 GOTO 400
440 REM
450 REM  ORDENACAO DE DADOS$
460 REM
470 FOR J=1 TO I
480 FOR K=J TO I
490 IF DADOS$(J) < DADOS$(K) THEN 530
500 TEMP$ = DADOS$(J)
510 DADOS$(J) = DADOS$(K)

```

```

520 DADOS$(K) = TEMP$
530 NEXT K
540 NEXT J
550 REM
560 REM  GRAVACAO DA LISTA NO ARQUIVO "NOMES/DAT"
570 REM
580 FOR N=1 TO I
590 LSET INFO$ = DADOS$(N)
600 PUT #1,N
610 NEXT N
620 CLOSE #1
630 END

```

PROGRAMA SIMPLES 3 - RELAÇÃO DE AMIGOS

Use este programa para gravar num disco o nome e o telefone de seus amigos. Os nomes serão armazenados em ordem alfabética. Para gravar o endereço destes amigos basta acrescentar algumas linhas ao programa.

```

10 REM  *** PROGRAMA "PESQUISA/DAS" ***
20 REM
30 REM  PARA EXECUTAR ESTE PROGRAMA E' PRECISO CRIAR O ARQUIV
40 REM  O "NOMES/DAT" ANTES. VEJA O PROGRAMA "CRIACAO/DAS".
50 REM  ESTE PROGRAMA PESQUISA UM ARQUIVO EM DISCO CONTENDO N
60 REM  OMES E TELEFONES EM ORDEM ALFABETICA. O ARQUIVO E' DE
70 REM  ACESSO DIRETO, SE CHAMA "NOMES/DAT" E POSSUI UM COMPR
80 REM  IMENTO DE 32 BYTES. A SUA ESTRUTURA E' A SEGUINTE :
90 REM  20 BYTES PARA O NOME; 2 BYTES PARA AS INICIAIS; 3 BYT
95 REM  ES PARA O DDD; 7 BYTES PARA O TELEFONE.
100 REM
110 REM
120 OPEN "D", #1,"NOMES/DAT",32
130 FIELD #1,22 AS NOME$, 3 AS DDD$, 7 AS TEL$
140 CLS
150 PRINT @ 99,"FORNECA O NOME E AS INICIAIS"
160 PRINT @130," ":LINE INPUT NM$
170 REM
180 REM  INICIALIZACAO DE VARIAVEIS
190 REM
200 N1$=NM$
210 IF LEN(NM$) < 22 THEN 800
220 IF LEN(NM$) > 22 THEN 820
230 INIC = 1
240 MEIO = INT((LOF(1)+1)/2)
250 FIM = LOF(1)
260 CNT = 0
270 REM

```

```

280 REM  O PROGRAMA TESTA PRIMEIRO O ULTIMO REGISTRO PORQUE E
290 REM  LE NAO E' PESQUISADO NA ROTINA.
300 REM
310 GET #1,FIM
320 IF NOME$ = NM$ THEN 450
330 REM
340 REM  O PROGRAMA COMPARA NOME$ COM NM$ ATE' ENCONTRAR NM$
350 REM  OU DESCOBRIR QUE NM$ NAO CONSTA DO ARQUIVO.
360 REM
370 REM
380 GET #1,MEIO
390 IF CNT > (LOF(1)+1)/2 THEN 710
400 IF NOME$ < NM$ THEN 570
410 IF NOME$ > NM$ THEN 640
420 REM
430 REM  QUANDO NM$ FOR ACHADO, ELE E' IMPRESSO NO VIDEO
440 REM
450 CLS
460 PRINT @ 104, NOME$
470 PRINT @ 136, "(:DDD$;)" ";" " :TEL$
480 PRINT @ 232, "PRESSIONE <ENTER> "
490 PRINT @ 264, "PARA CONTINUAR OU "
495 PRINT @ 296, "<E> PARA ENCERRAR "
500 OPCAO$ = INKEY$
510 IF OPCAO$ = "E" THEN CLOSE : END
520 IF OPCAO$ = CHR$(13) THEN 140
530 GOTO 500
540 REM
550 REM  ROTINA PARA NOME$ < NM$

```



```

560 REM
570 INIC = MEIO
580 MEIO = (MEIO+FIM)/2
590 CNT = CNT + 1
600 GOTO 380
610 REM
620 REM ROTINA PARA NOME$ > NM$
630 REM
640 FIM = MEIO
650 MEIO = (MEIO+INIC)/2
660 CNT = CNT + 1
670 GOTO 380
680 REM
690 REM ROTINA PARA QUANDO NM$ NAO EXISTIR
700 REM
710 CLS

```

