

MICROCOMPUTADOR

CURSO BÁSICO



EDIÇÃO REVISTA E ATUALIZADA



EDITORA
GLOBO

**MICRO
COMPUTADOR
CURSO BASICO**

Todos os direitos reservados. Nenhuma parte desta edição pode ser utilizada ou reproduzida — em qualquer meio ou forma, seja mecânico ou eletrônico, fotocópia, gravação, etc. — nem apropriada ou estocada em sistema de banco de dados, sem a expressa autorização da editora.

Tradutoras:

Maria Clara Cescato
Regina Maria M. Gonçalves

Consultor técnico:

Jakow Grajew

Adaptação dos programas BASIC:
SAD — Sistema de Apoio e Decisão

Fotos de equipamentos nacionais:
Salvador de Rosa

Composição:

Lídio Ferreira Jr. Artes Gráficas e Editora Ltda.

Impressão:

Cayfosa - Sta. Perpétua de la Mogoda, Barcelona
Depósito legal: B.525-1987 (I)

© Orbis Publishing Co., 1984

© Editora Globo, 1987, para a língua portuguesa.

MICRO COMPUTADOR CURSO BASICO



Introdução

de Jakow Grajew

Professor da Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas e Consultor em Informática

O surpreendente aumento da procura de microcomputadores constitui um dos maiores fenômenos mercadológicos dos últimos anos, em todo o mundo. Boa parte destes computadores está sendo usada profissionalmente em escritórios e fábricas. No entanto, uma parcela crescente de compradores está levando seu microcomputador para casa. Muitos destes usuários interessam-se principalmente pelo aspecto de lazer no uso de "videogames" ou pequenos programas de jogos. Por outro lado, boa parcela do público não se sente atraída por esse tipo de divertimento e desconhece as outras utilidades dos microcomputadores.

Normalmente, os vendedores apontam algumas funções em que o microcomputador pode ter um bom desempenho. Mas tarefas como a sugestão do cardápio para o jantar, o controle do saldo bancário ou o cálculo de probabilidades (para aumentar a chance de acerto numa loteria esportiva) certamente não justificam a posse de um microcomputador. "Será que não se trata de mais uma mania passageira, como tantas outras?", raciocina o comprador em potencial.

Tenho boas razões para responder negativamente a essa pergunta, explicando inclusive os motivos pelos quais os microcomputadores merecem ser vistos como algo muito mais importante do que fliperamas ou maquininhas que ajudam a passar o tempo. E posso acrescentar que o domínio de suas potencialidades, seu funcionamento interno e sua evolução é de extrema valia nos dias atuais.

Alguns usuários domésticos valorizam seus microcomputadores apenas em função do fascínio e do divertimento que oferecem. Aspectos referentes a eficácia, utilidade e vantagens parecem-lhes menos relevantes já que se trata de verdadeiros "hobistas".

Tais consumidores não precisam ser convencidos dos benefícios oferecidos pelo curso que estamos iniciando. Eles sabem que aprendendo mais sobre "hardware" e "software" permitirá melhorar o uso de seu microcomputador, conectar diferentes periféricos, aumentar a variedade de uso e até mesmo desenvolver seus próprios programas. Mas outros proprietários (ou futuros proprietários) desejam conhecer e usufruir o máximo de seus microcomputadores, não podendo ser considerados "hobistas". É também para estes usuários ou potenciais usuários que destinamos a presente obra.

O progresso da tecnologia eletrônica conduz a uma crescente expansão dos microcomputadores, introduzindo técnicas de computação em quase todos

os setores da vida. Nossa sociedade está se tornando dependente dos computadores para garantir sua prosperidade, sua saúde e sua defesa. A menos que haja uma catástrofe (natural ou provocada pelo homem), a tecnologia da informação não vai estagnar. Portanto, a familiaridade com os computadores, o conhecimento de seu mecanismo básico e de suas utilidades são fundamentais para o homem de hoje.

Esta é uma das razões que levam os pais a colocar seus filhos em contato direto com as máquinas do presente. Querem prepará-los para o futuro, e sua prosperidade poderá depender daquilo que venham a saber sobre os computadores.

Muitas vezes ouvi que o motorista não precisa entender o funcionamento de seu carro para dirigi-lo bem. A mesma linha de raciocínio leva a concluir que não é necessário entender como os computadores funcionam para usá-los adequadamente. No entanto, para o uso de computadores, o conhecimento da máquina é de extrema importância, principalmente porque revela as potencialidades do instrumento, eliminando assim os temores que a ignorância de seu funcionamento pode provocar.

A vantagem de possuir um computador e aprender sempre mais a seu respeito transcende o campo teórico. Depois de um período inicial, em que se adapta ao microcomputador, o usuário percebe que seu aparelho é a base de um sistema muito útil e eficiente.

Pequenas empresas estão se valendo de microcomputadores na contabilidade, no controle de estoques, na avaliação de desempenhos, nos registros. Muitos editores, escritores e professores passaram a usar a microcomputação em sistemas de processamento de textos. Tais aplicações exigem equipamentos acessórios, como unidade de discos e impressoras, além de programas produzidos profissionalmente. Portanto, esses itens também foram incluídos em MICROCOMPUTADOR — CURSO BÁSICO.

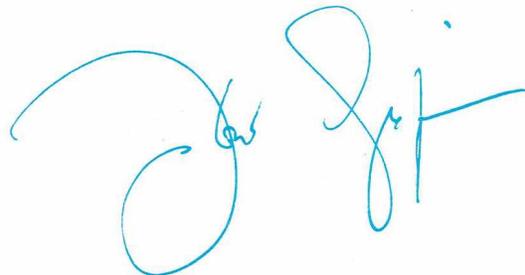
A maioria dos usuários de microcomputadores adotam a linguagem de programação BASIC, que pode ser aprendida com facilidade. Assim, está incluído nesta obra um curso completo de BASIC, em 24 partes. Mas, já que a verdadeira alfabetização em computadores exige também o conhecimento de outras linguagens de programação — como PASCAL ou LOGO, orientadas especialmente para o ensino de crianças —, no decorrer de MICROCOMPUTADOR — CURSO BÁSICO, você entrará em contato com mais algumas linguagens específicas, todas de muita utilidade.

O futuro certamente exigirá um maior uso de “geradores de programas” e de linguagem de alto nível, o que facilitará a transmissão das necessidades do usuário ao computador. As linguagens funcionais permitirão dizer-lhe o *que fazer*, em vez de apenas instruí-lo sobre *como fazer* uma tarefa. Dessa forma, os computadores tornar-se-ão, além de sistemas de informações, fontes de sugestões.

Embora fascinantes e extraordinários, os atuais microcomputadores tendem a ser superados pelos que ainda virão. O progresso nessa área é tão vertiginoso que não se pode prever com exatidão os rumos do desenvolvimento dos microcomputadores. Mas, sem dúvida, quem estiver familiarizado com o nível atual dos microcomputadores poderá compreender e manipular melhor sua inevitável evolução.

Muitos dos que se esforçam por aprender microcomputação estão motivados pelo desejo de se profissionalizar. MICROCOMPUTADOR — CURSO BÁSICO mostrará a diferença entre escrever programas para uso doméstico e projetar software de qualidade profissional para importantes aplicações em negócios. A distância é a mesma que existe entre construir um bonito planador de madeira e projetar um avião supersônico...

De todos os engenhos criados pelo homem, o computador digital é o mais versátil. Os programas que o controlam determinam sua capacidade e o uso é limitado apenas pela criatividade e engenhosidade dos programadores. O advento do microcomputador representa um passo a mais no campo da “engenharia da informação”: um campo fértil para o progresso individual e o desenvolvimento das nações. Esta obra nada mais almeja do que dar sua contribuição nesse rumo.

A handwritten signature in blue ink, consisting of a large, stylized initial 'D' followed by a name that appears to be 'D. J. F.'.

Para aproveitar o máximo de seu

MICROCOMPUTADOR — CURSO BÁSICO

Eis o primeiro guia completo ao mundo dos microcomputadores. Para facilitar ainda mais sua compreensão, o curso está dividido em seções, identificadas por diferentes faixas coloridas no alto de cada página.

Por dentro do hardware Talvez você esteja indeciso quanto à compra de seu primeiro microcomputador e julgue difícil avaliar os argumentos usados por fabricantes e vendedores. Pode ser também que você pretenda adquirir um aparelho mais potente e moderno, a fim de substituir seu modelo atual. Em qualquer desses casos, leia a seção *Por dentro do hardware*. Fizemos um estudo pormenorizado e colocamos ao seu alcance uma lista de especificações didaticamente elaboradas em forma de tabela, o que torna simples comparar as máquinas entre si e escolher os diversos periféricos disponíveis. Se você já tem um microcomputador, certamente desejará experimentá-lo — saber e testar tudo o que ele pode fazer.

Programação BASIC A seção *Programação BASIC* mostra inicialmente o que seu microcomputador *não* pode fazer — evitando assim que você escreva programas que o forcem a cometer erros, pois ele não teria condições de corrigi-los. A partir daí, são propostos exercícios que possibilitam, gradualmente, o domínio do assunto.

Fundamentos O que acontece *dentro* de um computador quando você escreve seus programas? A seção *Fundamentos* vai ajudá-lo a compreender as funções internas de seu microcomputador, pois explica a lógica eletrônica e como ela se aplica no projeto do computador.

Som e luz À medida que desenvolver habilidades mais sofisticadas, a seção *Som e luz* fará você percorrer o empolgante caminho que leva à geração de gráficos luminosos e efeitos sonoros (até música).

Conexões Existem também os periféricos: unidades de discos e impressoras, joysticks e modems. A seção *Conexões* permite que você veja o interior dessas máquinas, conhecendo detalhes de sua construção e seus fundamentos eletrônicos.

Software/hardware Essas duas seções apresentam os mais fascinantes programas (software) e equipamentos (hardware) atualmente em uso e esboçam os princípios a partir dos quais eles foram elaborados.

Perspectivas Sob este tema, você encontra as inúmeras aplicações práticas dos microcomputadores e percebe como eles estão cada vez mais presentes na vida cotidiana.

Chips & bytes Nesta seção, você fica sabendo como um chip é fabricado, o que é uma rede de processamento, e muitos outros “mistérios” do mundo da computação.

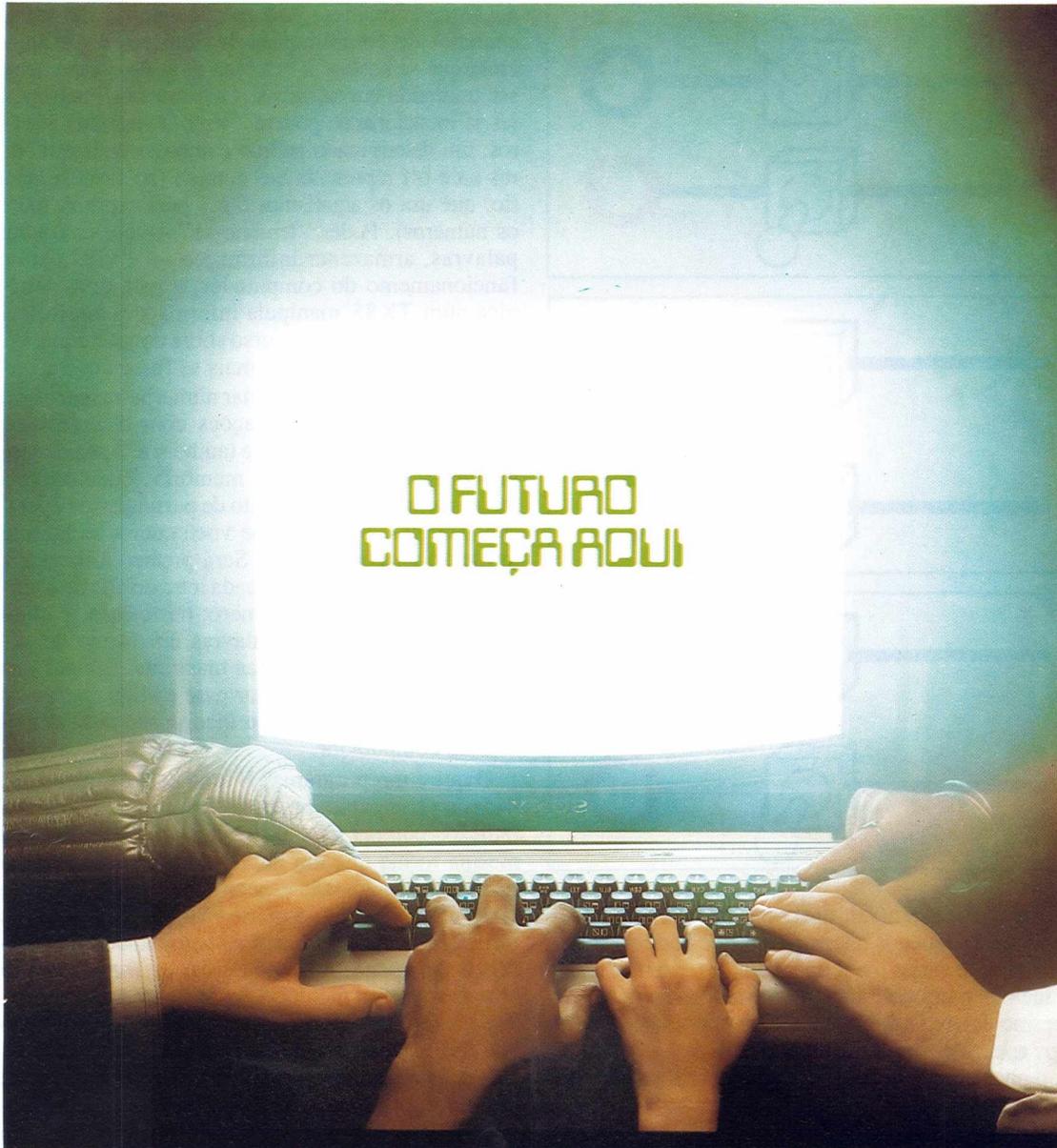
Precursores Para completar sua visão de como os computadores se desenvolveram, esta seção vai introduzi-lo na carreira e nas realizações de alguns dos pioneiros deste admirável mundo novo.

O futuro já começou. Seja bem-vindo ao MICROCOMPUTADOR — CURSO BÁSICO.



O que é computador?

O que “pensam” e o que “sabem” os computadores?
A resposta a essas questões é vital para compreendê-los.

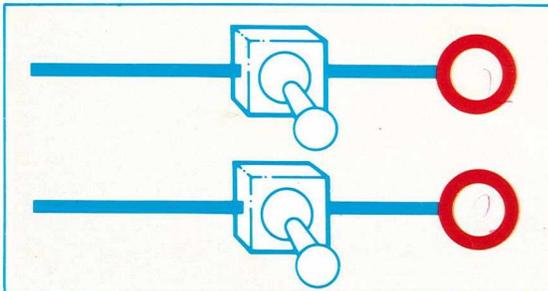
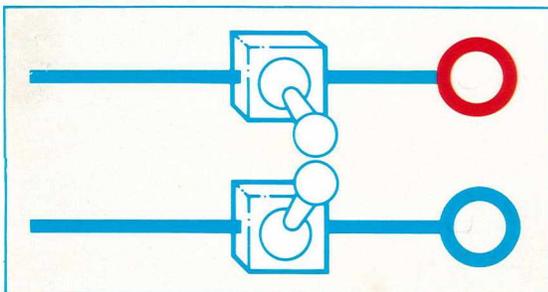
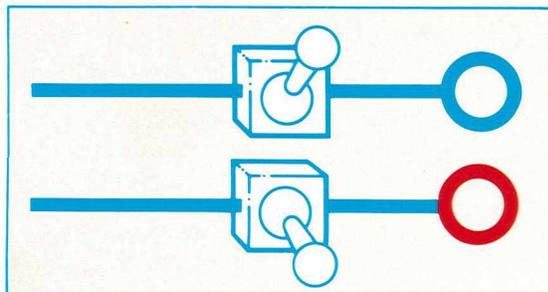
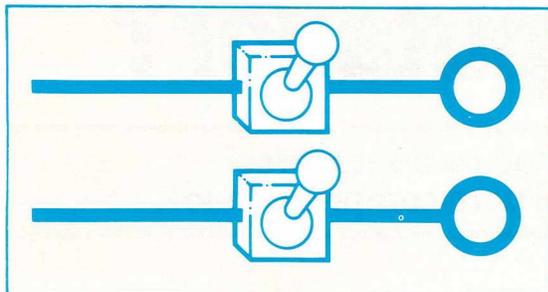


A questão “O que é um computador?” não é tão fácil de responder quanto “O que é um televisor?” ou “O que é uma máquina de lavar?”, porque o computador, ao contrário desses aparelhos, não se destina a um único uso. Computadores digitais, incluindo os que podem ser comprados por cerca de 100 dólares, são um novo tipo de máquina capaz de realizar uma grande variedade de tarefas, dependendo de como forem programados pelos usuários.

A idéia de “programabilidade” não é totalmente desconhecida nas casas modernas; afinal, aparelhos

domésticos como a máquina de lavar e os fogões já são programáveis para executar muitas funções. Com um computador, porém, todas as funções da máquina podem ser modificadas — basta introduzir um novo programa para que ele passe, em minutos, de um processador de palavras para um jogo eletrônico ou para um controlador de sua conta bancária.

Como um computador consegue realizar tantas tarefas? Você aprenderá mais sobre isso em outros capítulos, mas desde já vamos dar uma rápida olhada nos princípios envolvidos.



Transformando em números

Para representar números, o computador usa circuitos elétricos feitos basicamente de interruptores. Podem estar ligados (on) ou desligados (off). Dois interruptores, juntos, perfazem quatro combinações de on e off. Um sistema como o da ilustração é usado para representar números: off/off é zero; off/on é 1; on/off é 2; e on/on é 3.

O uso de grupos com mais de dois interruptores permite a representação de números maiores. Os computadores processam números e operações matemáticas complexas com extrema rapidez, graças a milhares de interruptores microscópicos.

Num primeiro nível, um computador é uma caixa repleta de pequenos interruptores que podem ser conectados de várias maneiras. Este, no entanto, não é o melhor meio de iniciar seu estudo, se você quer mesmo entender o que os computadores podem fazer; apenas as pessoas que projetam e constroem os computadores precisam conhecer esse estágio, os usuários (incluindo você) não.

Um computador é uma máquina extraordinariamente complexa: graças ao estonteante desenvolvimento da microeletrônica (o célebre chip de silício), é possível, mesmo para um pequeno computador doméstico, abrigar cerca de 250.000 desses pequenos interruptores. Cada um pode estar ligado (on) ou desligado (off). Ao mesmo tempo, o computador que você adquirir terá um programa embutido, permanente, que disfarça sua espantosa complexidade; permite até uma “conversa” com a máquina, usando-se algumas palavras em inglês, abreviadas mas facilmente reconhecíveis.

Muitas pessoas ficam surpresas quando ligam

pela primeira vez um computador e descobrem que a máquina, afinal, não sabe nada do que elas supunham que deveria saber. Embora pareça estranho, ainda não desapareceu o pressuposto de que o computador é um “cérebro eletrônico” capaz de conhecer todas as coisas. Mas precisa saber, por exemplo, o nome da capital do Afeganistão? Ou a altura do Kilimanjaro?

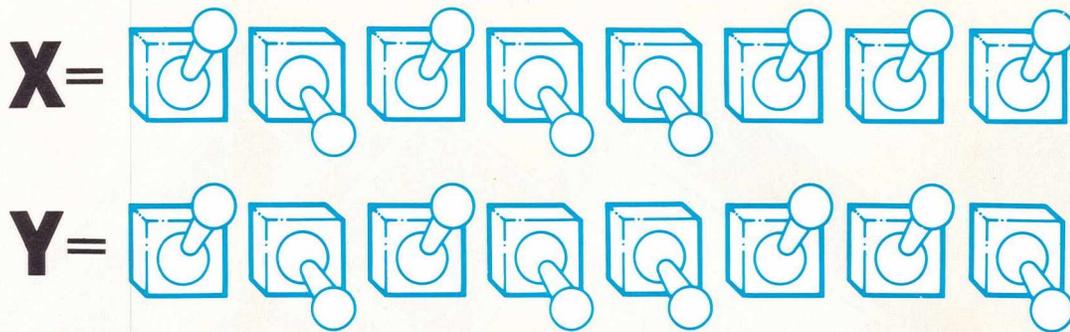
Na verdade, longe de dominar esses dados, o chip de silício que caracteriza o “cérebro” de um microcomputador não conhece sequer o alfabeto e não tem noção de aritmética. Tudo o que entende são várias centenas de combinações de números e qualquer coisa que se ensinar à máquina deve necessariamente ser traduzida em números. Os pequenos interruptores já mencionados podem “lembrar-se” dos números; um determinado padrão composto de interruptores on e off representa um número (no sistema binário, que usa os algarismos 0 e 1 para exprimir todos os números). Poder “lembrar-se” — ou, em outras palavras, armazenar informação — é vital para o funcionamento do computador; a memória eletrônica num TK 85 manipula informações equivalentes a seis páginas deste curso sobre computadores (e poderia armazenar muito mais se fosse em fita).

Tão bem quanto armazenar números na memória, um computador faz operações com eles (somadas, subtrações, comparações) e também é capaz de movimentá-los no interior da memória. Tudo o que a máquina faz tem como ponto de partida essas operações simples. Suponha que você queira armazenar um texto no computador. Será preciso utilizar um código, de modo que para cada letra do alfabeto corresponda determinado número: neste caso, o computador pode arquivar palavras em forma de números e dispô-los de várias maneiras. É claro que você não necessita inventar esse código, porque o fabricante da máquina já o incorporou nos programas do computador.

O que é um programa? É uma lista de instruções dadas ao computador para que cumpra as operações (adição, comparação etc.) numa certa ordem, da mesma forma que um molde de tricô indica como realizar uma sucessão específica de pontos, para se fazer uma peça de vestuário. Mas o que são essas instruções e como chegam ao computador? Na verdade, são apenas mais números, da mesma forma armazenados na memória do computador!

Será isso um paradoxo semelhante à questão: o que surgiu primeiro, o ovo ou a galinha? O computador não pode fazer nada sem um programa que lhe instrua o que fazer; toda vez que a letra “A” é teclada, um programa no interior do computador precisa “varrer” o teclado, verificar que tecla foi apertada e dizer então ao computador o número do código para tal letra. Quando o computador foi pela primeira vez projetado, o programa de varredura do teclado não existia. Alguém precisava colocar meticulosamente os números certos diretamente na memória do teclado, recorrendo a instrumentos especiais, de forma que ela pudesse compreender as letras do teclado e exibi-las na tela.

Uma vez prontos os programas básicos, tudo fica mais fácil. Agora pode-se arquivar números novos na memória do computador — basta teclá-los. O pro-



Código para letras e números
Um grupo de oito interruptores permite 256 combinações de on e off. É mais do que suficiente para um código individual (usando apenas zeros e uns) que representa cada uma das letras, numerais e sinais especiais no teclado de um computador. A ilustração mostra como as letras X e Y são representadas no interior de um computador usando o código ASCII.

cesso é chamado de programação em linguagem de máquina; futuramente falaremos mais dele. Como esse tipo de programação é difícil e cansativo, alguns técnicos elaboraram programas (em linguagem de máquina) que traduzem palavras inglesas, como PRINT (imprimir), BEEP (alarme), LOAD (carregar) e LIST (listar), em instruções de código de máquina que o computador pode usar. Quase todos os microcomputadores têm, embutido, um programa desse tipo. Assim, você pode programá-los utilizando uma linguagem simples de computador chamada BASIC, em vez de seqüências de números. Mas, toda vez que recorrer ao BASIC, lembre que o produto de muitas horas de trabalho do programador já está embutido no computador, trabalhando para você.

Com linguagens de computador como o BASIC é bem fácil fazer programas que realizem tarefas úteis ou de entretenimento, sem a necessidade de preocupar-se com a complexa e intensa atividade iniciada no interior da máquina para verificar que foi teclada a letra "A". Você poderá, por exemplo, escrever um programa que arquivará os nomes das capitais do mundo e que dará à questão "Qual é a capital do Afeganistão?" a resposta: "Cabul". Ou seja, o cérebro eletrônico sabe apenas aquilo que lhe foi dito antes; não descobre nada por si mesmo.

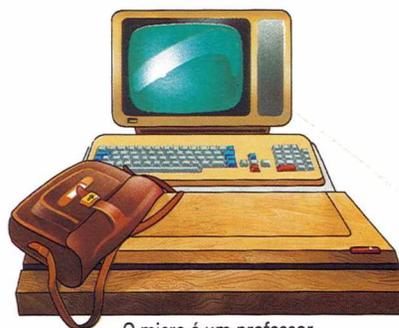
Se é assim, por que os computadores são tão úteis? Porque podem armazenar enorme quantidade de informações e manipulá-las muito melhor do que as pessoas. Naturalmente, não precisa ser *you* a pessoa que colocará pela primeira vez a informação no computador. É possível adquirir um programa elaborado por outra pessoa, com todas as capitais do mundo armazenadas. Neste caso, o computador será uma espécie de livro eletrônico de referências. Como alternativa, é possível comprar um programa que trabalhe sobre informações que você mesmo lhe fornece: um "processador de palavras" que permita digitar, corrigir e fazer uma nova redação de documentos e cartas; ou um programa de banco de dados que permita catalogar, por exemplo, informações sobre uma biblioteca e obter em poucos segundos respostas para questões do tipo: "Que livros de Jorge Amado publicados até 1970 constam desta biblioteca?"

O fato de o tolo computador compreender apenas números é, na prática, muito mais uma demonstração de eficácia que de incapacidade. Se os computadores lidassem com determinadas coisas que nos interessam, como palavras ou cores, seriam muito mais complexos do que os modelos atuais, e para cada tipo de tra-

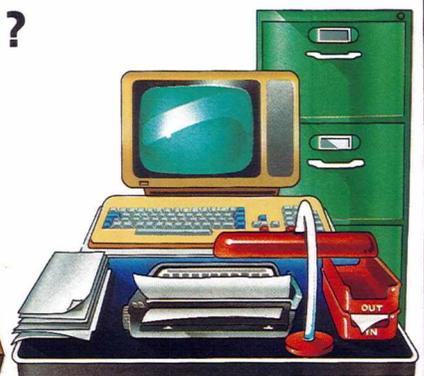
balho seria necessário um modelo diferente de máquina. De qualquer modo, como se poderia armazenar (a noção de) verde na memória de um computador? Já que esta máquina não precisa "compreender" as informações que processa, da forma como uma pessoa o faz, ela pode trabalhar com praticamente tudo. Basta que o programador descreva o tema de um modo que seja redutível a números.

Que tal, por exemplo, fazer um computador produzir música? Não espere encontrar sons reais no seu interior; em vez disso, faça corresponder a cada nota da escala um número proporcional a seu tom ou freqüência. Então, é possível fazer com que o computador envie os sinais elétricos (usados na repre-

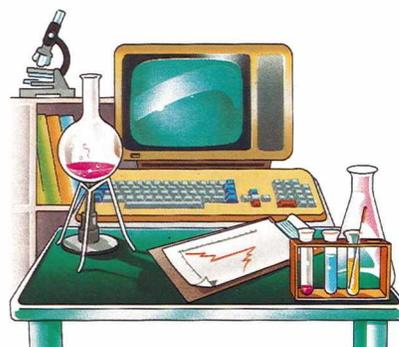
POR QUE SOFTWARE ?



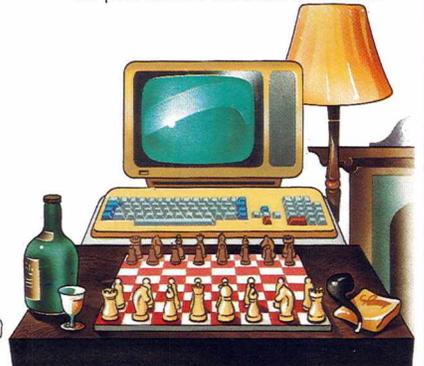
O micro é um professor



Útil para cartas e controles comerciais

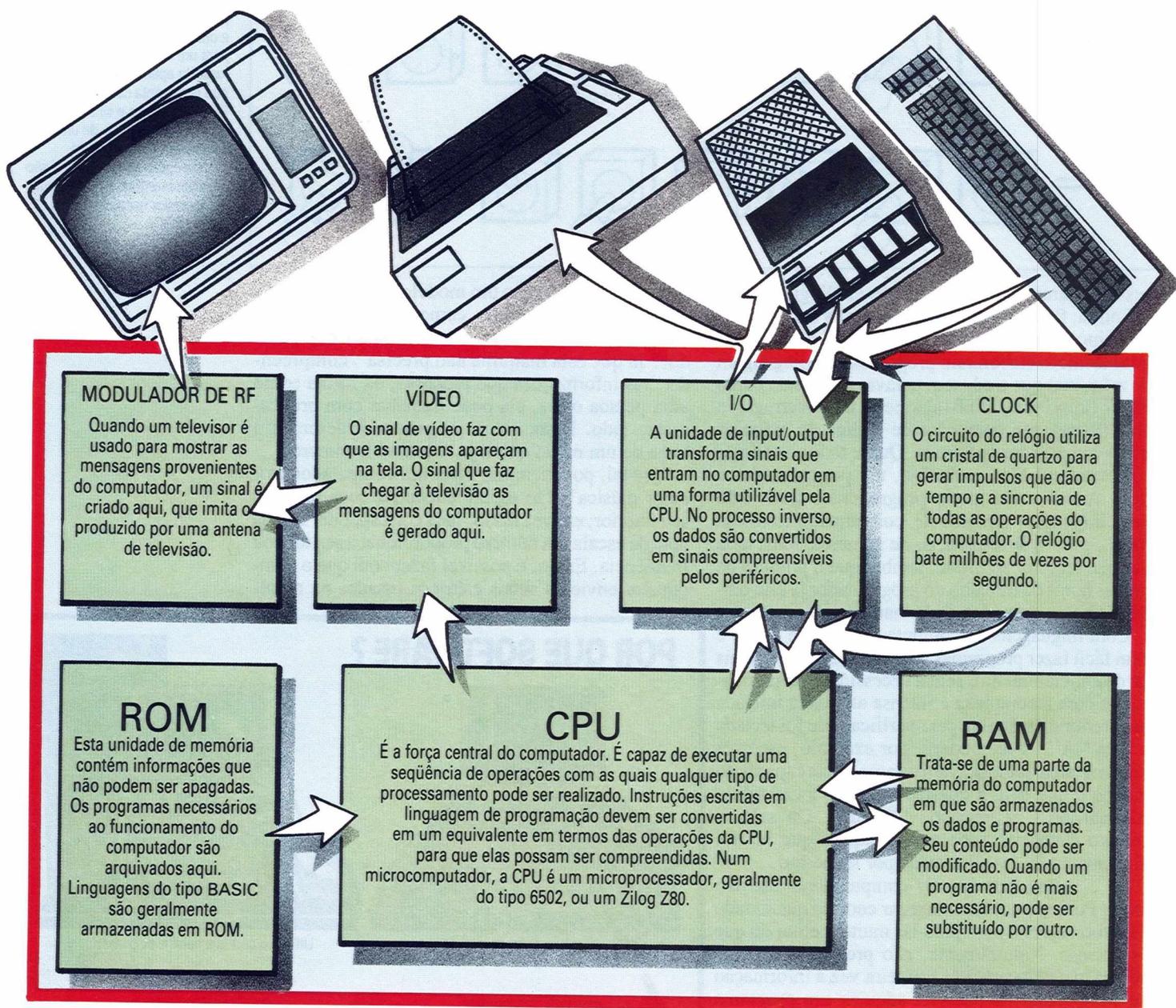


Os computadores auxiliam a ciência



Jogos e entretenimento

O computador é uma máquina versátil, capaz de realizar muitas tarefas. A mesma máquina pode ser usada pelo homem de negócios com software apropriado, pelos tecnólogos, com software para estatística; ou para entretenimento. O software, como se vê, é que determina o que o computador faz.



O que se passa por dentro

Para estabelecer um sistema completo de computador e torná-lo pronto para uso é necessário fazer uma conexão de diversas unidades. Os chips de silício que tornaram possível o microcomputador estão embutidos numa caixa, normalmente sob o teclado. Tire a tampa e o que você encontrará serão os principais componentes do seu equipamento.

sentação de números), não para uma tela como é o mais comum, mas para um alto-falante.

Como se pode disparar mísseis, ao longo da tela, contra os "invasores do espaço" que se aproximam? Simplesmente movimentando alguns números que representam o formato de um míssil para outro local da memória do computador que atua como um "mapa" da tela. Imagens, movimento, cores e sons: tudo isso pode ser traduzido para um código numérico e manuseado pelo computador; depois, recorrendo a um adequado "transmissor", como a televisão ou um alto-falante, é possível transformar novamente (as imagens, as cores, os sons, etc.) em sinais que tenham sentido para as pessoas.

Portanto, a resposta final para a pergunta "O que é um computador?" poderia ser: é uma máquina que armazena sinais elétricos que representam números.

Alguns desses números são instruções que dizem ao computador o que fazer com os outros números. A máquina seguirá essas instruções com exatidão, sem se cansar e sem cometer erros (embora reproduza fielmente os erros do programador), a uma média de vários milhares de operações por segundo. Resultado dessa incansável manipulação: mais números, por sua vez "traduzidos" nas informações que desejamos e numa forma que podemos compreender. É a atividade dos programadores (seres humanos) que torna o computador um instrumento útil, explorando sua destreza com números para realizar tarefas que têm significado para você; ou recebendo informações de vários tipos e transformando-as em soluções que seriam, se não dispuséssemos dos computadores, tão cansativas como complexas e consumiriam um tempo enorme de sua vida.



Domine seu micro

O hardware, conjunto de componentes físicos do computador, não funciona sem o software conveniente. Conheça o significado deste termo fundamental e avalie os softwares existentes.

Software é a parte invisível de um sistema de computador. Sem ele, o computador é uma simples massa inerte de componentes eletrônicos e nada pode realizar.

Com uma olhada nos chips de silício de um computador verifica-se que consistem em milhares — talvez milhões — de interruptores eletrônicos. Assim como o interruptor de uma lâmpada comum não pode ligar e desligar sozinho, os interruptores do computador precisam ser acionados, mas não simultaneamente: cada um deles deve ser ligado ou desligado separadamente e na seqüência exata em relação às centenas de outros interruptores. É esta a função do software.

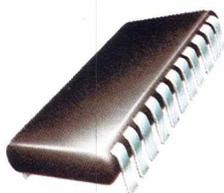
O software consiste em instruções que fazem o computador funcionar. As instruções são dadas sob a forma de números que, apresentados à CPU (o coração do computador; ver p. 4), ligam e desligam os interruptores internos, fazendo-os executar operações específicas. O computador só “compre-

ende” esses números se estiverem na forma binária (mediante a conversão nos dígitos 1 e 0, como é explicado no capítulo Bits e bytes, p. 32).

Esses dois dígitos, inteligíveis para o computador (no sentido de que o instruem para a execução de determinadas tarefas), são o resultado final de uma longa série de operações, originadas na mente do programador. Podem-se encontrar os programas (“programa” é o termo que designa qualquer unidade de software) sob formas diversas. A única afirmação precisa que se pode fazer sobre todos os programas é a de que devem resultar em algo compreensível pelo computador.

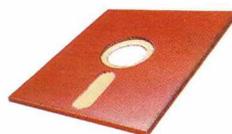
Veja um exemplo: um engenheiro de tráfego deseja controlar uma série de semáforos, utilizando um computador. Para isso, o computador que fará o controle vai precisar de um programa que o instrua a induzir a seqüência correta de mudança de cores (todos os faróis verdes ao mesmo tempo seria inconcebível). Antes de desenvolver esse software, o en-

Para fazer funcionar seu computador é necessário “alimentá-lo” com software (uma série de dados eletrônicos). Nos dispositivos aqui mostrados, esses dados são armazenados. Representam os quatro métodos mais comuns de fornecimento de software, tendo cada um vantagens específicas. Os softwares são feitos “sob medida” para cada marca de computador — um programa desenvolvido para determinada marca não funcionará necessariamente em outra.



ROM

A ROM (Read Only Memory) é um dos principais dispositivos de memória do computador que permite o armazenamento permanente de programas. A maioria dos microcomputadores é equipada com chips ROM que contêm a linguagem BASIC de programação. Outras ROMs podem ser compradas para aprimorar o desempenho do computador pelo acréscimo de mais uma linguagem, ou para transformá-lo em um processador de palavras.



DISCO FLEXIVEL

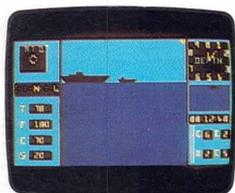
Pode-se armazenar software gravando-o em um disco feito com um filme magnético. A gravação é efetuada em trilhas como as faixas de LP comum, através de uma cabeça magnética de leitura/gravação, que também lê o programa, quando necessário. Os discos possuem grande capacidade de armazenamento e operação em alta velocidade.



CASSETTE

Geralmente, o software é fornecido em fitas cassete idênticas às de gravação de som e é nessa forma que em geral os programas de jogos se encontram à venda.

O computador é alimentado mediante a conjugação do equipamento a um gravador cassete comum que “toca” a fita do programa. A fita pára após o carregamento e geralmente o computador não precisa voltar a “consultá-lo”.



CARTUCHO

É basicamente uma ROM embalada em proteção adequada. Alguns microcomputadores têm encaixes de fácil acesso, nos quais os cartuchos são inseridos. Os softwares geralmente vendidos em cartucho são linguagens de programação (como o BASIC) ou sofisticados videogames.





genheiro deve decidir com cuidado que tarefas o computador executará.

Normalmente, as instruções seriam formuladas em sentenças da linguagem comum (ex.: em determinado momento, quero que o sinal n.º 1 mude para amarelo, mantendo ao mesmo tempo o vermelho; em seguida, que as luzes vermelha e amarela se apaguem e a verde acenda). Mas, assim estruturadas, essas instruções são incompreensíveis ao computador; portanto, precisam ser convertidas em um programa. Para tal, deve ser usada uma linguagem de programação, como BASIC, por exemplo.

Esse tipo de linguagem permite que pensamentos logicamente ordenados em linguagem comum sejam reescritos de forma inteligível para o interpretador BASIC. Interpretador BASIC é, na verdade, um programa que converte o programa original (escrito em BASIC) em uma forma compreensível à CPU — Unidade Central de Processamento. Nesta forma, o software é chamado “linguagem de máquina” ou “código de máquina”.

O software que você comprar para seu computador estará quase sempre em linguagem de máquina e armazenado de modo imediatamente acessível ao computador. Às vezes, o software é armazenado na memória ROM (interna) do computador. Mais frequentemente, é fornecido em cassete ou disco flexível. Esses elementos não são exatamente software: apenas os meios pelos quais o software é fornecido. Para uso do computador, é preciso transferir o software do cassete (ou do disco flexível, ou da ROM) para o computador. Só após o carregamento (nome do processo de transferência de software) o programa poderá ser executado.

FIRMWARE

A origem do termo hardware (equipamento físico) é evidente — é a parte física e eletrônica do computador: conexões para fornecimento de energia, teclado, chips de silício etc. Já o termo software (suporte lógico) tem origem em sua natureza intangível, pois consiste simplesmente em uma série de instruções.

No início da era da computação — há uns trinta anos —, o software era codificado e armazenado em fitas de papel perfuradas, semelhantes às de telex. Mais tarde, cassetes e discos magnéticos as substituíram. Na década de 70, inventou-se uma nova técnica para armazenamento de software em ROMs: os chips. Os chips ROM têm instruções software neles incorporadas no processo de fabricação. Essa combinação de software “intangível” e hardware “concreto” chama-se firmware (suporte lógico inalterável).

Compra de software

Com algum dinheiro disponível, talvez você pense: “Acho que vou comprar um carro”. Mas ninguém pensaria: “Acho que vou comprar um equipamento”, pois as perguntas decorrentes seriam: “Que tipo de equipamento? Que trabalho espero que realize?”

Ocorre o mesmo com o software. O computador não funciona sozinho: é o software com que você o equipa que o transforma em videogame, em máquina de escrever automatizada ou em máquina de contabilidade. Assim, a primeira decisão a tomar refere-se ao que você deseja de seu computador.

Feito isso, o problema seguinte é definir-se sobre o software adequado a suas necessidades. No processo de busca do software mais conveniente, você irá naturalmente aperfeiçoando sua escolha, ao analisar suas necessidades concretas. Se o ponto de partida for como divertir as crianças no domingo à tarde, o passo seguinte será descobrir que tipos de programa podem fazer isso: os jogos abrangem desde videogames do tipo “caça aos inimigos” até complexas e motivadoras simulações de fantasia. Feita a escolha, a etapa seguinte é verificar se o software está ou não disponível para o seu equipamento.

Uma vez que diferenças entre computadores não são apenas superficiais (cada um tem seus próprios componentes eletrônicos internos e requer desenvolvimento especial de software), dificilmente os modelos são compatíveis: um programa feito para o

Manipulação de palavras



Processador de texto

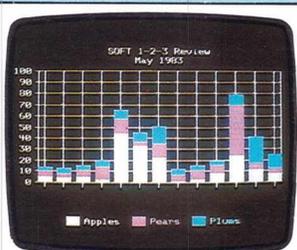
Com um software processador de palavras, seu computador ultrapassa a máquina de escrever. Mesmo boas datilógrafas cometem erros, mas com um processador de palavras as cartas serão sempre impressas com perfeição e grande produtividade. O teclado do computador substitui o da máquina de escrever, e a tela substitui o papel. As palavras escritas aparecem imediatamente na tela, assim como na folha datilografada. Mas aí terminam as semelhanças e começam as vantagens do computador.

Podem-se corrigir erros, na tela, instantaneamente, reescrever ou suprimir palavras e até eliminar parágrafos. Além disso, se a reordenação de sentenças permitir melhor expressão de seu pensamento, você pode fazê-lo na própria tela. As palavras ou sentenças que deseja reordenar na “página” são suprimidas temporariamente (o programa processador de palavras elimina-as da tela, armazenando-as na memória do computador) e você pode depois inseri-las onde quiser.

Quando o documento atingir exatamente a forma desejada, poderá ser impresso usando-se a impressora do computador ou ser armazenado em cassete ou disco flexível, para utilização posterior.



Balanco



Pacote de contabilidade

Uma vez que os computadores efetuam operações matemáticas, é natural haver muitos programas que auxiliem na área financeira. Numerosos são os softwares desse tipo: da manutenção automática de livros fiscais ao serviço completo de contabilidade. Esses programas precisam ter acesso a grande quantidade de informações e armazenar grande quantidade de registros. Por isso, exigem, geralmente, ao menos uma unidade de discos flexíveis para atender às exigências de armazenamento em larga escala.

Programas contábeis geralmente funcionam por meio de uma série de perguntas (apresentadas na tela do computador) e de respostas (fornecidas pelo operador). O programa utiliza as informações registradas pelo operador, realiza os cálculos necessários e armazena os resultados no disco ou os imprime na forma adequada.

Esses programas incluem emissão automática de faturas, reorganização de estoque, controle de livros fiscais e acompanhamento de trabalhos em andamento. Esse tipo de software, embora caro, pode significar bom investimento, pois diminui despesas com funcionários e fornece resultados mais rapidamente.

Arquivamento



Bancos de dados

Os computadores podem acessar arquivos de informações muito mais depressa do que o fazem as pessoas; quanto maior a quantidade de informações desejada, maior a utilidade do computador. Em sua forma mais simples (e barata), o banco de dados é um pouco mais do que uma caderneta de endereços computadorizada, em que são localizados nomes, endereços e números telefônicos. Programas de bancos de dados mais sofisticados e caros executam operações muito mais complexas.

Para se ter idéia do potencial de um banco de dados, pense em um botânico que compila dados para um livro sobre cogumelos exóticos e venenosos: ele fará arquivos extensos sobre as diferentes espécies e habitats, extrairá notas de obras de consulta amplamente variadas e organizará listas infundáveis de especialistas.

Antes de existirem os computadores, essas informações tinham de ser escritas em cartões e arquivadas. Com um computador e um programa de banco de dados as informações são armazenadas na memória e o botânico obtém respostas imediatas a suas questões: se necessitar de uma lista em ordem alfabética de todos os livros que incluem as palavras "venenoso", "cogumelos" ou "fungos", o banco de dados pode fornecê-la.

Os bancos de dados são, em geral, bastante caros e normalmente encontram-se disponíveis apenas em discos flexíveis.

Manipulação de números



"Folha eletrônica"

A folha eletrônica (spreadsheet) é a solução fornecida pelo computador para questões hipotéticas, antes resolvidas com calculadora e muito papel. O comércio de qualquer produto abrange muitas variáveis e a mudança de uma delas geralmente afeta as demais.

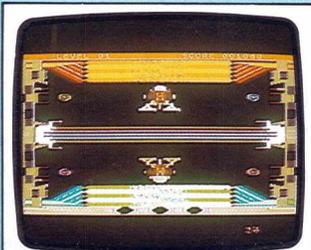
Observe problemas que um dono de sorveteria gostaria de resolver: "Teríamos mais lucro com a redução do preço do sorvete mantendo o mesmo número de balconistas ou devemos aumentar o preço e empregar mais dois?"

A escolha provavelmente influenciará toda a transação comercial — reduções nos preços podem representar aumento nas vendas mas diminuição nos lucros. A folha eletrônica vai proporcionar resultados imediatos para esse tipo de problema.

Todos os números essenciais a serem manipulados estão dispostos em um quadriculado de linhas e colunas e a relação entre cada coluna e linha é especificada (por exemplo, cada número na coluna C é obtido subtraindo-se o número da coluna A do número da coluna B). Uma vez agrupados todos os dados reais e hipotéticos, a alteração de qualquer número isolado pode ser efetuada e seu efeito sobre os outros números é imediatamente visível.

Folhas eletrônicas são utilizadas por empresários para planejar custos, ou engenheiros e cientistas, que precisam utilizar dados numéricos muito variáveis e exigem geralmente unidades de disco e impressoras.

Lazer



Jogos

Os computadores não servem apenas para processamento de números e palavras. Podem fornecer horas de lazer, se usados com um dos muitos programas de jogos existentes (xadrez, gamão e diversões eletrônicas, como o "pouso lunar" e simulações de voo). Há também jogos de aventuras extraordinariamente complexos que duram dias e mesmo semanas (ver p. 22). Muitos deles não são apenas divertidos: têm considerável valor educacional.

Esses jogos exigem atenção intensa e o fornecimento contínuo de dados pelo jogador. O fornecimento dá-se geralmente por meio do teclado; uma tecla pode ser usada para bombardear um "laser" ou um "míssil" ou para controlar o movimento de alguma coisa na tela. O número de teclas usadas varia em função do jogo e dos controles exigidos pelo programa.

Uma alternativa comum ao fornecimento de dados por teclado são os comandos joystick. São ligados ao computador, funcionam de modo semelhante ao dos manches de avião, permitem maior controle e tornam os jogos ainda mais divertidos.



Software pronto para uso

A maioria dos softwares é vendida em lojas na forma de disco flexível ou em cassete, acompanhados de um manual. Apple Writer ou Editex são programas padrões processadores de palavras. Consistem em um único disco flexível e em um manual que inclui explicações completas e concisas, de modo que iniciantes podem usá-lo imediatamente. Os padrões de documentação são muito variados. Alguns softwares vêm com manuais tão incompletos e mal escritos que pode ser difícil, e mesmo impossível, usá-los.

CP 500 não funcionará no Unitrón AP II (a não ser que se fabrique algum modelo especial deste). Por esta razão, você deve sempre comprar software especialmente produzido para seu equipamento.

Mesmo sabendo disso, você ainda não tem condições para efetuar a compra. O próximo ponto a ser considerado são as limitações físicas do seu equipamento. Verifique qual é a sua capacidade de memória: se for de 16 K RAM, veja se o jogo escolhido necessita de memória suplementar. De modo geral, jogos mais interessantes e sofisticados exigem programas mais longos; sendo assim, será necessária maior quantidade de memória. Lembre-se também de que o software é encontrado em grande variedade de formas físicas (ver a página anterior). Se um programa é vendido apenas em disco flexível e você tiver só um gravador cassete, não poderá usá-lo antes de comprar uma unidade de discos de alto custo. Alguns softwares (especialmente os jogos) exigem outros suplementos, como controles joystick. Provavelmente uma impressora será dispensável para os

jogos, ao contrário do que ocorre com os softwares comerciais que, com frequência, necessitam de uma para registrar os resultados.

Tipos de software

De certa forma, jogos são uma questão à parte: afinal, sua função é de entretenimento. A maioria dos softwares é desenvolvida para execução de um tipo específico de trabalho, do modo mais fácil e rápido. São inúmeras as maneiras de os programadores de software conseguirem aumentar a rentabilidade e eficiência deste. Consideremos a datilógrafa ineficiente, cujo trabalho não satisfaz ao chefe. Com um microcomputador e um tipo de software chamado processador de palavras, o computador substitui a máquina de escrever, realizando as correções na tela. Depois de corrigida, a página inteira pode ser transcrita em papel automaticamente, ao toque de uma simples tecla, evitando, desse modo, erros e perda de tempo.

Outra tarefa cansativa que se presta à computarização é a administração financeira. Muitas atividades que exigiam grande número de funcionários — como o balanço de contas de uma empresa e o cálculo de salários — são agora executadas por um software especialmente desenvolvido para esse fim. Os próprios programas já são bastante especializados; é improvável que uma só unidade de software possa atender a todas essas necessidades. Os tipos de software incluem: programas de cálculo de salários e emissão de holerites, programas de controle de estoque, para saber quais as mercadorias vendidas ou os gastos efetuados (em alguns casos, o programa automaticamente encomenda novos estoques), e até programas que auxiliam no cálculo de tamanhos e tipos mais econômicos de papel para impressão de livros e revistas.

Outra tarefa extremamente bem executada por computadores é o arquivamento e classificação de informações. Este tipo de programa é denominado banco de dados. Os bancos de dados podem substituir unidades inteiras de arquivo de referência cruzada, possibilitando rápido acesso a qualquer informação.

A última classe importante de software é a conhecida como folha eletrônica. Permite o preparo e a correção ilimitados de complexos orçamentos e previsões financeiras, dispensa o uso de enormes quantidades de papel e substitui a conhecida calculadora.

Todos os tipos de software aqui descritos são vendidos “prontos para uso”: quem os desenvolveu originariamente visualizou uma série de soluções específicas para problemas específicos. Todavia, poderá ocorrer que não se encontre à venda o software para instruir seu computador exatamente na execução do que você precisa. O que fazer, neste caso? Uma solução, embora dispendiosa, é contratar um programador que desenvolva um programa de acordo com suas necessidades. Outra, é aprender você mesmo a desenvolver programas. Munido de uma linguagem como o BASIC é possível criar programas que instruem seu computador na execução das mais surpreendentes operações. E o único gasto consiste no tempo usado para desenvolver o programa.





CP 500

Um microcomputador que se presta às mais diversas aplicações — técnicas, didáticas, comerciais —, o CP 500 possui inúmeros softwares já desenvolvidos em quase todas as áreas.

Primeiro microcomputador nacional totalmente compatível com o TRS-80, o CP 500 da Prologica possui memória RAM de 48 K (para armazenar programas e dados), 16 K de ROM (para o interpretador BASIC) e 1 K de RAM (para buffer de vídeo). Um dos modelos mais vendidos entre nós, o CP 500 foi desenvolvido dentro do conceito americano do "all together": já vem acoplado a um terminal de vídeo.

Sua tela de 12 polegadas é de fósforo verde (a escrita aparece em verde-fosforescente sobre fundo negro). Pode ser formatada por software, operando na configuração básica de 16 linhas de 64 colunas ou, usando-se o modo expandido, de 32 caracteres de largura dupla. Pode-se obter também uma resolu-

ção de 128 pontos horizontais e 48 pontos verticais. Apresenta 64 símbolos gráficos e 160 caracteres especiais, exclusivos do CP 500. Entre eles estão vários tipos de sinais idiomáticos, inclusive os 96 caracteres normais de nosso alfabeto, maiúsculos e minúsculos. Todas as teclas possuem função de auto-repetição. O versátil CP 500 também é capaz de executar gráficos de alta resolução (502 x 192).

O teclado do CP 500 é muito semelhante ao de uma máquina de escrever. Permite a digitação de textos normais e dos caracteres de controle. Tem teclado numérico separado, o que torna muito mais ágil a digitação de números.

A unidade central de processamento do CP 500 é





um microprocessador Z80 de 8 bits. Trata-se do componente eletrônico onde nascem todos os "pensamentos" do computador, processados a uma velocidade acima de 2 milhões de ciclos por segundo.

O CP 500 permite ainda a conexão de vários periféricos, como gravador cassete, 4 discos flexíveis de 5 1/4 polegadas de dupla densidade, impressoras matriciais, joystick (para jogos e gráficos), plotter e uma porta serial, padrão RS232 C. Um periférico de aplicação eminentemente didática é o Didata 16, um acessório que permite ligar o CP 500 com até 16 CPs 300 ou CPs 500. Ideal para centros de treinamento e escolas, o Didata facilita o ensino de matérias como física, geometria e matemática.

Finalmente, o CP 500 é capaz de produzir sons, quando se utiliza um programa que transforma o teclado do computador em teclado musical. Mas a grande vantagem do CP 500 sobre outros computadores é, sem dúvida, sua total compatibilidade com os modelos da família TRS-80. Dessa forma, ele pode utilizar a imensa variedade de softwares aplicativos prontos, abrangendo todas as áreas que se possam imaginar: educacional, financeira, recreativa, administrativa etc.



Impressora P 500

Modo de impressão:

- normal - 80 col.
- normal/expandido - 40 col.
- comprimido/expandido - 70 col.
- comprimido - 140 col.

Caracteres:

ASCII, caracteres especiais e gráficos, acentuação da língua portuguesa

Velocidade:

100 CPS

Buffer de memória:

2 K

Alimentação:

110/220 V

Video

Monitor profissional em fósforo verde com alta resolução

RAM
48 K de memória
RAM

ROM
16 K de memória
ROM

CPU
Z80 com clock
de 2 MHz

Unidade sonora



CP 500

MICROPROCESSADOR

Z80

CLOCK

2,00 MHz

MEMÓRIA

Possui 48 K de memória RAM e 16 K de ROM que inclui o interpretador BASIC.

VÍDEO

Tela de 12" de fósforo verde que pode operar com 16 linhas e 64 colunas ou 32 colunas. Possui todos os caracteres de nosso alfabeto (maiúsculos e minúsculos), 64 símbolos gráficos e 160 caracteres especiais.

TECLADO

Teclado alfanumérico, tipo máquina de escrever, teclado numérico separado e função de auto-repetição.

LINGUAGENS

BASIC

PERIFÉRICOS

Cassete, monitor, até 4 unidades de disco (5 1/4"), Winchester, impressora, interface serial RS232 C, modem, joystick.

DOCUMENTAÇÃO

O manual fornecido pelo fabricante contém instruções sobre a operação do equipamento e uma descrição da linguagem BASIC utilizada. Cada capítulo possui uma explicação básica sobre o assunto antes do enfoque técnico. Possui vários apêndices onde são apresentados os códigos de erro e de caracteres, um resumo dos comandos e instruções e um glossário de termos técnicos.

Unidade de disco

Uma ou duas unidades de disco (5 1/4") já incluídas no gabinete

Fonte de alimentação

Possui ventilação própria

Função reset

Teclado numérico

Teclado

Mecânico, tipo máquina de escrever



A ficha técnica

Preste atenção nas especificações técnicas de um micro. Elas revelam muitas de suas características e possibilidades.

Teclado

É especialmente projetado para uso fácil e agradável, com as teclas na disposição padrão das máquinas de escrever. Mantendo-se uma tecla abaixada, seu caractere aparece repetidamente. Apresenta letras maiúsculas e minúsculas e dispõe de um grupo separado de teclas (teclado numérico) para a entrada de números.

Display

Os caracteres ASCII (American Standard Code for Information Interchange) são um conjunto padrão de letras, números e símbolos usados por muitos computadores. Grande parte deles possui uma tela com 40 colunas e 24 linhas. Um televisor ou um monitor especial podem ser utilizados.

Memória

Os números indicam a capacidade de memória em kilobytes, ou milhares de bytes. A ROM contém os elementos necessários à operação fundamental do computador, geralmente incluindo uma linguagem como o BASIC. A RAM (Random Access Memory) é usada para armazenar os programas e dados do usuário.

Interfaces

Há entradas especiais através das quais uma impressora, equipamentos de comunicação de dados, cassete e cartuchos podem ser conectados ao computador. Um cartucho é uma ROM (Ready Only Memory) especial que pode conter um programa, uma linguagem ou mesmo um novo O/S (sistema operacional).

CPU

A CPU é a Unidade Central de Processamento — o chip de silício que é o coração do computador. O microprocessador Zilog Z80 pode ser citado como um dos mais comuns. O clock, que cronometra todas as suas operações, pode medir o tempo com precisão de até 2,2 milhões de vezes por segundo.

Componentes de um computador "típico"

- Memória** 16 Kbytes ROM, 32 Kbytes RAM, passível de expansão até 48 Kbytes RAM.
- Display** Pode apresentar um conjunto de caracteres ASCII de 24 linhas cada um, com 40 caracteres, saídas para televisor doméstico e monitor. Z80 com ciclo de 2,2 MHz.
- CPU**
- Teclado** Teclado padrão (QWERTY), dispositivo de repetição, teclado numérico com maiúsculas e minúsculas.
- Interfaces** Interface para impressora, entrada para cassete, abertura para cartucho.
- BASIC** Instruções para som e gráficos, verificação da sintaxe, mensagens de erro, descarga de tela, componentes estruturados.
- Gráficos** São obtidos com o uso de caracteres específicos ou com uma resolução de 256 linhas de 640 pontos.
- Som** Sintetizador de música, 5 oitavas, saída para hi-fi.
- Periféricos disponíveis** Unidade de cassete, unidades de discos flexíveis e fixos, impressoras, plotter, digitalizador, joystick, modem, sintetizador de voz.
- Linguagens disponíveis** FORTH, PASCAL, LOGO, LISP, PROLOG, ASSEMBLER.

BASIC

A linguagem residente do computador tem instruções que permitem utilizar suas possibilidades gráficas e sonoras. Verifica as instruções dadas, para assegurar-se de que estão corretas. Se estiverem erradas, uma mensagem de erro é emitida. Pode-se também reproduzir uma tela na impressora. Há instruções BASIC adicionais para garantir que os programas sejam feitos com boa "estrutura", isto é, sejam de fácil leitura e correção.

Som

Notas separadas ou acordes podem ser tocados com um alcance de cinco oitavas e o sinal sonoro pode ser emitido por um sistema de alta fidelidade.

Periféricos disponíveis

As unidades que podem ser ligadas ao computador incluem um gravador cassete, unidades de discos flexíveis e fixos. Todos armazenam programas e dados. Uma impressora, um plotter e um digitalizador para entrada e saída de gráficos também podem ser usados; controles do tipo joystick podem ser anexados para jogos.

Gráficos

As exposições criadas pelo videotexto podem ser mostradas na tela, através da combinação de vários caracteres para apresentar gráficos e também imagens em perspectiva de objetos.

Linguagens disponíveis

Outras linguagens podem ser usadas em lugar do BASIC. Cada uma é apropriada a uma aplicação específica.



Como sobreviver às tentações do micro

O mercado oferece uma enorme variedade de opções de compra. Saiba escolher a melhor solução para seu problema.

Videocassetes, televisores e aparelhos de som têm, pelo menos, um ponto em comum: cada qual executa uma tarefa específica. O nível de aperfeiçoamento dos diferentes modelos varia, mas um equipamento estéreo apenas reproduz som, a máquina de lavar roupa só faz isso e o televisor apenas recebe e apresenta sinais da estação transmissora.

Já o computador tem diversas aplicações. Muitas pessoas compram um mesmo modelo, mas cada qual pode determinar um ou mais tipos de tarefa para a máquina. Por esse motivo, comprar um microcomputador envolve vários fatores, que não ocorrem ao se adquirir qualquer outro produto.

Quando você for fazer essa compra, é preciso atenção cuidadosa. Antes de mais nada, deve fazer uma lista do que deseja que o computador execute. Talvez você queira aprender os fundamentos da programação BASIC; neste caso, o CP 200 ou o TK85 podem ser a máquina indicada. Talvez você queira usar o computador para jogos, ou como processador de palavras, ou para fazer a contabilidade doméstica; então um modelo CP 500, um TK2000 ou um DGT 1000 seriam a melhor escolha. Fatores como preço e confiabilidade provavelmente estarão no início da lista, que deve ser completa, e assim você evitará comprar um computador inadequado aos seus objetivos.

O microcomputador escolhido é o centro do equipamento. Para aproveitar ao máximo seu potencial, você também necessitará de um meio de guardar programas para uso futuro. Gravadores cassete ou unidades de disco são os meios comumente usados. Você precisará de um televisor para observar os programas e responder aos jogos. Muitas vezes, isso exigirá a compra de mais um aparelho de televisão, especialmente se você não mora sozinho — sua família não vai gostar de perder os programas prediletos enquanto você trabalha com o microcomputador. Algumas atividades exigirão cópias impressas e neste caso uma impressora será necessária.

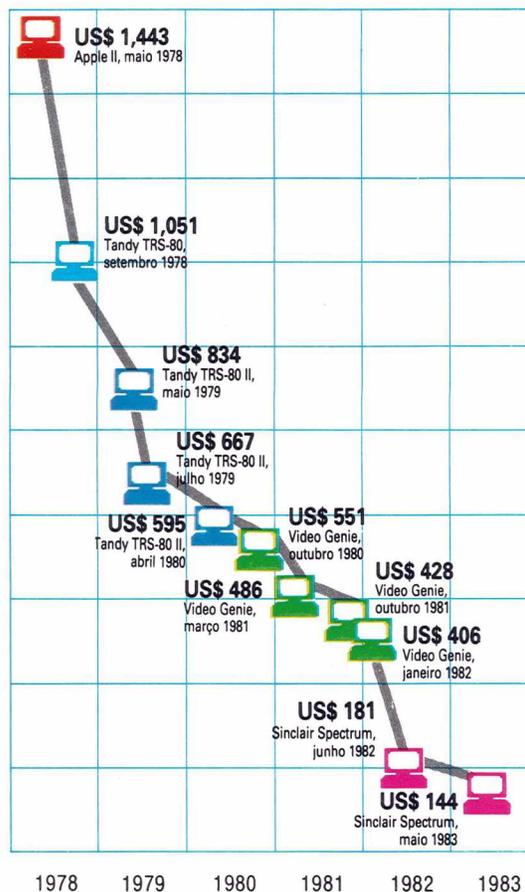
Se você pretende usar o computador apenas para jogos, o principal fator a considerar são as opções em software existentes. Microcomputadores como o TK85 ou o CP 200 serão convenientes, pois grande variedade de programas em fita cassete para eles é disponível no mercado.

Qual a capacidade de memória de que você vai precisar? Isso depende da complexidade dos programas imaginados: os mais complexos geralmente são maiores e, desse modo, exigem maior capacidade

de memória para contê-los. Processadores de palavras necessitam de grande capacidade de memória para armazenar textos. Geralmente, 32 K de RAM podem atender à maior parte das exigências, embora 16 K provavelmente sejam suficientes para rodar software de jogos de lazer com bons gráficos. Como regra geral, procure um equipamento com a maior capacidade possível de memória.

Alguns microcomputadores mais caros (como o CP 500, o Unitron AP II e o DGT 1000) são considerados adequados para uso em escritório. Em geral, têm dispositivos para acréscimo de unidade de disco, impressoras e modems, e podem ser conectados a outros computadores através de telefone.

Desse modo, assegure-se de que não esqueceu nada em sua lista: você precisa saber com total clareza o que espera de seu computador, antes de gastar dinheiro com ele.



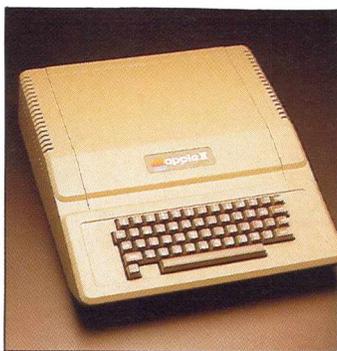
A queda do preço do micro

Esta é a evolução dos preços de venda de computadores com pelo menos 16 K de RAM nos Estados Unidos e em países europeus.

O mercado brasileiro passou a oferecer alguns destes modelos a partir de 1982, podendo-se prever uma curva de queda de preços semelhante a esta.



Qual deles?



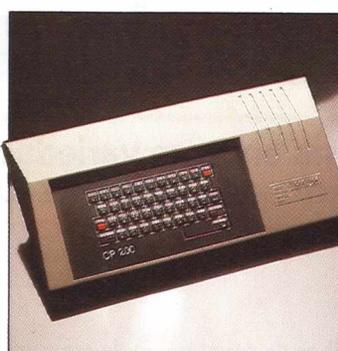
APPLE II PLUS

Memória RAM mínima: 48 K
Memória RAM máxima: 64 K
Memória ROM: 12 K
Microprocessador: 6502
Clock: 1,0 MHz
Sistema operacional: APPLIEDOS - (CP/M)
Teclado: mecânico, tipo máquina de escrever
Família: Apple
Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora, joystick, paddle.



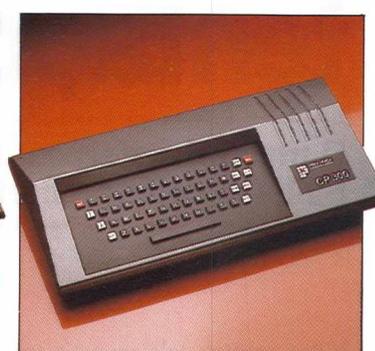
APPLE-TRONIC

Memória RAM mínima: 48 K
Memória RAM máxima: 128 K
Memória ROM: 12 K
Microprocessador: 6502
Clock: 1,0 MHz
Sistema operacional: APPLIEDOS - (CP/M)
Teclado: mecânico, tipo máquina de escrever
Família: Apple
Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4" e 8"), impressora.



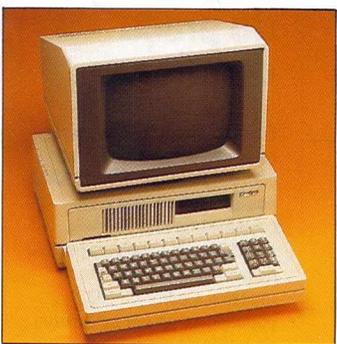
CP 200

Memória RAM mínima: 16 K
Memória RAM máxima: 16 K
Memória ROM: 8 K
Microprocessador: Z80A
Clock: 3,25 MHz
Sistema operacional: P 1
Teclado: tipo calculadora
Família: Sinclair
Periféricos disponíveis: cassete, TV, joystick.



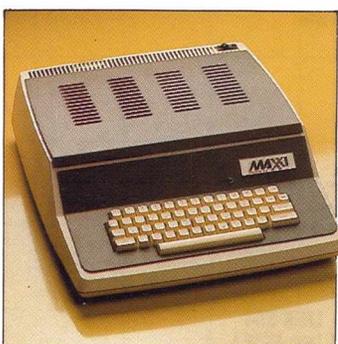
CP 300

Memória RAM mínima: 48 K
Memória RAM máxima: 48 K
Memória ROM: 16 K
Microprocessador: Z80
Clock: 2,00 MHz
Sistema operacional: DOS 500 - TRSDOS
Teclado: tipo calculadora
Família: TRS-80
Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora, joystick.



ITAUTEC JR.

Memória RAM mínima: 64 K
Memória RAM máxima: 64 K
Memória EPROM: 4 K + 32 K
Microprocessador: NSC 800
Clock: 4,0 MHz
Sistema operacional: SIM/M - CP/M
Teclado: mecânico, tipo máquina de escrever
Família: —
Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4", 8" e Winchester), impressora.



MAXXI

Memória RAM mínima: 16 K
Memória RAM máxima: 48 K
Memória ROM: 12 K
Microprocessador: 6502
Clock: 1,0 MHz
Sistema operacional: APPLIEDOS - (CP/M)
Teclado: mecânico, tipo máquina de escrever
Família: Apple
Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora, joystick.



MICRO ENGENHO 2

Memória RAM mínima: 64 K
Memória RAM máxima: 192 K
Memória ROM: 16 K
Microprocessador: 6502
Clock: 1,0 MHz
Sistema operacional: APPLIEDOS - (CP/M)
Teclado: mecânico, tipo máquina de escrever
Família: Apple
Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora, paddle.



NEXUS 1600

Memória RAM mínima: 256 K
Memória RAM máxima: 512 K
Memória ROM: 64 K
Microprocessador: 8087 e 8088
Clock: 5,0 MHz
Sistema operacional: MSDOS
Teclado: mecânico, tipo máquina de escrever
Família: IBM PC
Periféricos disponíveis: monitor, unidade de disco (5 1/4", 8" e míni-Winchester), impressora, joystick, caneta óptica.



CP 500

Memória RAM mínima: 48 K

Memória RAM máxima: 48 K

Memória ROM: 16 K

Microprocessador: Z80

Clock: 2,00 MHz

Sistema operacional: DOS 500 - TRSDOS - NEWDOS

Teclado: mecânico, tipo máquina de escrever

Família: TRS-80

Periféricos disponíveis: cassete, monitor, unidade de disco (5 1/4", Winchester), impressora.



DACTRON E

Memória RAM mínima: 64 K

Memória RAM máxima: 128 K

Memória ROM: 12 K

Microprocessador: Z80A e 6502

Clock: —

Sistema operacional: DOS 3.3 e CP/M

Teclado: mecânico, tipo máquina de escrever

Família: Apple

Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4", 8" e Winchester), impressora.



DGT-1000

Memória RAM mínima: 16 K

Memória RAM máxima: 64 K

Memória ROM: 14 K

Microprocessador: Z80

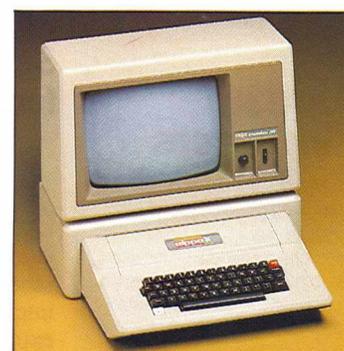
Clock: 2,5 MHz

Sistema operacional: CP/M

Teclado: mecânico, tipo máquina de escrever

Família: TRS-80

Periféricos disponíveis: cassete, monitor, unidade de disco (5 1/4"), impressora.



ELPPA II PLUS

Memória RAM mínima: 48 K

Memória RAM máxima: 48 K

Memória ROM: 12 K

Microprocessador: 6502

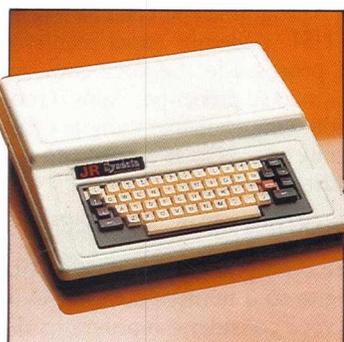
Clock: 1,0 MHz

Sistema operacional: ELPPADOS - (CP/M)

Teclado: mecânico, tipo máquina de escrever

Família: Apple

Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora.



SYSDATA JR.

Memória RAM mínima: 16 K

Memória RAM máxima: 64 K

Memória ROM: 16 K

Microprocessador: Z80A

Clock: 1,78 MHz

Sistema operacional: TRSDOS - (CP/M)

Teclado: mecânico, tipo máquina de escrever

Família: TRS-80

Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora.



TK85

Memória RAM mínima: 16 K

Memória RAM máxima: 48 K

Memória ROM: 10 K

Microprocessador: Z80A

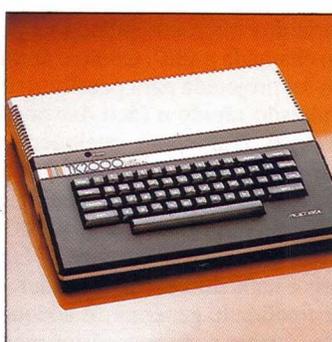
Clock: 3,25 MHz

Sistema operacional: P 1

Teclado: membrana de borracha

Família: Sinclair

Periféricos disponíveis: cassete, TV, impressora térmica, joystick.



TK2000

Memória RAM mínima: 64 K

Memória RAM máxima: 64 K

Memória ROM: 16 K

Microprocessador: 6502

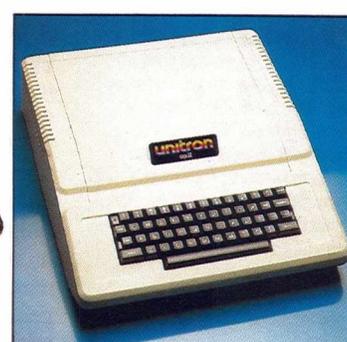
Clock: 1,0 MHz

Sistema operacional: —

Teclado: mecânico, tipo máquina de escrever

Família: Apple

Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora, joystick, paddle.



UNITRON AP II

Memória RAM mínima: 16 K

Memória RAM máxima: 512 K

Memória ROM: 12 K

Microprocessador: 6502

Clock: 1,0 MHz

Sistema operacional: UD - APPLIEDOS - (CP/M)

Teclado: mecânico, tipo máquina de escrever

Família: Apple

Periféricos disponíveis: cassete, TV, monitor, unidade de disco (5 1/4"), impressora.

Às suas ordens

Outras linguagens

O BASIC é a linguagem de programação utilizada na maioria dos microcomputadores. Mas não é, sem dúvida, a única linguagem no gênero. Antes do advento dos micros, quando para a computação eram utilizados grandes computadores, os cientistas e engenheiros usavam a linguagem chamada FORTRAN. No mundo dos micros, são também empregadas as linguagens PASCAL, FORTH e LOGO.

Pascal

Tal como o BASIC, o PASCAL foi desenvolvido inicialmente como linguagem para estudantes de programação. Tem ótimo conceito entre os professores, pois possibilita elaborar programas refinados. A linguagem PASCAL é geralmente fornecida em discos flexíveis e tende a ser de alto custo. A linguagem PASCAL pode ser utilizada para escrever programas sofisticados e extensos.

Forth

Os programas escritos em FORTH são menos parecidos com a língua inglesa do que os feitos em BASIC ou PASCAL. O FORTH é também mais difícil para se aprender. É uma linguagem poderosa, pois programas complexos podem ser escritos em poucas linhas. Com a linguagem você define seus próprios comandos, enquanto no BASIC esses comandos são predefinidos.

Logo

A linguagem LOGO é relativamente nova e tem-se difundido na educação. Possui a grande vantagem da simplicidade para o ensino de jovens e crianças. Auxilia no aprendizado das técnicas de programação e também possibilita um enfoque lógico no desenvolvimento de programas. Com essa linguagem, desenhos são facilmente produzidos na tela. Uma "tartaruga" mecânica pode ser conectada ao computador. Comandos simples digitados no teclado movimentam a "tartaruga", fazendo-a desenhar linhas e figuras.

Se você "falar" da maneira certa com seu computador, ele fará exatamente aquilo que você pedir. Sem cometer erros.

É perfeitamente possível a qualquer pessoa usar um computador — em casa ou no trabalho — sem nada saber sobre a maneira como ele funciona. Aqui tem início o MICROCOMPUTADOR — CURSO BÁSICO, que explica passo a passo, desde o princípio, tudo o que você precisa conhecer para criar seus próprios programas.

Muitas pessoas, após algum tempo, acham enfadonhos os programas e jogos que compraram prontos para o microcomputador. Elas querem saber se podem modificá-los ou mesmo escrever programas próprios. Como um computador nada pode fazer por si só, ele precisa receber uma série de instruções que lhe diga exatamente, em pormenores, o que fazer e como fazer. Essas instruções são chamadas "programa" e a arte de criá-lo é denominada "programação".

Não há nada difícil em programação. Você nem mesmo precisa ser bom em matemática, a menos, é claro, que queira escrever programas que realizem tarefas matemáticas. Para começar, basta apenas conhecer o BASIC.

Sua primeira linguagem

A maioria dos microcomputadores possui uma linguagem chamada BASIC. Como o nome diz, essa linguagem foi projetada para possibilitar aos iniciantes o aprendizado rápido e fácil das noções básicas de programação. Tal como qualquer linguagem humana, o BASIC tem seu próprio vocabulário e sua gramática, embora bem menos complexos que os da língua inglesa ou portuguesa. O BASIC utiliza pequeno número de palavras em inglês, facilmente reconhecíveis e nada difíceis de aprender. Como linguagem de finalidade genérica, é adequada tanto para iniciantes quanto para os usuários mais experientes.

Há um inconveniente, porém: durante os últimos anos, diversos fabricantes de computadores introduziram suas próprias modificações. Por isso existe grande número de variações no BASIC, especialmente com relação aos comandos de controle dos aspectos mais recentemente desenvolvidos da máquina — como cores, gráficos e sons. Todas as variações do BASIC existentes nos computadores mais populares encontram-se no quadro "A propósito...".

De um computador para outro, há variações no BASIC, tornando praticamente impossível escrever um programa nessa linguagem, menos ou mais complexo, que funcione em qualquer computador. Mas a linguagem tem um núcleo comum, que é normalmente o mesmo em todas as máquinas. Vamos concentrar-nos nesse núcleo e, à medida que o curso progredir, veremos programas de maior complexidade.

Os primeiros passos

Começaremos escrevendo um pequeno programa para ver o que acontece. Esse programa mostrará que o computador aparentemente cometeu um erro. Ligue o micro e digite o programa exatamente como é apresentado, incluindo os espaços. O <CR> ao final de cada linha é para lembrá-lo de que deve bater a tecla RETURN. Nos diversos micros, essa tecla pode ser também a ENTER ou a <↵>.

```
10 REM COMPUTADORES JAMAIS COMETEM
    ERROS <CR>
20 PRINT "DIGITE UM NUMERO" <CR>
30 INPUT A <CR>
40 LET A = A + 1 <CR>
50 PRINT "ACHO QUE O NUMERO QUE VOCE
    DIGITOU FOI "; <CR>
60 PRINT A <CR>
70 END <CR>
```

Após ter digitado o programa, datilografe LIST <CR>. O programa que você acaba de digitar deve reaparecer na tela. LIST é uma instrução para o computador "imprimir" uma listagem do programa que está na memória. Se o programa apareceu na tela de modo certo após o LIST, podemos experimentar o comando RUN, para processá-lo. Se você cometeu um erro de digitação, não se preocupe. Após o programa ter sido LISTado, simplesmente redigite qualquer linha que contenha erro. Não se esqueça do número da linha. Experimente digitar

```
25 REM AQUI ESTA OUTRA LINHA "REM" <CR>
```

e, em seguida, LISTe o programa novamente. Para eliminar a linha, digite só o número dela, seguido por <CR>. Quando você considerar o programa digitado corretamente, pode processá-lo, digitando RUN <CR>. Experimente isso e verá na tela:

```
DIGITE UM NUMERO
```

Continue e digite um número. Por exemplo, o 7. (Faça uso de numerais; o computador não reconhecerá "sete" como 7, a menos que o tenhamos programado para tal.) Se você digitou 7, na tela deverá aparecer:

```
ACHO QUE O NUMERO QUE VOCE DIGITOU FOI 8
```

Será que realmente o computador cometeu um erro, ou estava simplesmente obedecendo a ordens? Se observarmos o programa, linha por linha, veremos qual instrução fez com que ele agisse assim. Eis a primeira linha:

```
10 REM COMPUTADORES JAMAIS COMETEM
    ERROS
```



REM significa observação (REMark em inglês). Tudo o que aparecer na mesma linha após REM é ignorado pelo computador. As observações são uma forma útil de lembrá-lo do que o computador está fazendo. Este REM em particular é apenas um título, não nos indica o que o computador está fazendo. Posteriormente, veremos neste curso como são úteis essas observações REM, quando escritas de forma adequada. Agora, vejamos:

20 PRINT "DIGITE UM NUMERO"

Em BASIC, a parte que se segue à palavra PRINT é "impressa" na tela do computador. Observe que a sentença está entre aspas. Uma das regras do BASIC é que os caracteres (letras) que se seguem ao comando PRINT entre aspas aparecerão exatamente como foram digitados. Veremos outra forma de utilização do PRINT na linha 60. Em seguida, temos:

30 INPUT A

Pularemos esta linha por enquanto, retomando-a após observar a linha 40.

40 LET A = A + 1

A letra A é aqui empregada como uma variável. A variável pode ser considerada uma caixa rotulada que contém um número ou alguns caracteres. Em vez de termos de lembrar o que está na caixa, devemos apenas saber o nome da caixa para consultá-la. É como dizermos: "Dê-me a caixa rotulada B", em vez de: "Dê-me a caixa que contém parafusos de 15 mm".

Nessa linha temos uma caixa chamada A, conhecida como variável, pois o valor que nela colocamos pode variar. Podemos atribuir qualquer valor a uma variável. Um valor foi dado à variável A, na linha 30; assim, vejamos como isso se fez:

30 INPUT A

Usar a palavra INPUT é uma das formas de fornecer, em BASIC, um valor específico a uma variável. Quando um programa BASIC chega a uma linha que se inicia com INPUT, ele espera que algo seja digitado no teclado. INPUT A informa ao computador que temos uma variável chamada A e tudo o que for digitado no teclado será associado a essa variável. Digitar 7 <CR> neste ponto coloca 7 na caixa A, ou, utilizando o jargão de computador, associa o valor 7 à variável A. Agora sabemos o que é uma variável e vamos novamente observar a linha 40.

40 LET A = A + 1

O nome da variável à qual um valor foi associado sempre aparece à esquerda do sinal de igualdade. Aqui, estamos dando um novo valor para A. A declaração significa "LET (Deixe) o novo valor de A igual ao antigo valor mais 1". O antigo valor era 7. Agora temos 7 + 1, o novo valor é 8.

50 PRINT "ACHO QUE O NUMERO QUE VOCE DIGITOU FOI ";

Temos, novamente, o comando PRINT. Ele "imprime" o conjunto de caracteres (ou seja, as palavras ou números que você digitou) entre as aspas. Ob-

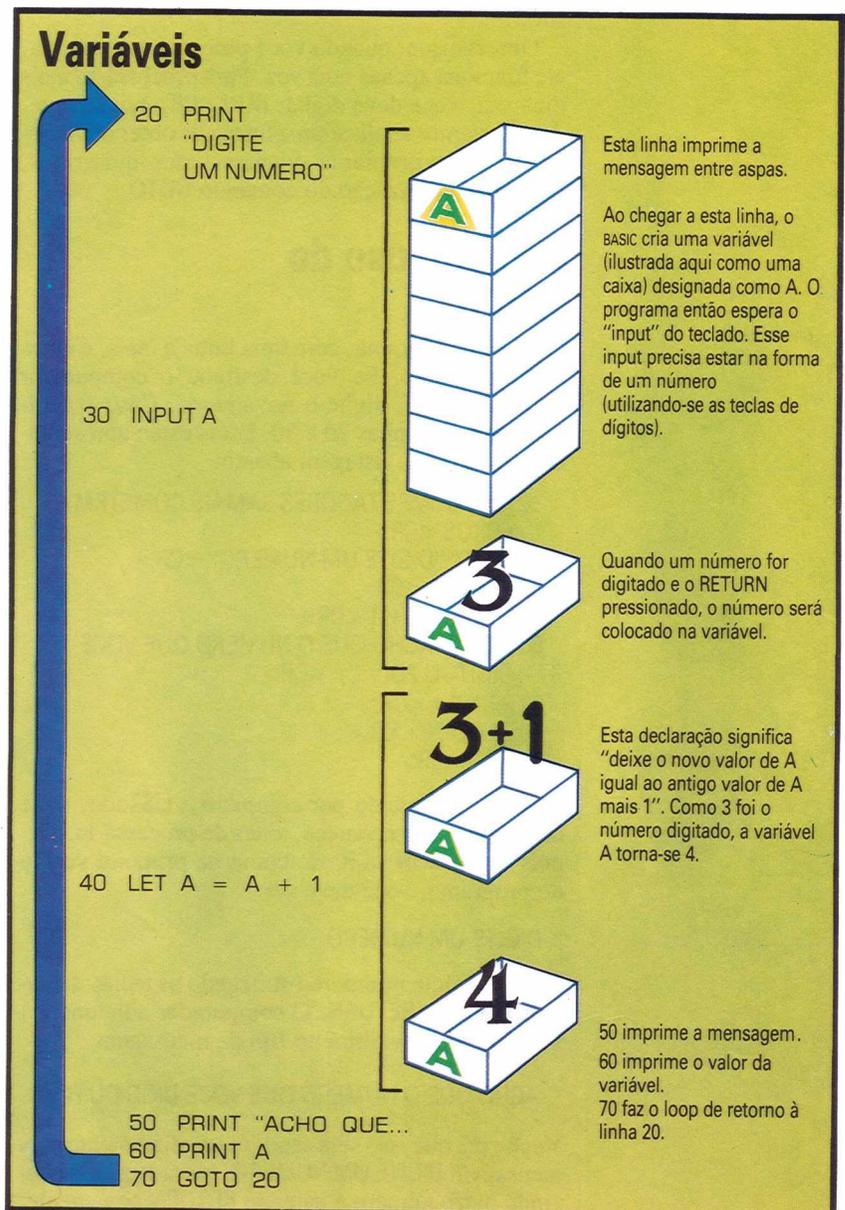
serve o ponto e vírgula no fim da linha. Ele auxilia a especificar as posições em que os itens são mostrados na tela. Neste curso, voltaremos a falar do ponto e vírgula, com maiores detalhes. Agora vejamos:

60 PRINT A

Eis outra declaração PRINT, mas aqui o A não está entre aspas. Já sabemos que o programa não mostrará um A na tela, porque precisaríamos das aspas para isso. Sem as aspas, o BASIC procura pôr uma variável com o mesmo nome do caractere após o PRINT. Se encontrar um, apresentará o valor da variável. (Se não encontrar um, dará uma mensagem de erro.) Este programa já possui uma variável chamada A e assim o BASIC mostrará seu valor. Qual é ele? Se você pensou que a resposta é 7, lembre-se de que o BASIC funciona por meio de programmas, linha por linha. Quando estávamos na linha 60, o valor do A foi alterado para 8, e é isto que será mostrado. Finalmente, chegamos em:

70 END

O quadro abaixo mostra como as "variáveis" são usadas no BASIC e como o comando GOTO é empregado para formar um loop.



A propósito...



O TK85 e o CP 200 utilizam LET como parte da instrução. Na maioria dos outros computadores isso pode ser omitido. Por exemplo, a linha 20 pode ser escrita como $A = A + 1$, em vez de LET $A = A + 1$.



Não é utilizado no TK85 e no CP 200. A última linha do programa digitado é considerada o fim do programa.



Pode aparecer na tela como duas palavras (GO TO) no TK85 ou CP 200, embora apenas uma tecla seja pressionada. A maioria dos computadores aceitará a instrução digitada em duas palavras.

A declaração END diz ao BASIC que o fim do programa chegou. Algumas versões do BASIC insistem no fato de que todos os programas devem terminar com END, enquanto outras não (ver o quadro "A propósito...").

Observe que, quando você processa o programa, ele funciona apenas uma vez. Para processá-lo mais uma vez, você deve digitar RUN <CR> novamente. Agora, vamos analisar uma forma de obter o funcionamento do programa quantas vezes quisermos, mediante a utilização do comando GOTO.