```

720 PRINT @ 104, N1$
725 PRINT @ 136, "NAO FOI ENCONTRADO"
730 PRINT @ 232, "PRESSIONE <ENTER> "
735 PRINT @ 264, "PARA CONTINUAR OU "
740 PRINT @ 296, "<E> PARA ENCERRAR "
745 OPCAO$ = INKEY$
750 IF OPCAO$ = "E" THEN CLOSE : END
755 IF OPCAO$ = CHR$(13) THEN 140
760 GOTO 745
770 REM
780 REM ROTINA PARA TRANSFORMAR NM$ NUM STRING DE 22 BYTES
790 REM
800 NM$ = NM$ + " "
810 GOTO 210
820 NM$ = LEFT$(NM$,22)
830 GOTO 220

```

PROGRAMA SIMPLES 4 - PESQUISA DE UM NOME

Como o arquivo criado pelo programa "CRIACAO/BAS" (3) está em ordem alfabética é fácil pesquisar um nome dentro dele. Este programa mostra como fazê-lo.

```

10 REM *** PROGRAMA "ATUALIZA/BAS" ***
20 REM
30 REM PARA EXECUTAR ESTE PROGRAMA E' PRECISO CRIAR O ARQUIV.
40 REM O "NOMES/DAT" ANTES. VEJA O PROGRAMA "CRIACAO/BAS".
50 REM ESTE PROGRAMA PERMITE ALTERAR OU EXCLUIR UM REGISTRO
60 REM DO ARQUIVO "NOMES/DAT".
70 REM
80 REM
90 J=0 : CLS
100 PRINT @ 105, "*** SELETOR ***"
110 PRINT @ 164, "1 .. ALTERAR UM REGISTRO "
120 PRINT @ 196, "2 .. EXCLUIR UM REGISTRO "
122 PRINT @ 228, "3 .. FIM "
124 PRINT @ 297, "INDIQUE A OPCAO"
130 PRINT @ 331, "(1, 2 OU 3) "
140 OPCAO$ = INKEY$
150 IF OPCAO$ = "" THEN 140
155 IF OPCAO$ >= "1" AND OPCAO$ <= "3" THEN 160 ELSE 140
160 ON VAL(OPCAO$) GOSUB 180,590,850
170 CLOSE : GOTO 90
180 OPEN "D",#1,"NOMES/DAT",32
190 FIELD #1,22 AS NOME$, 3 AS DDD$, 7 AS TEL$
200 FOR I=1 TO LOF(1)
210 GET #1,I
220 CLS
230 PRINT @ 128, "NUMERO :";I
240 PRINT @ 160, "NOME :";NOME$

```

```

250 PRINT @ 192, "DDD :";DDD$
260 PRINT @ 224, "TELEFONE: ";TEL$
265 PRINT @ 296, "VOCE DESEJA ALTERAR"
270 PRINT @ 328, "ESTE REGISTRO (S/N)?"
280 OPCAO$ = INKEY$
290 IF OPCAO$ = "S" THEN 320
300 IF OPCAO$ = "N" THEN 560
310 GOTO 280
320 PRINT @ 392, "ALTERAR NOME (S/N)?"
330 OPCAO$ = INKEY$
340 IF OPCAO$ = "N" THEN NM$ = NOME$ : GOTO 400
350 IF OPCAO$ = "S" THEN 370
360 GOTO 330
370 PRINT @ 168, " :";:LINE INPUT NM$
380 IF LEN(NM$)<22 THEN NM$=NM$+" ":GOTO 380 ELSE 390
390 IF LEN(NM$)>22 THEN NM$=LEFT$(NM$,22)
400 PRINT @ 424, "ALTERAR DDD (S/N)?"
410 OPCAO$ = INKEY$
420 IF OPCAO$ = "N" THEN D$ = DDD$ : GOTO 460
430 IF OPCAO$ = "S" THEN 450
440 GOTO 410
450 PRINT @ 200,":":LINE INPUT D$
460 PRINT @ 456, "ALTERAR TELEFONE (S/N)?"
470 OPCAO$ = INKEY$
480 IF OPCAO$ = "N" THEN T$ = TEL$ : GOTO 520
490 IF OPCAO$ = "S" THEN 510
500 GOTO 470
510 PRINT @ 232,":":LINE INPUT T$

```



```

520 LSET NOME$=NM$
530 LSET DDD$ =D$
540 LSET TEL$ =T$
550 PUT #1,I
560 NEXT I
570 CLOSE #1
580 RETURN
590 OPEN "D",#1,"NOMES/DAT",32
600 FIELD#1,22 AS NOME$, 3 AS DDD$, 7 AS TEL$
610 OPEN "D",#2,"TEMP/DAT",32
620 FIELD#2,22 AS INOME$,3 AS TDDD$,7 AS TTEL$
630 FOR I=1 TO LOF(1)
640 GET #1,I
650 CLS
660 PRINT @ 128, "NUMERO  :";I
670 PRINT @ 160, "NOME    :";NOME$
680 PRINT @ 192, "DDD     :";DDD$

```