Fazendo uso do GOTO

O mesmo programa, com uma linha a mais, é apresentado abaixo. Se você desligou o computador para descansar, digite-o novamente. Caso contrário, digite as linhas 70 e 80. Estas estão apresentadas em azul, na listagem abaixo.

```
10 REM COMPUTADORES JAMAIS COMETEM
  ERROS <CR>
20 PRINT "DIGITE UM NUMERO" <CR>
30 INPUT A <CR>
40 LET A = A + 1 <CR>
50 PRINT "ACHO QUE O NUMERO QUE VOCE
  DIGITOU FOI "; <CR>
60 PRINT A <CR>
70 GOTO 20 <CR>
80 END <CR>
```

Após tê-lo digitado por completo e LISTado, tente descobrir o que acontece, antes de processá-lo. Depois digite RUN <CR> e, como na primeira versão do programa, você deve ver:

```
DIGITE UM NUMERO
```

Digite qualquer número (utilizando as teclas de numeral) e bata RETURN. O computador adicionará 1 ao número e o exibirá no fim da mensagem.

```
ACHO QUE O NUMERO QUE VOCE DIGITOU FOI 8
```

Você verá que isso será seguido imediatamente pela mensagem DIGITE UM NUMERO, outra vez. Introduzindo outro número e batendo RETURN novamente,

você pode observar que o ciclo do programa recomeça *ad infinitum*. A razão para que isso aconteça encontra-se na linha 70:

```
70 GOTO 20
```

Quando o BASIC chega à declaração GOTO, não continua na próxima linha; em vez disso, vai para a linha de número especificado. Neste caso, está dirigido de volta à linha 20 e o programa inteiro recomeçará. Se você quiser parar o programa, perceberá que não existe forma de sair do loop. O programa simplesmente continua, aguardando sua entrada de dados.

Como seria de esperar, existem várias formas de escrever um programa, de modo que podemos sair do loop se quisermos. No próximo segmento deste curso veremos uma dessas formas. Por enquanto utilizaremos a tecla BREAK para fazê-lo. Digitar RUN <CR> fará com que o programa recomece.

Observe que ainda temos a declaração END, ao final do programa. O modo pelo qual escrevemos o programa com o comando GOTO 20 criou um loop infinito. Entretanto, algumas versões do BASIC exigem que sempre utilizemos um END no fim.

Se você não encontrou uma forma de parar o programa, tente a tecla RESET. Esta certamente vai pará-lo. Então, tente LISTá-lo novamente. Desse modo você será capaz de "editar" o programa nos exercícios abaixo. Se o programa não aparecer na tela, isso significa que o RESET em seu micro o destrói na memória e, assim, você deve digitar tudo de novo.

Exercícios

Estas questões foram cuidadosamente classificadas e pretendem ser de agradável resolução. A solução dos exercícios é uma das melhores maneiras de verificar se você compreendeu o material apresentado e está fazendo progressos reais.

Antes de começar os exercícios, procure alterar algumas linhas para ver o efeito no modo de processar-se o programa. O computador não será danificado se você cometer erros ou bater teclas erradas. Para alterar uma linha, digite o programa e depois, para verificação do resultado, LISTe-o. Todo o programa aparecerá na tela novamente. Digite o número da linha que deseja alterar seguido pela nova linha. Experimente isto:

```
10 REM COMPUTADORES ALGUMAS VEZES
  COMETEM ERROS <CR>
```

Em seguida digite LIST novamente. Observe como a primeira linha foi alterada. Se você desejar eliminar toda a linha, simplesmente digite o número da linha seguido de <CR>. Experimente:

```
10 <CR>
LIST
```

A linha 10 deve ter desaparecido. Reintroduza a linha 10, digitando a linha inteira novamente, e não se esqueça do número da linha.

■ Reescreva o programa, para que o computador re-



almente imprima o número digitado. Sugestão: retire uma linha inteira.

■ Redigite a linha 70, para que o programa vá para a linha 80. LISTE o programa. Acione o comando RUN. Por que ele não se processou do mesmo modo que antes?

■ Altere a linha 60, para que o computador imprima um A na tela em vez do valor da variável A.

■ Reescreva a linha 60, para que o computador imprima o valor da variável A outra vez. Retire a linha 10 (a linha REM) completamente. Processe o programa. Notou alguma diferença?

■ Coloque um novo REM na linha 25. Novas linhas podem ser adicionadas, simplesmente digitando o novo número seguido pela nova declaração. Coloque uma observação na linha 25 para lembrá-lo do que deve fazer a seguir (pode ser “esperar uma entrada de dados pelo teclado”). Após ter digitado a nova linha, bata <CR>, LISTE o programa novamente e verifique se suas observações aparecem no lugar certo.

■ Reescreva o programa, a fim de ele multiplicar por 10 o número que você digitou. Você precisará alterar a linha 50 para imprimir algo como o NÚMERO QUE VOCE DIGITOU MULTIPLICADO POR 10 E. Desta vez, não queremos somar o valor da variável antiga, e sim multiplicá-la por 10. O BASIC faz uso do sinal * para significar “multiplicar”. (Não utilize um X, porque o BASIC apenas o reconhece como uma letra e não como sinal de multiplicação.)

Até aqui, examinamos uma boa quantidade de assuntos. Vimos como escrever comentários que o BASIC chama de REM, como imprimir conjuntos de caracteres na tela, qual o modo de imprimir o valor das variáveis na tela e como fazer um programa ir a um número de linha especificada utilizando o GOTO.

Da próxima vez, veremos como sair de um loop, utilizando a declaração IF-THEN. Descobriremos como conseguir que o programa seja executado um número especificado de vezes, sem ficar em loop contínuo.

Além disso, veremos como a velocidade do programa pode ser diminuída, fazendo com que o computador pareça estar realmente pensando.

E o BASIC foi criado

Hoje em dia, o BASIC é a linguagem de programação mais popular do mundo. As linguagens de computador foram inventadas para permitir que o operador humano se comunique de modo mais fácil com a máquina. O BASIC compõe-se de instruções em inglês que foram combinadas de forma simples com símbolos matemáticos.

Em pouco tempo, utilizando o BASIC, você já conseguirá elaborar pequenos programas. O BASIC foi concebido em 1965 por dois professores da Faculdade de Dartmouth (New Hampshire, Estados Unidos), Thomas Kurtz e John Kemeny, com a finalidade de simplificar as linguagens existentes. Com o tempo, ligeiras variações na linguagem original foram ocorrendo. Mas o núcleo do BASIC permanece comum a todos os fabricantes.

Um programa é uma seqüência de instruções que o computador executa para desenvolver uma tarefa específica. O programa aparece na tela do micro como uma série de linhas numeradas. Cada linha contém uma instrução e o número permite ao computador obedecer aos comandos na ordem correta. Os comandos são rapidamente aprendidos e até mesmo programas mais complicados fazem uso apenas de combinações e repetições de comandos elementares.



O BASIC eliminou o mistério da programação e tornou a computação acessível a todos.

A maioria dos micros vem da fábrica com a linguagem BASIC embutida. Os computadores podem também ser programados em “linguagem de máquina” (designada como linguagem de “baixo” nível, pois é a que está mais próxima da estrutura da lógica encontrada nos circuitos eletrônicos). O BASIC é linguagem de “alto” nível, pois é muito semelhante ao inglês cotidiano. Existem muitas outras linguagens de alto nível planejadas para aplicações mais técnicas e especializadas, mas o BASIC é a melhor introdução a todas elas. É uma linguagem simples e poderosa.

Rumo à expansão

Para transferir e receber dados, para armazenar programas e até para alguns jogos, é necessário um hardware complementar.

Impressora

É usada quando são necessárias cópias em papel de programas ou de resultados fornecidos pelo computador. Há diversos tipos de impressora e o preço é proporcional à velocidade e qualidade da impressão.



Unidade de disco

Armazena programa à semelhança do cassete, usando um "disco flexível" em vez de fita. O disco é muito mais caro do que o gravador cassete, porém armazena mais dados e trabalha muito mais rapidamente. Em geral, é necessário ao processamento comercial.

Gravador cassete

O gravador cassete comum proporciona um método econômico de guardar o programa. Este permanece na memória enquanto estiver em uso pelo computador. Desliga a energia elétrica, o conteúdo dessa memória desaparece. Antes disso, o programa pode ser gravado em fita, que será lida pelo computador, sempre que necessário.

Controlador "track ball"

É usado em jogos. Rolando-se a bola em seu suporte, um elemento do jogo pode ser movido na tela. Permite posicionamento mais preciso e rápido do que os joysticks. É munido de botões para bombardear "lasers" etc.



O computador

Embora necessite de complementos para ajudá-lo a comunicar-se com o usuário, o computador é o centro do sistema de computação. Possui teclado semelhante ao da máquina de escrever, com algumas teclas suplementares. É munido de encaixes (geralmente na parte posterior) para acoplamento a equipamentos acessórios, como o gravador cassete, a unidade de disco ou um televisão.



Televisor

Um aparelho de televisão comum permite ao computador exibir mensagens. Quando você estiver escrevendo um programa, o que for digitado no teclado aparecerá na tela. O monitor (na figura acima, atrás do televisor) é utilizado para proporcionar imagens mais detalhadas e de melhor qualidade.

Joysticks

São semelhantes aos controles de alguns fliperamas. O modo como devem ser efetivamente usados depende do jogo escolhido. Podem, por exemplo, controlar um carro de corrida ou um personagem em um labirinto. Alguns joysticks têm um conjunto de dez ou mais teclas (dispostas como em uma calculadora); o modo como as teclas são usadas também depende do jogo escolhido.





O enigma das barras

Às vezes, misteriosas barras surgem impressas em capas de livros e nas embalagens industriais. São apenas mensagens para um computador a serviço das lojas.

Decodificação das listras



A ilustração acima mostra um código de barras, representando o número 72. Sua configuração é uma série de linhas pretas de diferentes larguras. Nesse exemplo cada conjunto inclui cinco barras, duas das quais mais largas. A posição dessas duas em cada conjunto fornece o número. As barras adicionais indicam o início e o fim de cada unidade de informação.

Há diversos métodos de codificação de dados em código de barras. Uma vez que as barras podem ser largas ou estreitas, cada uma pode representar os dígitos 1 e 0, o que conduz diretamente à matemática binária, própria dos computadores. Um código de barras um pouco diferente é o Universal Product Code, para uso comercial. Neste caso, as barras, de várias larguras, são poucas e os dados interpretados conforme a largura da linha. Quando o código de barras é iluminado, o total de luz refletida é registrado. Comparadas ao fundo branco contra o qual se apresentam, as barras pretas quase não refletem luz. A luz refletida é convertida em sinal elétrico e ampliada. Se a luz for registrada, haverá um sinal (o número 1, em código binário); se não for registrada, não haverá sinal (o número 0, em código binário). Os dados agora podem ser compreendidos pelo computador.

As barras fornecem séries de dígitos. Quanto mais largas, maior quantidade de dígitos 0 elas contêm. De modo semelhante, o fundo branco fornece séries de dígitos 1. Assim, a caneta óptica fornece ao equipamento combinações de dígitos binários a partir das quais o computador determina a composição do código de barras.

Vá a uma livraria e solicite um livro importado, de edição recente. Na contracapa da brochura, você encontrará um conjunto de listras: é um código de barras, recurso engenhoso que pode ser “lido” em fração de segundo por uma “caneta” óptica; o código, aplicado em produtos diversos, seja uma lata de refrigerante, uma caixa de chocolate ou livros e revistas, contém dados sobre a mercadoria que a caneta óptica, ao percorrer as listras, fornece diretamente ao computador. Assim, informações sobre alterações de preço, níveis de estoque e outras podem ser rapidamente obtidas. Isso contribui para o funcionamento mais eficiente da casa comercial e melhor atendimento ao consumidor.

Vejam como o método funciona no caso de uma edição em brochura. Todos os livros publicados nos principais países têm uma numeração segundo o International Standard Book Number (ISBN, Padrão Internacional para Numeração de Livros). A numeração compõe-se de um ou mais dígitos para indicar o idioma ou o país em que o livro foi publicado (os dígitos 8 e 5 referem-se aos livros publicados no Brasil); de dois a sete dígitos para identificar o editor; e de um a seis dígitos para identificar o título e a edição do livro. Assim, chegamos a um total de nove dígitos, além de um dígito de controle (que o

computador utiliza para garantir que todos os dígitos sejam fornecidos na ordem correta).

Na codificação de barras, os livros são numerados de acordo com o sistema European Article Numbering (EAN, Numeração Européia de Produtos), que utiliza treze dígitos (a maioria dos produtos comestíveis emprega, geralmente, número menor, com oito dígitos). Os três primeiros dígitos são o “sinalizador” EAN — 978 para livros; os seguintes são os números do ISBN; e, finalmente, um dígito alternativo de controle EAN. Nos Estados Unidos e na Grã-Bretanha, o ISBN é também impresso em números dispostos acima do código de barras, podendo ser lidos tanto pelo olho humano, quanto por uma leitora óptica, facilitando assim a identificação do produto.

A leitora óptica é outro aperfeiçoamento de grande interesse, com amplas consequências futuras. As máquinas atuais podem, literalmente, ler as palavras impressas, através da exploração óptica da linha. O sinal de saída da leitora é fornecido diretamente ao computador, que pode, assim, processar as informações de diferentes formas. As palavras lidas por esse dispositivo são, por exemplo, mostradas na tela, evitando a necessidade de um demorado trabalho de datilografia.

Código de um tablete de chocolate

Sensor óptico

É uma célula fotolétrica, sensível à luz. A corrente que produz aumenta em função do total de luz que a atinge. Uma vez que superfícies em preto praticamente não refletem luz, enquanto as brancas refletem quase toda a luz que as alcança, o sensor produz uma corrente maior ou menor, ao percorrer o código de barras.

Digitalizador

Os sinais do detector óptico são ampliados e convertidos em dígitos binários. Sinais altos são convertidos no dígito 1; os baixos, no dígito 0. Os dados fornecidos pela caneta óptica podem, desse modo, alimentar o computador diretamente.





Jogos e brincadeiras

Castelos e dragões, mercado financeiro e lutas espaciais, divertimento e educação — tudo está nos jogos de computador.

O computador pode ser muito divertido: o mundo dos jogos eletrônicos é um caleidoscópio fascinante de brincadeiras, enigmas e novos desafios.

Os populares fliperamas estão sendo substituídos por baratos e potentes microcomputadores, que proporcionam muita variedade e emoção até mesmo enquanto você está aprendendo a jogar. Esses jogos nos micros conseguiram atrair mais pessoas para a computação do que qualquer software de contabilidade. Você não deve pensar que, por isso, não está utilizando bem a máquina. Os jogos estão aí porque divertem.

Aventura

Nem todos os jogos precisam ter som e cor para que sejam apaixonantes. Toda uma nova gama deles tem

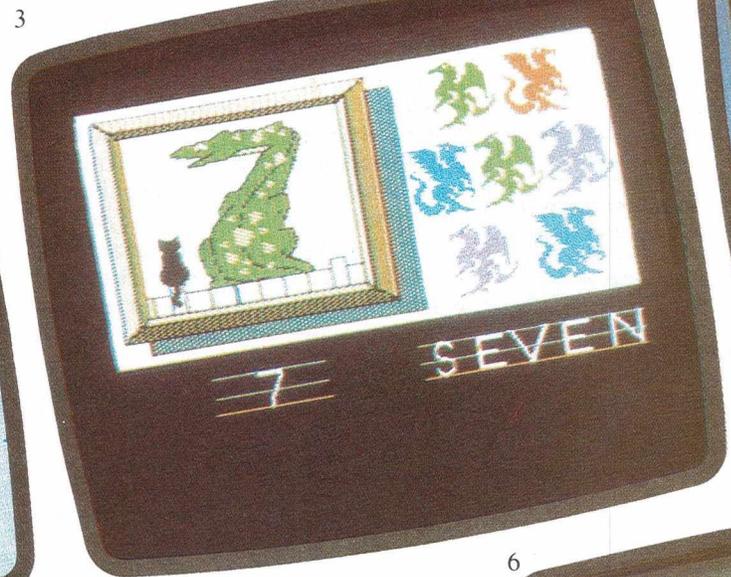
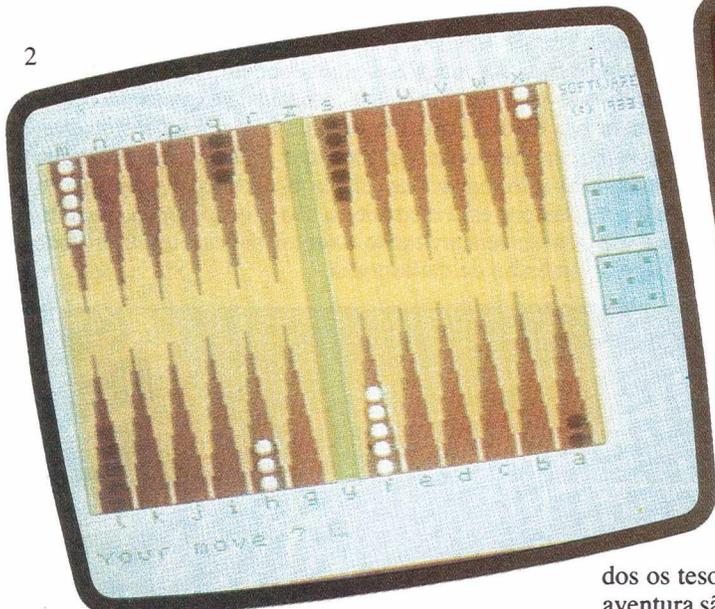
aparido com a difusão do micro; jogos que estimulam a imaginação com palavras, da mesma forma que livros e revistas sempre o fizeram.

Esses são os chamados “jogos de aventura”, e os primeiros programas foram feitos por programadores para brincar com seu grande computador no tempo livre de que dispunham. A idéia é que o programa cria um mundo que o jogador explora dirigindo seu *alter ego*; mas a direção é dada com palavras digitadas no teclado e não com joysticks.

O pequeno caractere da tela é movido quando se digitam instruções tais como “norte” e “acima”, e o computador mostra uma descrição do ambiente e de qualquer objeto próximo. O jogador pode “pegar”, “largar”, “virar” ou “quebrar” os objetos, ou pode experimentar alguma ação que lhe pareça interessante. O único limite é a imaginação.

Um programa de computador pode ser um labirinto de salas e porões, misturado com baús e soldados alemães, como no conhecido jogo Castelo Wolfenstein, ou pode ser uma nave espacial extraterrestre abandonada, ou mesmo uma casa de campo onde se procura solucionar um crime.

Seja qual for o cenário, o jogador tem de explorar, achar objetos valiosos ou tesouros e resolver enigmas intelectuais. A contagem total só aparece quando todos os problemas forem solucionados e to-



dos os tesouros encontrados. Os melhores jogos de aventura são como um bom romance e pode-se levar muito mais tempo para completá-los do que para ler um livro.

Mesa e tabuleiro

Muito naturalmente, os jogos preferidos de mesa e de tabuleiro foram transferidos para os microcomputadores, tão logo a tecnologia pôde fazê-lo. Não é necessário encontrar um oponente, pois o computador o substitui, e, se você errar uma jogada, pode corrigi-la, sem que o computador o acuse de trapacear. A máquina pode também aperfeiçoar o jogo do parceiro, apontando e corrigindo erros que você cometa.



O xadrez por computador alcançou um bom padrão e os gráficos do micro podem agora produzir traçados completos e detalhados do tabuleiro com um movimento suave da peça. Diferentes estratégias e aberturas foram exaustivamente analisadas e um programa de computador poderia facilmente ser campeão mundial. Os jogos de gamão, bridge e outros têm estado disponíveis em várias máquinas domésticas. Os computadores são fortes adversários no tabuleiro, na mesa ou na tela.

Jogando e aprendendo

Muitas empresas de programação se especializam em jogos educativos, com o computador ministrando testes e quebra-cabeças e aplaudindo quando a resposta está certa. Ou, às vezes, um jogo parece simplesmente um jogo mas contém informações

Estratégia grandiosa

Os jogos que requerem concentração e planejamento, as táticas e a tenacidade de um general comandando uma batalha também se transferiram para o microcomputador.

O jogador pode ser qualquer general, em qualquer guerra, deslocando exércitos e tentando descobrir os planos do adversário, o computador. Este se apresenta na sua melhor forma, atuando como árbitro e controlador do tabuleiro, à medida que decifra regras densamente escritas e pedaços de papelão que se perdem com facilidade — o que fez dos jogos de guerra de tabuleiro os preferidos apenas de uma minoria.

O jogador pode ser também o rei de um pequeno país, planejando um meio de administrar a colheita e o tesouro e, ao mesmo tempo, mantendo os traba-

1 **Magia:** Entre num mundo misterioso. Seus companheiros podem ajudar. Mas quem escolher? Um guerreiro, uma jovem ou um cientista?

2 **Gamão:** O seu adversário neste jogo é apoiado por uma força formidável — a lógica implacável do computador.

3 **ABC Dragon:** Um jogo educativo para crianças.

4 **Zaxxon:** Um dos primeiros jogos do tipo fliperama. A tela é o limite do piloto, à medida que ele avança dando voltas entre os mísseis e lutando para atingir seu objetivo.

5 **Frente Oriental:** Talvez você possa ter sucesso na Rússia, onde Hitler falhou.

6 **Simulação do voo de uma aeronave.** Você se esquivou ou sofre uma colisão.

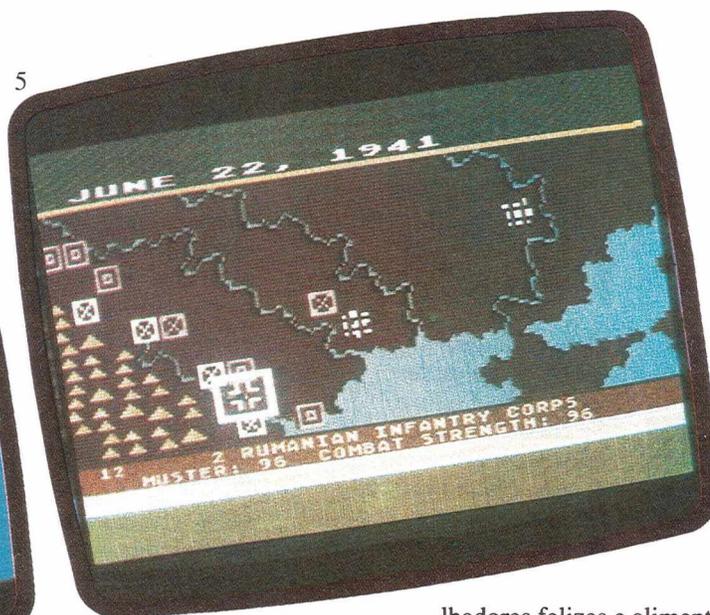
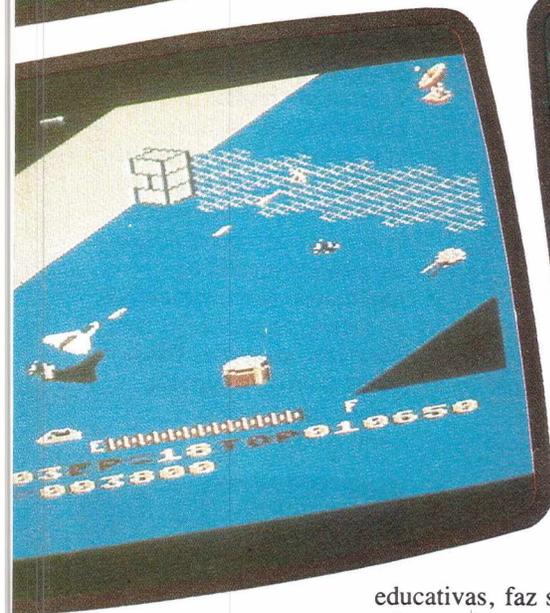
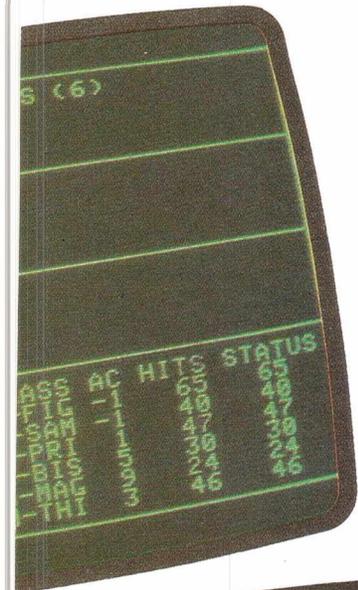
lhadores felizes e alimentados e os ladrões longe de seus domínios.

Da mesma forma, o jogador pode estar encarregado do fornecimento de energia de um país, avaliando os preços de minérios, do petróleo e da energia nuclear, contra seus perigos e efeitos a longo prazo. O computador pode ajudá-lo a avaliar e aperfeiçoar os seus planos de conquistar o mundo.

Voando alto

Os programas de jogos podem colocá-lo na cabina de um avião veloz, lendo os instrumentos e manejando os controles para decolar e aterrissar com perfeição, numa variedade de simulações de aeroportos verdadeiros; os jogos podem transformá-lo no piloto de uma missão espacial, que avista a Terra através de vigias; ou podem torná-lo um magnata como Rockefeller nas bolsas de valores internacionais, formando e destruindo empresas poderosas.

Precisão é tudo nos jogos de simulação. Siga as regras do mundo real e o jogo lhe mostrará o que realmente aconteceria. Mas, se cometer um erro, você não ficará preso nos destroços nem precisará saltar de uma janela de Wall Street. Os computadores são mais complacentes do que o mundo real!



educativas, faz somas, soletra e até impõe a lei da oferta e da procura.

Favorito entre os jogos é aquele em que as crianças dirigem, sobre uma folha de papel, um robô que segura uma caneta e é conhecido como “tartaruga”, devido à sua forma. Assim, as crianças se divertem desenhando, enquanto aprendem geometria.

Fliperamas

Os fliperamas, jogos de muita ação e movimento, já atraíram bilhões de fichas para suas máquinas. Nos microcomputadores, você pode divertir-se com os mais conhecidos: ataques de invasores, sapos que saltam, aventuras de Tarzan e gorilas gigantescos.

Mas as empresas de programação têm suas próprias idéias e criaram jogos que concorrem com os fliperamas em emoção e em gráficos espetaculares. Com um micro há maior possibilidade de escolha de jogos rápidos e emocionantes, todos à sua disposição, sem a presença voraz da máquina caça-níqueis. Esses jogos testam os microcomputadores — e os programadores — até o limite de sua possibilidade.



Perguntas e respostas

Aqui estão algumas dúvidas sobre o mundo da computação, que nem sempre são respondidas nos livros e revistas.

Em que um micro pode ser utilizado, além dos jogos?

Microcomputadores são usados para controlar pequenos negócios, manipular contas bancárias, processar palavras, auxiliar num projeto de decoração de interior ou armazenar informações sobre recordes esportivos. Estes são exemplos típicos do emprego de microcomputadores. Em resumo, sua utilidade pode ser assim classificada: lida com números e com palavras, armazena e apresenta informações de acordo com as necessidades do usuário.



Os computadores parecem estar ficando cada vez menores e mais baratos. Esse processo tende a ter um fim?

Os computadores ficam menores à medida que a tecnologia avança. Os componentes eletrônicos no seu interior estão ficando reduzidos, mas o teclado deve permanecer de tamanho suficiente para ser acionado por nossos dedos. Por isso, é improvável que os computadores fiquem muito menores do que já são, até que o teclado tradicional seja substituído por outros modos de comunicação com a

máquina. Com vários micros custando cerca de 200 dólares, é improvável que os preços caiam muito abaixo disso.



O BASIC é uma linguagem difícil de aprender?

Em si, não é difícil. Se comparado com o português, que, afinal de contas, é uma língua que todos aprendemos, o BASIC é uma linguagem com regras fixas. Isso a torna de mais fácil aprendizado que uma língua estrangeira. Embora os elementos do BASIC sejam simples, não é tão fácil redigir programas longos e complicados.

Quando é que devo começar a aprender o BASIC?

A melhor época para começar é quando você sentir necessidade. É possível que o computador faça exatamente aquilo que você quer, utilizando um programa comprado pronto. Neste caso, não é preciso aprender o BASIC. Infelizmente, os programas comprados nem sempre fazem exatamente o que você precisa. Ao aprender um pouco de BASIC, você pode adaptar os programas às suas necessidades. Assim,

para que o micro faça tudo o que você quer, de modo a se tornar um computador pessoal, é necessário que você aprenda o BASIC logo que puder.

Um monitor parece ser mais caro do que um televisor. Qual seria a vantagem em adquiri-lo?

Você tem uma imagem mais clara e nítida na tela. Se pretende passar algum tempo utilizando o computador, ficará a maior parte dele olhando para a tela; se ela for nítida e de fácil leitura, a possibilidade de dores de cabeça será menor. Além disso, as imagens gráficas terão melhor qualidade num monitor.

Qual o consumo de eletricidade de um microcomputador?

Menor do que o de uma lâmpada de 60 watts.



Os anúncios dizem que se pode utilizar o computador para ajudar a ensinar as crianças. Isso é verdade?

Sim. E também pode ajudar a ensinar adultos. Muitas escolas brasileiras já estão utilizando microcomputadores nas salas de aula, com excelentes resultados. Este é um

indício de que os microcomputadores devem desempenhar um papel cada vez mais importante na educação. O segredo para utilizá-los como um auxílio na aprendizagem está em possuir um software adequado. Há programas que buscam desenvolver o raciocínio matemático, outros que ensinam regras de ortografia. Contudo, este não é um modo muito estimulante de utilizar o computador. Uma utilização educacional mais criativa é feita através da linguagem LOGO. O LOGO permite que as crianças aprendam explorando seu micromundo e fazendo experiências para ver o que acontece. Desse modo a criança aprende enquanto programa o computador.



Alguns computadores (caso do TK85) têm teclados no estilo de uma calculadora; outros (caso do CP 500) têm teclados como uma máquina de escrever. Qual a diferença entre eles?

Muito pouca, a não ser que você seja um datilógrafo experiente; nesse caso, poderá digitar seus programas BASIC mais rapidamente num teclado em estilo máquina de escrever.

O professor eletrônico

Mesmo seus filhos pequenos podem usar facilmente o computador. Eis o melhor modo de ajudá-los nessa iniciação.

Um dos mais interessantes complementos pedagógicos que o computador oferece a crianças é a "tartaruga que gira". Funciona ligada a um microcomputador e comandada por um programa LOGO. A tartaruga permite à criança executar desenhos no chão e é ótimo auxiliar no ensino de conceitos matemáticos, como forma, distância e relação entre objetos. Além disso, é muito divertida.



Muitos pais se perguntam se um microcomputador seria útil a seus filhos. A maioria reconhece a vantagem de adolescentes aprenderem a usar o computador em casa e na escola; mas a questão que permanece é saber se crianças pequenas podem ser beneficiadas pela computação.

A resposta é afirmativa. Há diferentes modos, porém, de iniciar o contato de uma criança com o conceito de computação — e alguns são melhores do que outros.

Em países desenvolvidos já teve início o uso de computadores por crianças na escola primária pública. Na Inglaterra, por exemplo, o projeto *Micros in Primary Education* atingiu avultado custo, com o objetivo de dotar, em curto prazo, cada uma das 29.000 escolas primárias inglesas com pelo menos um microcomputador.

Os computadores não servem apenas para uso em

matemática. Mediante um bom programa, podem ajudar as crianças menores a aprender música, balé, geografia, línguas estrangeiras e, naturalmente, aritmética, geometria e física elementar.

▶ Há duas maneiras principais de o computador ajudar crianças menores: elas podem usá-lo para conhecer melhor o mundo que as envolve; ou o computador pode funcionar como professor, instruindo e exercitando a criança em uma série de matérias do currículo escolar.

Não é aconselhável tentar ensinar crianças com, por exemplo, seis anos de idade a programar um computador em linguagem BASIC. Algumas conseguem desenvolver programas em BASIC aos nove anos e mesmo antes, mas as pesquisas do psicólogo suíço Jean Piaget mostram que, antes dos doze ou treze anos, a maioria tem dificuldade para entender conceitos abstratos.

Para contornar essa dificuldade, os pesquisadores tornaram possível a crianças o controle e programação de computador, sem necessidade de conceitos abstratos (ver o quadro a respeito do LOGO). Normalmente, os professores iniciam o contato de crianças com o computador por meio da mistura dos dois métodos, alcançando, desse modo, ótimos resultados.

“Tartaruga que gira”

Mesmo crianças pequenas podem ser auxiliadas pelo computador em seu aprendizado. Na foto da página anterior, um garoto brinca com a “tartaruga”, um robô mecânico ligado a um microcomputador. As tartarugas são projetadas para uso escolar; porém, seu princípio de funcionamento é simples: a tartaruga tem duas rodas e uma caneta. A criança a instrui para que se mova para a frente, sobre uma cartolina, e se deve ou não desenhar uma linha — o “rastros da tartaruga” — conforme se move.

Por esse procedimento, a criança executa desenhos ao instruir o robô sobre o modo de formar ângulos e unir linhas. Sendo incentivada a planejar os movimentos que a tartaruga deve executar para desenhar determinada forma, acaba por descobrir sozinha os elementos constituintes da geometria básica. Essa iniciativa individual é o cerne do método LOGO. É fundada na idéia de que lições aprendidas “heurísticamente” (por tentativa e erro) são mais bem assimiladas do que pela apresentação de exemplos.

Essas duas escolas de pensamento encontram-se na base dos dois métodos de utilização de computador por crianças pequenas. Através do LOGO, crianças maiores — de nove ou dez anos — passam a utilizar a tartaruga na tela do computador, desenhando figuras complexas e “ensinando” o robô a memorizar vários procedimentos. Ao ensinar a tartaruga a executar movimentos, numa cartolina ou na tela, a criança está, de fato, programando o computador. O LOGO é uma linguagem que permite à criança fazer programas antes de desenvolver a compreensão abstrata, necessária para a maioria das linguagens de computador. “Brincar com a tartaruga” possibilita à criança se acostumar ao controle do computador, ajudando-a a conhecer o meio em que vive.

A outra abordagem usa o que poderíamos chamar de “paciência” do computador para ensinar por meio de exemplos.

Crianças com dificuldade na compreensão de determinada matéria ou conceito são auxiliadas por programas de “exercício e prática” que fazem perguntas e depois fornecem os resultados, indicando o nível de acerto. Vários desses programas são visualmente muito atrativos, com gráficos em cores vivas, efeitos sonoros e acordes interessantes. Esses programas incentivam o aprendizado; além do que o computador jamais se cansa ou desiste se a criança fornece seguidamente uma resposta errada. Essa “paciência” mostra-se valiosa no ensino de crianças de raciocínio lento. Instrumentos educacionais muito eficazes são também os programas de exercício e prática que, por exemplo, pedem à criança que escolha um substantivo em um grupo de palavras, ou forme uma palavra a partir de um conjunto de letras.

Apesar de todas essas utilizações, a figura do professor continua sendo essencial. Contato humano é o elemento mais importante do aprendizado e, embora o computador seja o complemento pedagógico mais rico em possibilidades, não pode substituir a atenção individual dada pelo professor.

Se você pensa em comprar um computador para uso de seus filhos, é aconselhável verificar que modelo é ou vai ser utilizado na escola. A compra de um modelo semelhante possibilitará o uso dos mesmos programas em casa e estabelecerá um vínculo entre as atividades de computação na escola e em casa.

Os computadores também divertem e os jogos são numerosos. Muitos pais se preocupam com a possibilidade de jogos como Space Invaders e Pac-Man apresentarem efeitos nocivos; mas não há evidência alguma de que o interesse despertado por esses jogos seja prejudicial.

Crianças com menos de sete anos necessitam de auxílio e supervisão para ligar o computador e o televisor, bem como para carregar programas. Se o programa for bem feito, poderão usá-lo sozinhas, em-

A lógica LOGO

Vemos aqui como se constroem figuras na tela usando a linguagem LOGO.

Logo é a linguagem de computador desenvolvida especialmente para que crianças menores, até mesmo com quatro ou cinco anos, programem um computador. Foi desenvolvida no Instituto de Tecnologia de Massachusetts, no fim da década de 60, por um grupo liderado por Seymour Papert, matemático que trabalhara com o educador Jean Piaget no seu Centro de Genebra.

Para crianças menores, o LOGO é uma “tartaruga” sob a forma de robô mecânico ou representada na tela do computador por um triângulo luminoso. O comando FORWARD 10 faz a tartaruga avançar dez unidades para a frente, desenhando uma linha atrás de si. O comando RIGHT 90 a faz virar à direita em ângulo reto. Pode-se elaborar seqüências de comando que fazem a tartaruga desenhar quadrados, triângulos, círculos e formas não previstas. Pode-se ensinar a tartaruga a “lembrar-se” dos comandos: sem percebê-lo, a criança está, de fato, programando o computador.

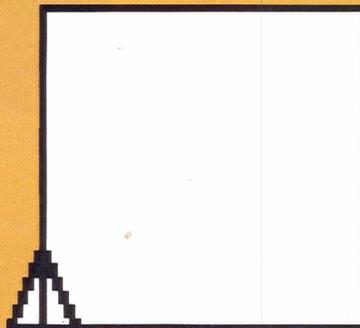
No Brasil encontra-se disponível uma versão do LOGO em português (MLOGO), cujos comandos são, por exemplo, DIREITA, FRENTE, REPITA etc.

Um modo de desenhar um quadrado:

```
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
```

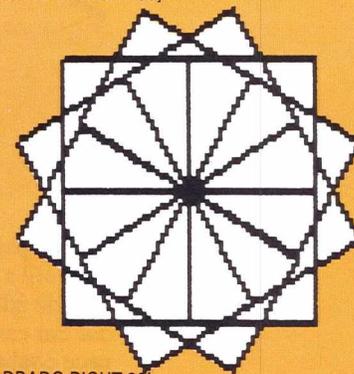
Criando um comando para “QUADRADO”:

```
TO QUADRADO
REPEAT 4 [FORWARD 50 RIGHT 90]
END
```

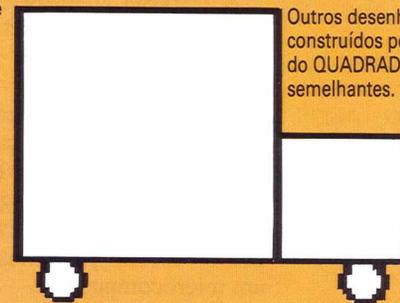


Comando “ESTRELA”:

```
TO ESTRELA
REPEAT 12 [QUADRADO RIGHT 30]
END
```



Outros desenhos podem ser construídos pela combinação do QUADRADO com comandos semelhantes.





Tensão ocular

Alguns adultos que precisam olhar fixamente a tela de computador ao trabalhar sofrem de tensão e fadiga ocular, mas as crianças não costumam concentrar-se por períodos longos o suficiente para que isso ocorra. No caso de adultos, o problema parece ser a "fixação acomodativa", fenômeno pelo qual o olho se fixa a uma distância focal, para cujo reajuste é necessário um certo tempo. Se achar que seu filho pode ficar tão entusiasmado por computação que tenderá a fixar o olhar na tela por longos períodos, isso poderá ser evitado assegurando-se de que faça um intervalo a cada quinze minutos.

Um problema técnico quanto ao uso de velhos aparelhos de televisão deve ser mencionado. Sabe-se que alguns televisores em cor fabricados antes de 1970 emitem uma pequena carga de radiação. Por isso convém assegurar-se de que seu aparelho tenha sido fabricado após este ano.

bora isso dependa também da capacidade de leitura e fornecimento de respostas às questões do programa. As exigências do equipamento (hardware) são muito simples. O microcomputador deve ser forte: as crianças às vezes batem no teclado com o punho cerrado, puxam as ligações elétricas e com frequência tocam e batem na tela. Se o equipamento for frágil, com ligações malfeitas ou de difícil controle, deixará de interessar à criança pequena.

Alguns especialistas são da opinião de que o teclado de um micro para crianças pequenas deve ter teclas grandes e bem distintas. Mas, à medida que a habilidade motora atinge desenvolvimento pleno (geralmente, pelos sete anos), as crianças tornam-se capazes de lidar com teclados bem menores, o que pode parecer difícil mesmo para adultos. Os teclados que funcionam por sensibilidade ao toque (tipo "touch"), encontrados nos modelos mais baratos como o TK83, não são realmente adequados para crianças abaixo de nove ou dez anos.

Jovens programadores

A escolha do software é difícil quando se trata de crianças pequenas. Se você pretende usar um equipamento que funcione com cassete, precisará supervisionar todo o processo de carregamento e armazenamento. Contudo, se seu equipamento opera por meio de disquetes, você verá que seus filhos menores conseguem manipulá-los muito bem. Uma das melhores formas de armazenamento de programas para crianças pequenas, abaixo de sete anos, é o cartucho ROM, uma embalagem em material plástico que contém um chip com um programa eletronicamente incorporado. A desvantagem desse equipamento é não permitir ao usuário a armazenagem de seus trabalhos. Mas os cartuchos são praticamente indestrutíveis e permitem a crianças pequenas usar o computador sem que você precise preocupar-se com eventuais danos ao equipamento.

Ao comprar um micro especialmente para seus filhos, procure instalá-lo em um local permanente. A mudança do computador de uma sala para outra,

com o conseqüente ligar e desligar de terminais, não o prejudicará (a menos que sofra uma queda), mas poderá fazer com que a criança mude para algum outro entretenimento, como a televisão.

O local de trabalho ideal

Em termos ideais, o computador da criança deveria ser colocado no quarto dela, complementado por seu próprio televisor. Se você quer que seus filhos desenvolvam uma atitude positiva em relação aos computadores, instale um local de trabalho em um de seus quartos, junto com um televisor de segunda mão, para uso exclusivo (coloque a parte central do computador no quarto da criança mais velha: ela poderá querer usá-lo após as demais terem ido dormir). Um velho televisor em preto e branco pode ser adquirido a baixo preço e, desde que sintonize canais, está em condições de apresentar os dados do computador.

Há muita discussão acerca da validade da cor em computação para crianças; alguns especialistas afirmam que ela é um elemento vital, outros a consideram um atrativo adicional, porém desnecessário. Todavia, parece evidente que se a escolha tiver de ser feita entre a ligação permanente a um televisor em branco e preto no quarto da criança e a ligação temporária no televisor colorido da família, a permanente será preferível pois incentivará o uso do computador a qualquer momento.

Se você montar um centro permanente ou semi-permanente de computação para jogo/trabalho no quarto de uma das crianças, é aconselhável dispor o equipamento de modo que possibilite a mudança do micro sem alterar muito a disposição. Ao instalar a mesa do computador, prenda todos os terminais e ligações, de tal modo que as crianças não desliguem algum terminal por acidente (todas as ligações de saída devem estar bem protegidas e presas de forma que não se soltem). É fundamental que o microcomputador fique estável e não balance. Alguns fabricantes fornecem um suporte que mantém firme o aparelho quando este é muito leve, e, se seus filhos forem irrequietos, você deve pensar na possibilidade de construir um reforço ou algum outro meio de fixá-lo com segurança.

Naturalmente, o computador comprado para seus filhos será útil para toda a família; assim, se estiver sendo usado por várias pessoas, é útil comprar-lhe também terminais adicionais e (se necessário) um segundo suprimento de saídas. Este é comparativamente barato e lhe permitirá colocar seus filhos para dormir, desligar o computador, o gravador cassete ou a unidade de discos do quarto deles (deixando todos os terminais presos em seus lugares) e ligá-los em seu próprio televisor através dos terminais adicionais. Se o micro for montado na sala para ser usado aí pelas crianças, a confusão com as ligações de terminais e as possíveis desavenças com pessoas da família que preferem ver um programa na TV destruirão a idéia de um computador como atividade de lazer para seus filhos, antes mesmo de iniciar-se o processo. Assim, precauções bastante simples como as aqui sugeridas são fundamentais para tornar o trabalho com o computador agradável.

O big trak

Assemelha-se a um tanque de brinquedo, mas é um ótimo instrumento de aprendizado, programável, que permite à criança planejar exatamente os movimentos que deseja que o tanque execute. O aparelho tem memória para dezesseis procedimentos e pode ser programado para andar pela casa, antes de voltar à base. A criança diverte-se mas, ao mesmo tempo, o computador ajuda-a a explorar seu ambiente e a elaborar os procedimentos necessários para a execução de um programa elementar.



O futuro chegou

A revolução trazida pelos computadores está mudando rapidamente hábitos e características da sociedade. É o mundo do amanhã, acontecendo hoje.

A revolução do computador coincidiu com a corrida espacial e com a viagem do primeiro homem à Lua. Os bilhões de dólares empregados nesse esforço resultaram na concentração dos maiores cérebros científicos e em grandes inovações tecnológicas — tudo para realizar um trabalho quase impossível. Os subprodutos têm variado, desde novos materiais de cerâmica, plásticos e adesivos até a microminiaturização do potencial incrível da computação.

Quando o “computador” significava milhares de componentes eletrônicos conectados um a um, os custos de fabricação eram enormes. Agora, esses mesmos circuitos podem ser reunidos em chips (minúsculas pastilhas) de silício tão pequenos que cabem num estojo do tamanho de um teclado.

Os chips de um TK85 ou CP 500, além de pequenos, também podem ser produzidos em massa, a um custo praticamente irrisório. O silício, matéria de que os chips são feitos, é uma das substâncias mais comuns do mundo. Todo grão de areia compõe-se quase que só desse elemento químico.

Agora que o computador é usado em casas e escritórios de todo o mundo desenvolvido, temos a oportunidade de testemunhar, em primeira mão, o início da segunda Revolução Industrial. A primeira substituiu o penoso trabalho manual pelas máquinas a motor. A revolução do computador poupará o tempo dos trabalhadores especializados e substituirá os operários por robôs controlados por computadores.

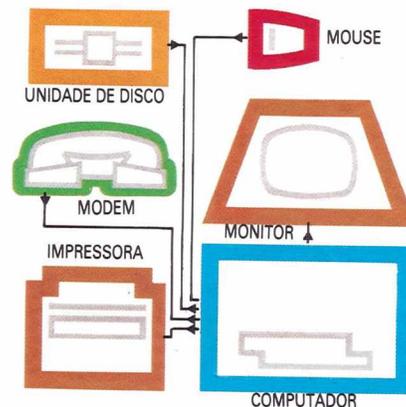
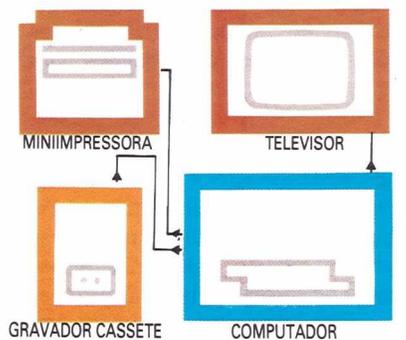
A longo prazo, o impacto que essa revolução exercerá em nossa vida ainda é de difícil avaliação. O certo é que padrões de trabalho e de lazer mudarão rapidamente. Os robôs, que nada mais são do que extensões mecânicas de computadores, vêm substituindo trabalhadores. Essas pessoas estão diante de duas possibilidades: ou aprendem novas técnicas ou ficam sem emprego.

Ofícios tradicionais, como a impressão e a composição, tornam-se igualmente vulneráveis. Até mesmo professores podem vir a ser substituídos com a utilização da tecnologia existente. E uma visita ao médico da família logo poderá significar uma entrevista com um terminal de computador, que identificará os sintomas do paciente e, a partir deles, terá condições de elaborar um diagnóstico.

A conseqüência social da primeira Revolução Industrial foi o deslocamento de milhões de pessoas do campo e a criação da poluição industrial. Estamos iniciando uma nova revolução industrial igualmente dramática, em que o computador será a força motriz da sociedade. Uma transformação em que o conhecimento e a utilização dos computadores poderão ser a chave da sobrevivência.

Em casa

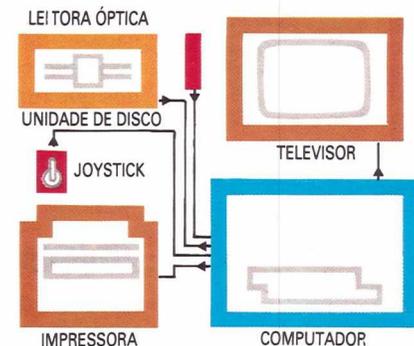
Um sistema doméstico barato como este pode ser o seu passaporte para o futuro.



Negócios

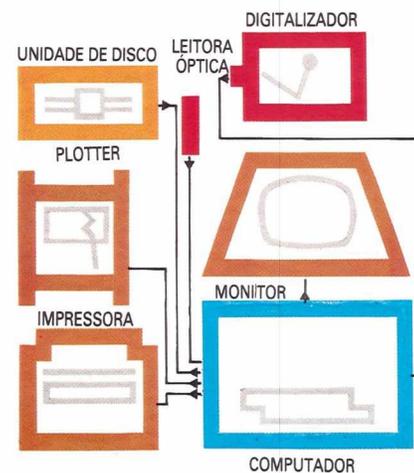
Pequenos sistemas de microcomputador estão ajudando — e substituindo — empregados de escritório e homens de negócios em todo o mundo.

Os modems dão acesso, através de aparelhos de telefone comuns, a computadores centrais e a seus vastos bancos de dados. Máquinas de escrever tornam-se coisa do passado, à medida que o processamento de palavras assume trabalho equivalente. Os softwares de contabilidade e as folhas eletrônicas diminuem custos e permitem a tomada de decisões financeiras em poucos minutos. Unidades de disco substituem salas cheias de arquivos e as impressoras fornecem desde cartas perfeitamente impressas até tabelas instantâneas das cotações diárias do mercado de ações. Os executivos que não sabem datilografar poderão utilizar o mouse, que fornece comandos ao computador sem a utilização do teclado.



Jogos

Alguns complementos abrem para você o mundo dos jogos. Além de divertir, os jogos ensinam princípios essenciais de computação.

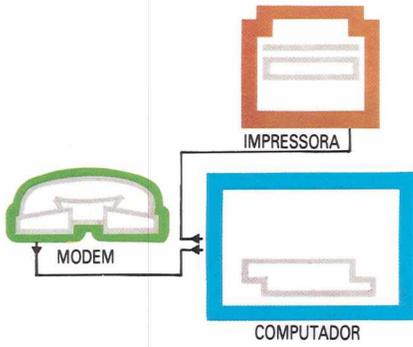


Pesquisa

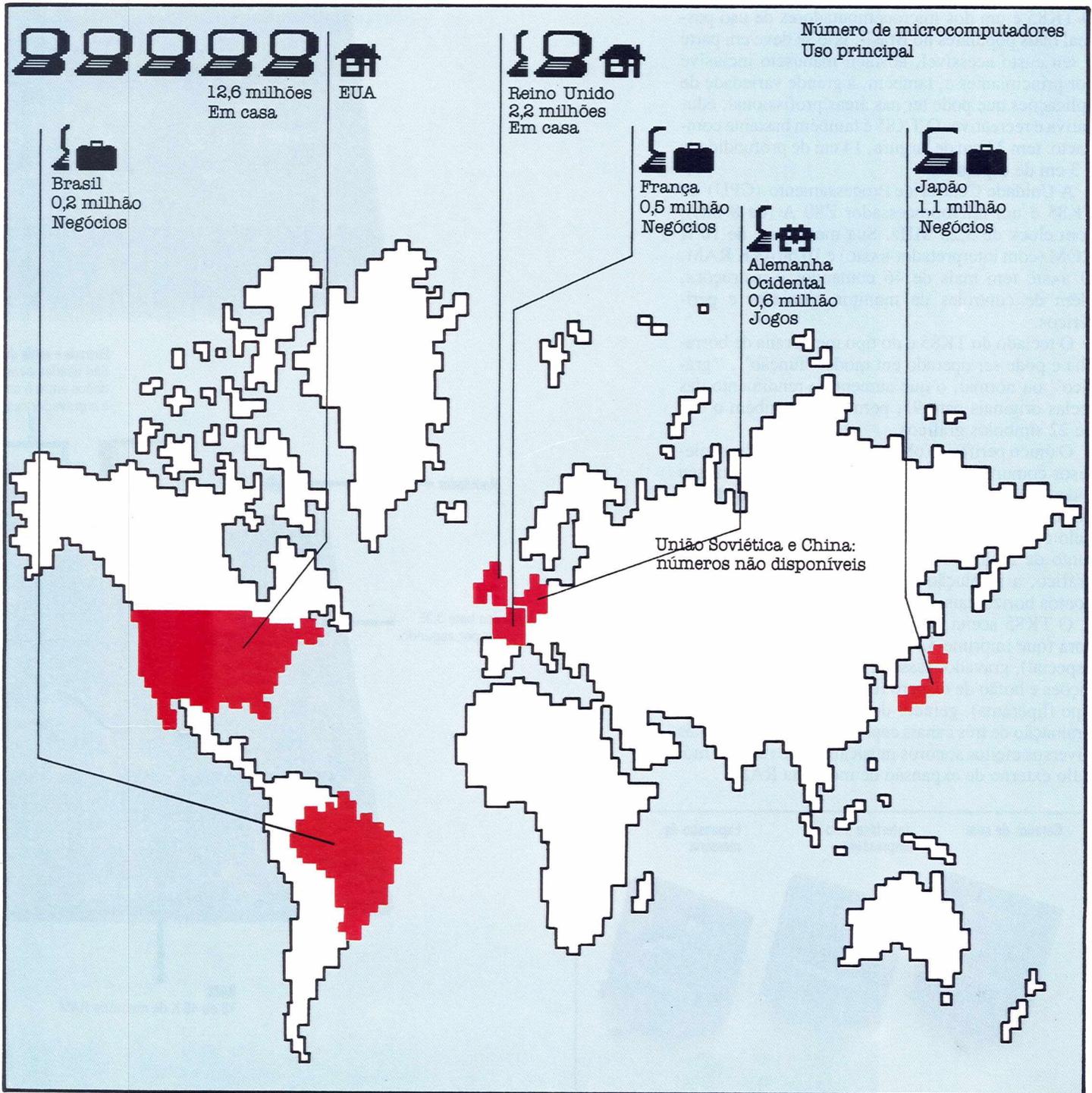
A computação avançada já não é mais privilégio dos laboratórios de universidades ou das grandes corporações. Sistemas sofisticados instalam-se em escolas e lares do mundo inteiro. Já estão disponíveis plotters que produzem desenhos técnicos e digitalizadores que introduzem figuras complexas, tais como mapas ou diagramas. Atualmente, estudos de computação fazem parte do currículo escolar, assim como o latim fazia há alguns anos. As crianças de hoje já estão aprendendo a conhecer o mundo computadorizado de amanhã.



Comunicação



A rede telefônica é difundida em todo o mundo e muitos computadores já podem ser facilmente transportados. O executivo atarefado pode digitar relatórios num avião a caminho de Nova Iorque e enviá-los para seu escritório com a velocidade do elétron, assim que chegar a um aeroporto. O modem acopla microcomputadores ao telefone e dá acesso instantâneo a praticamente todos os outros computadores do mundo, incluindo gigantescas centrais de computadores industriais. Os bancos de dados estão disponíveis 24 horas por dia e até mesmo o jornalista, fazendo uma reportagem na Ásia ou na América Central, pode enviá-la assim que encontrar um telefone.





TK85

De fácil operação, o TK85 é um equipamento que possibilita um grande número de aplicações.

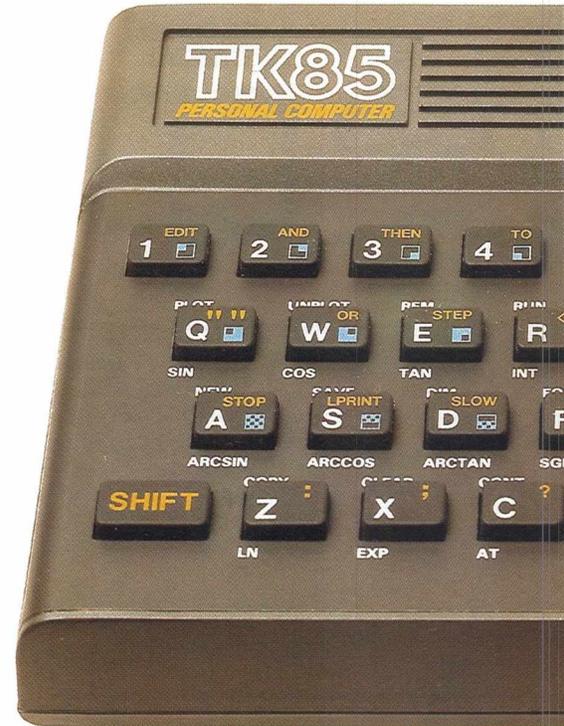
O TK85 é um dos microcomputadores de uso pessoal mais populares no Brasil. Isso se deve em parte a seu custo acessível, ao fácil manuseio inclusive por principiantes e, também, à grande variedade de aplicações que pode ter nas áreas profissional, educativa e recreativa. O TK85 é também bastante compacto: tem 23 cm de largura, 14 cm de profundidade e 3 cm de espessura.

A Unidade Central de Processamento (CPU) do TK85 é um microprocessador Z80 A (de 8 bits), com clock de 3,25 MHz. Sua memória é de 10 K ROM (com interpretador BASIC) e 16 ou 48 K RAM. O BASIC tem mais de 46 comandos e instruções, além de controles de monitor, gravador e periféricos.

O teclado do TK85 é do tipo membrana de borracha e pode ser operado em modo "função", "gráfico" ou normal, o que aumenta o rendimento das teclas originais para 91, permitindo também o uso de 22 símbolos gráficos.

O único periférico obrigatório ao TK85 é um televisor comum, em cores ou preto e branco: em seu vídeo serão visualizados os dados digitados no teclado, bem como os resultados do processamento feito pelo computador. A tela é organizada por um conjunto de 24 linhas de 32 caracteres e, para display gráfico, a resolução é de 44 pontos verticais e 64 pontos horizontais.

O TK85 aceita vários periféricos, como impressora (que imprime 32 colunas em papel metalizado especial), gravador cassete, joystick com quatro posições e botão de disparo (especialmente para jogos tipo fliperama), gerador de som (permitindo a programação de três canais capazes de produzir os mais diversos efeitos sonoros musicais ou de ruído) e módulo externo de expansão de memória RAM.



Entrada e saída de gravação
São usadas para transferir dados entre o computador e o gravador cassete.

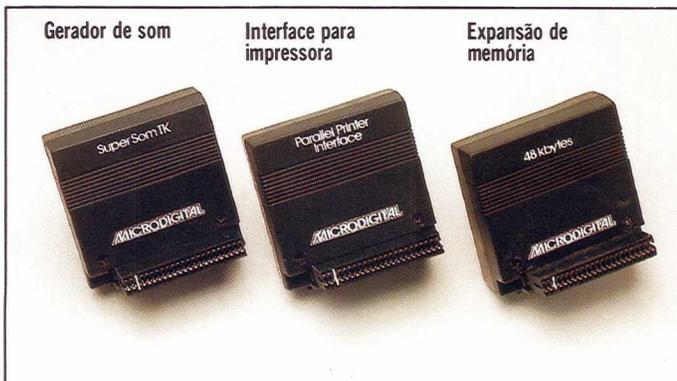
Modulador

Clock

O clock eletrônico bate 3,25 milhões de vezes por segundo.

RAM

16 ou 48 K de memória RAM





TK85

MICROPROCESSADOR

Z80 A

CLOCK

3,25 MHz

MEMÓRIA

Possui 16 K ou 48 K de memória RAM e 10 K de memória ROM que inclui o interpretador BASIC.

VÍDEO

Usa um televisor normal em preto e branco ou colorido, sendo feita a conexão aos terminais da antena. A tela é organizada em 24 linhas e 32 colunas, possui 22 símbolos gráficos e vídeo inverso.

TECLADO

Tipo membrana de borracha e pode ser usado em modo "função", "gráfico" e teclas simples.

LINGUAGENS

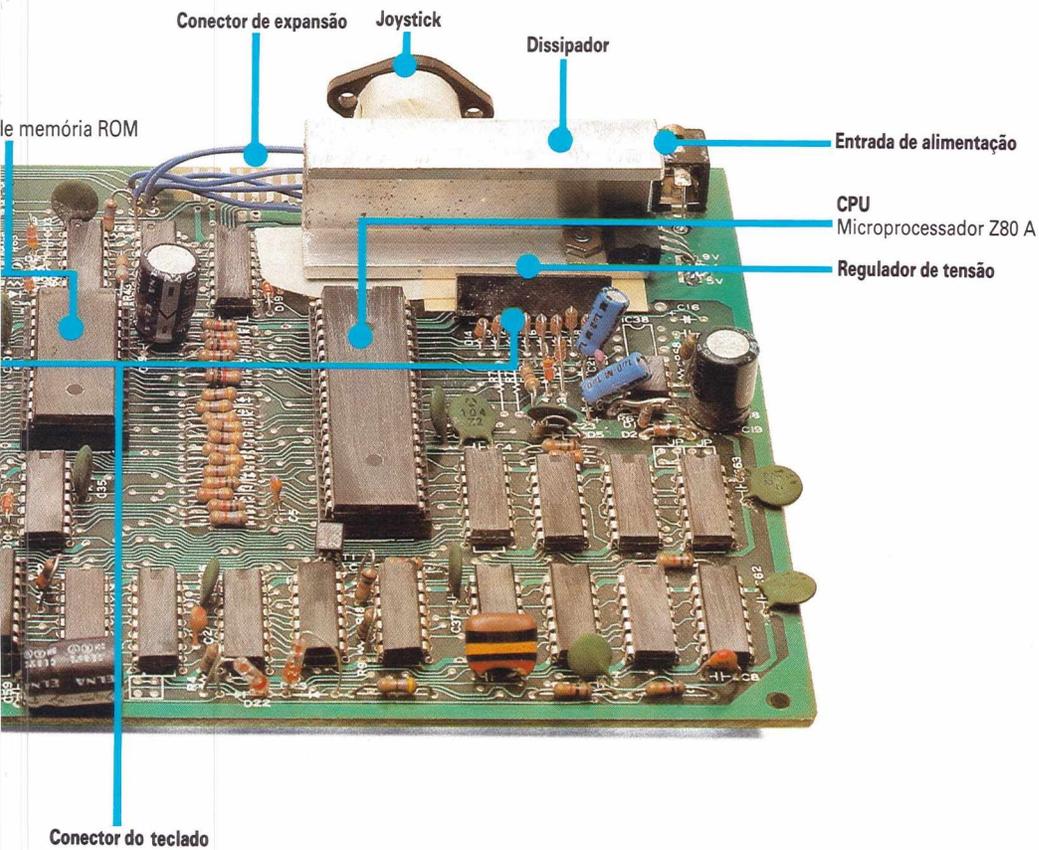
BASIC

PERIFÉRICOS

Cassete, impressora de 32 colunas (papel metalizado), joystick, gerador de som.

DOCUMENTAÇÃO

O manual que acompanha o TK85 é dirigido ao iniciante. Possui instruções detalhadas sobre o modo de operação do equipamento e inclui uma descrição pormenorizada da linguagem BASIC que ele utiliza. Para os que já conhecem o BASIC, o manual apresenta um apêndice onde são enumeradas resumidamente as características peculiares ao TK85.



Bits e bytes

O computador só entende de números, e tudo o que podemos fazer com dez dígitos ele faz com apenas dois — 0 e 1.

As palavras bits e bytes são sempre empregadas quando se escreve sobre computadores. Referem-se ao modo pelo qual os computadores armazenam e usam os números — um jeito absolutamente diferente daquele empregado pelas pessoas.

Representamos os números com dez símbolos diferentes (de 0 a 9) e os manipulamos em múltiplos de 10 (o que é chamado sistema de base 10). Já os computadores, para toda a sua prestidigitação matemática, precisam de apenas dois símbolos: 0 e 1. Para representar suas combinações é que se recorre aos bits e bytes. Um bit é a menor unidade de informação que um computador pode usar. É com um bit que o computador representa os dois dígitos — 0 e 1.

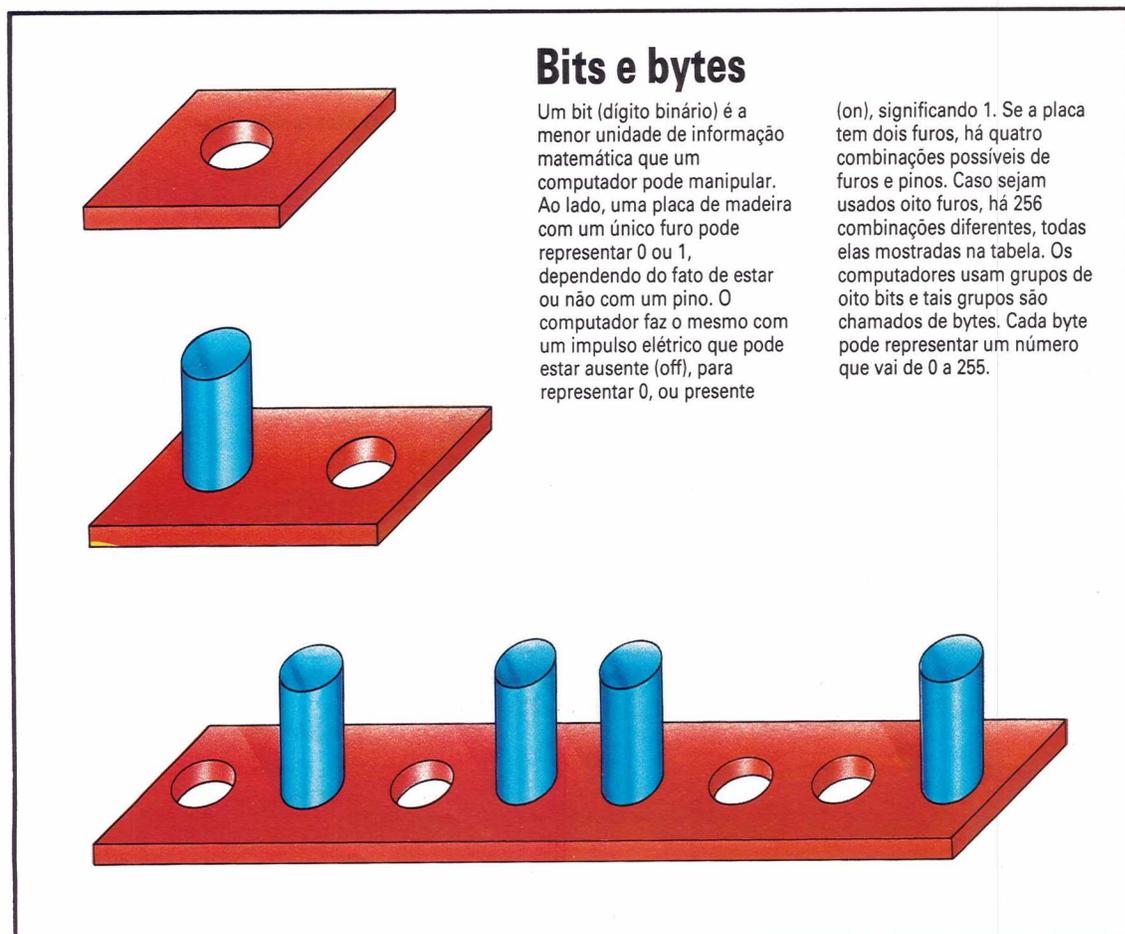
Em primeiro lugar, vamos ver o que são os bits e a razão por que são assim chamados. Computadores são instrumentos eletrônicos; tudo o que realizam é feito com sinais ou impulsos elétricos. Um único impulso pode estar “presente” (on) ou “ausente” (off): este é o princípio que permite representar números com impulsos.

A ilustração mostra uma placa de madeira com um furo em que se pode colocar um pino. Embora se trate de um único furo, pode representar dois números e serve de excelente analogia à maneira de funcionar do computador. Ou o furo está sem o pino e nesse caso representa 0, ou com o pino e representa 1. Uma única placa pode, então, simbolizar 0 ou 1.

Num computador, obtém-se o mesmo resultado com um impulso elétrico. Quando este está ausente (off), significa 0; caso contrário (on), representa 1. Um único dispositivo eletrônico, ou o pedaço de madeira com o furo, pode então ser usado para representar dois estados: sem pino ou com pino; ausente ou presente; off ou on; 0 ou 1.

A menor unidade de informação chama-se bit. A palavra, em inglês, já sugere algo de tamanho reduzido e representa dois estados possíveis. A palavra deriva de **BI**nary **digiT** (dígito binário). Pensado de outra forma, um bit é capaz de contar, mas apenas de 0 a 1.

Uma placa com dois furos pode indicar quatro di-





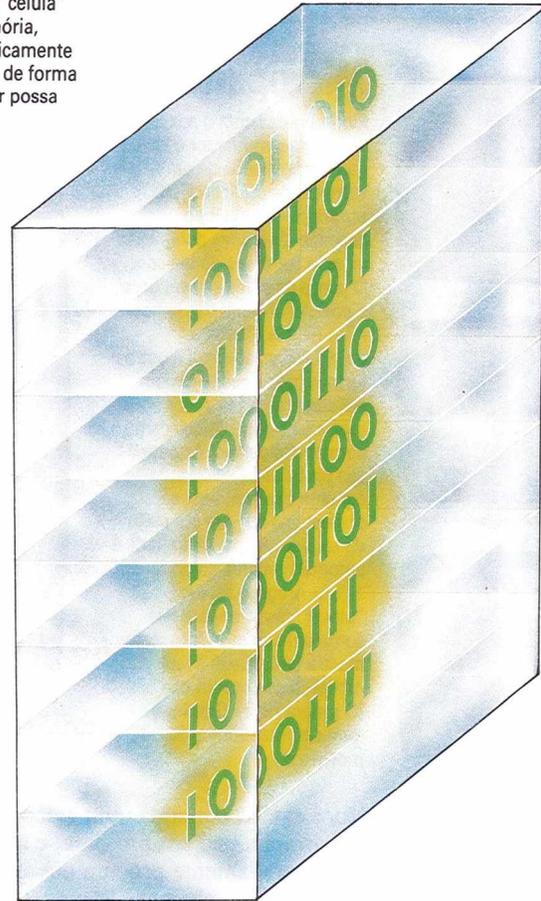
0	00000000	128	10000000
1	00000001	129	10000001
2	00000010	130	10000010
3	00000011	131	10000011
4	00000100	132	10000100
5	00000101	133	10000101
6	00000110	134	10000110
7	00000111	135	10000111
8	00001000	136	10001000
9	00001001	137	10001001
10	00001010	138	10001010
11	00001011	139	10001011
12	00001100	140	10001100
13	00001101	141	10001101
14	00001110	142	10001110
15	00001111	143	10001111
16	00010000	144	10010000
17	00010001	145	10010001
18	00010010	146	10010010
19	00010011	147	10010011
20	00010100	148	10010100
21	00010101	149	10010101
22	00010110	150	10010110
23	00010111	151	10010111
24	00011000	152	10011000
25	00011001	153	10011001
26	00011010	154	10011010
27	00011011	155	10011011
28	00011100	156	10011100
29	00011101	157	10011101
30	00011110	158	10011110
31	00011111	159	10011111
32	00100000	160	10100000
33	00100001	161	10100001
34	00100010	162	10100010
35	00100011	163	10100011
36	00100100	164	10100100
37	00100101	165	10100101
38	00100110	166	10100110
39	00100111	167	10100111
40	00101000	168	10101000
41	00101001	169	10101001
42	00101010	170	10101010
43	00101011	171	10101011
44	00101100	172	10101100
45	00101101	173	10101101
46	00101110	174	10101110
47	00101111	175	10101111
48	00110000	176	10110000
49	00110001	177	10110001
50	00110010	178	10110010
51	00110011	179	10110011
52	00110100	180	10110100
53	00110101	181	10110101
54	00110110	182	10110110
55	00110111	183	10110111
56	00111000	184	10111000
57	00111001	185	10111001
58	00111010	186	10111010
59	00111011	187	10111011
60	00111100	188	10111100
61	00111101	189	10111101
62	00111110	190	10111110
63	00111111	191	10111111
64	01000000	192	11000000
65	01000001	193	11000001
66	01000010	194	11000010
67	01000011	195	11000011
68	01000100	196	11000100
69	01000101	197	11000101
70	01000110	198	11000110
71	01000111	199	11000111
72	01001000	200	11001000
73	01001001	201	11001001
74	01001010	202	11001010
75	01001011	203	11001011
76	01001100	204	11001100
77	01001101	205	11001101
78	01001110	206	11001110
79	01001111	207	11001111
80	01010000	208	11010000
81	01010001	209	11010001
82	01010010	210	11010010
83	01010011	211	11010011
84	01010100	212	11010100
85	01010101	213	11010101
86	01010110	214	11010110
87	01010111	215	11010111
88	01011000	216	11011000
89	01011001	217	11011001
90	01011010	218	11011010
91	01011011	219	11011011
92	01011100	220	11011100
93	01011101	221	11011101
94	01011110	222	11011110
95	01011111	223	11011111
96	01100000	224	11100000
97	01100001	225	11100001
98	01100010	226	11100010
99	01100011	227	11100011
100	01100100	228	11100100
101	01100101	229	11100101
102	01100110	230	11100110
103	01100111	231	11100111
104	01101000	232	11101000
105	01101001	233	11101001
106	01101010	234	11101010
107	01101011	235	11101011
108	01101100	236	11101100
109	01101101	237	11101101
110	01101110	238	11101110
111	01101111	239	11101111
112	01110000	240	11110000
113	01110001	241	11110001
114	01110010	242	11110010
115	01110011	243	11110011
116	01110100	244	11110100
117	01110101	245	11110101
118	01110110	246	11110110
119	01110111	247	11110111
120	01111000	248	11111000
121	01111001	249	11111001
122	01111010	250	11111010
123	01111011	251	11111011
124	01111100	252	11111100
125	01111101	253	11111101

Bytes na memória

Bytes são grupos de oito dígitos binários (bits). Cada byte é usado pelo computador para armazenar números que podem ir de 0 a 255. Além disso, todo byte é armazenado numa espécie de “célula” separada da memória, disposta sistematicamente junto com outras, de forma que o computador possa

localizar o byte de que precisa. Para isso, ele tem de saber onde está o compartimento de memória que contém o byte.

- 1.ª posição de memória
- 2.ª posição de memória
- 3.ª posição de memória



ferentes estados, ou contar de 0 a 3. Os furos podem estar vazios; o furo da direita pode estar com um pino; o da esquerda pode estar com um pino; ou ambos estarem com pinos. A ilustração mostra uma placa com oito furos. Há, portanto, 256 combinações possíveis de pinos e furos — todas mostradas na tabela ao lado com números 1 representando pinos e zeros representando furos.

Um conjunto como esse, de oito dígitos binários (bits), é chamado de byte. Um único byte, portanto, pode representar 256 diferentes estados (ou pode contar de 0 a 255).

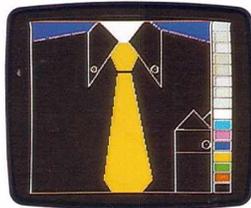
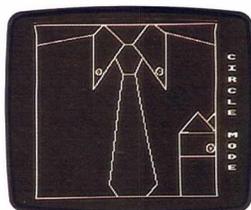
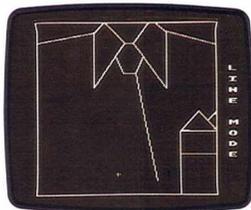
Quando se diz que um computador “armazena” um byte, significa que um número (de 0 a 255) está na memória do computador para ser utilizado no momento que for necessário. Cada byte tem seu próprio compartimento e tais compartimentos podem ser dispostos em seqüência (a ilustração acima mostra os bytes como se estivessem empilhados). Se o computador quer recuperar um número, tudo que precisa saber é em qual compartimento o byte (que contém esse número) está armazenado.

Todos os números de 0 a 255 podem ser representados mediante combinações de 1 e 0 (veja tabela). Os bits são armazenados e usados por computadores em grupos de 8. Oito bits juntos formam um byte.



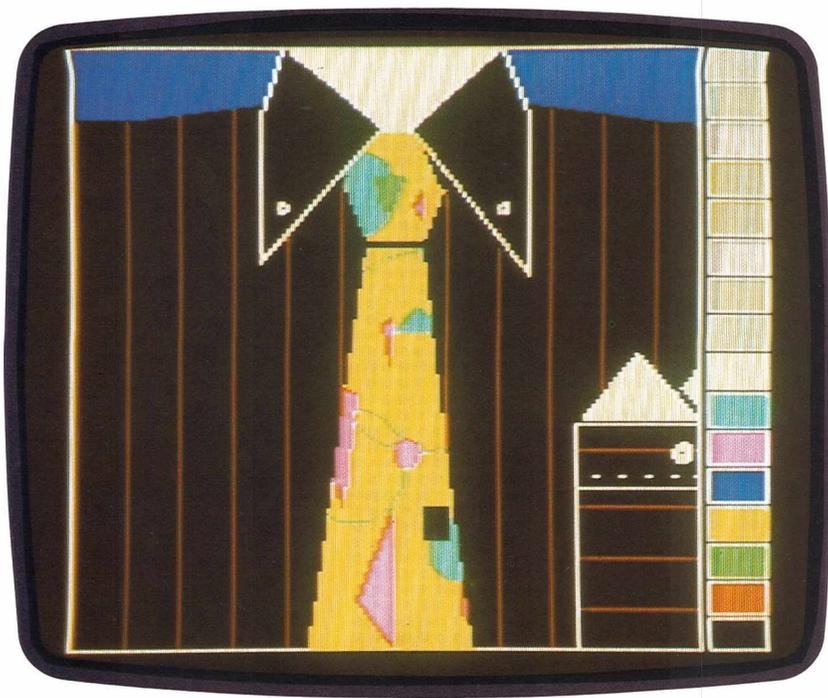
O micro: um artista

Simple ou sofisticadas, as figuras do vídeo são formadas por milhares de pontos. Memorizando a cor e a luminosidade de cada um deles, o computador é capaz de criar imagens.



Micropintura

Pintar figuras com o auxílio de um microcomputador é fácil, quando se dispõe de um software de gráficos. Esse programa possibilita o desenho de figuras complexas usando controles do teclado. O ponto de partida é, em geral, o posicionamento das linhas retas na tela. Círculos, triângulos, quadrados e outras formas predefinidas podem ser acrescentados. A cor é usada tanto para linhas como para partes da tela. As partes são pintadas selecionando-se a cor com base em uma escala disposta na tela, ou por meio do teclado. A área delimitada é automaticamente colorida. Em alguns programas, um "pincel" move-se pela tela como um cursor. A foto maior mostra o resultado desse processo de "construção". (O programa foi Graphkey, num computador inglês BBC modelo B.)



Você está lá em cima num avião, o motor em pane, um arranha-céu se aproxima, e a pista em que você desesperadamente procura descer está ocupada por um avião em chamas. Isso aparece à sua frente na tela do televisor. É apenas uma das formas de gráfico de computador.

Em programas educativos, como os destinados a alfabetizar ou ensinar aritmética, é evidente que, se não proporcionarem uma exposição visual interessante, a atenção das crianças não será mantida por muito tempo.

Outros usuários de microcomputadores estarão preocupados com números representando quantias recebidas ou gastas com o estoque de mercadorias e assim por diante. Mas esse tipo de informação é muito mais prontamente compreendido e interpretado quando mostrado pictoricamente. A habilidade do computador em "redesenhar" rapidamente uma figura, de forma a incorporar novas informações, propor opções e produzir cópias impressas quando necessário é também de imenso valor em muitas aplicações empresariais.

A tela do computador

Como um computador cria figuras? Para responder a essa questão, observe primeiramente a "tela" em que o computador trabalha. Um microcomputador produz imagens na sua tela de exposição, iluminando ou "acendendo" um ou vários pontos em de-

terminadas posições na tela. Esses pontos são dispostos em linhas no sentido horizontal e em colunas no sentido vertical, de modo que a posição de qualquer ponto é dada por sua localização na linha e na coluna.

Imagens específicas (figuras ou impressões) são produzidas pela iluminação de certos pontos, enquanto os outros permanecem apagados. Isso se aplica não apenas às telas monocromáticas, mas também às exposições coloridas. Nesse caso, para que apareçam as figuras, os pontos adquirem cores apropriadas.

Para expor uma única letra ou número, o computador usa uma disposição retangular de pontos. Essa disposição é conhecida como "matriz de ponto". Num microcomputador típico, deverá consistir em um bloco de oito linhas com oito pontos cada.

O número de pontos na tela não é igual em todos os microcomputadores, mas uma disposição que poderia ser considerada típica consistiria em 192 linhas com 280 pontos cada — em outras palavras, 192 linhas e 280 colunas.

É evidente que, quanto mais pontos existirem na tela do micro, mais detalhadas serão as imagens. O grau de detalhe obtido quando se expõem gráficos é conhecido como "resolução". Diz-se que um computador que pode expor 192 linhas com 280 pontos cada tem uma resolução de 280×192 . Quanto mais alta a resolução, isto é, quanto mais pontos forem



dispostos na tela, menor “granulação” terá a imagem.

Todos os micros têm uma densidade máxima de pontos que podem ser dispostos em suas telas de gráficos, mas alguns são também programados para usar disposições menos densas de pontos. Por exemplo, o micro inglês BBC tem uma resolução máxima de 640×256 , isto é, uma densidade máxima de pontos dada por 256 linhas, cada uma com 640 pontos. Entretanto, pode também ser programado usando-se apenas 320 dessas colunas, uma resolução de 320×256 , ou resoluções mais baixas, se necessário. Numa máquina capaz de proporcionar esse tipo de variação, a resolução deve ser definida no começo da parte do programa que produz os gráficos.

As linhas e curvas produzidas por um computador que usa um sistema de gráficos baseado em pontos não são propriamente contínuas, como seriam se fossem desenhadas com uma caneta. São caminhos de pontos iluminados, menos ou mais unidos, dependendo da resolução do sistema. Um sistema com resolução mais baixa poderá produzir apenas curvas desajeitadas e uma linha reta não será tampouco perfeitamente reta, uma vez que a posição dos pontos na tela está em linhas retas apenas em certas direções — como ao longo de linhas e colunas e sobre as diagonais da disposição de pontos.

Para traçar uma linha reta, o sistema de gráficos deve iluminar os pontos que estão mais próximos da trajetória da linha escolhida. O resultado pode ter um efeito de “degrau”. Novamente, quanto mais alta a resolução do sistema, menos perceptíveis serão tais “degraus”.

Para se ter continuidade na exposição dos gráficos na tela de televisor, a imagem deve ser continuamente “refeita” ou “redesenhada” — do contrário, seria visível apenas por um momento. Por essa ra-

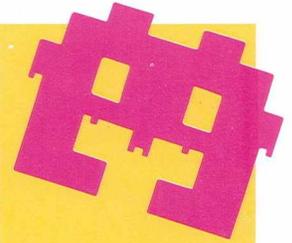
ção, a imagem deve ser representada de alguma forma no computador, para ser consultada quando necessário. A representação da imagem é armazenada numa parte específica da memória do computador, conhecida como memória de tela. Em uma exposição monocromática, cada ponto corresponde a um bit na memória de tela. Uma imagem é representada estabelecendo-se os bits correspondentes a pontos iluminados, 1, e os bits correspondentes a pontos apagados, 0. Assim, se um computador pode manter uma exposição monocromática com resolução de 280×192 , deve ter memória de tela de 280×192 ou 53.760 bits — ou 6,5 kilobytes, uma vez que 1 kilobyte é igual a 8.192 bits.

Com os gráficos em cor é necessário mais memória. Dois bits podem ser usados para representar quatro cores, por exemplo da seguinte forma:

bit 1	bit 2	cor
0	0	branco
0	1	vermelho
1	0	azul
1	1	preto

Para representar qualquer imagem de quatro cores é necessário designar dois bits para cada ponto na tela. Da mesma forma, para oito cores, três bits na memória são designados para cada ponto na tela, enquanto para exposições de dezesseis cores há quatro bits por ponto e assim por diante. Desse modo, um microcomputador que expõe cores com resolução de 160×256 deve ter uma memória de tela de $160 \times 256 \times 4$, o que é equivalente a 160 kilobits (20 kilobytes).

A necessidade de uma memória de tela explica em parte por que alguns computadores são planejados para trabalhar com diferentes resoluções. Sem uma memória de tela suficiente, o computador não pode armazenar e portanto não expõe figuras de resolução mais alta.

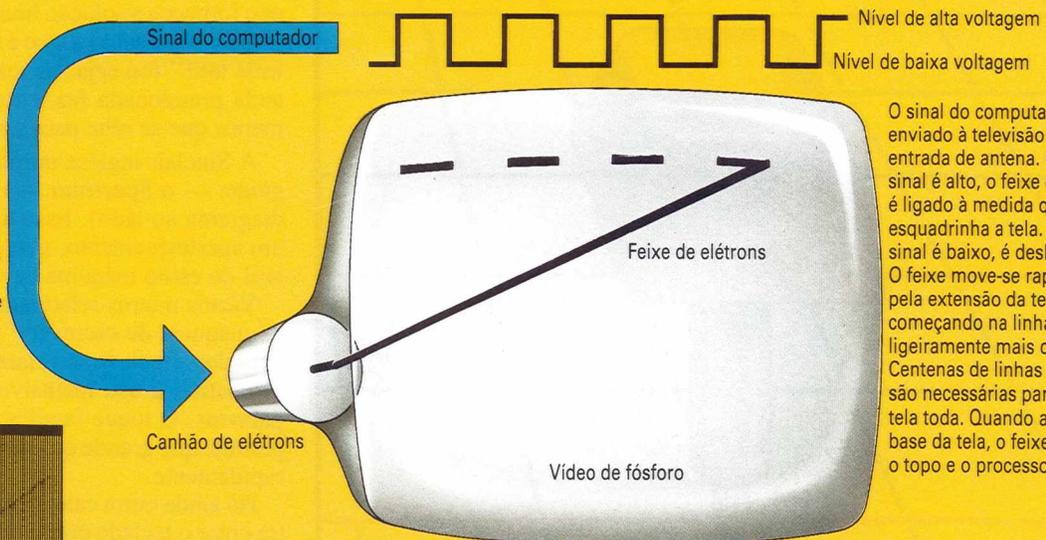
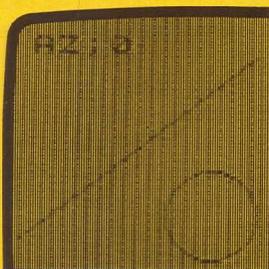


Criação de gráficos

Construir imagens no microcomputador é fácil. A maioria dos modelos tem controles especiais em BASIC, para facilitar o “desenho” ou definir objetos na tela, desde invasores alienígenas até figuras completas. Existem também programas especiais para criar figuras animadas na tela.

Construção da imagem

É desta forma que os sinais produzidos pelo computador são transformados em imagens na tela do televisor. A imagem é construída linha por linha e o sinal de entrada, que produz partes iluminadas em cada linha, é cronometrado de forma que todas as partes se ajustem para formar as figuras, linhas e caracteres (ver abaixo).



O sinal do computador é enviado à televisão através da entrada de antena. Quando o sinal é alto, o feixe de elétrons é ligado à medida que esquadrinha a tela. Quando o sinal é baixo, é desligado. O feixe move-se rapidamente pela extensão da tela e volta, começando na linha seguinte ligeiramente mais devagar. Centenas de linhas como essa são necessárias para cobrir a tela toda. Quando atinge a base da tela, o feixe volta para o topo e o processo se repete.

Fechando contato

Aparentemente, todos os teclados se assemelham. Na verdade, há uma série de diferenças no seu modo de funcionamento.

Por meio do teclado é que você se comunica com o computador. O teclado tem a mesma importância atribuída à capacidade de memória ou à qualidade dos gráficos.

Os microcomputadores herdaram o estilo de teclado Qwerty, da máquina de escrever, assim chamado porque as primeiras seis letras na série superior das teclas estão dispostas dessa maneira. Nos primeiros anos deste século, os caracteres foram posicionados nas teclas de modo a reduzir a velocidade das datilógrafas, para que não danificassem o delicado mecanismo!

No início dos anos 50, quando os computadores começaram a ser utilizados comercialmente, a disposição QWERTY era o sistema padrão das máquinas de escrever e tornou-se também o dispositivo padrão de entrada para os computadores. Esse tipo de teclado é sem dúvida familiar a datilógrafas experientes, mas às vezes difícil de ser dominado pelo principiante.

Quando os computadores custavam dezenas de milhares de dólares, o preço de um teclado mecânico era irrisório. Mas o desenvolvimento da tecnologia do microprocessador reduziu drasticamente o custo dos componentes eletrônicos do microcomputador.

Na época em que o micro inglês Sinclair ZX81 começou a ser desenvolvido, um teclado em estilo máquina de escrever podia representar proporção significativa no custo de fabricação de um microcomputador. O teclado móvel mecânico, de modelos como o CP 500 ou Micro Engenho, utiliza um tipo de interruptor sob os botões das teclas (ver ilustração na página seguinte). Quando a tecla é pressionada, os contatos internos se fecham para completar um circuito. Esses interruptores contêm inúmeros componentes e aumentam consideravelmente o custo do teclado.

Uma solução para o problema é um tipo novo e mais barato de teclado. Ao ser feito o teclado tipo "touch", do TK83, por exemplo, a idéia era que quase todos os compradores potenciais de microcomputadores estariam interessados principalmente em jogos e em escrever pequenos programas.

Como essas atividades envolvem uma quantidade mínima de trabalho com o teclado, parecia lógico que os primeiros usuários dos micros estivessem preparados para aceitar um teclado de menor qualidade. Se as vantagens de um teclado convencional em estilo máquina de escrever pudessem ser sacrificadas, seria possível uma substancial economia.

O ZX81 da Sinclair inglesa foi projetado com um teclado sensível ao toque, eliminando grande parte das peças. Isso fez baixar o preço do modelo, mas não forneceu a solução final. O inconveniente de um teclado sensível ao toque é não oferecer muito "controle tátil" (ou seja, nunca se tem certeza de que a tecla pressionada fez o registro no computador, a menos que se olhe para a tela).

A Sinclair inglesa introduziu em seu produto seguinte — o Spectrum — o teclado membrana (ver diagrama ao lado). Esse tipo de teclado representa um aperfeiçoamento, mas ainda não tem o controle tátil do estilo máquina de escrever.

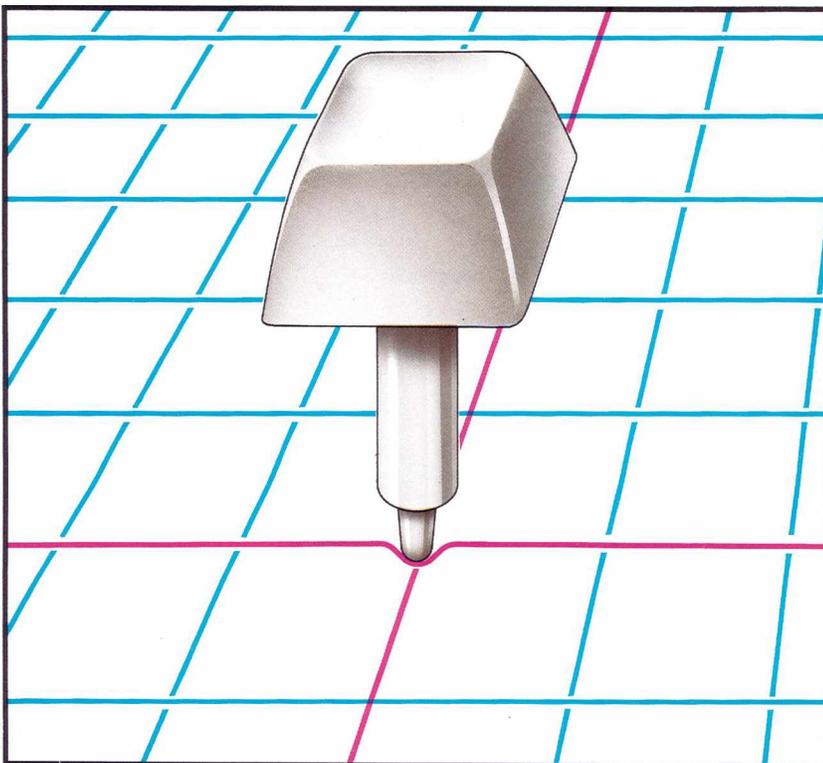
Vários micros relativamente baratos têm teclado de máquina de escrever "profissional". As vantagens desse teclado são claras quando o computador é submetido a uso intensivo para processamento de palavras. O toque familiar da máquina de escrever permite que grande quantidade de trabalho seja feita rapidamente.

Há ainda outra categoria de teclado que se encontra entre o teclado completamente móvel e o do tipo membrana do Spectrum inglês, comumente chamado "teclado tipo calculadora". As teclas oferecem melhor "toque", mas são pequenas, duras e menos adequadas à digitação do que as de estilo máquina de escrever, completamente móveis.

Um modo de resolver parcialmente a falta de controle tátil dos teclados "sensíveis ao toque" e

A matriz do teclado

As teclas de um computador são, na verdade, interruptores conectados a uma rede de fios. A ilustração mostra de que modo, ao se pressionar uma tecla, dois fios são conectados. Para cada tecla há um (e apenas um) par de fios envolvidos. Assim, cada tecla faz uma única conexão, permitindo ao computador perceber que tecla foi pressionada.



Teclado Sinclair

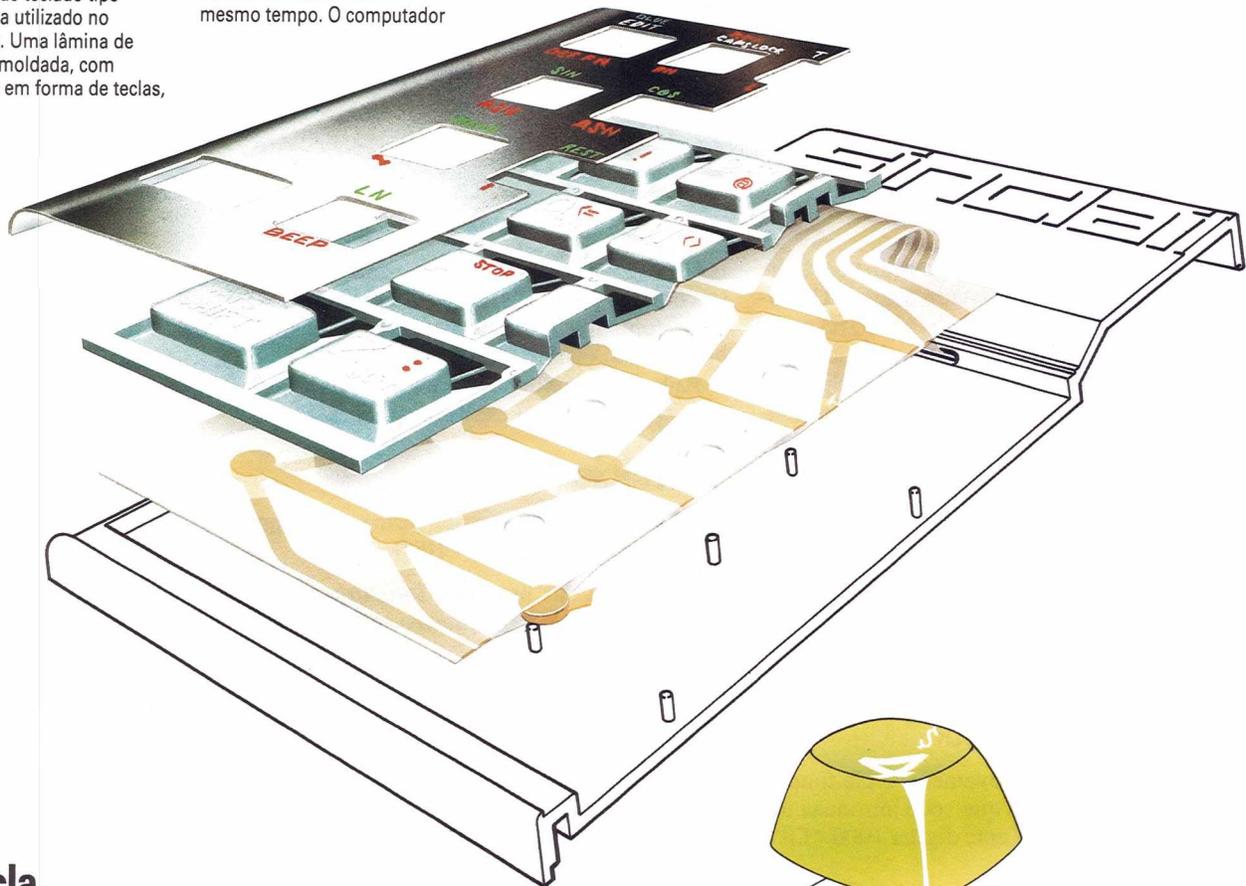
A Sinclair inglesa desenvolveu o projeto de teclado tipo membrana utilizado no Spectrum. Uma lâmina de borracha moldada, com saliências em forma de teclas,

é montada sobre um conjunto de contatos, formando a rede do teclado ou a matriz. Ao ser pressionada a tecla, a saliência sob o botão aciona os contatos ao mesmo tempo. O computador

verifica que par de contatos foi fechado e descobre a que tecla ele corresponde. Os contatos fechados pela tecla são normalmente separados por

uma bolha de ar presa entre duas lâminas plásticas. A força que devolve a tecla à sua posição inicial é fornecida pela elasticidade da borracha, que

estica quando a tecla é pressionada. Essa engenharia do teclado possibilita que o produto seja fabricado a custo mais baixo.

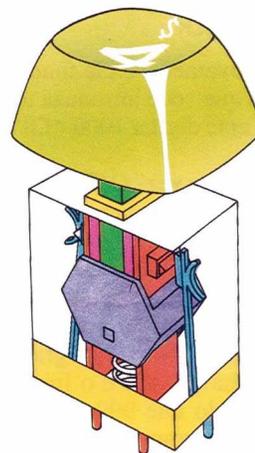


A tecla

As teclas em estilo máquina de escrever em geral incorporam um par de contatos, que são normalmente separados e impedem a passagem da eletricidade. Quando a tecla é pressionada, uma moldagem de plástico move-se para baixo e faz com que os contatos se unam e fechem um circuito.

Uma mola interna tem como função devolver a tecla à posição superior. O fechamento dos contatos permite o fluxo da corrente e o computador detecta isso. Os fios conectados aos contatos em cada interruptor estão dispostos numa rede. O computador "sabe" que tecla foi pressionada ao

verificar qual dos fios "verticais" da rede e qual dos "horizontais" conduzem a corrente. Teclas desse tipo, mecanicamente complexas, oferecem muita confiabilidade e têm um toque mais firme do que as teclas de membrana de borracha. Esse toque provém da resistência oferecida pela



mola. Uma tecla bem desenhada fornece controle tátil, de modo que o usuário percebe facilmente quando ela foi pressionada. A superfície das teclas também é desenhada para uma digitação mais cômoda. Esse tipo de tecla é aconselhável quando o computador vai ser utilizado com muita frequência.

"membrana" é produzir um bip audível ao ser pressionada a tecla. Isso assegura ao usuário que a tecla foi realmente pressionada e reconhecida pelo computador.

Os projetistas do Sinclair ZX81 e do Spectrum introduziram uma nova maneira, muito útil, de reduzir a quantidade de digitação exigida pelo registro de programas BASIC. Cada tecla é feita para representar mais de uma única letra do alfabeto ou número. Utilizando-se uma tecla de "função" especial em conjunção com as teclas comuns, pode-se fazer aparecer palavras BASIC inteiras na tela, sem necessidade de escrever a palavra toda. Por exemplo, a palavra PRINT do BASIC é produzida simplesmente pressionando-se a tecla de função especial e a tecla da letra P, ao mesmo tempo. No Brasil, este é o caso de micros como o TK83, TK85 e CP 200.

O fim da datilografia

Há não muito tempo, o único modo de fazer um computador funcionar era registrar uma instrução através do teclado. Por ser esta uma tarefa cansativa, que poderia estar fazendo as pessoas deixarem de utilizar computadores, os fabricantes



encontraram a solução brilhante e simples do "mouse", ainda não disponível no nosso mercado. O mouse é colocado em qualquer superfície plana e, quando se move, movimentam-se também o cursor na tela. Assim, uma pessoa pode fazer um movimento para qualquer parte da tela, pressionar o botão e dar andamento à operação desejada. O mouse é também utilizado em gráficos para traçar linhas ou "pintar" cores na tela.

Loops sob controle

Você pode interromper os loops e voltar a eles quantas vezes quiser. Nesta segunda parte do curso de programação você também aprende o melhor modo de numerar as linhas.

A primeira parte do curso de Programação BASIC terminou com o programa relacionado abaixo. O programa funcionou bem, mas, devido ao GOTO na linha 70, sempre voltava para o início. A única maneira de sair do loop era com o uso da tecla BREAK ou com o da tecla RESET. Conheça agora uma das maneiras de sair de um loop como este, incluindo um teste no programa.

Geralmente se testa com um número que, na verdade, nunca iríamos querer usar no programa. O programa permitia digitar um número que o computador então imprimia na tela somando-lhe 1. Poderíamos decidir que nunca seria usado um número superior a 999. Nesse caso, há possibilidade de testar para saber se o número que entrou é maior do que 999. Digite o programa e então acrescente:

```
35 IF A > 999 THEN GOTO 80 <CR>
```

Agora execute o programa novamente e ele funcionará como antes — a menos que você introduza um número maior do que 999. Tente digitar 1000 <CR> e veja o que acontece.

Por que o programa parou desta vez? O IF na linha 35 é o motivo. Quando o BASIC encontra um comando IF sabe que se segue a ele um teste de lógica. O sinal > significa “maior que”. A linha 35, portanto, significa IF (variável) A (é maior que) 999 THEN GOTO (linha) 80. Se você apenas digitar 1000, o valor de A será 1000, que é maior do que 999, e o programa THEN (então) GO (vai) TO linha 80, que é o fim do programa. Se A não é maior do que 999, a parte THEN da linha é ignorada e o programa continua na linha seguinte.

Ao executar este programa, você poderá introduzir quantos números quiser, desde que não sejam maiores do que 999. Logo que um número superior a 999 for introduzido, o comando IF-THEN detecta o fato e encerra o programa (GO TO END). Quando um programa BASIC atingir o fim ou for concluído, aparecerá um sinal na tela. Dependendo do seu computador, esse sinal pode ter diversas formas. No microcomputador Unitron AP II, é um sinal como este:]; no CP 500, é READY. Qualquer que seja a sua forma, este sinal é como o BASIC informa que nenhum programa está sendo executado, e aguarda novas instruções.

Há inúmeras variações na forma em que as diferentes versões do BASIC usam THEN. Mais detalhes você encontrará na seção “A propósito...”, na página 40.

Outras comparações usadas em BASIC são < (me-

nor que); = (igual a); ≥ (maior que ou igual a); ≤ (menor que ou igual a) e ≠ (diferente de). Veremos essas comparações usadas com frequência à medida que o curso evoluir.

Antes de continuar, faremos alguns exercícios para aprender o uso das comparações.

Exercícios

- Mude uma das linhas de forma que o programa seja interrompido se A = 1000.
- Mude uma das linhas de forma que o programa seja interrompido se o número introduzido for menor que zero.
- Mude a linha GOTO de forma que faça o programa voltar para o começo, se A for igual ou menor que 500. Sugestão: você não precisará de uma linha separada IF-THEN e de uma linha GOTO.

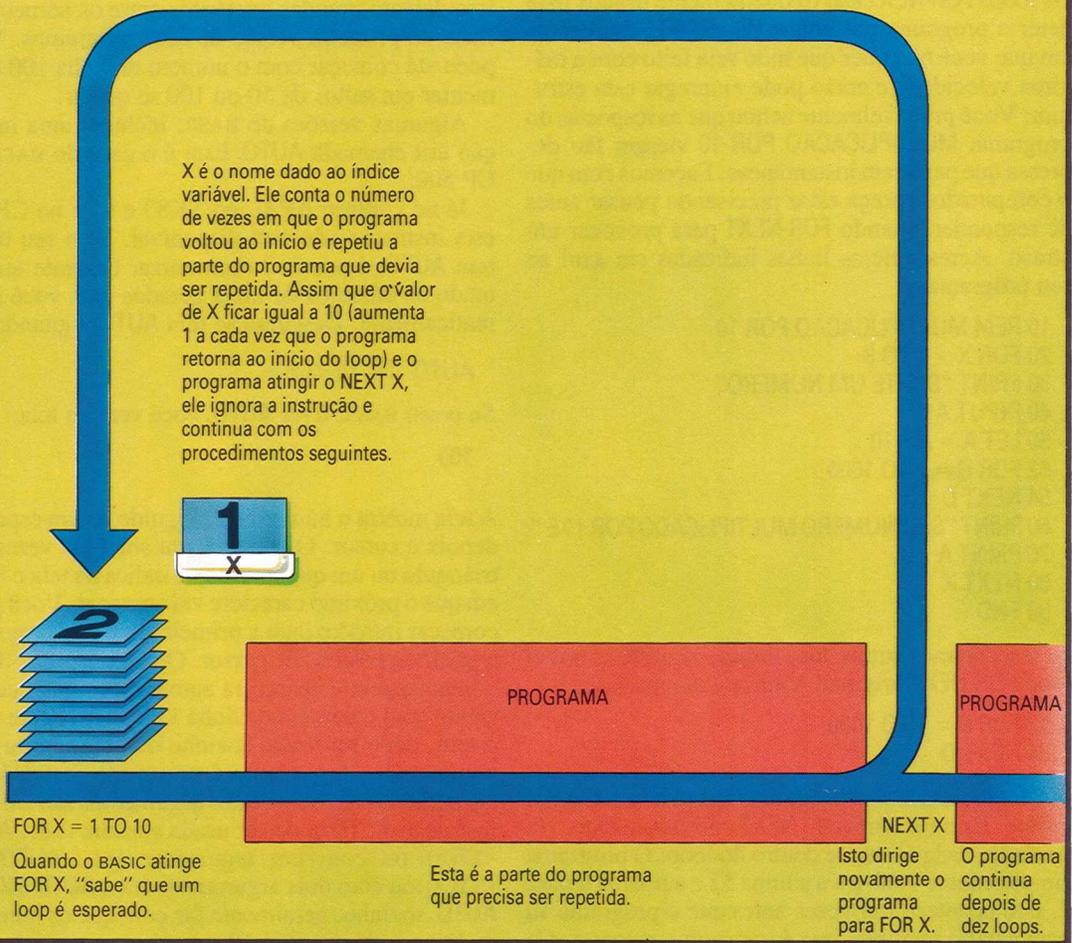
Descobrimo FOR-NEXT

Ao escrever programas, haverá muitas ocasiões em que você gostará de que alguns itens do programa se repitam determinado número de vezes. O GOTO na linha 70 possibilitou que o programa fizesse quantos loops quiséssemos. Mais tarde acrescentamos um comando IF-THEN na linha 35, que nos permitiu sair, introduzindo um número “fora de limite”.

```
10 REM COMPUTADORES NUNCA ERRAM
20 PRINT "DIGITE UM NUMERO"
30 INPUT A
40 LET A = A + 1
50 PRINT "EU ACHO QUE O NUMERO QUE VOCE DIGITOU FOI"
60 PRINT A
70 GOTO 20
80 END
```



Os comandos FOR-NEXT em BASIC criam um loop de forma que parte do programa possa ser repetida um número exato de vezes. Quando o BASIC encontra a palavra FOR sabe que a parte seguinte do programa será repetida. Ele cria uma variável (X, neste caso) e determina seu valor como sendo 1. A parte seguinte do programa (a parte a ser repetida) é "executada". Quando o BASIC encontra o comando NEXT X, ao invés de continuar com o restante do programa, volta para a linha FOR X, soma 1 na variável X (ilustrada aqui como o cartão sendo colocado em um suporte) e continua novamente com a parte loop do programa. Este processo é repetido dez vezes.



Esta é a parte inicial do programa.

Quando o BASIC atinge FOR X, "sabe" que um loop é esperado.

Esta é a parte do programa que precisa ser repetida.

Isto dirige novamente o programa para FOR X.

O programa continua depois de dez loops.

Entretanto, há ocasiões, como vimos na primeira parte do curso, em que usar o GOTO para fazer um loop não é a melhor maneira de agir.

Voltemos ao nosso antigo programa, desta vez modificado, realmente, para multiplicar o número que entra (input) por 10 e fazê-lo exatamente oito vezes.

```
10 REM MULTIPLICACAO POR 10
20 FOR X = 1 TO 8
30 PRINT "DIGITE UM NUMERO"
40 INPUT A
50 LET A = A * 10
60 PRINT "SEU NUMERO MULTIPLICADO POR 10 E ";
70 PRINT A
80 NEXT X
90 END
```

Digite este programa, LISTe para verificar erros e, então, RUN. Um número será solicitado apenas oito vezes. Depois disso, o programa simplesmente pára. A razão para que isso aconteça está na linha 20.

```
20 FOR X = 1 TO 8
```

Isto faz parte de um loop FOR-NEXT. É uma das estru-

turas mais úteis que o BASIC tem a oferecer e, por isso, merece um estudo cuidadoso.

Da forma como fizemos aqui, criamos uma variável chamada X. (As variáveis estão explicadas na primeira parte do curso, na p. 17.) Poderíamos ter-lhe dado qualquer nome (exceto A, que estamos usando para outra coisa). FOR deve sempre ser usado com o correspondente NEXT, mas NEXT aparecerá mais adiante no programa, depois da parte a ser repetida. A parte FOR de um loop FOR-NEXT sempre tem a seguinte forma:

FOR variável = valor inicial TO valor final. No nosso exemplo FOR X = 1 TO 8, chamamos a variável de X e lhe demos valor inicial de 1. A parte seguinte do programa é então executada pelo computador. O número que nós digitamos é multiplicado por 10 e então mostrado na tela. Depois disso vamos para NEXT X e o programa volta para onde a variável X está, na linha 20. Assim que isso tenha sido feito, X aumenta 1, e passa a ter o valor 2. A parte do programa dentro do loop FOR-NEXT é então outra vez executada. Ao chegar a NEXT novamente na linha 80, o programa volta e aumenta X para 3.

O programa se repete desta forma até que X tenha sido aumentado para 8. Depois disso, o loop termina. NEXT X não volta para FOR X e o programa continua na linha seguinte.

Mais usos para os loops FOR-NEXT

Os loops FOR-NEXT são freqüentemente usados para deter o programa por algum tempo. Há momentos em que você não quer que tudo seja feito com a máxima velocidade e então pode empregar esta estrutura. Você provavelmente achou que as respostas do programa MULTIPLICACAO POR 10 vieram tão depressa que pareciam instantâneas. Façamos com que o computador pareça estar precisando pensar antes de responder, usando FOR-NEXT para provocar um atraso. Acrescente as linhas indicadas em azul ao seu programa.

```
10 REM MULTIPLICACAO POR 10
20 FOR X = 1 TO 8
30 PRINT "DIGITE UM NUMERO"
40 INPUT A
50 LET A = A * 10
52 FOR D = 1 TO 1000
54 NEXT D
60 PRINT "SEU NUMERO MULTIPLICADO POR 10 E ";
70 PRINT A
80 NEXT X
90 END
```

Acrescentamos outras duas linhas, 52 e 54, ao nosso loop FOR-NEXT original. Vamos examiná-las.

```
52 FOR D = 1 TO 1000
54 NEXT D
```

D é fixado em 1 e o programa vai para a linha seguinte. Este é o comando NEXT correspondente. Na verdade, nada acontece dentro do loop. O programa simplesmente volta para a linha 52 e aumenta D para 2. Isto acontece mil vezes antes que o programa vá para a parte seguinte, que é imprimir a resposta. Os computadores são rápidos, mas tudo tem um tempo finito, de forma que fazer mil vezes um loop gasta um tempo facilmente perceptível. Os computadores variam no tempo que levam para fazer um loop. No Unitron AP II, o loop FOR-NEXT leva 1,9 segundo, enquanto no TK85 leva 4,5 segundos. Experimente, mudando o número que você usa como limite mais alto na linha 52.

Para fazer com que o computador se comporte mais como um ser humano, acrescente estas três linhas:

```
56 PRINT "AGORA DEIXE-ME VER..."
57 FOR E = 1 TO 1000
58 NEXT E
```

LISTe o programa e RUN. Temos agora dois atrasos que não levam a nada, a não ser a perder tempo.

Acrescente estas duas linhas:

```
51 REM ESTE LOOP DESPERDICA TEMPO
55 REM ISTO DESPERDICA MAIS TEMPO
```

Agora LISTe o programa e preste atenção a ele. Veja como todas as linhas extras que acrescentamos se encaixaram nos lugares certos. O que nos leva ao último ponto desta parte do curso: número de linhas.

Começamos nosso programa original com a linha 10 e fomos aumentando, em saltos de 10 para cada nova linha, terminando com a linha 90. Poderíamos

ter escolhido quaisquer números, por exemplo 1, 2, 3... 9. Mas, se tivéssemos feito isso, como encaixaríamos as linhas extras? Os programadores sempre têm novas idéias e fazem melhoramentos, e por isso deixam grandes intervalos entre os números de linha na primeira versão de seus programas. Você pode até começar com o número de linha 100 e aumentar em saltos de 50 ou 100 se quiser.

Algumas versões do BASIC incluem uma instrução útil chamada AUTO. Este é o caso do BASIC do CP 500.

Já no Unitron AP II, no TK83 e 85 e no CP 200 esta instrução não está disponível. Se o seu BASIC tem AUTO, é possível economizar bastante tempo, tendo os números de linha gerados para você automaticamente. Descubra se tem AUTO digitando:

```
AUTO 100, 10 <CR>
```

Se o seu BASIC tiver AUTO, você verá na tela:

```
100
```

A tela mostra o número 100 seguido de um espaço e depois o cursor. O cursor é um sinal (às vezes um triângulo ou um quadrado) que indica na tela o lugar em que o próximo caractere vai aparecer. Você pode começar introduzindo a primeira linha do programa a partir da posição do cursor. Quando atingir <CR>, a linha seguinte aparecerá automaticamente, começando com o número de linha 110. O AUTO, se você o tem, pode ser usado sozinho ou com um ou dois argumentos. Argumento é um termo matemático. Na expressão $2 + 3 = 5$, os argumentos são 2 e 3. A instrução AUTO pode ser usada sozinha (i.e., AUTO <CR>), ou com um argumento (ex.: AUTO 100 <CR>), ou com dois argumentos (ex.: AUTO 300,50). AUTO, sozinho, geralmente faz com que os números de linha comecem em 10 e aumentem em saltos de 10. Se apenas um argumento é usado (ex.: AUTO 100 <CR>), o primeiro número será 100 (neste caso) e então os números aumentarão de acordo com o "valor default", que geralmente é 10. Se você especificar dois argumentos, o primeiro número especifica o número de linha inicial e o segundo especifica o incremento. AUTO 250,50 <CR> indica um número inicial de 250; o próximo número será 300 e assim por diante, aumentando sempre 50. Mesmo no micro mais simples é improvável que a numeração das linhas chegue ao seu limite.

Na próxima parte deste curso, veremos as várias maneiras de melhorar a apresentação visual do programa na tela e diferentes formas de imprimir dados.

A propósito...

IF

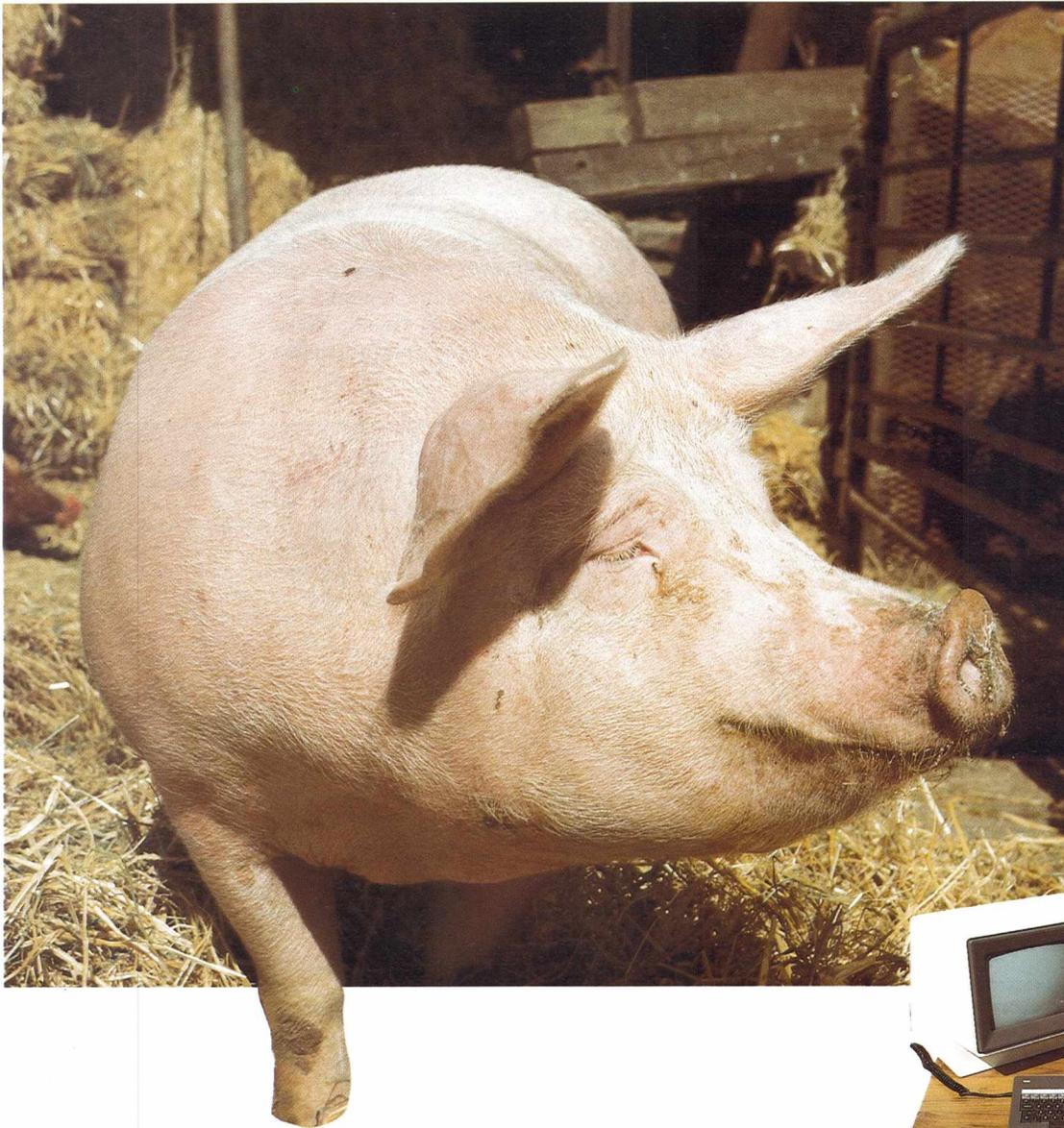
A maioria dos microcomputadores pode usar esta instrução na forma IF A > 999 THEN 80 ou IF A > 999 GOTO 80.

AUTO

Esta instrução não existe nos computadores da linha Apple (Unitron AP II) e Sinclair (TK83).



Nos bastidores



O computador na fazenda

Os computadores já atuam não só em escritórios, mas em casa e na fazenda.

Um engenheiro inglês especializado em software desenvolveu um programa chamado Optimiser, que reduz o custo da alimentação dos porcos. Uma fazenda de porcos consome normalmente 400 toneladas de ração por mês; entretanto, a mistura de cereais muda diariamente, enquanto os leitões estão sendo engordados para a venda. O uso do computador faz com que o fazendeiro possa suprir as necessidades diárias de proteínas e vitaminas do animal. O programa é capaz ainda de determinar a mistura economicamente mais vantajosa.

Com o aumento constante do preço das rações, esse sistema de programação tem ajudado os fazendeiros a fazer um cálculo mais eficaz da sua composição. Mas nem todos os problemas estão resolvidos, já que os animais se acostumam a uma dieta regular, sem mudanças drásticas de um dia para outro. Apesar disso, esse programa é vendido no mundo inteiro.



O computador economiza muito tempo e trabalho na empresa. Com isso, a lucratividade se tornará maior.

O computador teve origem em laboratórios militares e universitários. Os primeiros aparelhos foram construídos para calcular a trajetória de cápsulas disparadas de um navio de guerra e prever a temperatura em alto-mar.

Entretanto, não foi preciso muito tempo para que o uso comercial do computador fosse valorizado. Se, inicialmente, apenas as grandes empresas podiam arcar com as despesas de um sistema de computação, bastou a evolução dos microaparelhos nos anos 70 para que o poder dos computadores ficasse ao alcance também das pequenas e médias empresas.

Por que, afinal, esse produto é tão útil para cien-

tistas militares quanto para donos de supermercados? O computador pode receber e acumular enorme quantidade de dados, além de reorganizar informações com absoluta eficiência. Em um supermercado, há grande estoque de alimentos, utensílios domésticos e produtos variados. As cifras de venda são registradas no computador da matriz e um programa especial confere o estoque. No momento em que se registra insuficiência na reserva de determinado item, o aparelho envia uma mensagem a fim de que nova compra seja efetuada. E essa mesma máquina, previamente programada, apresenta a quantidade da compra a ser feita e o seu valor.

A utilidade do computador não termina aí, pois

ele ainda poupa o tempo do empregador ao calcular salários e tributos e ao fazer a contabilidade anual. Empresas comerciais podem ainda cadastrar nomes e endereços de clientes e, ao realizarem uma promoção de venda nos seus estabelecimentos, terão condições de emitir, com um simples comando, todas as etiquetas de endereçamento desejadas. Bastará então colá-las nos envelopes e enviá-los. O tempo e o custo despendidos no trabalho de datilografia foram significativamente reduzidos.

São vantajosos, portanto, e muito mais do que isso: os computadores abrem novos rumos para a humanidade. Sem eles, o homem nunca teria chegado à Lua, pois apenas um computador seria capaz de solucionar os problemas apresentados no projeto de um foguete. Mas, se o pequeno e o médio proprietário perguntam o que um computador pode fazer melhor do que qualquer outro aparelho, a resposta é:

economia. O advento do microcomputador fez com que atividades anteriormente dispendiosas e demoradas se tornassem agora lucrativas.

Antes de possuir um microcomputador, o dono de um supermercado tinha inúmeras "complicações": precisava de vários departamentos para efetuar o controle de estoque, a administração do pessoal e a contabilidade geral da empresa. Cada departamento desses representava um custo elevado, com funcionários que nem sempre forneciam os resultados corretos no tempo desejado.

Com um microcomputador adequado a suas necessidades, em pouco tempo esse quadro se modifica e o empresário consegue grande agilidade nos seus negócios. Pode desenvolver ele mesmo os seus programas ou comprar um dos muitos sistemas que se encontram à venda em forma de pacote. Neste caso, deve verificar com cuidado se o software esco-

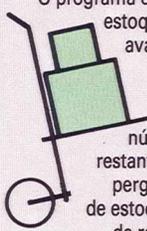
O microcomputador comercial



O modelo adequado para a pequena empresa deve estar em condições de suportar horas de uso ininterrupto. Um bom teclado é essencial e unidades de disco serão necessárias ao

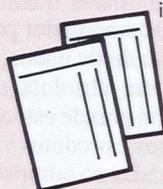
funcionamento dos softwares comerciais. Um equipamento do tipo aqui ilustrado custa cerca de 3.500 dólares e é um sistema conveniente.

Controle de estoque



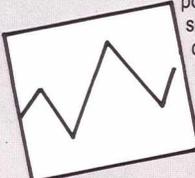
O programa de controle de estoque possibilita ao operador avaliar quando e como o estoque está se esgotando. Se um operador quer saber sobre certo produto, basta digitar o número de embalagens restantes (em resposta a uma pergunta na tela) e o programa de controle de estoque transmite um pedido de reposição de estoque, à medida que este cai abaixo da média desejada. Em alguns países, os programas de controle de estoque obtêm informações mediante a identificação dos produtos por um código de barras, detectadas por uma leitora óptica no momento da venda, e a baixa no estoque é dada automaticamente.

Faturamento



Esse tipo de programa produz faturas impressas com menos trabalho e mais precisão. Quando ele está em funcionamento, o operador responde a algumas questões e as respostas apropriadas são registradas no teclado. O programa verifica nos registros se as faturas são legítimas e se os detalhes estão corretos. Alguns programas podem até mesmo trocar referências com o programa de controle de estoque, para que o balanço da empresa se mantenha em dia.

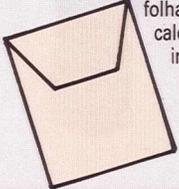
Contabilidade



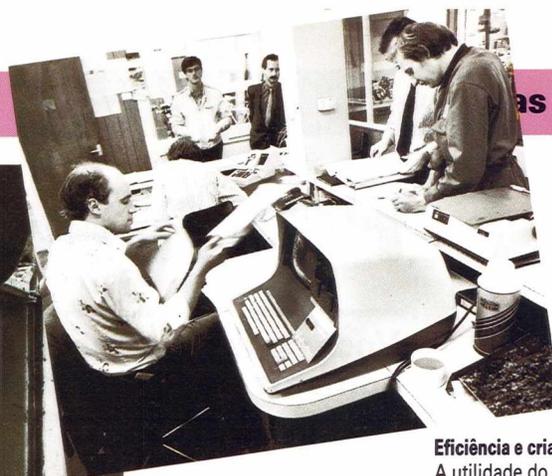
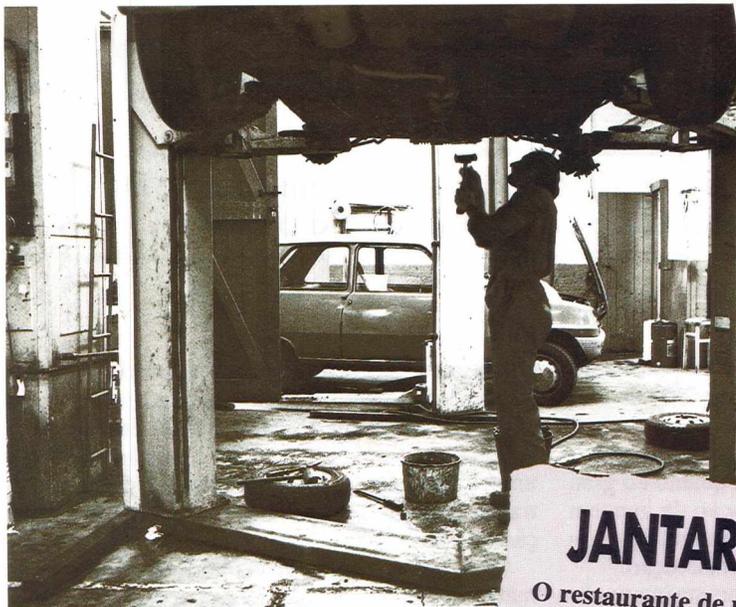
Toda a contabilidade de uma empresa pode ser feita com o auxílio de um software especial. Tanto o diário de vendas como o livro caixa podem ser automatizados para produzir cálculos anuais, trimestrais ou mensais, visando a satisfazer os contadores e auditores.

Os melhores sistemas de contabilidade são os compostos de "módulos" que trabalham em conjunto, de tal forma que um dado fornecido a uma seção seja automaticamente atualizado em todas as outras. O administrador tem, assim, a vantagem de gastar menos tempo com a contabilidade rotineira, ciente de que os cálculos são exatos e maior o potencial de lucratividade.

Folha de pagamento



O programa da folha de pagamento calcula os salários e imprime os holerites de todos os funcionários. Este sistema deve considerar as horas regulares de serviço e as horas extras, fazer a dedução de I.R. e o cálculo dos encargos. O operador deste sistema geralmente responde a perguntas referentes ao número de empregados que constam da folha de pagamento, aos salários, descontos, faltas, atrasos etc. As respostas a essas perguntas são fornecidas através do teclado do computador.



Eficiência e criatividade

A utilidade do computador estende-se praticamente a todas as áreas de atuação. Você pode até mesmo levar seu carro para uma revisão eletrônica e obter do computador uma lista de checagem a ser efetuada no veículo. O mecânico executa o trabalho tendo a lista como guia para revisão das peças, sem perder tempo na procura das que apresentem defeito. Quando a revisão estiver pronta, o computador calcula a fatura das peças e serviços.

O computador faz a verificação de rotina e libera o usuário para trabalhos mais criativos.

JANTAR IMPECÁVEL

O restaurante de uma famosa avenida parecia ser bem-sucedido, pois permanecia sempre lotado. Mas o que preocupava seus administradores era saber se as despesas e os salários dos funcionários não estavam sendo excessivos.

A aquisição de um computador resolveu a questão. A gravação dos pedidos efetuados e a elaboração de contas exatas minimizaram a margem de erro. A eficiência gerada pelo sistema trouxe ainda vantagens no atendimento ao público. Os donos do restaurante queriam também manter um esquema de gratificação; por isso, adquiriram um sistema para calcular a folha de pagamento e um outro para manter o controle de estoque e o custo dos alimentos.

UMA FATIA PARA TODOS

Um dia o Manuel instalou um computador na sua panificadora. Havia percebido que a massa pronta para encomendas de rotina poderia ser aprimorada com o emprego do computador. E foi o que aconteceu.

A nova máquina se encarrega ainda de faturas semanais, da folha de pagamento, do custo das receitas, das entregas e dos descontos.

Manuel está contente com o equipamento, que possui uma unidade de disco e uma impressora.

A única parte difícil da tarefa foi registrar 350 encomendas fixas no arquivo. Mas, feito isso, o computador tem trabalhado sozinho e poupado muito tempo.

O LEITE DO FUTURO

Um fazendeiro da região de Piracicaba, em São Paulo, estava empenhado em aumentar a produtividade do seu rebanho. E conseguiu ótimos resultados com a aquisição de um computador. Agora ele obtém rapidamente relatórios que informam sobre o número de vacas que deram cria, as que já foram inseminadas e as que estão produzindo menos. Se um animal produz menos leite, o criador logo saberá, destinando-o ao corte. Utilizando o micro, faz previsões da produção de leite anual, bem como dos lucros que serão obtidos com a venda do produto. Assim, a produção leiteira é cada vez mais bem controlada.

lhido realmente se adapta à realidade de sua empresa. Outra opção seria contratar um programador profissional que desenvolveria um software especificamente voltado para as necessidades da empresa. Dependendo do porte e complexidade do empreendimento, esta pode ser a solução mais aconselhável.

Muitos outros sistemas podem ser comprados já prontos; entre eles, o de processamento de palavras, que interessa particularmente aos homens de negócios. Neste sistema, faz-se a correção com facilidade na própria tela; quando completada, uma cópia correta pode ser impressa várias vezes. A repetição é cansativa para quem trabalha, mas um computador nunca se cansa.

Com o avanço da tecnologia, um número crescente de pessoas terá acesso ao computador no trabalho ou mesmo em casa. A comunicação direta entre computadores no futuro pode eliminar as enco-

mendas feitas por carta e as faturas que nunca se completam. E o uso do computador então se tornará tão necessário quanto o do telefone.

Pintando com números

Você pode criar imagens e gráficos com seu computador. Os números são os instrumentos, sua imaginação é o limite.

Cria-se um quadro aplicando centenas de pinceladas sobre uma tela. Mas como poderia um artista criar imagens com um computador?

Há três métodos principais para a criação de gráficos com computação e eles diferem quanto ao grau de controle da resolução ou granulação da imagem final. Todos os computadores utilizam gráficos em bloco, "pixels" ou gráficos de alta resolução.

Nos gráficos de alta resolução, o artista busca obter o controle sobre cada ponto de fósforo no monitor de televisão. O controle é limitado apenas pela capacidade de memória do computador, que determina exatamente como a memória da tela modela o vídeo de televisão. Em um computador de 32 K, cada ponto de fósforo tem um correspondente no modelo da tela no computador.

Nos gráficos em bloco, o artista ganha em facilidade o que pode perder em controle sobre cada ponto que compõe a tela. Formas básicas já se en-

contram prontas nos softwares para a construção de imagens. São controladas diretamente pelo teclado e em geral aparecem na parte frontal de cada tecla. Acionada uma tecla que altera a função das outras, um teclado semelhante ao de máquina de escrever se transforma em "paleta", que possibilita a elaboração de gráficos em bloco.

Cada figura é formada no interior de uma pequena matriz de pontos, de oito linhas por oito colunas. Alguns microcomputadores possibilitam até mesmo definir seus próprios caracteres em bloco. Um programa menor é usado para definir os novos caracteres e acrescentá-los aos do computador.

Os "pixels" situam-se entre os gráficos em bloco e os de alta resolução. Possibilitam ao artista o controle sobre a célula de imagem (picture cell, daí o nome "pixel"), que contém mais do que um ponto de fósforo, porém menos do que um bloco de oito pontos por oito. Cada "pixel" pode ser chamado se-

Os três tipos de resolução



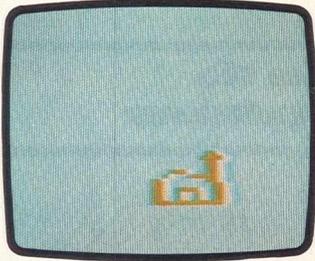
Nos gráficos em bloco de baixa resolução, pouca capacidade de memória, como 1 Kbyte, é suficiente para armazenar todos os detalhes do que deve aparecer na tela. Se houver apenas quarenta blocos em cada linha e só 24 linhas na tela, o número total de blocos será de 960, quase 1 kilobyte. Assim, 1 kilobyte de memória para a tela pode armazenar até 8 bits de dados para cada bloco na tela.

Em gráficos de média resolução, há necessidade de maior precisão de detalhes; assim torna-se necessária maior capacidade de memória. Oito kilobytes de RAM podem fornecer 65.536 pontos de luz na tela, que se apresentam iluminados ou não. Se a composição for em cores, parte da capacidade da memória será necessária para especificar a cor de cada ponto.

Uma figura em cores, de alta resolução, exige grande capacidade de memória. Composições de alta resolução na tela às vezes têm até 640 pontos em cada linha e 240 linhas na tela, num total de 153.600 pontos. Se a composição na tela for em uma cor, bastarão 19.200 bytes (153.600/8). Será necessária uma quantidade maior de memória se quisermos utilizar mais cores.



Gráficos em dimensão



Os gráficos em dimensão foram inicialmente desenvolvidos pela Texas Instruments, mas são hoje encontrados para vários microcomputadores, inclusive o Texas Instruments TI99/04A, o Commodore 64, os computadores da Atari e o Sord M5. (Até o momento esses computadores não estão disponíveis no mercado brasileiro.)



Em gráficos convencionais, as figuras são construídas em uma tela única, como aconteceria se você as pintasse numa folha de papel. Nos gráficos em dimensão, o artista do computador dispõe de vários "planos" ou camadas, cada qual podendo ter suas próprias figuras. Em alguns computadores com gráficos em dimensão, podem existir até 32 planos.



O modo mais fácil de ter noção do que seriam esses planos consiste em imaginá-los como folhas de plástico transparente: se a folha mais "próxima" ao observador mostrar a figura de uma árvore, enquanto a que estiver atrás apresentar a de uma nuvem, esta será vista passando atrás da árvore, como se estivesse sendo carregada pelo vento, no ar.



Desse modo, podem-se criar efeitos tridimensionais bem convincentes.

Uma vez "desenhado" um "objeto" qualquer (um pássaro, por exemplo), o programador pode deixar de lado os detalhes da construção da figura. Se quiser que se mova na tela, basta especificar a velocidade e a direção.

paradamente e localizado na posição desejada na tela.

Em gráficos de alta resolução, para computadores da linha Apple, é usado um comando BASIC para o desenho das linhas. O comando HLOT x,y identifica o ponto na posição (x,y) da tela, enquanto o comando HLOT x,y TO w,z faz com que seja traçada uma linha que vai do ponto (x,y) até o ponto (w,z) . As linhas são geralmente numeradas no sentido da parte superior da tela para a inferior, e as colunas numeradas da esquerda para a direita. Assim, o ponto situado na linha zero da coluna zero encontra-se no canto superior do lado esquerdo da tela.

O programa abaixo pode ser processado nos modelos compatíveis com Apple (Unitron, Maxxi, Micro Engenho e outros). Lembre-se de passar o microcomputador para a modalidade gráfica. Digite exatamente como indica o programa, que poderá ser processado em qualquer computador com gráficos de alta resolução.

```
10 HGR
20 HLOT 92,95 TO 57,125
30 HLOT 57,125 TO 100,110
40 HLOT 100,110 TO 143,125
50 HLOT 143,125 TO 108,95
60 HLOT 108,95 TO 100,50
70 HLOT 92,95 TO 100,50
80 END
```

Quando você PROCESSAR esse programa, uma forma trinácia surgirá na tela, semelhante à da ilha da Sicília (e assim se denomina porque Trinácia é o antigo nome latino da ilha). O esquema do programa serve para desenhar qualquer forma constituída por uma rede de linhas unidas. Os números apresentados após o comando HLOT indicam a posição, na linha e coluna, do ponto até o qual a figura deve estender-se. Pode-se desenhar qualquer coisa na tela usando unicamente esses dois comandos. Esse uso é limitado apenas pela resolução da tela do computador e pela persistência do usuário, pois mesmo as curvas devem ser representadas por pontos e os co-

mandos HLOT x,y e HLOT x,y TO w,z possibilitam o controle desses pontos.

O programa apresentado a seguir fornece a imagem de um cone traçado por uma combinação de círculos. Foi desenvolvido de modo a ser processado pelo micro inglês Spectrum, da Sinclair.

```
10 FOR K = 2 TO 40
20 CIRCLE 40 + K,40 + K,K
30 NEXT K
```

Os dois programas ilustram o processo pelo qual um computador pode dar origem a composições gráficas na tela por meio de comandos que utilizam números. Todavia, os procedimentos digitais são falhos sempre que se tenta imitar processos de fluxo contínuo. Por isso, há dispositivos especiais de hardware para desenho de imagens, que são conjugados aos microcomputadores e permitem ao artista livrar-se de fornecer os milhares de números necessários para imagens mais apuradas. O digitalizador é um desses dispositivos. O artista "executa" o desenho com uma "caneta" especial e o digitalizador traduz seu movimento para os números de colunas e linhas compatíveis com o computador.

A propósito...

ALTA RESOLUÇÃO

Em computadores da linha Apple (Unitron, Micro Engenho, Maxxi, Exato etc.), a composição da tela é de até 53.760 pontos com uma matriz de 192×280 pontos.

PROGRAMA COM CIRCLE

O comando CIRCLE não existe na maioria dos micros encontrados no Brasil. Para se ter o mesmo efeito deste comando, seria necessária a utilização de fórmulas trigonométricas junto aos comandos HLOT e HLOT TO.

Outras instruções podem ser utilizadas na elaboração de gráficos. Elas vão variar dependendo do modelo de seu computador e do BASIC que ele utiliza.

Contato!

Os surpreendentes progressos da microeletrônica são responsáveis pela revolução causada pelos computadores.

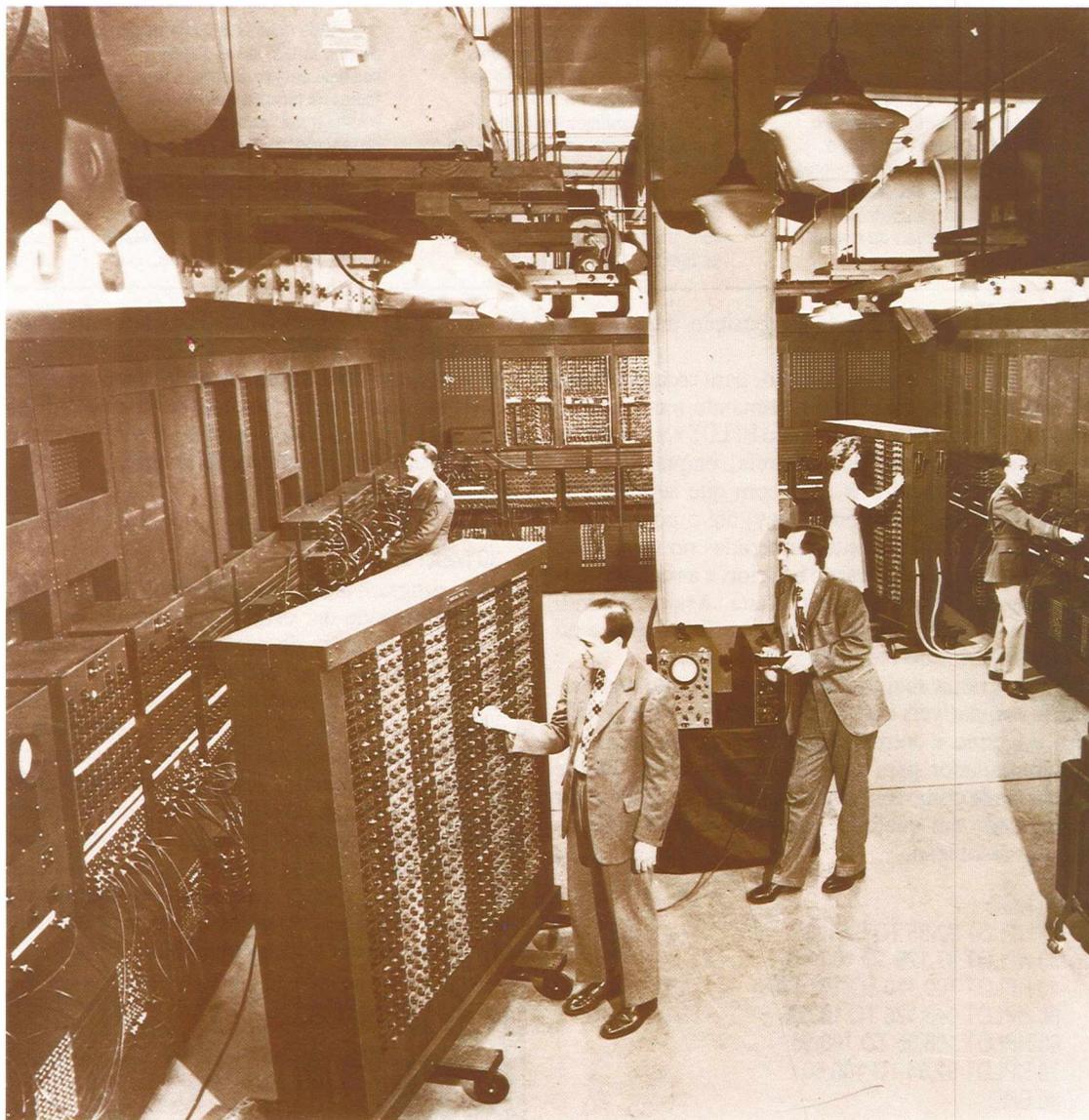
O computador a válvulas

Em 1943, no auge da Segunda Guerra Mundial, um coronel do exército dos Estados Unidos sugeriu que se fizesse uma máquina de calcular para fins de artilharia. O desafio foi aceito pela Universidade da Pensilvânia, que, em 1946, apresentou seu invento, tendo empregado 7.237 horas-homem para concluí-lo.

O equipamento recebeu o nome de ENIAC (Electrical Numerical Integrator and Calculator) e foi o primeiro computador a válvulas. Possuía 18.000 válvulas, 1.500 relés e emitia o equivalente a 200 quilowatts de calor. Essa enorme máquina foi alojada em uma sala de 9 m por 30 m.

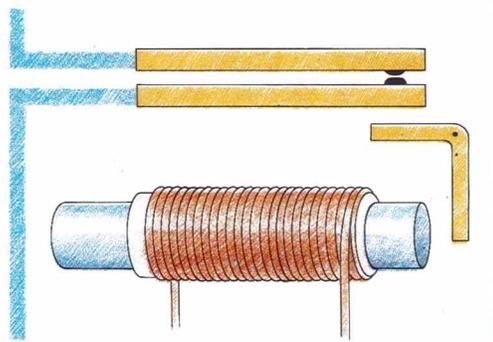
Capacidade de memória e confiabilidade foram os primeiros problemas. O ENIAC tinha condições de armazenar apenas vinte números de dez dígitos e toda programação precisava ser feita reordenando-se a rede elétrica. Em 1952, mais de 19.000 válvulas tinham sido substituídas: elas começavam a queimar dois minutos depois de ligar-se o equipamento.

O ENIAC teve vida ativa curta e foi aposentado em 1952.



Interruptor de relé

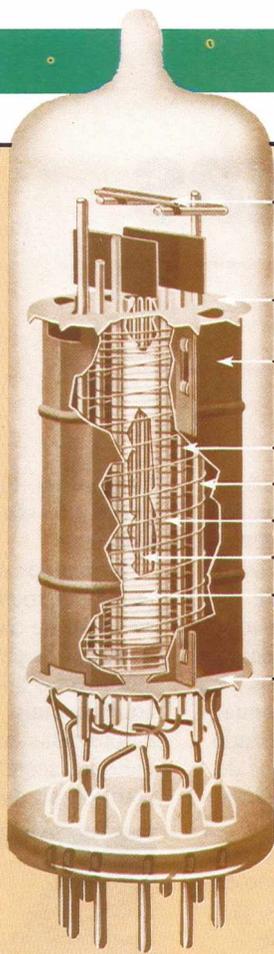
Quando a corrente passa pela bobina que envolve o eixo de ferro, produz-se força magnética. Essa força atrai a peça de ferro em forma de L que gira em ângulo reto. Ao girar sobre esse eixo, a tira junta os dois contatos e, desse modo, fecha o interruptor.



O moderno computador possui milhões de pequenos interruptores eletrônicos, essenciais a sua estrutura. Sem eles, a revolução tecnológica, que aconteceu após a Segunda Guerra Mundial, não teria sido possível.

Em 1938, o engenheiro eletrônico Claude Shannon provou que operações lógicas podem ser executadas utilizando-se interruptores de circuitos elétricos. Uma vez evidenciado que o trabalho do computador consistia em uma seqüência de operações lógicas, o objetivo da pesquisa passava a ser a produção do interruptor eletrônico.

A primeira tentativa teve como resultado a cria-



SUPORTE DE ABSORVEDOR DE GASES

MICA

ÂNODO

GRADE AUXILIAR

GRADE SUPRESSORA

GRADE DE CONTROLE

CALEFATOR

CÁTODO

MICA

Além dos três componentes fundamentais (cátodo, ânodo e grade), a maioria das válvulas contém alguns componentes adicionais para melhorar o desempenho. O princípio da geração, todavia, permanece o mesmo.

Válvula

O tubo de vidro na ilustração possui um terminal positivo e um negativo (o cátodo e o ânodo) separados por uma grade de fios. Os elétrons do cátodo são atraídos pelo ânodo e fazem uma corrente de elétrons passar através da válvula.

O fluxo natural dos elétrons negativos para o terminal positivo é intensificado de duas maneiras: o cátodo se aquece até a incandescência e sua superfície é tratada com um produto químico especial.

A grade de controle está situada entre o cátodo e o ânodo e geralmente não interfere no movimento dos elétrons. Mas, quando uma carga negativa é aplicada à grade, os elétrons são repelidos e não atingem o ânodo. A força repulsora da grade supera a força de atração do ânodo e o fluxo dos elétrons cessa. A corrente é interrompida e o interruptor se desliga.

ção do relé, que foi usado com resultados satisfatórios no início da era da computação, mas cuja estrutura limitava o desenvolvimento e a capacidade do computador. O relé não operava apenas eletronicamente; por isso, os componentes mecânicos com frequência ocasionavam paralisações e seu funcionamento era lento e de pouca confiabilidade.

A primeira geração de computadores caracterizou-se pelo uso da válvula que funcionava apenas eletronicamente e, por conseqüência, de modo mais rápido. Todavia, além de tornar o equipamento muito volumoso, necessitava de alta capacidade energética, o que aumentava o custo e causava problemas em decorrência do calor produzido.

A invenção do transistor deu origem a uma nova geração de computadores. Os transistores teoricamente funcionam de modo semelhante às válvulas, mas têm desempenho superior, são menores e de produção mais barata. Esses avanços levaram os computadores para fora das universidades e das instituições militares, introduzindo-os no mundo comercial.

Os computadores de hoje ainda usam o transistor como interruptor, mas ele não mais constitui um elemento distinto e separado. Sobre um chip de silício, do tamanho de uma unha, pode haver quase 250.000 transistores, todos pequenos demais para serem percebidos a olho nu. Pela concentração dos milhares de interruptores necessários ao funcionamento do computador em um pequeno chip de silício, tornou-se possível economizar em outros itens importantes do custo. Os computadores mais potentes e caros da década de 50, que ocupavam um laboratório inteiro, foram reduzidos a um simples chip, que constitui a matriz dos procedimentos dos modernos microcomputadores.

Interruptor por transistor

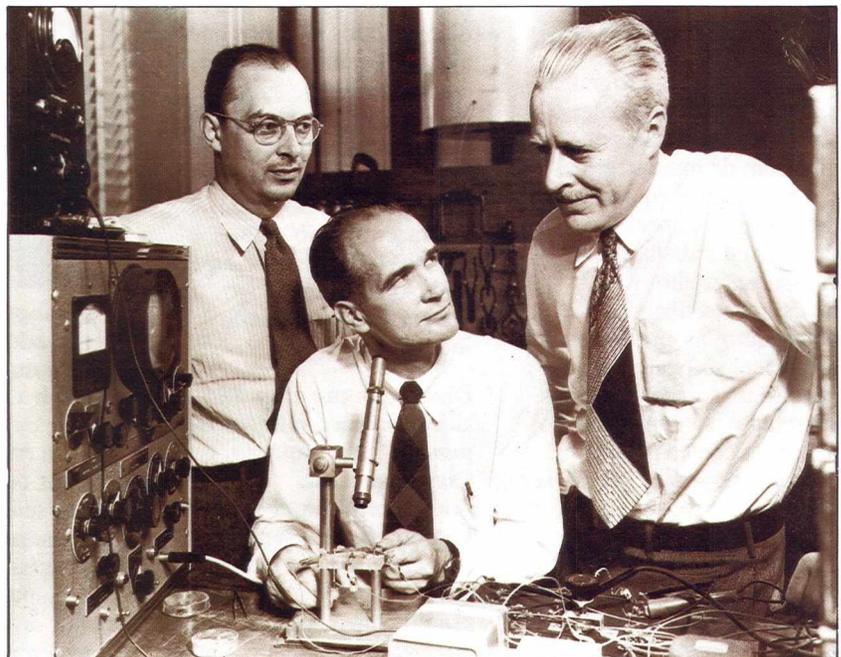
O interruptor pode estar ligado ou desligado. Em um transistor, os dois estados



resultam de uma corrente em fluxo ou interrompida. Isso é controlado pela aplicação de voltagem ao fio de controle.

Os inventores do transistor

O Prêmio Nobel de 1956 foi conferido à equipe cuja pesquisa levou à invenção do transistor, em 1947. Nesta foto, tirada nos laboratórios da Bell Telephone, encontram-se, da esquerda para a direita, os premiados John Bardeen, William Shockley e Walter Brattain.



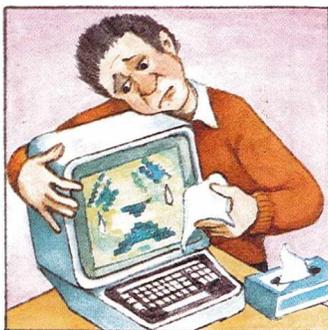


Perguntas e respostas

Aqui estão algumas dúvidas sobre o mundo da computação, que nem sempre são respondidas nos livros e revistas.

Os computadores podem ter emoções?

Computadores não têm e não podem ter emoções, pelo menos até o momento. Os computadores de 1984 não são inteligentes e é difícil saber quanto tempo passará até que se produzam computadores que pensem. Alguns pesquisadores são da opinião de que o pensamento criativo é inseparável das emoções. De acordo com esse ponto de vista, os computadores capazes de pensar deverão ter emoções.

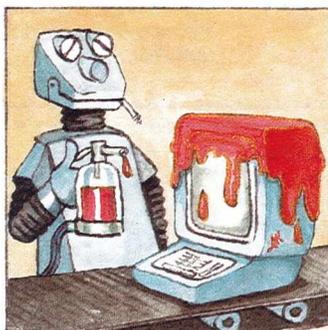


Por que existe uma diferença de preço tão grande entre os diversos tipos de software?

Para desenvolver programas extensos, especialmente softwares comerciais que precisam de muitos testes, são necessárias equipes de programadores bem pagos, ou anos de trabalho. Para compensar o alto investimento financeiro e obter lucro, as empresas que produzem softwares têm de vendê-los a preços elevados. Entretanto, se considerarmos os jogos, centenas e até milhares de cópias podem ser vendidas; dessa forma, o preço unitário pode baixar.

Se o potencial de mercado for muito restrito, o preço ao consumidor terá de ser elevado. Muitos programas

são altamente especializados; um pacote de previsão para impressão (permite que cálculos para trabalhos de impressão sejam feitos rapidamente e com precisão) tem um potencial de mercado limitado ao número total de gráficas do país. Um investimento de centenas de milhares de dólares terá de compensar, quer sejam vendidas dezenas, quer milhares de unidades.



Qual a diferença entre computador e robô?

Os robôs são prolongamentos mecânicos dos computadores: seus braços e olhos. Todos os robôs que ajudam a montar carros e equipamentos de som possuem microcomputadores, mas ainda são "burros". Diante de uma nova situação, simplesmente não sabem o que fazer. Os robôs de amanhã terão computadores mais sofisticados e logo haverá robôs dotados de inteligência.

Diz-se que um chip de silício deixará milhões de pessoas sem emprego.

Como pode um microcomputador tornar o homem dispensável?

Os efeitos sociais a longo prazo do microcomputador são difíceis de prever, mas o que parece bem claro é que

estamos assistindo ao início de uma segunda Revolução Industrial. Computadores especialmente miniaturizados e microcomputadores de baixo custo, quando ligados a robôs mecânicos, podem facilmente ser adaptados para substituir o trabalho manual. Até o trabalho especializado corre esse risco: serviços de contabilidade podem agora ser executados por programas de computador, e empregos de composição em jornal estão ameaçados, já que as processadoras de palavras usadas por redatores podem ser diretamente ligadas ao equipamento eletrônico de composição. Os computadores realizam operações matemáticas complexas tão rapidamente e os robôs executam operações mecânicas tão bem que será necessário um menor número de funcionários para realizar certos trabalhos.



Os computadores poderiam ser usados para roubo de banco ou para dar início à Terceira Guerra Mundial?

Já que os computadores podem comunicar-se uns com os outros utilizando linhas telefônicas comuns, teoricamente seria possível "penetrar" em uma central de computador de banco e emitir ordens de transferência de fundos para

outra conta. Na prática, as coisas não são assim tão simples.

Os bancos utilizam processos modernos de proteção de dados para evitar o acesso de pessoas não autorizadas a informações confidenciais. As técnicas empregadas envolvem métodos secretos de codificação das informações.



É quase impossível decifrar esses códigos e, em muitos casos, nem mesmo os funcionários do banco têm acesso a eles. Um dos códigos usados para informações altamente confidenciais é tão difícil de decifrar que se calcula que o computador mais potente do mundo levaria bilhões de anos para fazê-lo.

Violar um sistema de computação militar seria ainda mais difícil. Os computadores de instituições militares geralmente não usam linhas telefônicas públicas justamente para evitar esse risco. As ligações feitas através de microondas e via satélite não são facilmente acessíveis a pessoas comuns. Alguém que fosse capaz de interceptar uma ligação em microondas, tendo acesso aos dados do computador, ainda assim precisaria decifrar os códigos.



CP 300

Desenvolvido a partir de avançadas técnicas de hardware, o CP 300 é um equipamento modular, que pode ser expandido aos poucos, conforme a necessidade do usuário.

O CP 300 é um microcomputador pessoal de porte médio, com capacidade de expansão e preço relativamente acessível. Sua unidade central de processamento (CPU) é um microprocessador Z80, igual ao do CP 500. A CPU é responsável pelo controle e organização de todas as funções do sistema. A memória do CP 300 é de 16 K ROM (para armazenamento de programas e linguagem BASIC residente), 48 K RAM (para armazenamento temporário de programas) e 1 K RAM (para buffer de vídeo).

O CP 300 possui teclado alfanumérico simplificado e uma barra de espaço, com 54 teclas auto-repetitivas e reset com proteção contra acionamento

involuntário. A formatação do vídeo do CP 300 pode ser realizada por software, obtendo-se então 16 linhas de 64 caracteres (normal), 16 linhas de 32 caracteres (expandido) ou 48 linhas de 128 caracteres (gráficos). Caso o usuário prefira não adquirir logo um monitor de vídeo apropriado, poderá conectar seu CP 300 a qualquer televisor comum, em cores ou preto e branco. Para isso, existe um cabo co-axial próprio, que deve ser ligado à entrada da antena do televisor.

Como se trata de um equipamento modular, o CP 300 pode ser expandido aos poucos, conforme as necessidades de seu usuário. Um dos periféricos

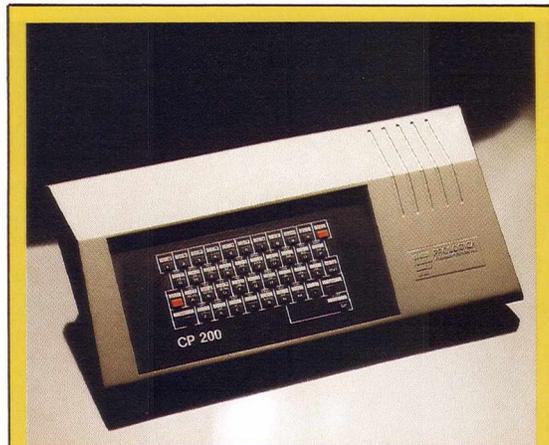




mais comumente acoplados ao equipamento é um gravador cassete (que funciona como memória auxiliar e para armazenamento permanente dos programas e dados). Para a transferência de dados entre o micro e o gravador podem utilizar duas velocidades: 500 e 1.500 bauds.

Também é possível trabalhar com até quatro unidades de disco flexível de 5 1/4 polegadas. Desse modo, pode-se armazenar e carregar rapidamente programas e dados, com um máximo de confiabilidade. Empregando esse sistema, torna-se possível dispor de 720 K.

Caso haja necessidade constante de listar programas, relatórios ou copiar o conteúdo do vídeo, o melhor é conectar uma impressora com interface paralela ou serial ao CP 300. E, com a Interface RS 232 C, é possível colocar o CP 300 em comunicação com outros computadores que possuam a mesma interface. Devidamente programado, o equipamento também é capaz de se comunicar com centrais de computação por telefone, o que eleva enormemente o número de dados e programas acessíveis ao usuário.



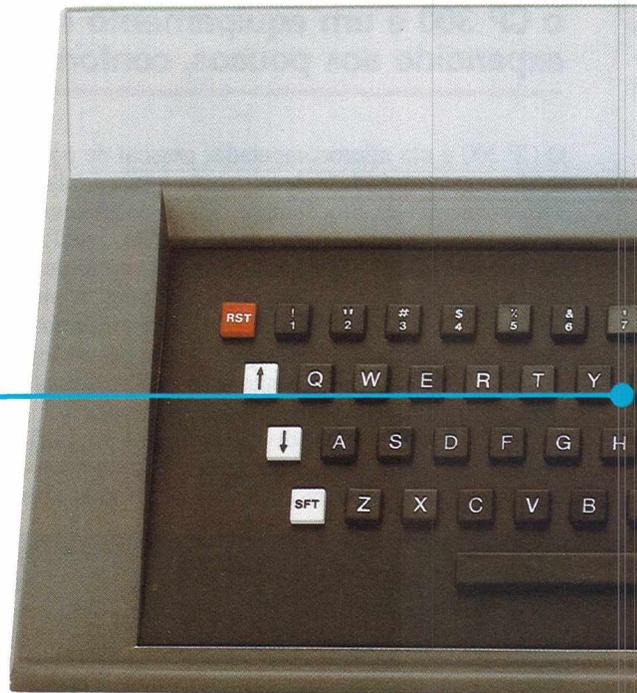
CP 200

O CP 200 é um microcomputador para iniciantes, ideal para as primeiras lições de computação e para recreação (videogames). A CPU desse equipamento é o microprocessador Z80 A, com relógio de 3,25 MHz. Tem memória de 16 K RAM, para armazenamento de programas e dados, e 8 K ROM (sistema operacional e interpretador BASIC).

O CP 200 tem um teclado semelhante ao das calculadoras. Cada tecla tem de 3 a 5 funções, num total de 160, todas elas impressas no próprio teclado. Isso facilita o aprendizado do iniciante na linguagem BASIC. Para visualização, o CP 200 precisa ser conectado a um televisor comum por um modulador RF. A tela é composta de 24 linhas por 32 colunas ou 44 linhas por 64 colunas (modo gráfico).

O CP 200 aceita como periféricos gravador cassete e joystick. Pode ser utilizado para controlar contas bancárias, organizar orçamentos domésticos e auxiliar nas tarefas escolares.

Teclado
Alfanumérico com função de auto-repetição



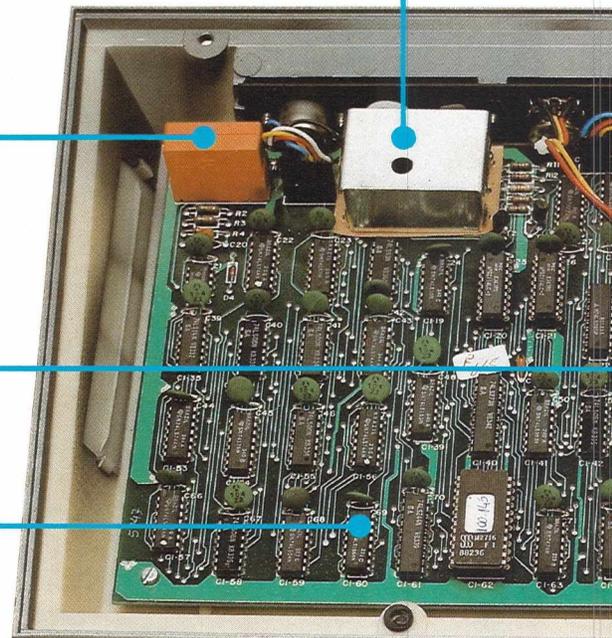
Relé
Controle do gravador cassete

Modulador de RF

Int gr

CPU
Z80 com clock de 2 MHz

Circuitos integrados para apoio ao microprocessador





CP 300

MICROPROCESSADOR

Z80

CLOCK

2 MHz

MEMÓRIA

16 K de memória ROM onde estão armazenados os programas internos, incluindo a linguagem BASIC. 48 K de memória RAM para armazenamento temporário de programas e dados. 1 K de memória RAM para buffer de vídeo.

VÍDEO

Utiliza um televisor comum ou um monitor. O vídeo pode ser formatado com 16 linhas e 64 colunas em modo normal, 16 linhas e 32 colunas em modo expandido e 48 linhas e 128 colunas em modo gráfico.

TECLADO

Alfanumérico com 54 teclas, tipo calculadora com função de auto-repetição.

LINGUAGENS

BASIC

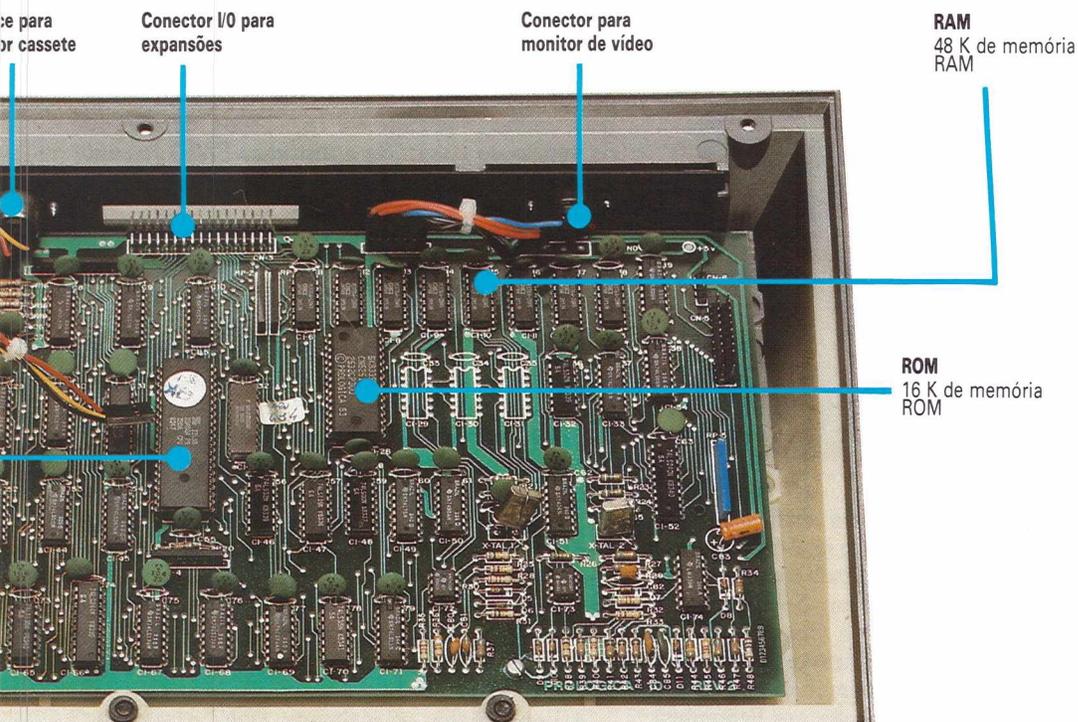
PERIFÉRICOS

Cassete, monitor, até 4 unidades de disco (5 1/4"), impressora, interface serial RS232 C, joystick.

DOCUMENTAÇÃO

O manual é destinado a iniciantes e explica em detalhes como instalar o equipamento. Descreve os comandos da linguagem BASIC, incluindo também os principais endereços de memória e as prováveis causas de defeitos. Apresenta um sumário final de instruções e um glossário de termos técnicos.

Auto-falante da unidade sonora



Direto ao ponto

Esteja atento aos detalhes de pontuação quando escrever um programa para seu micro. Eles são muito importantes.

Você deve ter observado no programa da primeira parte da programação BASIC que há um ponto e vírgula no final da linha 50. A função desse sinal de pontuação não foi explicada naquele momento, mas é muito importante. O sinal é utilizado em quase todas as versões BASIC para concatenar as seções impressas. As linhas 50 e 60 da página 38 são:

```
50 PRINT "ACHO QUE O NUMERO QUE VOCE
    DIGITOU FOI ";
60 PRINT A
```

A linha 50 imprimiu as palavras contidas entre as aspas. A linha 60 imprimiu o valor da variável A. A colocação do ponto e vírgula fez com que o valor da variável A fosse impresso diretamente após as palavras entre aspas, na linha 50. Se o ponto e vírgula não fosse empregado, o valor da variável teria sido impresso na linha seguinte.

O programa a seguir foi projetado para ilustrar algumas propriedades úteis do ponto e vírgula, tal como é usado no BASIC. Experimente digitá-lo e processá-lo. De agora em diante, omitiremos o <CR> no final de cada linha, para indicar que você deve pressionar a tecla RETURN. O programa a seguir permite a você registrar uma série de temperaturas em graus centígrados (conhecidos como Celsius) e tê-los automaticamente convertidos aos graus equivalentes em Fahrenheit.

```
10 REM PROGRAMA PARA CONVERTER GRAUS C EM F
20 PRINT "FORNECA A TEMPERATURA MAIS BAIXA"
30 INPUT L
40 PRINT "FORNECA A TEMPERATURA MAIS ALTA"
50 INPUT H
60 FOR X = L TO H
70 LET F = (X * 9 / 5) + 32
80 PRINT X; " EM CENTIGRADO E "; F; " EM FAHRENHEIT "
90 NEXT X
100 END
```

Digite este programa. LISTE-o para verificação e, depois, utilize o comando RUN para fazê-lo funcionar. Em primeiro lugar lhe será solicitado fornecer a temperatura mais baixa. Tente digitar -5. Em seguida, você será solicitado a fornecer a temperatura mais alta. Tente digitar 10. O programa converterá todas as temperaturas com 1 grau de intervalo de -5 a 10 graus centígrados aos equivalentes em Fahrenheit. Você deve obter uma impressão em blocos na tela semelhante a esta:

```
- 6 EM CENTIGRADO E 21.2 EM FAHRENHEIT
- 5 EM CENTIGRADO E 23 EM FAHRENHEIT
- 4 EM CENTIGRADO E 24.8 EM FAHRENHEIT
```

Observe que as colunas não estão uniformes, devido aos pontos decimais, mas cada valor em centígrado tem seu equivalente em Fahrenheit, em uma única linha. Após ter processado esse programa algumas vezes, redigite a linha 80, tal como é, mas substitua todos os pontos e vírgulas por vírgulas e acione o comando RUN novamente. Como você pode notar, a impressão tornou-se uma desordem.

Para verificar por que isso ocorreu, experimentemos um programa muito simples, comparando o efeito das vírgulas com o dos pontos e vírgulas. Digite NEW<CR>. Depois introduza:

```
10 REM COMPARACAO DO ; COM A,
20 PRINT "ESTA LINHA USA PONTO E VIRGULA"
30 PRINT "H";"E";"L";"P"
40 PRINT "ESTA LINHA USA VIRGULAS"
50 PRINT "H", "E", "L", "P"
60 END
```

A linha 30 fará com que apareça na tela a palavra HELP, enquanto na linha 50 aparecerá H E L P. Reporte-se ao quadro "A propósito..." para ver as variações entre as diferentes máquinas. A vírgula possui diversos empregos em BASIC, mas nos comandos PRINT tem o efeito de fazer os itens aparecerem espaçados na tela (ou na impressão em papel), comumente entre 8 e 16 espaços, dependendo da versão do BASIC. Se a instrução PRINT for utilizada sem vírgula ou ponto e vírgula, os itens serão impressos em linhas separadas.

Além de ilustrar o emprego do ponto e vírgula no BASIC, nosso programa de conversão de temperatura fez a revisão de diversas instruções abordadas nas duas primeiras partes do curso de programação BASIC. As linhas 30 e 50 associaram as variáveis L e H aos valores mais alto e mais baixo de temperaturas que desejamos converter. A linha 60 é a primeira parte do loop FOR-NEXT. Este loop parece ser diferente dos outros que encontramos até aqui, por utilizar letras ao invés de números. Na realidade, não há diferença. As letras que estamos utilizando aqui, L e H, são variáveis com valores numéricos correspondentes aos valores digitados no estágio do programa do INPUT L e INPUT H. Se, como anteriormente foi sugerido, você introduziu -5 e 10, a declaração FOR X = L TO H é, portanto, equivalente a FOR X = -5 TO 10.

A linha 80 diz, na realidade: PRINT o valor de X (que começa na temperatura mais baixa e aumenta 1 grau de cada vez até a temperatura mais alta) segui-



do diretamente na mesma linha (por isso utilizamos o ponto e vírgula) pelas palavras entre aspas, seguida outra vez diretamente (outro ponto e vírgula) pelo valor de F. Se você olhar atentamente para F, verá que é o valor, naquele momento, da temperatura em graus centígrados, convertida em Fahrenheit. A linha NEXT X assegura que continuemos as conversões até que o limite superior no loop FOR-NEXT tenha sido alcançado.

Antes de observarmos uma variação mais sofisticada da declaração PRINT, é importante que paremos um segundo para examinar a linha 70 do nosso programa de conversão de temperatura:

```
70 LET F = (X * 9 / 5) + 32
```

Esta linha associa um valor à variável F (que significa Fahrenheit). O programa, em primeiro lugar, obtém o valor de X (a temperatura em graus centígrados), multiplica-o por 9, divide-o por 5 e, depois, soma 32. Essa mesma fórmula seria apresentada em um livro comum de matemática do seguinte modo: $F = [(C \times 9) \div 5] + 32$. O BASIC usa * para a multiplicação, / para a divisão, + para a soma e - para a subtração.

Na matemática comum, e no BASIC também, a ordem de execução das operações aritméticas é importante. A multiplicação tem sempre prioridade, seguida pela divisão, seguida pela soma, seguida pela subtração. Se partes da expressão aritmética estão encerradas entre parênteses, devem ser realizadas em primeiro lugar. Se você deseja fazer antes da multiplicação uma soma, esta deverá estar encerrada entre parênteses. Por exemplo, se você quisesse saber o equivalente em dólares da soma de suas contas bancária e de poupança, deveria expressá-la em seu programa assim:

```
D = (C + P) / taxa do dólar
```

Se na conta corrente você tem Cr\$ 500.000,00 (C) e na poupança Cr\$ 700.000,00 (P), deverá somar em primeiro lugar os cruzeiros (C + P) e, depois, dividir pela taxa do dólar. Sem os parênteses, o valor de sua conta em poupança seria em primeiro lugar dividido pela taxa do dólar e, em seguida, o valor de sua conta corrente seria somado ao resultado — e não é isso que desejamos! Tenha sempre certeza de haver verificado as operações aritméticas, para que sejam calculadas na ordem correta.

O PRINT USING

A fim de verificar um último aspecto em nosso programa de conversão, experimente digitá-lo outra vez e acione o comando RUN. Introduza, digamos, -10, como o valor inferior de temperatura, e 10 para o valor superior. Como já vimos, a impressão em blocos da tela é muito confusa. Isso se deve aos pontos e vírgulas na linha 80, que reúnem todas as partes que estão sendo impressas, ao invés de imprimi-las em linhas separadas. O que seria bom, exceto pelo fato de que varia o espaço tomado pelos números, tanto os graus centígrados quanto os Fahrenheit. Isso tem como efeito o desalinhamento das colunas,

tornando desordenada a impressão em blocos.

Quase todas as versões do BASIC possuem uma forma de PRINT denominada PRINT USING. Esta permite que a aparência dos números impressos, ou palavras, seja “formatada”, ou harmonizada. Se você desejar imprimir o valor de X e souber antecipadamente que esse valor varia de, digamos, -99 a 99, os números podem ser impressos corretamente alinhados o utilizando PRINT USING “###”;X. Os três sinais de numeral permitem que três dígitos, ou dois dígitos precedidos pelo sinal de menos, sejam impressos. Se mais de três dígitos forem introduzidos, eles não serão impressos corretamente. Sendo necessário, pontos decimais podem ser incluídos na posição adequada dentro dos sinais de numeral. Por exemplo, a declaração pode tomar a forma PRINT USING “###.###”;X. Utilize um sinal de numeral para cada dígito. Todos os pontos decimais serão alinhados automaticamente.