```

690 PRINT @ 224, "TELEFONE:";TEL$
700 PRINT @ 296, "VOCE DESEJA EXCLUIR"
705 PRINT @ 328, "ESTE REGISTRO (S/N)?"
710 OPCAO$ = INKEY$
720 IF OPCAO$ = "S" THEN 800
730 IF OPCAO$ = "N" THEN 750
740 GOTO 710
750 LSET INOME$ = NOME$
760 LSET TDDD$ = DDD$
770 LSET TTEL$ = TEL$
780 J=J+1
790 PUT #2,J
800 NEXT I
810 CLOSE
820 KILL "NOMES/DAT:0"
830 RENAME "TEMP/DAT" TO "NOMES/DAT"
840 RETURN
850 CLS:NEW:END

```

PROGRAMA SIMPLES 5 - ATUALIZAÇÃO DA LISTA

Com este programa você pode atualizar qualquer informação gravada pelo programa "CRIAÇÃO/BAS" (3) no arquivo "NOMES/DAT".

```

10 REM *** PROGRAMA "PROVAS/BAS" ***
20 REM
30 REM O OBJETIVO DESTES PROGRAMA E' CRIAR VARIOS ARQUIVOS: U
40 REM M ARQUIVO DE NOMES E VARIOS DE NOTAS DE PROVAS.
50 REM OS ARQUIVOS PODEM ENTAO SER PESQUISADOS. E AS NOTAS P
60 REM ROCESSADAS PARA DETERMINAR MEDIAS E DESVIOS PADROES.
70 REM TODOS OS ARQUIVOS SAO DE ACESSO SEQUENCIAL.
80 REM
90 REM
100 REM MODULO PRINCIPAL DO PROGRAMA
110 REM
120 DIM NOME$(30),GRAU(6,30)
130 CLS
140 PRINT @ 105, "*** SELETOR ***"
150 PRINT @161, "1 .. CRIAR UM ARQUIVO DE NOMES"
160 PRINT @193, "2 .. CRIAR NOVO ARQ. DE PROVAS"
170 PRINT @225, "3 .. PROCESSAR AS PROVAS "
180 PRINT @257, "4 .. FIM DE PROCESSAMENTO "
190 PRINT @321, "INDIQUE A OPCAO (1,2,3 OU 4) "

```

```

200 REM
210 REM
220 OPCAO$ = INKEY$
230 IF OPCAO$ = "" THEN 220
235 IF OPCAO$ < "1" OR OPCAO$ > "4" THEN 220
240 ON VAL(OPCAO$) GOSUB 290,430,640,1430
250 GOTO 130
260 REM
270 REM ROTINA PARA CRIAR UM ARQUIVO DE NOMES
280 REM
290 OPEN "O",#1,"NOMES/ARQ".
300 CLS
310 PRINT @ 96, "FORNECA O NOME DO ALUNO"
320 PRINT @127, " ":LINE INPUT NOME$
330 WRITE #1,NOME$
340 PRINT @ 200, "PRESSIONE <ENTER> "
345 PRINT @ 232, "PARA CONTINUAR OU "
347 PRINT @ 264, "<E> PARA ENCERRAR "
350 OPCAO$ = INKEY$

```



```

360 IF OPCAO$ = CHR$(13) THEN 300
370 IF OPCAO$ = "E" THEN CLOSE #1 : RETURN
380 GOTO 350
390 REM
400 REM
410 REM  ROTINA PARA CRIAR UM ARQUIVO DE PROVA
420 REM
430 CLS
440 PRINT C64
450 INPUT "FORNECA O NUMERO DA PROVA";NP$
460 IF NP$ = "" THEN 450
470 NP$ = "PROVA" + NP$
480 OPEN "I",#1,"NOMES/ARQ"
490 OPEN "O",#2,NP$
500 IF EOF(1) THEN 560
510 INPUT #1,NOME$
520 PRINT "NOME :";NOME$
530 INPUT "NOTA :";GRAU
540 WRITE #2,GRAU
550 GOTO 500
560 CLOSE #1,#2
570 RETURN
580 REM
590 REM  ESTA ROTINA LE OS ARQUIVOS DE NOMES E DE PROVAS A SE
600 REM  REM PROCESSADOS. O CALCULO PODERA' SER FEITO POR TUR
610 REM  MA OU POR ALUNO. TODOS OS RESULTADOS SAO IMPRESSOS N
620 REM  O VIDEO.
630 REM
640 OPEN "I",#1,"NOMES/ARQ"
645 Y=0
650 IF EOF(1) = -1 THEN 690
660 Y = Y + 1
670 INPUT #1, NOME$(Y)
680 GOTO 650
690 YFIM = Y
700 CLOSE #1
710 CLS
720 PRINT @ 96
730 INPUT "QUANTAS PROVAS EXISTEM";N
740 FOR X=1 TO N
750 NP$ = "PROVA" + RIGHT$(STR$(X),1)
760 OPEN "I",#1,NP$
770 FOR Y=1 TO YFIM
780 INPUT #1,GRAU(X,Y)
790 NEXT Y
800 CLOSE #1
810 NEXT X
820 CLS
830 PRINT@96,"FAZER A ANALISE POR TURMA OU PORALUNO (T/A)? ";
840 OPCAO$ = INKEY$
850 IF OPCAO$="A" THEN 900
860 IF OPCAO$="T" THEN 1180
865 GOTO 840
870 REM
880 REM  ROTINA PARA ANALISE POR ALUNO
890 REM
900 FOR Y=1 TO YFIM
910 CLS
920 PRINT @ 105, NOME$(Y)
930 PRINT @ 137, "NOTAS"
940 ALTOT = 0
950 FOR X=1 TO N

```