Modifique o programa original alterando a linha 80 e adicionando as linhas 82, 84 e 86:

```
80 PRINT USING “###”;X;
82 PRINT “ EM CENTIGRADO E “;
84 PRINT USING “###.###”;F;
86 PRINT “ EM FAHRENHEIT”
```

LISTe o programa novamente e acione o comando RUN, para executá-lo. Todas as colunas devem agora estar alinhadas perfeitamente.

No próximo segmento do curso vamos descobrir como “guardar” programas, para que não precisem ser redigitados cada vez que se queira executá-los.

Exercícios

Experimente introduzir a “temperatura inferior” de -1.000. Por que o programa não funcionou desta vez? Como você modificaria a declaração PRINT USING na linha 80 para fazê-lo funcionar?

Altere a linha 84, de modo que apenas os números inteiros (sem frações decimais) sejam impressos.

Escreva um programa para converter uma determinada quantidade de valores em cruzeiros para dólares, fazendo uso da taxa de câmbio do dia.

A propósito...



Esta característica não se encontra nos modelos compatíveis com o Apple (por exemplo, Unitron), nem nos compatíveis com o Sinclair (TK83 e 85, e CP 200).



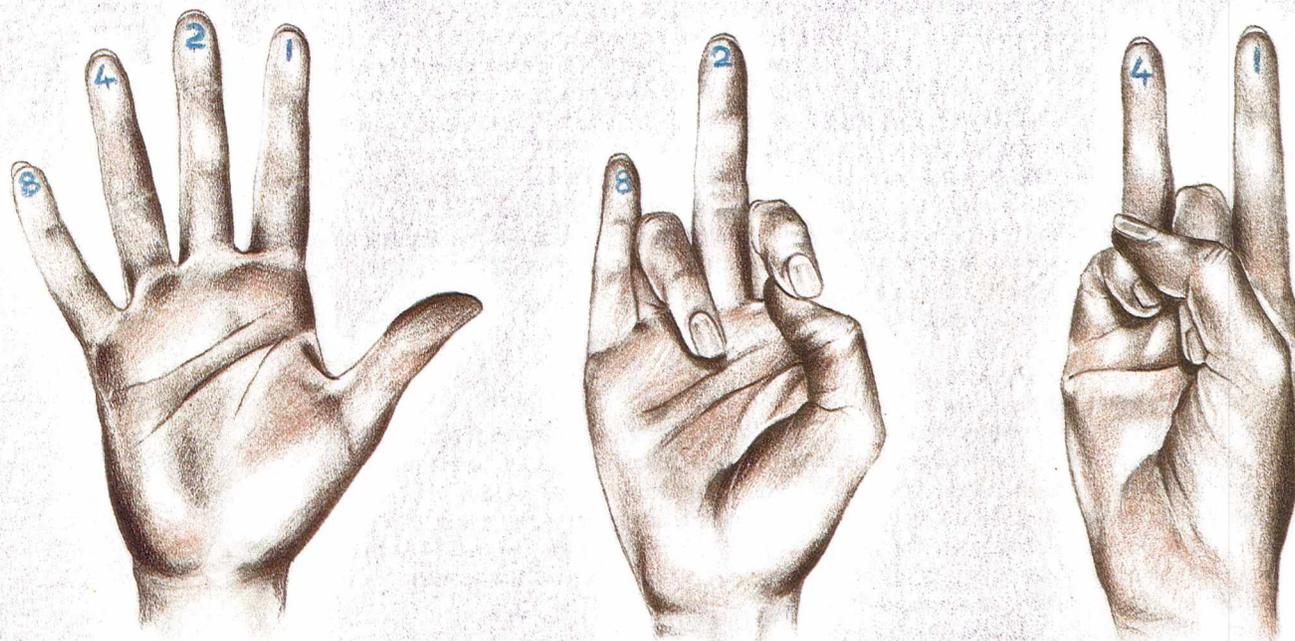
A vírgula entre campos de impressão separará os itens a serem impressos, com a inserção de um número de espaços, que varia conforme o tipo de computador utilizado. Nos computadores da linha Apple, a tela é dividida em três campos, dos quais dois possuem 16 espaços e um possui 8 espaços. Nos computadores da linha Sinclair (TK's, CP 200) e TRS-80 (CP 500), a tela é dividida em dois campos de 16 espaços.



Quando 1+1=10

Computadores são capazes de realizar seus incríveis cálculos utilizando apenas dois dígitos: 0 e 1.

Conversão ao sistema binário



O meio mais fácil de converter números binários pequenos em seus equivalentes decimais é imaginar que o valor de posição de cada coluna binária esteja escrito nos dedos da mão direita. Contudo que o número binário não tenha mais de quatro dígitos, tudo o que você precisa fazer é erguer o dedo correspondente ao respectivo dígito binário 1 e abaixar o dedo quando for um 0.

Ao erguer os dedos correspondentes para formar 1010, você obterá um 8 e

um 2; o resultado de sua soma é o número decimal 10. A terceira ilustração mostra como decodificar 0101: dá um 4 e um 1, e somando os dois obtém-se 5 na forma decimal. Use o método para calcular o equivalente decimal de 1110 e 0110. O método pode ainda ser ampliado, usando-se ambas as mãos para ilustrar números binários mais longos. Para fazê-lo, os dedos da mão esquerda (palma da mão virada para você) deverão ser identificados como 16, 32, 64 e 128, com o 16 no dedo mínimo e o 128 no indicador.

A maioria das pessoas confia tanto em nosso habitual sistema numérico decimal que nunca lhe ocorre a idéia de que outro possa ser utilizado.

Os romanos adotaram um sistema para representar números, valendo-se de letras do alfabeto. Assim, X correspondia a 10; L, a 50; C, a 100; D, a 500; M, a mil. O sistema romano funcionou bem como meio para gravar números simples, através da combinação destas letras. Não serviu, todavia, para fazer cálculos. Mesmo as adições em algarismos romanos são difíceis, por uma razão: não existe conceito para o “valor de posição”.

Veja os números 506 e 56. A única diferença aparente entre eles é o zero no meio. Este algarismo nos informa que no número 506 não há dezenas, mas apenas cinco centenas e seis unidades. Cada coluna ou posição em um número convencional tem um valor a ela associado. A coluna à direita do número é

a das unidades, a seguinte (movendo-se para a esquerda) é a das dezenas; a próxima é a coluna das centenas e assim por diante. O dígito usado em qualquer coluna significa quantos valores dela estão envolvidos.

E o que tudo isso tem a ver com computadores e o sistema binário? Computadores são máquinas eletrônicas que lidam facilmente com números, utilizando níveis de voltagem. Cinco volts representam o 1 e zero volt representa o 0. Assim, como você leu no artigo “Bits e bytes” (p.32), o 1 e o 0 são perfeitamente adequados a representar qualquer número, mesmo que seja grande.

Utilizando-se o sistema decimal (também conhecido como sistema de base 10), o número 506 é o meio conciso para representar o equivalente a quinhentos e seis elos de uma corrente, ou quinhentos e seis nós em uma corda. Na aritmética binária, o



mesmo número é representado por um deselegante 11111010.

Devido ao fato de que o sistema usado é binário, o “valor de posição” de cada dígito em cada coluna é diferente. Ao invés de aumentar um valor em múltiplos de 10, as colunas aumentam em múltiplos de 2.

A coluna à direita continua sendo a das unidades, mas, como existem apenas dois símbolos (0 e 1), faltam-nos dígitos assim que acrescentamos 1. No sistema decimal, somente nos faltam símbolos quando chegamos ao 9; desta maneira, utilizamos a coluna das dezenas e usamos 1 para indicar que agora temos um “lote” de dez.

O sistema binário opera exatamente da mesma forma. Em vez de agrupar em dezenas escrevendo 10 para dezenas, o sistema binário faz agrupamentos em dois, com o que o dígito binário 10 representa o número decimal 2.

Segue-se o número quinhentos e seis escrito de forma decimal e de forma binária, dando a perceber a similaridade dos dois sistemas:

Centenas	Dezenas	Unidades	
5	0	6	
= 5x100 +	0x10 +	6x1	(= 506)

256s	128s	64s	32s	16s	8s	4s	2s	1s	
1	1	1	1	1	1	0	1	0	

= 1x256 + 1x128 + 1x64 + 1x32 + 1x16 + 1x8 + 0x4 + 1x2 + 0x1 (= 506)

As regras de aritmética no sistema binário são exatamente as mesmas do já conhecido sistema decimal; a

única diferença é que nos faltam símbolos de contagem após o 1, ao invés de após o 9. Vamos tentar algumas somas para prová-lo. Os equivalentes decimais são demonstrados entre parênteses.

$$\begin{array}{r} (3) \quad 11 \\ + (5) + 101 \\ \hline (8) \quad 1000 \end{array}$$

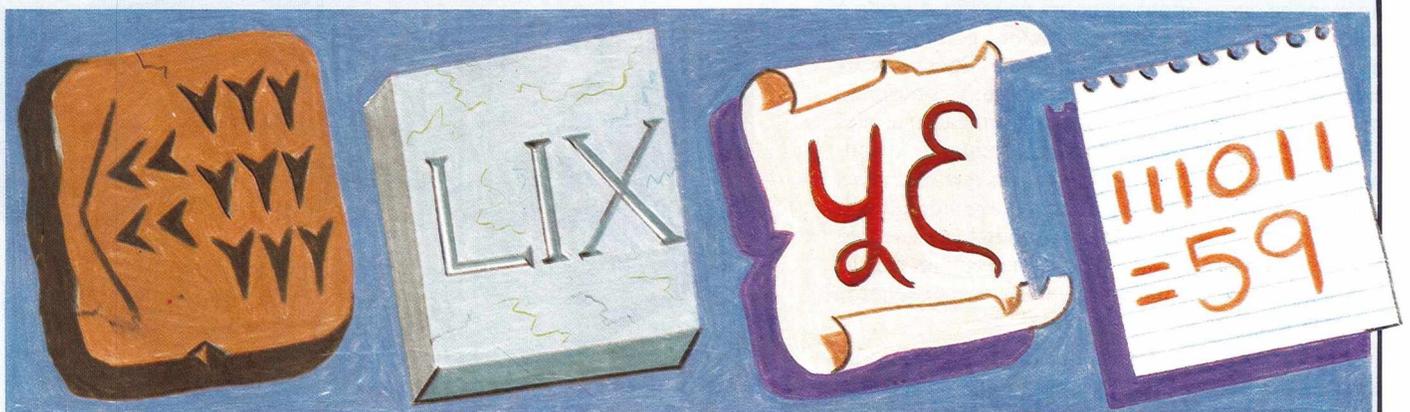
$$\begin{array}{r} (1 + 1 = 0 \text{ transporta } 1) \\ (1 \text{ (transportado)} + 1 = 0 \text{ transporta } 1) \\ (1 \text{ (transportado)} + 1 = 0 \text{ transporta } 1) \\ (1 \text{ (transportado)} + 0 = 1) \end{array}$$

No sistema binário, tal como acabamos de ver, ao adicionarmos 1 a 1 faltam-nos símbolos, uma vez que somente os números 0 e 1 são permitidos. Assim, dizemos: “um mais um é igual a zero e vai um”. Exatamente como, na adição decimal, quando adicionamos 1 a 9 não temos mais símbolos — não existem algarismos maiores que o 9 —, por esta razão dizemos: “nove mais um é igual a zero e vai um”. Eis aqui outra soma que demonstra esse princípio e mais duas para você mesmo tentar resolver:

(4)	100	(7)	111	(3)	11
+(6)	+110	+(2)	+10	+(12)	+1100
(10)	1010	(?)	?	(?)	?

Você nota que os números binários são bem mais longos que seus equivalentes decimais. Veja se pode somar 11010110 a 1101101. Lembre-se de manter as colunas à direita alinhadas exatamente como se você somasse um número decimal mais longo a um mais curto.

A história dos números



Babilônios

Romanos

Hindus

Binários

Os antigos babilônios tinham um sistema numérico avançado, baseado em 60 ao invés de 10. A representação do número 59 na escrita cuneiforme babilônica é ilustrada acima. O uso de 60 como número base tinha muitas vantagens e ainda hoje há vestígios desse sistema. Existem 60 segundos em um minuto e 60 minutos em uma hora e seis vezes 60 graus em um círculo — vestígios de um sistema matemático aperfeiçoado há 4.000 anos.

O sistema romano foi um considerável retrocesso. Letras eram usadas para representar números, mas sua posição não dava indicação de seu valor, tornando impossível mesmo a aritmética mais simples.

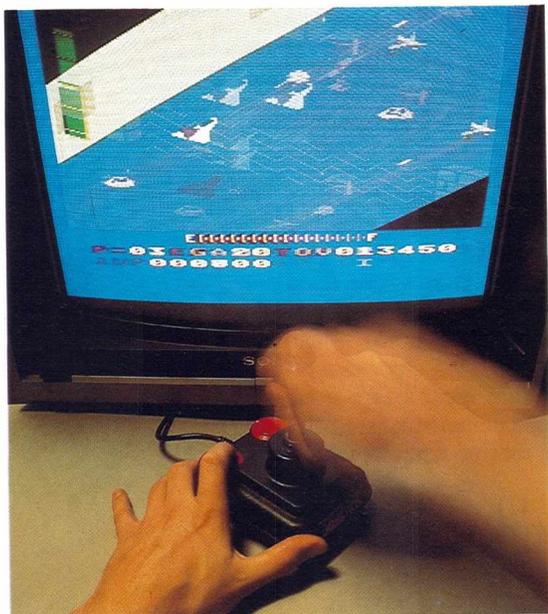
Os hindus utilizavam nove símbolos para os números de 1 a 9 e mais tarde

acrescentaram um símbolo para representar o 0. Sua contribuição vital foi o “valor de posição” — a idéia era de que a posição de um dígito em um número determinava o “valor” desse dígito. Dessa forma, 3 em 30 “vale” três dezenas. O sistema hindu foi adotado pelos árabes e gradualmente se espalhou pela Europa. Um dos mais ilustres matemáticos árabes era chamado Al Khwarizmi. A pronúncia latinizada de seu nome deu-nos o termo matemático algoritmo e o seu livro *Al-jabr wa'l Mugabalah* é lembrado na palavra álgebra.

Os computadores utilizam o sistema binário, pois números de qualquer tamanho são representados usando-se somente os números 0 e 1.

Ação rápida

Você pode tornar os jogos de computador mais emocionantes acrescentando alguns periféricos ao seu equipamento.



Joystick

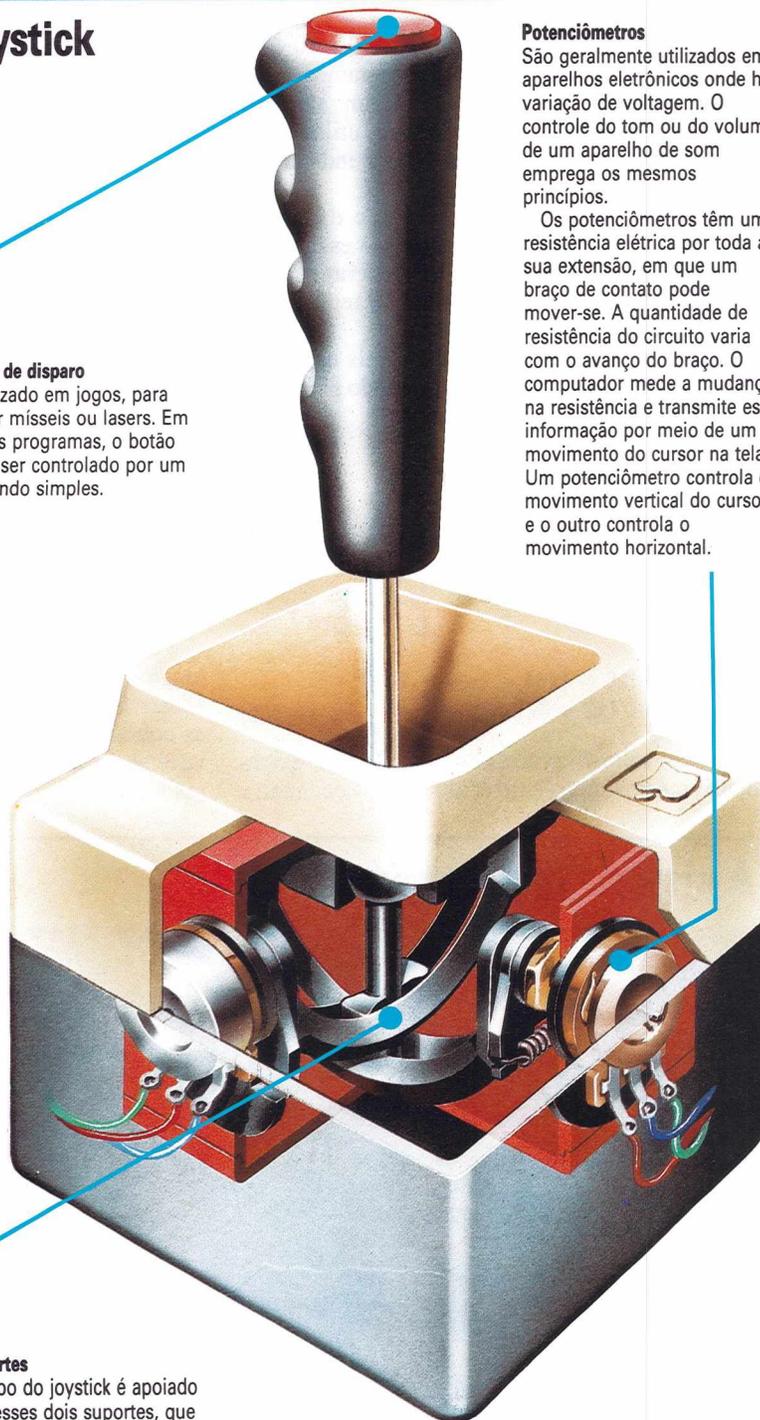
Botão de disparo

É utilizado em jogos, para lançar mísseis ou lasers. Em outros programas, o botão pode ser controlado por um comando simples.

Potenciômetros

São geralmente utilizados em aparelhos eletrônicos onde há variação de voltagem. O controle do tom ou do volume de um aparelho de som emprega os mesmos princípios.

Os potenciômetros têm uma resistência elétrica por toda a sua extensão, em que um braço de contato pode mover-se. A quantidade de resistência do circuito varia com o avanço do braço. O computador mede a mudança na resistência e transmite esta informação por meio de um movimento do cursor na tela. Um potenciômetro controla o movimento vertical do cursor e o outro controla o movimento horizontal.



Suportes

O cabo do joystick é apoiado por esses dois suportes, que se elevam formando ângulos retos. São ligados aos potenciômetros. Quando o cabo do joystick é movido, os braços deslizam nos potenciômetros e mudam a resistência elétrica.

De repente, um videogame leva você a pilotar uma espaçonave em território inimigo e disparar mísseis para destruir determinado alvo. Em vez de um teclado, detalhado demais para momentos de tanta tensão, o joystick é capaz de transferir o controle da espaçonave para suas próprias mãos, possibilitando ação mais rápida.

Esse tipo de controle foi desenvolvido com base no manche que o piloto manobra no avião. Ligado à parte de trás do microcomputador, geralmente é utilizado para jogos semelhantes aos fliperamas. A espaçonave, ou qualquer outro objeto controlado pelo joystick, move-se na mesma direção que ele. Quando o joystick é movido para a frente, a espaçonave sobe na tela. O aparelho possui quatro chaves elétricas dispostas de tal forma que, quando o joystick é movido, apenas um dos contatos se fecha. Cada chave envia sua própria mensagem para o computador: para cima, para baixo, para a esquerda ou para a direita.

Alguns deles são dotados ainda de um botão lateral de disparo de mísseis para ser operado com a mão que estiver desocupada. Em alguns modelos, no entanto, basta que se aperte um botão disparador com o polegar.

Alguns microcomputadores mais simples não possuem saídas para joystick. Nesse caso, torna-se necessário digitar nas teclas os movimentos desejados, ou adquirir uma interface, que estabelece o contato entre o joystick e o computador. Algumas empresas fabricam interfaces para esses equipamentos; mas, mesmo com a utilização de tal recurso, a programação dos jogos deve ser feita incluindo tanto o teclado como o joystick.

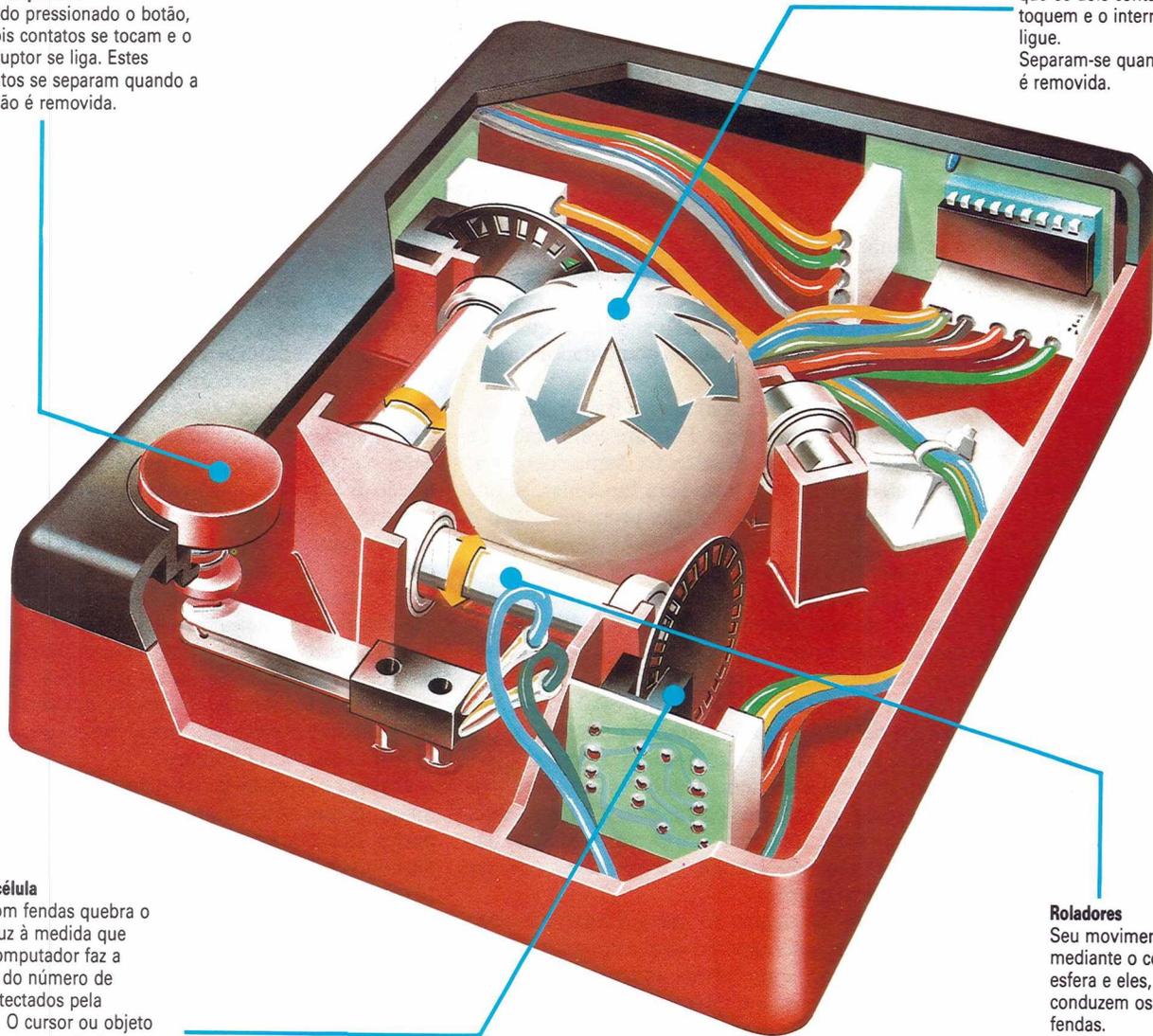


Botão disparador

Quando pressionado o botão, os dois contatos se tocam e o interruptor se liga. Estes contatos se separam quando a pressão é removida.

Esfera rotatória

Pressionada, a esfera faz com que os dois contatos se toquem e o interruptor se ligue. Separam-se quando a pressão é removida.



Luz e fotocélula

O disco com fendas quebra o fecho de luz à medida que gira e o computador faz a contagem do número de flashes detectados pela fotocélula. O cursor ou objeto movimenta-se na tela de acordo com esse número. A fim de controlar o movimento horizontal e vertical na tela, há dois grupos de discos e fotocélulas.

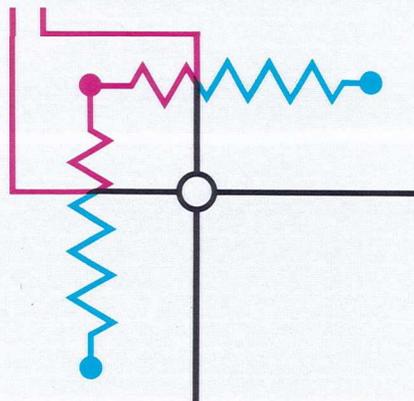
Roladores

Seu movimento ocorre mediante o contato com a esfera e eles, por sua vez, conduzem os discos com fendas.

Track ball

Suponha que você esteja guiando um cursor na tela, num labirinto, e precise avançá-lo e conduzi-lo em todas as curvas e mudanças de direção. O controlador tipo track ball foi projetado para esse tipo de problema e é dotado de uma esfera do tamanho de uma bola de bilhar que você rola na palma da mão. À medida que a bola rola, o objeto se move na mesma direção, dando-lhe controle total e imediato. Dentro do aparelho há duas rodas colocadas em ângulo reto e que estão em contato com a bola. Ao rolar a bola na palma da mão, uma roda capta a parte vertical do movimento e a outra capta a parte horizontal. O computador une os dois sinais, a fim de recriar o movimento.

Circuito do potenciômetro



A alavanca do joystick é ligada a duas resistências variáveis chamadas potenciômetros. Uma ligação mecânica move o ponto de contato (braço) ao longo de uma ou das duas resistências representadas na figura por um ziguezague. Assim, a posição do joystick estabelece a resistência elétrica dos dois potenciômetros. O computador checa as voltagens e calcula a posição do joystick. Encarrega-se também de converter essas informações em movimentos na tela.

Memória infalível

O computador armazena no seu interior um grande número de informações. Esses dados devem ser organizados cuidadosamente para que possam ser recuperados a qualquer momento.

Memória RAM

O chip RAM (abaixo) constitui um dos mais recentes avanços no que se refere à tecnologia de computadores. RAM (Random Access Memory) é uma das variedades de memória totalmente eletrônica, categoria que também abrange ROM (Read Only Memory). As fitas cassete e os discos magnéticos flexíveis são exemplos de outro gênero importante, a memória eletromagnética.

A memória RAM é fabricada com silício e utiliza processo fotográfico e gravação química, a fim de criar milhares de minúsculos transistores. Cada bit de memória requer pelo menos um transistor.

O tempo que se leva para "escrever" um simples bit em qualquer das 16.384 células de memória é de cerca de 200 nanossegundos, ou seja, 5 milionésimos de segundo.

No ser humano, a memória é um arquivo da mente; um lugar em que detalhes de experiências vividas são armazenados para uso posterior. Em computação, a palavra "memória" significa quase a mesma coisa; mas a memória do computador limita-se a determinadas funções.

Para um ser humano, a memória fraca pode causar momentos embaraçosos. Para um computador, a falta de memória é algo desastroso. Sem a memória, o computador não teria base para funcionar, já que faz uso dela para armazenar os programas que o dirigem.

Em ambos os casos, o termo memória refere-se a duas coisas: armazenagem e lembrança. Armazenar informações sem poder retirá-las seria inútil; e tentar lembrar-se de informações que não foram armazenadas obviamente não tem sentido.

Ambas as memórias são similares também sob outro aspecto. A memória humana parece atuar a curto e a longo prazo. Um homem atravessando a rua, por exemplo: certamente ele "se lembrará" de esperar que o último carro passe. Entretanto, quando já estiver do outro lado da rua, o homem não se lembrará mais do veículo. Sua memória é, neste caso, de curto prazo.

No entanto, se dentro do mesmo carro ele visse dois homens mascarados no banco traseiro, ameaçando sua esposa, com certeza se lembraria do modelo e da cor do carro e talvez até da placa do veículo. A memória é, neste caso, de longo prazo.

O computador possui esses dois tipos de memória. A memória a longo prazo ou permanente

contém programas e informações que o usuário deseja manter. São armazenados na forma de gravações magnéticas em fitas cassete, discos flexíveis ou cartuchos ROM.

O tipo de memória a curto prazo ou transitória é chamado RAM, um chip que fica dentro do próprio computador. Seu uso restringe-se ao período em que o computador está em funcionamento. Se o computador não receber energia, mesmo que seja por uma fração de segundo, todo o conteúdo memorizado desaparece instantaneamente.

Entretanto, a analogia feita com a memória humana não é inteiramente exata. Para que o computador trabalhe, os programas e dados devem ser transferidos da memória a longo prazo para a memória a curto prazo; só assim o computador poderá ter acesso instantâneo às informações. E o modo como são armazenados e transferidos os dados num computador é completamente diferente do mecanismo da memória humana.

É ainda um mistério o mecanismo da memória humana, já que as lembranças não são depositadas em partes identificáveis do cérebro. Não precisamos descobrir a localização de determinada informação para trazê-la à lembrança. E quando uma lembrança não é mais necessária não a depositamos novamente numa pequena divisão de nosso cérebro.

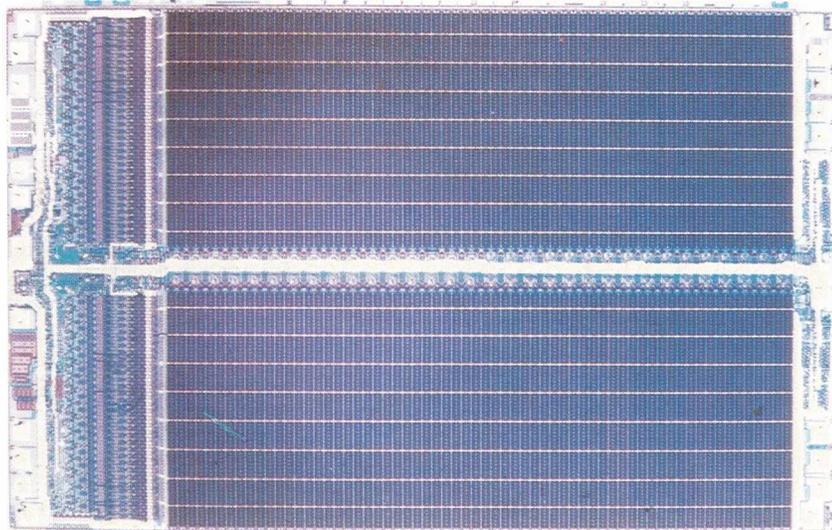
Caos organizado

Na memória do computador, a localização de cada item é vital. O computador deve encontrar um byte de informação, saber se ela é parte do programa ou dos dados do programa e saber, finalmente, onde depositar a informação.

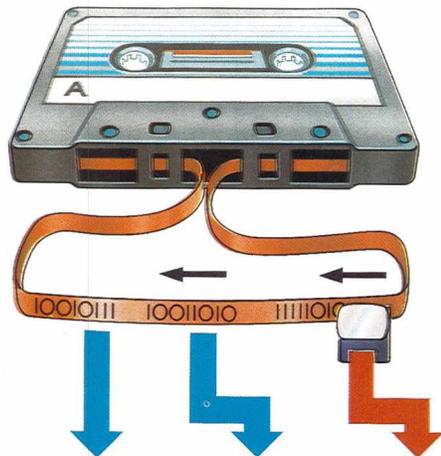
A memória humana se parece com uma caixa desorganizada, abarrotada de informações. As informações são depositadas aparentemente ao acaso e se confundem com outras lembranças, à medida que novas experiências ocorrem. Mesmo assim, o cérebro é capaz de conseguir a informação correta, no momento certo.

A memória do computador é como um enorme pombal em que cada parte é totalmente separada da outra. Tudo é muito organizado: cada parte do pombal tem um número (conhecido como seu "endereço") e contém apenas um byte, nem mais nem menos. O computador encontra a informação pelo número do compartimento do pombal e não pelo que nele está armazenado.

O computador não tem inteligência, portanto não



Memória

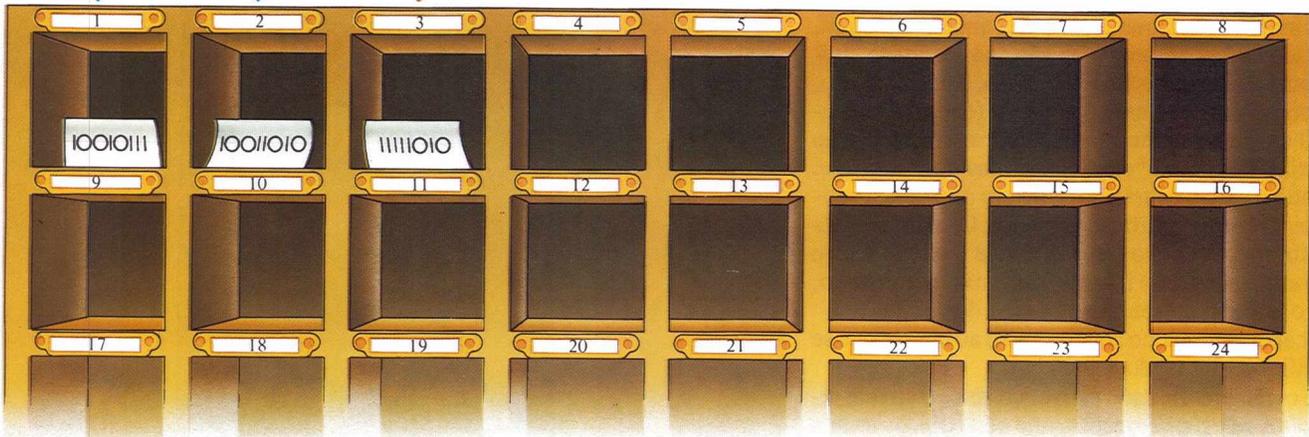


Os programas registrados em fita são armazenados em seqüência, cada bit de cada byte registrado em ordem. Quando a fita é utilizada novamente, o computador faz a leitura de cada bit, mas armazena-os em grupos de oito (bytes) em cada célula de memória. O primeiro byte da fita é colocado na primeira célula disponível; o segundo byte, na segunda, e assim por diante. Quando o computador for executar o programa, só precisa saber o endereço de início. O computador transfere o conteúdo de cada célula de memória para a CPU (Unidade Central de Processamento) e estes bytes fazem com que se "realizem" as ações requisitadas pelo programa.

Parte da memória do computador é ocupada por programas de "controle interno", responsáveis pelos aspectos fundamentais de sua operação. Eles verificam qual das teclas foi pressionada, expõem caracteres na tela etc. Esse software interno pode também incluir a linguagem BASIC de programação. Esses programas ocupam lugar na memória e deixam

menos espaço para a armazenagem de programas feitos pelo usuário ou adquiridos comercialmente. Algumas versões BASIC, por exemplo, ocupam 16 Kbytes de memória. Quando um programa é carregado de fita cassete, a primeira locação de memória disponível não será a primeira locação em RAM. É uma das tarefas dos programas de controle interno saber a primeira locação de memória disponível para o usuário. Depois de o programa ter sido carregado na memória RAM do computador, os programas de controle dizem: "Comece olhando para a locação x de memória, depois continue examinando cada locação sucessiva de memória, registrando o conteúdo da locação na CPU". A ordem original em que o programa foi registrado no teclado pelo programador é a mesma ordem registrada na fita.

Quando o programa passa da fita para a memória do computador, é colocado nas células de memória na mesma ordem em que estava, como se tivesse sido digitado no teclado.



consegue organizar sua memória sozinho. O que leva o computador a armazenar dados é o fato de que alguém preencheu a parte certa do pombal, na hora e na ordem exatas. E como isso acontece com o sistema de computador típico?

Quando você liga seu microcomputador, em geral uma mensagem aparece na tela para indicar que ele está funcionando. Na maioria das vezes, ele também informa que você pode começar a escrever o programa. A mensagem e as condições que lhe permitem a programação estão armazenadas na parte interior da memória do computador e devem ter sido armazenadas na memória a longo prazo (em geral, a Read Only Memory ou chip ROM).

Esta parte da memória do computador contém programas que verificam se as teclas foram pressionadas, imprimem os caracteres (letras, números, símbolos) na tela e executam outras tarefas essenciais a seu funcionamento. Contém ainda um programa especial que traduz comandos geralmente escritos em BASIC na mais simples linguagem binária de uns e zeros, compreendida pelo computador.

Quando o computador é ligado, a mensagem na tela diz: "x bytes livres" — onde x é algo como 15, 797, ou qualquer outro número estranho. Isso indica a quantidade de compartimentos do pombal que está disponível. À medida que as teclas são pressionadas, os compartimentos livres vão sendo preenchi-

dos; e aqui chegamos a outro fator importante em relação à memória do computador: a ordem de armazenagem das informações.

A pressão de uma tecla envia um byte que representa a letra pressionada (ver p. 3), para que seja armazenado na memória. Pressionando-se o D, por exemplo, essa letra ocupa espaço em um compartimento de memória na forma binária.

Mas que compartimento do pombal será ocupado pela letra D? Ela vai para o primeiro espaço vazio da memória não permanente do computador. Se pensarmos na seqüência de compartimentos de um pombal, a letra D iria para o primeiro espaço à esquerda.

Pressione outra tecla, digamos I, e a configuração apropriada de bits irá para o segundo espaço vazio, à direita do D. Pressione uma terceira tecla, o A, e irá para o terceiro espaço vazio, ao lado do I. Olhando para o pombal completo, veremos os códigos da palavra DIA aparecerem na primeira fila.

O computador tem um contador interno para avaliar que parte do pombal foi atingida. Ele sabe onde começar porque um programa interno de controle avisa onde começa a próxima área livre de memória. À medida que cada letra é armazenada, o contador é acrescido de um, a fim de dizer qual é o compartimento seguinte do pombal para a próxima letra digitada.



Tudo sob controle

O caixa automático de seu banco pode, em minutos, resolver muitos de seus problemas, mesmo num domingo à tarde.

Clientes de banco têm agora acesso ao caixa a qualquer hora do dia ou da noite e em quase todo lugar, graças ao microcomputador. Para isso, precisam apenas de um pequeno cartão de plástico que contém informações codificadas sobre sua conta. Quando o "bolso está vazio" e há necessidade imediata de dinheiro, é suficiente saber a localização da filial mais próxima do banco que dispõe de caixa automático.

Esses caixas automáticos são, na realidade, microcomputadores disfarçados. Para obter dinheiro, o cliente insere seu cartão em uma fenda da máquina. O cartão tem uma tira magnética semelhante ao material magnético empregado nas fitas cassete. Quando o cartão é inserido, a leitora magnética verifica os números codificados na tira. Os números demonstram se o cartão é verdadeiro e também dizem ao computador, no interior do caixa automático, qual o número da senha ou código do cliente.

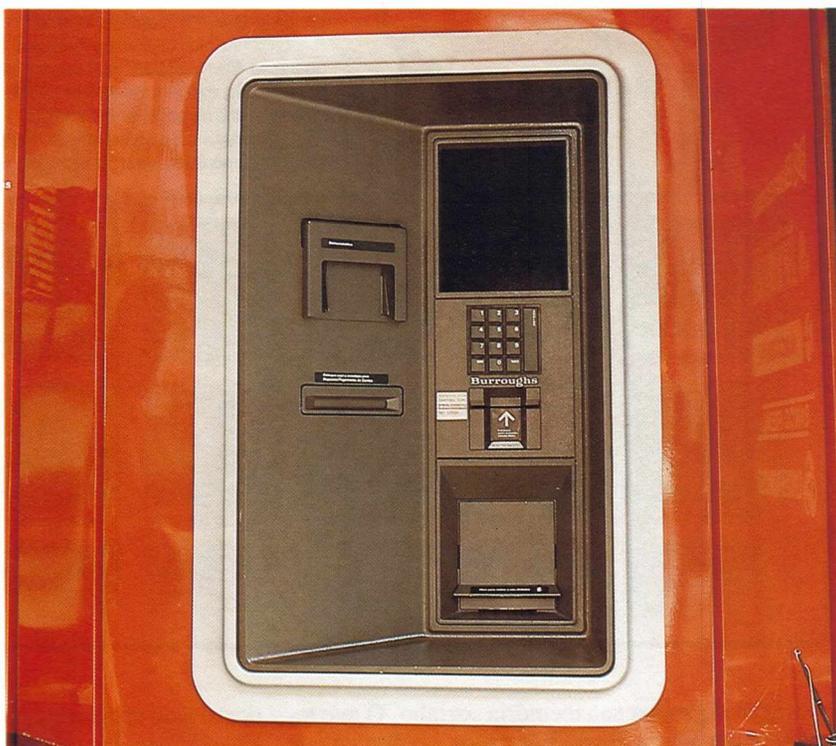
O caixa automático solicita então ao cliente que se identifique, digitando seu código.

Se o número digitado estiver correto, o cliente recebe uma seleção de opções, que são exibidas na tela do caixa automático. Elas geralmente incluem saque em dinheiro, consulta ao saldo da conta ou da poupança, depósitos e pagamentos. Se o código digitado não estiver correto, o sistema solicitará ao cliente que realize novamente a operação. Desse modo, o acesso às informações é bloqueado a quem não conheça a senha adequada.

Se desejar sacar dinheiro, o cliente pressiona o botão adequado e, em seguida, digita o total da quantia necessária. Nesta etapa, os detalhes de processamento no interior do sistema são diferentes. Um código na tira magnética diz ao caixa automático se ele precisa entrar em contato com o computador central do banco ou se a solicitação de dinheiro pode ser liberada no local.

De qualquer modo, quando a quantia a ser sacada não for prefixada (cartão SOS, por exemplo), o computador deverá checar se está disponível na conta do cliente um total igual ou superior ao solicitado. Se não estiver disponível, a operação será interrompida. Pagamentos de contas diversas também poderão ser efetuados nesses caixas automáticos, bem como transferências, consultas e movimentações de cadernetas de poupança.

Se o cliente desejar fazer uma retirada em dinheiro, o caixa automático comunica-se diretamente com o computador central do banco, o que significa que uma conexão se estabelece por linhas telefônicas especiais. Dessa forma, o microcomputador do caixa recebe as informações sobre a quantia de dinheiro existente no saldo da conta corrente do cli-



ente que está efetuando a operação.

Sabendo que há saldo suficiente, o computador central instruirá o microcomputador no caixa para liberar a quantia desejada. Uma máquina de contagem de dinheiro, no caixa automático, conta as notas e as empurra por uma fenda. Se o caixa está funcionando "on-line", o computador central é informado sobre a quantia de dinheiro retirada e a conta corrente do cliente é devidamente atualizada.

O caixa automático foi introduzido para fornecer serviços bancários fora do horário comercial. Durante o dia ou à noite, o cliente retira dinheiro, solicita o saldo da poupança, verifica o saldo de sua conta ou um extrato detalhado. O cliente recebe um cartão magnético que é associado a uma senha.

Problemas com "bilhete amarelo"

Os cartões magnéticos são usados no metrô de Londres. Há poucos anos, um amplo sistema foi planejado para permitir que os passageiros entrassem e saíssem por portões automáticos, valendo-se de um bilhete amarelo magnetizado. Infelizmente, as máquinas de saída rejeitaram tantas vezes os bilhetes, irritando os passageiros, que foi preciso abandonar o sistema.





Processador portátil

O processamento de palavras está se tornando rapidamente uma das aplicações mais comuns do microcomputador. Novos equipamentos são projetados com características especialmente dirigidas para essa função, entre os quais as telas com 80 colunas (para apresentar as linhas do texto em sua completa extensão), a incorporação de unidades de disco (em alguns casos, um processador de palavras em disco vem incluído no preço) e teclas com funções programáveis, usadas para trabalhar com o texto. Alguns equipamentos, como o da foto, são portáteis — ideais, portanto, para jornalistas e executivos.

O texto perfeito

Com um software adequado, seu micro se transforma em um processador de palavras em condições de revisar, organizar e armazenar textos — e mesmo corrigir erros ortográficos.

O processamento de palavras é uma das tarefas mais úteis realizadas pelo microcomputador. Mas a expressão “processamento de palavras” não deixa claro para a maioria das pessoas do que se trata exatamente. “Como é possível processar palavras?” Costuma ser esta a reação dos que ouvem falar nisso pela primeira vez.

Até recentemente, vistosos anúncios ofereciam “processadoras de palavras para seu escritório”. Mas deixavam de informar que o caro equipamento em questão consistia simplesmente em um microcomputador adaptado para rodar programas processadores de palavras. Processadores de palavras são menos versáteis que o microcomputador comum porque executam apenas uma atividade.

Talvez o tipo de programa que a expressão “processamento de palavras” indica fosse mais corretamente designado por “datilografia assessorada por computador”. Pelo acréscimo de uma impressora, a maioria dos microcomputadores pode rodar programas de processamento de palavras ou de edição de textos. Assim, basta que o usuário do computador experimente o processamento de palavras para perceber sua grande utilidade.

Quando usado como processador de palavras, o computador as apresenta na tela, à medida que são datilografadas no teclado, assim como a máquina de escrever as imprime em papel. Os computadores maiores podem apresentar 80 caracteres em linha na tela, representando a “página”. Em computadores

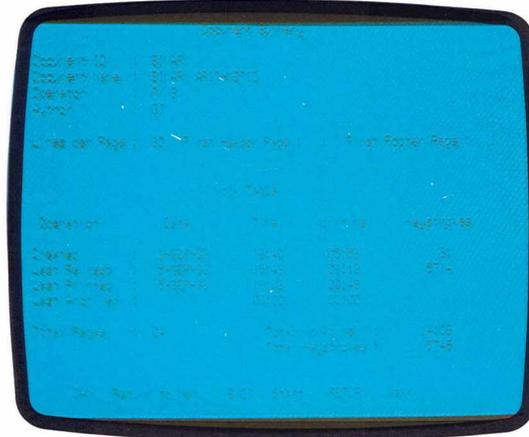


menores, o usuário precisa ser mais paciente. Deve se habituar a uma tela de reduzidas dimensões, e no caso de alguns modelos não terá letras minúsculas à disposição. O usuário também tem de se lembrar que um equipamento menor armazena apenas uma extensão limitada de texto.

O programa proporciona vários recursos sofisticados. Todos os programas processadores de palavras detectam o final da linha à medida que este se aproxima e automaticamente transferem a última palavra por inteiro para o início da linha seguinte. Isso significa que o usuário não precisa ficar atento à mudança de linha, ao atingir a margem: datilografa o texto em seqüência ininterrupta, enquanto o programa cria, quando necessário, uma nova linha. Todavia, o início de parágrafo tem de ser indicado pressionando-se a tecla RETURN.

Em máquinas de escrever convencionais, a correção de erros é feita por procedimentos mecânicos, usando-se borracha ou encobrimdo o erro com um corretor e reescrevendo o trecho incorreto. O resultado não é de boa qualidade. Se houver duas ou mais correções, as únicas opções serão enviar uma carta com má apresentação ou refazê-la. Com o processamento de palavras, o problema está solucionado. O cursor luminoso indicará sua posição na tela a cada momento. Você pode movê-lo no texto já feito, até atingir o ponto em que foi escrita a letra ou palavra errada e então eliminar o erro e datilografar corretamente.

Tendo percebido a capacidade de "correção" do processador de palavras, os usuários serão motiva-

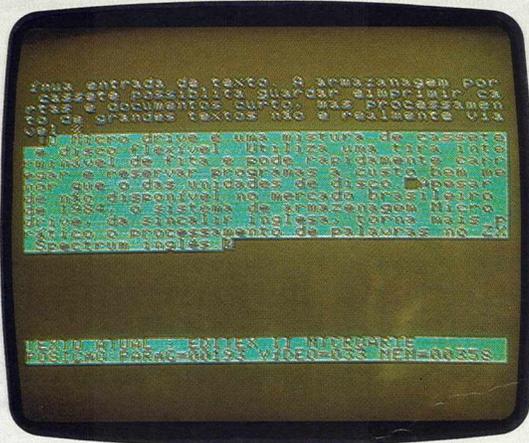
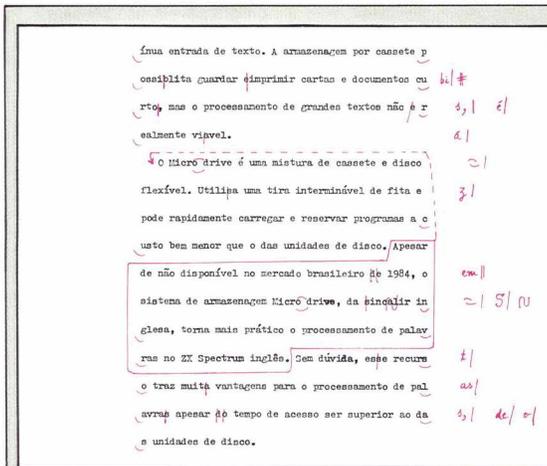


Vendo o índice

Na foto, o índice de um sofisticado pacote de processamento de palavras. O índice surgirá na tela imediatamente após a inserção do software e indica as várias funções existentes no processador de palavras. Como exemplo poderíamos citar: tabulação e posicionamento de margem, espaçamento de linhas, contagem do número de palavras no documento, reordenação dos parágrafos e elaboração de índices.

texto remanescente se reestruture, recuperando a aparência perfeita da página. Jornalistas e escritores profissionais que já usam processadores de palavras confirmam a melhora de seu trabalho, tanto em qualidade quanto em quantidade.

Mesmo os menores microcomputadores proporcionam certa capacidade de processamento de palavras. O TK83 e o CP 200 funcionam com um simples programa de edição ou revisão de textos, que permite ao usuário escrever uma carta ou documento na tela e depois fazer as correções. Edição de texto é a expressão geralmente usada para programas limitados de processamento de palavras que trabalham com textos de uma ou duas páginas, mas não podem lidar com documentos mais extensos. A pequena ca-



Revisão eletrônica

A principal vantagem de um processador de palavras controlado por computador é a surpreendente flexibilidade e velocidade. Enquanto um texto copiado em máquina de escrever requer revisão e correção trabalhosa, o processador de palavras faz isso em velocidade eletrônica. E todas as operações podem ser realizadas enquanto o texto ainda está

na tela. Sistemas sofisticados fazem pesquisas no texto, substituem palavras, trocam linhas de posição, efetuam correções no texto automaticamente e corrigem erros de ortografia e gramática. Compare as "confusas" correções no texto datilografado com a impecável revisão realizada no processador de palavras.

dos a dar mais atenção ao preparo do texto. Por exemplo, é possível usar o comando INSERT para acrescentar uma palavra, uma sentença inteira ou mesmo um parágrafo, tão facilmente como se se tratasse de uma única letra. Isso incentiva o usuário a reexaminar o que foi escrito na carta ou no documento. A instrução para eliminar trechos do texto é igualmente fácil. Um comando faz com que palavras e letras simplesmente desapareçam da tela e o

pacidade de RAM do TK83 e de computadores de tamanho semelhante limita muito a extensão do texto a ser composto e elaborado na tela.

Um problema com o TK83 é o teclado tipo "touch", que impede a datilografia rápida. Embora o teclado do TK85 seja muito melhor nesse aspecto, ainda não é do tipo mecânico ao qual as datilógrafas estão acostumadas. Se você pensa em comprar um microcomputador para uso em processamento de



palavras, deve examinar o teclado do equipamento, já que tem influência considerável na velocidade e comodidade de execução do trabalho (ver p. 36).

Após a aquisição da impressora, o principal problema quanto ao hardware vem a ser a existência de capacidade suficiente de memória para armazenar o texto. É possível utilizar um gravador cassete para a armazenagem. Mas esse tipo de equipamento limita a extensão do texto que pode ser escrito, pois a memória rapidamente se esgota. Isso não acontece com discos flexíveis, que vão automaticamente extraíndo texto da memória, deixando-a livre para contínua entrada de texto. A armazenagem por cassete possibilita guardar e imprimir cartas e documentos curtos, mas o processamento de grandes textos não é realmente viável.

Apesar de não disponível no mercado brasileiro em 1984, o sistema de armazenagem Microdrive, da Sinclair inglesa, torna mais prático o processamento de palavras no ZX Spectrum inglês. O Microdrive é uma mistura de cassete e disco flexível. Utiliza uma tira interminável de fita e pode rapidamente carregar e reservar programas a custo bem menor que o das unidades de disco. Sem dúvida, este recurso traz muitas vantagens para o processamento de palavras, apesar do tempo de acesso ser superior ao das unidades de disco.

Eficiência e economia

O processamento de palavras é um método eficiente porque separa o ato de composição do ato de impressão. Tanto a escrita manual como a datilografia exigem que a palavra seja efetivamente escrita ao mesmo tempo que ocorre o processo de pensa-

mento. Pelo processamento de palavras, nenhuma palavra aparece no papel até que a composição na tela alcance a forma desejada. Mas o surgimento de palavras no papel exige uma impressora — e uma impressora de baixo custo, adequada à emissão de listagens de programa, dificilmente poderá produzir cópias de boa qualidade (ver p. 74).

Os programas processadores de palavra estão agora disponíveis também em chips que podem ser conectados ao painel de circuitos do computador. São muito úteis na ausência de uma unidade de disco. Proporcionam a vantagem de rápido carregamento do programa, que se torna imediatamente disponível para uso. Se a memória RAM do computador tiver capacidade suficiente (32 Kbytes ou acima disso), você poderá elaborar documentos de até 5.000 palavras e fazer sucessivas revisões até chegar à forma satisfatória. Se quiser arquivar o texto revisado após a impressão, será necessário guardá-lo em fita cassete, pois o chip de processamento de palavras não pode armazenar o texto que você cria.

Alguns sofisticados processadores de palavras executam funções suplementares muito úteis; o dicionário automático, ou conferidor de ortografia, é um dos mais conhecidos. Para beneficiar-se desta comodidade, você necessitará de uma unidade de disco. O processador verifica as palavras escritas em um documento e as compara com aquelas armazenadas em seu arquivo, indica as que não reconhece e chama a atenção do usuário, que as corrigirá, se for o caso.

À medida que o uso do processador de palavras se amplia, é provável que em pouco tempo seja considerado um instrumento essencial tanto para os serviços de escritório quanto para os de correspondência.

Aplicações básicas de um sistema de processamento de texto

CORRESPONDÊNCIAS SEMI-STANDARD	Cartas de transferência Propostas Certificados Convocações Cartas de cobrança etc.	Caracterizam-se por textos repetitivos (standard), onde deverão ser inseridos dados variáveis
CORRESPONDÊNCIAS COMPLEXAS	Contratos Fórmulas jurídicas Descrição de produtos Ofertas	Obtidas mediante a composição de vários parágrafos standard
CORRESPONDÊNCIAS LONGAS	Manuais Apostilas Relações Editais Relatórios Atas etc.	Comportam a datilografia de várias páginas, que devem ser corrigidas e logo após impressas
CORRESPONDÊNCIAS PERSONALIZADAS	Mala direta Circulares Cartas promocionais Cartas de aniversário Cartas convites etc.	Correspondências que deverão ser enviadas a uma grande lista de endereços
CORRESPONDÊNCIAS VARIÁVEIS	Memorandos Pró-memórias Requisições Comunicados Solicitações	Correspondências cujo teor será sempre diferente

Perguntas e respostas

Aqui estão algumas dúvidas sobre o mundo da computação, que nem sempre são respondidas nos livros e revistas.

O que é a “quinta geração” de computadores e o que foram as primeiras quatro gerações?

A quinta geração é o estágio de desenvolvimento dos computadores em cuja direção os engenheiros e programadores de computação estão trabalhando. Estes computadores não terão teclados nem precisarão ser programados em linguagem de computador, como BASIC e PASCAL. Ao contrário, poderemos conversar com os computadores e eles nos responderão na linguagem desejada. Esses computadores provavelmente serão capazes de desenvolver seus próprios programas, solucionando problemas que lhes apresentarmos.

A primeira geração de computadores foram os primeiros equipamentos para cálculo completamente eletrônicos, inventados logo após a Segunda Guerra Mundial. A segunda geração trabalhava basicamente de acordo com os mesmos princípios, só que com transistores ao invés de válvulas e relés, tornando os computadores menores e mais baratos. A terceira geração usava circuitos integrados (circuitos eletrônicos montados em silício). Esses circuitos integrados foram a mais antiga forma de microchip e marcaram o início da redução no preço dos computadores. Mas esses computadores ainda eram caros demais para um escritório ou casa comuns.

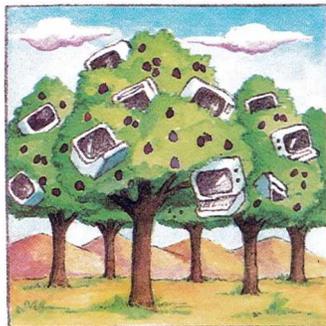
A quarta geração representa a tecnologia atual. Os computadores utilizam circuitos LSI (Large Scale Integration — Integração em

Grande Escala). O desenvolvimento desses microchips colocou os computadores nos limites de orçamento da maioria das pessoas.



O que é o Vale do Silício?

Vale do Silício é o nome dado a uma região que se estende das proximidades de San José ao sul de San Francisco, nos Estados Unidos, onde estão as sedes ou departamentos de pesquisa da maioria das grandes companhias americanas de computação e microeletrônica. A razão pela qual tanto conhecimento técnico se reúne em uma só área diminuta é puramente histórica — não há recursos naturais propícios à manufatura dos microchips! Até há vinte anos, a área era conhecida apenas pela produção de frutas.



Será realmente necessária aquela confusão de fios na parte posterior de alguns computadores?

A maioria dos microcomputadores de hoje é projetada com tanto cuidado com a aparência externa como com a parte eletrônica. Fios emaranhados geralmente estão escondidos. Contudo, para alguns computadores de pesquisa, mais aperfeiçoados, os fios soltos são importantes. A eletricidade move-se à velocidade da luz, mas ainda leva algum tempo para seguir pelo fio. Esses computadores de pesquisa trabalham tão rapidamente que os dados têm de chegar ao local certo e na hora precisa. O comprimento dos fios é calculado com exatidão, de modo a assegurar que a cronometragem seja perfeita.



É freqüente o anúncio comercial de computadores com um microprocessador Z80 ou 6502. Qual o significado desses números?

Por si mesmos, esses números nada significam — 6502 e Z80 são referências para identificação ou nomes para chips microprocessadores. Todos os computadores com base no mesmo microprocessador compreendem a mesma série de instruções fundamentais (chamada “linguagem de máquina”), a partir da qual os programas são elaborados. Entretanto, os programas são geralmente desenvolvidos

pelos usuários através de uma linguagem de alto nível, como BASIC, e depois traduzidos para um código de máquina pelo computador. Assim, a menos que você queira especificamente desenvolver programas já em linguagem de máquina, não fará muita diferença a espécie de microprocessador que seu computador possui.

Apesar de alguns tipos de microprocessador operarem a velocidade maior do que outros, o espaço de tempo observado em uma aplicação padrão depende muito mais da forma na qual o software foi escrito.



Como os computadores poderiam auxiliar no trabalho da polícia?

No Brasil, os computadores estão começando a ser utilizados nesse tipo de atividade. Em outros países, já há alguns anos, o computador tem sido poderoso auxiliar da polícia. Grandes arquivos catalogam dados de todas as pessoas com antecedentes criminais. Nesses países utilizam-se microcomputadores nos distritos policiais para que as viaturas sejam mobilizadas nas emergências com a maior rapidez. Alguns carros da polícia são equipados com terminais de computador e através deles têm acesso a informações do computador central.



Micros em movimento

Não será mais necessário consultar mapas e roteiros, nem se preocupar com o tanque de combustível: o carro do futuro levará você com segurança e economia ao seu destino.



Painel dinâmico

O painel do seu próximo carro poderá ser semelhante ao mostrado acima. Os ponteiros e mostradores dos painéis convencionais foram substituídos por um terminal de computador. Através da emissão de luz dos diodos é

possível controlar a velocidade, a quantidade de gasolina e a temperatura com muito mais eficiência e precisão. Painéis desse tipo já estão sendo utilizados em modelos como o Austin Maestro.

Em um carro pode-se ver a importância prática da tecnologia do computador. Em muitos países já existem à venda veículos com dispositivos que indicam a necessidade de assistência mecânica e avisam o momento em que é preciso ir à oficina.

O carro tem sensores ligados a vários pontos do motor, que fornecem pormenores — tais como quilometragem e temperatura — a um microcomputador, que avalia as condições em que o veículo foi conduzido. O motorista fica sabendo quando é necessário assistência mecânica porque o minúsculo computador aciona uma série de luzes (vermelhas, amarelas e verdes), no painel de instrumentos. Quando as luzes verdes se apagam e aparecem as vermelhas, é sinal de que o carro precisa de atendimento mecânico.

Há carros produzidos na Europa e no Japão que falam com o motorista, chamando-lhe a atenção para o uso do cinto de segurança ou para algum problema iminente com o motor — o superaquecimento ou os baixos níveis de água ou óleo, por exemplo. O carro pode fazê-lo porque possui um sintetizador de voz, equipamento no qual as características da voz humana estão programadas digitalmente. É o que se chama som digital e tem esse nome porque as ondas sonoras são transformadas em números binários, compreendidos pelo computador.

Quando os sensores do motor detectam algum problema, o computador ativa o sintetizador de voz; este transforma os dados digitais do computador em voz humana, por meio de um alto-falante.

Uma viagem confortável

Um computador pode também trazer aperfeiçoamentos ao carro pelo controle da suspensão. Os car-

ros Lotus funcionam por um processo chamado “suspensão ativa” — técnica que utiliza um computador para ajustar a resistência e flexibilidade dos amortecedores várias vezes por segundo; desse modo o carro se mantém estável na estrada, quer esteja lotado com passageiros e bagagens, ou leve apenas o motorista.

Geralmente, os carros esporte têm suspensão dura, para se manter estáveis na estrada. O inconveniente disso é o desconforto, com os passageiros sentindo solavancos. Mas, se o carro for equipado com suspensão macia para proporcionar viagem agradável, não poderá fazer as curvas tão bem, pois terá menor estabilidade. A suspensão controlada por computador terá a vantagem das duas situações.

A Honda e a Toyota estão desenvolvendo computadores também para controle do trajeto. Eles informam ao motorista a direção a seguir — e podem fazê-lo medindo a velocidade do carro, a direção e a distância percorrida e estabelecendo uma comparação com o mapa da rota correta, guardado em sua memória. O motorista decide se deve virar à direita, à esquerda ou seguir em frente, de acordo com as indicações dadas no painel de controle.

Um dos computadores mais comuns para uso em carro mede o consumo de combustível e calcula o tempo provável de viagem. O motorista tem conhecimento de quanto combustível está sendo usado em determinado momento e é igualmente informado pelo computador sobre a velocidade média do percurso. Alguns dos computadores para carros mais sofisticados tornam possível ao motorista programar a velocidade de todo o trajeto. O carro manterá a velocidade programada, sem que o motorista necessite pressionar o acelerador.

O computador instalado em veículos para longos percursos tem finalidade mais importante: pode ser utilizado como um relatório eletrônico, permitindo à autoridade de trânsito verificar quanto tempo o motorista esteve ao volante, a que velocidade dirigia e qual a distância percorrida.

Uma das vantagens mais importantes proporcionadas pelo computador é a ampliação do rendimento do combustível. A BMW, indústria automobilística alemã, já constrói carros com um equipamento que determina a mistura adequada de gasolina e ar, necessária a cada condição de percurso.

Na verdade, é como se o carro fosse “ajustado” muitas vezes por segundo para possibilitar o melhor rendimento possível do combustível. O equipamento faz a contínua medição da mistura de combustível e ar e os ajustes necessários, levando em consideração a velocidade do carro, as marchas adequadas e a temperatura do motor.

O futuro

Como será o carro do futuro? Teoricamente, seria possível ao computador assumir o controle total do carro. Tudo o que o motorista teria a fazer seria programar o trajeto. O computador dirigiria o carro automaticamente, mediante informações transmitidas por sensores instalados na estrada ou pela comunicação com computadores de centrais de tráfego. Poderíamos contar também com outro aperfeiçoamento: um radar que automaticamente ajustaria a velocidade do carro, se este se aproximasse em demasia do carro da frente.

Em poucos anos, os instrumentos convencionais serão substituídos por uma tela de monitor. O motorista do carro terá condições de chamar dados, tais como temperatura do motor e nível de combustível, que serão mostrados na tela. Os dados sobre o percurso poderiam ser apresentados no pára-brisa, evitando que o motorista necessite tirar os olhos da estrada. Também é provável que os carros venham a ser equipados com um computador que informe ao mecânico qualquer problema. O mecânico ligaria o computador do carro ao computador pré-programado da oficina, que imediatamente analisaria o estado de todos os componentes, identificando desgastes e defeitos. Na foto abaixo, um modelo computadorizado para cuidar do trajeto, montado no painel de controle de um veículo Honda.





Mensagem recebida

Pressione uma tecla e o lado oculto de seu micro — o software — entra em ação: decodifica instruções, consulta a memória e verifica o teclado à espera do comando seguinte.

O computador é um conjunto de metal, plástico e silício que, sem um programa na memória, é incapaz de desenvolver algum tipo de tarefa útil, assim como um toca-discos sem disco no prato. O processo para conseguir dele um desempenho específico é chamado programação. Mesmo os principiantes em programação percebem a existência de duas fases distintas para resolver um problema ou desenvolver determinado tema. Em primeiro lugar, o tema deve ser expresso e escrito de modo compreensível para o computador. Em segundo lugar, esse programa precisa ser carregado no computador e “processado”. Essas duas fases podem ser subdivididas em dois outros estágios: o programador participa do primeiro estágio, enquanto no segundo o computador deve agir sozinho (comumente sem o conhecimento ou a interferência do usuário).

Suponha que você queira escrever um programa para preparar uma folha de pagamento. A primeira coisa necessária é o conhecimento claro do tema. O que você deseja como saída (output ou “produto”) de seu computador? Que informações o computador precisa para calcular os salários? O passo seguinte e essencial é especificar com mais detalhes o processo pelo qual a saída será produzida, por exemplo: “De que modo são calculados os impostos e descontos?”

Em uma grande empresa, isso é executado, em geral, por um analista de sistemas, especialista em analisar a forma pela qual a empresa opera e descrevê-la de modo que seja facilmente traduzida em um programa. Para programas domésticos ou educacionais, tudo isso poderá ser normalmente elaborado pelo próprio programador.

Se os computadores compreendessem perfeita-

mente o português, essa “especificação de programa” poderia ser logo processada; infelizmente, eles ainda não podem fazê-lo. Muitos iniciantes encontram dificuldades porque tentam escrever o programa do início ao fim, do mesmo modo que se traduz um texto do português para outra língua. O mais aconselhável é analisar o problema de forma diferente. Programadores mais experientes poderiam, por exemplo, dividir a especificação da folha de pagamento em quatro “módulos”: fornecimento dos dados mensais, cálculos, armazenamento dos resultados cumulativos, como “impostos pagos este ano”, e impressão do holerite.

Cada módulo é dividido em estruturas menores, o que é chamado de “programação estruturada”; estas seções menores são simples e podem ser expressas em uma ou duas linhas de programa. Finalmente, todo o agrupamento de linhas — a listagem do programa — é digitado no computador.

Um bom programador sempre mantém anotações de todos os estágios, que vão refletir os diversos e distintos níveis, desde o problema escrito em português até o programa escrito em uma linguagem de alto nível como o BASIC.

O que acontece desde o momento em que você digita RUN está sob total controle do computador e, novamente, abrange diferentes níveis ou estágios. Todavia, as operações internas do computador encontram-se “escondidas”; o usuário tem apenas conhecimento de seu programa, que a ele solicita as informações relevantes e produz a saída necessária.

Como o microprocessador não compreende linguagens de alto nível, a primeira tarefa do computador consiste em traduzir as instruções em código de

Do problema ao programa

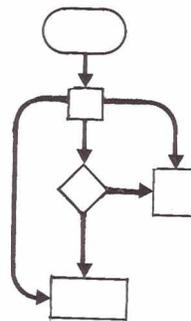
Um programa de computador nasce da consciência de que há um problema a ser solucionado; por exemplo, como manter a temperatura de uma estufa em nível constante. Para obter a resposta, este problema precisa passar por diversos níveis de processamento, que resultam em um programa final.



O problema aparece...



A idéia é rabiscada em um pedaço de papel.



Um fluxograma é formulado para analisar o problema e desenvolver a estrutura do programa...



É, então, traduzido para uma linguagem de computador; por exemplo, BASIC.



máquina. Em muitos microcomputadores isso é executado pelo interpretador, que está permanentemente armazenado na ROM (Read Only Memory).

O interpretador é um programa em código de máquina, executado diretamente pelo microprocessador. Ao digitar RUN, o interpretador começa a examinar o programa do usuário, caractere por caractere, consultando todas as frases em seu próprio dicionário. Ao encontrar um caractere que não compreende (um simples erro de datilografia, por exemplo), ele interrompe a interpretação do programa e imprime uma mensagem de erro na tela, como SIN-TAX ERROR (erro de sintaxe).

Se a palavra estiver contida no seu dicionário (por exemplo, PRINT), ela é imediatamente passada ao interpretador, que sabe como lidar com essa função. Nesse caso, a rotina examina o que se segue à palavra PRINT no programa do usuário, e prepara esses dados como um conjunto de caracteres a ser mostrado na tela.

Nesse ponto, inicia-se o nível seguinte da operação. Em outro lugar da memória do computador há uma rotina, que pode aceitar um fluxo de caracteres, armazenando-os em outra área da memória reservada para a tela, dispondo-os de forma que sejam convertidos no tipo de sinal necessário ao monitor. Isso deve ser realizado continuamente, até mesmo enquanto o próprio programa esteja efetuando cálculos.

O mesmo se aplica à outra extremidade do computador — o teclado. Uma rotina especialmente escrita dentro do computador deve verificar o teclado para descobrir se qualquer uma das teclas foi pressionada. Em caso afirmativo, coloca os códigos apropriados em uma determinada área da memória, para uso como input (entrada de dados) ao programa do usuário. E, como você pode desejar parar a operação do programa em qualquer momento, usando a tecla BREAK, o teclado precisa ser verificado continuamente, mesmo que o programa esteja sendo processado.

Na realidade, o microprocessador encontrado na maioria dos microcomputadores pode realizar apenas uma tarefa de cada vez; por isso ele divide o tempo entre a interpretação do programa do usuário e suas próprias funções internas, como a verificação



do teclado e o controle da tela. Um método para isso é “interromper-acionar”, em que um circuito eletrônico especial interrompe o microprocessador praticamente cinquenta vezes por segundo e “lembrar” de desempenhar suas tarefas de manutenção e outras funções relativas à tela e ao teclado, antes de retomar o que estava fazendo.

Mesmo quando o programa está sendo digitado, vários níveis de processamento devem ser executados pelo computador, antes que se produzam os resultados desejados. Embora o processo possa parecer complexo, o computador se encarrega dele com muita eficiência e rapidez.

Hoje em dia, a tendência é fazer com que os computadores executem o maior número possível de tarefas rotineiras. A próxima geração de computadores será capaz, provavelmente, de escrever o programa inteiro, a partir de uma especificação completa feita pelo usuário na sua própria língua.

O software oculto

Em todos os computadores, há uma hierarquia complexa e oculta de software, que trabalha de modo contínuo. Dentre suas tarefas, o software controla o momento em que a tecla é pressionada; verifica qual é a tecla e o que está na tela; que instruções são dadas aos periféricos e qual a situação e o conteúdo da memória RAM. Essas atividades se processam continuamente, enquanto o operador se preocupa apenas com a função seguinte do programa. O princípio da hierarquia oculta do software é sempre o mesmo, em qualquer tipo de computador.



O programa é introduzido na memória do computador via teclado.



O programa BASIC passa a seguir por uma série de chips, que o convertem em código de máquina compreensível pela CPU.



O processamento tem lugar na CPU. Os dados resultantes são transmitidos em seguida a um periférico (por exemplo, impressora, monitor, unidade de disco etc.).



Neste caso a impressora produz uma listagem.



Se o programa estiver correto, o problema estará solucionado.

Verdadeiro ou falso ?

Os computadores ainda não podem “pensar” como você, mas são capazes de seguir facilmente todas as leis da lógica.

A CPU (Unidade Central de Processamento) é comumente considerada o coração do computador. É o lugar onde ocorrem todos os cálculos e as tomadas de decisão lógica. Mas de que modo são feitos esses cálculos e tomadas essas decisões?

Para compreender isso é necessário conhecer os fundamentos da aritmética binária e estar familiarizado com as portas lógicas. Essas portas são circuitos elétricos capazes de tomar decisões lógicas e fazer comparações. Isso parece mais complicado do que é na realidade e os princípios podem ser facilmente ilustrados com exemplos da vida diária.

Existem três tipos fundamentais de portas — a AND (E), a OR (OU) e a NOT (NÃO).

Conexões lógicas

A porta AND é um circuito que dá saída “verdadeira”, se todas as entradas forem “verdadeiras”. Vamos ver o que isso significa. Suponha que você queira fazer uma viagem para o campo. Se tem carro E (AND) gasolina, pode fazer a viagem. Se tem gasolina, mas não tem carro, não pode viajar. De modo semelhante, o carro sem a gasolina significa que não haverá viagem.

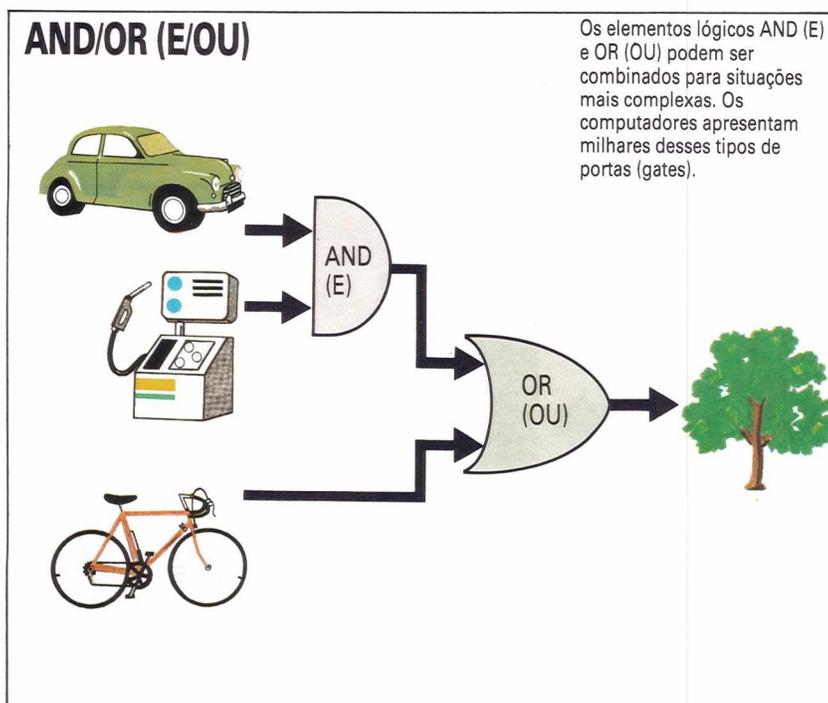
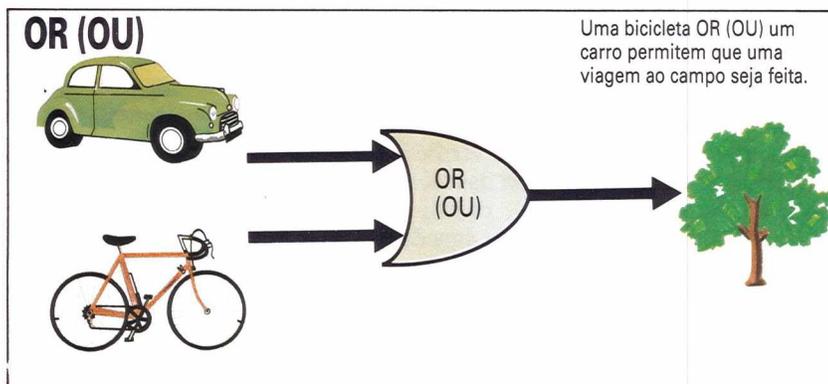
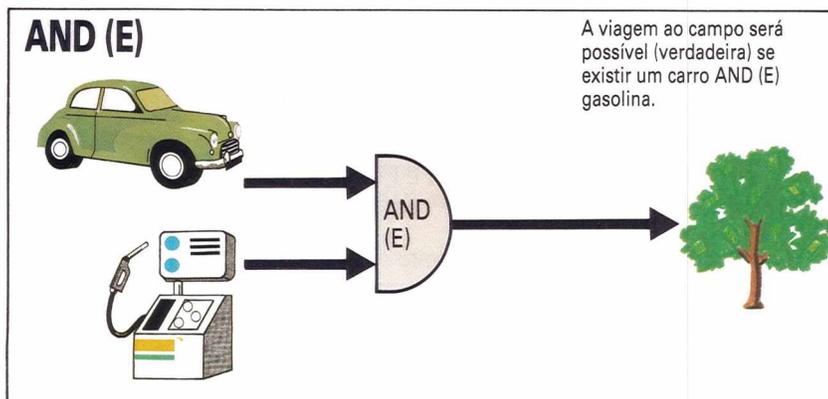
Neste “circuito” AND existem duas condições de entrada e ambas precisam ser “verdadeiras”. Para poder viajar (a “saída”) precisa haver um carro E (AND) precisa haver gasolina, e só neste caso a saída se torna “verdadeira”. Posteriormente, veremos de que modo este diagrama lógico pode ser apresentado na forma de uma equação lógica, além de poder ser representado em uma “tabela de validação”.

Imagine agora uma situação ligeiramente diferente. Alguém deseja viajar para o campo. A viagem será possível se a pessoa tiver carro OU (OR) bicicleta (vamos supor que o carro esteja com gasolina). Se a pessoa tiver carro poderá viajar. Se tiver bicicleta, também. Contudo, se nenhuma das condições de entrada for verdadeira, a viagem se tornará impossível — no jargão do computador, a saída se torna falsa (i.e., não é verdade que a pessoa vai viajar para o campo).

Há mais uma porta lógica fundamental, a porta NOT. Esta porta simplesmente fornece uma saída oposta à entrada. Se a entrada for verdadeira, a saída será falsa. Se a entrada for falsa, a saída será verdadeira. Ampliando nossa metáfora, a viagem ao campo de carro ou bicicleta será falsa se houver um pneu furado.

Se a entrada (pneu furado) é verdadeira, então a saída (a viagem) será falsa.

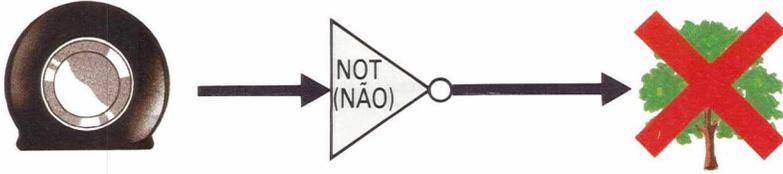
Esses elementos lógicos podem ser combinados e ilustramos isso com o exemplo da viagem ao campo.





NOT (NÃO)

O NOT (NÃO) fornece uma saída que é oposta à entrada. Se for verdade que o pneu está vazio, não será verdade que haverá uma viagem ao campo.



As combinações de AND, OR e NOT permitem que todas as decisões baseadas na lógica convencional sejam feitas.

Experimente solucionar o que seria necessário para se fazer um churrasco no jardim, por exemplo. Fica bem complicado. Para termos um churrasco no jardim (a saída verdadeira), precisaríamos de várias condições de entrada: dinheiro OU (OR) cheque OU (OR) cartão de crédito (para comprar comida e bebida) E (AND) um dia livre E (AND) um tempo agradável E (AND) uma churrasqueira E (AND) carvão E (AND) fósforo E (AND) espetos.

Nos computadores empregamos 0 e 1 para significar falso e verdadeiro. O computador interpreta a voltagem positiva como 1 e a voltagem zero como 0. E o circuito AND pode ser feito, utilizando transistores, de modo que, se ambas as entradas são voltagens positivas, a saída é também uma voltagem positiva. Se uma ou ambas as entradas forem de voltagem zero, a saída do circuito também será zero.

Um circuito eletrônico OR fornece uma saída de voltagem positiva, se uma ou ambas as entradas forem positivas. Se ambas as entradas forem 0, a saída também será 0. Em um circuito NOT, a entrada é simplesmente invertida: se a entrada for positiva, a saída será 0; se a entrada for 0, a saída será positiva.

Tabelas de validação

Os símbolos que utilizamos na ilustração são os mesmos dos diagramas de circuito do computador. A fim de sabermos como as decisões lógicas podem ser tomadas empregando circuitos elétricos, vamos analisar a “tabela de validação” para o AND. Se utilizarmos a letra (c) para representar a condição de entrada “possuir um carro” e a letra (g) para “ter gasolina”, podemos representar a condição de saída “fazer uma viagem para o campo” com a letra (v). Podemos também empregar V para verdadeiro e F para falso. A tabela de validação apresenta todas as combinações possíveis de condições de entrada e o efeito da utilização de AND na saída. Assim temos:

CARRO	F	V	F	V	(c)
GASOLINA	F	F	V	V	(g)
VIAGEM	F	F	F	V	(v)

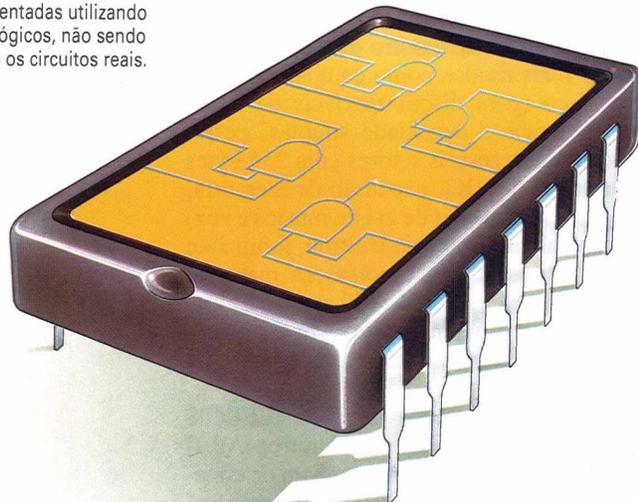
A tabela de validação para duas entradas AND

CARRO	0	1	0	1	(c)
GASOLINA	0	0	1	1	(g)
VIAGEM	0	0	0	1	(v)

A mesma tabela usando 0 e 1 para Falso e Verdadeiro

O chip 7408

Os grandes chips de silício normalmente contêm milhares de portas (gates) para executar a lógica de AND, OR e NOT. O pequeno chip 7408 da ilustração contém todos os transistores e circuitos para quatro portas AND. As portas são representadas utilizando símbolos lógicos, não sendo mostrados os circuitos reais.





Unitron AP II

O Unitron AP II é eficiente e versátil — editando textos, controlando estoques ou resolvendo equações.

O Unitron AP II é um microcomputador de uso pessoal que também pode ser utilizado por empresas. Para isso, bastam algumas modificações em sua configuração, mantendo-se a mesma Unidade de Processamento Central (CPU).

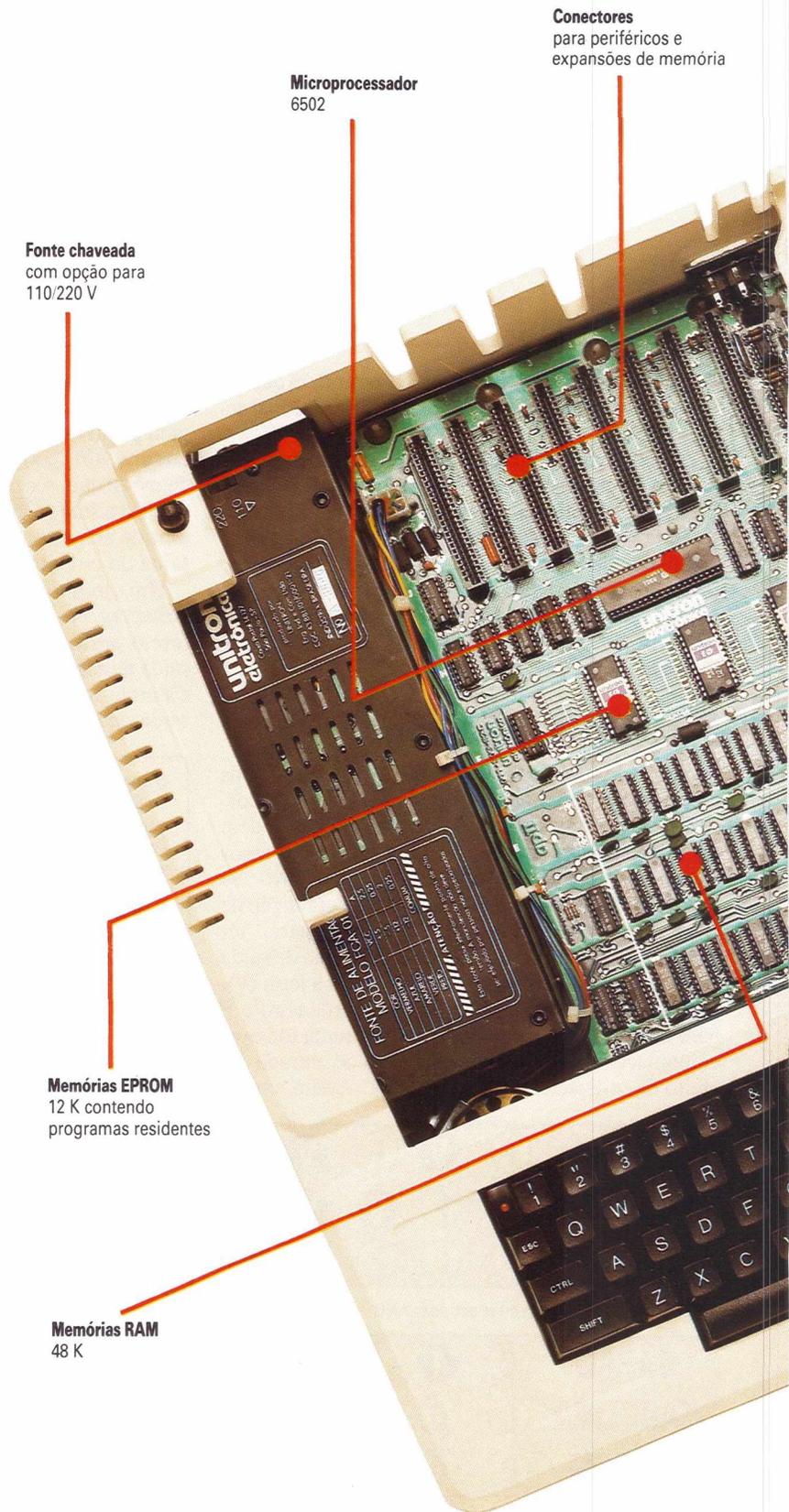
A CPU do Unitron AP II é um microprocessador 6502, com 48 K de memória RAM (para armazenamento de programas e dados) e 12 K de memória EPROM (que inclui a linguagem BASIC). A memória RAM pode ser expandida por meio de módulos especiais de 32, 64 ou 128 K. Neste último caso, o módulo vem acompanhado de um programa que permite sua utilização como pseudodisco.

O teclado do equipamento é alfanumérico, semelhante ao de uma máquina de escrever normal. Compõe-se de 52 teclas, a maioria delas com mais de uma função; com esse teclado é possível escrever textos acentuados ou não, em letras maiúsculas e minúsculas.

O Unitron AP II pode ser adquirido com ou sem monitor de vídeo. No primeiro caso, a definição de imagem é superior a 50.000 pontos. No segundo, o computador deve ser conectado a um aparelho de televisão, em cores ou preto e branco, por ligação via placa conversora PAL. O padrão de vídeo do equipamento é de 24 linhas, formadas por quarenta caracteres, ou — usando-se um módulo especial — de 24 linhas, compostas de oitenta caracteres.

O Unitron AP II tem oito conectores para periféricos, mais um para cassete e outro para videogames. Os periféricos mais usados como equipamento são unidade de disco (para discos flexíveis de 5 1/4 polegadas, com face e densidade simples, perfazendo 35 pistas de dezesseis setores com 256 bytes cada, num total de 143 K), expansão de memória, vídeo composto, som (sinais de aviso, música e voz) e impressora paralela. A própria Unitron, fabricante do computador, possui uma impressora matricial bidirecional, com velocidade de 100 cps e carro de oitenta colunas, que também imprime em caracteres comprimidos, aumentando a capacidade da linha para 132 caracteres.

Além disso, usando-se uma interface IEEE488 também é possível compatibilizar o Unitron AP II com instrumentos elétricos que disponham de conexão desse tipo. Finalmente, o sistema operacional do Unitron AP II faz com que o equipamento seja totalmente compatível com o Apple II e, portanto, com grande quantidade de software disponível no mercado brasileiro.



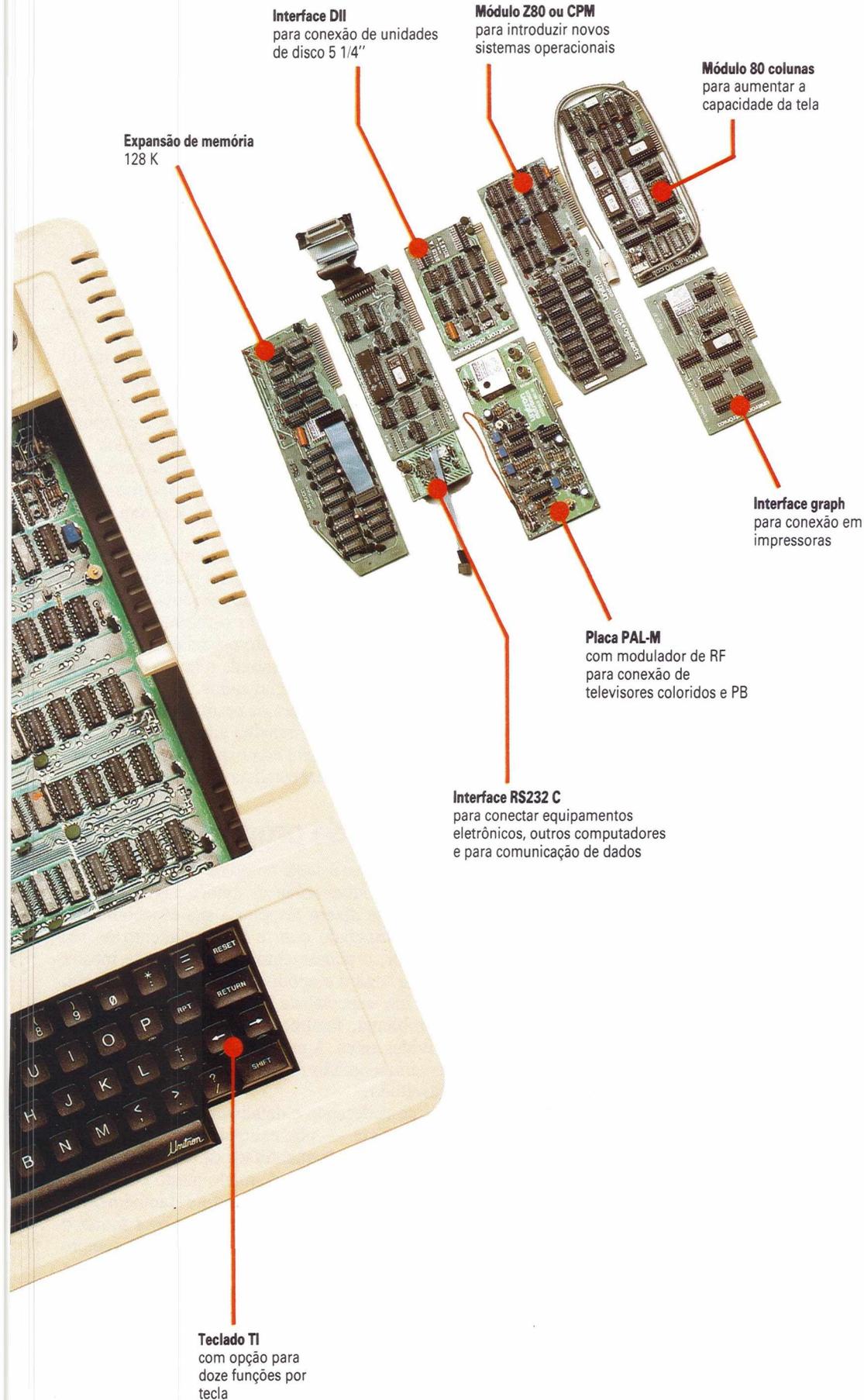
Microprocessador 6502

Conectores para periféricos e expansões de memória

Fonte chaveada com opção para 110/220 V

Memórias EPROM 12 K contendo programas residentes

Memórias RAM 48 K



Interface DII
para conexão de unidades
de disco 5 1/4"

Módulo Z80 ou CPM
para introduzir novos
sistemas operacionais

Módulo 80 colunas
para aumentar a
capacidade da tela

Expansão de memória
128 K

Interface graph
para conexão em
impressoras

Placa PAL-M
com modulador de RF
para conexão de
televisores coloridos e PB

Interface RS232 C
para conectar equipamentos
eletrônicos, outros computadores
e para comunicação de dados

Teclado TI
com opção para
doze funções por
tecla

UNITRON AP II

MICROPROCESSADOR

6502

CLOCK

1,00 MHz

MEMÓRIA

48 K de memória RAM e 12 K de EPROM, que contém um BASIC avançado.

VÍDEO

O Unitron possui saída de vídeo própria para conexão a monitor ou ligação via placa conversora PAL a televisores preto e branco ou coloridos.

TECLADO

Teclado tipo máquina de escrever, com 52 teclas, várias com dupla função, letras maiúsculas e minúsculas e alguns acentos.

LINGUAGENS

BASIC, COBOL, FORTRAN, PASCAL

PERIFÉRICOS

Cassete, monitor, até 14 unidades de disco (5 1/4"), impressora, interface RS232 C e IEEE488, joystick.

DOCUMENTAÇÃO

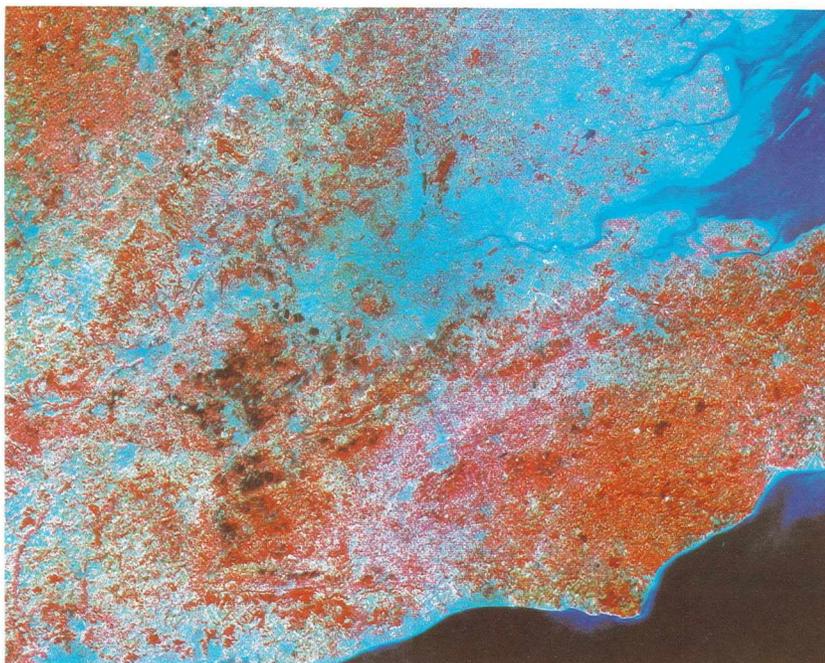
O manual que acompanha o equipamento possui um capítulo inicial que descreve o sistema e dá instruções detalhadas sobre como conectar os vários periféricos. A partir do segundo capítulo é descrita a linguagem BASIC utilizada pelo Unitron. Constam dos apêndices uma lista das mensagens de erro e um resumo das características de edição e das instruções BASIC.



Consulte o chip

Muitos profissionais poderão ficar livres da rotina graças aos sistemas especializados: computadores programados para analisar e responder a perguntas complexas e variadas.

Inteligência artificial, computadores que pensam e tomam decisões como seus criadores humanos, por enquanto, são uma fantasia da ficção científica. Ainda não se chegou ao conhecimento total de como funciona o cérebro humano e, apesar de alguns avanços nesse terreno, há poucos projetos de um computador “estilo 2001”.



Uma visão do espaço

O satélite Landsat 4 foi lançado em julho de 1982 e está numa órbita que cobre toda a superfície do planeta. Ele retorna ao mesmo ponto de sua órbita a cada vinte dias. Todos os dados enviados pelo sensor vêm em forma digital. Na ilustração a informação foi processada fotograficamente, a fim de apresentar aspectos de Londres e do Sudeste da Inglaterra. A água limpa é representada em azul-escuro; a água rasa com sedimentos, em azul-claro; as cidades e os campos cultivados, em azul-acinzentado; os campos abertos, em marrom-avermelhado; o milho bom para colheita, em verde; e outras vegetações aparecem em vermelho-vivo.

Mas, no caso de uma tarefa específica, e se um computador precisa apenas “parecer” inteligente, atuando num campo muito restrito da atividade humana, então a simples simulação da inteligência torna-se algo bem mais fácil.

Esta é a teoria que está por trás dos sistemas especializados. A idéia se baseia no trabalho realizado por um especialista de determinada área. Por exemplo, um geólogo ou um cirurgião podem alimentar um computador com códigos e conhecimentos especializados. Depois de trabalhar as informações recebidas, o computador responde a pessoas não especializadas, sobre um campo específico.

Os sistemas especializados poderiam ter várias utilidades. Um programa foi desenvolvido para diagnosticar a causa de dor de estômago mediante perguntas aos pacientes sobre os sintomas. Outro programa se vale de nosso conhecimento de geologia para apontar os locais onde provavelmente seria achado molibdênio ou outros minerais. E outro deduz as prováveis estruturas para moléculas orgânicas, a partir de massas de dados experimentais não

estruturados. Todas essas tarefas seriam normalmente realizadas por um profissional com muita experiência e devidamente treinado. Graças ao computador, essas pessoas podem agora dedicar seu tempo a atividades mais originais.

Mas os sistemas especializados significam muito mais que a simples substituição dos especialistas humanos. Uma vez que o conhecimento de um especialista está sendo usado pelo programa, o computador frequentemente revela dados inesperados. Às vezes, o equipamento aponta uma relação entre itens de informação que os homens não conseguiram captar; e sugere ainda novos caminhos a serem explorados.

Assim, pode-se dizer que os sistemas especializados são, ou serão brevemente, um passo importante no que se refere à utilização do computador. Muitos computadores podem utilizar o mesmo programa, substituindo o conhecimento especializado de uma única pessoa pelo de um grande número de computadores igualmente especializados.

O único problema para os pesquisadores é escrever um programa que trabalhe adequadamente; um programa que seja exato como um computador e inteligente como um especialista humano.

A criação do programa

O primeiro passo é saber como os especialistas humanos se decidem sobre as evidências e sobre as perguntas que dizem respeito à sua especialidade. Quando comparado com o funcionamento de um computador, o pensamento humano não se mostra totalmente lógico e apóia-se muito na experiência. Quando um especialista humano depara com um novo problema, fatalmente o compara com situações vividas antes. A partir dessas comparações, algumas tentativas são feitas a fim de se obter o melhor resultado.

Mas, considerando o detalhado conhecimento de um especialista — por exemplo, um médico —, veremos que grande quantidade de regras tem de ser armazenada e associada de modo muito complexo. Além disso, modificações posteriores também serão necessárias para que o computador possa imitar o comportamento humano. Um médico nem sempre está seguro do que afirma e usa as expressões “estou quase certo de que...” ou “bastante confiante”, ao expressar uma opinião. Baseado em alguns sintomas, o médico pode apenas ter 30% de certeza de seu diagnóstico.

Assim, as regras num computador devem ter valores de probabilidade relativos às suas conclusões, indo de 100%, quando há apenas uma conclu-



são possível, até 1%, quando há outras cem conclusões igualmente prováveis.

A maioria dos sistemas especializados funciona por meio de um diálogo entre o entrevistado e o computador. O entrevistado deve detalhar os problemas a serem resolvidos pelo computador; e o modo mais simples de evitar problemas aqui é ter o computador fazendo uma série de perguntas de múltipla escolha ao usuário. Desse modo, o usuário não corre o risco de usar termos e frases que o computador não compreende. A seguir, o computador testa a informação de acordo com as normas depositadas no seu estoque de conhecimento. O modo como o com-

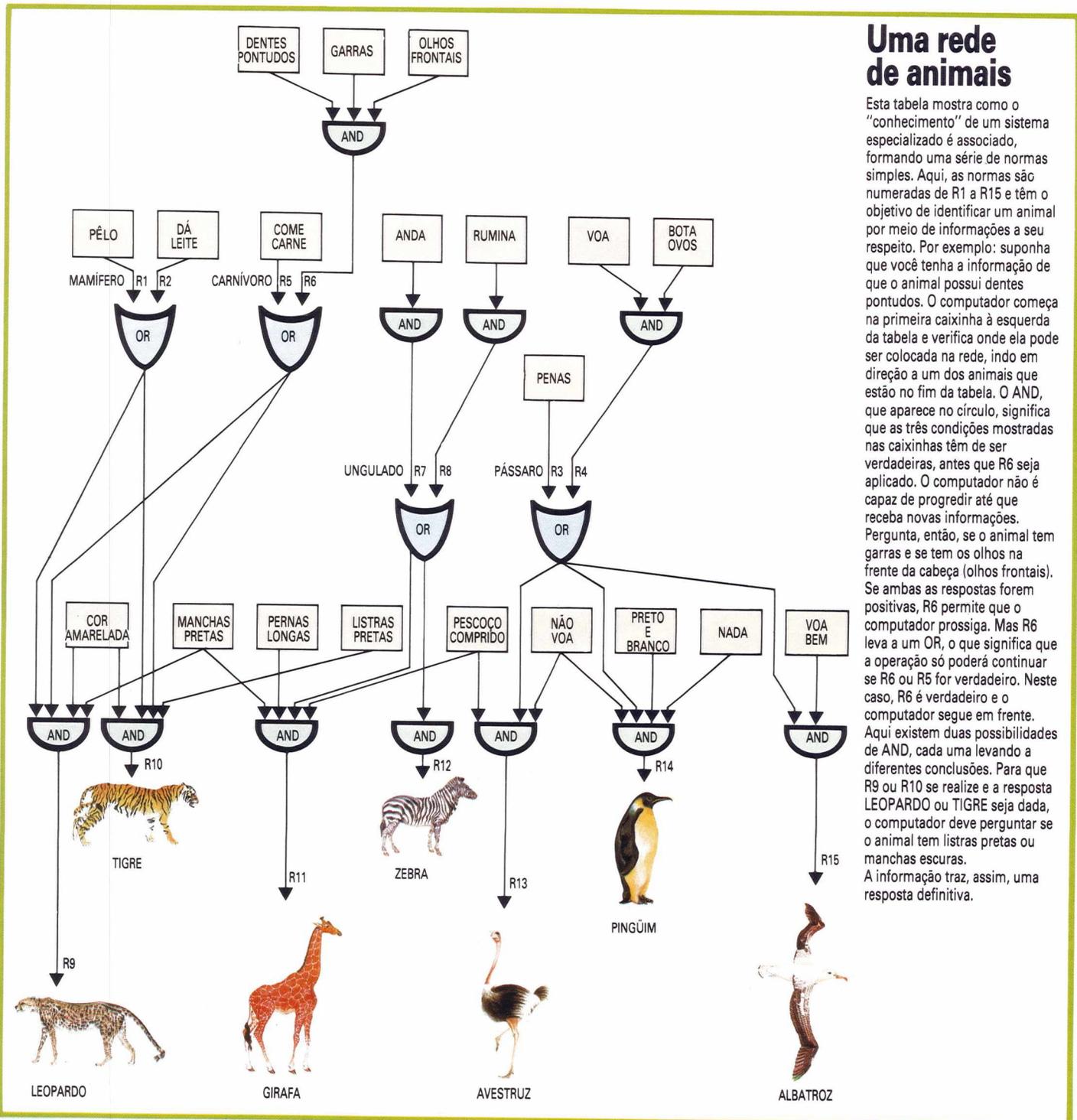
putador chega ao circuito das normas depende das respostas que o usuário dá às suas perguntas. À medida que o programa avança, surge nova pergunta do computador.

O mais conhecido sistema especializado na Inglaterra, o programa Mickie de diagnóstico médico, funciona num microcomputador. E outros produtos comerciais a serem lançados transformarão o computador que se tem em casa em um especialista de vários assuntos.

Talvez ainda não possamos conversar com os computadores. Mas eles já nos dão respostas em que certamente podemos confiar.

Uma rede de animais

Esta tabela mostra como o "conhecimento" de um sistema especializado é associado, formando uma série de normas simples. Aqui, as normas são numeradas de R1 a R15 e têm o objetivo de identificar um animal por meio de informações a seu respeito. Por exemplo: suponha que você tenha a informação de que o animal possui dentes pontudos. O computador começa na primeira caixinha à esquerda da tabela e verifica onde ela pode ser colocada na rede, indo em direção a um dos animais que estão no fim da tabela. O AND, que aparece no círculo, significa que as três condições mostradas nas caixinhas têm de ser verdadeiras, antes que R6 seja aplicado. O computador não é capaz de progredir até que receba novas informações. Pergunta, então, se o animal tem garras e se tem os olhos na frente da cabeça (olhos frontais). Se ambas as respostas forem positivas, R6 permite que o computador prossiga. Mas R6 leva a um OR, o que significa que a operação só poderá continuar se R6 ou R5 for verdadeiro. Neste caso, R6 é verdadeiro e o computador segue em frente. Aqui existem duas possibilidades de AND, cada uma levando a diferentes conclusões. Para que R9 ou R10 se realize e a resposta LEOPARDO ou TIGRE seja dada, o computador deve perguntar se o animal tem listras pretas ou manchas escuras. A informação traz, assim, uma resposta definitiva.



Pronta para imprimir

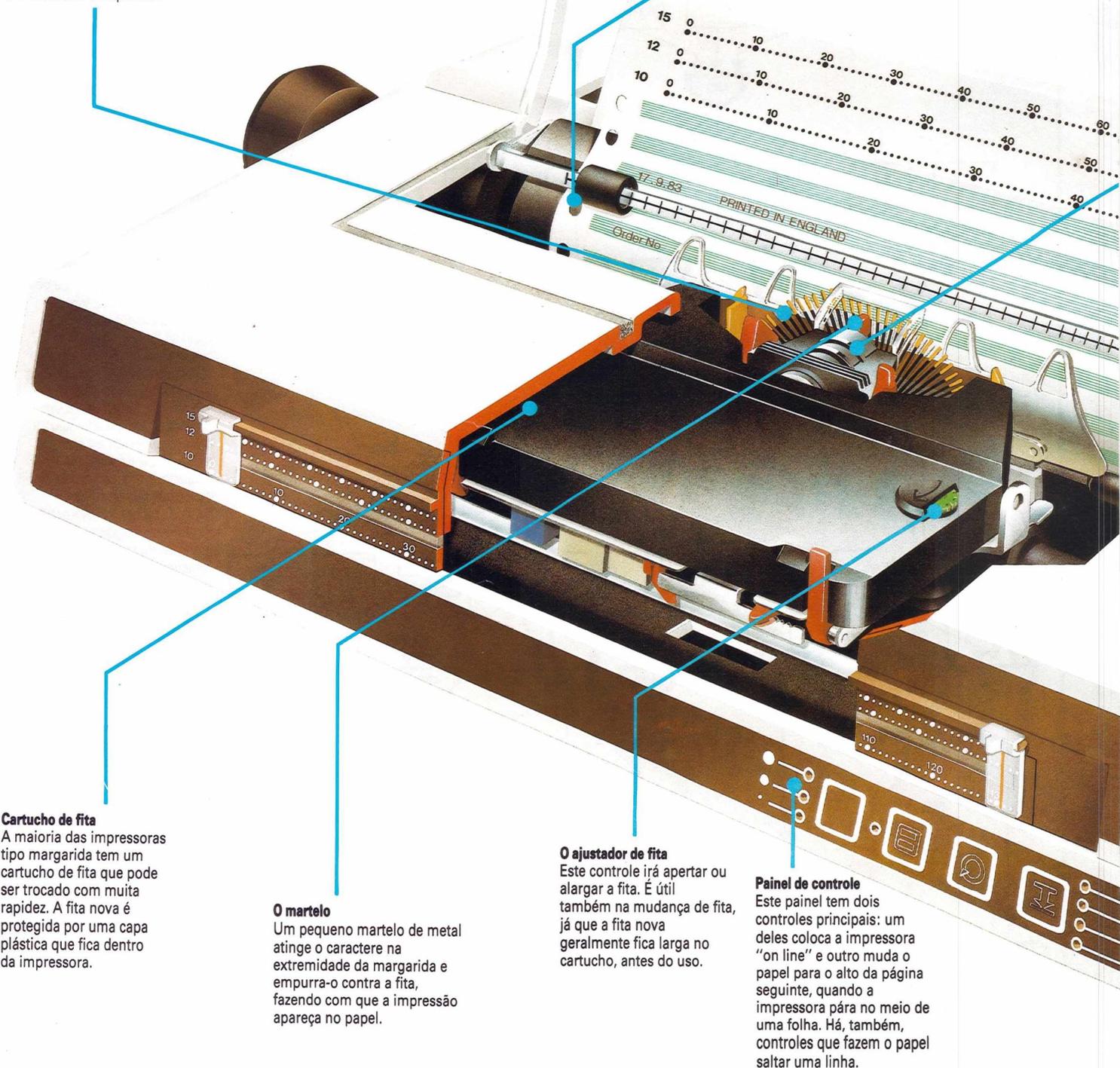
A tecnologia da impressão já produziu modernos e eficientes equipamentos que podem ajudá-lo em muitos de seus trabalhos. Escolha o tipo mais adequado a suas necessidades.

Guias de papel

A alimentação de papel é automática, por meio de engrenagens, nas quais vão se encaixando os furos da borda do papel.

A impressora tipo margarida (daisy wheel)

Esta impressora traz os caracteres na extremidade das "pétalas". A pressão de uma tecla move a roda em direção à pétala desejada e o caractere é impresso.



Cartucho de fita

A maioria das impressoras tipo margarida tem um cartucho de fita que pode ser trocado com muita rapidez. A fita nova é protegida por uma capa plástica que fica dentro da impressora.

O martelo

Um pequeno martelo de metal atinge o caractere na extremidade da margarida e empurra-o contra a fita, fazendo com que a impressão apareça no papel.

O ajustador de fita

Este controle irá apertar ou alargar a fita. É útil também na mudança de fita, já que a fita nova geralmente fica larga no cartucho, antes do uso.

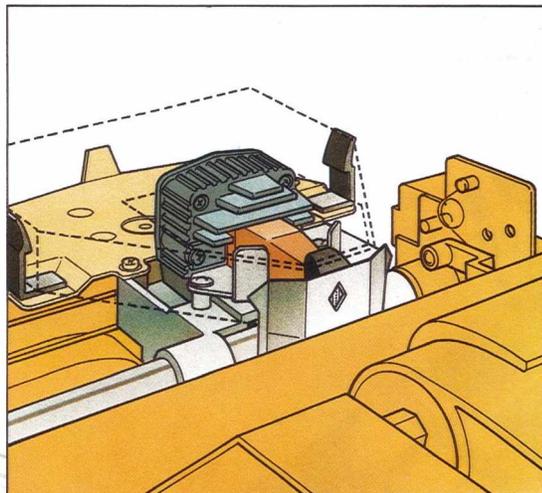
Painel de controle

Este painel tem dois controles principais: um deles coloca a impressora "on line" e outro muda o papel para o alto da página seguinte, quando a impressora pára no meio de uma folha. Há, também, controles que fazem o papel saltar uma linha.



O motor da impressora tipo margarida

O motor gira a margarida e alinha os caracteres com o martelo, que os comprime contra a fita e o papel.

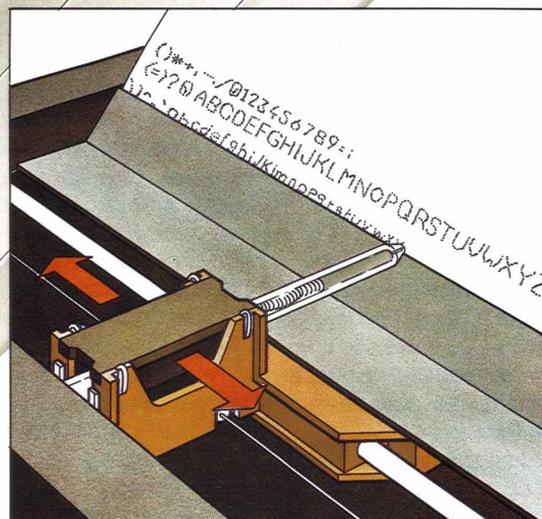


A impressora de matriz de pontos (dot matrix)

O método de matriz de pontos utiliza um conjunto (matriz) de pontos que formam um caractere. A cabeça de impressão contém um grupo de agulhas que batem na fita, deixando um ponto no papel. Com o movimento da cabeça de impressão no papel, os pinos são lançados por impulsos elétricos na sequência correta, para formar o caractere.

A impressora por jato de tinta (ink-jet)

Esta impressora tem um nome bastante apropriado, já que lança um jato de tinta por um tubo que transforma a tinta em gotas minúsculas. Cada gota recebe uma carga elétrica e passa por placas defletoras de metal. A carga faz com que as gotas de tinta atinjam o papel, na configuração exata para formar um caractere.



Até agora, provavelmente, você não pensou em utilizar uma impressora. De fato, se você está satisfeito com o uso de seu computador para jogos de lazer ou balanço de contas domésticas, não há necessidade de cópias impressas daquilo que aparece na tela de seu televisor ou monitor.

À medida que você se especializar no uso do microcomputador, as limitações causadas pela falta de uma impressora ficarão evidentes, sobretudo na eventualidade de precisar fazer um trabalho mais complexo. Se você quiser escrever seus próprios programas, certamente vai se interessar em guardar cópias de suas listagens. E, se usar seu computador para elaborar uma contabilidade detalhada, o registro impresso do resultado de seus cálculos se tornará necessário.

Para escolher uma impressora adequada, você deve considerar que o custo do aparelho vai depender da velocidade na produção de palavras e da qualidade do resultado.

A escolha de sua impressora

Existem três tipos principais de impressora para microcomputadores: matriz de pontos, margarida e impressora térmica.

O método mais comum de impressão é o da matriz de pontos, que funciona por meio de uma cabeça de impressão que contém um grupo de agulhas. Os caracteres são impressos no papel mediante a combinação dessas agulhas. A vantagem da matriz de pontos está na rapidez e no preço. Entretanto, como as letras ou números são feitos com séries de pontos, a qualidade da impressão deixa a desejar; além disso, essa impressora faz muito barulho.

Algumas impressoras de matriz de pontos resolvem o problema da má qualidade da impressão, pois imprimem os pontos duas ou três vezes. Nesse caso, a cabeça move-se levemente para que novos pontos preencham os espaços deixados na primeira impressão.

As impressoras de matriz de pontos são aceitáveis, caso você não se importe com a qualidade das cópias impressas. Elas podem produzir tabelas e gráficos, pois a cabeça é capaz de imprimir modelos predeterminados além dos caracteres normais. Mas, caso você queira uma boa qualidade de impressão, precisará de uma máquina diferente.

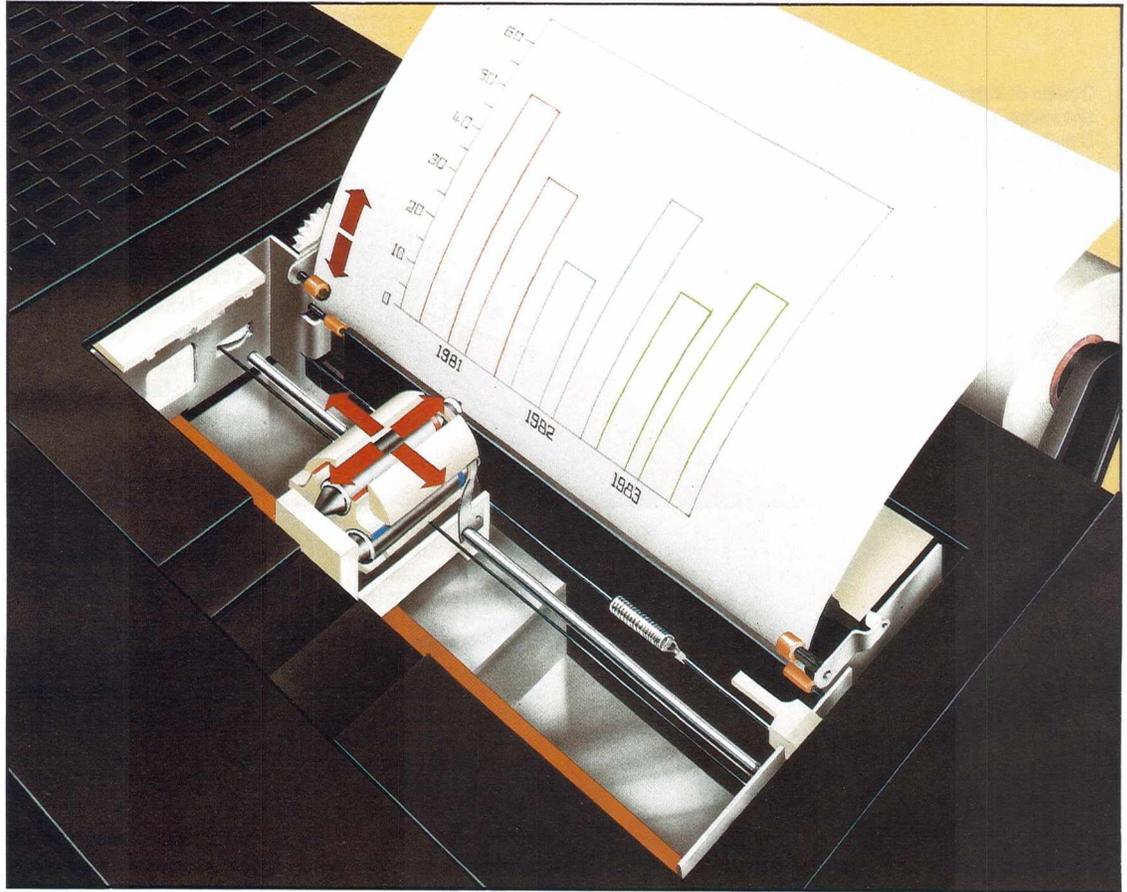
A impressora tipo margarida, assim chamada porque usa uma roda com "pétalas" que se parecem com as da margarida, oferece qualidade superior de impressão. Para imprimir, as rodas giram, colocando cada pétala na linha com um pequeno martelo de metal que empurra o caractere no fim da pétala contra a fita de impressão. A margarida é feita de metal ou plástico. Você pode trocá-la para obter outras variedades de impressão, como numa máquina de escrever com esfera.

O problema com a impressora tipo margarida é ser mais lenta e mais cara do que a de matriz de pontos. Além disso, mais trabalhosa, quando se trata de produzir formas gráficas variadas, que necessitariam de trocas constantes das margaridas.

As impressoras por jato de tinta são de preço ainda mais elevado. Injetam gotas de tinta que for-

A impressora "esferográfica" (ball-pen)

Este tipo de impressora representa uma novidade em técnicas de impressão. A cabeça de impressão controla quatro canetas "esferográficas" especialmente projetadas. Quando é dado o comando PRINT, o papel sobe e desce a fim de criar os traços verticais, ao mesmo tempo que as canetas se movem para os lados, para criar os traços horizontais. Este sistema apresenta a vantagem de poder ser usado para imprimir tabelas coloridas e gráficos. Oferece, ainda, qualidade de impressão superior à do método de matriz de pontos, já que tem seus caracteres formados por traços simples de caneta. Entretanto, comparado a outros métodos, tal sistema é vagaroso, e as canetas precisam ser trocadas regularmente, quando textos longos forem impressos com muita frequência.



mam o caractere a ser impresso. As gotas passam por um eletrodo e recebem carga elétrica. A seguir, um par de placas de metal desvia as gotas em diferentes direções, para formar os caracteres. Esse tipo de impressora é tão rápido que pode imprimir por segundo uma quantidade de caracteres que equivaleria a 20 metros!

A impressora térmica consiste num método alternativo que emprega um papel sensível ao calor. A cabeça de impressão transfere seu calor para o papel, de tal modo que este se escurece na área tocada e forma, ali, o caractere apropriado. As impressoras térmicas são silenciosas e rápidas. Uma das mais conhecidas é o modelo Silent Type, da Apple. Você terá de utilizar um papel especial, sensível ao calor, mais caro do que o papel comum. A qualidade de impressão dessas máquinas não é tão boa quanto a da impressora tipo margarida.

Esse tipo de impressora não é adequado a determinadas aplicações comerciais, pois não possibilita a produção de cópias.

Interface apropriada

Ao decidir comprar uma impressora é preciso saber exatamente em que você vai usá-la; e ter certeza de que tanto ela quanto o plugue de conexão são compatíveis com o seu computador. O soquete para a ligação da impressora está, geralmente, atrás do computador, e tem o nome de interface.

Os três tipos de interface mais comuns são o Centronics, o IEEE488 e o RS232. O Centronics também é conhecido como interface paralela. O micro-

computador tem uma abertura com o nome de um desses três tipos.

Entretanto, a indústria de computadores caracteriza-se por sua incompatibilidade; por isso, uma impressora e um microcomputador que usam a mesma ligação podem não ser compatíveis. Isso ocorre porque a interface deve funcionar com a mesma velocidade na impressora e no microcomputador. Essa velocidade é conhecida como "baud" e mede o ritmo de transferência dos bits da memória do computador para a impressora. Os bits são enviados para a impressora de duas maneiras: ou descem alinhados num fio único, como na interface RS232 (método serial) ou são transmitidos por vários fios, como na interface Centronics e na IEEE488 (método paralelo).

As impressoras funcionam de dois modos com o papel: alimentam uma folha por vez, como uma máquina de escrever, ou usam dois dentes de engrenagem, cada um encaixando-se nos furos que devem existir em ambos os lados da folha, assim como uma máquina fotográfica muda o filme de um lado para outro. Esse último método é mais conveniente, já que a impressora pode se alimentar de papel sozinha. Entretanto, tal método não aceita o papel timbrado, utilizado em geral pelas empresas, pois, nesse caso, as folhas não possuem a serrilha com orifícios, necessária ao funcionamento da impressora.

Assim, alguns itens devem ser observados na compra de uma impressora: sua utilização, a compatibilidade com o seu microcomputador, a quantidade de linhas ou de caracteres que ela imprime por minuto, a qualidade final da impressão, e, sem dúvida, o seu preço.



Problemas de rotina

Programas dentro de programas: conheça um novo aspecto do BASIC que tornará seus programas concisos e fáceis de ler.

Em partes anteriores deste curso de programação BASIC, pudemos digitar, processar e modificar programas, limpando depois a memória (pelo uso do comando NEW) para poder processar outros programas. Mas, sempre que havia necessidade de processar o programa anterior, era preciso digitar tudo outra vez.

Para evitar esse trabalho repetitivo, todos os computadores que operam em linguagem BASIC são equipados com um comando que permite a armazenagem de qualquer programa em fita cassete. O programa abaixo pode ser guardado em fita usando-se o comando SAVE, seguido por um nome de arquivo. Vejamos, por exemplo, o cálculo do número de azulejos necessários para revestir uma sala.

```

10 REM ESTE PROGRAMA CALCULA O NUMERO DE
  AZULEJOS
20 REM NECESSARIOS PARA REVESTIR UMA SALA
30 PRINT "FORNECER O TAMANHO DA LATERAL DO
  AZULEJO EM MM"
40 INPUT A1
50 REM LINHA 60 CALCULA A AREA DO AZULEJO
60 LET A2 = A1 * A1
70 PRINT "FORNECER O NUMERO DE PAREDES"
80 REM W ESTABELECE OS LIMITES DO LOOP
90 INPUT W
100 FOR X = 1 TO W
110 PRINT "COMPRIMENTO DA PAREDE NO. "; X;
  "EM METROS"
120 REM D E A DIMENSAO DA PAREDE
130 INPUT D
140 REM E CONVERTIDA EM MM
150 REM NA SUB-ROTINA
160 GOSUB 380
170 REM LINHA 190 ESTABELECE L COMO
180 REM COMPRIMENTO DA PAREDE EM MM
190 LET L = D2
200 PRINT "ALTURA DA PAREDE NO. "; X; "EM
  METROS"
210 REM LINHAS 230 A 250 ESTABELECE H
220 REM COMO ALTURA DA PAREDE EM MM
230 INPUT D
240 GOSUB 380
250 LET H = D2
260 REM LINHA 270 ESTABELECE A3 COMO AREA
  DA PAREDE
270 LET A3 = L * H
280 REM S (SUBTOTAL) E A AREA DA PAREDE
  DIVIDIDA
290 REM PELA AREA DO AZULEJO
300 LET S = A3/A2
310 REM T (TOTAL) TEM O NOVO SUBTOTAL
320 REM ACRESCENTANDO CADA VEZ ATRAVES
  DO LOOP

```

```

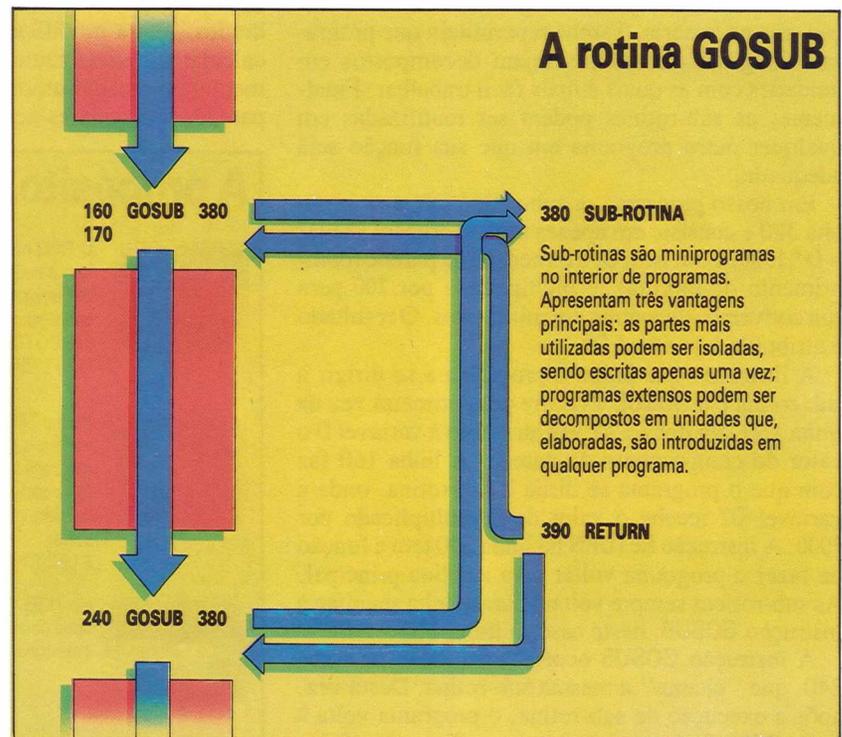
330 LET T = T + S
340 NEXT X
350 REM IMPRIMIR O TOTAL
360 PRINT T
370 END
375 REM SUB-ROTINA DE CONVERSAO M x MM
380 LET D2 = D*1000
390 RETURN

```

Após digitar o programa, você usará o comando SAVE para guardá-lo em fita cassete. Naturalmente, o gravador cassete deve ser preparado de acordo com as instruções do manual de seu computador.

O comando SAVE é muito simples: digite SAVE, seguido de um nome de arquivo entre aspas. Arquivo, em computação, assemelha-se a um arquivo qualquer: um programa ou um conjunto de dados são, mediante seu nome de arquivo, armazenados ou recuperados quando necessário. É sempre melhor usar um nome de arquivo que faça você se lembrar da função do programa. Como nosso programa calcula o número de azulejos necessários para revestir uma sala, podemos chamá-lo AZULEJOS. Estando o gravador cassete preparado, introduza uma fita limpa para guardar o programa.

Os gravadores com tomada para controle remoto geralmente têm motor controlado diretamente pelo computador. Caso contrário, aperte a tecla para gra-



vação e depois a de pausa. Digite o comando SAVE e o nome do arquivo; ligue o gravador soltando a tecla de pausa e pressione a tecla RETURN.

Para certificar-se de que o programa foi gravado corretamente, limpe a memória do computador digitando o comando NEW <CR>. Retroceda a fita, aperte a tecla para reprodução e carregue novamente o computador com o programa, usando o comando LOAD seguido do nome do programa em questão. Coloque: LOAD "AZULEJOS" e pressione a tecla RETURN.

Após o computador ter recebido o programa, uma mensagem na tela, como READY ou OK, indicará que o programa já está carregado. LISTe o programa e verifique se é idêntico ao que você digitou inicialmente.

GOSUB

A instrução que desvia o fluxo de um programa para uma sub-rotina é GOSUB. A sub-rotina pode ser definida como um miniprograma ou um programa no interior de outro. No programa aqui usado para ilustração, a sub-rotina é bastante simples e sua função é mostrar o princípio do procedimento.

Nosso programa destina-se a encontrar o número de azulejos para o revestimento de uma sala. Ele calcula a área da parede, depois de o comprimento e a altura serem convertidos de metros em milímetros. O total de azulejos é encontrado pela divisão da área de cada parede pela área do azulejo e a subsequente soma dos resultados. A conversão do comprimento e da altura da parede em milímetros é efetuada pela sub-rotina, que simplesmente multiplica o comprimento ou a altura (em metros) por 1000, para encontrar o equivalente em milímetros.

As sub-rotinas apresentam três vantagens. As partes do programa utilizadas freqüentemente podem ser separadas e necessitam ser escritas apenas uma vez — não importando quantas vezes a operação seja necessária. Também permitem que programas longos e complexos sejam decompostos em unidades com as quais é mais fácil trabalhar. Finalmente, as sub-rotinas podem ser reutilizadas em qualquer outro programa em que sua função seja adequada.

Em nosso programa, a sub-rotina inicia-se na linha 380 e consiste em apenas uma instrução: LET D2 = D * 1000. Com isto, D, a dimensão da parede (comprimento ou largura), é multiplicada por 100 para sua conversão de metros em milímetros. O resultado é atribuído à variável D2.

A instrução que induz o programa a se dirigir à sub-rotina é o GOSUB e ocorre pela primeira vez na linha 160. Na linha 130, era atribuído à variável D o valor do comprimento da parede. A linha 160 faz com que o programa se dirija à sub-rotina, onde a variável D2 recebe o valor de D multiplicado por 1000. A instrução RETURN na linha 390 tem a função de fazer o programa voltar para a rotina principal. As sub-rotinas sempre voltam para a linha seguinte à instrução GOSUB; neste caso, à linha 170.

A instrução GOSUB ocorre novamente na linha 240, que "chama" a mesma sub-rotina. Desta vez, após a execução da sub-rotina, o programa volta à linha 250. Embora o programa utilize uma única

sub-rotina, é possível usarmos quantas forem necessárias. Em cada caso, a instrução GOSUB terá de incluir o número da linha da sub-rotina desejada. Observe que a instrução END ocorre na linha 370, antes da sub-rotina. END indica o fim do programa principal e serve igualmente para evitar que, após o término do programa, o processamento continue, invadindo as sub-rotinas.

Este é um programa um pouco mais extenso do que os já apresentados neste curso, mas sua complexidade não é maior. Experimente acompanhá-lo linha por linha, observando o que se passa em cada estágio. Com exceção da instrução GOSUB e das sub-rotinas, ele introduz um único conceito novo: nomes de variáveis mais extensos.

Será útil traçar quadros, escrevendo neles os nomes das variáveis e os valores que adquirem em cada estágio.

Linha 300: LET S = A3/A2 em alguns casos resultará em um número com fração decimal. Experimente processar o programa fornecendo a medida do azulejo como 110 mm e o comprimento e altura da parede como 2,3 e 1,8 m, respectivamente, calculando uma única parede: a resposta deverá ser 342, 149 azulejos. Uma vez que os azulejos não são vendidos em unidades fracionárias, este resultado não é completamente satisfatório. Na próxima oportunidade, examinaremos um dos métodos de atingir resultados satisfatórios com números inteiros.

Exercícios

■ Observe o que acontecerá se você fornecer o tamanho do azulejo como sendo 0 mm. Por que ocorre uma indicação de erro? Por que não acontece o mesmo quando você fornece o comprimento da parede como 0 m? Dica: os procedimentos de multiplicação e de divisão por zero não são os mesmos; verifique isso em sua calculadora.

■ O programa só dá certo no caso de azulejos quadrados. Tente modificar as linhas de 30 a 60 para calcular a área de azulejos retangulares (o procedimento é o mesmo usado para o cálculo da área das paredes retangulares neste mesmo programa).

A propósito...



O TK83 não possui este comando; por isso é necessário outro procedimento para se desviar das linhas 380 e 390, o que é conseguido pela mudança da linha 370: 370 GOTO 400 e acréscimo da linha 400 PRINT "END".



Para o TK83 é necessário que todas as variáveis sejam definidas antes da realização de cálculos matemáticos. Para fazer com que a linha 330 não perca seu sentido, deve-se acrescentar a seguinte linha: 5 LET T = 0.

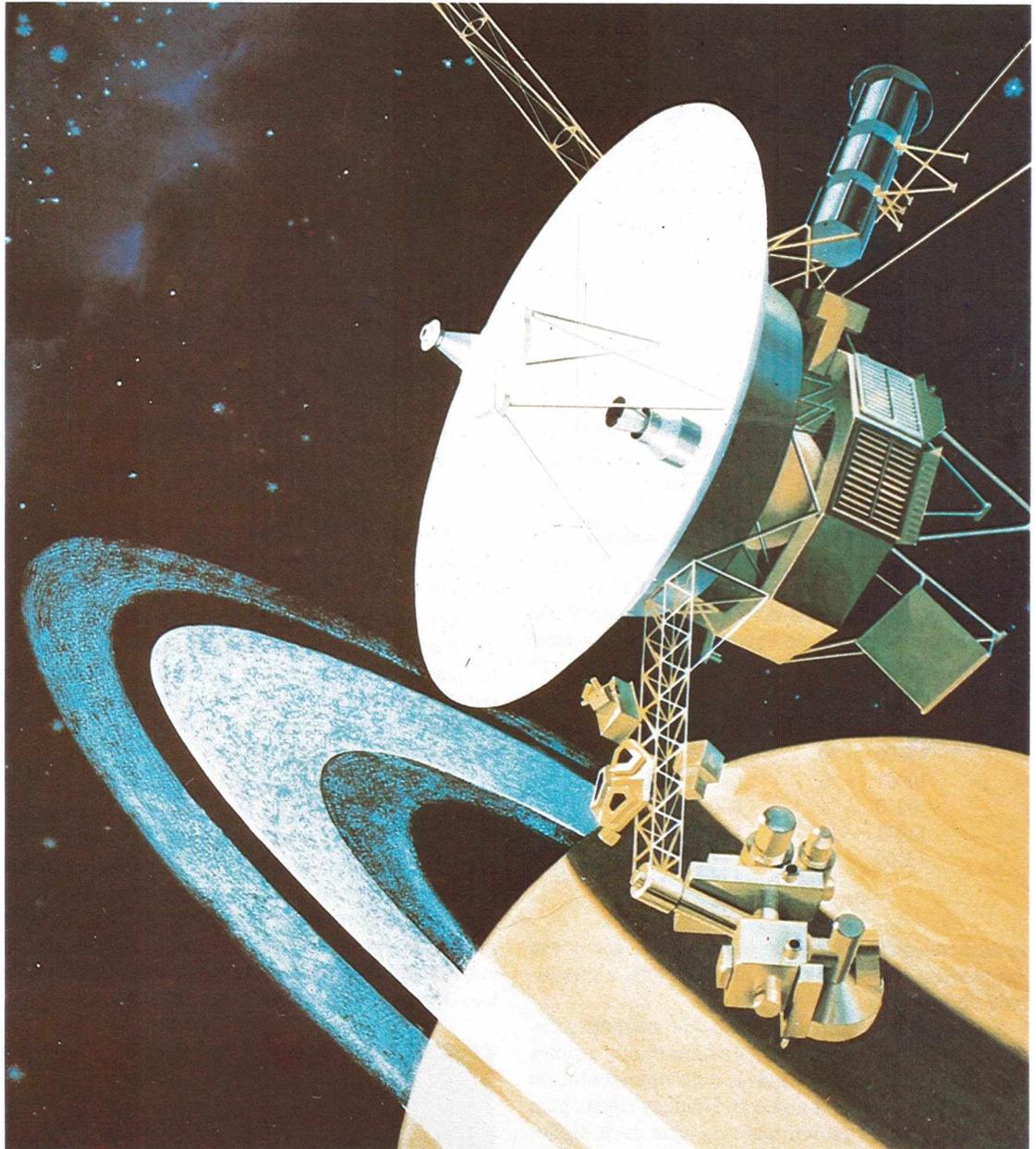


No TK83, este comando está escrito em duas palavras, bastando, entretanto, pressionar uma única tecla.



Menos igual a mais?

Circuitos simples são mais adequados ao computador. Criou-se então um artifício para efetuar subtrações através de adições.



Voyager 2

O Voyager foi a primeira sonda espacial que ultrapassou o sistema solar. Fotografou e coletou informações, enquanto o computador a bordo convertia os dados em dígitos binários, que eram enviados à Terra a uma velocidade de emissão de 116.000 bits, em média, por segundo, e processados pelo computador da NASA, em Houston, Texas.

Na primeira parte desta série, vimos que os dígitos binários podem ser usados para representar qualquer número decimal. Os números binários têm a desvantagem de ser mais extensos que seus equivalentes no sistema decimal, porém são ideais para computadores, pois os dígitos 1 e 0 podem ser representados por voltagem positiva e por voltagem zero. Também vimos que números binários podem ser somados por meio de um procedimento muito simples.

No papel, a subtração dos números binários é tão fácil quanto a dos números decimais, e segue as mesmas regras da subtração decimal. Todavia, os projetistas de computadores perceberam há bastante

tempo que os circuitos de adição podem realizar tanto a adição como a subtração, sem necessidade de circuitos especiais de subtração. Veja agora a explicação de como isso é feito.

Complemento

Um dos métodos de representação de números negativos no computador é conhecido como complemento. Através dele, a subtração se apresenta simplesmente como um outro lado da adição. Observe o seguinte problema aritmético:

$$16 - 12 = 4 \text{ ou } 16 + (-12) = 4$$

Aqui, o número 12 é extraído de 16, porém o processo de subtração pode ser considerado uma adição: a adição de 16 ao 12 negativo. Em ambos os casos, a resposta é a mesma e a única diferença está no uso de sinais aritméticos e parênteses. Este pequeno artifício pode ser usado pelo computador para representar números negativos, simplificando o procedimento da subtração.

Imaginemos que nosso computador tenha capacidade para lidar apenas com 5 dígitos, embora os computadores possam, na verdade, lidar com milhares de dígitos. Nosso hipotético computador de 5 dígitos tem o seguinte método de trabalho: o primeiro dígito do lado esquerdo deve ser considerado isoladamente dos demais. Se o primeiro dígito for 1, estará representado o número 16 negativo, e se for 0 será simplesmente o próprio zero. Os quatro outros dígitos são positivos e seguem as convenções do código binário, apresentadas no fascículo anterior do curso.

[]
[]
[]
[]
[]
 -16 ou 0 8s 4s 2s 1 ou 0

Assim, por exemplo, o número binário 01000 corresponde ao decimal 8 e o número 10000 ao decimal -16. Mas, e o número 10100? Este inclui -16 e +4, resultando em -12.

Quantos números podem ser representados com apenas cinco dígitos, de acordo com essa convenção? O maior número positivo é 01111, ou o decimal 15, e o maior número negativo é 10000, ou -16. Com um pouco de experiência, você poderá comprovar que todos os números entre -16 e +15 podem ser representados.

Binário	Decimal	Binário	Decimal
10000	-16	11111	-1
10001	-15	00000	0
10010	-14	00001	1
10011	-13	00010	2
10100	-12		etc.
	etc.	01110	14
		01111	15

Se aumentarmos o total de dígitos com que nosso computador pode trabalhar, ampliaremos a quantidade de números.

No início do desenvolvimento da aritmética binária dos computadores, um artifício muito simples foi elaborado para encontrar-se o complemento, ou forma negativa, de um número. Há duas etapas para essa execução. Primeiramente, inverta cada dígito: onde houver um dígito 1, substitua-o por zero, onde houver um 0, por 1. Em seguida, some o número 1 ao número invertido.

Siga o método, como mostrado no exemplo abaixo. Usamos o número +12, cujo equivalente binário é 01100 (o primeiro 0 à esquerda não é estritamente necessário, já que 01100 equivale ao próprio 1100; porém, como nosso computador tem cinco dígitos, é necessário preencher todos eles).

Primeiro estágio: 01100 (= +12)
 10011
Segundo estágio: 00001 (+1)
 10100 (= -12)

Agora examinemos como nosso computador resolve o problema da subtração; por exemplo: 12 menos 4.

$$\begin{array}{r}
 +12 \text{ é } \quad 01100 \\
 -4 \text{ é } \quad \quad 11100 \text{ (usando o complemento)} \\
 \hline
 12+(-4) \quad 101000
 \end{array}$$

Observe que agora temos seis dígitos. Uma vez que nosso computador tem capacidade para registro de somente cinco dígitos, o primeiro dígito à esquerda é chamado dígito de estouro e deve ser ignorado, restando 01000 ou 8 no sistema decimal, a resposta correta. Exemplo um pouco mais complexo é: 4 menos 12.

$$\begin{array}{r}
 +4 \text{ é } \quad \quad 00100 \\
 -12 \text{ é } \quad \quad 10100 \\
 \hline
 4+(-12) \quad 11000
 \end{array}$$

Como último exemplo, examinemos o procedimento, usando dois números negativos: -3 -4 = -3 + (-4) = -7

$$\begin{array}{r}
 3 \text{ é } \quad \quad \quad 00011 \\
 \text{assim, } -3 \text{ é } \quad 11101 \text{ (usando o complemento)} \\
 \text{e } -4 \text{ é } \quad \quad \quad 11100 \\
 \hline
 \quad \quad \quad 111001
 \end{array}$$

Novamente, temos um número de seis dígitos. Deixando de lado o dígito de estouro, temos o número binário: 11001 ou -7, pelo sistema decimal.

Para essas subtrações, foi necessária apenas a adição e o artifício do complemento (que se resume na utilização dos opostos dos dígitos e na adição). Para o computador, sua vantagem consiste em poderem os dígitos ser facilmente convertidos, pelo uso de uma porta NOT (ver p. 68).

A porta NOT possui uma entrada e uma saída. É uma porta "teimosa" porque, para qualquer valor fornecido, a saída será seu oposto. Assim, se a entrada for 1, a saída será 0 e vice-versa. Esta propriedade de "inversão" é exatamente o recurso necessário para o primeiro estágio do artifício do complemento. Na próxima parte do curso, examinaremos como a adição é efetuada por um computador, com o uso de uma combinação de portas lógicas.

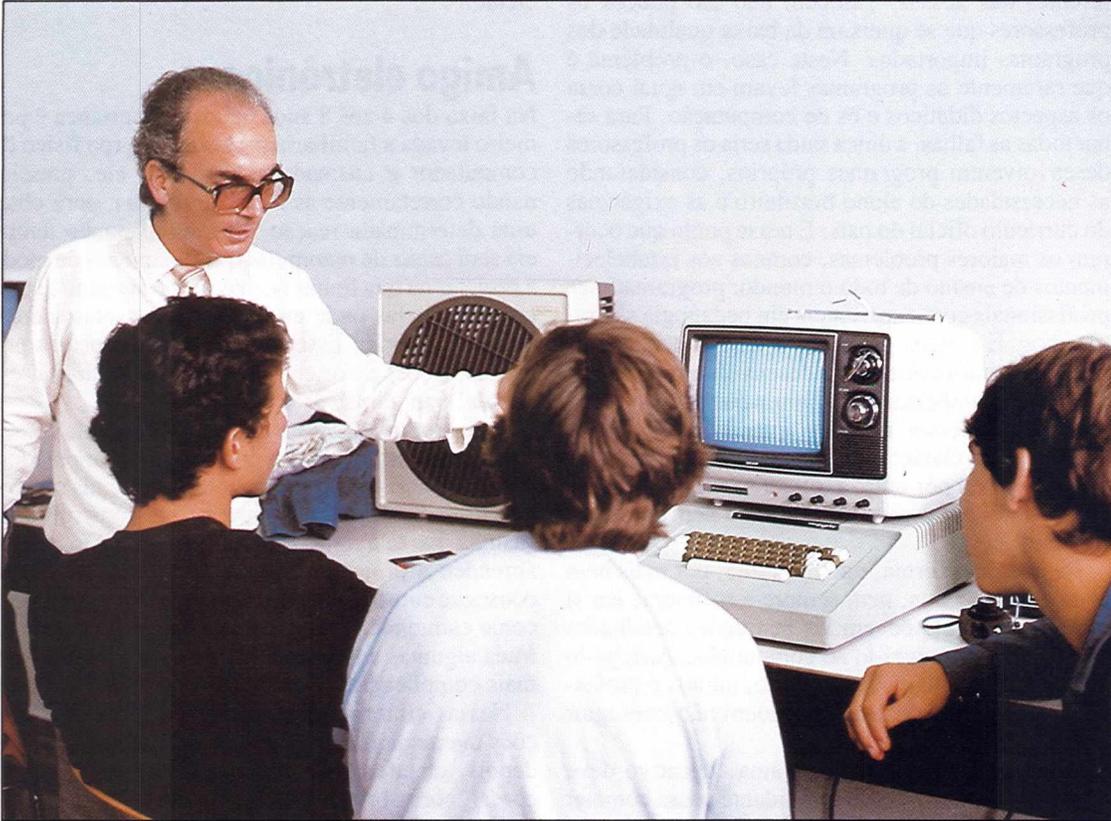


Pontos e traços

O código Morse é um dos exemplos mais antigos do código binário em eletrônica. Em 1837, o primeiro telégrafo elétrico foi instalado em Londres, com cabos de 3 km de comprimento, ligando duas estações ferroviárias. No mesmo ano, Samuel Morse, nos EUA, apresentou seu código para transmissão de mensagens. Cada letra era a combinação de dois tipos de sinal: ponto e traço.



Um novo aluno



Familiarizar crianças com o computador é hoje uma necessidade que poucos pais e professores discutem. Todos compreendem que num futuro próximo essas máquinas serão parte de nosso dia-a-dia, tanto como são hoje a calculadora de bolso e o automóvel. Pensando nisso, muitas escolas já começam a utilizar tais equipamentos na pré-escola, para fornecer aos alunos as primeiras noções de orientação espacial. Mais tarde, o computador facilita a alfabetização e o aprendizado de aritmética; depois disso, acompanhará o aluno até a universidade.

As salas de aula estão sofrendo mudanças: aos poucos, o microcomputador abre caminho entre livros e cadernos.

Muitos pais, em dúvida sobre estimular ou não a convivência de seus filhos com microcomputadores, buscam auxílio na pedagogia. Aí encontram dois tipos de opiniões, opostas uma à outra. A primeira vê a invasão dos micros como um modismo passageiro, talvez pernicioso. A segunda considera que os computadores ajudam a criança a soltar a imaginação e a raciocinar logicamente.

De um modo ou de outro, cada vez mais os jovens se interessam vivamente pelos microcomputadores. E muitos pais — geralmente usuários de computadores — ajudam os filhos na compra e na utilização desses equipamentos. No Brasil, entre 1980 e 1983, o mercado dos micros saltou de 8.000 para mais de 65.000 equipamentos vendidos. E foi possível detectar que a esmagadora maioria dos compradores tinha 30 anos de idade ou menos.

Em várias escolas brasileiras, o computador já faz parte do dia-a-dia dos alunos. Nesses estabelecimentos, localizados geralmente nas capitais dos Estados, existem dois tipos de atitude em relação ao uso do computador na escola: no primeiro caso, os alunos podem, se quiserem, seguir um curso de informática e aprender alguma linguagem de computador — geralmente LOGO ou BASIC. No segundo caso, os computadores são integrados aos currículos escolares normais, como auxiliares de ensino em al-

gumas disciplinas. Frequentemente os pequenos alunos contam apenas 5 anos de idade — às vezes menos — quando têm o primeiro contato com o computador, que atua como auxiliar na alfabetização, no ensino dos primeiros conceitos de aritmética ou para fazer a criança adquirir noções de orientação espacial ou de língua estrangeira.

Mesmo nos estabelecimentos de ensino que não fazem uso pedagógico da informática, o computador muitas vezes já é utilizado na área administrativa, para controlar gastos e custos da escola, pagamentos de mensalidades e, até mesmo, as notas e o comparecimento dos alunos.

Nas escolas públicas, as coisas ainda não chegaram tão longe. Mas já se começa a traçar os primeiros planos e a formar grupos de estudos e centros pilotos de educação pela informática, considerando-se como certa a presença dos computadores na escola, num futuro não muito longínquo.

Programando o futuro

No mercado brasileiro existem muitos programas pedagógicos para microcomputadores. No entanto, a grande maioria é diretamente importada dos Estados Unidos ou da Inglaterra e, por isso, vem com legendas e instruções em inglês, dificultando seu

uso por alunos brasileiros não iniciados nesse idioma. Além disso, os exemplos e figuras de apoio desses programas foram desenvolvidos para as necessidades de estudantes americanos e ingleses, bem diferentes das nossas. Também não são poucos os professores que se queixam da baixa qualidade dos programas importados. Neste caso, o problema é que raramente os programas levam em igual conta os aspectos didáticos e os de computação. Para sanar todas as falhas, a única saída seria os professores desenvolverem programas próprios, considerando as necessidades do aluno brasileiro e as exigências do currículo oficial do país. É nesse ponto que ocorrem os maiores problemas, comuns aos estabelecimentos de ensino de todo o mundo: programadores profissionais com experiência em pedagogia são raríssimos. E poucos professores têm as necessárias noções de informática para desenvolver programas eficazes. É também muito comum que um programa feito por professor funcione quando aplicado aos alunos de sua classe; mas, quando o mesmo programa é utilizado por outro professor, em outra classe do mesmo grau e série, o resultado nem sempre é satisfatório.

De qualquer forma, um programa, por mais bem elaborado que seja, nem sempre é suficiente em si mesmo. Com ele devem vir instruções detalhadas sobre como introduzi-lo no computador, carregá-lo e fazê-lo funcionar. Do contrário, alunos e professores leigos em computação podem não conseguir trabalhar com ele.

Além disso, um bom programa educativo deve prever todos os erros que o estudante possa cometer ao utilizá-lo. Isto significa que o programador precisa assegurar-se de que o aluno perceberá seu engano assim que pressionar a tecla errada. Por fim, um programa bem desenvolvido não deve executar

operações indesejadas quando o aluno cometer enganos. Esta é a parte mais difícil da programação educacional, tão importante quanto certificar-se de que o programa é um complemento pedagógico eficiente.

Amigo eletrônico

Na faixa dos 4 aos 8 anos de idade, a criança é primeiro levada a familiarizar-se com o corpo físico do computador e ensinada a lidar com ele, pressionando corretamente as teclas desejadas, para obter uma determinada reação. Em muito pouco tempo ela será capaz de manipular o equipamento de modo a projetar na tela linhas horizontais e verticais, apertando as teclas onde estão impressas setas indicadoras de direção. Esses sinais dão à criança as primeiras noções de orientação no espaço (direita, esquerda, para cima, para baixo).

Em aproximadamente quinze minutos, ela consegue formar um quadrado no vídeo do computador, isto é, já inicia um programa. Numa segunda sessão, sugere-se à criança que experimente construir uma casinha na tela utilizando as linhas simples que aprendeu a projetar. Com um pouco de ajuda ela consegue construir a forma sugerida e muitas outras, como caminhõezinhos e várias figuras geométricas. Mais algumas aulas e poderá criar figuras cada vez mais complicadas.

Nessas crianças muito novas, as primeiras reações diante da máquina são de intensa curiosidade, depois, de afeição: o computador sempre "concorda" com a brincadeira escolhida por seu pequeno mestre e nunca se cansa de repeti-la vezes sem conta.

Aumentando a faixa etária para 12 ou 13 anos, o sentimento dominante em relação ao computador é

Muitos acreditavam que o uso sistemático do computador transformasse jovens normais em pessoas introvertidas e pouco sociáveis. Hoje em dia, pode-se observar justamente o contrário: a "geração do computador" é descontraída, entusiasmada e muito amistosa, embora raciocine mais logicamente e seja capaz de maior concentração, se isso for necessário. Com o uso do computador, as aulas parecem sessões de laboratório, com um trabalho de equipe e muito debate em torno do objeto estudado.





de franca admiração: o computador sabe tudo e nunca erra. Com o tempo, porém, essas crianças percebem que a máquina só faz o que lhe é ordenado pelo programa, segundo as instruções do operador — que é a própria criança. Com isso, a máquina deixa de ser uma caixinha mágica para tornar-se um ajudante fiel em jogos e brincadeiras, e um instrumento capaz de auxiliar na resolução dos problemas escolares mais complicados.

Os alunos do segundo grau usam com frequência o computador para reproduzir graficamente seus problemas de matemática, facilitando assim a resolução. Desse modo, termos como representação gráfica de funções matemáticas, indeterminação da função, levantamento da função inversa e resolução da equação de n graus tornam-se parte do vocabulário corriqueiro desses jovens.

Não é só no ensino da matemática que o computador pode ser útil. No entanto, nas outras disciplinas existe maior dificuldade para obter ou desenvolver bons programas. Para solucionar o problema, as escolas estão ministrando cursos de computação a seus professores, para que eles possam desenvolver com eficiência seus próprios programas; nesses casos, o computador torna-se um auxiliar de ensino verdadeiramente útil.

Apesar de todas as dificuldades, existem muitos programas educativos bons, para serem usados em casa ou na escola. Se os pais pretendem usar um desses programas em casa, devem escolhê-lo cuidadosamente, procurando conhecer o funcionamento do computador. Um programa adequado certamente deixará a criança fascinada e desejosa de aprender cada vez mais.

O tipo mais comum de programa educacional é o chamado de “exercício e prática”, no qual o computador dá vários exemplos de um determinado problema à criança, solucionando-os em seguida; depois, propõe a ela que solucione outros problemas, semelhantes aos exemplificados. A maioria dos programas faz uma tabela dos resultados finais alcançados pela criança e, se ela tem muitos acertos, recebe elogios; se erra muito, é gentilmente estimulada a “tentar de novo”.

A escolha certa

Se você ainda não possui um microcomputador mas pretende comprar um, para uso principalmente pedagógico, verifique antes de que tipo é o computador usado na escola de seu filho (se lá houver um). Prefira um modelo igual e, desse modo, a criança poderá usar em casa os mesmos programas que utiliza na escola. A maioria dos estabelecimentos de ensino fornece cópias de seus programas aos alunos. Caso a escola de seu filho proceda assim, essa “lição de casa” adicional será muito proveitosa para ele. No entanto, se você já adquiriu um micro e ele não é compatível com o computador da escola, não se preocupe — seu filho pode tirar proveito de ambas as experiências em computação.

Naturalmente, o número de programas existentes no mercado atual destina-se, sobretudo, aos computadores mais sofisticados. Mas há também muitos programas bons para equipamentos mais modestos. Tais programas podem ser adquiridos no mesmo lo-

cal onde você adquiriu seu micro ou em lojas de software. Estas últimas costumam anunciar seus produtos em revistas especializadas no ramo e nas lojas que vendem microcomputadores.

Para crianças com menos de 8 anos de idade, a maior parte dos programas se concentra em exercícios elementares de alfabetização e aritmética, ou



O computador é um mestre habilidoso e capaz. Uma de suas maiores virtudes é permitir a cada aluno trabalhar em seu próprio ritmo, sem medo de atrapalhar a classe inteira ou de parecer menos inteligente do que os outros. Além disso, sua simples utilização deixa a criança notar como um problema é analisado e resolvido.

em jogos que desenvolvem a coordenação motora.

À medida que a criança cresce, aumentam a quantidade e a complexidade de programas adequados a ela. Para a faixa etária situada entre os 8 e os 11 anos, por exemplo, já existem à venda programas que ensinam teoria dos conjuntos, explicam a composição do sistema solar ou dão noções de música e língua estrangeira.

Entretanto, os alunos mais beneficiados pelo uso do computador são os que estão no final do segundo grau. Para eles, há um sem-número de programas destinados a transmitir todos os tipos de conceitos abstratos, por meio de boa resolução gráfica e recursos de animação. Muitos programas de geometria (definição de ponto, retas e planos e suas relações no espaço) e álgebra já foram traduzidos para o português e são facilmente obtidos em lojas do ramo.

Para os universitários, há bons programas em quase todas as disciplinas. Alguns dos melhores estão na área de química. Usando um deles, o aluno pode, por exemplo, visualizar a estrutura molecular de um composto ou os átomos de cada molécula em três dimensões. O conjunto move-se lentamente no espaço, permitindo a observação detalhada das ligações atômicas.

Contudo, é preciso não esquecer que para escolher corretamente um desses programas para uma criança deve-se pedir o auxílio de seu professor: o estudo feito em casa não deve conflitar com o que é dado na escola.



Código decifrado

Experimente digitar uma instrução qualquer na linguagem do seu computador: quase imediatamente, um único programa interno irá convertê-la em um código que a máquina entenda.

Embora os computadores pareçam semelhantes na execução de funções, cada modelo é diferente de outro. Há os que trabalham com programas já integrados, e existem os que requerem programas a ser "lidos" de um disco externo ou fita cassete.

Algumas máquinas contêm um único programa abrangente, que tanto permite a entrada do programa como o uso de instruções diretas, tais como SAVE ou LOAD. Outros modelos exigem programas separados para realizar essas funções.

Há, entretanto, princípios básicos segundo os quais operam os microcomputadores mais comuns. O movimento de informação entre elementos de memória externa (discos e fitas) e a tela é, neste caso, controlado pelo teclado. Toda máquina pode, ainda, comunicar-se com aparelhos externos, como impressoras, plotters ou instrumentos científicos. A maioria dos micros permite ainda que o usuário escreva programas em linguagens similares ao inglês, como o BASIC, por exemplo.

Ao digitar um programa BASIC no teclado do seu computador, um programa chamado "sistema operacional" leva o que você digitou para a tela e para um programa interpretador BASIC. Isto significa que três programas estão sendo processados ao mesmo tempo: o sistema operacional, o interpretador BASIC e o seu próprio programa.

Quando você opera o computador, os três programas parecem funcionar ao mesmo tempo. Cada instrução BASIC é traduzida pelo programa interpretador e, uma a uma, as instruções resultantes em código de máquina são transferidas para o microprocessador. Ao mesmo tempo, o sistema operacional verifica o teclado para a entrada de dados e, quando for o caso, ele os envia para a tela ou para qualquer outro periférico especificado.

Se uma das instruções de seu programa exigir que algo seja impresso ou gravado no disco, por exemplo, o programa interpretador enviará um pedido ao sistema operacional, a fim de que isso seja realizado.

A incrível velocidade do microprocessador dá a idéia de que várias coisas estão acontecendo simultaneamente. Ele processa instruções do sistema operacional e do programa interpretador com tal rapidez que ambos parecem estar ocorrendo ao mesmo tempo.

Algumas máquinas operam com maior velocidade, permitindo que a chegada de dados "interrompa" o processo normal. Nesse caso, o sistema operacional não precisa checar as várias fontes externas de dados como o teclado e as unidades de disco.

Um tipo de processador de palavra menos sofisticado é chamado "editor". Sua qualidade tende a variar muito, e provavelmente você encontrará um bom editor incluído no interpretador BASIC do seu microcomputador.

Em vez de usar um interpretador, para seus programas em BASIC, você poderia utilizar um compilador. O interpretador traduz as informações à medida que aparecem; o compilador traduz todo o programa para código de máquina de uma só vez. Os programas "compilados" funcionam mais rapidamente que os "interpretados".

Linguagens diferentes

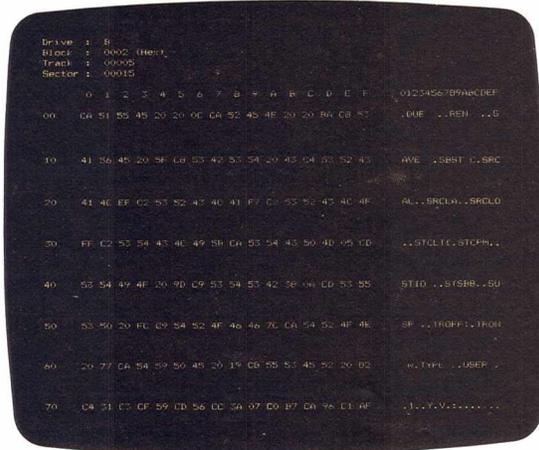
A linguagem BASIC é agradável e simples para se escrever um programa. Tem a vantagem de ser similar ao inglês comum, portanto ideal para iniciantes. Mas o programador mais audacioso pode fazer seu programa ser executado com maior rapidez usando uma linguagem de montagem (assembly) como o Assembler. Esse tipo de linguagem é bem diferente do inglês e o programador precisa saber como o microprocessador realiza suas funções. Cada instrução que você dá ao seu computador possui uma equivalência direta em código de máquina. Uma linguagem assembly compõe-se de abreviaturas como MVI (Move Immediate) ou JZ (Jump on Zero). Estes códigos são usados para ajudar o programador a lembrar-se da função dessas instruções.

Se você dominar uma linguagem desse tipo, seu próximo passo será trabalhar com o código de máquina. Isso não lhe trará problemas, a não ser que você realmente precise economizar minúsculas frações de segundo do tempo de execução de um programa.

O código de máquina nos microcomputadores geralmente é escrito na forma chamada hexadeci-

Sistema operacional de disco
(DOS - Disk Operating System)
Quando os programas são armazenados num disco flexível, a informação é distribuída ao acaso, por toda a superfície do disco. O sistema operacional de disco é um programa que guarda, automaticamente, a localização de cada byte de informação.

A ilustração mostra a informação armazenada numa seção de disco. A representação é feita na forma hexadecimal à esquerda; e o caractere equivalente é mostrado à direita.





mal, que é uma numeração de base 16. Você conta de 0 a 9 normalmente, então passa a usar as letras de A até F para os números de 10 a 15. Cada posição de memória contém 8 dígitos binários (bits), que podem ser representados por um par de dígitos hexadecimais.

Por exemplo, a configuração binária 01011101 seria primeiramente dividida em duas partes: 0101 e 1101. Estas partes seriam traduzidas nos números decimais 5 e 13, equivalentes a 5 e D em hexadecimal. Assim, 01011101 é referido como 5D, quando se programa em código de máquina. Este é o modo mais lento de desenvolver programas, mas reduz seu tempo de execução. A seguir, alguns exemplos de programas típicos:

BASIC

```
100 INPUT "INFORME HORAS TRABALHADAS";
    HORAS
200 PAGAMENTO = HORAS * TAXA
```

A primeira linha mostra a mensagem na tela e pede ao usuário que informe o número de horas trabalhadas. O computador aceita a mensagem; e a seguir, na segunda linha, multiplica esse número pela taxa de pagamento (informada anteriormente), e dá o valor total como resultado.

LINGUAGEM ASSEMBLY

```
MVL C, 01
CALL 05
```

A primeira instrução acima move o valor 1 para uma parte da memória do micro chamada "registrador C". A segunda instrução pede que o sistema operacional assuma o controle. Depois, o sistema operacional transfere novamente o controle para o programa. Abaixo, uma tradução direta das linhas acima:

CÓDIGO DE MÁQUINA

```
0E01
CD0500
```

A tradução em código de máquina das duas declarações BASIC compreenderia muitos comandos. É melhor deixar que um compilador ou um interpretador faça esse trabalho.

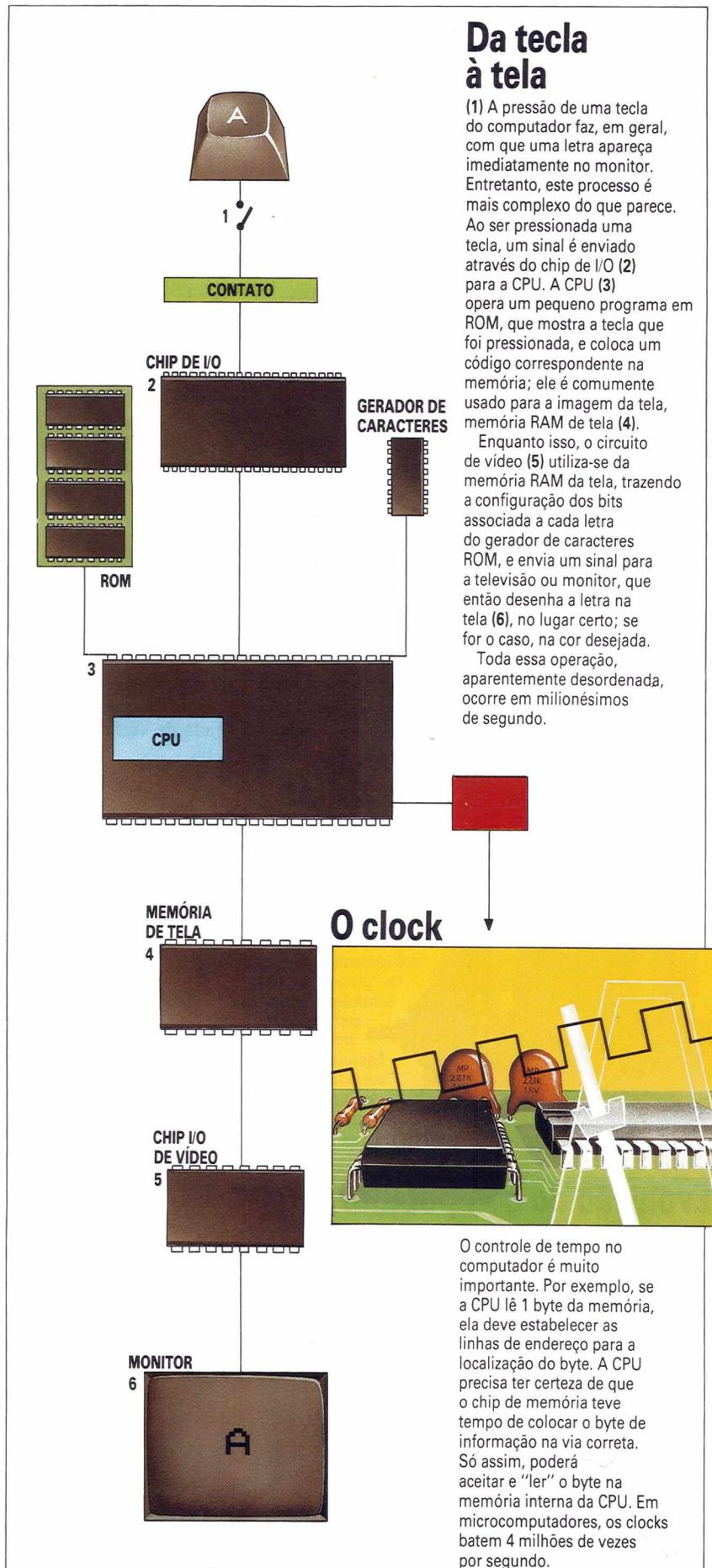
Alguns comandos de sistemas operacionais são tão enigmáticos como os programas em linguagem assembly. Abaixo, alguns exemplos de CP/M (Control Program for Microcomputers):

```
DIR *.BAS
```

A frase acima quer dizer: "Liste todos os arquivos do disco, cujo sufixo é BAS". DIR é uma abreviação de Directory (Diretório).

Apesar da aparência estranha, o CP/M é usado em milhões de máquinas, pois apresenta muitas vantagens para os profissionais que escrevem software. Os programas podem ser facilmente adaptados de uma máquina para outra, mas escritos de modo que transfiram o controle dos discos, do teclado, da impressora e da tela para o CP/M.

Alguns equipamentos oferecem diferentes versões de sistemas operacionais. Nestes casos, você pode optar, no momento da compra, por aquele que mais lhe convier, sempre levando em conta o tipo de aplicação e a disponibilidade de softwares prontos compatíveis com o sistema escolhido.



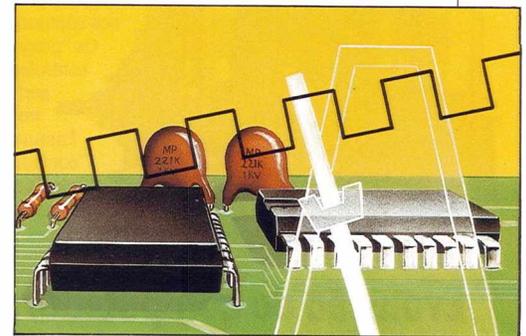
Da tecla à tela

(1) A pressão de uma tecla do computador faz, em geral, com que uma letra apareça imediatamente no monitor. Entretanto, este processo é mais complexo do que parece. Ao ser pressionada uma tecla, um sinal é enviado através do chip de I/O (2) para a CPU. A CPU (3) opera um pequeno programa em ROM, que mostra a tecla que foi pressionada, e coloca um código correspondente na memória; ele é comumente usado para a imagem da tela, memória RAM de tela (4).

Enquanto isso, o circuito de vídeo (5) utiliza-se da memória RAM da tela, trazendo a configuração dos bits associada a cada letra do gerador de caracteres ROM, e envia um sinal para a televisão ou monitor, que então desenha a letra na tela (6), no lugar certo; se for o caso, na cor desejada.

Toda essa operação, aparentemente desordenada, ocorre em milionésimos de segundo.

O clock



O controle de tempo no computador é muito importante. Por exemplo, se a CPU lê 1 byte da memória, ela deve estabelecer as linhas de endereço para a localização do byte. A CPU precisa ter certeza de que o chip de memória teve tempo de colocar o byte de informação na via correta. Só assim, poderá aceitar e "ler" o byte na memória interna da CPU. Em microcomputadores, os clocks batem 4 milhões de vezes por segundo.

Do ábaco ao micro

Os modernos chips dos computadores devem sua existência ao trabalho de inventores geniais, durante três séculos!

Quando alguém está apaixonado, sente-se capaz de conseguir qualquer coisa. Este foi, pelo menos, o pensamento de Charles Babbage em 1830. Em consequência, ele quase construiu o primeiro computador do mundo, cem anos antes de isto se tornar realidade.

O projeto de Babbage apresentava desvantagens; uma delas era o fato de que o computador tinha de ser mecânico, e outra era a própria precariedade da engenharia da época. Apesar desses problemas, Charles Babbage construiu um aparelho que impressionou muito o governo inglês; a partir daí, o inventor recebeu vultosas subvenções.

Entretanto, a história da computação começou muito antes. O computador é uma máquina capaz de efetuar cálculos com um grupo de números, "lembrar" o que foi computado e, ainda, pode ser adaptado para efetuar outros cálculos com um novo grupo de números. O primeiro "modelo" foi o ábaco,

usado desde 2000 a.C. e ainda encontrado no Japão e em outros países. É um tipo de computador em que se pode ver claramente a soma nos fios: a posição das contas forma uma "memória" da soma. Mas não são automáticos e não comportam números muito extensos.

Blaise Pascal, matemático, físico e filósofo francês, inventou a primeira calculadora mecânica em 1642, aos 18 anos de idade, talvez para ajudar o pai, fiscal de impostos. A calculadora trabalhava perfeitamente; transferia os números da coluna de unidades para a coluna de dezenas por um dispositivo, parecido com o velocímetro de automóvel. Pascal chamou a invenção de pascalina.

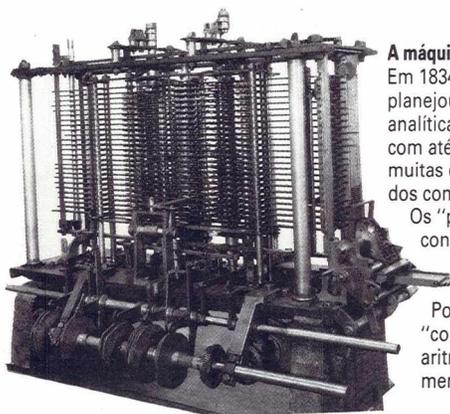
Apesar de o aparelho não ter vendido bem, despertou grande interesse científico, e nos anos que se seguiram vários projetos foram feitos com intuito de aperfeiçoar essa primeira calculadora. Entretanto, nada de significativo aconteceu até que Babbage e

Blaise Pascal

A pascalina foi a primeira calculadora mecânica do mundo, planejada por Blaise Pascal em 1642.

Originalmente, ele pretendia construir uma máquina que realizasse as quatro operações fundamentais. O instrumento usava uma agulha para mover as rodas, e um mecanismo especial levava dígitos de uma coluna para outra.

Pascal recebeu uma patente do rei da França para que lançasse a calculadora no comércio. O engenho, apesar de útil, não obteve aceitação.



A máquina analítica

Em 1834, Charles Babbage planejou uma máquina analítica capaz de computar com até 80 dígitos; tinha muitas das características dos computadores modernos.

Os "programas" eram controlados por cartões perfurados e os resultados impressos automaticamente.

Possuía também um "contador" aritmético e dispositivos de memória separados.



Evolução do computador

1000 a.C.
ÁBACO
Ainda usado atualmente.
"Armazena" números reunindo contas.

1614
LOGARITMOS
John Napier criou este recurso, que reduzia a divisão à subtração e a multiplicação à adição.

1623
BINÁRIO
Francis Bacon foi o primeiro a usar aritmética de base 2.

1642
MÁQUINA DE SOMAR
Blaise Pascal inventou a calculadora mecânica que chamou de pascalina.

1671
CALCULADORA
Gottfried von Leibniz fez uma máquina que efetuava multiplicações e divisões.

1802
CARTÕES PERFURADOS
Joseph Jacquard construiu um tear que memorizava os modelos da fábrica em cartões perfurados.

1822
APARELHO DE DIFERENÇA
A primeira calculadora matemática, de Charles Babbage.

1000 BC

1614

1623

1642

1671

1802

1822



Ada Lovelace começaram a considerar melhor o problema.