```

960 PRINT TAB(10) GRAU(X,Y)
970 ALTOT = ALTOT + GRAU(X,Y)
980 NEXT X
990 MED(Y) = ALTOT/N
1000 NUM = 0
1010 FOR X=1 TO N
1020 NUM = (MED(Y)-GRAU(X,Y))^2 + NUM
1030 NEXT X
1040 DP = SQR(NUM/N)
1045 PRINT
1050 PRINT USING "%      %###.###";"MEDIA:";MED(Y)
1060 PRINT USING "%      %###.###";"DESVIO PADRAO:";DP
1065 PRINT
1070 PRINT "PRESSIONE <ENTER> PARA CONTINUAR"
1080 OPCAO$ = INKEY$
1090 IF OPCAO$ = CHR$(13) THEN 1100 ELSE 1080
1100 NEXT Y
1110 CLS
1120 PRINT @ 105, "NAO EXISTEM MAIS NOMES"
1130 GOTO 1350
1140 REM
1150 REM  ESTA ROTINA PROCESSA AS NOTAS POR PROVA E TURMA
1160 REM
1170 REM
1180 CLS
1190 PRINT @ 32, "FORNECA O NUMERO DA PROVA";
1195 INPUT X
1200 REM
1210 PRINT @ 96, " NOMES"; TAB(25) "NOTAS"
1220 TTOT = 0
1230 FOR Y=1 TO YFIM
1240 TTOT = TTOT + GRAU(X,Y)
1250 PRINT TAB(1) NOME$(Y);TAB(25) GRAU(X,Y)
1260 NEXT Y
1270 MED = TTOT / YFIM
1280 NUM = 0
1290 FOR Y=1 TO YFIM
1300 NUM = NUM + (MED-GRAU(X,Y))^2
1310 NEXT Y
1320 DP = SQR(NUM/(YFIM-1))
1325 PRINT
1330 PRINT USING "%      %###.###";"MEDIA DA PROVA #";X;": ";MED
1340 PRINT USING "%      %###.###";"DESVIO PADRAO      : ";DP
1345 REM
1350 PRINT
1355 PRINT TAB(9) "PRESSIONE <ENTER> "
1356 PRINT TAB(9) "PARA CONTINUAR OU "
1360 PRINT TAB(9) "<E> PARA ENCERRAR ";
1370 OPCAO$ = INKEY$
1380 IF OPCAO$ = CHR$(13) THEN 820
1390 IF OPCAO$ = "E" THEN 1400 ELSE 1370
1400 RETURN
1410 REM
1420 REM  ROTINA DE FIM DE PROGRAMA
1430 REM
1440 CLOSE : CLS : NEW : END

```


PROGRAMA SIMPLES 6 - NOTAS DE PROVAS

Este programa é excelente para professores. Ele cria vários arquivos com nomes e notas de alunos. Baseado nestes dados, ele determina a média e o desvio padrão para toda a turma ou para cada aluno individualmente.

```
10 REM *** PROGRAMA "SELETOR/BAS" ***
20 REM
30 REM O OBJETIVO DESTES PROGRAMAS E' MOSTRAR COMO UM PROGRAMA
40 REM PODE CHAMAR OUTRO PROGRAMA. O PROGRAMA PRINCIPAL "SEL
50 REM ETOR/BAS" CHAMA, DEPENDENDO DE UM PARAMETRO, OS PROGR
60 REM AMAS "CASA/BAS", "SALA/BAS" E "QUARTO/BAS".
70 REM EVIDENTEMENTE, E' PRECISO GRAVAR ESTES PROGRAMAS NO D
80 REM ISCO ANTES DE EXECUTAR O PROGRAMA "SELETOR/BAS".
90 REM
100 REM
110 CLS
120 PRINT @ 105, "*** SELETOR ***"
130 PRINT @ 171, "1 .. CASA"
140 PRINT @ 203, "2 .. SALA"
150 PRINT @ 235, "3 .. QUARTO"
```