Charles Babbage nasceu numa família rica, em 1791. Foi talentoso matemático e, por ficar frustrado em apenas corrigir os erros que encontrava nas tabelas de logaritmos, decidiu construir uma máquina que eliminasse o trabalho dos cálculos.

Em 1822, ele apresentou à Sociedade Real de Astronomia o primeiro modelo de uma máquina de "diferença", capaz de fazer os cálculos necessários para elaborar uma tabela de logaritmos. O nome da máquina derivou de uma técnica de matemática abstrata, o método das diferenças. Com o incentivo da sociedade, ele continuou a trabalhar no aperfeiçoamento da máquina.

Com Ada Lovelace, filha de Lord Byron, iniciou um projeto mais ambicioso para construir uma "máquina analítica". Foi projetada para calcular valores de funções matemáticas bem mais complexas que as funções logarítmicas.

O projeto apresentava inúmeros problemas, e simplesmente não funcionava. Os desenhos que restaram nos mostram que o aparelho era imenso e ocupava toda a oficina de Babbage. As centenas de engrenagens, barras e rodas apresentavam problemas ao serem acionadas, pois a tecnologia do metal da época não era suficientemente boa. Ao construir um modelo menor, as pequenas imperfeições podiam ser desprezadas; mas, feita a máquina no tamanho real, essas imperfeições tornaram-se enormes.

Contudo, Babbage estava no caminho certo; se tivesse montado as peças de modo satisfatório, sua máquina analítica provavelmente teria funcionado.



A primeira programadora
A condessa Ada Lovelace, companheira de Charles Babbage e filha de Lord Byron, é uma das poucas mulheres a figurar na história da computação. Matemática talentosa, Ada compreendeu a máquina analítica de Babbage e escreveu um dos melhores relatos sobre ela. Criou programas para a máquina, tornando-se a primeira programadora de computador do mundo.

Grande parte da arquitetura lógica e da estrutura dos computadores atuais provém dos projetos de Charles Babbage, que é lembrado como um dos fundadores da computação moderna.

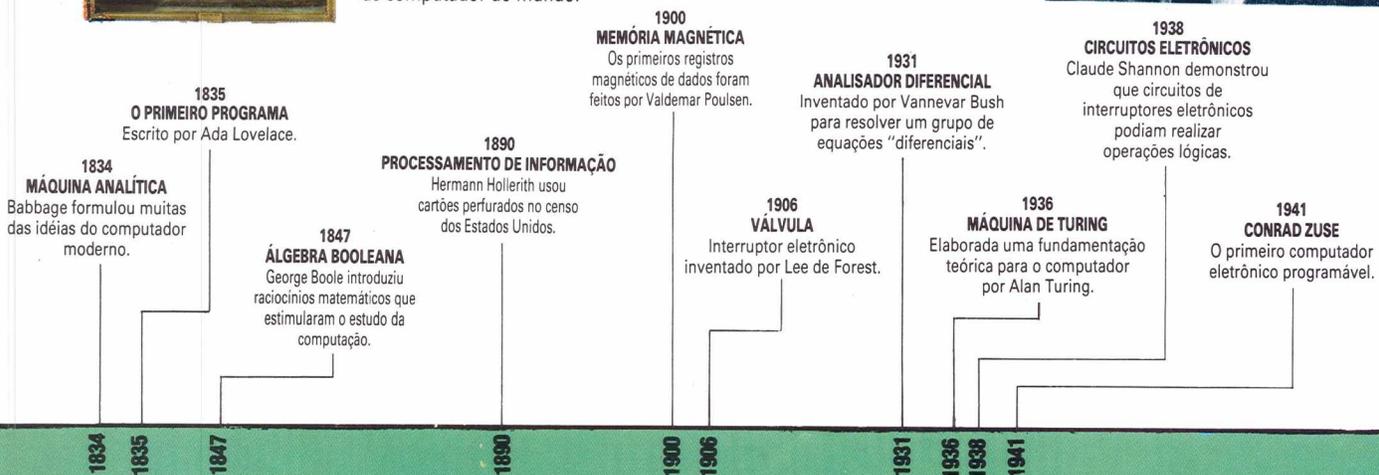
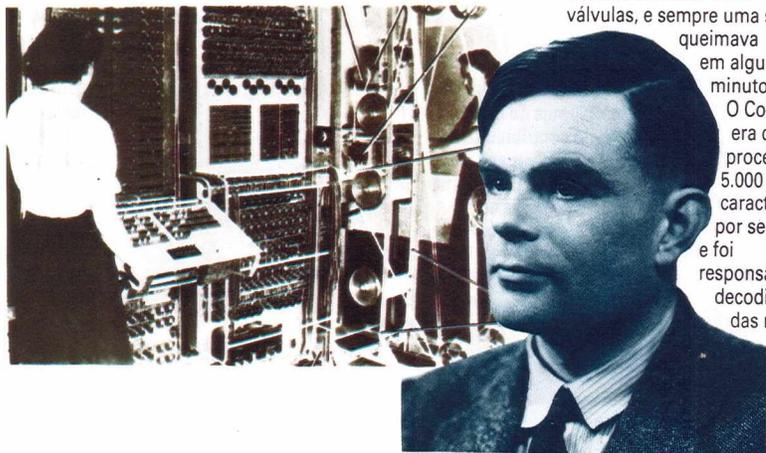
Durante seu trabalho, Babbage observou que se podia "programar" ou "ensinar" sua máquina a fazer tarefas matemáticas. Se ele tivesse podido testar isso, ou se tivesse criado uma máquina para fazê-lo, os vitorianos teriam movido o império por meio do computador a vapor.

A estrutura lógica

Só por volta de 1936, as idéias de Babbage foram comprovadas, quando um jovem matemático de Cambridge, Alan Turing, publicou um artigo, pouco conhecido, *On computable numbers*. O nome de Turing é quase desconhecido para o público, mas sua contribuição foi fundamental para o desenvolvimento de idéias que ocorreriam antes de o computador propriamente dito tornar-se realidade. Os cientistas admitiam que a matemática não era uma arte misteriosa, e sim uma ciência inteiramente relacionada com regras lógicas. Se uma máquina recebesse essas regras e o problema a ser solucionado, ela seria capaz de resolvê-lo. No entanto, os esforços dos mais competentes matemáticos foram inúteis para desenvolver tal máquina. Turing decidiu examinar o impasse de outra maneira. Verificou os tipos de problemas que uma máquina poderia resolver seguindo regras lógicas, e tentou fazer uma lista de todos eles.

Alan Turing e o Colossus

Alan Turing demonstrou que um conjunto de estruturas simples poderia resolver qualquer problema complexo. Ele e sua equipe desenvolveram o Colossus, um dos primeiros computadores do mundo, visto aqui em operação durante a Segunda Guerra Mundial. Essa máquina enorme continha 1.500 válvulas, e sempre uma se queimava em alguns minutos. O Colossus era capaz de processar 5.000 caracteres por segundo, e foi responsável pela decodificação das mensagens alemãs "Enigma".





Se abrangessem toda a matemática, a questão estava resolvida.

Turing liderou uma equipe de pesquisa na Inglaterra e desenvolveu a mais secreta invenção da Segunda Guerra Mundial, o Colossus, primeiro computador eletromecânico do mundo, que pôde decifrar os códigos alemães de mensagens "Enigma", durante a guerra.

Depois da guerra, Turing colaborou no projeto do primeiro computador dos Estados Unidos, o ENIAC, desenvolvido na Universidade da Pensilvânia. Ainda imperfeito, era composto de 18.000 válvulas. De dois em dois minutos uma válvula se queimava!

Uma das razões de ser desconhecido o nome de Turing foi por trabalhar para o serviço secreto inglês. O governo não liberou detalhes sobre o trabalho pioneiro de Turing até 1975.

O desenvolvimento do computador continuou, mas só com a invenção do transistor de silício, em 1947, tornou-se possível aumentar a velocidade das operações na computação.

Os transistores substituíram as válvulas: são mais rápidos, mais exatos e não geram calor. Assim como as válvulas, são interruptores eletrônicos que se ligam e desligam e podem representar os algarismos 0

e 1 do código binário. Durante os anos 50 e começo dos 60, construíram-se computadores maiores e mais rápidos, usados em grandes empresas e órgãos do governo.

Em meados dos anos 60, os cientistas observaram que um circuito eletrônico funcionaria de modo igualmente satisfatório se tivesse o tamanho bem reduzido. Com bilhões de dólares despendidos na corrida espacial, os laboratórios começaram experimentando a colocação de um projeto de circuito num único chip de silício, gravando o circuito no chip. Antes do fim dos anos 60, nasceu o "circuito integrado", e com isso a computação deu um grande passo à frente.

O desenvolvimento de um circuito em um único chip levou naturalmente à construção de múltiplos circuitos em um só chip; e o resultado inevitável da colocação de vários chips juntos foi o conhecido microprocessador.

Apesar da pouca semelhança entre a tecnologia do microchip e os diversos projetos de Babbage, Ada Lovelace e o Colossus de Turing, a "arquitetura" prática criada por Charles Babbage ainda é utilizada nos microprocessadores atuais. E a teoria matemática de Turing, que possibilitou tudo isso, ainda não foi superada.

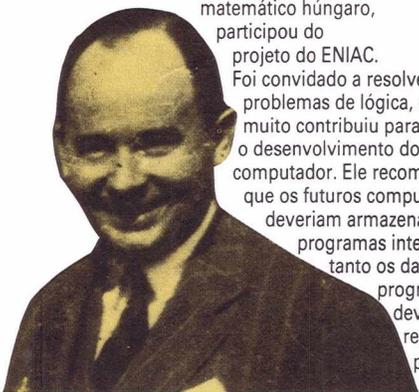
O Sinclair Spectrum

Desenvolvido 40 anos depois dos primeiros e incômodos computadores, instalados em grandes salas, o Spectrum é portátil e barato. Foi o primeiro micro em cores. Os primeiros computadores eram grandes máquinas complexas, projetadas para instituições governamentais, operadas a portas fechadas. Atualmente, graças a máquinas como o Spectrum, muitas pessoas podem ter um micro em casa.



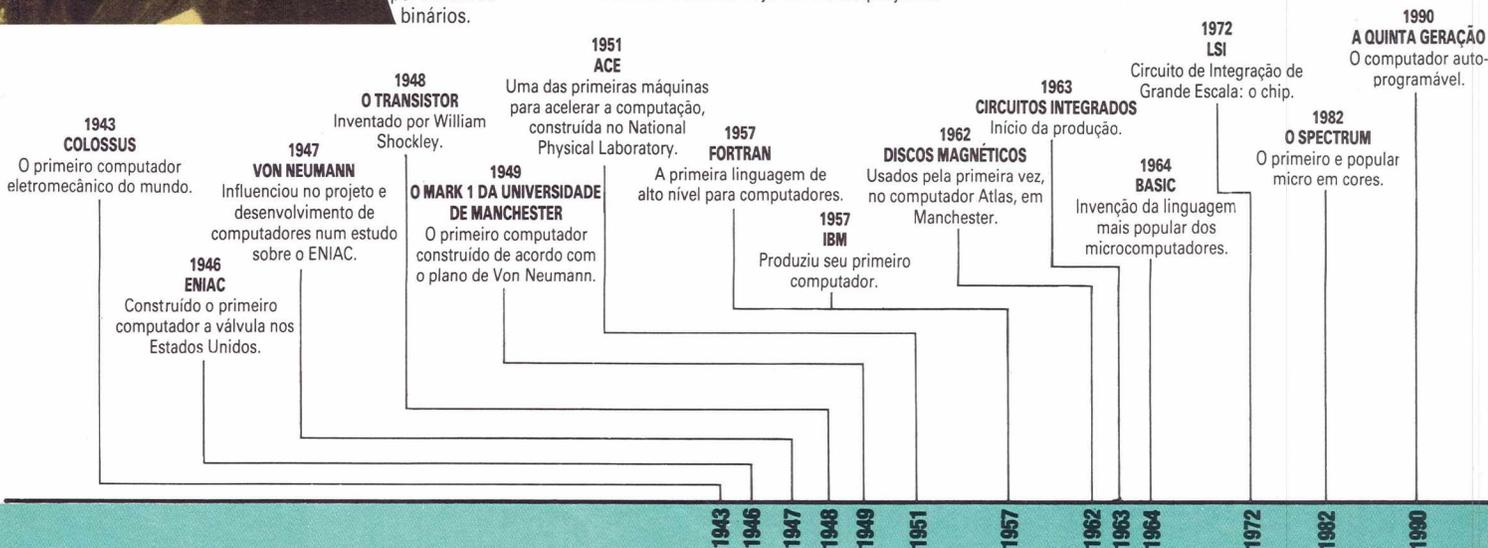
O "arquiteto" do computador moderno

John von Neumann, célebre matemático húngaro, participou do projeto do ENIAC. Foi convidado a resolver problemas de lógica, e muito contribuiu para o desenvolvimento do computador. Ele recomendou que os futuros computadores deveriam armazenar os programas internamente e tanto os dados como os programas deveriam ser representados por números binários.



Steve Wozniak

Conhecido como uma "lenda viva" no comércio da computação, Wozniak, aqui, está ao lado de seu primeiro invento, o Apple I (hoje um objeto de colecionador). Ele projetou computadores para a escola e, apesar de não ser engenheiro, empenhou-se em reduzir o tamanho e o preço do microcomputador. O Apple II, por ele desenvolvido em 1976, foi o primeiro computador doméstico a ter unidade de disco, e é um dos mais vendidos no mundo. Na Apple Corporation, Wozniak trabalha hoje em novos projetos.





Nexus 1600

Projetado e desenvolvido com a mais avançada tecnologia, o Nexus 1600 dá solução adequada às mais variadas aplicações e atua como terminal inteligente para outros computadores.

O Nexus 1600 é um microcomputador de 16 bits, de uso pessoal-profissional e empresarial, projetado e construído no Brasil. Formado por três módulos — unidade de sistema, vídeo e teclado —, pode ser utilizado de forma concentrada (módulos empilhados) ou distribuído ao redor de uma mesa.

A Unidade Central de Processamento deste equipamento (CPU) é um microprocessador 8088 de 16 bits, com velocidade de processamento de 8 MHz; opcionalmente, pode acompanhar a unidade um coprocessador aritmético 8087. Esta configuração tem 64 Kbytes de memória EPROM, 256 Kbytes de memória RAM e duas interfaces para comunicação serial.

O teclado do Nexus 1600 é serial, tipo máquina de escrever, com 86 teclas (sendo 10 de função), bloco

numérico separado, simulador de ruído mecânico e repetição automática. O design do teclado possui características que facilitam sua operação: a conexão à unidade de sistema através de cabo espiralado, o que aumenta sua mobilidade; e as dimensões reduzidas, leveza e pés retráteis, que lhe conferem maior flexibilidade no uso diário.

O monitor de vídeo possui um cinescópio móvel que se adapta ao melhor ângulo de visão do operador. Atualmente, o Nexus 1600 dispõe de um monitor monocromático, de 12 polegadas, em fósforo verde, com as cores virtuais representadas em tonalidades distintas. Este monitor é controlado por uma interface gráfica que opera em modo alfanumérico (80 x 25 ou 40 x 25) e gráfico branco e preto (640 x 200 pontos), além de aceitar conexão light-



As características ergonômicas do Nexus 1600, desenvolvidas pela Scopus, permitem uma melhor adaptação do operador à máquina. O vídeo basculante com tela anti-reflexo, o teclado leve e de inclinação variável, o cabo espiralado que liga o teclado à unidade de sistema são importantes requisitos que visam sobretudo o conforto e produtividade do usuário.



pen. (O Nexus 1600 poderá ser conectado a um monitor de vídeo colorido, de 14 polegadas, controlado pela mesma placa gráfica já descrita, oferecendo, neste caso, um modo alfanumérico colorido (16 cores, 80 x 25 ou 40 x 25), ou gráfico colorido (4 cores, 320 x 200 pontos), e o mesmo modo gráfico monocromático (640 x 200 pontos). O monitor de vídeo dispõe também de máscara anti-reflexo, que elimina cintilações indesejáveis.

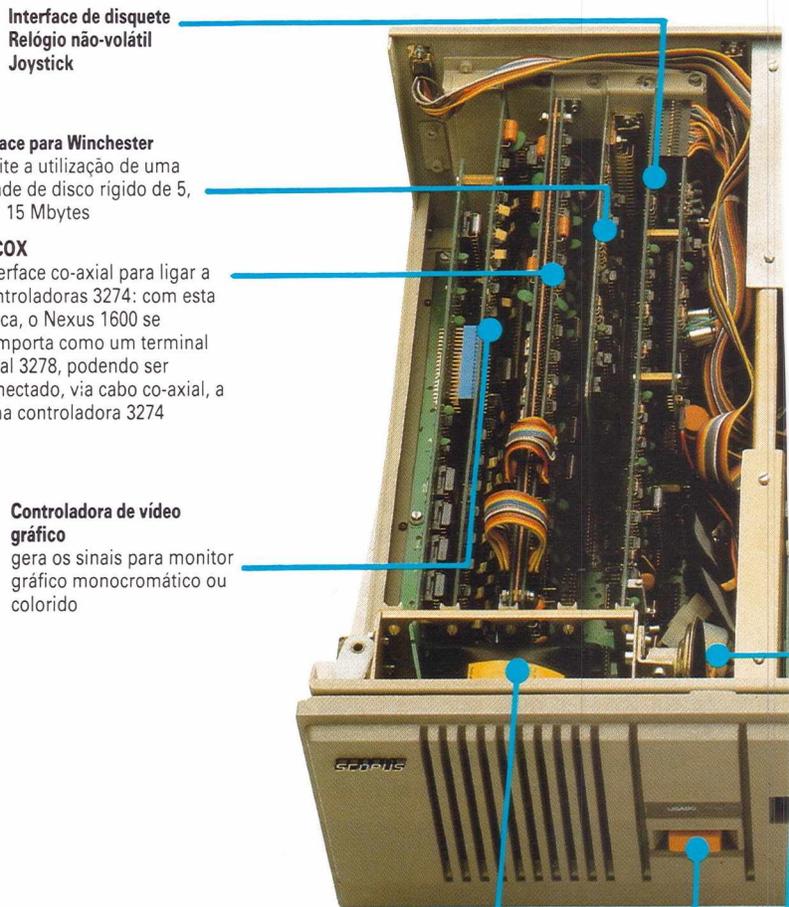
Uma placa de funções múltiplas engloba interface para impressoras e para discos flexíveis de 5 1/4 polegadas, relógio-calendário não-volátil e interface para joystick. O Nexus 1600 dispõe de mais duas placas de expansão: a primeira é uma interface para míni Winchester de 10 Mbytes, e a segunda é a INCOX, para cabo co-axial IBM 327x, que possibilita ligação direta com controladoras IBM 3274/3276, emulando os terminais IBM 3278 e 3279.

O sistema operacional do Nexus 1600 é o SISNE. Foi escrito em Pascal e, em sua versão inicial, emula o PC-DOS 1.1 da IBM. Quanto aos softwares aplicativos, o usuário pode se utilizar dos inúmeros pacotes já existentes no mercado brasileiro.

Entre novas implementações do Nexus 1600, destacam-se a possibilidade de multiprogramação (execução simultânea de vários programas) e a sua interligação em sistemas de rede com outros microcomputadores, a fim de compartilhar arquivos e periféricos, reduzindo em muito os custos e agilizando as operações de processamento.



Monitor de 12" em fósforo verde dotado de máscara anti-reflexo.



Interface de disquete
Relógio não-volátil
Joystick

Interface para Winchester
permite a utilização de uma
unidade de disco rígido de 5,
10 ou 15 Mbytes

INCOX
interface co-axial para ligar a
controladoras 3274: com esta
placa, o Nexus 1600 se
comporta como um terminal
local 3278, podendo ser
conectado, via cabo co-axial, a
uma controladora 3274

Controladora de vídeo
gráfico
gera os sinais para monitor
gráfico monocromático ou
colorido

Ventilador

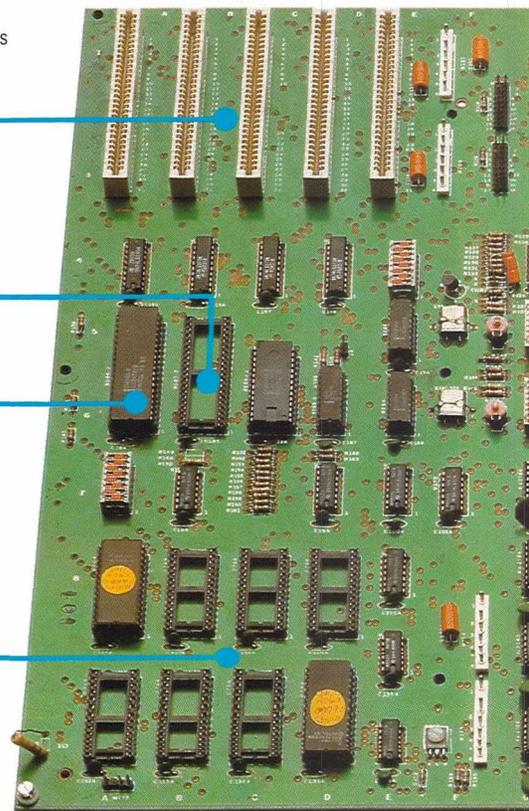
Chave liga-desliga

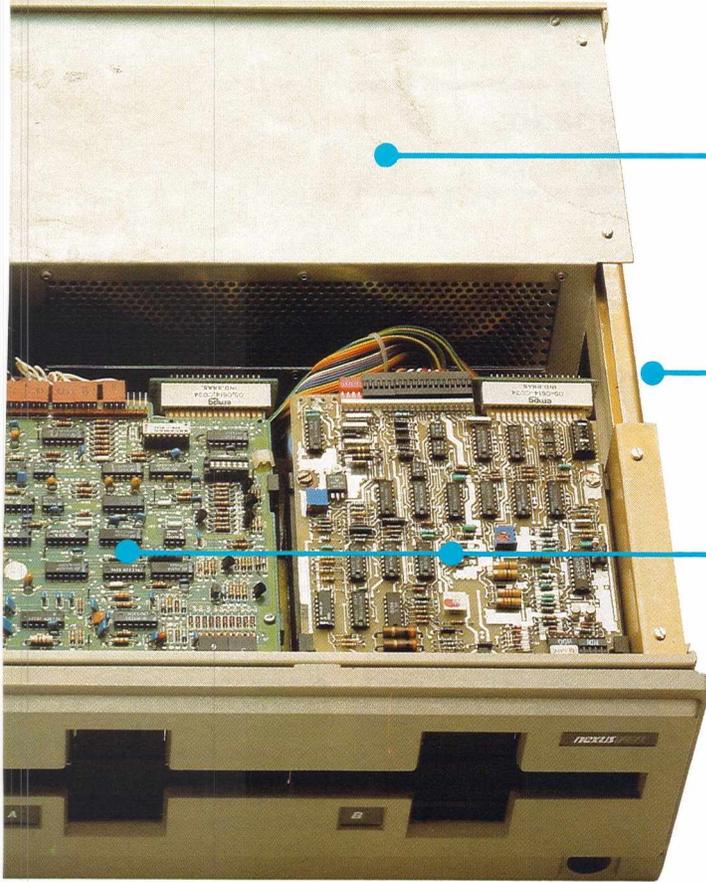
Conectores para placas de
expansão
diversas placas acessórias
estão disponíveis para o Nexus
1600

Expansão para 8087
o 8087 é um processador
aritmético especializado que
realiza funções numéricas de
ponto flutuante em alta
velocidade

CPU 8088

Área de EPROM
onde reside o núcleo básico do
sistema operacional





Fonte de alimentação
fonte chaveada compacta para alimentação dos módulos do sistema e das placas de expansão

Unidade de sistema
neste gabinete estão instaladas a placa de sistema (ver foto abaixo), as unidades de disco flexível, a fonte e as placas de expansão

Unidades de disco
disco flexível de 5 1/4", face dupla

NEXUS 1600

MICROPROCESSADOR

8088 - Processador Central
8087 - Co-processador aritmético de ponto flutuante (opcional).

CLOCK

8 MHz

MEMÓRIA

64 Kbytes de memória EPROM e 256 Kbytes de memória RAM (expansíveis até 704 K).

VÍDEO

Monitor de 12", em fósforo verde, com capacidade multinível (múltiplas tonalidades). Interface gráfica — modo alfanumérico (80 x 25 ou 40 x 25) e gráfico monocromático (640 x 200 pontos).

TECLADO

Tipo máquina de escrever com 86 teclas (10 teclas de função), teclado numérico separado, repetição automática e simulador de ruído mecânico.

LINGUAGENS

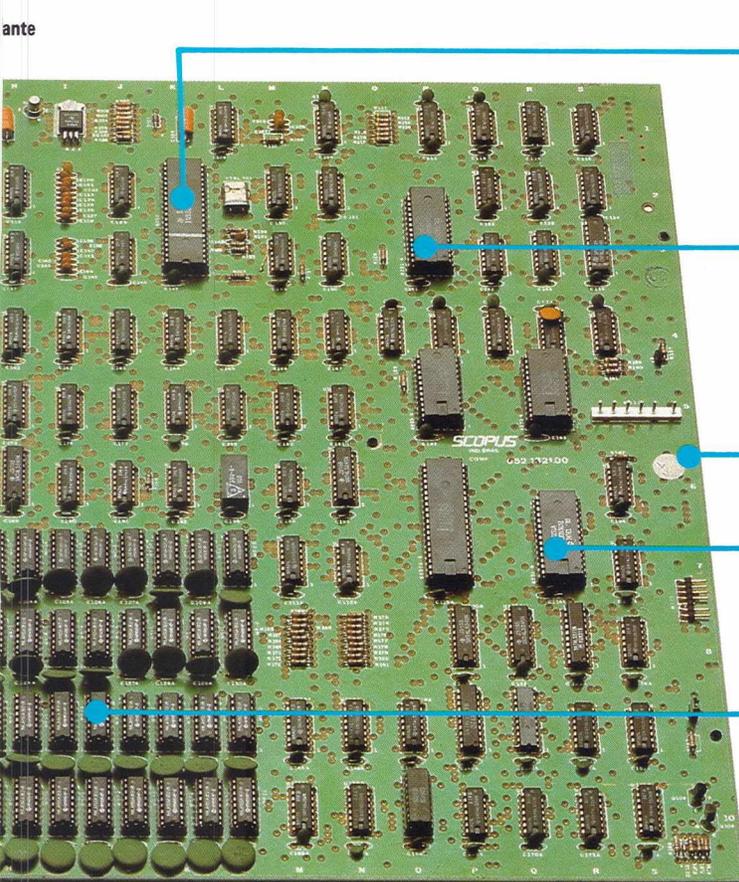
BASIC

PERIFÉRICOS

Unidades de disco (5 1/4" e míni Winchester), impressora, interface serial RS232 C, joystick e light-pen.

DOCUMENTAÇÃO

Dois manuais acompanham o equipamento: um de operação e outro de referência BASIC. O primeiro fornece uma visão das aplicações possíveis do equipamento, da constituição do sistema e das possibilidades de expansão. Seguem-se explicações sobre o modo de operação. A parte final é constituída por um manual de referência com informações mais detalhadas sobre o equipamento. O outro manual apresenta toda a orientação necessária à programação na linguagem BASIC do Nexus 1600.



UART
interface de comunicação assíncrona

USART
interface de comunicação síncrona assíncrona: para comunicação remota com computadores, através da emulação de protocolos diversos

Placa de sistema

Interface de teclado

Memória RAM
a configuração básica do Nexus 1600 dispõe de 256 K; a expansão para a capacidade máxima de 704 K é realizada através de um módulo opcional

Caixa-forte

Os dígitos binários 0 e 1 podem ser facilmente somados quando você utiliza as lógicas do AND, OR e NOT.

Em artigo anterior (ver p. 68) vimos como era relativamente simples o uso de circuitos transistorizados para realizar decisões lógicas, como AND, OR e NOT. É surpreendente como essas mesmas “portas lógicas” (logic gates) são igualmente a base para a execução de procedimentos aritméticos no interior do computador.

Pela lógica, as entradas das portas (gates) são 0 volt, para representar o valor “falso”, ou uma voltagem positiva para o valor “verdadeiro”. A ausência de voltagem é geralmente representada por 0, e a voltagem positiva, por 1. Quando as portas lógicas são usadas para realizar procedimentos aritméticos, os mesmos dígitos 0 e 1 são usados; desta vez, porém, representando exatamente os algarismos 0 e 1 que estiveram sendo somados.

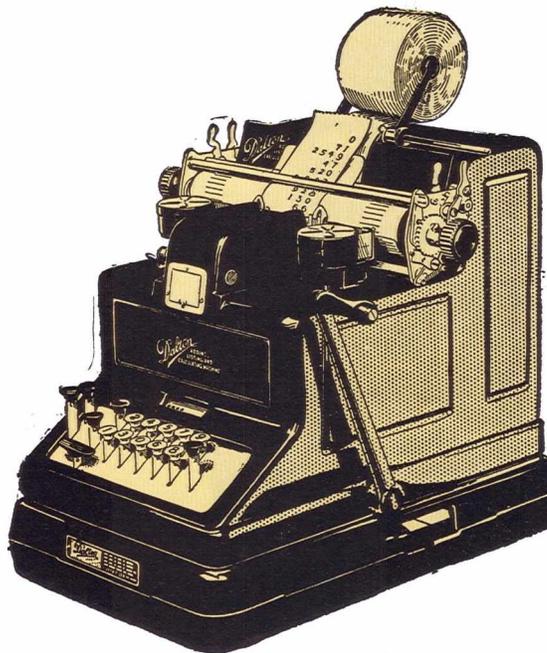
Se quisermos somar dois dígitos binários, haverá apenas duas entradas para o circuito de soma e pode haver apenas quatro combinações de entrada: 0 + 0, 0 + 1, 1 + 0 e 1 + 1. Na matéria anterior sobre aritmética binária, vimos que 0 mais 0 é igual a 0 (como na aritmética do sistema decimal). Também vimos que 0 mais 1 (ou 1 mais 0) é igual a 1 (novamente como na aritmética do sistema decimal). A diferença entre a aritmética que aprendemos na escola e o sistema binário é que, neste, 1 mais 1 é igual a 0, e um dígito é transportado para a coluna da esquerda. A representação aritmética dessas quatro adições é a seguinte:

X	Y	Z
0	+	0 = 0
0	+	1 = 1
1	+	0 = 1
1	+	1 = 10

Usando uma porta OR para realizar a adição, se ambas as entradas tiverem valor falso (0), obteremos uma saída de valor falso (0); se uma das entradas tiver o valor verdadeiro (0 + 1 ou 1 + 0), teremos uma saída com valor verdadeiro (1).

Até este ponto, o recurso a uma única porta OR pode parecer perfeitamente adequado para a soma de dois dígitos binários. Porém, cuidado: se ambas as entradas forem verdadeiras, a saída de uma porta OR simples terá também valor verdadeiro, mas esta seria uma resposta errada em aritmética binária. A resposta deveria ser um dígito 0 e um outro dígito 1, de transporte. Uma simples porta OR dará respostas corretas para três das quatro combinações, porém três em quatro ainda não é uma resposta satisfatória.

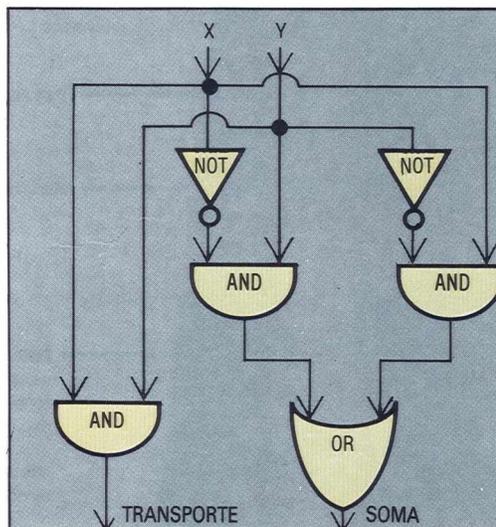
O que é necessário é um circuito que dê 0 como



A máquina de somar

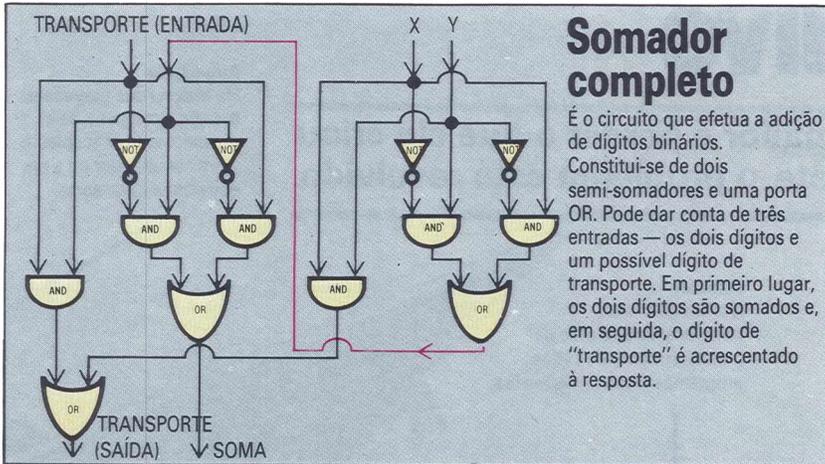
Até a invenção recente da calculadora eletrônica, a máquina de somar mecânica (ou caixa registradora) era uma presença comum em lojas e escritórios. Com exceção de algum aperfeiçoamento, quase não sofreu mudanças por 300 anos, funcionando com uma combinação de rodas dentadas e engrenagens. O potencial comercial da calculadora foi logo percebido. Pascal inventou a primeira máquina de somar para uso, por seu pai, no recolhimento de impostos. Com a criação, por Leibniz, dos procedimentos para cálculo de divisão e multiplicação, a calculadora estava preparada para ser utilizada no comércio.

resposta, se ambas as entradas forem 0; 1, se uma delas for 1 e a outra 0; e ainda 0, se ambas forem 1 e mais 1 de transporte. Isto não é tão difícil quanto parece. Se tivermos duas portas AND, com as duas entradas conectadas passando em ambas as portas, mas com uma entrada sendo invertida através de uma porta NOT em uma das portas AND, e se a outra entrada for invertida através de outra porta NOT e na outra porta AND (ver ilustração), nós teremos uma situação em que 0 em ambas as entradas resultará em uma saída de valor falso em ambas as portas AND, e um dígito 1, em ambas as entradas, dará analogamente uma saída com valor falso, em ambas as por-



Semi-somador

É um dispositivo para soma de dois números binários que, para tal, usa uma combinação de portas lógicas. É chamado semi-somador porque não pode dar conta do dígito de transporte que geralmente resulta da realização de somas. Lembre-se de que uma porta NOT converte o 1 em 0 e o 0 em 1. Para uma saída 1, as duas entradas de uma porta AND devem ser 1. A saída de uma porta OR será 1, se uma ou ambas as entradas forem 1. A saída será 0, somente se ambas as entradas forem 0.



Tem o nome de semi-somador porque, de certo modo, é adequado apenas parcialmente. Se pretendêssemos apenas a soma de uma coluna única de dois dígitos binários, seria suficiente. Entretanto, geralmente queremos somar 2 bytes de dados e cada byte contém 8 bits. O somador que controla a coluna da extrema direita necessitaria, na verdade, de apenas um semi-somador. Todavia, todos os somadores à esquerda dessa coluna têm de estar em condições de admitir três entradas: os dois dígitos de “sua” coluna e eventuais dígitos de transporte da coluna à direita. Observemos esta adição:

$$\begin{array}{r} 011 \\ + 111 \\ \hline 1010 \end{array}$$

tas. Por outro lado, 0 em uma entrada e 1 na outra darão duas saídas de valor “verdadeiro” em uma das portas AND. Desse modo, uma dessas portas produzirá uma saída de valor verdadeiro. Se as duas portas AND tiverem suas saídas ligadas a uma porta OR, a saída da porta OR somente terá o valor verdadeiro se uma e apenas uma das entradas for verdadeira.

Agora, já temos quase o circuito desejado: o único problema é que uma entrada de dois dígitos 1, embora resultando em uma “soma” correta 0, não produz o sinal de transporte. Todavia, o acréscimo de uma porta AND, ligada paralelamente às duas entradas, produzirá esse sinal, quando e somente quando ambas as entradas tiverem valor verdadeiro. A tabela de validação do circuito, na ilustração, chamada semi-somador, é a seguinte:

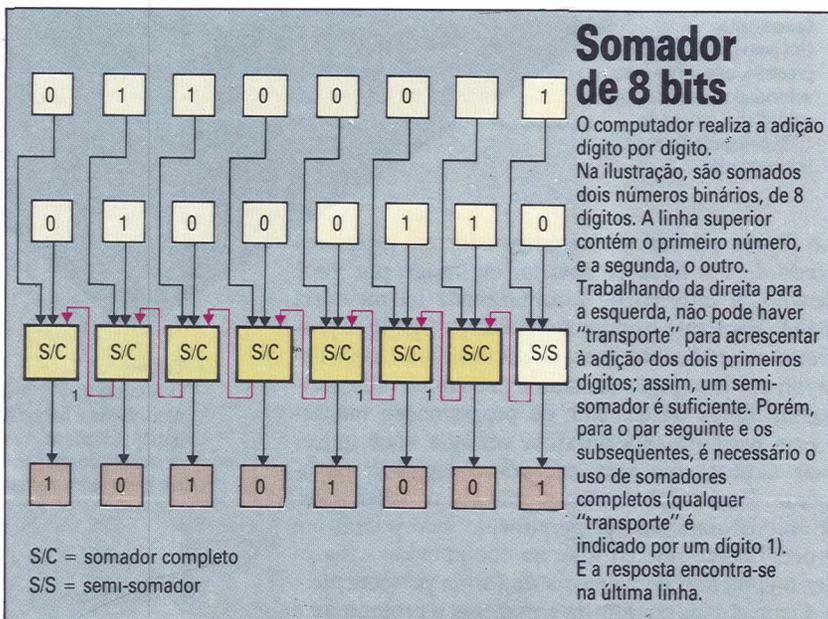
X (entrada 1)	Y (entrada 2)	T (saída “transporte”)	S (saída “soma”)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Ao somar a coluna “um”, dizemos 1 e 1 é 0, transportamos o 1 e escrevemos 0 sob a coluna “um”. Ao somar a coluna “dois”, dizemos 1 e 1 é 0, transportamos o 1, que somado ao transporte da coluna “um” dá 1, restando 1, de transporte. Escrevemos 1 sob a coluna “dois” e transportamos o número 1 para a coluna “quatro”. Aqui, dizemos 0 e 1 é 1, que, somado ao número de transporte da coluna “dois”, é 0, restando 1, de transporte. Escrevemos o 0 na coluna “quatro”, restando 1 de transporte, que podemos escrever sob a coluna “oito”. Ou seja, a tabela de validação para um “somador completo”, capaz de operar tanto transportes como dois dígitos binários, seria a seguinte:

X	Y	T (entrada)	T (saída)	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

O somador completo pode ser obtido pelo uso de dois semi-somadores, além de uma porta OR suplementar. Para cada somador completo, a “saída do transporte” está diretamente ligada à “entrada do transporte” do somador à sua esquerda; desse modo, podem ser encadeados tantos somadores completos quantos necessários.

Nos microcomputadores modernos, a maior parte das somas e outras operações aritméticas é realizada com um grande número de circuitos de somadores, idênticos em seu princípio de funcionamento aos descritos acima. O interior de uma CPU comum consiste em um simples chip de silício que incorpora pequenas áreas de memória RAM e ROM, um número muito grande de circuitos interruptores e a ALU (Unidade Aritmética e Lógica), que é a parte que contém todas as portas lógicas e somadores necessários ao computador, para realizar cálculos e decisões lógicas.



Grave e archive

A grande preocupação do programador é perder o que ele criou no vídeo. Com o uso de fitas cassete, o problema está resolvido.

Alto-falante
Na maioria dos gravadores cassete, desliga-se, se o gravador estiver acoplado a um computador ou a um equipamento de som.

Cabeçote de limpeza
Apaga qualquer sinal anteriormente gravado na fita, durante a realização da gravação.

Contador
Um compimento indispensável, no caso de armazenamento de vários programas em uma mesma fita.

Cabeçote de gravação/reprodução
Este cabeçote com dupla função grava os sinais sonoros na fita magnética e os reproduz.



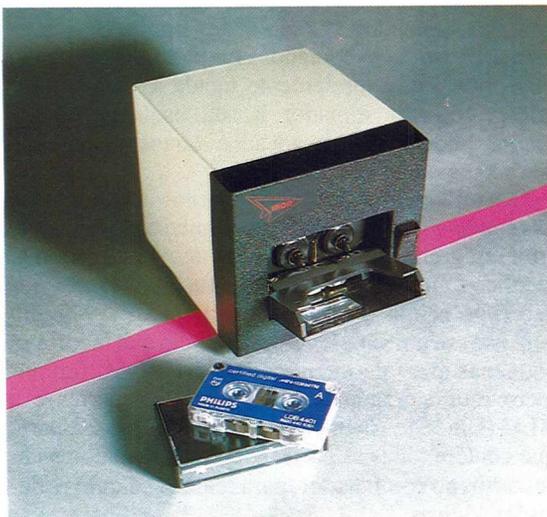
O Hobbit
O Hobbit da Ikon inglesa é um gravador cassete muito eficiente, projetado exclusivamente para armazenar programas de computadores. É superior a um gravador comum, pois é inteiramente controlado pelo software. Não há necessidade de uso de teclas de avanço, retrocesso, reprodução, ou gravação: o Hobbit faz tudo isso por você. Se quiser carregar um programa, basta digitar o nome, e o Hobbit seleciona em seu catálogo o programa e sua posição, e coloca a fita no lugar correto.

Cabrestante
Um pino produzido com precisão, que gira a uma velocidade cuidadosamente controlada, movendo 1 7/8 polegada de fita por segundo.

Volume
Determina o volume da reprodução e necessita de ajuste cuidadoso, senão os cassetes podem não ser carregados corretamente.

Ao digitar em seu microcomputador um programa tirado de um livro ou revista, ou criado por você mesmo, a informação é armazenada na memória RAM. Enquanto o computador estiver ligado, o programa estará intacto, mas, ao ser desligado, tudo desaparece! Isso pode constituir um grande aborrecimento: horas de trabalho de programação inteiramente perdidas. Na próxima vez que você quiser usar o programa será necessário digitá-lo novamente. Para resolver esse problema, os fabricantes de microcomputadores geralmente incorporam um procedimento que permite ao computador armazenar o conteúdo da memória de forma permanente.

O meio mais comum de armazenar o programa é a





Motor

Dirige o cabrestante a uma velocidade constante e também gira o carretel fornecedor e o carretel receptor, para avançar e retroceder a fita.

fita cassete, facilmente disponível e utilizada em quase todos os microcomputadores. A forma pela qual cada computador armazena os dados tende a variar um pouco: um programa criado e armazenado em um computador CP 500 não poderá ser carregado no TK85, por exemplo. Entretanto, o procedimento utilizado para converter o programa em uma forma que permita o armazenamento é quase universal.

Um gravador cassete do tipo usado na maioria dos microcomputadores é evidentemente mais adequado para armazenar sons — e o programa está armazenado no computador sob a forma de dígitos binários. Estes devem ser convertidos em sons, de modo que permitam ao computador reconhecer a di-

para ser representado. Entretanto, para indicar o início e o fim, o computador geralmente coloca um tom a mais no fim de cada byte. Estes são chamados bits de início e bits de parada e seu valor é sempre o mesmo: 1 ou 0, de acordo com cada computador.

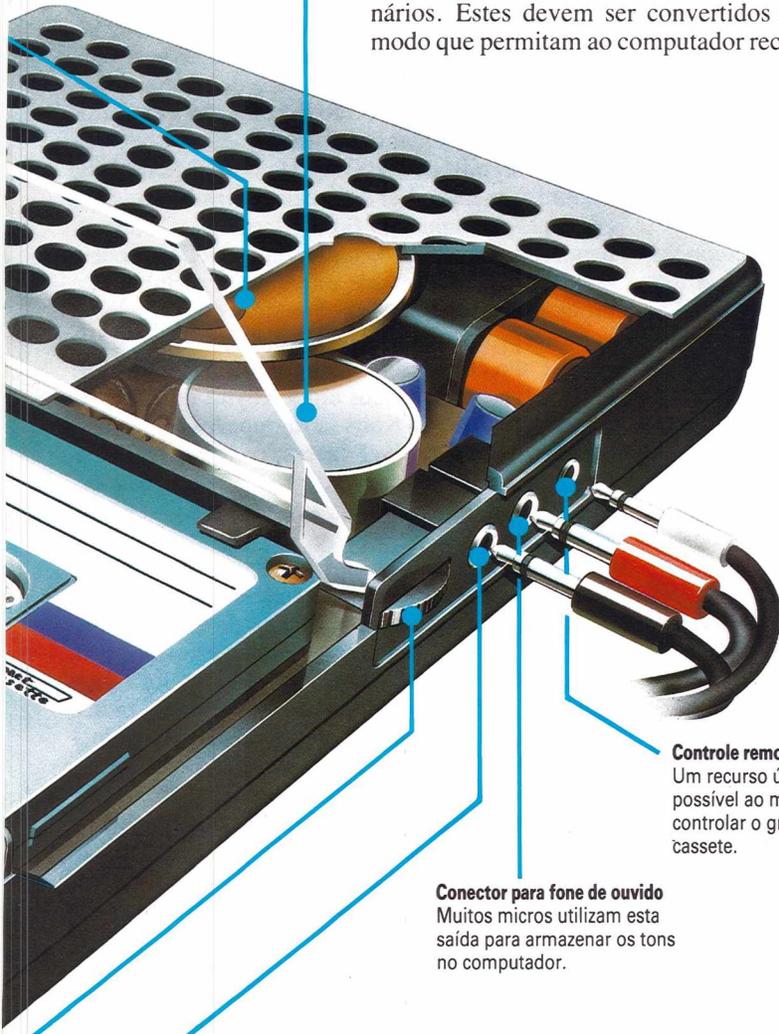
O programa é armazenado quase do mesmo modo, exceto pelo fato de ser, normalmente, dividido em segmentos. De modo geral, estes têm 256 bytes de comprimento e muitas vezes incluem dados adicionais que permitem ao computador verificar se estão sendo carregados os dados corretos. O sistema aqui utilizado é bem simples e chama-se soma de verificação. O primeiro byte do segmento contém o número de bytes presentes no segmento e o último contém um número calculado de modo especial que representa o total de todos os bytes somados. Quando procede à nova leitura do cassete, o computador compara os números encontrados na fita com os calculados por ele mesmo e, se não coincidirem, informa o usuário sobre o erro.

Alguns equipamentos cassete chegam a proceder a essa comparação atribuindo nomes e uma numeração a todos os segmentos. Se um erro ocorrer, é possível simplesmente voltar a fita alguns centímetros e tentar de novo. Ao contrário, a maioria dos equipamentos não apresenta o nome do programa que está sendo carregado.

A velocidade Baud

A velocidade com que os tons são produzidos e gravados na fita é geralmente (e erroneamente) chamada de velocidade Baud. O nome tem origem no código Baudot, utilizado nas primeiras formas de telégrafo elétrico, que na verdade se refere ao número de vezes em que o sinal se altera por segundo. Uma medida mais exata seria o número de bits gravados por segundo. Quanto maior a velocidade — que varia de 300 a 1.200 bits por segundo —, mais rapidamente os programas serão armazenados na fita; e, também, menos tempo será necessário para carregá-los outra vez para o computador. Infelizmente, a confiabilidade do sistema é afetada na proporção direta da maior rapidez com que os tons são armazenados; a velocidade de 1.200 bits por segundo é confiável e rápida o suficiente para evitar problemas. Alguns equipamentos têm duas velocidades, geralmente uma baixa (muito confiável), de 300 bits por segundo, e uma mais rápida — de 1.200 ou de 2.400 bits por segundo. Cópias de programas importantes podem ser guardadas em ambas as formas, para evitar acidentes.

A fita cassete deve ser de boa qualidade: não há problema algum em usar fitas comuns de gravador, em vez de fitas especialmente feitas para computador, mas deve-se tomar cuidado na escolha de uma marca de confiança, bem como evitar fitas com duração acima de C-60. A capacidade aproximada de duração de uma fita pode ser determinada pela divisão da velocidade da interface do cassete por 10, o que fornece o número de bytes que será armazenado na fita por segundo; uma fita C-60, com 30 minutos de cada lado, na qual a interface opera a 1.200 bits por segundo, poderia conter cerca de 432 Kbytes de programa.



Controle remoto

Um recurso útil, que torna possível ao microcomputador controlar o gravador cassete.

Conector para fone de ouvido

Muitos micros utilizam esta saída para armazenar os tons no computador.

Conector para microfone

Freqüentemente usado como entrada para os dados provenientes do computador para o gravador. Entretanto, deve ser usado apenas se os cabos Auxiliar e DIN não servirem.

Geralmente, é necessário ajuste cuidadoso do tom e do volume, quando esta entrada é utilizada.

ferença entre um bit “ligado” e outro “desligado” — os dígitos 0 e 1 do sistema binário. O método mais simples de fazê-lo é criar um som que represente o dígito 1 e outro que represente 0. Conventiou-se representar o dígito 1 por tons de 2.400 ciclos, e o dígito 0, por tons de 1.200 ciclos.

Quando o comando SAVE é digitado no computador, a primeira coisa a ser gravada será um certo número de segundos em um tom constante. Isso é feito de modo que, mais tarde, quando o gravador cassete reproduzir a gravação para alimentar o computador, este possa distinguir entre a fita em branco e a parte que contém o programa. Os primeiros dados reais a ser gravados são os tons que representam os caracteres do nome dado ao programa. Cada um desses caracteres consiste em 1 byte — total de 8 bits —; assim, cada um dos caracteres necessita de oito tons

Lógica misteriosa

O computador armazena na memória milhares de bytes de informação. E consegue lembrar onde está cada uma delas!

Uma das formas de referência à memória do computador é o armazenamento a curto e a longo prazo. Este segundo tipo não perde a informação, podendo retê-la por longos períodos de tempo, até mesmo quando a energia é desligada. A fita magnética e o disco flexível enquadram-se nessa categoria. Mas os computadores também necessitam de memória para armazenamento temporário dos programas e resultados.

Outra forma de referência à memória do computador é considerá-la como interna ou externa. A memória interna, localizada no interior do computador, é totalmente "eletrônica"; a externa, como o próprio nome diz, é periférica ou fora do computador. A externa quase sempre é parcialmente mecânica, envolvendo mecanismos como fita cassete, unidades de disco, ou fitas perfuradas e leitoras.

A memória interna eletrônica é geralmente chamada de memória principal, enquanto a externa é mencionada como memória secundária, memória

auxiliar ou memória de reserva. Atualmente, há dois tipos de memória interna: RAM e ROM. Ambas são dispositivos inteiramente eletrônicos, na forma de chips de silício, colocados em caixas retangulares de plástico, com conjuntos de terminais paralelos de estanho ou banhados com prata. Há outras semelhanças, na forma como são selecionadas e "endereçadas" pela CPU do computador, e falaremos delas mais adiante.

uns em cada local da memória é fixo e ajustado na fabricação, e não pode, conseqüentemente, ser alterado. As ROMs podem ser consideradas verdadeiras "bibliotecas de referência" do computador. Este consulta o conteúdo ROM mas não é capaz de "escrever" qualquer coisa nela.

ROM significa Read Only Memory (Memória de Leitura Somente). Ler é o termo utilizado para indicar o que o computador faz ao "acessá-la" ou recuperar as informações da memória. As ROMs podem ser de tipos ligeiramente diferentes e algumas têm programas que podem ser removidos ou apagados, e depois reprogramados. Uma ROM típica é a 2364 da Intel. É um chip descrito como sendo uma ROM de 65.536 bits, organizados em 8 Kbytes de 8 bits. O que significa que 64 Kbits são agrupados em 8 bits, e cada local "endereçável" acessa ou lê um byte completo. Em matemática, $1\text{ K} = 2^{10}$ (2 elevado à décima potência) ou 1.024; assim, $64\text{ K} = 64 \times 1.024$ ou 65.536.

Portanto, o computador é capaz de selecionar qualquer um dos 8.192 (8 K) endereços. Um exame mais minucioso das especificações do chip 2364 revela que possui 28 pinos, com um reservado para fornecimento de energia de +5 volts, e um para a conexão do fio terra, o que dá o total de 26 pinos. Cada byte contém 8 bits, de modo que, quando 1 byte é lido do chip, os 8 bits deste byte devem ser conduzidos por meio de fios do chip à CPU. Conseqüentemente, há oito fios para transportar os bits no byte que está sendo lido pela CPU. Estes fios são chamados de "data bus" (bus de dados). Oito dos pinos no chip são dedicados a essa tarefa, um para cada bit no byte.

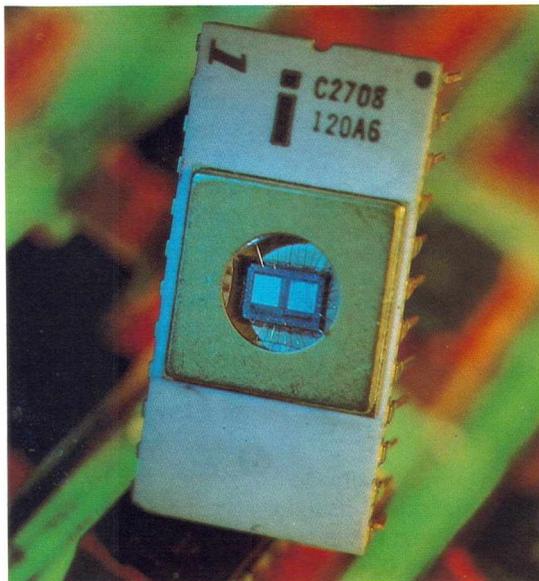
Isto nos deixa dezoito pinos. Um pino não é necessário e não está conectado. Ele existe, pois é mais fácil fabricar chips com um número par de pinos. Quatro pinos são utilizados para "selecionar" o chip de diversas maneiras. Estes são o pino de "capacidade de saída", o de "capacidade de chip" e dois de "seleção de chip". Estes pinos recebem os sinais do computador e possibilitam ao chip "saber" quando está sendo requerido.

Os pinos restantes são os de "endereço". Cada pino é conectado a um fio de "address bus" (bus de endereço) e estes transportam o endereço do byte necessário, codificado na forma binária. Treze dígitos binários podem fornecer 2^{13} ou 8.192 combinações únicas dos algarismos 1 e 0, de modo que as treze linhas de endereço são suficientes para selecionar cada um e todos os 8.192 bytes armazenados na ROM.

As RAMs são os quadros-negros do computador. Ou seja, programas e dados são armazena-

EPROM

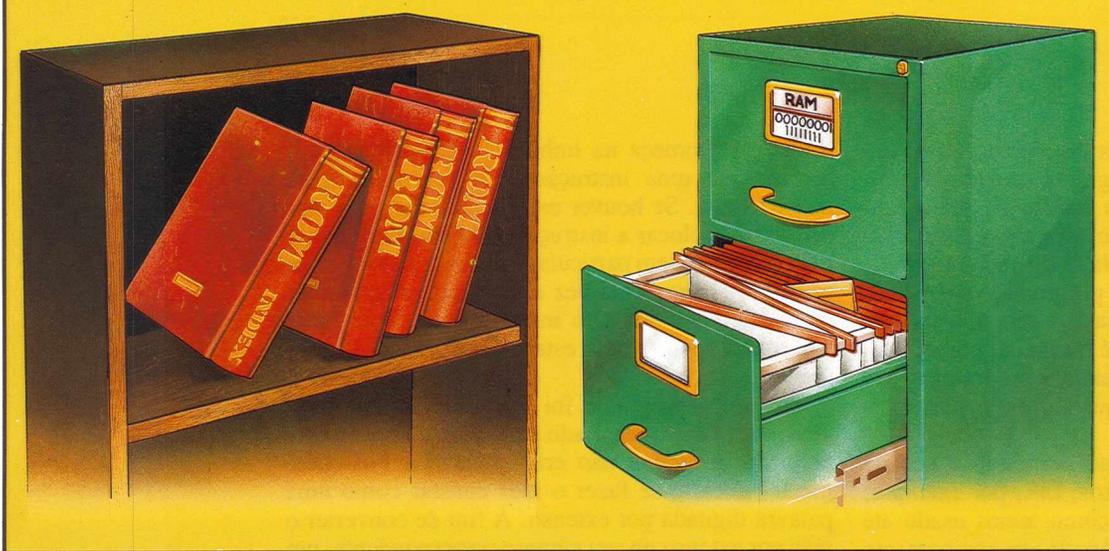
O inconveniente das ROMs comuns é que o conteúdo da memória é "embutido" e não pode ser alterado. A EPROM (Erasable Programmable Read Only Memory) é bem mais flexível. Uma vez programada, pode ser reprogramada apagando-se o conteúdo e "escrevendo" um novo programa. A EPROM possui uma "janela" de sílica que permite que raios ultravioleta atravessem o interior, fazendo com que os capacitores que armazenam os bits na EPROM se descarreguem. Na ausência da luz ultravioleta, os capacitores mantêm a carga indefinidamente, e o conteúdo da memória é retido.



A principal diferença funcional é que os chips da memória ROM são utilizados para armazenar programas de forma permanente. O padrão de zeros e



ROM e RAM



Memória de longo prazo

A ROM (Read Only Memory, Memória de Leitura Somente) é análoga a um livro em que estão “armazenadas” as informações permanentemente. Não se pode alterar ou remover os dados, nem alterar as palavras impressas nas páginas.

Memória de curto prazo

A RAM (Random Access Memory, Memória de Acesso Aleatório) mais se parece com um sistema de arquivo, uma vez que as informações podem ser alteradas, e os dados não são permanentes. A RAM fica limpa quando o computador é desligado.

dos deles temporariamente, enquanto o computador está trabalhando, e os resultados e outros dados são também “escritos” temporariamente. A memória RAM, em geral, é mais complexa internamente que a ROM, pois cada bit em cada byte da RAM deve ser capaz de alterações, se assim for “escrito”.

Um chip típico RAM é o Intel 2114. Cada chip RAM 2114 contém 4.096 bits de memória, e estes são organizados como 1.024 “nibbles” (meios bytes) de 4 bits. Isto significa que cada endereço dará saída a 4 bits de dados. Dois destes chips serão necessários para produzir 1 Kbyte total de dados.

Cada chip 2114 possui apenas dezoito pinos, dois dos quais são para fio terra e fornecimento de energia. Quatro são usados para linhas de dados de entrada/saída. Um é utilizado para sinal de seleção de chip (o sinal diz ao chip que este é solicitado ou “selecionado”) e outro diz ao chip quando deve ser escrito ou lido. Os dez pinos restantes são empregados para os address bus.

Dez linhas de endereço podem identificar unicamente 2^{10} locais, ou 1.024. Se o computador possuir 64 Kbytes de RAM, e tiverem sido utilizados chips 2114 Intel, um total de 128 chips RAM seriam necessários, já que 2 chips formam um byte completo. Atualmente, é mais comum empregar chips RAM de maior densidade, que contêm mais memória no mesmo espaço. Com os chips RAM modernos, tal como o 4164, é possível obter 64 Kbytes de RAM com apenas 8 chips.

Os chips RAM e ROM têm-se tornado cada vez mais compactos e de menor custo e, agora, existem 128 Kbits em um único chip. Contudo, o acondicionamento de densidades mais altas em chips únicos não poderá evoluir indefinidamente. O circuito no silício tem ficado tão diminuto que as técnicas ópticas empregadas para “gravar” os circuitos estão chegando ao seu limite.

Os chips de memória de alta densidade do futuro serão provavelmente fabricados utilizando-se feixes de elétron ou raios X.

De modo mais abrangente, há dois tipos de memória RAM em uso: a RAM estática e a RAM dinâmica. Há vantagens e desvantagens em ambas, mas a RAM dinâmica é atualmente mais utilizada. Ambos os tipos apagam o conteúdo da memória, logo que o fornecimento de energia é interrompido, mas a memória dinâmica precisa ter seu conteúdo “restaurado” em intervalos de alguns milissegundos.

Cada bit na memória precisa ser restaurado ou reescrito sem diminuir a velocidade com que a CPU acessa os dados ali contidos. Isto significa que um circuito de controle de tempo muito especial deve ser projetado, o que torna o trabalho do projetista do circuito mais difícil.

A memória dinâmica oferece duas vantagens em relação à memória estática. A dinâmica requer apenas um transistor por bit, ao passo que três transistores são normalmente necessários para cada bit na memória estática. Isso permite que maior quantidade de memória seja colocada em chips menores. A maioria dos chips RAM dinâmicos possui apenas dezesseis pinos. A outra vantagem da RAM dinâmica é utilizar menos energia, comparada a sua equivalente estática. Portanto, gera menos calor e tem baixo consumo de força.

A vantagem da RAM estática reside na simplicidade do projeto do circuito. Escritos os conteúdos da memória, ali permanecem, sem a necessidade de ser restaurados. Cada célula de memória de 1 bit requer três transistores; assim sendo, é difícil conseguir as altas densidades que a RAM dinâmica permite. A RAM estática consome energia, o calor extra produzido complica o sistema de resfriamento do computador e pode demandar o uso de um ventilador para resfriamento, tornando o modelo mais dispendioso.

À espera do Natal

Novos comandos para trabalhar com dados e escrever um programa que calcula quantos dias faltam para o Natal.

Este programa retoma todos os tópicos vistos até agora em nosso curso de programação e também introduz algumas importantes instruções de linguagem BASIC. A finalidade do programa é calcular quantos dias faltam para o Natal do ano corrente.

Se observar a listagem do programa, você perceberá que ela se inicia por uma lista de variáveis usadas. Este procedimento não é essencial, porém é recomendável, pois pode tornar seus programas mais fáceis de ser entendidos, quando você posteriormente vier a examiná-los.

Algumas versões do BASIC permitem o uso de variáveis com nomes extensos, DIA, por exemplo, em vez de letras isoladas, como temos usado até agora. Se tiver a sorte de possuir um equipamento em linguagem BASIC, que possibilita nomes extensos de variáveis, escolha nomes sugestivos. Nomes como DIA, MES, NUMDIA são preferíveis a símbolos como A, X ou D. Se não tiver escolha, porque o BASIC que você utiliza não possibilita nomes extensos de variáveis, a anotação das variáveis no início do programa torna-o mais "legível".

Quando o programa é processado, o primeiro sinal que surge na tela são as instruções PRINT, que se iniciam na linha 230. Elas indicam resumidamente o que o programa fará e então solicitam ao usuário digitar a data no formato apresentado, separando dia, mês e ano com vírgulas.

A primeira instrução não familiar encontra-se na linha 300. É uma instrução DIMensão (DIMension), utilizada para determinar o total de itens ou elementos que entram na tabela designada como X. As tabelas, às vezes chamadas variáveis indexadas, assemelham-se a variáveis comuns, mas apresentam várias subdivisões. Na linha 300, criamos uma variável denominada X, com treze subdivisões no seu interior. Voltaremos, mais adiante, à questão das tabelas e da instrução DIM, detalhadamente.

```
310 INPUT D, M$, A
```

Esta linha é uma instrução comum para entrada de dados (INPUT), com a diferença de que prevê três entradas. D é uma variável numérica que contém a data do dia; A é outra variável numérica, para o ano. A variável M\$ é um pouco diferente. É denominada variável alfanumérica (variável string), indicada pelo sinal \$ (cifrão). Este tipo de variável aceita caracteres, bem como números, do teclado. Se, por exemplo, digitarmos 23, JANEIRO, 1983, à variável D será atribuído o valor 23, à variável M\$ a série de caracteres JANEIRO e à variável A o valor 1983.

```
330 GOSUB 560 REM "ROTINA NO. DO MES"
```

Esta instrução faz o programa desviar-se para a sub-

rotina que começa na linha 560. Observe, igualmente, que uma instrução REM foi inserida na mesma linha. Se houver espaço na linha, não será necessário colocar a instrução REM em outra linha. Esta sub-rotina em particular é usada pelo programa principal apenas uma vez e, a rigor, poderia com igual facilidade ter sido inserida nele. Fazer uma sub-rotina apenas isola esta parte do resto do programa.

Quando o programa foi inicialmente desenvolvido, um número era usado para indicar o mês e esta parte do programa não era necessária. Posteriormente, decidiu-se fazer o mês constar como uma palavra digitada por extenso. A fim de converter o mês por extenso no seu número correspondente, um programa suplementar foi desenvolvido separadamente, na forma dessa sub-rotina. A única mudança necessária no programa principal (original) foi acrescentar uma única instrução GOSUB. Esta sub-rotina exemplifica a facilidade com que os programas podem ser elaborados em blocos e unidos com a utilização dos comandos GOSUB e RETURN.

A rotina em si é muito simples, mas exemplifica a engenhosidade da linguagem BASIC em lidar com séries de caracteres. Admitamos que fornecêssemos JANEIRO como o mês solicitado na instrução de entrada (INPUT). À variável M\$ seria atribuída a série de caracteres JANEIRO. A primeira linha da sub-rotina é:

```
560 IF M$ = "JANEIRO" THEN LET M = 1
```

Esta instrução compara os conteúdos da variável M\$ com os caracteres entre aspas. Se forem os mesmos (como acontece neste caso), a linha prossegue a fim de colocar 1 como valor da variável numérica M. Não confunda a variável M com a variável M\$. São diferentes: apenas uma pode conter a variável alfanumérica: aquela com o sinal \$. Após verificar se a variável M\$ corresponde a JANEIRO, o programa segue para a próxima linha e verifica se o conteúdo da variável M\$ corresponde a FEVEREIRO. Caso contrário, a variável M não passa a ter valor 2. Apenas onde a correspondência for correta a variável M receberá um valor, que será correspondente ao número do mês: 1 para janeiro, 3 para março etc.

Ao chegar à linha 680, o programa retorna (RETURN) ao programa principal, na linha seguinte à instrução GOSUB: é a linha 340, que contém uma instrução REM, sem nenhuma observação; foi inserida simplesmente para dar mais espaço entre as linhas, tornando mais fácil a leitura do programa.

As linhas 350 e 370 são um loop do tipo FOR-NEXT, que incrementa o valor da variável I, iniciando com 1 e seguindo até 13. A variável I é usada



como indexador da tabela X, na linha 360, e deve ser cuidadosamente examinada.

```
360 READ X(I)
```

READ é uma instrução nova, ainda não usada anteriormente. É sempre utilizada com uma instrução DATA correspondente. A instrução DATA para esta linha está na linha 510:

```
510 DATA 31,28,31,30,31,30,31,31,30,31,30,25,0
```

Esses números, com exceção dos dois últimos, representam o total de dias em cada mês do ano. As duas linhas equivalem a treze instruções LET.

```
LET X(1) = 31
LET X(2) = 28
LET X(3) = 31
LET X(4) = 30
LET X(5) = 31
LET X(6) = 30
LET X(7) = 31
LET X(8) = 31
LET X(9) = 30
LET X(10) = 31
LET X(11) = 30
LET X(12) = 25
LET X(13) = 0
```

O loop montado na linha 350 faz a variável I percorrer de 1 a 13; desse modo, foi possível substituir X(I) por X(1), X(2), X(3) etc.

Antes de voltar a este programa, examinemos um outro menor e bem mais simples:

```
10 READ A, B, C
20 LET D = A + B + C
30 PRINT D
40 DATA 5, 10, 20
```

Aqui, a instrução READ, na linha 10, faz a leitura do primeiro item da instrução DATA na linha 40 e “registra” seu valor para a primeira variável. Em outras palavras, atribui o valor 5 à variável A. A instrução READ lê, em seguida, o próximo item e o coloca na variável seguinte. Este programa torna A = 5, B = 10 e C = 20. Em seguida, soma-os e atribui o resultado à variável D. O resultado, 35, é impresso (PRINT) na linha 30.

Voltemos ao nosso programa principal. Na primeira passagem pelo loop, iniciado na linha 350, o valor da variável I é 1. Assim, a linha 360 equivale a READ X(1). O item correspondente, na linha 510, é 31 (o primeiro item). Conseqüentemente X(1) passa a ter 31 como valor.

Na segunda passagem pelo loop, a variável I torna-se 2 e, com isso, a linha 360 passa a equivaler a READ X(2). O próximo item na linha DATA é 28. Isto significa que a variável X(2) passa a valer 28. Desse modo, todas as treze “subdivisões” da variável X indexada são preenchidas com o número de dias de cada mês; com exceção do 12.º mês, que tem apenas 25 dias (você percebe por quê), e da décima terceira subdivisão, de que falaremos mais adiante.

```
390 GOSUB 750 REM “ROTINA DE ANO BISSEXTO”
```

Essa linha dirige o programa para uma sub-rotina que verifica se o ano fornecido é ou não um ano bissexto. Vejamos seu procedimento.

```
100 REM LISTA DE VARIÁVEIS
110 REM
120 REM D = DATA DE HOJE
130 REM M$ = NOME DO MES
140 REM A = ANO
150 REM I = INDEXADOR 1
160 REM X = TABELA DE DIAS EM CADA MES
170 REM R = DIAS RESTANTES
180 REM M = NO. DO MES
190 REM L = INDEXADOR 2
200 REM Z = VALOR INTEIRO DE A/4
210 REM
220 REM
230 PRINT “ESTE PROGRAMA CALCULA”
240 PRINT “O NUMERO DE DIAS QUE FALTAM”
250 PRINT “ATE O NATAL DESTES ANO”
260 PRINT
270 PRINT “FORNECER O DIA DE HOJE, MES, ANO”
280 PRINT “EX.: 12, JULHO, 1984”
290 PRINT
300 DIM X(13)
310 INPUT D, M$, A
320 REM
330 GOSUB 560 REM ROTINA “NO. DO MES”
340 REM
350 FOR I = 1 TO 13
360 READ X(I)
370 NEXT I
380 REM
390 GOSUB 750 REM ROTINA “ANO BISSEXTO”
400 REM
410 LET R = X(M) - D
420 FOR L = M TO 11
430 LET M = M + 1
440 LET R = R + X(M)
450 NEXT L
460 REM
470 IF R = 1 THEN GOTO 500
480 PRINT “FALTAM”; R; “DIAS PARA O NATAL”
490 GOTO 520
500 PRINT “FALTA 1 DIA PARA O NATAL”
510 DATA 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 25, 0
520 END
530 REM
540 REM
550 REM
560 IF M$ = “JANEIRO” THEN LET M = 1
570 IF M$ = “FEVEREIRO” THEN LET M = 2
580 IF M$ = “MARCO” THEN LET M = 3
590 IF M$ = “ABRIL” THEN LET M = 4
600 IF M$ = “MAIO” THEN LET M = 5
610 IF M$ = “JUNHO” THEN LET M = 6
620 IF M$ = “JULHO” THEN LET M = 7
630 IF M$ = “AGOSTO” THEN LET M = 8
640 IF M$ = “SETEMBRO” THEN LET M = 9
650 IF M$ = “OUTUBRO” THEN LET M = 10
660 IF M$ = “NOVEMBRO” THEN LET M = 11
670 IF M$ = “DEZEMBRO” THEN LET M = 12
680 RETURN
690 REM
700 REM
710 REM
720 REM NOTA: ESTA ROTINA NAO VERIFICA
730 REM OS ANOS BISSEXTOS DO FINAL
740 REM DE CADA SEculo
750 LET A = A/4
760 LET Z = INT (A)
770 IF A - Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

```

750 LET A = A/4
760 LET Z = INT (A)
770 IF A-Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
    
```

O ano bissexto é definido como "divisível por 4". No caso de um século, deve poder ser divisível por 400 para ser um ano bissexto. Para simplificar, não verificamos o século, apenas a divisibilidade por 4.

A linha 750 coloca na variável A o seu valor anterior dividido por 4. O novo valor da variável A resultará em um número inteiro, se o ano for divisível por 4. Caso não, terá uma fração decimal.

A linha 760 utiliza a função INT para calcular o valor "inteiro" de A. Se o número do ano fornecido for 1985, o novo valor da função INTeiro transforma o número com fração decimal no número inteiro imediatamente inferior. O número a ser arredondado é colocado entre parênteses após a instrução INT. Outra alternativa é a colocação do nome de uma variável entre parênteses. Assim, LET Z = INT (496,25) atribuiria à variável Z o valor 496.

A linha 770 subtrai Z de A e verifica se o resultado é 0. Se for, significa que o ano é bissexto (pois não há fração decimal no novo valor da variável A). Se for este o caso, o programa se dirige para a linha 790 pelo uso da instrução GOTO. A linha 790 acrescenta 1 ao segundo item da tabela (o segundo item era 28, o número de dias em um mês de fevereiro de ano não bissexto).

Se o resultado da subtração na linha 770 não for zero, X(2) é mantido como está e a sub-rotina RETURN ao programa principal, na linha 400.

A linha 400 é outra instrução REM usada apenas para deixar mais espaço entre as linhas e tornar o programa mais fácil de ler. A próxima linha que efetivamente realiza alguma operação é a 410, onde R é a variável que contém o número dos dias restantes. A ela foi atribuído o número de dias no mês dado, menos o número do dia fornecido. Se tivéssemos fornecido, por exemplo, 12, FEVEREIRO, 1983, a variável D seria igual a 12 e a variável M seria 2. Assim, X(M) seria igual a X(2) e o segundo item da tabela X seria 28 (e não teria o número 1 somado a ele, pois 1983 não é ano bissexto). Por consequência, a variável R terá o valor de 28 - 12, isto é, 16, o número de dias restantes no mês fornecido, fevereiro.

A linha 420 inicia um outro loop. Ele tem por objetivo incrementar o valor da variável M. Você pode perceber por que dizemos FOR L = 1 TO 11, em vez de FOR L = 1 TO 12? Se a variável M for 2, porque fornecemos o mês de FEVEREIRO, a linha 430 aumentará seu valor para 3. Na linha 440 o valor da variável R, que representa o número de dias restantes, passa a ser o último valor de R mais X(M). A variável X(M) agora equivale a X(3), já que M foi aumentado em uma unidade. O valor de X(3) é 31, o total de dias em março. Desse modo, na linha 440 o novo valor da variável R passa a ser 16 + 31 (16 foi o resultado da subtração 28 - 12). Na próxima passagem pelo loop, a variável M aumentará para 4 e o número de dias em abril, X(4), será somado ao último valor da variável R. Assim, esta se torna 16 + 31 + 30.

A última passagem pelo loop ocorre quando o úl-

timo valor de M, 11, for alcançado (na linha 420). A linha 430 acrescenta 1 ao valor da variável M, pela última vez, e ela passa a valer 12. O valor de X(12) na instrução DATA é 25. Bem, não há 25 dias em dezembro, porém nos interessa apenas o número de dias até o Natal. Quando o mês atual for dezembro, neste loop M assumirá o valor 13 (12 + 1), e R já estará calculado corretamente com o número de dias até o Natal (R = X(12) - D). É por este motivo que a 13.ª subdivisão de X é igual a zero, para que, quando M for igual a 13, X(13) seja igual a zero, não alterando o valor de R.

```
470 IF R = 1 THEN GOTO 500
```

Esta linha verifica se falta só um dia para o Natal, de modo que obtenha uma resposta na tela gramaticalmente correta. Se a variável R não for 1, deve faltar mais de um dia e com isso a instrução PRINT da linha 480 estará gramaticalmente correta.

Isso é tudo. A versão da linguagem BASIC que usamos pode ser processada na maioria dos computadores (ver o quadro "A propósito..."), com exceção possivelmente da sub-rotina do "ano bissexto". O uso do comando LET varia muito, dependendo da versão do BASIC utilizada. Se linhas como IF M\$ = "SETEMBRO" THEN LET M = 9 não derem certo em seu computador, a sub-rotina poderá ser mudada para:

```

560 IF M$ = "JANEIRO" THEN GOTO 900
570 IF M$ = "FEVEREIRO" THEN GOTO 910
580 IF M$ = "MARCO" THEN GOTO 920
:
900 LET M = 1
905 RETURN
910 LET M = 2
915 RETURN
920 LET M = 3
925 RETURN
(... e assim por diante)
    
```

Esta solução consome mais espaço e é mais difícil de ser acompanhada com as diversas instruções GOTO e RETURN. Todavia, ela mostra que há vários modos de resolver um mesmo problema.

A propósito...

INPUT

O TK85 ou CP 200 não são compatíveis com a entrada de séries de valores separados por vírgulas. Desse modo, a linha 310 deve ser substituída por: 310 INPUT D, com o acréscimo das linhas: 312 INPUT M\$ e 314 INPUT A. Como a data vai agora ser fornecida em três estágios, deve-se substituir as instruções PRINT por: 280 PRINT "DIA?"; acrescentam-se também as linhas: 311 PRINT "MES?" e 313 PRINT "ANO?"

REM

Caso sejam usadas no TK85 ou CP 200, é aconselhável que as instruções REM estejam em linhas separadas.

END

Não existe esta instrução no TK85 ou CP 200 e, assim, a linha 520 deve ser mudada para: 520 GOTO 1000, e deve ser acrescentada a linha: 1000 REM FIM DO PROGRAMA.



Quem é o quê?

Muitas são as categorias profissionais na área da computação. Conheça as atividades de cada uma delas.



Os gigantes amáveis

Os enormes computadores comerciais como esse (conhecidos como computadores de grande porte, para diferenciá-los dos míni e microcomputadores) exigem uma equipe de operadores altamente treinados, para manter todas as unidades em operação com alto rendimento. As máquinas desse porte são capazes de processar centenas de programas simultaneamente e atender a milhares de usuários em qualquer lugar do mundo, por meio de linhas telefônicas, microondas e satélites de comunicações.

O uso crescente de computadores em casa e na escola está produzindo bons programadores — muitas pessoas que, de outro modo, não considerariam a possibilidade de uma carreira na área da computação. Mas a realidade é que, como sempre, um conhecimento superficial é algo perigoso, especialmente em se tratando da linguagem BASIC.

É importante compreender que os requisitos de um programador profissional são fundamentalmente diferentes dos de um usuário doméstico, e muitos dos atributos não podem ser transferidos de um para o outro.

Para o estudante que concluiu o colegial, e tenha profundo interesse em computadores, um curso de nível universitário ou o ingresso direto na carreira de computação parecem ser as opções mais aconselháveis. Diversas faculdades oferecem cursos com qualificação em computação, e os estudantes bem-suce-

didados poderão escolher seu trabalho entre as muitas ofertas do mercado. O problema de desemprego na área da computação tem sido restrito ao escalão de menor nível — digitadores e operadores —, mas a demanda de programadores, analistas de sistemas e engenheiros de projetos e de manutenção continua bastante alta.

Uma alternativa que está se tornando cada vez mais procurada é a do ensino de computação nas escolas. Até agora, a computação, como matéria, tem sido do domínio de faculdades. A área educacional sofre a carência de pessoal treinado em computador, e tal carreira, sem dúvida, será muito gratificante.

Faça sua escolha

Há, talvez, seis níveis principais de hierarquia na área da computação. O primeiro grau pode ser considerado o do “usuário habilitado”. Esta categoria inclui funcionários que aprenderam como operar computadores em determinadas tarefas, tais como processamento de palavras ou contabilidade. Muitas vezes, esses funcionários são selecionados de outras ocupações — por exemplo, administração de escritório, controle de estoque — e devem aprender as tarefas de operador de terminal, operador de perfuração de cartão etc. Esses empregos exigem conhecimentos de nível colegial ou universitário e capacidade de pensar claramente. Qualificações como a operação do teclado são normalmente ensinadas no próprio emprego.

O passo seguinte é o operador de computador. Embora os computadores utilizados em grandes empresas sejam bastante diferentes na aparência dos computadores domésticos, são baseados nos mesmos princípios e, assim sendo, alguma familiaridade com estes é de grande valia. Os operadores rapidamente compreendem o modo de operação dos computadores, e tornar-se um operador é um bom trampolim para a carreira de programador. Deve-se ter em mente, contudo, que o trabalho pode exigir algum esforço físico. A maioria das grandes instalações está em operação 168 horas por semana.

Para tornar-se programador, os atributos necessários são mente clara e lógica e capacidade de se concentrar nos mínimos detalhes. Um programador hábil deve ter uma aptidão muito especial; apesar de, em geral, ser exigida somente formação colegial, a capacidade natural para trabalhar logicamente conta mais pontos. Muitas vezes, os programadores começam a trabalhar sem qualificações formais, o que atrai muitos pais, esperançosos na capacidade de programação de seus filhos.

Como mencionamos antes, o conhecimento do



Uma opção de vida

Analistas



Antes de iniciar uma carreira, é importante observar os objetivos e recursos disponíveis. O **analista de sistemas** tem a tarefa de entrevistar os usuários, para definir suas necessidades, combinar essas necessidades com os recursos e sugerir um método para a solução do problema. A fim de desenvolver um sistema de trabalho para

outras pessoas, o analista precisa ter pensamento lógico, boa habilidade de comunicação e criatividade. É ele freqüentemente o vendedor do departamento, portanto deve sempre causar impressão favorável a seus "usuários" — os que utilizam o computador na empresa.

Programadores

O **programador** recebe a ampla estratégia estabelecida pelo analista e a converte, em primeiro lugar, em um plano tático, dividindo o trabalho em segmentos; em seguida, codifica-os para que o computador possa reconhecê-los e interpretá-los.

Programadores de aplicações dedicam-se à elaboração de programas para executar uma tarefa específica, enquanto os

programadores de sistemas estão mais

envolvidos com o desempenho global do sistema de processamento de dados. Os programadores de aplicações trabalham em isolamento, apesar de fazerem parte de

uma equipe de projetos. Para eles, a capacidade de concentrar a atenção na tarefa existente é realmente importante. Os programadores de sistemas precisam dessa capacidade, mas, também, de uma visão de conjunto. "Se você puder manter a calma quando todos já a perderam...", talvez possa ser um programador de sistemas.

Os **operadores** em instalações menores são comumente chamados para auxiliar programadores e engenheiros no diagnóstico de erros, além de processar o trabalho existente. Contudo, o mais importante é o conhecimento minucioso do método de operação do programa. Os softwares do tipo "user-friendly" tornam mais fácil o trabalho do operador, e um programa bem elaborado pode ser processado por pessoal relativamente inexperiente, com pequena perda de eficiência.

Em qualquer empresa moderna, o departamento de computação é organizado em linha hierárquica. O chefe é o gerente de processamento de dados, responsável pelas diversas atividades necessárias ao processamento de informações.

Todos os profissionais de computador são primeiramente técnicos, e adquirem habilidades gerenciais à medida que progredem nas categorias. As três principais especializações são a operação do computador, programação e análise de sistemas; e há um elemento de mobilidade entre essas categorias que possibilita as promoções.

Em comum com outras profissões, está a necessidade de especialização profissional. Mesmo que não pareça fazer muita diferença no início, uma pessoa menos qualificada logo encontrará barreiras. Será difícil obter um grau universitário enquanto se trabalha em período integral. Por

Engenheiros de desenvolvimento

Enquanto não chega o tempo em que os próprios computadores desenvolverão a nova geração de máquinas, é no cérebro do



engenheiro de desenvolvimento que ocorre esse processo de inovação. Este engenheiro é em parte cientista e em parte um técnico. Sua tarefa é beneficiar-se das

novas descobertas e desenvolvimentos teóricos, para aperfeiçoar e melhorar o desempenho de determinada peça do equipamento. Os títulos de pós-graduação são muito freqüentes neste campo.

Engenheiros de manutenção

Não raro, a única chance de um operador descansar é quando o computador apresenta um defeito, e o **engenheiro de manutenção** é

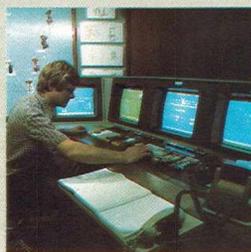
chamado para consertá-lo. Devido à capacidade dos computadores modernos em diagnosticar suas próprias falhas, e à adoção quase universal da construção

modular, a tarefa do engenheiro tornou-se mais simples. Ele precisa ser competente em eletrônica digital, deve ser habilitado em mecânica e capaz de trabalhar com elementos de precisão.



Operadores

Em uma indústria, o trabalho que mais exige fisicamente do funcionário é operar um grande computador em multiprogramação/multiusuário.



Contudo, além de andar quilômetros entre prateleiras de discos, fitas ou caixas de papel, o operador precisa ter muita familiaridade com o sistema operacional do computador, e saber a qualquer momento a importância relativa das tarefas que estão

sendo processadas na máquina. Um **operador sênior** participará de decisões que afetam os negócios da empresa, através da permissão ou não de acesso ao sistema do computador.

isso, os que já possuem essa formação terão mais facilidade de ascender nessas carreiras, embora excelentes profissionais da área constituam exceções a essa regra.

Intelectualmente falando, a categoria que apresenta menores exigências é a do **operador de entrada de dados** (digitador). As

qualificações necessárias aqui são velocidade e precisão. A tarefa é um tanto cansativa e repetitiva, porém em muitas instalações isso é compensado pela oportunidade de se envolver com outras atividades do departamento de computador.





BASIC não é necessariamente uma porta aberta à atividade da computação. Embora seja uma linguagem popular nos microcomputadores, para a maioria dos profissionais é mal estruturada. Consideram que ela favorece maus hábitos de programação e um fio de pensamento descuidado. Este é um problema real, pois a maioria das crianças com experiência em microcomputadores aprendem BASIC com maior facilidade do que as linguagens mais bem estruturadas como LOGO ou COBOL.

Alguns professores de faculdades expressam preferência por alunos que não tenham aprendido o BASIC, porque acham que essa linguagem forma determinados hábitos, difíceis de ser alterados.

Apesar disso, muitos jovens estão encontrando uma forma de empregar suas habilidades em programação BASIC. Escrevem nessa linguagem jogos que interessam a outros jovens, e empresas de software desenvolvem esses entretenimentos adequados à mente adolescente. Algumas das “crianças gênios”, que aparecem nos jornais ganhando apreciáveis somas em dinheiro, escrevem apenas em BASIC e não têm realmente profundo conhecimento de computação. Outros são fenômenos genuínos, escrevendo em Assembler (linguagem de nível inferior que controla o código de máquina de modo muito eficiente), e fazem prever um futuro realmente promissor. Os repórteres de jornais nem sempre distinguem os dois tipos, e tais notícias podem levar os pais a pensar que o interesse de seus filhos por computadores é uma chance de se tornarem milionários, o que não é impossível, mas pouco provável.

No mundo dos negócios

Na indústria dos computadores propriamente dita, os programadores são divididos em dois grupos: programadores de aplicações e programadores de sistemas. Os primeiros escrevem programas para executar uma tarefa específica. Os programadores de sistemas são os “controladores”; escrevem programas para manter em ordem o sistema do computador — por exemplo, para detectar erros. O programador de aplicações provavelmente encontrará os clientes fora da sala do computador e é possível que trabalhe como parte de uma equipe que desenvolva programas para uma tarefa específica. Os programadores de sistemas são mais especializados e trabalham sozinhos. Eles se dirigem diretamente à “inteligência” da máquina.

Mas, a partir deste ponto, a computação traça uma linha divisória além da qual é negado o acesso a muitos, exceto aos programadores mais brilhantes e universitários mais qualificados. Este é o domínio dos analistas de sistemas e dos engenheiros de projetos.

O analista de sistemas considera um problema e, depois, decide de que modo o computador pode auxiliar na sua resolução. Por exemplo: uma companhia petrolífera descobre um novo depósito sob o leito marítimo. Calculam a extensão da jazida e descobrem que a qualidade do petróleo varia muito. A companhia deve decidir sobre o investimento ou não dos bilhões de dólares necessários à exploração do campo de petróleo.

A decisão será baseada em projeções sobre o estado do mercado internacional de petróleo durante a vida do campo (digamos, vinte anos), e a companhia vai decidir qual porção do campo deverá perfurar em primeiro lugar. Dado o volume do investimento, a companhia envia o problema ao pessoal do computador para análise.

O analista considera o problema, consulta economistas, especialistas no mercado de petróleo, geólogos e outros técnicos, e durante um extenso período de tempo constrói um “modelo” de computador que representa o campo de petróleo.

Os dirigentes da empresa podem então fazer perguntas do tipo “E, se...?” a esse modelo, descobrindo as diversas opções sobre preço, técnicas de refinaria e enfoques de mercado que deverão afetar o desempenho global. Eles recebem todas as informações necessárias para a decisão final: o melhor modo de explorar a jazida de petróleo.

Há diversos outros papéis importantes na área da computação, embora poucos tenham tanto prestígio quanto o do analista de sistemas. A exceção talvez seja, na indústria que fabrica o computador, o engenheiro de projetos do hardware. Existem oportunidades para engenheiros eletrônicos em todos os níveis, desde os centros de manutenção aos departamentos de pesquisas; porém, as áreas de desenvolvimento de produtos e pesquisa pura estão abertas somente aos engenheiros eletrônicos altamente qualificados.

Muitos dos analistas e projetistas, tanto das máquinas quanto do software, são promovidos a posições de nível gerencial e de consultoria, e estes títulos freqüentemente indicam que o indivíduo está trabalhando em uma posição de maior poder. O teor do trabalho, entretanto, permanece o mesmo.

Atualmente, há grande escassez de pessoal habilitado em computador e, ao mesmo tempo, numerosos desempregados, muitos deles com graus técnicos e universitários. Esse desencontro de qualificações preocupa educadores e industriais, e algumas providências estão sendo tomadas para retificar essa situação, incluindo programas de treinamento para os qualificados em outros campos e uma variedade mais ampla de oportunidades para o aprendizado em todos os níveis.

Os governos de muitos países consideram que a microeletrônica pode oferecer a resposta a alguns dos problemas de desemprego a curto prazo. Nesses países são desenvolvidos projetos que fornecem treinamento e experiência de trabalho para os recém-formados desempregados. Através desses projetos, os recém-formados aprendem os diversos aspectos da microcomputação, ao mesmo tempo que recebem um subsídio de treinamento que possibilita a sua sobrevivência.

Outros projetos oferecem alguma familiarização com o computador àqueles que nunca tiveram contato direto com a máquina na escola (porque deixaram a escola antes da chegada do computador, ou porque não foram “escolhidos” para utilizá-lo). Isso aumentará, sem dúvida, suas probabilidades de encontrar trabalho, pois para aqueles que deixaram a escola sem algum conhecimento do computador as perspectivas de emprego dentro em breve poderão parecer sombrias.



David Simmonds

Com 17 anos, ele ganhou o equivalente a 15.000 dólares durante as férias.

É um gênio da programação, que escreve programas para a Commodore (fabricante estrangeiro dos computadores PET e Vic). David desenvolve software “sério”, de aplicações comerciais, e pretende encontrar uma posição adequada na indústria do computador, ao terminar seus estudos.

David começou brincando, mas logo abandonou os jogos, absorvendo-se na tarefa de descobrir de que forma eram escritos os programas de computador. Inicialmente, David teve alguns de seus programas publicados em uma revista e, aos poucos, começou a vender cópias de seus programas.

Desta forma, a Commodore tomou conhecimento do fato e David mostrou ao fabricante o que ele poderia fazer. O resultado foi o seu primeiro contrato de programação.



Eugene Evans

Tem 17 anos de idade e diz-se que ganha 60.000 dólares anuais!

Eugene é um dos muitos gênios que estão surgindo na programação. Ele está auxiliando a manter a Imagine Software de Liverpool entre os maiores produtores de jogos para computador na Inglaterra.

Os altos ganhos destes gênios da programação são obtidos mediante *royalties* nas vendas. Os jovens são os mais indicados para o desenvolvimento de jogos, que se destinam a outros jovens — o principal mercado dos jogos de computador.



O mapa lógico

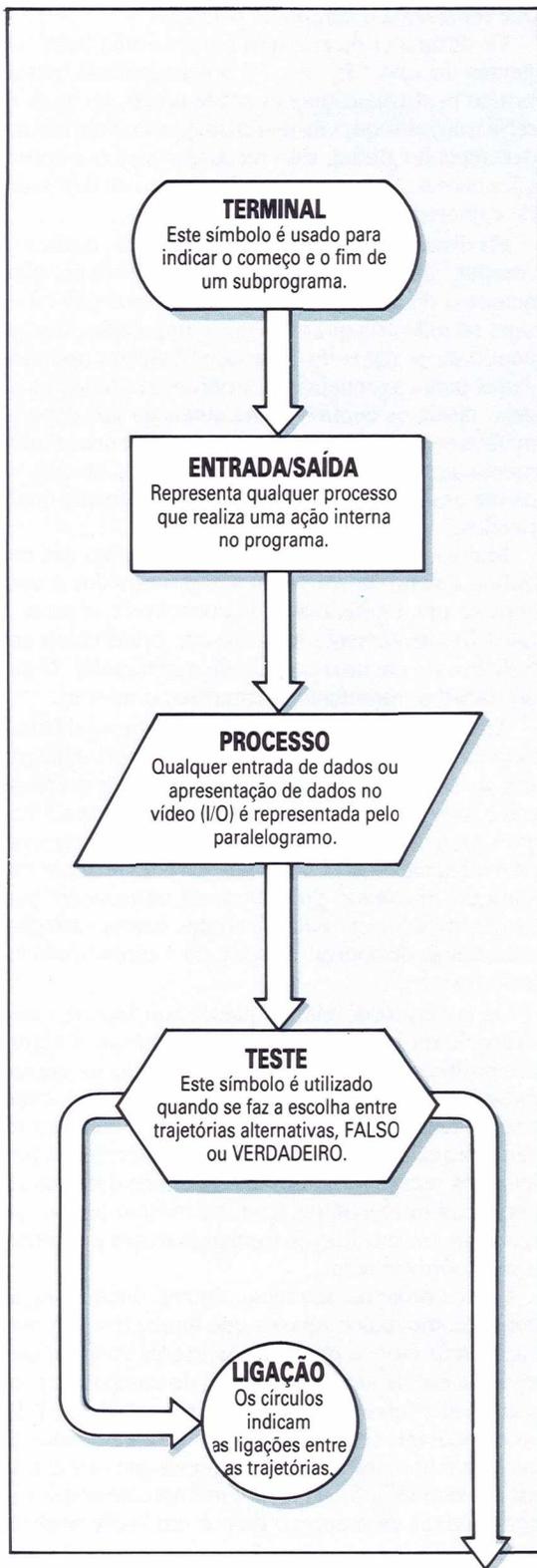
Os fluxogramas ajudam você a criar programas eficientes e com uma estrutura lógica bem elaborada.

O fluxo de informação

O objetivo do diagrama é indicar, de maneira concisa e simples, o fluxo de informação e controle em um programa de computador.

Os pontos de teste são os principais, quando o controle passa para um ponto que não é o seu consecutivo. Uma simples representação gráfica desta passagem de controle é de mais fácil compreensão do que um comando semelhante em forma escrita. Às vezes uma representação gráfica vale bem mais que milhares de palavras.

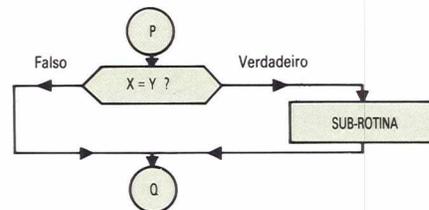
O símbolo de "teste" pode ser representado por um hexágono plano, como é mostrado aqui, ou por um losango.



Um problema pode ser representado de maneira ilustrativa por diagramas que mostram as fases de um processamento e a trajetória dos fluxos que o ligam. Esses fluxogramas são úteis para se compreender um problema e elaborar uma solução.

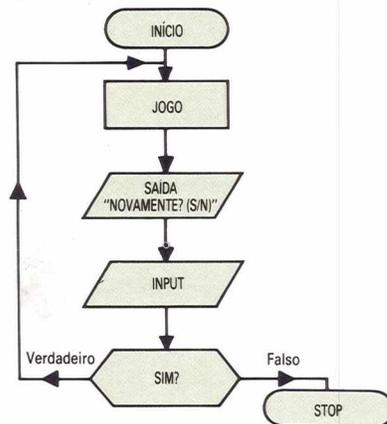
No fluxograma, cada símbolo representa um processo ou ação, e as linhas que ligam esses símbolos representam as possíveis trajetórias a serem feitas. Em um "fluxo de tráfego" de mão única, as setas indicam a direção, que no diagrama é normalmente de cima para baixo e da esquerda para a direita.

Todas as vezes que uma decisão deve ser tomada, um "símbolo de decisão" com a forma de um hexágono ou de um losango é usado. O controle flui em uma única trajetória, como anteriormente, mas pode sair em uma das direções, dependendo do resultado do teste em questão. Se o objetivo do teste é adotar ou não um procedimento, apenas uma das trajetórias de saída terá um "símbolo de procedimento". Aqui está um exemplo de um teste que visa a decidir se o desvio para a sub-rotina deve ser feito:



```
120 IF X = Y THEN GOSUB 300
```

O símbolo de decisão é também utilizado para indicar o teste que termina um loop. No exemplo abaixo, o controle retorna para o começo do programa, se houver uma resposta positiva para a pergunta "NOVAMENTE?"



```
90 REM** COMEÇO DO JOGO**
100
800
```

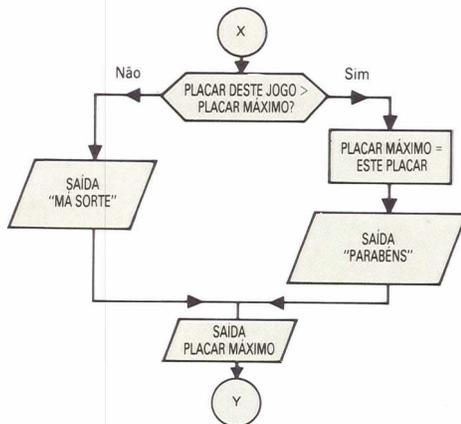


```

810 PRINT "NOVAMENTE? (S/N)";
820 INPUT R$
830 IF R$ = "S" THEN GOTO 100
840 END

```

Talvez queiramos tomar uma decisão que irá resultar em um de dois cursos de ação totalmente diferentes a serem seguidos. No exemplo abaixo, comparamos o placar do jogador com o placar máximo dos jogos anteriores:



```

1200 IF PLACJOGO > PLACMAX THEN GOTO 1230
1210 PRINT "MA SORTE, CONTINUE TENTANDO";
1220 GOTO 1250
1230 LET PLACMAX = PLACJOGO
1240 PRINT "PARABENS! NOVO RECORDE!";
1250 PRINT PLACMAX

```

Note que o valor de PLACMAX é impresso em ambos os eventos, e as duas trajetórias de fluxo possíveis reúnem-se no processo e tornam-se uma única entrada para esta operação de saída.

Todas as decisões são tomadas com base em testes semelhantes a este, que trazem um resultado negativo ou positivo, falso ou verdadeiro. Como se pode notar, este processo de decisão totalmente binário não admite respostas duvidosas. Você pode usar quantos símbolos de decisão quiser, mas é importante que duas trajetórias de saída sejam adequadamente marcadas.

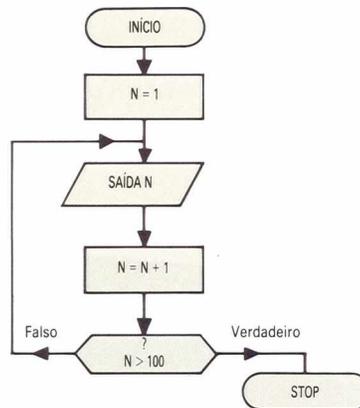
Toda linguagem de programação tem um comando de decisão básico. Se o resultado positivo (verdadeiro) for alcançado, ele leva a uma declaração condicional; caso o resultado seja negativo (falso), ele transfere o controle para o comando seguinte. No caso de versões de BASIC que permitam somente o IF-THEN simples, temos de imitar a escolha condicional através de um comando GOTO, como na linha 1200 do último exemplo. O comando da linha 1210 só será executado se o resultado do teste na linha 1200 for falso.

Mas o que dizer do segundo uso de GOTO na linha 1220? Como se pode perceber, o uso do GOTO no fim do teste, para resolver o problema do destino da escolha condicional, forçou-nos a usar este método para reunir as duas trajetórias de controle possíveis, neste caso, na linha 1250.

O uso dos fluxogramas estimula a introdução dos

GOTOs como forma de seguir toda a representação do programa. Geralmente, o uso de saltos incondicionais é muito perigoso. No caso de a versão do BASIC utilizada forçar essa solução, o fluxograma é um excelente meio de se avaliar o modo como o controle sai da sucessão normal do programa.

Um último exemplo mostra claramente como o uso do fluxograma nos permite representar exatamente as etapas necessárias para se realizar uma simples tarefa: imprimir todos os números entre 1 e 100.



```

10 LET N = 1
20 PRINT N
30 LET N = N + 1
40 IF N > 100 THEN END
50 GOTO 20

```

O uso de fluxogramas como o aqui mostrado tende a incentivar um processo, passo a passo, de programação escrita que, especialmente em grandes projetos, leva a um resultado bastante deslegante. O uso da repetição FOR-NEXT geralmente é indicado até mesmo para pessoas com um conhecimento superficial da linguagem BASIC. Por exemplo:

```

10 FOR N = 1 TO 100
20 PRINT N
30 NEXT N
40 END

```

O fluxograma é incapaz de representar esse pequeno programa BASIC, e a tentativa de segui-lo traria outros problemas. Entretanto, o fluxograma dá algumas informações sobre a estrutura de repetição FOR-NEXT, por isso tem seu valor. Ele demonstra de modo claro como são construídos esse e muitos outros comandos BASIC.

Os fluxogramas são úteis no estágio de planejamento e definição da programação, especialmente nas partes mais complicadas. Os programadores experientes tendem a usá-los menos que os iniciantes, e apenas recorrem aos fluxogramas para ilustrar e documentar um software escrito sem sua participação direta.

O fato de o fluxograma ser desenhado no papel ou estar apenas no subconsciente do programador não faz diferença. A diagramação do fluxo de informação e controle é essencial como forma de solucionar problemas.



A casa automática

Não diga que você não tem computadores em casa antes de ter absoluta certeza disso.

Quantos chips microprocessadores você tem em casa? É claro que haverá um dentro do seu microcomputador, mas o que dizer da máquina de lavar roupa, do aparelho de som e talvez do videocassete?

Tudo, desde o forno até o controle de ignição e o painel de seu carro, pode eventualmente ter um chip como responsável por seu funcionamento.

Não se esqueça da calculadora e do relógio digital, exemplos mais usados de microprocessadores de produção em massa.

O brinquedo das crianças

Em casa, as crianças geralmente usam mais o computador do que os adultos, pois o aceitam tão naturalmente quanto um aparelho de TV. O conhecimento de novos brinquedos eletrônicos faz com que as crianças desenvolvam um raciocínio lógico e aprendam sozinhas. Mesmo os jogos mais simples podem ajudar as crianças em casa.

Um software educacional de boa qualidade também é útil (ver p. 81). Os jogos podem educar e estimular; mas o interesse e o talento das crianças serão mais desenvolvidos ao deparar com problemas, quando elas têm oportunidade de programar o computador.

A linguagem LOGO está se tornando bem conhecida entre as várias linguagens disponíveis. Além de ser de fácil adaptação para muitos computadores, oferece enorme potencial, que leva à melhor maneira de resolver um problema em campos de aprendizagem bastante diversos.



O chip que limpa

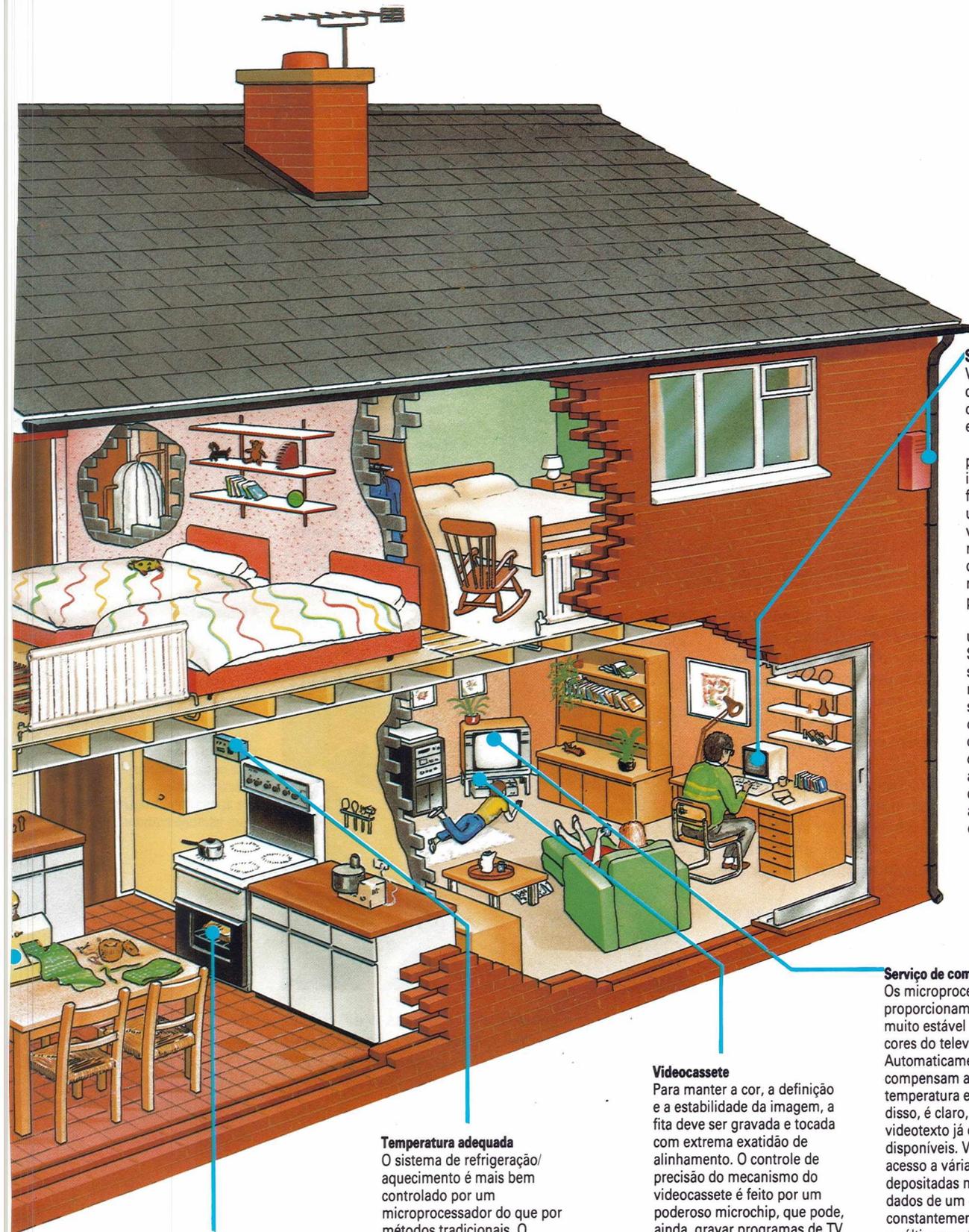
Algumas máquinas de lavar utilizam um microprocessador para selecionar e controlar os vários ciclos de lavagem. As melhores combinações de temperatura, ciclo de lavagem, nível de água e velocidade de rotação são escolhidas com um único toque. Por não haver partes móveis, exceto o tambor, é claro, a vida de sua máquina de lavar roupa será mais longa e o custo da manutenção mais baixo.

O micromóvel

Os computadores em automóvel estão sendo feitos para proporcionar economia e novas funções. Calculam o consumo de combustível, a velocidade, dão alarme em tentativa de roubo e falam com o motorista para avisar a baixa pressão do óleo ou a bateria fraca.

O chip de costura

A máquina de costura tradicional faz pontos bonitos e firmes, requerendo apenas paciência e habilidade da costureira. A máquina de costura controlada por um microprocessador ajuda na criação de bordados complicados ou pontos difíceis, sem grande esforço. Seleciona pontos preestabelecidos e produz outros pontos que ficam armazenados na memória, mesmo quando a máquina estiver desligada.



O fim da comida queimada
Um forno com computador faz assados perfeitos, mediante preciso controle de tempo e temperatura. Enquanto o alimento é assado com toda a precisão, você armazena receitas no seu microcomputador e pode acessá-las facilmente quando desejar.

Temperatura adequada
O sistema de refrigeração/aquecimento é mais bem controlado por um microprocessador do que por métodos tradicionais. O relógio eletrônico possibilita que se programem diferentes temperaturas para os dias úteis e os fins de semana. Áreas diversas, quartos, estufa de plantas e garagem podem ter programas próprios de controle de tempo e temperatura. Isto significa gastar menos energia elétrica.

Videocassete
Para manter a cor, a definição e a estabilidade da imagem, a fita deve ser gravada e tocada com extrema exatidão de alinhamento. O controle de precisão do mecanismo do videocassete é feito por um poderoso microchip, que pode, ainda, gravar programas de TV enquanto você estiver fora. Basta programar o sintonizador automático para os canais e horários que você desejar.

Seu computador doméstico
Você já deve ter usado seu computador em jogos, mas que tal pensar em algo mais emocionante?

Sendo um comerciante preocupado, ou um pai interessado na educação dos filhos, provavelmente já está usando seu computador de várias maneiras. Você pode registrar toda a sua contabilidade ou usar, ainda, o microcomputador como um professor muito paciente.

Mas existem várias outras utilidades não aproveitadas. Se você está interessado em segurança, pode perceber que não há limites para a sofisticação de um sistema que você mesmo constrói. O computador pode localizar qualquer detector escondido, acionar alarmes e até mesmo comunicar-se automaticamente com serviços de emergência.

Serviço de compensação

Os microprocessadores proporcionam um controle muito estável do circuito de cores do televisor. Automaticamente, compensam as variações de temperatura e voltagem. Além disso, é claro, serviços de videotexto já estão disponíveis. Você pode ter acesso a várias informações depositadas no banco de dados de um computador constantemente atualizadas: as últimas notícias de vários setores de atividades, esporte, lazer, cotações de bolsas de valores, saldos bancários, reservas de avião etc.

A ligação que faltava

Com o uso do modem, informações podem ser transmitidas através de linhas telefônicas por quilômetros de distância.



Acoplador acústico

Muitos modems em uso são aparelhos eletrônicos ligados diretamente à linha telefônica, através da saída do aparelho. As companhias telefônicas têm normas muito rígidas em relação a dispositivos como os modems; e isto tende a torná-los mais caros. Uma solução mais barata é o uso do acoplador acústico. Trata-se de um tipo de modem que converte os sinais de som com frequência de onda senoidal em sons reais que alimentam um pequeno alto-falante.

O termo "modem" é a contração das palavras modulador/demodulador. Apesar de existirem no mercado há cerca de cinco anos, só ultimamente os modems passaram a ser usados por um maior número de possuidores de microcomputadores. Para quem tem um micro em casa, quais seriam as eventuais vantagens de ligá-lo ao sistema telefônico através de um modem?

O modem permite que o computador "converse" com qualquer outro computador do mundo; mas isso só pode ocorrer se o outro também tiver um modem. Ele tanto pode ser adaptado a um microcomputador, como ao poderoso equipamento central de uma universidade ou instituição bancária.

A ligação de seu micro a um grande equipamento pode lhe dar acesso a grandes bancos de dados, a serviços de informação e às últimas cotações da Bolsa de Valores. Se ligar seu micro ao de um amigo, vocês podem trocar software, enviar cartas eletrônicas, além de praticar jogos bidirecionais.

Como funciona

O modem trabalha de modo semelhante à interface cassette de muitos microcomputadores. Tanto a interface cassette como os modems convertem os algarismos 0 e 1 do computador em frequências de som. No caso de interface cassette, essas frequências podem ser facilmente registradas em uma fita como se fossem sinais sonoros. Com o modem, as frequências de áudio são simplesmente enviadas pela linha telefônica e convertidas em números binários pelo outro modem.

Assim, a interface cassette apenas precisa converter os dígitos binários em sinais de som, a fim de gravar a fita (este processo é chamado modulação). Ou fazem o contrário: convertem os sinais de som do cassette em dígitos binários (este processo recebe o nome de demodulação).



A máquina FAX

A máquina FAX, ou fac-símile, está se popularizando na Europa e nos Estados Unidos. E no Japão pequenas empresas e residências a utilizam. A máquina FAX pode transmitir documentos longos, incluindo desenhos e fotos para outras máquinas FAX, em questão de segundos. Para isto usa apenas um modem interno e um telefone comum.

Entretanto, a maioria dos modems é desenhada tendo em vista a comunicação bidirecional por uma única linha telefônica. Por isso, precisam de duas faixas de frequência e quatro frequências individuais.

Um modelo popular usa a frequência de 1,070 Hz para 0 (zero) e 1,270 Hz para 1 (um) na transmissão; e 2,025 Hz para 0 e 2,225 Hz para 1, na recepção. Percebe-se que as duas frequências em cada uma das duas faixas (a de baixa frequência e a de alta) estão muito próximas. Existe apenas uma diferença de 200 Hz na frequência de 1 e 0 em ambas as faixas.

Nota-se um contraste com interface cassette, em que a frequência que representa 1 é geralmente duas vezes mais alta que a frequência para 0. A decodificação de frequências tão próximas exige um circuito eletrônico complexo, o que faz com que os modems sejam um artigo de luxo e tenham quase o mesmo preço de microcomputadores domésticos.



TK2000

O TK2000 é um microcomputador de bom desempenho, com características básicas dos equipamentos mais sofisticados.



O TK2000 é um microcomputador pessoal cujos recursos servem também a aplicações profissionais leves. A Unidade Central de Processamento (CPU), a memória e os controladores de entrada e saída estão impressos em uma única placa; o conjunto insere-se numa pequena embalagem plástica, onde também se encontram os conectores para ligação dos periféricos padrão.

O microprocessador do TK2000 é um 6502 de 8 bits, com relógio de 1 MHz. Tem memória de 16 K ROM (para o sistema operacional) e 64 K RAM (para armazenamento de programas e dados). O teclado, incluído na unidade central, constitui-se de 54 teclas do tipo máquina de escrever, com 65 comandos, 46 funções, 59 caracteres alfanuméricos e 49 caracteres gráficos; os caracteres são do tipo ASCII, com matriz de caracteres de 5 x 7 pontos.

Para visualização, o TK2000 requer monitor de

vídeo ou aparelho de televisão, em cores ou preto e branco, ligado à máquina em VHS, canal 2, por meio de cabo co-axial. Em modo de texto, a tela é formada por 24 linhas de quarenta caracteres, ocupando 1 K de memória RAM; em modo gráfico de baixa resolução, é formada por 40 x 48 blocos de até seis cores, ocupando também 1 K de memória RAM; em modo gráfico de alta resolução, o formato será de 280 x 192 pontos, ocupando 8 K RAM. Em modo de texto de alta resolução (exclusivo do TK2000), ocupando 8 K RAM, pode-se observar, ao mesmo tempo, texto e gráficos de alta resolução.

Como todos os microcomputadores, o TK2000 só armazena dados e programas enquanto está ligado. Por isso, para sua utilização plena é preciso conectar a ele um gravador cassete comum, para atuar como memória auxiliar. Desejando-se rendimento máximo do equipamento, utilizam-se dois gravadores

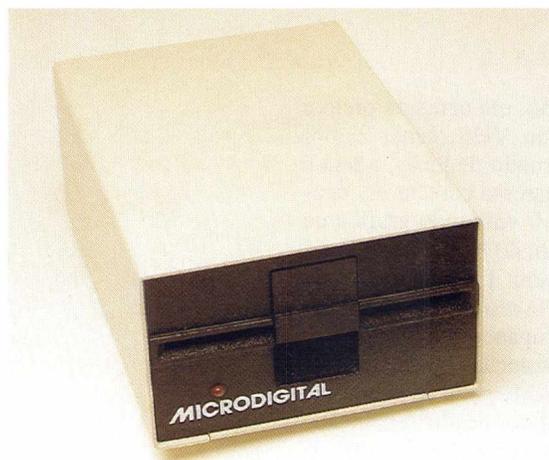


cassete, mantendo um deles em posição de reprodução (play) e o outro em gravação (record). A velocidade tanto de reprodução quanto de gravação é de 1.500 bauds. Se, em vez de gravadores, forem usados discos flexíveis de 5 1/4 polegadas e densidade dupla, pode-se expandir a memória auxiliar do TK2000 para até 173 K por disco.

O TK2000 tem interfaces incorporadas ao sistema, para conexão de monitor de vídeo (ou TV), gravador cassete e impressoras (tipo Centronics) compatíveis com Epson, Seikosha, Globus e Elebra. Além desses, pode-se conectar também cabos para um segundo gravador, cabo de conexão para outros tipos de impressoras, joysticks simples ou analógicos e interface serial assíncrona (RS232 C), para conexão de modems, plotters etc.

O software básico do TK2000 inclui um interpretador BASIC e utilitário (miniassembler e monitor disassembler), operando diretamente em hexadecimal, com capacidade para examinar posições e alterar o conteúdo da memória, transferir e comparar blocos de dados, armazenar e carregar blocos de memória em fita cassete de formato Applesoft ou próprio, somar e subtrair valores hexadecimais e executar programas em linguagem de máquina.

A maior parte dos programas para equipamentos Apple DOS e Apple II Plus é compatível com o TK2000.



Led indicador

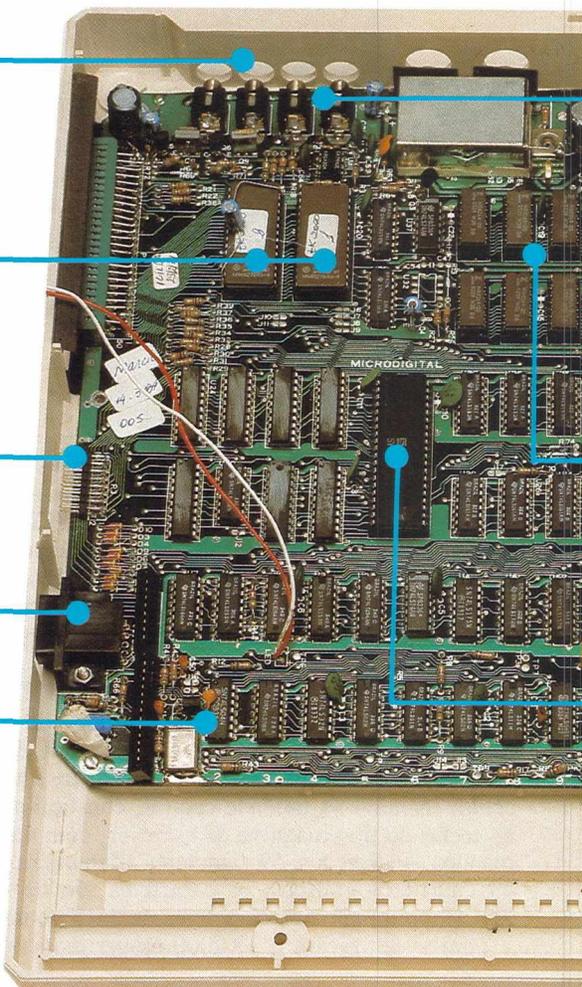
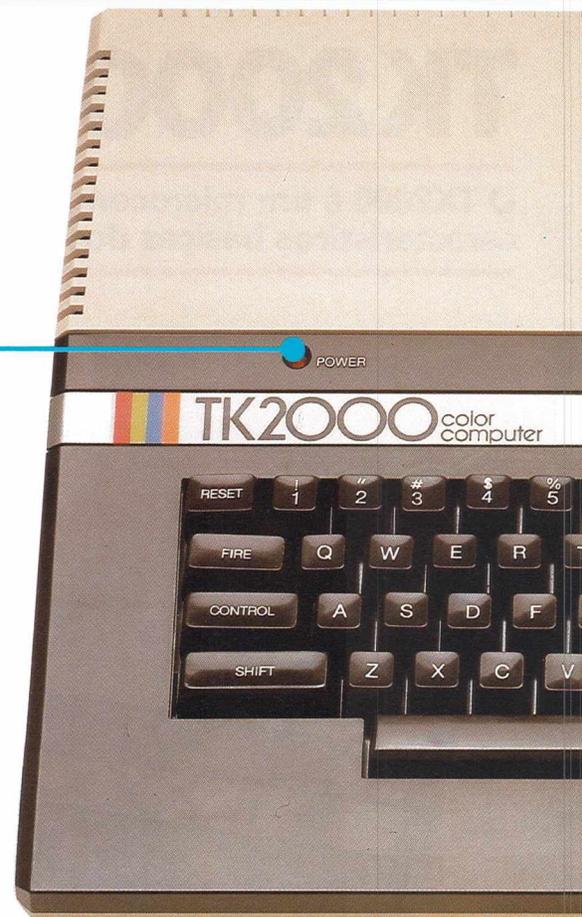
Saída controle de gravador

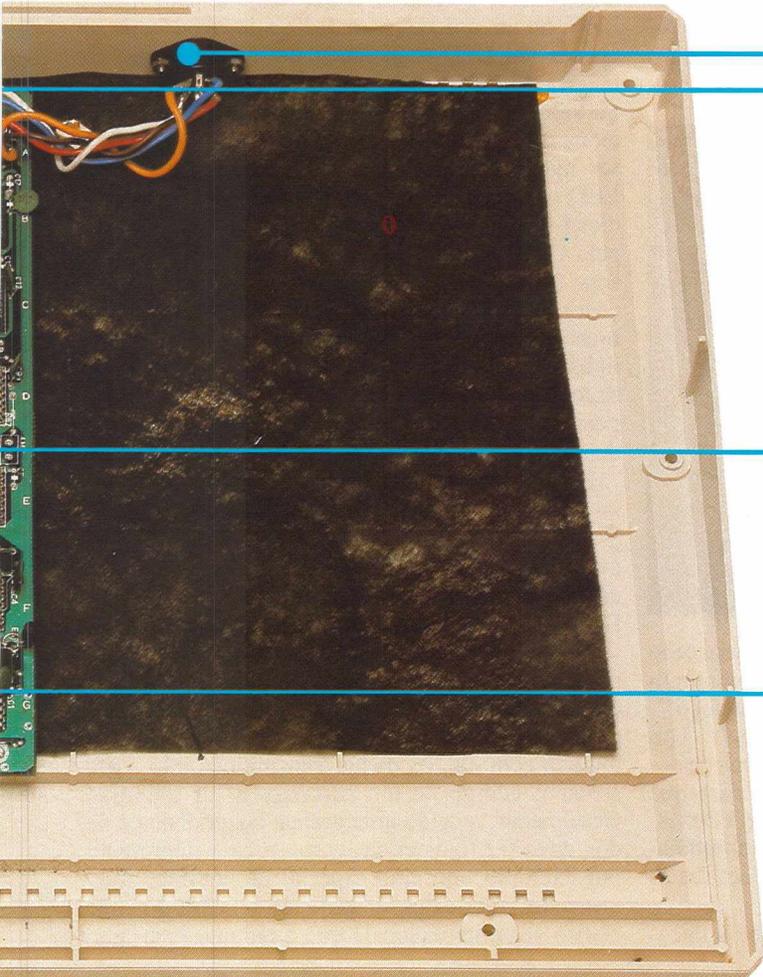
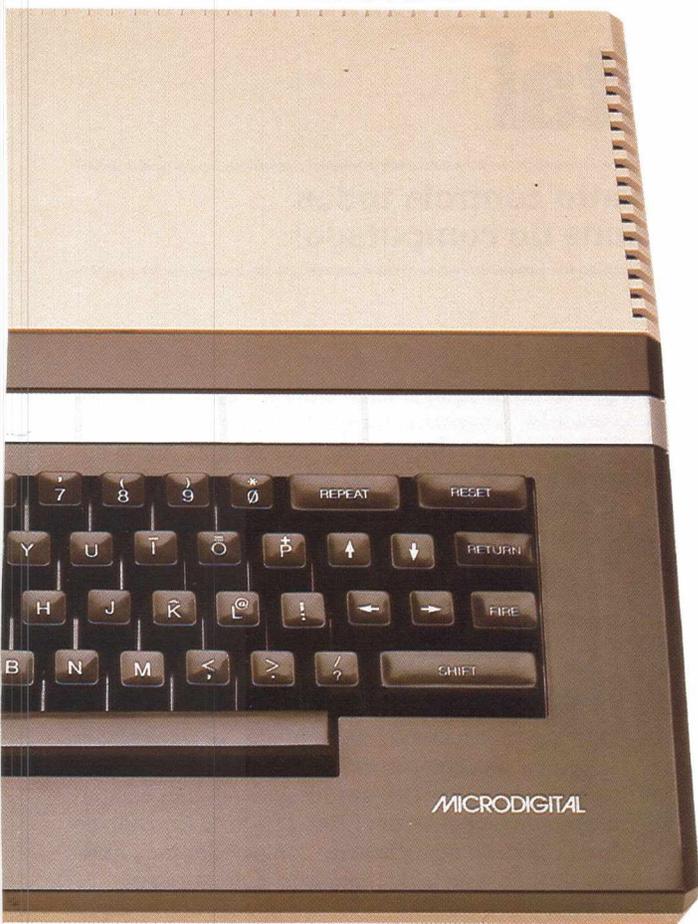
ROM
16 K de memória
ROM

Saída para impressora

Entrada para joystick

Cristal





Entrada de alimentação múltipla

Entrada e saída de gravador

RAM
64 K de memória RAM

CPU
6502 com clock de 1,00 MHz

TK2000

MICROPROCESSADOR

6502

CLOCK

1,00 MHz

MEMÓRIA

Possui 64 K de memória RAM e 16 K de memória ROM com BASIC incluído.

VIDEO

Pode ser ligado a um televisor preto e branco ou em cores (padrão PAL/M) ou a um monitor monocromático ou em cores. São quatro os modos de display: texto (40 caracteres por 24 linhas), gráfico de baixa resolução (40 x 48 pontos), gráfico de alta resolução (280 x 192 pontos) e texto de alta resolução simultâneo.

TECLADO

Alfanumérico, tipo máquina de escrever com 54 teclas multifuncionais e 50 caracteres gráficos.

LINGUAGENS

BASIC

PERIFÉRICOS

Cassete, monitor, unidade de disco (5 1/4"), impressora, interface serial RS232 C, joystick, paddle.

DOCUMENTAÇÃO

O manual do TK2000 apresenta as principais características técnicas do equipamento e da linguagem BASIC. Ao fim de cada capítulo é colocada uma página onde alguns dados devem ser preenchidos pelo usuário e que poderá ser utilizada como resumo. Nos apêndices são descritas as mensagens de erro, as palavras reservadas e é feita uma comparação do TK2000 com o Apple II Plus.

Diálogo digital

A Unidade Central de Processamento controla todos os procedimentos de entrada e saída do computador.

Analogico para digital

No mundo real, poucas informações vêm em unidades digitais discretas. Pelo contrário, elas são muito variáveis como os níveis de ruído, ou as marés.

Para que esses dados sejam compreendidos pelo computador, o sinal deve primeiramente ser digitalizado. O conversor de analógico para digital (A/D) recebe amostras de uma fonte de sinais numa taxa constante, talvez 100 por segundo. Cada uma dessas amostras é depositada num local separado de memória como um valor digital, permitindo que cálculos de variação sejam feitos e as condições fora do limite sejam conhecidas.

Os conversores de digital para analógico (D/A) trabalham de modo similar, mas ao contrário, usando técnicas estatísticas para encaixar os picos numa curva regular.

Input/Output, ou I/O (abreviação comum), são os termos usados para indicar a transferência de informação entre a CPU (a Unidade Central de Processamento que forma o centro do computador) e o "mundo exterior". Aqui, "mundo exterior" significa qualquer aparelho ligado ao computador. Nesse caso não se incluem as memórias ROM e RAM, consideradas partes integrantes do computador. A distinção entre o que faz parte ou não do computador é arbitrária. Mas todos os circuitos lógicos (ver p. 92), que trabalham em estrita conexão com a CPU e a memória central, são considerados parte integrante do computador.

Os equipamentos externos que têm I/O para a comunicação com o computador incluem uma variedade de periféricos: teclado, unidades de discos flexíveis, joysticks, impressora e unidades de vídeo.

Quando a CPU quer acessar dados da memória, a primeira coisa que faz é "endereçar" o local onde o byte de dados é depositado. Do mesmo modo, caso a CPU queira depositar um byte de dados para uso posterior, ela deve primeiramente estabelecer o local onde será depositado. Este processo recebe o nome de "endereçamento de memória", e pressu-

ção com um equipamento externo, ele terá de "localizá-lo" da mesma forma. Somente oito linhas de endereço estão disponíveis; isso permite que o número total de locações I/O separadas que podem ser selecionadas seja de 256. Este número é pequeno, comparado ao poder de dezesseis linhas de endereço; mas, na prática, 256 é uma quantidade adequada. Não há necessidade de ser conectada ao computador uma quantidade maior de equipamentos.

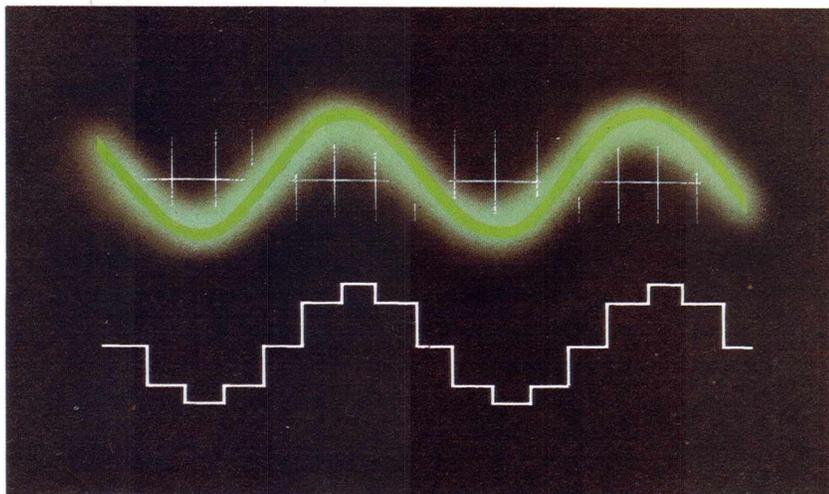
A escolha dos equipamentos

Para descobrir como, na verdade, o computador seleciona um equipamento externo e envia dados para ele, consideremos um dos mais simples equipamentos de output — um LED (Diodo de Emissão de Luz) colocado no teclado do computador para mostrar quando a tecla de maiúscula é pressionada.

Para o computador, o LED é apenas outro dispositivo externo a que envia dados. No caso de um único LED, o dado será um único 1 para ligar o LED; ou um único 0 para desligá-lo. Apesar de ser apenas um LED requerendo um bit de dados, ele precisa ter endereço e posição. A CPU não pode gastar todo seu tempo endereçando um LED. Ela seleciona o LED para que possa dizer quando ele deve ser ligado e depois desligado. Admitamos que o LED tenha um endereço de I/O de 32. A fim de selecioná-lo, as linhas de endereço são estabelecidas pela CPU através do binário equivalente a 32, ou seja, 00100000. O LED terá um circuito "decodificador" especial que irá ignorar todas as outras combinações de bits nas linhas de endereço. Quando a linha de endereço torna-se 00100000, este circuito decodificador a reconhece, produz uma alta voltagem e, conseqüentemente, um output "verdadeiro".

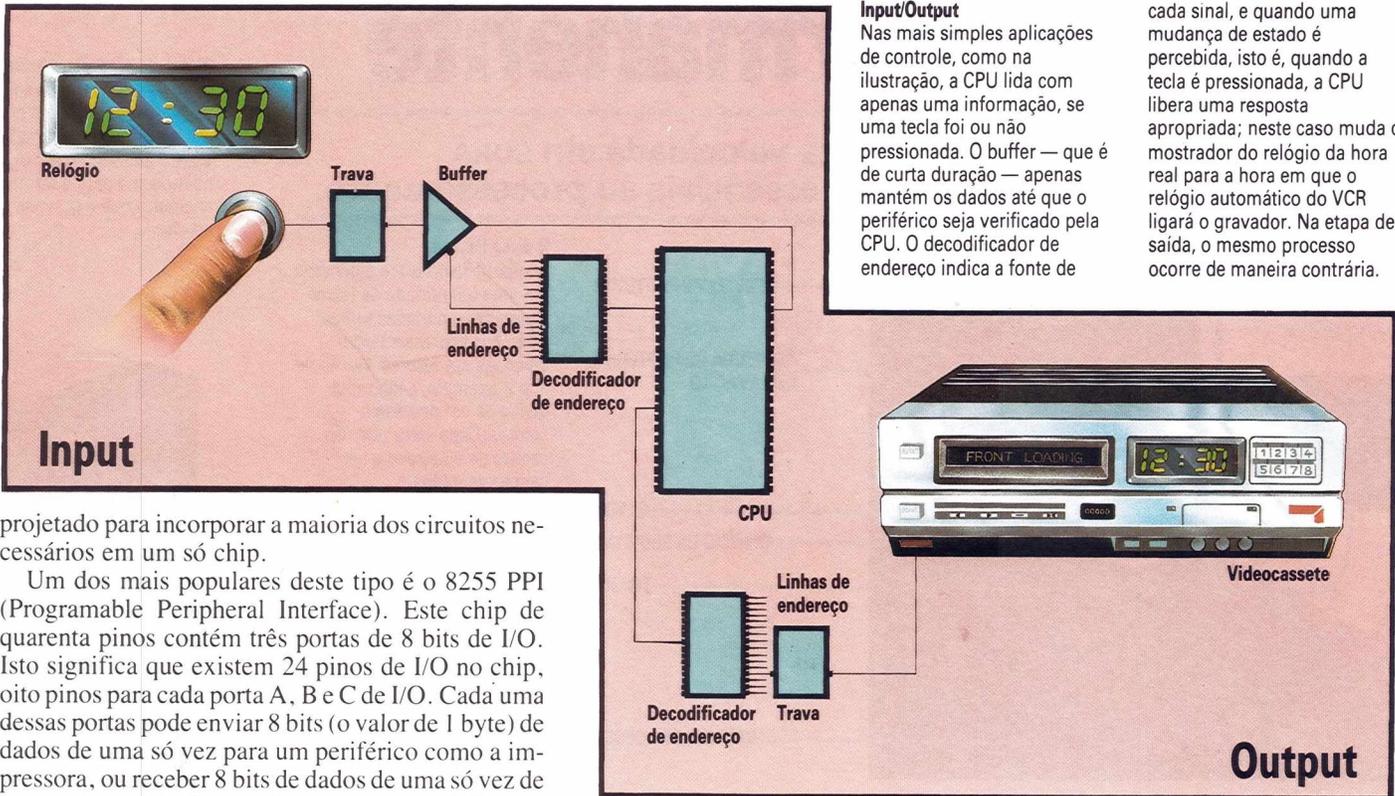
A parte seguinte do circuito que precisa ser ligada no LED é um pequeno chip chamado trava de dados ("data latch"). Este prende o dado enviado, para que o LED fique ligado ou desligado até o próximo endereçamento e novos dados sejam enviados a ele. Este processo é chamado "toggling".

A maioria dos equipamentos externos com que o computador se comunica são mais complexos que um simples LED. A impressora é considerada um periférico típico e, toda vez que o computador se comunica com ela, os dados transmitidos irão representar o código de um caractere a ser impresso. Geralmente, quando uma grande quantidade de dados deve ser transferida, como para a impressora, por exemplo, um chip especial de interface I/O é utilizado. Tais chips simplificam a tarefa do engenheiro de computador, já que o circuito interface é



põe que a CPU coloque os dígitos binários de acordo com a posição de memória desejada num conjunto de dezesseis fios ligados aos "pinos de endereço". Estes fios recebem o nome de "address bus". Um circuito especial na seção de memória é capaz de decodificar estes dezesseis dígitos binários para que selecionem a posição de memória correta. (Dezesseis dígitos binários possibilitam 65.536 combinações dos algarismos 0 e 1 e podem endereçá-las às diferentes locações de memória.)

Caso o computador queira estabelecer comunica-



Input/Output

Nas mais simples aplicações de controle, como na ilustração, a CPU lida com apenas uma informação, se uma tecla foi ou não pressionada. O buffer — que é de curta duração — apenas mantém os dados até que o periférico seja verificado pela CPU. O decodificador de endereço indica a fonte de

cada sinal, e quando uma mudança de estado é percebida, isto é, quando a tecla é pressionada, a CPU libera uma resposta apropriada; neste caso muda o mostrador do relógio da hora real para a hora em que o relógio automático do VCR ligará o gravador. Na etapa de saída, o mesmo processo ocorre de maneira contrária.

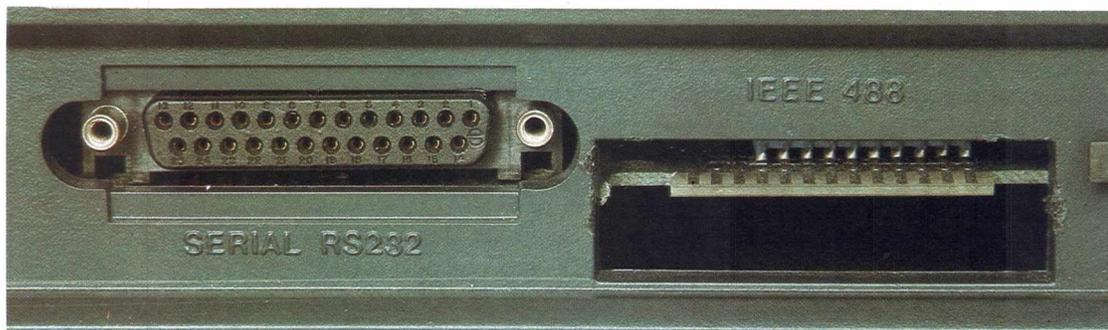
projetado para incorporar a maioria dos circuitos necessários em um só chip.

Um dos mais populares deste tipo é o 8255 PPI (Programable Peripheral Interface). Este chip de quarenta pinos contém três portas de 8 bits de I/O. Isto significa que existem 24 pinos de I/O no chip, oito pinos para cada porta A, B e C de I/O. Cada uma dessas portas pode enviar 8 bits (o valor de 1 byte) de dados de uma só vez para um periférico como a impressora, ou receber 8 bits de dados de uma só vez de um equipamento de entrada, como um teclado, por exemplo.

A fim de enviar 8 bits de dados para uma impressora, a CPU primeiro terá de endereçar o PPI; só então enviará a ele os 8 bits de dados através do "data bus". Esses dados serão depositados temporariamente numa célula de memória de 1 byte, dentro do chip, chamada registrador. A seguir, o PPI fará com

nicas principais que podem ser usadas. A CPU interrompe periodicamente a execução do programa e verifica todas as portas de entrada. Se ela encontra dados à espera, instrui a porta para que os coloque no "data bus". O processo de inquirição dos periféricos de entrada é conhecido como "polling".

A outra técnica utiliza "interrupções". O equipa-



que esse dado esteja disponível a um grupo apropriado de pinos de I/O.

Um princípio similar, mas que trabalha ao contrário, permite que dados de periféricos de entrada sejam armazenados num registrador no chip, e depois colocados no "data bus", quando a CPU lhes enviar o sinal apropriado.

Como se nota acima, os periféricos não podem ser liberados para colocar continuamente seus dados no "data bus" do computador; é necessário que se permutem dados da memória e de outros equipamentos de I/O. O chip de I/O armazena os dados temporariamente, e só coloca os dados no "data bus" (para serem colhidos pela CPU) quando a CPU ordena que isso seja feito.

Como a CPU sabe se um periférico está tentando enviar dados para o computador? Existem duas téc-

mento que está aguardando envia um sinal de interrupção diretamente à CPU, o que força o programa que está sendo executado a parar, enquanto a porta de entrada envia seus dados. As vantagens e desvantagens de ambos os métodos serão discutidas no curso, mais adiante. O I/O descrito até agora é chamado "I/O paralelo", porque os dados entram ou saem de 1 byte por vez, usando oito fios ou linhas de I/O (8 bits paralelos).

Uma outra técnica recebe o nome de "I/O serial". Neste caso, a informação é alimentada em 1 bit de cada vez, 1 bit após o outro. Algumas impressoras utilizam interfaces seriais, e a saída de modems (ver p. 108) também é serial. A vantagem principal é que, essencialmente, a comunicação serial permite que um único par de fios seja usado, em vez de oito ou mais.

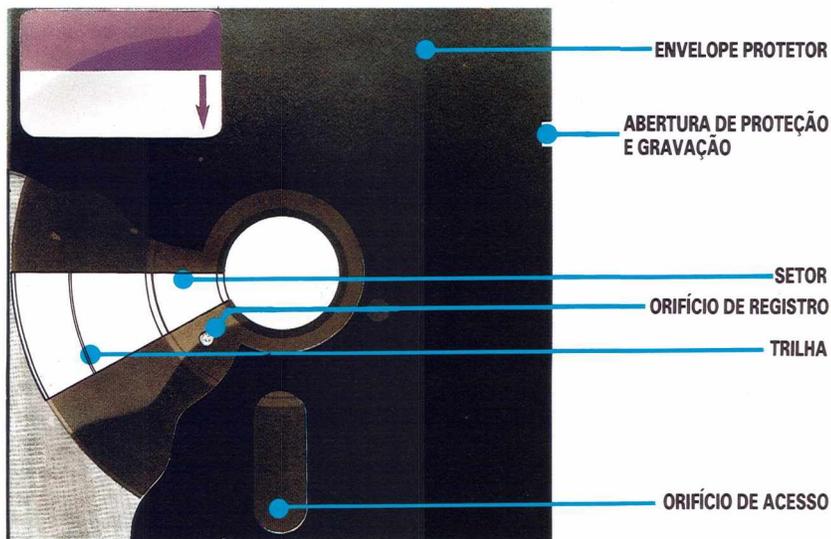
Portas paralelas e seriais

A maioria dos microcomputadores modernos oferece tanto as portas paralelas como as seriais. As portas seriais transmitem os dados em 1 bit de cada vez, e as paralelas em bytes completos. O tipo mais comum de porta serial é conhecido como RS232 C, usa um conector do tipo "D", um exemplo de 25 pinos (ilustração à esquerda), ou, mais raramente, um plugue DIN semelhante aos utilizados em aparelhos de som.

A porta paralela (à direita) segue a convenção IEEE 488, desenvolvida por Hewlett Packard e adotada como modelo industrial pelo Instituto de Engenharia Elétrica e Eletrônica, dos Estados Unidos.

Memória permanente

Os discos magnéticos giram em alta velocidade em suas unidades, fornecendo informações essenciais ao processamento.



O disco flexível

A superfície do disco é dividida em uma quantidade de faixas separadas chamadas trilhas. As trilhas são, mais tarde, divididas em setores. No Apple II, por exemplo, cada trilha divide-se em dezesseis setores. Cada setor tem um campo de endereço e um campo de dados.

O Sistema de Operação de Disco acessa setores individuais da trilha, usando o campo de endereço que contém os números da trilha e do setor e, ainda, um identificador (que verifica se o usuário está lendo o disco certo). Assim, ele pode recuperar a informação de modo semelhante (usando o endereço).

Seu microcomputador esquecerá tudo o que você programou, assim que a energia for desligada: algo bastante desagradável, porque se perde um trabalho resultante, às vezes, de horas de programação. Por isso, os fabricantes de microcomputadores criaram um dispositivo que mantém o conteúdo da memória gravado permanentemente. O dispositivo em geral utilizado é uma fita cassete, em que o programa é armazenado digitalmente, como se fosse uma série de tons (ver p. 94).

Entretanto, quando se trabalha com programas longos, ou com uma coleção de programas pequenos utilizados com frequência, muito tempo é gasto para que sejam encontrados e trazidos do cassete. Esse inconveniente tem duas razões. A primeira é que a fita deve ser lida desde o começo para que se localize o programa; neste caso, um gravador que tenha um contador facilitará o trabalho.

O segundo problema é o modo como o programa é armazenado. As configurações de bits que estão na memória devem ser convertidas em uma seqüência de tons correspondentes: um tom mais alto representa 1 bit que está ligado (ou 1); e um mais baixo representa 1 bit desligado (ou 0). Esses tons devem, então, ser gravados na fita cassete. A maior velocidade em que essa transferência ocorre é 150 bytes por segundo. Se a velocidade for aumentada, existe a possibilidade de os erros aumentarem, o que afetará a confiabilidade do sistema.

Um sistema convencional com uma fita C-10 leva aproximadamente cinco minutos para encontrar um programa. Isto pressupõe que um sistema de carregamento rápido esteja sendo utilizado. Alguns sistemas trabalham mais lentamente, com a velocidade de 30 bytes por segundo. Para programas longos torna-se necessária a utilização de um sistema de gravação que encontre o início do programa, e o carregue em alguns segundos, mantendo sempre a integridade dos dados.

Esse sistema de armazenagem é o disco flexível, usado em muitos microcomputadores. Ele é conhecido também como disquete ou floppy disk. Se você pensar que a grande quantidade de fita de um cassete é transformada em um disco de algumas polegadas, vai perceber a rapidez com que a informação registrada no disco poderá ser localizada. Esse disco tem um envelope protetor e é colocado em uma unidade de disco para leitura e gravação.

A função da unidade é fazer o disco girar (dentro do envelope) numa velocidade constante e transferir programas ou dados do disco para o computador. Essa operação é feita através de uma cabeça de leitura e gravação, semelhante à de um gravador cassete comum, mas bem menor. A cabeça move-se para trás e para a frente na superfície do disco, ao contrário do gravador cassete, que só pode fazer a fita passar pela cabeça.

Ao contrário da fita, que é apenas uma longa se-

Placa de circuito analógico

O circuito converte os sinais recebidos ou enviados pela cabeça de leitura/gravação. Traduz a forma digital usada na máquina para a forma analógica que é gravada no disco.



Indicador

Este diodo de emissão de luz mostra se a unidade de disco está sendo acessada.

Eixo direcional

Ele sustenta o disco plástico e faz com que gire dentro do envelope.

Manutenção do disco

Os discos flexíveis são delicados e devem ser tratados com muito cuidado. Siga cuidadosamente as recomendações do fabricante.



NÃO DOBRE!



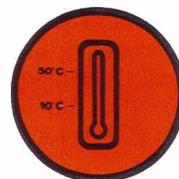
NÃO EMPILHE!



MANTENHA LONGE DE ÍMÃS



GUARDE COM CUIDADO



MANTENHA À TEMPERATURA AMBIENTE



Cabo de ligação

As informações são transferidas para a unidade de disco e dela saem pelo cabo de ligação, que contém as vias de 8 bits de dados e outros sinais de controle.

Conector do cabo de ligação

Oferece segura conexão do cabo de ligação.

Motor

Encarregado de girar o eixo direcional.

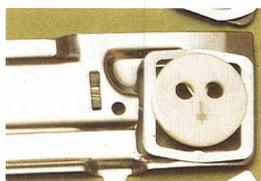
Cabeça de leitura e gravação

Motor gradual e parafuso condutor

Motor elétrico extremamente preciso que move a cabeça na superfície do disco.

"Cantilever" Mecanismo de carregamento

Este mecanismo de alavanca assegura o lugar preciso do disco no eixo direcional, quando a porta da unidade de disco é fechada.



Cabeça de leitura e gravação

Foto ampliada da cabeça que lê e grava os dados na superfície do disco. É semelhante à cabeça de um gravador cassette, mas quase invisível a olho nu.

qüência de bytes, o disco é constituído de uma série de círculos concêntricos; cada um deles é tratado como um pequeno grupo de, geralmente, 256 bytes. Cada um desses "setores" tem um endereço.

Quando o programa vai ser escrito no disco, a cabeça move-se até o diretório, que é um arquivo especial com a função de um índice de todo o disco. O diretório é examinado para se descobrir onde o arquivo deve ser colocado. Caso ele esteja sendo reescrito, o primeiro setor da cópia antiga é encontrado, e os novos dados são gravados a partir daí. Se o arquivo for novo, não constará do diretório; por isso, uma entrada deve ser criada. O primeiro setor vazio é preenchido com os dados, e assim sucessivamente outros setores serão ocupados de acordo com a necessidade.

As vantagens resultantes da alta velocidade e da

grande capacidade de memória que o disco oferece explicam a considerável diferença de preço entre os dois sistemas. As unidades de disco têm preço bem maior que os gravadores cassette.

O fator mais importante nessa diferença de preço é a precisão técnica necessária ao bom funcionamento. A cabeça de leitura e gravação da unidade de disco é quase invisível e deve ser colocada entre centésimos de polegada. Esse minúsculo mecanismo é movido por um motor elétrico, que gira em frações de graus. É acoplado a um eixo que carrega a cabeça e a leva à superfície do disco, em minúsculas distâncias cuidadosamente calculadas. Para possibilitar que o disco gire a uma velocidade constante, utiliza-se uma eletrônica complexa e todos os componentes são montados em uma estrutura fundida, a fim de reduzir os efeitos do calor e da vibração.

Desafie os elementos

As variáveis indexadas, ao contrário das variáveis simples, podem conter qualquer quantidade de elementos.

Em nosso programa anterior, no cálculo do número de dias que faltam para o Natal, encontramos um novo tipo de variável, a variável "indexada". É diferente das variáveis comuns, ou "simples", pois pode conter um número indeterminado de subdivisões ou elementos. As variáveis simples admitem duas letras ou uma letra seguida por um dígito, de 0 a 9 (algumas versões de BASIC admitem o uso de palavras inteiras, como nomes de variáveis). As variáveis A, B, B1, C3 e R2 são todas simples. As variáveis indexadas têm esta forma: A(6), B(12) ou X(20). O indexador é o número entre parênteses. A leitura dos exemplos acima é a seguinte: "A indexado de seis", "B indexado de doze" e "X indexado de vinte".

Se imaginarmos uma variável simples como uma caixa identificada por um nome, podemos visualizar uma variável indexada como uma caixa que contém determinado número de elementos em seu interior. Se desejamos uma variável com doze elementos, nós a criamos, utilizando a instrução DIM, da seguinte forma: DIM A(12). Pode-se usar qualquer letra do alfabeto.

Atribuem-se valores a variáveis simples pelas instruções LET ou INPUT, como: LET A = 35, LET B1 = 365, ou INPUT C3. De modo semelhante, valores são atribuídos aos elementos de uma variável indexada. Vejamos como é possível atribuir valores a uma matriz indexada ("matriz" é o nome alternativo dado a conjuntos de variáveis indexadas). Por exemplo:

```
10 DIM A(5)
```

cria uma variável indexada com cinco elementos. Podemos agora atribuir um valor a cada elemento:

```
20 LET A(1) = 5
30 LET A(2) = 10
40 LET A(3) = 15
50 LET A(4) = 20
60 LET A(5) = 100
```

A fim de perceber como essas variáveis diferem das variáveis simples, experimentemos atribuir valores a algumas variáveis simples:

```
70 LET X = 5
80 LET Y = 6
90 LET Z = 7
```

Experimente fornecer estes dados a seu computador e depois confira o conteúdo de cada variável, com o comando PRINT. Muitas das instruções da linguagem BASIC também funcionam como comandos. Após ter fornecido as instruções acima, confira-as, LISTando-as. Digite em seguida a instrução PRINT X < CR >. O número 5 deverá aparecer imediatamente na tela. Em seguida, digite o comando PRINT Y. O

computador responderá a este comando apresentando o número 6 na tela. Se quiser conferir os elementos da variável indexada, digite o comando PRINT A(1) para verificar o valor do primeiro elemento da tabela. O computador deverá apresentar o número 5 na tela. Experimente usar o comando PRINT para conferir os valores de A(3) e A(5).

A maior diferença entre variáveis indexadas e variáveis comuns está em que o próprio indexador pode ser uma variável. A fim de perceber o que isto significa, digite o comando PRINT A(X). A tela apresentará o número 100. Por quê?

Examine a lista que você digitou e verifique o valor da variável X. O valor é 5 e corresponde a A (o valor da variável X), o que equivale a A(5). Desse modo, digitar PRINT A(X) corresponde exatamente a digitar PRINT A(5). Que valor será obtido se você digitar PRINT A(Y-X)? Antes de fazê-lo efetivamente, experimente calcular você mesmo a resposta.

Atribuição de valores

Se houver apenas algumas variáveis simples, a instrução LET será o modo mais simples de atribuir-lhes valores. As variáveis indexadas podem ter grande número de elementos na matriz; assim, verifiquemos quais são as alternativas para o fornecimento de valores:

```
10 DIM A(5)
20 PRINT "FORNECER AS VARIAVEIS"
30 INPUT A(1)
40 INPUT A(2)
50 INPUT A(3)
60 INPUT A(4)
70 INPUT A(5)
```

Este método de digitação é tão cansativo quanto o uso das instruções LET, embora funcione. Se soubermos quantas variáveis existem realmente (neste caso, cinco), será mais fácil o uso do loop FOR-NEXT, da seguinte forma:

```
10 DIM A(5)
20 FOR X = 1 TO 5
30 INPUT A(X)
40 NEXT X
```

Este programa prevê a digitação de cinco valores no teclado do computador, ao ser processado o programa. A tecla RETURN deverá ser pressionada depois de cada novo número fornecido. Se soubermos previamente quais serão os valores incluídos, será mais fácil fornecê-los com a instrução READ combinada a uma instrução DATA, da seguinte forma:

```
10 DIM A(5)
```



```
20 FOR X = 1 TO 5
30 READ A(X)
40 NEXT X
50 DATA 5, 10, 15, 20, 100
```

Experimente este pequeno programa e verifique o conteúdo da matriz usando o comando PRINT (ou seja, use o comando PRINT após o programa ter sido processado). Por exemplo, PRINT A(1) <CR> e PRINT A(5). Agora, podemos acrescentar ao programa algumas linhas, de modo a instruí-lo a automaticamente fazer por nós o trabalho de impressão dos elementos incluídos na matriz:

```
60 FOR L = 1 TO 5
70 PRINT A(L)
80 NEXT L
90 END
```

Processe este programa e verifique se os valores corretos aparecem na tela. Em seguida, digite novamente a linha 50, usando cinco diferentes elementos DATA. Lembre-se de que os números incluídos na instrução DATA devem ser separados por vírgulas, mas não pode haver vírgulas antes do primeiro número, nem após o último.

O método mais simples de atribuição de valores consiste no uso de instruções DATA e READ. Se os valores variarem toda vez que o programa for processado, o uso de uma instrução INPUT como parte de um loop FOR-NEXT será provavelmente o melhor método. Se o número total de elementos na matriz for fixo, ele poderá ser usado para indicar o limite máximo, na instrução FOR.

Vamos recorrer a tudo que aprendemos até agora para elaborar um programa curto, porém bastante eficaz. Suponha que queremos colocar alguns números em ordem crescente. Antes de escrever o programa, o primeiro passo será elaborar sob forma lógica os procedimentos que podem resolver o problema. Quando o método para solucionar o problema estiver claro, anote as etapas sucessivas, usando sentenças curtas em português.

Vamos supor que iniciamos com cinco números: 4, 9, 2, 8, 3. Colocá-los em ordem crescente é um procedimento simples: basta examinar a linha, verificar qual o menor deles e colocá-lo à esquerda, procedendo do mesmo modo com os demais dígitos.

Todavia, o computador necessita de um conjunto muito preciso de instruções e temos de planejar muito claramente os passos exigidos. Eis uma possibilidade: compare o primeiro dígito com o segundo. Se o primeiro for maior que o segundo, inverta as posições; se for menor, mantenha-os no mesmo lugar.

Em seguida, compare o segundo dígito com o terceiro: se for menor que o terceiro, deixe-os na mesma posição; caso contrário, faça nova substituição.

Repita esse procedimento até atingir o último par de dígitos.

Não havendo necessidade de substituição na linha, todos os dígitos deverão estar na ordem correta. Se ocorreu alguma substituição, volte ao início e repita todo o processo.

Se observar este procedimento, você perceberá que, na verdade, ele pode colocar qualquer conjunto de números em ordem numérica crescente. Veja

como se processa a ordenação do conjunto inicial de números, por esse processo de comparação de dígitos:

4	9	2	8	3
4	2	9	8	3
4	2	8	9	3
4	2	8	3	9

Todos os pares foram comparados e trocados, quando necessário. Uma vez que houve necessidade de pelo menos uma substituição, volte ao início e repita o procedimento:

4	2	8	3	9
2	4	8	3	9
2	4	3	8	9

Ainda houve necessidade de substituições; assim, volte ao início e repita novamente:

2	4	3	8	9
2	3	4	8	9
2	3	4	8	9

Não houve necessidade de substituição, no último exame da linha; em consequência, todos os números devem ser menores, em comparação ao dígito à sua direita. Ou seja, os números estão em ordem crescente e a operação pode ser encerrada.

O uso de variáveis indexadas possibilita a elaboração de rotinas como essa, em linguagem BASIC, pois o indexador pode ser uma variável. Se nosso conjunto inicial de cinco números constituir os valores de uma matriz, de modo que A(1) = 4, A(2) = 9, A(3) = 2, A(4) = 8 e A(5) = 3, caso X tenha o valor 1,

Variáveis indexadas

As variáveis indexadas (variáveis com diversas subdivisões) aumentam sensivelmente a capacidade da linguagem BASIC. Aqui a variável A tem o indexador X + Y - Z. Cada um dos elementos desse indexador consiste em uma variável e o valor de cada uma delas é apresentado no interior das pequenas caixas. X tem o valor 5; Y, 6; e Z, 7. Assim, X + Y - Z equivale a 5 + 6 - 7, isto é, 4. A(4) é o quarto elemento na matriz e seu valor é 20. Desse modo, a instrução PRINT A(X + Y - Z) apresentará o número 20 na tela.

então A(X) terá o conteúdo de A(1), que é 4. Neste caso, A(X + 1) corresponderá ao conteúdo de A(2), que é 9, e assim por diante.

Examine o programa e veja se pode perceber o que exatamente está ocorrendo. A linha 20 atribui à variável N o total de números que desejamos ordenar. Suponha que decidimos ordenar cinco números: quando o programa for processado, digitaremos o número 5 e teclaremos RETURN.

A linha 30 apresenta a instrução DIMensão. Se a variável N for 5, ela estabelecerá uma matriz constituída de cinco números. Esta linha equivale a DIM A(5).

As linhas de 40 a 60 consistem em um loop que possibilita a digitação dos cinco números. A maioria das versões de BASIC alerta o usuário apresentando um ponto de interrogação na tela. A tecla RETURN deverá ser pressionada após o fornecimento de cada número. Os números podem ter mais de um dígito, bem como incluir frações decimais.

A linha 90 zera a variável S. Esta variável está sendo usada como "flag" (sinalizador). Mais adiante no programa, a variável S é testada a fim de verificar se corresponde a 1 ou não. Só corresponderá se dois números tiverem sido substituídos um pelo outro, como veremos na sub-rotina de substituição. Vamos examinar em maior detalhe o uso de "flags", mais adiante, no curso.

A linha 100 estabelece os limites de um loop; neste caso, de 1 a 4 (pois N corresponde a 5; assim, N - 1 equivale a 4). Na primeira passagem pelo loop, a variável L assume o valor 1; assim, A(L) na linha 110 corresponderá a A(1), ou seja, o primeiro elemento da matriz, e A(L + 1) corresponderá a A(2), o segundo elemento da matriz. Na próxima passagem pelo loop, o valor da variável L será aumentado para 2; assim, A(L) equivalerá a A(2) e A(L + 1) a A(3). A linha 110 verifica se A(L) é maior que o número imediatamente à sua direita na tabela. O sinal de "maior do que" é >.

Se o primeiro número for maior que o segundo, o programa desviará para uma sub-rotina que permuta números. Se o primeiro número não for maior que o seguinte, não há desvio para a sub-rotina e o programa simplesmente segue para a próxima linha, que é a instrução NEXT L. Após ter sido repetido quatro vezes, o loop se encerra e o programa se dirige à linha 130, que examina a "flag de substituição", S, a fim de verificar se possui ou não o valor 1. Se tiver o valor 1 (colocado na sub-rotina de "substituição"), o programa desvia, voltando à linha 90, para repetir o processo de comparação. Se a "flag" S não cor-

responder a 1, não ocorreu substituição, o que significa que todos os números estão ordenados. O resto do programa apenas realiza a impressão dos números na tela.

A sub-rotina da substituição exige uma variável que armazene temporariamente um dos números que será substituído. Após os dois números terem sido permutados, nas linhas 210, 220 e 230, a "flag de substituição" S assume o valor 1, após o que o programa retorna ao programa principal.

```

10 PRINT "QUANTOS NUMEROS DESEJA
    ORDENAR?"
20 INPUT N
30 DIM A(N)
40 FOR X = 1 TO N
45 PRINT "PROXIMO NUMERO"
50 INPUT A(X)
60 NEXT X
70 REM
80 REM ROTINA ORDENACAO (SORT)
90 LET S = 0
100 FOR L = 1 TO N - 1
110 IF A(L) > A(L + 1) THEN GOSUB 200
120 NEXT L
130 IF S = 1 THEN 90
140 FOR X = 1 TO N
150 PRINT "A("; X;") = ";A(X)
160 NEXT X
170 END
180 REM
190 REM
200 REM SUB-ROTINA SUBSTITUICAO
210 LET T = A(L)
220 LET A(L) = A(L + 1)
230 LET A(L + 1) = T
240 LET S = 1
250 RETURN
    
```

Exercícios

- Ampliar o programa a fim de calcular o valor médio dos números fornecidos. A média é obtida pela soma dos itens dividida pelo número total desses itens. O modo mais simples de fazê-lo consiste em colocar uma instrução GOSUB antes da instrução END, na linha 170. A sub-rotina deverá ler cada um dos elementos incluídos na tabela e somará os valores em uma variável de "soma". Após a adição de todos os elementos, o resultado deverá ser dividido pelo número total desses elementos. A soma é facilmente obtida pelo uso do número de elementos como limite máximo de um loop FOR-NEXT.

- Altere uma linha no programa, de modo que os números sejam dispostos em ordem decrescente.

- Este exercício é dirigido em especial a quem possui computadores que não admitem o uso de variáveis como indexadores. Refaça o programa de modo que a instrução INPUT possa prever uma quantidade determinada de números a serem fornecidos, digamos, 12. Isto eliminará a necessidade de recorrer a uma variável como indexador. As linhas 100 e 110 deverão ser alteradas, assim como a sub-rotina de substituição.

- Um exercício difícil: nosso método para ordenação de números não é o único possível. Experimente elaborar outro procedimento.

A propósito...



Se este programa for processado nos computadores compatíveis com o Sinclair (TK85, CP 200), a linha 130 deverá ser alterada para: 130 IF S = 1 THEN GOTO 90.



Esta instrução não existe no TK82 ou CP 200. A linha 170 deve ser alterada para 170 GOTO 260, acrescentando-se a linha 260 REM FIM DO PROGRAMA.



O grupo dois

Embora forneçam respostas rápidas a problemas complexos, os computadores manipulam dados do modo mais simples possível.

Na última parte do nosso curso sobre o sistema binário, vimos a forma pela qual os computadores fazem a adição. Agora, vamos analisar o processo da multiplicação.

Se você tivesse de multiplicar 14 por 12, uma forma simples seria fazer a adição contínua de $14 + 14 + 14 \dots$ (12 vezes). Já que a multiplicação é, de certa forma, uma soma repetida, esse recurso teria resultado correto, e era assim que os primeiros computadores realizavam a multiplicação. Entretanto, o método é muito demorado, e por isso os projetistas de computador desenvolveram um método eficiente.

Quando multiplicamos dois números, fazemos a operação em um papel desta forma:

$$\begin{array}{r}
 14 \\
 \times 12 \\
 \hline
 28 \\
 + 14 \\
 \hline
 168
 \end{array}$$

(um 0 final é sempre escrito para manter os dígitos na coluna correta)

O mesmo processo funciona em qualquer base de números. Vamos ver um exemplo no sistema binário:

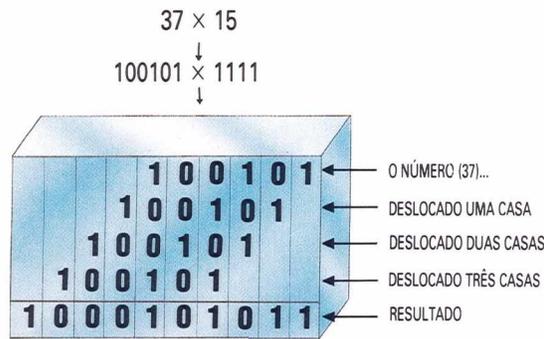
$$\begin{array}{r}
 101 \\
 \times 11 \\
 \hline
 101 \\
 + 101 \\
 \hline
 1111
 \end{array}$$

Com números maiores, o método é exatamente o mesmo; portanto, vamos voltar ao exemplo de 14×12 , utilizando o sistema binário:

$$\begin{array}{r}
 1110 \quad (14) \\
 \times 1100 \quad (12) \\
 \hline
 0000 \\
 0000 \\
 1110 \\
 1110 \\
 \hline
 10101000 \quad (168)
 \end{array}$$

A multiplicação é ainda mais simples em binário do que no sistema decimal, pois não há nunca um dígito transportado. Quando você multiplica um número por 1, o número não é alterado, $14 \times 1 = 14$, e quando você multiplica um número por 0, a resposta é 0, $14 \times 0 = 0$. Isso é verdadeiro em todos os sistemas numéricos.

Analisando esses cálculos, os matemáticos perce-



beram a existência de um padrão simples: a multiplicação binária consiste em apenas duas operações, “deslocamento” e adição. É precisamente esta a forma pela qual o computador faz a multiplicação. Em primeiro lugar, ele desloca “cópias” da linha superior para sua posição correta (determinada pelos algarismos 1 e 0 da linha inferior), e em seguida soma todas as “cópias”.

O computador precisa ter uma enorme capacidade de dígitos para realizar multiplicações. Quando, no exemplo dado, o número de quatro dígitos 1110 foi multiplicado pelo número de quatro dígitos 1100, a resposta obtida foi de oito dígitos (10101000), e, em geral, o resultado de uma multiplicação pode ter até duas vezes o comprimento do número maior.

Surpreende o fato de uma multiplicação feita por computador apresentar resultado incorreto, mas isso pode ocorrer. O erro pode ser explicado pela extensão de espaço designada pelo projetista da máquina para fornecer a resposta. Caso tenha sido alocado espaço insuficiente, ocorrerá um “overflow” (estouro): os dígitos menos significativos não aparecerão e o resultado estará errado.

Deslocamento na multiplicação
 A multiplicação binária é muito mais fácil do que a decimal. O mesmo processo de multiplicação longa utilizado no sistema decimal é aplicado; porém, como são envolvidos apenas dois números na multiplicação (0 e 1), a operação é muito mais simples. Quando um número é multiplicado por 1, o resultado é obviamente o mesmo número. Na ilustração, quando 100101 (37) é multiplicado por 1111 (15), quatro cópias do 100101 aparecem. Cada cópia é deslocada à esquerda, de acordo com a posição do 1, que a multiplica. Finalmente, todas as cópias são somadas, e é dada a resposta 1000101011 (555).



Gottfried Wilhelm Leibniz (1646–1716), contemporâneo de Isaac Newton, deu grandes contribuições à matemática e à filosofia. Inventou uma máquina de multiplicar e dividir. Investigou as possibilidades da aritmética binária em cálculos, embora a primeira referência aos números binários tenha sido feita por Francis Bacon, em 1623. Nos últimos anos de vida, entrou em disputa com Newton sobre a invenção do cálculo infinitesimal.

Sir Clive Sinclair



O empresário e gênio da eletrônica que, com suas invenções, revolucionou o mercado dos microcomputadores.

Um americano ou um europeu, mesmo com boa cultura, não saberiam responder o que é ou como funciona um Spectrum ou um ZX81. Aliás, é provável que se espantassem com semelhante pergunta, se alguém a fizesse. No entanto, o caso mudaria inteiramente de figura se a indagação se referisse à identidade de Sir Clive Sinclair, o gênio milionário da eletrônica (que já foi muitas vezes notícia de televisão, revistas e jornais).

Nascido no ano de 1940, em Londres, esse inglês irrequeto e criativo sempre demonstrou um interesse apaixonado por máquinas e miniaturas, desde a infância. Como resultado, já aos 12 anos de idade havia construído, sozinho, uma pequena calculadora mecânica. É fácil imaginar que para um garoto desses a escola convencional não ofereceria muitos atrativos. Apesar disso, Clive frequentou o St. George's College, de Weybridge, onde concluiu seus estudos de grau médio. Tinha então 17 anos e, em vez de ingressar numa faculdade, preferiu editar e escrever revistas técnicas, especializadas em eletrônica doméstica. Nessa ocupação, continuou a pesquisar e inventar até que, em 1962, fundou sua primeira empresa, a Sinclair Radionics. Seus primeiros produtos — vendidos por reembolso postal — foram kits de rádios portáteis para montar e amplificadores.

Em 1972, a Sinclair Radionics lançou no mercado um dos primeiros modelos de calculadora eletrônica de bolso, que também tinham preço acessível. (Um desses modelos, porém, era folheado a ouro e cus-

tava 4.000 dólares!) Os recentes relógios de pulso digitais, que utilizam microchips, também foram produzidos por esse Henry Ford do mundo do computador.

Mas foi somente no início da década de 80 que Clive Sinclair produziu seu primeiro computador de uso pessoal, o ZX80, uma verdadeira revolução no campo dos microcomputadores. Além de ser o mais barato dos microcomputadores então no mercado, o ZX80 usava a metade dos chips normalmente utilizada nesse tipo de computador e possuía um BASIC simples, mas muito potente. O modelo vendeu 50.000 unidades em poucos meses. O modelo seguinte, entregue ao mercado em 1981, surpreendeu ainda mais. Era o ZX81, menor e mais barato do que o ZX80, mas de eletrônica mais avançada (usava apenas quatro chips!). O ZX81 teve uma venda superior à do antecessor e, além de assegurar a seu criador fama e fortuna definitivas, colocou a Inglaterra outra vez na corrida tecnológica, juntamente com o Japão e os Estados Unidos. Mas isso ainda não era tudo: em 1982, Sinclair lançava o Spectrum, o primeiro micro de sua categoria com condições de produzir cor.

Em meio a tantas invenções, Sinclair não perdeu o gosto pela redação e edição de livros: sua editora, a Sinclair Brown, fundada em 1981, publica cerca de vinte títulos anuais, entre ficção e não ficção.

As mais recentes inovações de Sinclair são um televisor com minúscula tela plana do tamanho de um livro e um carro elétrico para uso na cidade.

1962

Clive Sinclair funda a Sinclair Radionics em Londres, para vender kits de rádio e amplificadores por reembolso postal.

1972

Sinclair fabrica uma das primeiras calculadoras de bolso do mundo, a Executive, a preço baixo, e lucra 3,7 milhões de dólares com a exportação.

1975

Sinclair lança um dos primeiros relógios digitais, o Black Watch. Entretanto, a empresa teve problemas financeiros, devido a dificuldades com o fornecimento de chips.

1976

O Conselho Nacional de Empresas (National Enterprise Board) forneceu-lhe subsídios para estudos de sua televisão de bolso, que será lançada brevemente.

1979

Sinclair funda a Sinclair Research, para desenvolver produtos de consumo no campo da eletrônica.

1980

A nova empresa lança seu primeiro produto, o ZX80, conhecido como o primeiro computador a ser vendido por menos de 150 dólares.

1981

Sinclair desenvolve o computador ZX81, que vende mais de 1 milhão de unidades em dois anos.

1982

O Spectrum é introduzido, para ser vendido juntamente com o ZX81, porém projetado para um conjunto maior de usos.

1983

Lança-se o tão esperado Microdrive, com as interfaces I e II, que expandem o Spectrum, utilizando cartuchos ROM. Sinclair também anuncia o novo televisor portátil de tela plana, após um programa de desenvolvimento de quatro anos, ao custo de 6 milhões de dólares.

1984

Sinclair espera lançar o computador ZX84 no mercado. Há rumores de que esse computador terá como característica a tela plana.



Microeletrônica

Cientistas descobriram que o desenvolvimento do chip dependia de um dos mais abundantes recursos naturais da Terra.

O silício é encontrado em sua forma natural por toda a superfície da Terra, em maior quantidade que qualquer outro elemento, exceto o oxigênio, com o qual se combina na formação da sílica (dióxido de silício). Muitas pessoas passam a maior parte de suas férias deitadas sobre sílica, construindo castelos com ela. Toda a revolução da microeletrônica acabou sendo erguida sobre areia!

A importância do silício na indústria microeletrônica reside em sua estrutura física. Em estado puro, o silício é péssimo condutor de eletricidade. Entretanto, quando quantidades controladas de determinadas impurezas são nele incorporadas, o silício torna-se semicondutor.

A condução de eletricidade através de uma substância é determinada pelo número de elétrons em cada um de seus átomos e pelo modo como estão ligados. Nos metais, a corrente elétrica é transportada pelos elétrons, sem ligações firmes: aí, os elétrons estão livres para transitar pelo interior da estrutura

atômica, transferindo sua ligação, com sua carga elétrica, de um átomo a outro. Em um material isolante, todos os elétrons estão firmemente ligados; assim, a corrente não pode transferir-se de um ponto a outro.

A fabricação do silício puro é um processo simples. Primeiramente, o óxido bruto é refinado por processos químicos, até tornar-se 99,99% puro. Em seguida, é colocado em um cadinho e aquecido até o ponto de fusão, a 1.410°C, em um ambiente de gás inerte purificado.

O processo de incorporação de quantidades controladas de impurezas (chamado “dopagem”) exige que o silício puro seja combinado com fósforo, o que produz silício “tipo n” (assim chamado por transportar a carga negativa), ou boro, que forma o silício “tipo p”, ou silício carregado positivamente.

Um cristal de grandes dimensões é produzido pela introdução de um cristal “semente” perfeito no óxido fundido e, depois, retirando-o lentamente,



O forno de chips

Vê-se aqui um forno de chips de silício. Após ter sido refinado, cortado em fatias, polido, mascarado e gravado, o silício recebe um banho de dióxido de silício. Isso é realizado pelo aquecimento das plaquetas a 1.050°C e subsequente passagem de uma corrente de oxigênio puro ou de vapor superaquecido sobre elas. Enquanto estão no forno, dentro de um recipiente feito de quartzo fundido, forma-se uma camada de dióxido de silício na superfície da plaqueta.

Essa camada é então seletivamente removida pelo próximo processo de gravação, sendo o ciclo repetido para cada camada do chip.



A soma das partes

O processo de fabricação do circuito integrado exige que cada camada do circuito seja colocada separadamente na superfície da plaqueta de silício — uma técnica muito semelhante ao modo pelo qual as ilustrações em cores desta publicação são impressas. Cada foto é primeiramente separada em três cores, além do preto. Em seguida, as quatro cores são impressas uma sobre a outra, em correspondência muito precisa, a fim de alcançar o efeito que é observado nestas páginas.

O processo de produção do chip recorre ao uso de camadas de silício e de outros materiais, no lugar da tinta, mas o processo de impressão é muito mais simples do que o de banho/mascaramento gravação. Como se observa na foto, as máscaras individuais são combinadas em seqüência, de modo a construir um microcircuito; neste caso, um componente transistorizado muito simples.



sempre em movimentos circulares. É desse modo que cristais de 7 a 10 cm de diâmetro e 60 cm, ou mais, de comprimento são produzidos. Em seguida, esses cristais são lapidados, até atingir um diâmetro padronizado, de 76 mm ou 100 mm. O cristal é montado, cortado em fatias e lapidado para ficar liso de ambos os lados, antes de ser polido em apenas uma das faces. A “pastilha” resultante tem geralmente 0,5 mm de espessura.

Se o processo é relativamente simples e a matéria-prima tão abundante, por que é tão caro o silício usado nos chips?

A resposta está na necessidade absoluta de manutenção da pureza. São necessários cuidados excepcionais para evitar impurezas. Os níveis de pureza do ar nas fábricas de plaquetas (ou pastilhas) são realmente extraordinários — menos de 3.000 partículas por metro cúbico. Significa mais de cem vezes a pureza do ar em hospitais modernos.

A fabricação do circuito integrado exige um método de microgravação da superfície dos chips. Na produção em massa, isso é conseguido por um processo chamado fotolitografia, semelhante, em muitos aspectos, ao usado para fazer esta publicação.

Cada “camada” do circuito é tratada como uma unidade isolada, durante todo o processo. O desenho original é produzido por métodos computadorizados e convertido em uma fotografia, que é então reduzida ao tamanho real. A máscara é formada pela reprodução dessa fotografia muitas vezes em uma configuração quadriculada, que cobre toda a superfície da plaqueta.

A plaqueta é primeiramente aquecida a 1.050°C,

em oxigênio puro. Isso faz com que se forme uma camada de dióxido de silício, na superfície, que age como isolante. Essa camada é então removida seletivamente, formando “janelas” no silício puro, de baixo. Esse processo repete-se em cada estágio sucessivo da construção do circuito integrado, na superfície da base de silício.

A superfície oxidada da plaqueta é primeiramente banhada com um material sensível à luz, cuja solubilidade é acentuadamente diminuída pela exposição a raios ultravioleta. Uma máscara na forma da primeira camada do microcircuito é introduzida entre a superfície banhada e a fonte luminosa. A superfície é exposta a raios ultravioleta e então “revelada” em um solvente que remove o material sensível à luz da região não exposta aos raios ultravioleta.

Esse processo é análogo ao usado na gravação em cobre: a superfície da folha de cobre é banhada em cera, o desenho é riscado através da camada de cera e o conjunto é imerso em ácido. Ao ser removido o ácido, o desenho permanece gravado na superfície do cobre. Nos locais onde a camada de cera estava intacta não ocorre nenhuma reação.

Terminado o primeiro estágio, todo o processo é repetido com diferentes máscaras e reagentes químicos, até que os circuitos desejados tenham sido construídos. A plaqueta inteira é finalmente banhada de modo uniforme, com uma última camada de dióxido de silício.

Este processo pode exigir dez ou mais repetições do procedimento de banho/mascaramento/gravação. A margem de erro é grande em cada estágio e o índice de defeitos, no estágio de teste e controle de



qualidade, é, na verdade, muito alto. Quanto mais complexo o microcircuito e mais compactamente comprimida a plaqueta, tanto maior o índice de perda.

O primeiro estágio do teste exige que a plaqueta inteira, com suas muitas centenas de circuitos integrados idênticos, seja encaixada em um dispositivo controlado pelo computador, que verifica cada circuito.

Geralmente, na plaqueta são reservadas nove posições, do tamanho de um chip, a fim de tornar possível o teste.

O aparelho de teste não apenas assinala cada unidade rejeitada com um ponto de tinta, mas também extrai dados sobre índices de perdas em cada plaqueta, localiza as falhas, tanto na plaqueta como em cada chip isolado, e identifica defeitos específicos.

As plaquetas são então cortadas em chips individuais, e as unidades rejeitadas são extraídas manualmente. Montam-se os chips que restarem em lâminas de metal miniaturizadas.

Os pinos conectores do chip são ligados à lâmina com fios muito finos — novamente em um processo controlado por computador — e o conjunto é selado em um suporte de cerâmica ou de plástico, do qual saem pinos conectores. Seguem-se testes finais meticulosos.

Atualmente, o limite no grau de miniaturização, que se situa no estágio fotolitográfico de fabricação, é o comprimento da onda luminosa — cerca de 2 milionésimos de metro. Recentes pesquisas centralizam-se no uso de raios X, que permitirão uma redução do circuito microeletrônico de, talvez, cinquenta vezes.

Quase todos os aspectos da fabricação dos chips, desde o estágio de elaboração do desenho, passando pela padronização, até a produção em massa e os testes finais, constituem um processo tão complexo que seriam inviáveis sem o uso dos próprios dispositivos que estão em fase de produção — um paradoxo notável!

A utilização de computadores no processo de elaboração do projeto, por exemplo, permite que subseções sejam predefinidas e chamadas da memória, sempre que necessário.

Vejamos o caso de um chip de memória de acesso aleatório (RAM), em que cada um dos bits é mantido em uma célula de armazenamento de um único transistor. Esse chip precisa conter um conjunto de 16.384 células idênticas, para armazenar 2 Kbytes de dados. Neste caso, o projetista define a estrutura uma única vez, e dará instruções ao computador para que ela seja repetida 16.384 vezes.

O uso de computadores no departamento de projetos não está limitado a este recurso útil e simples. Os desenhistas agora trabalham, não em pranchas de desenho, mas em terminais de apresentação visual, e utilizam canetas ópticas para “desenhar” diretamente na tela do monitor. O desenho final é preparado pelo computador, que utiliza para isso um plotter em cores.

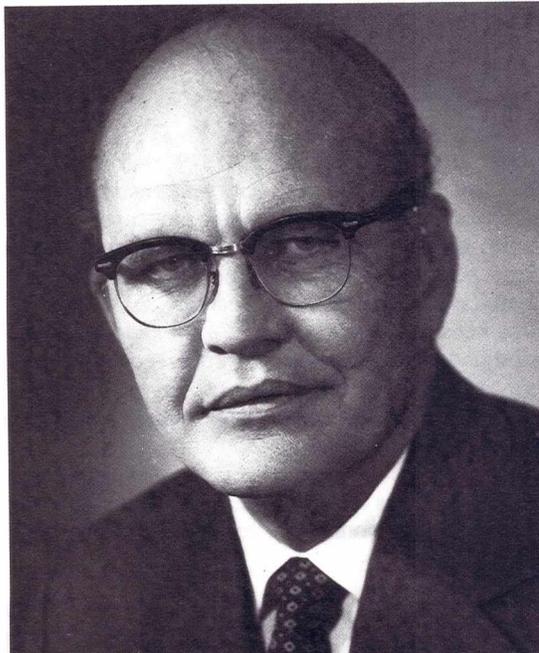
O processo de elaboração gráfica não foi o único a ser aperfeiçoado pela mecanização. Na página 103 aludimos à criação do modelo computadorizado de um campo de produção de petróleo. Essa mesma técnica de criação de modelos pode ser utilizada na

elaboração do projeto do circuito, permitindo também ao desenhista experimentar uma variedade de soluções, para não se comprometer com um processo de fabricação dispendioso. Desse modo, evita-se grande parte dos gastos causados por técnicas de tentativa e erro.

A “revolução microeletrônica” significou saltos quantitativos na velocidade e na redução de tamanho e preço. A esta altura, é interessante lembrar as características do computador de válvulas:

- Grande tamanho físico
- Baixa velocidade de processamento
- Exigência de alta capacidade de energia
- Limitação de memória e de quantidade de instruções
- Alto custo

Todos estes fatores foram afetados pela descoberta do transistor (ver p. 46); porém, no processo de produção, a indústria do computador ainda consumia intenso trabalho e, conseqüentemente, era dispendiosa. Os componentes isolados tinham, ainda, de ser montados em painéis de circuitos impressos.



Um revolucionário da microeletrônica

A descoberta do circuito integrado é, de modo geral, atribuída a Jack Kilby, enquanto trabalhava na Texas Instruments, em 1958. Ele construiu um bloco de cerca de 12,7 mm por 6,35 mm, que continha certo número de transistores. Os modernos circuitos eletrônicos contêm centenas de milhares de componentes ocupando o mesmo espaço.

A descoberta do circuito integrado — e, mais especificamente, do microprocessador — permitiu à indústria se beneficiar de progressos de fabricação controlados por computador.

Algumas estatísticas são surpreendentes. Em 1959, apenas um componente podia ser colocado em um circuito — um diodo, por exemplo, ou um transistor. Em 1978, o mais denso circuito integrado de grande escala (LSI) possuía mais de 250.000 componentes em um único chip. Em menor período, de 1973 a 1983, o custo por bit de memória do computador diminuiu em cerca de vinte vezes e o uso de componentes eletrônicos de todos os tipos aumentou, no mundo todo, aproximadamente mil vezes. Essas tendências continuarão: calcula-se que o número de componentes eletrônicos usados por ano aumentará em cem vezes nos próximos três anos.

Siga as pistas

O computador pode selecionar fatos e elaborar listas a partir das numerosas informações armazenadas em um banco de dados.

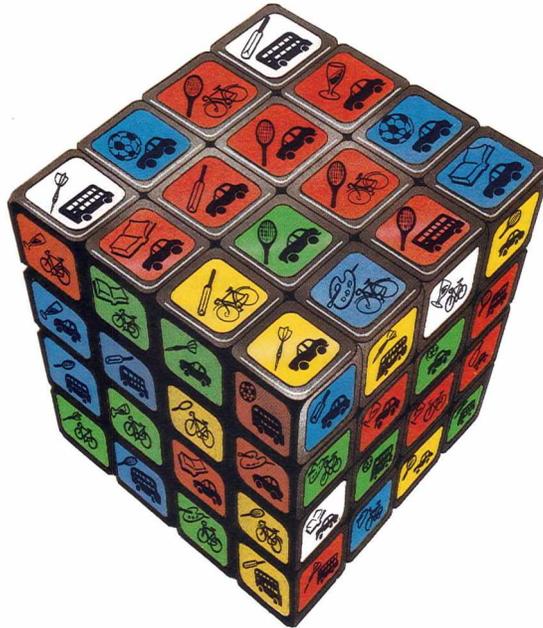
Um conjunto de dados mantidos e consultados pelo computador é conhecido pelo nome de banco de dados. Todos nós usamos vários bancos de dados não computadorizados em nosso dia-a-dia.

A lista telefônica é um exemplo de banco de dados não computadorizado. Entretanto, não é necessário catalogar ou armazenar dados em uma determinada ordem para termos um banco de dados. Em um computador, essa ordenação rígida dá origem a sérias limitações.

Programas de banco de dados são conjuntos de rotinas que permitem fazer seleções dos dados. Tais programas variam desde simples sistemas de fichário até linguagens completas. Geralmente, um banco de dados computadorizado é extenso e contém dados de muitos tipos. Mas isso não significa que você precisa ter um equipamento enorme. Qualquer computador pode controlar um eficiente banco de dados. As únicas limitações sérias estão no tamanho e velocidade dos recursos de armazenamento.

Poderíamos, por exemplo, fazer uma lista com dados sobre várias pessoas. Colocando essas informações em um fichário comum, teremos um conjunto semelhante a um fichário pessoal. Verificamos logo que, nesse caso, há muitas espécies de dados. Em cada categoria, muitos itens são palavras e alguns são números. Há possibilidades limitadas em algumas categorias: "sexo" (masculino e feminino) e "estado civil", que será um dos elementos do conjunto "solteiro, casado, divorciado, desquitado, viúvo".

Pode-se transformar certos itens em listas de palavras e números. Por exemplo: profissão, nome da



empresa, endereço comercial, telefone comercial e nome do superior hierárquico serão convenientemente agrupados sob o título "vida profissional", e modelo e ano do carro ficarão incluídos na lista "carro".

Ampliando essa idéia, podemos conservar todos os endereços em forma de listas. É melhor do que mantê-los em um único item, pois talvez desejemos saber apenas em que cidade uma pessoa reside, e não em que rua.

É possível também ampliar o item "estado civil" pelo acréscimo do nome do cônjuge, quando for necessário. O nome pode ser uma só palavra, mas,

Alguém se habilita?

Usamos o cubo mágico como uma analogia para um banco de dados rudimentar — isto é, contém todos os dados de que precisamos, mas ainda não colocados na ordem certa. Neste exemplo, estamos procurando um parceiro de tênis (o símbolo da raquete) que possua carro (o carro) e que esteja disponível no dia em questão (os quadrados vermelhos).