```
160 PRINT @ 267, "4 .. FIM"
170 PRINT @ 329, "INDIQUE A OPCAO"
175 PRINT @ 362, "(1,2,3 OU 4)"
180 OPCAO$ = INKEY$
190 IF OPCAO$ = "" THEN 180
195 IF OPCAO$ = "4" THEN 290
200 IF OPCAO$ >= "1" AND OPCAO$ <= "3" THEN 210 ELSE 180
210 CLS
220 PRINT @ 164, "PARA RETORNAR AO SELETOR"
230 PRINT @ 228, "PRESSIONE QUALQUER TECLA"
240 FOR I=1 TO 500 : NEXT I
250 ON VAL(OPCAO$) GOTO 260,270,280
260 LOAD "CASA/BAS",R
270 LOAD "SALA/BAS",R
280 LOAD "QUARTO/BAS",R
290 CLS : NEW : END
```

PROGRAMA SIMPLES 7 - LAR

A seguir apresentamos 4 programas que msotram, no vídeo, 3 vistas diferentes de uma casa. A casa, a sala e o quarto. Todas as vistas estão gravadas no disco.

```
10 REM *** PROGRAMA "CASABAS" ***
20 REM
30 PMODE 3,1
40 PCLS
50 SCREEN 1,0
60 DRAW "BM66,108:D48;R32;U48:L32"
70 DRAW "BM66,68;R132;DM46,96;R132;BM50,156;R128"
80 DRAW "BM50,96;D60;BM178,96;D60;BM206,88;D50"
90 DRAW "BM0,136;R50;BM206,136;R50"
100 LINE (46,96)-(66,68),PSET
110 LINE (178,96)-(198,68),PSET
120 LINE (198,68)-(206,88),PSET
```

```
130 LINE (174,156)-(206,136),PSET
140 CIRCLE (92,130),5,0
150 PAINT (0,0),3,4
160 PAINT (0,149),1,4
170 PAINT (67,70),4,4
180 PAINT (55,105),2,4
190 PAINT (194,96),2,4
200 PAINT (82,128),3,4
210 OPCAO$ = INKEY$
220 IF OPCAO$ = "" THEN 210
230 LOAD "SELETOR/BAS",R
```


10 REM *** PROGRAMA "SALA/BAS" ***

20 REM

30 PCLS

40 PMODE 3,1

50 SCREEN 1,0

60 DRAW "BM60,20;R140;D120;L40;U32;L4;D52;R4;U20;BM160,160;L128;U150"

70 DRAW "EM4,62;D130;BM28,166;U102;BM144,148;R12"

80 DRAW "EM40,72;D24;R36;U24;L36;BM44,76;D16;R28;U16;L28"

90 LINE (32,12)-(92,12),PSET

100 LINE (92,12)-(100,20),PSET

110 LINE (0,0)-(60,20),PSET

120 LINE (200,20)-(255,0),PSET

130 LINE (200,140)-(255,192),PSET

140 LINE (0,192)-(32,160),PSET

150 LINE (4,62)-(28,64),PSET

160 PAINT (120,4),2,4

170 PAINT (20,20),2,4

180 PAINT (230,20),2,4

190 PAINT (120,40),2,4

200 PAINT (60,16),3,4

210 PAINT (20,64),3,4

220 PAINT (158,124),4,4

230 PAINT (42,74),4,4

240 LINE (28,8)-(144,148),PSET

250 LINE (64,12)-(156,122),PSET

260 LINE (68,12)-(156,116),PSET

270 DRAW "BM144,148;U38"

280 FOR I=0 TO 9

290 DRAW "BM-8,28;U38"

300 NEXT I

310 DRAW "BM56,40;U28;BM48,31;U18;BM40,22;U10"

320 PAINT (56,84),2,4

330 CIRCLE (56,86),10,3,.4,0,.5

340 LINE (51,86)-(63,86),PSET

350 DRAW "BM56,84;L4;E7;D8"

360 FOR I=1 TO 32

370 CIRCLE (120,176),I*2,I/4,.25

380 NEXT I

382 DRAW "BM232,176;U100;R2;D100"

383 CIRCLE (232,180),15,4,1,.5,0

384 CIRCLE (232,178),6,4,2,.55,.1

385 CIRCLE (232,80),15,4,1,0,.5

386 CIRCLE (232,82),6,4,2,.1,.55

390 OFCA0\$ = INKEY\$

400 IF OFCA0\$ = "" THEN 390

410 LOAD "SELETOR/BAS",R


```
10 REM *** PROGRAMA "QUARTO/BAS" ***
```

```
20 REM
```

```
30 FCLS
```

```
40 FMODE 3,1
```

```
50 SCREEN 1,0
```

```
60 DRAW "BM104,60:D92:R48:U92:L48"
```

```
70 DRAW "BM44,20:R168:D132:L132:BL4:L12:BL4:L16:BM44,102:UB2"
```

```
80 DRAW "BM220,60:D100:BM244,58:D126"
```

```
90 DRAW "BM42,102:D38:R8:U38:L8"
```

```
100 DRAW "BM16,148:D40:R4:U40"
```

```
110 DRAW "BM64,148:D40:L4:U40"
```

```
120 DRAW "BM90,124:D40:L4:U36"
```

```
130 CIRCLE (144,108),4
```

```
140 CIRCLE (238,117),4
```

```
150 CIRCLE (45,140),15,4,.3,0,.7
```

```
160 CIRCLE (45,140),15,4,.3,.95,1
```

```
170 CIRCLE (53,136),4
```

```
180 LINE (0,192)-(16,176),PSET
```

```
190 LINE (20,172)-(44,152),PSET
```

```
200 LINE (256,192)-(212,152),PSET
```

```
210 PAINT (28,8),3,4
```

```
220 LINE (0,0)-(44,20),PSET
```

```
230 LINE (256,0)-(212,20),PSET
```

```
240 LINE (220,60)-(244,59),PSET
```

```
250 LINE (16,148)-(44,124),PSET
```

```
260 LINE (16,148)-(64,148),PSET
```

```
270 LINE (64,148)-(80,124),PSET
```

```
280 LINE (80,124)-(52,124),PSET
```

```
290 PAINT (10,10),3,4
```

```
300 PAINT (60,32),3,4
```

```
310 PAINT (240,20),3,4
```

```
320 PAINT (128,64),2,4
```

```
330 PAINT (228,70),2,4
```

```
340 PAINT (62,156),4,4
```

```
350 PAINT (78,150),4,4
```

```
360 PAINT (18,156),4,4
```

```
370 PAINT (68,128),1,4
```

```
380 PAINT (128,156),2,4
```

```
390 PAINT (40,140),4,4
```

```
400 PAINT (48,120),2,4
```

```
410 CIRCLE (46,98),5,2,2
```

```
420 OPCAD$ = INKEY$
```

```
430 IF OPCAD$ = "" THEN 420
```

```
440 LOAD "SELETOR/BAS",R
```


RELAÇÃO DE CÓDIGOS ASCII

A seguir, apresentamos a relação de códigos ASCII dos caracteres do teclado. A primeira coluna indica o caracter; a segunda representa o código em notação decimal (base 10); e a terceira é o código em notação hexadecimal (base 16).

CARACTER	CÓDIGO DECIMAL	CÓDIGO HEXADECIMAL
SPACEBAR	32	20
!	33	21
"	34	22
#	35	23
\$	36	24
%	37	25
&	38	26
'	39	27
(	40	28
)	41	29
*	42	2A
+	43	2B
,	44	2C
-	45	2D
.	46	2E
/	47	2F
0	48	30
1	49	31
2	50	32
3	51	33
4	52	34
5	53	35
6	54	36
7	55	37
8	56	38
9	57	39
:	58	3A
;	59	3B
<	60	3C
=	61	3D
>	62	3E
?	63	3F
@	64	40

CARACTER	CÓDIGO DECIMAL	CÓDIGO HEXADECIMAL
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D
N	78	4E
O	79	4F
P	80	50
Q	81	51
R	82	52
S	83	53
T	84	54
U	85	55
V	86	56
W	87	57
X	88	58
Y	89	59
Z	90	5A
␣ *	94	5E
␣ *	10	0A
␣ *	8	08
␣ *	9	09
BREAK	03	03
CLEAR	12	0C
ENTER	13	0D

RELAÇÃO DE CÓDIGOS DE TECLAS ESPECIAIS

CARACTER	CÓDIGO DECIMAL	CÓDIGO HEXADECIMAL	CARACTER	CÓDIGO DECIMAL	CÓDIGO HEXADECIMAL
	94	5E	(SHIFT) 	95	5F
	10	0A	(SHIFT) 	91	5B
	8	08	(SHIFT) 	21	15
	9	09	(SHIFT) 	93	5D
(BREAK)	03	03			
(CLEAR)	12	0C	(SHIFT)(CLEAR)	92	5C
(ENTER)	13	0D			

RELAÇÃO DE CÓDIGOS DE LETRAS MINÚSCULAS

A seguir apresentamos a relação de códigos ASCII das letras minúsculas. Para usar as letras minúsculas é preciso primeiro pressionar as teclas (SHIFT) e (Ø) simultaneamente. As letras minúsculas aparecerão no vídeo em cores reversas (teclas verdes e fundo preto).

CARACTER	CÓDIGO DECIMAL	CÓDIGO HEXADECIMAL	CARACTER	CÓDIGO DECIMAL	CÓDIGO HEXADECIMAL
a	97	61	n	110	6E
b	98	62	o	111	6F
c	99	63	p	112	70
d	100	64	q	113	71
e	101	65	r	114	72
f	102	66	s	115	73
g	103	67	t	116	74
h	104	68	u	117	75
i	105	69	v	118	76
j	106	6A	w	119	77
k	107	6B	x	120	78
l	108	6C	y	121	79
m	109	6D	z	122	7A

MENSAGENS DE ERRO

- /Ø Divisão por zero. Foi pedido ao computador para dividir um número por zero. Esta operação é impossível e este erro também ocorre se você escrever o nome de um arquivo sem aspas.
- AE Este nome de arquivo já existe. Você tentou usar os comandos RENAME e COPY com um nome de arquivo que já existe (Already Exists).
- AO Tentativa de abrir um arquivo de dados que já se encontrava aberto (Already Open).
- BR Número de registro incorreto. Você usou um número de registro impossível na instrução GET ou PUT. O número usado é muito pequeno (menor que 1) ou muito grande (maior que o número máximo de registros que o disco comporta). Use outro número da instrução PUT ou GET, ou diminua o tamanho do registro na instrução OPEN. (BAD RECORD).
- BS O índice de um vetor está fora dos limites admissíveis. Use a instrução DIM para dimensionar os vetores. Por exemplo, se você escrever no programa A (12), sem especificar na instrução DIM, que A é um vetor de 12 posições ou mais, obterá este erro. (Bad Subscript).
- CN Não é possível continuar o programa. Se utilizar o comando CONT estando no fim (END) do programa, obterá este erro. (CAN NOT)
- DD Tentativa de redimensionar um vetor. Um vetor só pode ser dimensionado uma vez. Por exemplo, você não pode ter DIM A (12) e DIM A (5Ø) no mesmo programa. (Dimension)
- DF Disco cheio. Você tentou colocar um arquivo num disco cheio. Use um outro disco (Disk Full).
- DN Erro no número da unidade de disco ou no número do "buffer". Erro no número da unidade de disco. Você usou um número de unidade maior que 3. Este erro também ocorre se usar os comandos DSKINI ou BACKUP sem indicar o número da unidade. Se possuir uma única unidade de disco, especifique o número zero com estes dois comandos DSKINI Ø ou BACKUP Ø. (Drive Number)

Erro no número do "buffer". Você usou mais buffers que o computador reservou. Use o comando FILES para reservar mais buffers. Este erro também ocorre se usar um número de buffer inválido(Por exemplo buffer #-3) ou omitir o número do buffer na instrução FIELD(por exemplo FIELD 1 AS A\$ em vez de FIELD #1, 1 AS A\$). (Device Number).

DS Instrução direta. Existe uma instrução direta no arquivo. Este erro ocorre se carregar um programa cujas instruções não tenham número de linha. (Direct Statement)

ER Tentativa de ler ou gravar um registro com número de ordem superior ao número do último registro do arquivo. Este erro só ocorre com arquivo de acesso direto. Você está tentando colocar (PUT) mais dados no arquivo do que ele admite ou pegar (GET) mais dados do que ele contém (End of Record).

FC Chamada incorreta de função. Este erro ocorre quando você usa uma palavra BASIC com um parâmetro (número) fora dos limites admissíveis. Por exemplo SOUND (260, 260) e CLS(10) provocarão este erro. A instrução, RIGHT\$(S\$,20) também provoca este erro se a variável S\$ possuir menos que 20 caracteres. existem outras situações em que ocorre este erro, como por exemplo, quando usamos vetores com índices negativos (A(-1)) ou quando chamamos funçõesUSR sem termos fornecido (POKE) antes o seu endereço. (Function Call).

FD Formato de campo incorreto. Este erro ocorre quando você lê (INPUT) ou grava (PRINT) um dado usando um formato inadequado. Se tentar, por exemplo, ler uma cadeia de caracteres (string) com a instrução INPUT #1,A este erro ocorrerá (File Data).

FM Tipo de arquivo incorreto. O tipo de arquivo ("O", "I" ou "D") especificado na instrução OPEN não corresponde ao que se quer executar. Você está tentando, por exemplo, usar as instruções GET e PUT num arquivo tipo "I" ou "O" (use "D"). Este erro também ocorre se for usada a instrução WRITE num arquivo "I" (use "O") ou a instrução INPUT num arquivo "O" (use "I"). (File Mode).

FN Nome de arquivo incorreto. Você está usando um formato incorreto para dar nome a um arquivo. (File Name)

- FO Estouro de campo. O tamanho do campo é maior que o registro (Field Overflow).
- FS A estrutura do arquivo não está correta. O seu arquivo em disco não está correto. Este erro ocorre se você gravar dados de forma incorreta ou se o diretório do disco apresentar problemas. Leia a descrição do erro IO para saber como proceder. (File Structure).
- ID Instruções direta incorreta. Você só pode usar a instrução INPUT dentro de um programa. Não é possível usá-la como um comando (Illegal Direct statement).
- IE Tentativa de ler um dado além do fim do arquivo. Use as instruções EOF e LOF para descobrir se atingiu o fim de um arquivo. Quando você atingir o fim de arquivo feche-o (ClOse). (Input past End).
- IO Erro de entrada ou saída. O computador está encontrando dificuldades para ler ou gravar informações num arquivo em disco.
- (1) Verifique se o disco foi inserido corretamente na unidade e se a porta esta fechada.
 - (2) Se este erro persistir, deve haver alguma coisa errada com o seu disco. Retire e insira-o novamente. Tente usar um outro disco ou "reformatá-lo". (Lembre-se que ao reformatamos um disco, apagamos o seu conteúdo.
 - (3) Se você continua recebendo este erro, o computador ou a unidade de disco está com defeito. Procure um representante.
- Este erro também poderá ocorrer com outros periféricos como por exemplo o gravador cassete. (Input/Output)
- LS Cadeia de caracteres muita longa. Uma cadeia de caracteres pode ter no máximo 255 caracteres. (String too Long)
- NE O computador não encontrou o arquivo em disco que você deseja. verifique o diretório deste disco para ter certeza que ele está nele. Caso você possua várias unidades de disco, pode ter esquecido de acrescentar o número da unidade. Se está usando os comandos COPY, KILL ou RENAME, pode ter esquecido de fornecer a extensão do arquivo.

- NF Instrução NEXT sem FORM correspondente. Você usou uma instrução NEXT sem a instrução FOR correspondente. Este erro também ocorre se inverter a ordem das instruções NEXT num ninho de FOR/NEXT. (NEXT without FOR)
- NO O arquivo não foi aberto. Você não pode ler ou gravar dados num arquivo sem antes abri-lo (OPEN). (Not Open)
- OB O "buffer" não tem espaço para armazenar todos os dados transmitidos. Use a instrução FILES para reservar mais espaço. (Out of Buffer)
- OD Não há dados suficientes nas instruções DATA para atender a todas as instruções READ. Verifique as instruções DATA READ de seu programa (Out of Data).
- OM Estouro na memória. Você já usou toda a memória disponível do computador. Tente alterar a estrutura do programa dividindo-o em vários módulos independentes e transformando cada módulo num programa autônomo (Out of Memory).
- OS Estouro de cadeia de caracteres. Não há espaço suficiente na memória para executar as operações envolvendo cadeias de caracteres. Use a instrução CLEAR no começo do programa para reservar mais espaço para as cadeias de caracteres. (Out of String).
- OV Estouro variável numérica. Uma das operações aritméticas, no seu programa, gerou um número muito grande ou muito pequeno e o computador não tem condições de processar este número. (OVerflow).
- RG Instrução RETURN sem o GOSUB correspondente. Você usou uma instrução RETURN sem a instrução GOSUB correspondente. (Return without Gosub)
- SE Tentativa de usar a instrução LSET ou RSET com uma variável que não foi definida numa instrução FIELD. Verifique a instrução FIELD do seu programa. (SEt).
- SN Erro de sintaxe. Este erro ocorre quando você escreve incorretamente uma palavra BASIC ou usa incorretamente a pontuação, os parenteses ou as aspas. Digite novamente a linha incorreta. (SiNtaxe)

- ST Tentativa de usar cadeias de caracteres em fórmulas muito complexas. Uma operação com cadeias de caracteres está muita complexa para ser processada pelo computador. Desdo bre esta operação em outras de menor porte. (SString).
- TM Incompatibilidade de tipo de variável. Este erro ocorre quando você tenta atribuir um número a uma variável "string" (A\$=3) ou uma cadeia de caracteres a uma variável numé rica (A="MICRO"). Este erro também ocorre se você não colocar o nome de um arquivo en tre duas aspas. (Type Mismatch).
- UL Linha não definida. Existe no programa uma ilustração GOTO ou GOSUB ou qualquer outra instrução de desvio pedindo ao computador que vá para uma linha cujo número não exis te. (Undefined Line).
- VF Verificação. este erro só ocorre quando o comando VERIFY estiver ligado e você tentar gravar informações no disco. O computador está informando que ocorreu uma falha na gra vação das informações. Leia a descrição do erro IO para saber como proceder. (VeriFy).
- WP Protegido contra gravação. Você tentou gravar informações num disco que está protegi do contra gravações. Use um disco diferente ou retire a etiqueta gomada que impede A\$ gravação. Se o seu disco não está protegido contra gravações, ocorreu um erro de en trada ou saída. Leia a descrição do erro IO para saber como proceder. (Write Protected).

1. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

2. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

3. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

4. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

5. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

6. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

7. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

8. A primeira seção do relatório descreve o trabalho realizado durante o período de observação. A segunda seção descreve os resultados obtidos e a terceira seção descreve as conclusões.