```

DATABASE HCCSOFT
ENTER DATE AS DD/MM/YY: 14/10/83
.ACCESS
*use FRIENDS
*while FRIENDS, do CARS:
*while HOBBIES, do TENNIS;
*end
.CLOSE
  
```

```

IFRIENDS, CARS subset hobbies: TENNIS
FIONA DAVIES      253 8146
CYNTHIA GRINSHAW 704 1121
ROBERT MITCHELL  877 6202
DAVE WHELAN      582 5368
MARGRET JENKINS  345 1314
JEND
  
```

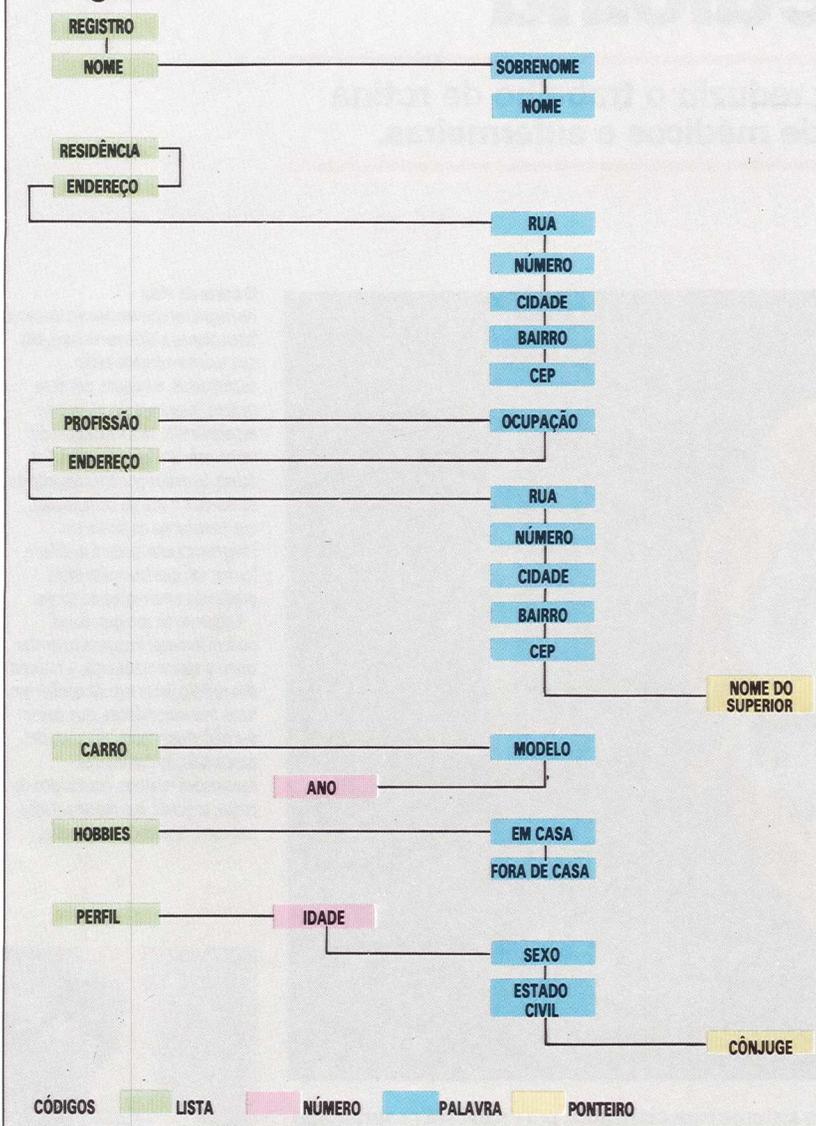
Fazendo as perguntas certas

Muitos programas de banco de dados comerciais usam códigos semelhantes a linguagens de programação. No exemplo acima o termo "ACCESS" indica ao programa que queremos consultar um

arquivo previamente criado. O termo "USE" indica que vamos usar um subconjunto do arquivo "FRIENDS" e, "enquanto" ("while") essas condições forem mantidas, poderemos obter todos os itens que estão sob os títulos "CAR" ou "TENNIS". O resultado é uma lista de amigos, que dirigem carro e jogam tênis.

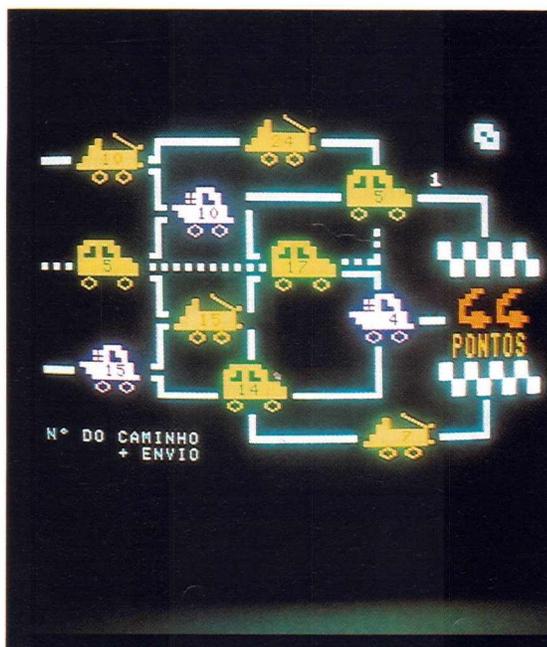


Registro

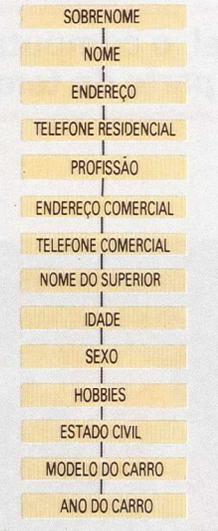


A enciclopédia de hoje

Serviços como o videotexto são tentativas para tornar grandes bancos de dados acessíveis ao público em geral, através de serviços que cada um pode ter em sua própria casa. Esse tipo de sistema usa a televisão comum como tela de monitor, e um teclado ligado ao computador central por uma linha telefônica comum. O acesso ao banco de dados faz-se por meio de um "menu" de serviços disponíveis. As opções são apresentadas na tela, e o usuário chega até a "página" de dados desejada percorrendo uma determinada hierarquia. Dentro de pouco tempo, o fornecimento do código de seu cartão de crédito lhe possibilitará fazer compras sem sair de casa.



Fichário pessoal



Informações organizadas

Um banco de dados organizado em hierarquia permite ao usuário passar de uma parte da informação a outra, tornando possível novas escolhas a cada vez. Nenhum conhecimento do conteúdo está aqui presumido.

como se refere a uma pessoa e o arquivo é sobre pessoas, seria conveniente que pudéssemos recorrer a outro registro.

Por estar em um lugar determinado do arquivo, cada registro tem um número. Assim, podemos usar o número do registro que descreve a pessoa, em vez de seu nome, para estabelecer uma referência cruzada.

Tal item de dados é chamado "ponteiro". Se usarmos essa técnica para referências ao superior da pessoa na empresa, o resultado será uma estrutura semelhante ao que se chama "registro".

A diferença entre um fichário e um banco de dados computadorizado está no fato de que o primeiro pode ter apenas uma ordem, geralmente alfabética.

O fichário será adequado se quisermos descobrir, por exemplo, qual empresa mantém como empregado uma determinada pessoa. Mas que faremos se quisermos saber quais os nomes de todos os empregados de uma empresa?

Pelo uso de um fichário, teríamos de examinar todos os cartões, retirando os que possuam o dado procurado. Esse procedimento não apenas consome muito tempo, mas também pode resultar em erros.

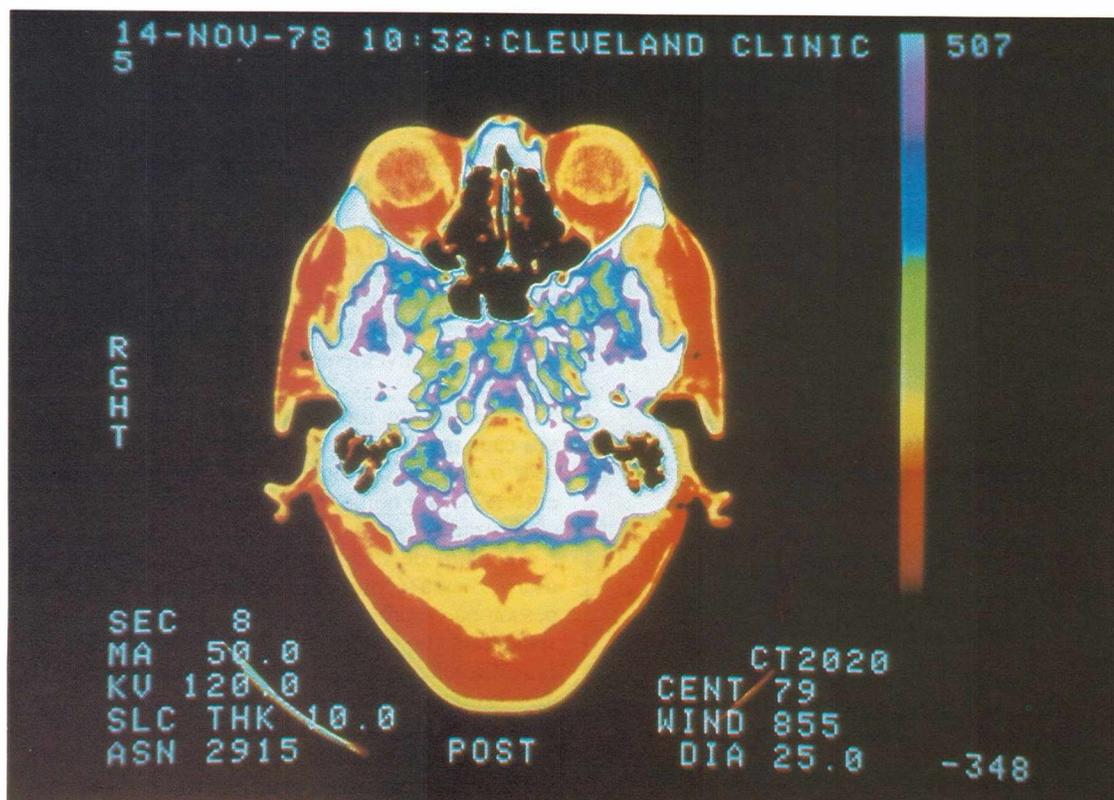
Com um sistema computadorizado, entretanto, podemos fazer com que a máquina verifique cada registro sucessivamente e imprima o nome de todas as pessoas que trabalham na empresa.

Outra possibilidade seria fazer o computador reorganizar o arquivo, colocando como item mais importante o campo de atividade da empresa. Isso resultará no mesmo banco de dados, com os mesmos dados, mas com uma estrutura completamente diferente. A reorganização colocará todas as ocorrências de determinada empresa em um grupo, e esta seção nos dará os nomes de todos os empregados. Em um fichário, há apenas um item básico para ordenação em geral: o nome, por exemplo. Porém, em um sistema computadorizado, a ordenação pode ser baseada em qualquer item.

Desse modo, pode-se acessar uma determinada informação do banco de dados com muita facilidade e rapidez.

Micros na medicina

Auxiliar inestimável, o computador reduziu o trabalho de rotina que consome boa parte do tempo de médicos e enfermeiras.



Assim como em outras profissões em que pessoal altamente especializado — e bastante dispendioso — precisava utilizar grande parte de sua capacidade em tarefas rotineiras, também a medicina foi consideravelmente favorecida por equipamentos econômicos e adaptáveis, como o microcomputador.

Com a introdução generalizada de equipamentos operados por microprocessadores, as unidades de terapia intensiva, em especial, modificaram de modo substancial seus procedimentos de trabalho. Em pouco tempo, os microcomputadores passaram a ser usados em monitores de pulsação, respiração e pressão sanguínea, por exemplo, fornecendo leituras instantâneas de estados patológicos, liberando, assim, grande parte da equipe de enfermeiras para tarefas menos rotineiras.

Progressos recentes aumentaram a confiabilidade na utilização dos micros na clínica geral e na administração hospitalar, para atualização do histórico dos pacientes, registro de consultas e controle do estoque farmacêutico.

Um sistema simples, mas eficiente, já utilizado em alguns países, é o Mickie, que fornece aos médicos uma visão geral do estado físico do paciente. Entretanto, ainda não pode ser considerado exatamente

um sistema especializado, pois não está programado para fornecer diagnósticos completos, mas apenas dados muito gerais.

As perguntas que o equipamento faz ao paciente são sempre formuladas de maneira que as respostas sejam fornecidas por expressões simples, como “sim”, “não”, “não sei”, “não compreendo”. Se a resposta for “não compreendo”, o sistema fará ainda algumas tentativas de auxiliar o paciente a responder. Em vez de um teclado completo, o paciente responde por meio de uma caixa com apenas quatro teclas, convenientemente identificadas. Durante o processamento, o sistema opera de modo lento, porém essa é uma limitação artificial, decorrente da velocidade média da leitura humana. O próximo passo no desenvolvimento de um sistema especializado para clínica geral será provavelmente a vinculação dos dados novos, obtidos através desse processo, ao histórico do paciente. Caso se verifique que o paciente se queixou dos mesmos sintomas em consulta anterior, o médico necessitará fazer apenas uma pergunta, para realizar o diagnóstico: “Você está se sentindo do mesmo modo que em...?” Entre outras vantagens que os registros médicos computadorizados proporcionam, estão algumas que talvez

O corte da vida

Radiografias convencionais fornecem fotos planas e bidimensionais, em que todos os órgãos estão superpostos, e exigem, por esse motivo, interpretação altamente especializada. Já a sondagem do corpo com um feixe de raios X, a coleta de dados por um conjunto de sensores e o uso do computador, que transforma os sinais em imagens na tela, podem auxiliar a formar um quadro muito mais preciso de uma região do corpo.

Enquanto os computadores podem fornecer imagens coloridas, como a apresentada aqui, a maioria dos radiologistas tem de confiar em fotos monocromáticas, que devem ser cuidadosamente interpretadas para avaliação correta das densidades relativas dos tecidos do corpo, segundo sua representação nos sombreados da radiografia.



Um olhar atento

O sistema de monitores, além de fornecer às enfermeiras indicações sobre o estado do paciente, armazena dados para análise posterior. Nesta foto, um médico extrai dados muito precisos sobre o estado do paciente durante a noite. Isso o auxilia a fazer um diagnóstico mais correto.



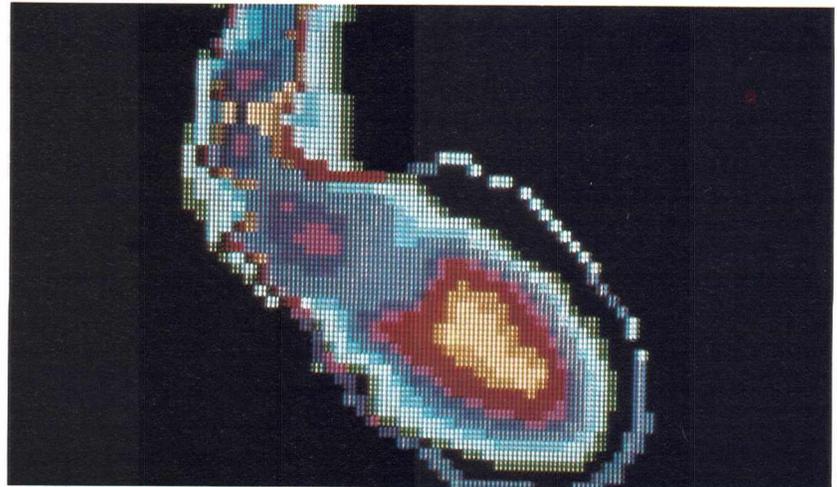
não sejam tão aparentes — o caráter confidencial das informações, por exemplo. É, sem dúvida, mais difícil ler um arquivo armazenado em um disco do que em um formulário de papel — mesmo que se conheça o modo de operação do equipamento.

Entretanto, os progressos mais significativos ocorreram na área dos diagnósticos. Até recentemente, os únicos métodos disponíveis para exame interno eram, na melhor das hipóteses, perigosos e, na pior, positivamente nocivos: radiografias, com a possibilidade de exposição excessiva à radiação; endoscopia — inserção de uma sonda —, que pode ocasionar traumatismos aos frágeis tecidos internos; e cirurgia exploratória.

O desenvolvimento de técnicas tomográficas computadorizadas, que utilizam um feixe fino de raios X, permitiu evitar a exposição do corpo a um manto de radiação, com um significativo avanço em precisão e segurança. O uso de procedimentos alternativos, tais como ultra-sons e ressonância magnético-nuclear, assegura um meio completamente sem riscos de realizar exames internos.

Equipamentos de emergência médica também se beneficiaram com a computadorização. É pouco provável que os transplantes de órgãos se encontrassem no nível de desenvolvimento atual sem os recursos proporcionados pelos bancos de dados.

Outro avanço extraordinário está no tratamento de deficientes físicos e mentais. Apesar de ainda no início, os potenciais progressos nesse campo são enormes. Imagine, por exemplo, um paciente mudo que se vê capaz, pela primeira vez, de “falar” através de um dispositivo de entrada, que acompanha o movimento ocular, e de um sintetizador de voz.



Muito mais segurança

A prospecção feita por ressonância magnético-nuclear, como a mostrada na foto, proporciona resultados análogos aos das sondas tomográficas computadorizadas, porém consideravelmente mais seguros por não utilizar raios X. Atualmente, a técnica ainda está em fase experimental. Cada exame pode durar até uma hora.



A imagem da saúde

Dados sobre o estado do paciente, obtidos por sensores ligados ao corpo, são interpretados por um microprocessador e então exibidos em um osciloscópio especialmente adaptado, semelhante aos usados em engenharia eletrônica.

mento, este poderá ser aproveitado como base para um sistema computadorizado de comunicação. O procedimento mais usual consiste na substituição do teclado comum por uma unidade construída sob medida. Ela pode possuir teclas grandes, por exemplo, para atender às dificuldades de pacientes com esclerose múltipla ou falta de coordenação dos movimentos do corpo. Uma adaptação possível, no caso, é a construção de teclados que não precisem ser pressionados.

Outra possibilidade de uso do equipamento é a de apresentar ao enfermo uma variedade de sugestões, como, por exemplo, o cardápio para sua próxima refeição. O usuário poderá informar ao computador o que deseja na refeição pelo simples pressionar de uma tecla. Caso não haja resposta, o computador presumirá uma resposta negativa. Nesse caso, o teclado funcionaria como um interruptor que poderia ser acionado por um simples movimento de cabeça.

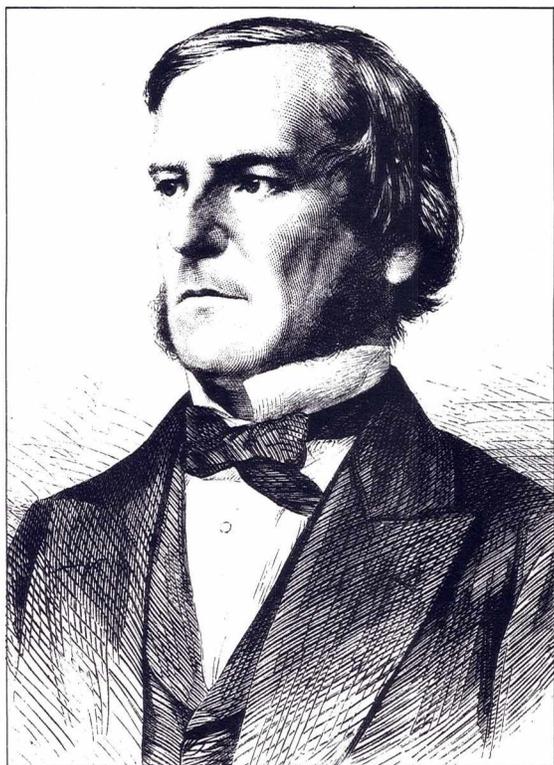
Esse engenhoso recurso permite ao usuário utilizar um pacote processador de palavras, que, embora lento, pode produzir trabalho escrito — um enorme progresso para vítimas da talidomida e pacientes com paralisia cerebral ou espasmos musculares acentuados.

Tais dispositivos medem o movimento do globo ocular à medida que este percorre uma linha. A partir de um ponto referencial, é possível inferir a posição em determinado momento e, daí, o caractere que está sendo lido. Chips sintetizadores de voz contêm, na memória ROM, os elementos básicos da fala, isto é, os fonemas. Sob o controle do programa, os fonemas isolados são encadeados, para formação de palavras.

Se um deficiente tiver um grau mínimo de movi-

Leis do pensamento

Um século antes da invenção dos modernos equipamentos eletrônicos, Boole expunha suas idéias sobre lógica matemática.



Em 1815, ano da derrota de Napoleão na batalha de Waterloo, outro acontecimento importante teve lugar na História da humanidade.

Nesse ano nasceu George Boole, em Lincoln, Inglaterra. Filho de um sapateiro, estava destinado a tornar-se um dos gênios que possibilitaram, mais tarde, a invenção do computador. Morreu em 1864, um século antes da revolução do microcomputador; entretanto, sem suas idéias o moderno computador não teria as características que tem hoje.

Boole sabia que os nossos processos de raciocínio na vida cotidiana podem ser representados em termos de lógica formal, desenvolvida primeiramente pelos gregos. Ele acreditava que, a rigor, poderíamos mesmo chegar a exprimir o raciocínio humano em termos matemáticos. Boole dispôs-se a fazer exatamente isso; estudou matemática sozinho e iniciou sua pesquisa sobre a lógica dos processos de pensamento.

Conjuntos de informação

Suponha que uma determinada pessoa vai a uma festa. Ela quer dançar e, na sala de dança, procura um parceiro.

Há pessoas ali que estão dançando ou não — porque não podem estar fazendo ambas as coisas ao

mesmo tempo. O par procurado será uma mulher — ou um homem.

Boole examinou essas alternativas de outro modo. Para ele, o salão contém “conjuntos” de pessoas, o conjunto de homens e o conjunto de mulheres, que podem ser simbolizados por H e M. Boole também levou em conta o conjunto das pessoas que estão dançando (D) e o conjunto das que estão esperando (E) para dançar.

O parceiro de dança (de um homem) teria de satisfazer duas condições: ser mulher e também estar esperando para dançar. Boole percebeu a importância do conetivo e (“and”) que liga as duas condições e atribuiu-lhe um símbolo: a letra U invertida. Com isto, tornou-se possível simbolizar o conjunto de eventuais pares como sendo $M \cap E$.

Entretanto, se não quisesse dançar, mas apenas conversar com uma pessoa amiga, o homem poderia escolher uma pessoa dos conjuntos H ou M, porque estes dois incluem qualquer pessoa na sala. Aqui, novamente, Boole notou a importância do conetivo ou (“or”) e atribuiu-lhe o símbolo U. Assim, em sua lógica algébrica, H U M inclui todos os homens e todas as mulheres presentes na sala.

As portas lógicas encontradas no computador são representadas de acordo com símbolos criados por Boole, tais como AND e OR. Na seção de programação em linguagem BASIC, examinaremos as funções desses dois úteis conetivos AND e OR. Porém, antes de prosseguir, há uma interpretação interessante da lógica booleana — feita pelos matemáticos John Venn (1834-1923) e Charles Dodgson (1832-1892), mais conhecido como o escritor Lewis Carroll.

Examinemos um problema prático. Vamos supor que você tenha armazenado uma lista de pessoas conhecidas na memória do computador. Junto a cada nome estão relacionados outros dados, como nú-

George Boole, 1815-1864

George Boole nasceu em uma época em que não era possível imaginar os computadores eletrônicos; ainda assim, ele é um dos fundadores da lógica matemática usada nos computadores de hoje. De família modesta, estudou matemática sozinho, nas horas vagas. Estava convicto de que os processos de pensamento de que nos valemos cotidianamente estão fundados na razão e que esta poderia ser depurada até alcançar a forma da lógica matemática. Publicou suas idéias em 1847 e tornou-se famoso da noite para o dia, sendo convidado para primeiro professor de matemática da nova Universidade de Cork, na Irlanda.

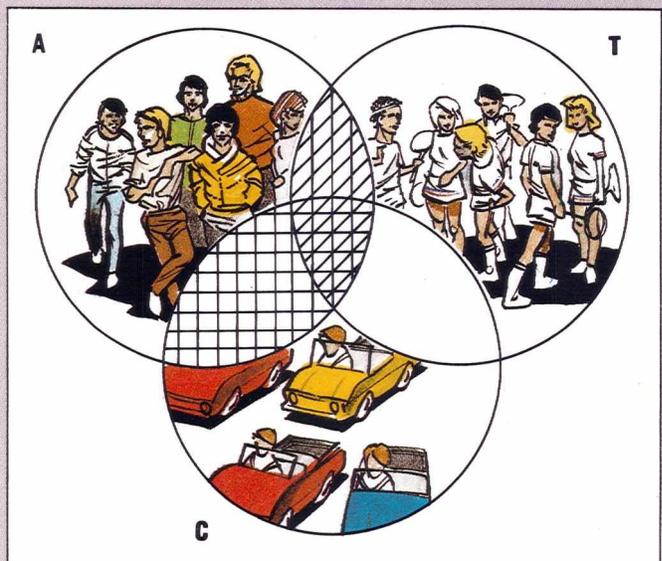
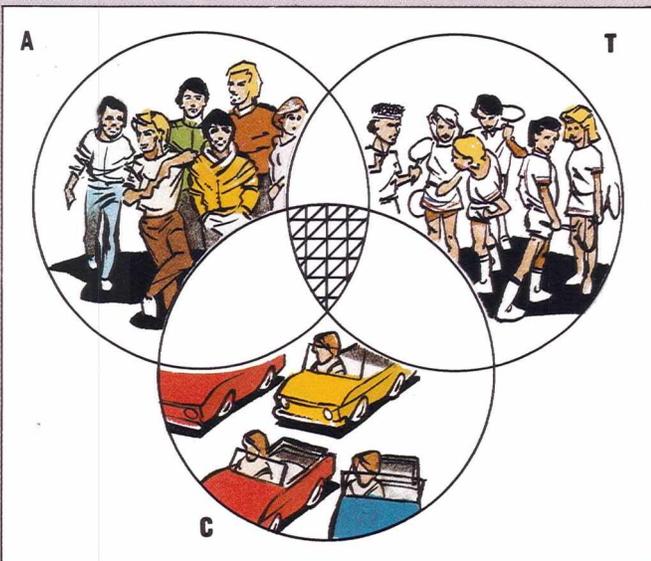
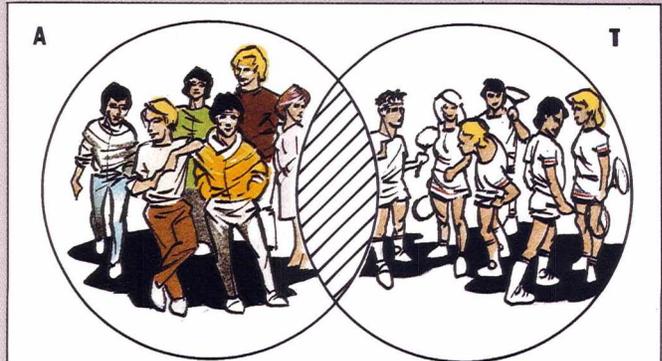
O programa Amigos

```

10 DIM NS(10), DS(10), AS(10), TS(10), CS(10)
15 REM NOME, NO. TEL., AMIGO?, TENIS?, CARRO?
17 PRINT "FORNECER DETALHES NO FORMATO:"
18 PRINT "NOME, TELEFONE, SIM/NAO, SIM/NAO, SIM/NAO"
20 FOR K = 1 TO 10
30 INPUT NS(K), DS(K), AS(K), TS(K), CS(K)
40 NEXT K
45 REM ROTINA DE PESQUISA
50 FOR J = 1 TO 10
60 IF AS(J) = "SIM" AND TS(J) = "SIM" AND CS(J) = "SIM" THEN GOSUB 100
70 NEXT J
80 END
100 PRINT NS(J), DS(J)
110 RETURN

```

Diagramas de Venn



O quadro representa o conjunto de todas as pessoas cujos nomes estão armazenados no computador e é chamado, nos diagramas de Venn, de conjunto Universo. Os círculos no interior do quadro representam conjuntos

individuais. Os amigos estão no conjunto A. Nem todas as pessoas jogam tênis, mas as que o fazem estão no conjunto T. Uma vez que há pessoas que tanto jogam tênis como são amigas, os dois círculos se sobrepõem

($A \cap T$). O conjunto C identifica as pessoas que têm carro. Este conjunto se sobrepõe aos outros dois e as pessoas que satisfazem as três condições estão incluídas na região em que os três círculos se interseccionam ($A \cap T \cap C$). O

último diagrama representa os mesmos conjuntos (A, T, C), mas as condições que devem ser satisfeitas são outras. Você resolveu jogar tênis ou passear de carro. O conjunto de conhecidos que são amigos e jogam tênis está riscado com

linhas diagonais. O conjunto de amigos que têm carro está riscado com linhas horizontais. A combinação dessas duas áreas representa os amigos com que você poderá jogar tênis ou passear de carro.

mero de telefone, hobbies etc. Uma tarde, você decide jogar tênis em um clube do outro lado da cidade. Você necessita de um amigo (não de um simples conhecido) que jogue tênis e tenha carro. Seu computador recebe a instrução para selecionar o nome e o telefone de todas as pessoas que satisfaçam as três condições: jogar tênis E (AND) ter carro E (AND) ser amigo.

O programa (na página anterior) primeiramente busca dados sobre todos os conhecidos: são amigos, têm carro, jogam tênis? O programa prevê um total de dez conhecidos, mas você pode mudar esse número para quantos quiser (desde que não se esqueça de mudar o número 10 entre parênteses, na instrução

DIM, linha 10). A lista é então examinada pelo uso da instrução IF - THEN, na qual foi introduzida uma condição múltipla. A maioria dos programas em linguagem BASIC admite que o comando IF-THEN opere com condições constituídas de subcondições conectadas por AND e OR. Finalmente, são apresentados o nome e o número telefônico dos conhecidos que satisfazem todas as condições — são amigos, jogam tênis e têm carro.

Em alguns programas, encontram-se combinações muito complexas de funções lógicas. Só na era do computador é que a álgebra de Boole, que em sua época era considerada uma curiosidade, veio a ter o merecido reconhecimento.



D-8100

O D-8100 torna o trabalho de profissionais de diversas áreas, estudantes ou pesquisadores mais produtivo e eficiente.

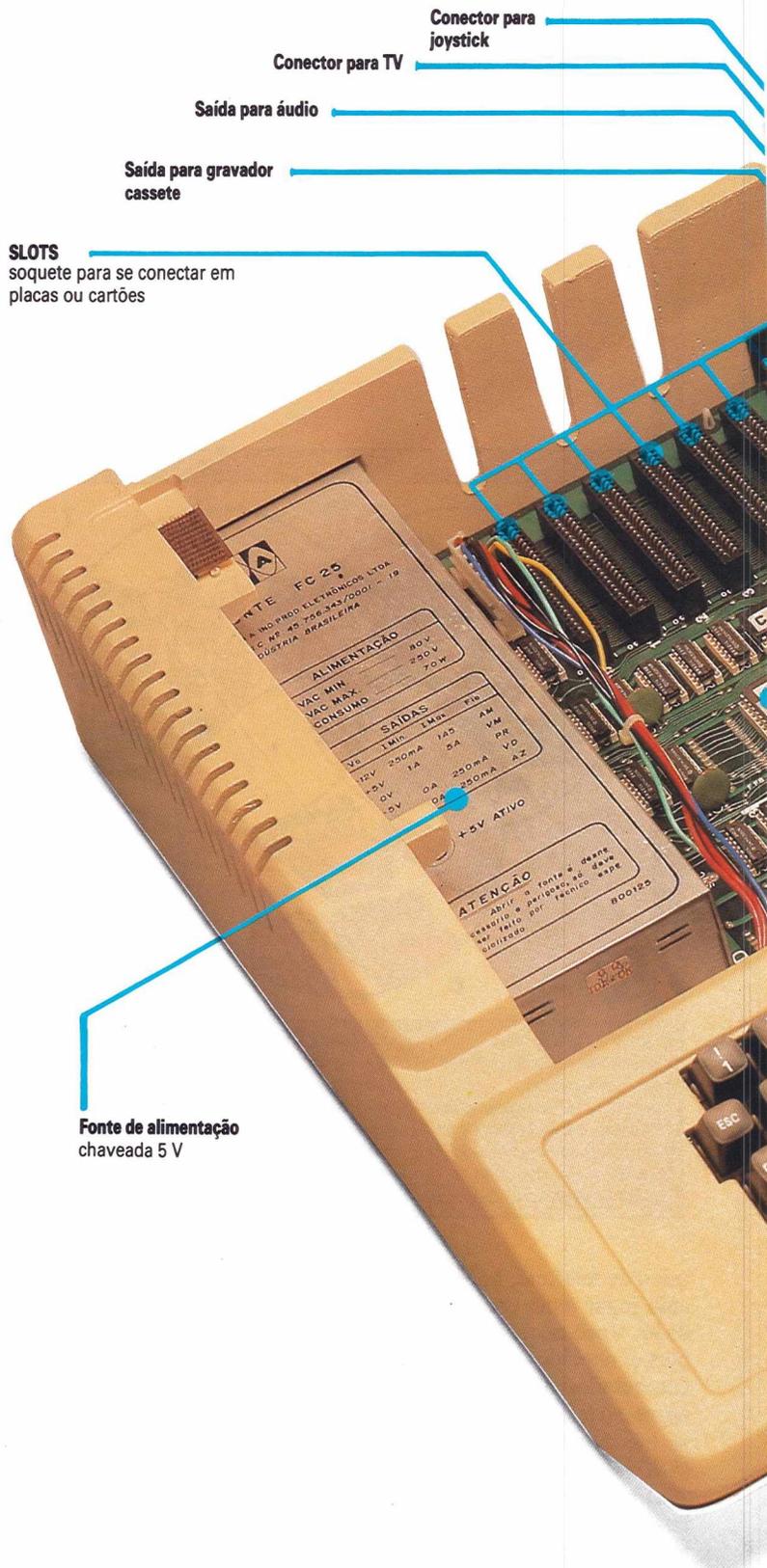
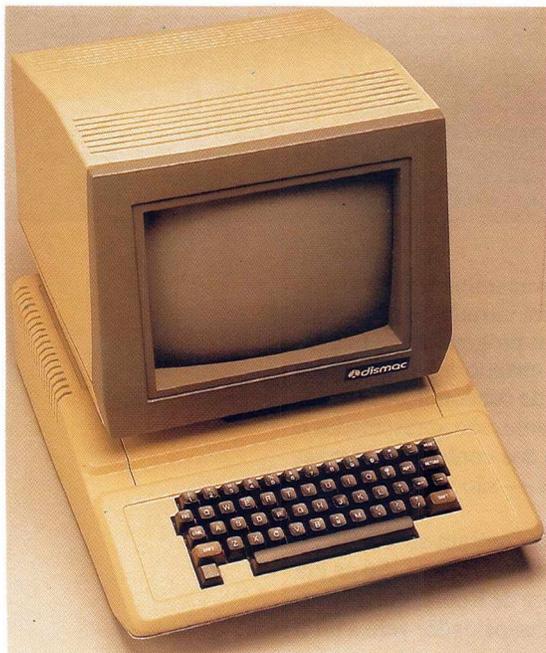
O Dismac 8100 é um microcomputador de uso pessoal em sua configuração básica, transformando-se em um micro profissional na configuração completa. Tem uma gama de aplicações que vai desde o lazer até a comunicação com equipamentos de grande porte. Uma de suas principais características é a facilidade de operação e programação.

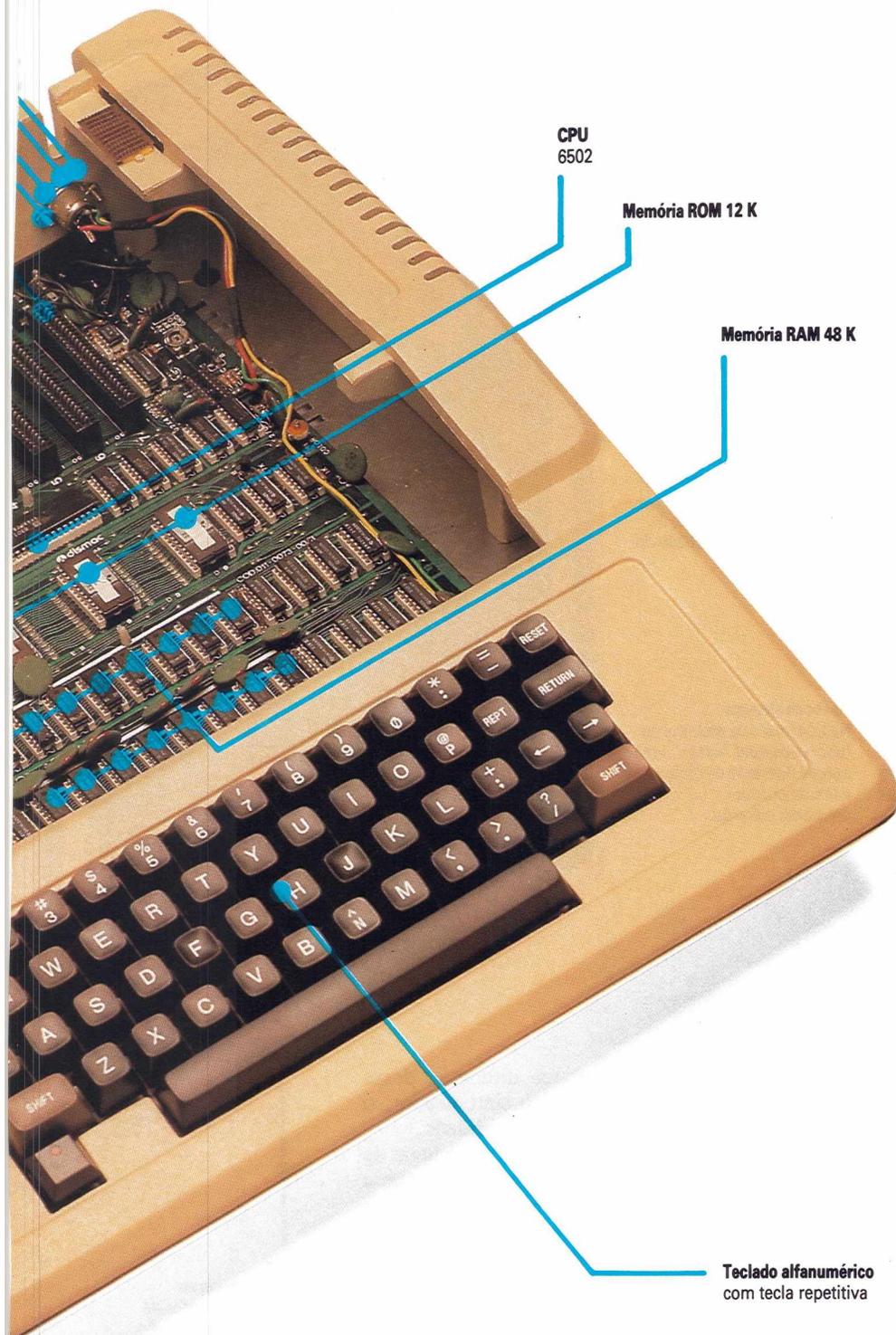
O D-8100 opera com um microprocessador 6502 com clock a 1,023 MHz, podendo opcionalmente utilizar um Z80 com clock a 4 MHz. Possui memória de 12 Kbytes ROM (2 Kbytes para monitor e 10 Kbytes para BASIC residente) e 48 Kbytes RAM expansíveis até 256 Kbytes.

Seu teclado apresenta 53 teclas dispostas de acordo com o padrão QWERTY. Dispõe de letras maiúsculas e minúsculas e tecla de auto-repetição. O monitor é de fósforo verde, de 12 polegadas, podendo operar com 16 linhas e 64 ou 32 colunas (opcionalmente, é possível expandi-lo para 80 colunas com a utilização do cartão "Videx").

Possui conexão para unidades de disco (5 1/4 pol.), oito slots para expansões, interface serial RS232 C, onde podem ser ligados modems e impressoras, interface paralela padrão Centronics, entrada/saída para gravador cassete, conector para controle de jogos (joystick) e saída para áudio.

A alimentação do D-8100 é fornecida por uma fonte chaveada de 5 V e utiliza o sistema operacional D.O.S. 3.3 ou o CP/M.





CPU
6502

Memória ROM 12 K

Memória RAM 48 K

Teclado alfanumérico
com tecla repetitiva

D-8100

MICROPROCESSADOR

6502

CLOCK

1,023 MHz

MEMÓRIA

Possui 48 K de memória RAM e 12 K de ROM.

VÍDEO

Tela de 12 pol., de fósforo verde, que opera com 16 linhas e 64 colunas.

TECLADO

53 teclas com arranjo similar ao da máquina de escrever, com letras maiúsculas e minúsculas. Tecla de auto-repetição.

LINGUAGENS

BASIC, ASSEMBLER, COBOL, FORTRAN, MUMPS, PASCAL.

PERIFÉRICOS

Cassete, interface para impressora paralela (padrão Centronics), unidade de disco (5 1/4 pol.), interface serial RS232 C, modem, joystick.

DOCUMENTAÇÃO

O manual fornecido pelo fabricante contém instruções sobre o sistema operacional e linguagem BASIC. Acompanha também disco do usuário e um Catálogo Nacional de Software, com uma relação de mais de 1.800 aplicativos e jogos disponíveis ao usuário.

Mantendo o foco

Para aumentar a nitidez e a qualidade dos gráficos e jogos, você pode adquirir um monitor e conectá-lo ao seu micro.

À medida que aumenta o uso de computadores, videocassetes e outros equipamentos que necessitam de telas, os preços de monitores especializados estão diminuindo.

Um bom monitor em cores ainda tem preço bastante elevado. Já os monitores monocromáticos apresentam preços bem mais acessíveis.

Com a contínua ampliação da eficiência dos computadores na elaboração de gráficos, em sua maioria coloridos, adquirir um monitor em cores seria a melhor opção.

Há dois tipos principais de monitor em cores, um deles conhecido como RGB (as iniciais de Red-Green-Blue, vermelho-verde-azul), e o outro, como vídeo de sinal composto. O monitor RGB é controlado diretamente, possuindo três canhões que efetivamente permitem ao computador ligar e desligar as cores. Os impulsos usados para sincronizar o computador com o monitor também são produzidos diretamente pelo computador.

Há dois tipos de impulsos sincronizadores, um para cada linha da imagem, outro para a imagem completa. No final de cada composição, o monitor recebe um curto impulso, indicador de que a composição está completa e que o feixe de elétrons (ou seja, o ponto que ele produz) deve retornar ao canto superior esquerdo da composição.

Processo semelhante ocorre no final de cada linha, indicando que a linha está completa, e que o feixe de elétrons deve retornar para o lado esquerdo da tela, preparado para a linha seguinte. Em um monitor RGB, cada um desses sinais (canhão vermelho, canhão azul, canhão verde, sincronizador de linha, sincronizador de composição) é enviado ao monitor através de fios isolados.

Um monitor de sinais compostos, por sua vez, está mais próximo da televisão, porque, nele, todos os sinais são combinados em um único, que é, então, enviado ao monitor por um cabo co-axial. Uma vez no interior do monitor, o sincronizador de composição, o sincronizador de linha e os três sinais de cor são novamente separados e usados no controle da imagem.

O monitor é um televisor sem o sintonizador. É possível transformar o monitor em televisor com o acréscimo de um sintonizador, ou modificar um aparelho comum de TV pela eliminação do mecanismo seletor de canais.

Entretanto, isso é totalmente desaconselhável, pois há voltagens perigosamente altas no interior do equipamento que contém um tubo de raios catódicos. Os técnicos enfrentam esse problema com precaução. Outra razão importante está em que a dispo-

Grade da tela

Para garantir que os canhões de elétrons apontem exatamente para o local correto na tela, uma grade ou máscara é incorporada à superfície do tubo.

Fósforo da tela

A imagem colorida é constituída (como mostra a ilustração) por três cores. Substâncias diversas são depositadas no vídeo. Quando atingidas pelo feixe de elétrons, essas substâncias se refletem em vermelho, ou verde, ou azul, produzindo assim imagem colorida, de acordo com a intensidade do feixe nesse ponto.

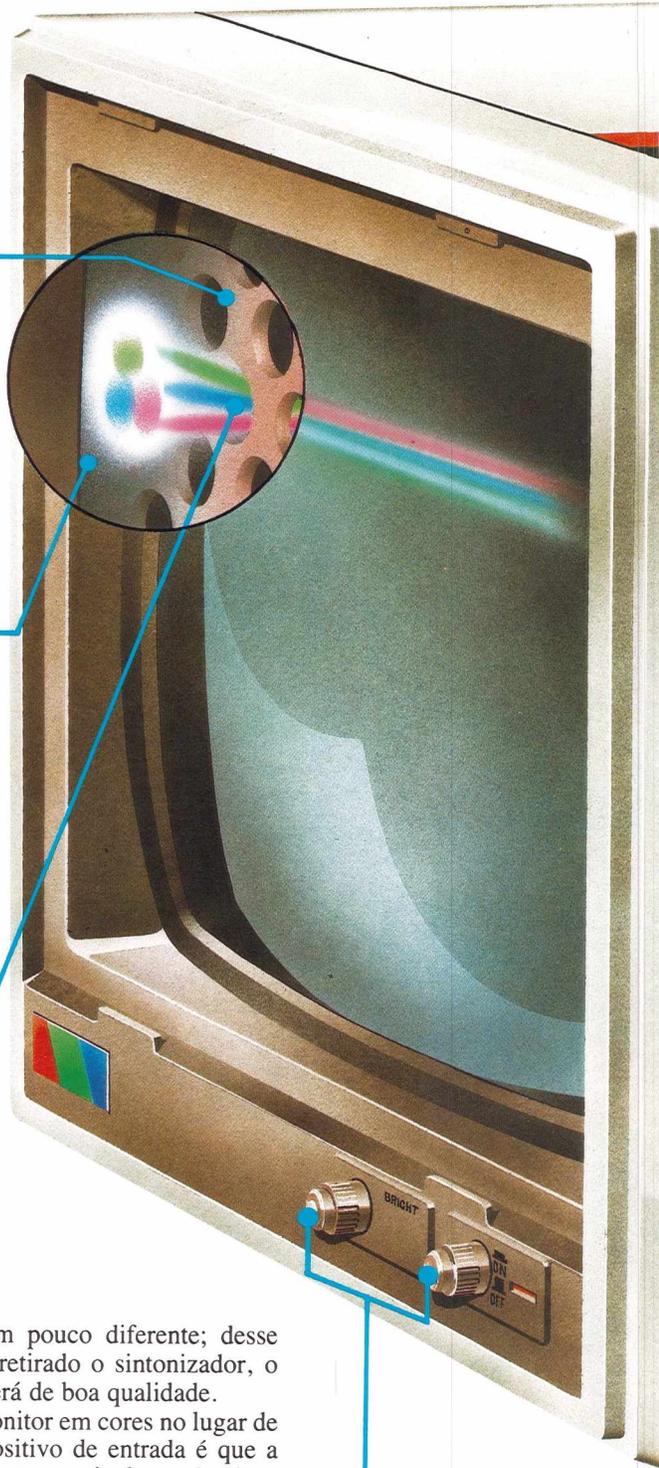
Feixes de elétrons

Há três feixes de elétrons no tubo; cada qual "excita" uma porção diferente de fósforo, a fim de produzir um ponto colorido.

sição dos circuitos é um pouco diferente; desse modo, mesmo que seja retirado o sintonizador, o monitor resultante não será de boa qualidade.

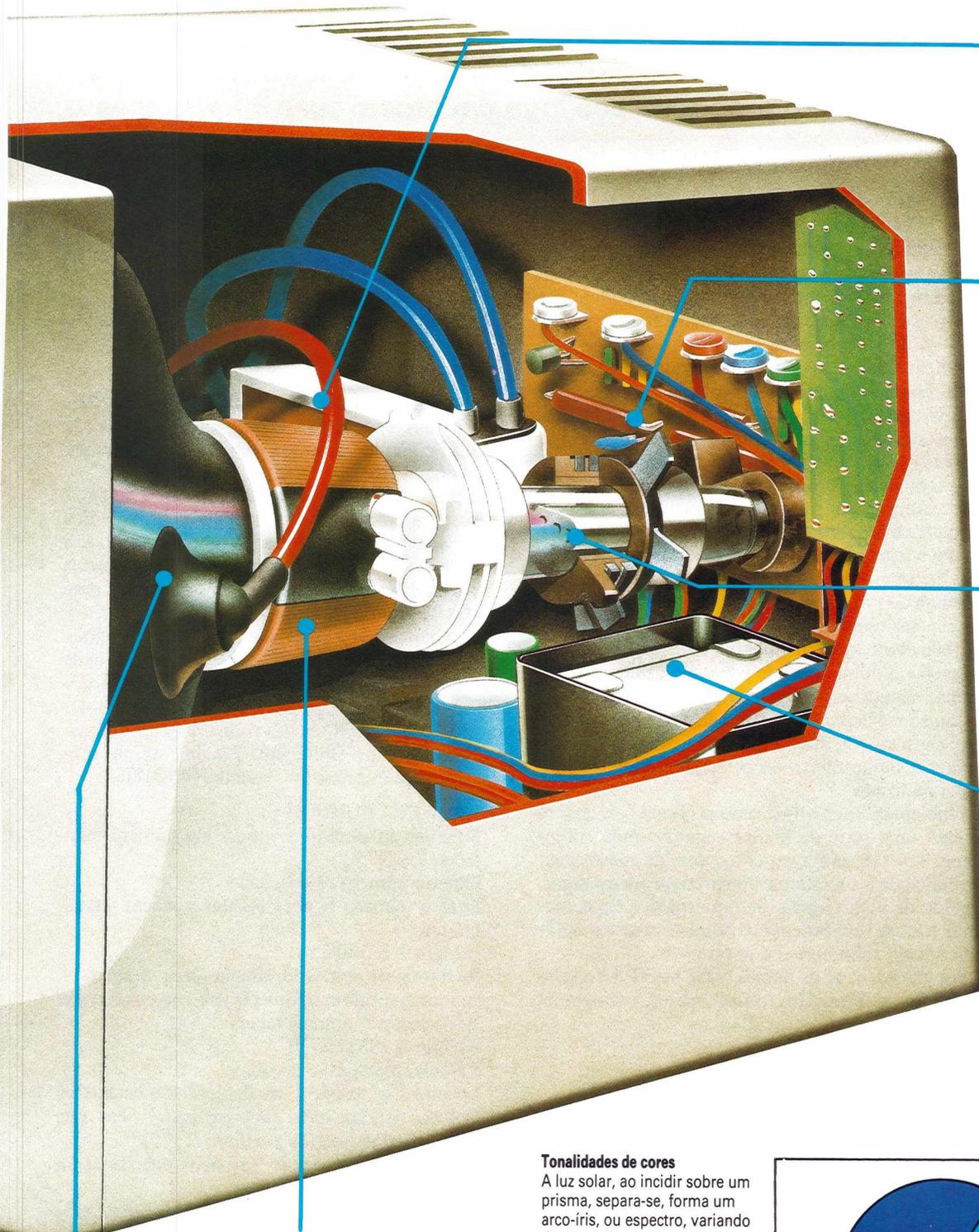
A razão para usar o monitor em cores no lugar de um televisor como dispositivo de entrada é que a televisão apenas opera com um sinal que é sobreposto por uma onda condutora de Ultra-Alta Frequência (UHF, Ultra-High Frequency). Isto significa que o sinal preciso e nítido produzido pelo computador deve ser codificado, enviado através do fio e, então, decodificado novamente. Esse processo produz um sinal "embaralhado", tornando as imagens manchadas.

O monitor, entretanto, não necessita dessa modulação e demodulação do sinal; produz imagens claras e precisas, torna mais fácil a observação visual e dá aos programas uma aparência muito mais profissional.



Controles

O monitor tem vários controles. Os controles de vertical e de horizontal são acessíveis ao usuário. Controle de intensidade de cor e outras variáveis ficam geralmente protegidos porque não têm ajustes frequentes.



Circuitos de alta-tensão
Devido ao fato de exigirem voltagens muito altas, os tubos de raios catódicos necessitam de um circuito de retificação para elevar a voltagem principal (240 V) ao nível adequado.

Painel de circuito principal
Os circuitos necessários à produção das correntes de controle, que movem o feixe e ligam e desligam os canhões, são encontrados aqui. Parte da seção de sincronização de linha, que opera a frequências muito elevadas, é utilizada como suprimento de energia de distribuição.

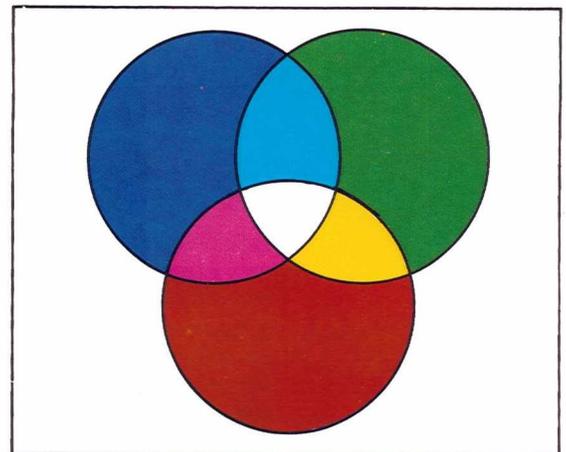
Canhões
Um monitor em cores, assim como a televisão colorida, tem três canhões, de cores vermelha, verde e azul, dispostos em uma linha, na parte posterior do tubo.

Fonte de energia
O tubo de raios catódicos deve ser controlado por voltagens de corrente contínua muito estáveis e requer correntes altas; assim, há necessidade de um transformador bastante potente.

Ligação de ânodo
Emitido pelos canhões, o feixe é acelerado pelo campo de alta voltagem, que deve estar situado na outra extremidade do tubo, devendo ser aplicado por meio desta placa larga e fortemente isolada que está em uma extremidade do cabo.

"Yoke"
É constituído por várias bobinas que produzem poderosos campos magnéticos, que variam rapidamente, e assim o ponto sobre o fósforo é movimentado, produzindo a imagem.

Tonalidades de cores
A luz solar, ao incidir sobre um prisma, separa-se, forma um arco-íris, ou espectro, variando do vermelho, em uma extremidade até o violeta-azulado, na outra. Se esse espectro atravessar outro prisma as cores se recombinam reconstituindo a luz solar original (a chamada "luz branca"). Esse processo de recombinação ou adição é usado no monitor em cores. Pelo acréscimo de várias intensidades das três cores principais, vermelha, verde e azul, qualquer cor pode ser obtida.



Organize seus dados

Veja como um programa põe logo em ordem suas informações.

Este capítulo do seu curso mostra como programas relativamente complexos são subdivididos em sub-programas ou sub-rotinas simples, que podem ser desenvolvidos e testados separadamente.

Além da vantagem de serem testadas separadamente, as sub-rotinas possibilitam que a elaboração de programas siga um desenvolvimento lógico. Há muitos modos de desenvolver programas em linguagem BASIC. Um dos mais comuns é conhecido como "tentativa e erro": você se senta diante do computador e começa a fornecer instruções em linguagem BASIC, sem refletir previamente no que seria necessário para o programa funcionar. Esse procedimento resulta em programas mal estruturados, que não podem dar bons resultados na primeira tentativa. Se a estrutura do programa não estiver clara, não será fácil detectar os erros ou falhas.

Um método muito melhor consiste, antes de tudo, em elaborar numa folha de papel a estrutura do programa, em etapas de detalhamento cada vez maior, até chegar ao desenvolvimento de um programa correto e eficaz. Os diagramas de bloco também poderão ser muito úteis (ver p. 104). Vejamos como isso é realizado.

Problema: desenvolver um programa que deverá receber uma série de nomes seguidos pelo sobrenome. Você deverá inverter a ordem de cada nome, de modo que o sobrenome venha em primeiro lugar, seguido de uma vírgula, de um espaço e do nome. Após isto, deverá ordenar os nomes em ordem alfabética (pelo sobrenome) e imprimi-los.

Por exemplo, se os nomes JOSÉ MARTINS e LUÍS LOPES forem fornecidos (nessa ordem), o programa imprimirá:

LOPES, LUÍS
MARTINS, JOSÉ

Antes mesmo de tentar desenvolver o programa que vai executar essas tarefas, anote os procedimentos de entrada e saída necessários, em termos genéricos:

Estágio 1

Entrada dos nomes em ordem aleatória, nome em primeiro lugar

Saída dos nomes em ordem alfabética, sobrenome em primeiro lugar

Esse procedimento mostra claramente o que desejamos executar. É o primeiro estágio, essencial para obtermos um programa bem elaborado. O passo seguinte consiste em aprimorar os procedimentos do primeiro estágio, mantendo-se sempre atento à eficácia do programa. Não se aprofunde em detalhes, apenas especifique um pouco mais os procedimentos necessários:

Estágio 2

Determinar o total de nomes a ser fornecido

Fornecer os nomes

Inverter os nomes

Ordenar os nomes

Imprimir os nomes

Observe a lista acima e verifique se é adequada. Percebe algum erro nela? Há algum problema de lógica? Se não há, prossiga para o estágio seguinte de detalhamento.

Os procedimentos a que chegamos no Estágio 2 são simples e distintos e podem ser desenvolvidos separadamente, como pequenos subprogramas. Em linguagem BASIC, subprogramas são denominados sub-rotinas. Vamos dar nomes às sub-rotinas para ficar fácil a identificação. A sub-rotina 1, para determinar o número de nomes que será fornecido, pode ser designada por NUMNOM. A sub-rotina 2, onde serão solicitados os nomes, pode ser chamada DADOS. A sub-rotina 3, para invertê-los, pode ser denominada INVERTER. A sub-rotina 4, para ordená-los, chamaremos ORDENAR. Finalmente, a sub-rotina 5, para imprimi-los, será IMPRINOMES.

Estágio 3.1 NUMNOM

Solicitar ao usuário a entrada do número desejado

Obter o número N solicitado

Usar o número N para montar a tabela alfanumérica

Estágio 3.2 DADOS

Se o total de nomes for menor que o número N, sugerir ao usuário o fornecimento de outro nome

Acrescentar o nome à tabela

Estágio 3.3 INVERTER

Calcular a extensão do nome

Localizar o "espaço" no conjunto de caracteres
Colocar os caracteres até o "espaço", em uma variável alfanumérica auxiliar

Colocar os caracteres do "espaço" até o final em outra variável auxiliar

Acrescentar vírgula e espaço para completar a variável

Colocar a segunda variável auxiliar, seguida da primeira, na série original

Estágio 3.4 ORDENAR

Comparar o primeiro componente da tabela com o seguinte

Se o primeiro componente for maior que o seguinte (i.e., posterior na seqüência alfabética), substituir um pelo outro

Comparar o segundo componente com o terceiro
Permutá-los se necessário

Repetir o procedimento até que todos os pares tenham sido comparados



Retornar ao início da tabela e repetir o procedimento de comparação dos pares, até que nenhuma permuta seja necessária, por toda a tabela.

Nota: essa rotina de ordenação é exatamente idêntica à usada na unidade anterior do curso de programação em linguagem BASIC. A seção da “permuta” será usada como uma sub-rotina chamada do interior da sub-rotina ORDENAR.

Estágio 3.5 IMPRINOMES

Imprimir cada nome da tabela até que todos os componentes tenham sido impressos

Todos os estágios necessários à elaboração desse programa estão agora formulados em um nível conveniente de detalhamento. A rotina ORDENAR foi apenas esboçada, pois a estudamos em detalhe anteriormente. A rotina PERMUTA, que é chamada do interior desta sub-rotina, foi deixada completamente de lado. Examinemos agora a facilidade com que programas em linguagem comum podem ser convertidos em linguagem BASIC.

Estágio 4

1. NUMNOM

As três linhas do estágio 3.1 traduzem-se facilmente em instruções em linguagem BASIC. O usuário recebe uma sugestão, através da instrução PRINT; o número é fornecido pelo uso de uma instrução INPUT e uma instrução DIM determina a extensão da tabela.

```
PRINT "QUANTOS NOMES DESEJA FORNECER?"
INPUT N
DIM A$(N)
RETURN
```

A variável N contém agora o número máximo de nomes a serem fornecidos. A instrução DIM dimensiona a tabela de variáveis alfanuméricas. Os nomes dessas variáveis sempre terminam com o sinal de “cifrão”. A variável A\$ sozinha pode conter apenas um conjunto de caracteres. A instrução DIM A\$(N) cria uma tabela que pode conter um número “N” de séries. Tratamos de variáveis indexadas anteriormente, no curso.

A instrução RETURN transfere o controle de volta para o programa principal, na linha seguinte à chamada da sub-rotina. Os valores atribuídos a variáveis na sub-rotina serão “transportados de volta” ao programa principal e poderão ser retomados em alguma outra parte do programa e mesmo em outras sub-rotinas.

2. DADOS

Se o total de nomes for menor que o número N, o programa deverá indicar ao usuário a necessidade de fornecer um nome que deve ser acrescentado à tabela. Isso pode ser feito através de um loop FOR-NEXT. Sabemos que o primeiro nome na tabela será seu primeiro elemento e que o último será o enésimo, deste modo:

```
FOR X = 1 TO N
PRINT "FORNECER O NOME"
INPUT A$(X)
NEXT X
RETURN
```

Isto deve bastar para o fornecimento de todos os nomes para a tabela. Leitores atentos terão percebido o que vai acontecer quando precisarmos inverter a ordem do primeiro e do último nome na sub-rotina INVERTER. Cada componente (nome) da tabela terá de ser obtido novamente, em seguida invertido e recolocado na tabela. Para não aumentar a extensão do programa com esse procedimento, seria mais simples chamar a sub-rotina INVERTER, no interior da sub-rotina DADOS, após a digitação de cada nome. O nome poderá então ser invertido antes de ser colocado na tabela. Para obter esse resultado, temos apenas de acrescentar uma linha, assim:

```
FOR X = 1 TO N
PRINT "FORNECER O NOME"
INPUT A$(X)
GOSUB [INVERTER]
NEXT X
RETURN
```

Todos os nomes da tabela se apresentarão agora em ordem inversa (sobrenome seguido do nome), prontos para ser ordenados.

3. INVERTER

Para inverter a ordem dos nomes, necessitamos saber onde se localiza o “espaço” que separa o primeiro nome do sobrenome. Sabendo sua localização, poderemos utilizar várias funções a fim de obter partes da variável e colocá-las em outras áreas. As funções em linguagem BASIC são comandos que executam operações predeterminadas sobre o valor apresentado após o nome da função. Este valor é sempre apresentado entre parênteses. Muitas funções vêm incorporadas, mas você também pode definir suas próprias funções. Um exemplo caracterís-

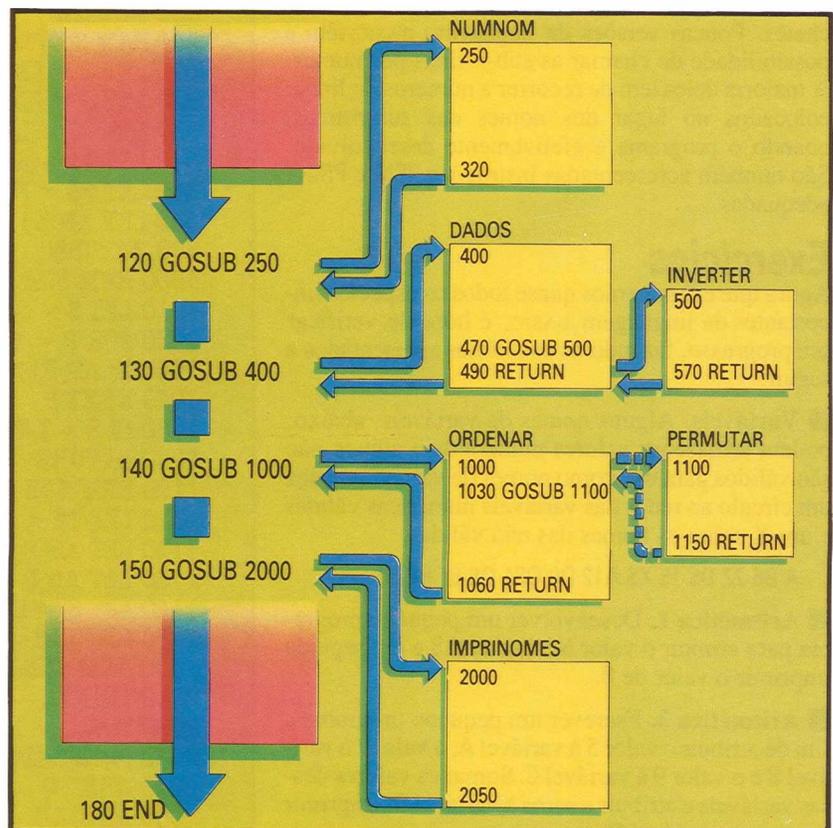
Programas incluídos em programas

Desta vez, o programa principal é muito pequeno. Todo trabalho é, na realidade, realizado pelos subprogramas (chamados sub-rotinas, em BASIC). Cada um dos procedimentos necessários para fazer o programa funcionar está separado e desenvolvido como “miniprogramas”, unidos entre si pelo programa principal.

Quando é processado, toda vez que chega a uma instrução GOSUB o programa desvia para o número de linha da sub-rotina especificada e essa seção do programa é, então, executada. O final da sub-rotina é indicado pela instrução RETURN. Ao alcançá-la, o programa retorna à linha imediatamente seguinte à instrução GOSUB que chamou a sub-rotina.

As sub-rotinas podem ser incluídas em outras sub-rotinas. A sub-rotina DADOS chama uma outra sub-rotina, chamada INVERTER, e a rotina ORDENAR algumas vezes chama uma outra, denominada PERMUTAR.

Subdividir um problema em sub-rotinas separadas e unidas por um programa principal torna mais fácil o desenvolvimento e teste dos programas.



tico de função “incorporada” é a função SQR (). Esta função calcula a raiz quadrada do valor apresentado entre parênteses. Assim: na instrução LET A = SQR (9), PRINT A imprimirá o número 3.

A sub-rotina INVERTER utiliza a função LEN (para calcular a extensão da variável), a função INSTR (para localizar a posição do espaço), a função LEFT\$ (para obter um número determinado de caracteres à esquerda) e a função RIGHT\$ (para obter um número determinado de caracteres à direita). Por enquanto, não discutiremos de modo detalhado como exatamente essas funções funcionam. Faremos um exame mais minucioso das funções em linguagem BASIC na próxima parte do curso.

4. ORDENAR

A sub-rotina ORDENAR e a sub-rotina PERMUTAR, nela incluída, são muito semelhantes às rotinas usadas na unidade anterior do curso.

5. IMPRINOMES

Este procedimento é o seguinte:

```
FOR Q = 1 TO N
PRINT A$(Q)
NEXT Q
RETURN
```

Agora, resta apenas desenvolver o programa principal, que é bastante simples:

```
REM PROGRAMA PRINCIPAL
GOSUB [NUMNOM]
GOSUB [DADOS]
GOSUB [ORDENAR]
GOSUB [IMPRINOMES]
END
```

Colocamos os “nomes” das sub-rotinas entre colchetes. Poucas versões da linguagem BASIC têm a possibilidade de chamar as sub-rotinas por nomes. A maioria deles tem de recorrer a números de linha, colocados no lugar dos nomes das sub-rotinas, quando o programa é efetivamente desenvolvido. São também acrescentadas instruções REM e PRINT adequadas.

Exercícios

Agora que examinamos quase todos os aspectos importantes da linguagem BASIC, é hora de verificar seu progresso, fazendo os exercícios apresentados a seguir.

■ **Variáveis.** Alguns nomes de variáveis, abaixo, podem armazenar valores numéricos e alguns não são válidos para uso como nomes de variáveis. Faça um círculo ao redor das variáveis numéricas válidas e um X sobre os nomes das não válidas.

A B6 2Z D\$ 15 X\$ A12 D9 Q81 Q5 6F H\$

■ **Aritmética 1.** Desenvolver um pequeno programa para atribuir o valor 6 à variável B e em seguida imprimir o valor de B.

■ **Aritmética 2.** Escrever um pequeno programa a fim de atribuir o valor 5 à variável A, o valor 7 à variável B e o valor 9 à variável C. Somar os valores dessas variáveis e atribuir a soma à variável D. Imprimir o valor da variável D.

```
10 REM ESTE PROGRAMA COLOCA NOMES
20 REM EM ORDEM ALFABETICA
30 PRINT "PRIMEIRAMENTE DETERMINE QUANTOS"
40 PRINT "NOMES VOCE DESEJA FORNECER"
50 PRINT "ENTAO FORNECA OS NOMES"
60 PRINT "NOME(ESPACO)SOBRENOME"
70 PRINT "ORDENAR"
80 REM
90 REM ESTE E O PROGRAMA PRINCIPAL
100 PRINT
110 PRINT
120 GOSUB 250
130 GOSUB 400
140 GOSUB 1000
150 GOSUB 2000
160 REM
170 REM FIM DO PROGRAMA PRINCIPAL
180 END
250 REM SUB-ROTINA PARA DETERMINAR QUANTOS
260 REM NOMES SERAO FORNECIDOS
270 PRINT "QUANTOS NOMES VOCE"
280 PRINT "DESEJA FORNECER?"
290 PRINT
300 INPUT N
310 DIM AS (N)
320 RETURN
400 REM SUB-ROTINA PARA FORNECER NOMES
410 PRINT "FORNECER O NOME, DESTA FORMA."
420 PRINT "NOME(ESPACO)SOBRENOME(CR)"
430 PRINT "EX. RAUL AGUIAR"
440 FOR X = 1 TO N
450 PRINT "FORNECER NOME"
460 INPUT AS (X)
470 GOSUB 500
480 NEXT X
490 RETURN
500 REM SUB-ROTINA PARA INVERTER A ORDEM DOS NOMES
510 LET L = LEN(AS(X))
520 LET S = INSTR(AS(X), " ")
530 LET CS = LEFT$(AS(X),S - 1)
540 LET FS = RIGHT$(AS(X),L - S)
550 LET FS = FS + " "
560 LET AS(X) = FS + CS
570 RETURN
1000 REM SUB-ROTINA ORDENAR
1010 LET S = 0
1020 FOR P = 1 TO N - 1
1030 IF AS(P) > AS(P + 1) THEN GOSUB 1100
1040 NEXT P
1050 IF S = 1 THEN GOTO 1000
1060 RETURN
1100 REM SUB-ROTINA PERMUTAR
1110 LET TS = AS(P)
1120 LET AS(P) = AS(P + 1)
1130 LET AS(P + 1) = TS
1140 LET S = 1
1150 RETURN
2000 REM SUB-ROTINA IMPRIMIR
2010 PRINT
2020 FOR Q = 1 TO N
2030 PRINT A$(Q)
2040 NEXT Q
2050 RETURN
```



■ **Aritmética 3.** Observe as seguintes linhas em linguagem BASIC e calcule o valor de C.

```
LET C = 5 + 4 * 3
PRINT C
```

■ **Aritmética 4.** Qual será o resultado impresso pelo seguinte programa?

```
LET A = 3
LET B = 2
LET C = 9
LET D = 4
LET E = (A + B) * (C - D)
PRINT E
LET E = 5
LET E = E * E
PRINT E
```

■ **Comparações 1.** Qual será o valor de X necessário para que a mensagem PRINT seja impressa?

```
70 LET A = 5
80 LET B = X
90 LET R = B - A
100 IF R = 0 THEN GOTO 120
110 GOTO 10
120 PRINT "PARABENS! VOCE ACERTOU"
999 END
```

■ **Comparações 2.** Qual o menor valor de X que fará o programa desviar para a linha 300?

```
250 IF X > 6 * 100 THEN GOTO 300
```

■ **Comparações 3.** Qual o menor valor de Z que fará o programa desviar para a mensagem "Parabéns"?

```
340 IF Z < 10000 THEN GOTO 500
350 IF Z >= 10000 THEN GOTO 520
:
:
500 PRINT "SEU NUMERO ESTA MUITO BAIXO.
TENTE NOVAMENTE"
510 GOTO 600
520 PRINT "PARABENS. VOCE AGORA E UM
MESTRE"
530 GOTO 700
```

■ **Print 1.** Suponha que o valor da variável T seja 50. Escreva uma instrução PRINT que imprima "O VALOR DE T E 50". Sugestão: coloque a "mensagem" entre aspas, use um sinal de ponto e vírgula e o nome da variável.

■ **Print 2.** Examine o pequeno programa abaixo e complete a instrução PRINT, para que se imprima uma mensagem de avaliação como esta:

```
SINTO, MAS SUA CLASSIFICACAO DE 175 FOI MUITO BAIXA
```

Complete a linha de modo que o valor real da avaliação possa variar em cada vez.

```
610 REM A VARIAVEL S E A CLASSIFICACAO
ATE AGORA
620 IF S <= 500 THEN GOTO 640
630 GOTO 700
640 PRINT "SINTO"
```

■ **Print 3.** Que mensagem será impressa quando o programa for processado?

```
200 LET A$ = "MICROCOMPUTADOR — CURSO
BASICO?"
210 LET B$ = "O QUE VOCE ACHA DO "
220 PRINT B$; A$
```

■ **Input 1.** A instrução INPUT é um modo de atribuição de valores a uma variável. Se o programa a seguir for processado, qual tecla deverá ser pressionada a fim de que o programa imprima a resposta "12"?

```
60 INPUT N
70 LET N = N * 2
80 PRINT N
```

■ **Input 2.** O que será impresso neste caso?

```
100 PRINT "POR FAVOR DIGITE SEU NOME"
110 INPUT N$
120 PRINT "OLA"; N$; " SOU SEU COMPUTADOR"
```

A propósito...



Os compatíveis Sinclair (TK82, TK83, TK85, CP 200) possuem esta instrução, mas sua utilização não segue o padrão; assim, elimine a linha 310 e a substitua por: 310 DIM A\$(N, 30)



Na linha 1050, o comando GOTO 1000 vem logo após a palavra THEN. Nesse caso, a maioria dos computadores admite a omissão da palavra GOTO; assim, a linha 1050 pode ser escrita da seguinte forma: 1050 IF S = 1 THEN 1000



Os equipamentos das linhas Apple e TRS-80 (Micro Engenho, Unitron, Maxxi, CP 500, etc.) não possuem esta função. Assim, suprima a linha 520 e a substitua por estas cinco linhas: 515 FOR P = 1 TO L 520 CH\$ = MID\$(A\$(X),P,1) 522 LET S = 0 523 IF CH\$ = " " THEN LET S = P: LET P = L 525 NEXT P



Os equipamentos compatíveis com o Sinclair (TK82, TK83, TK85, CP 200, Ringo) não possuem esses comandos, mas você pode criar sua própria versão, substituindo a linha 460 e as linhas de 510 a 570 por:



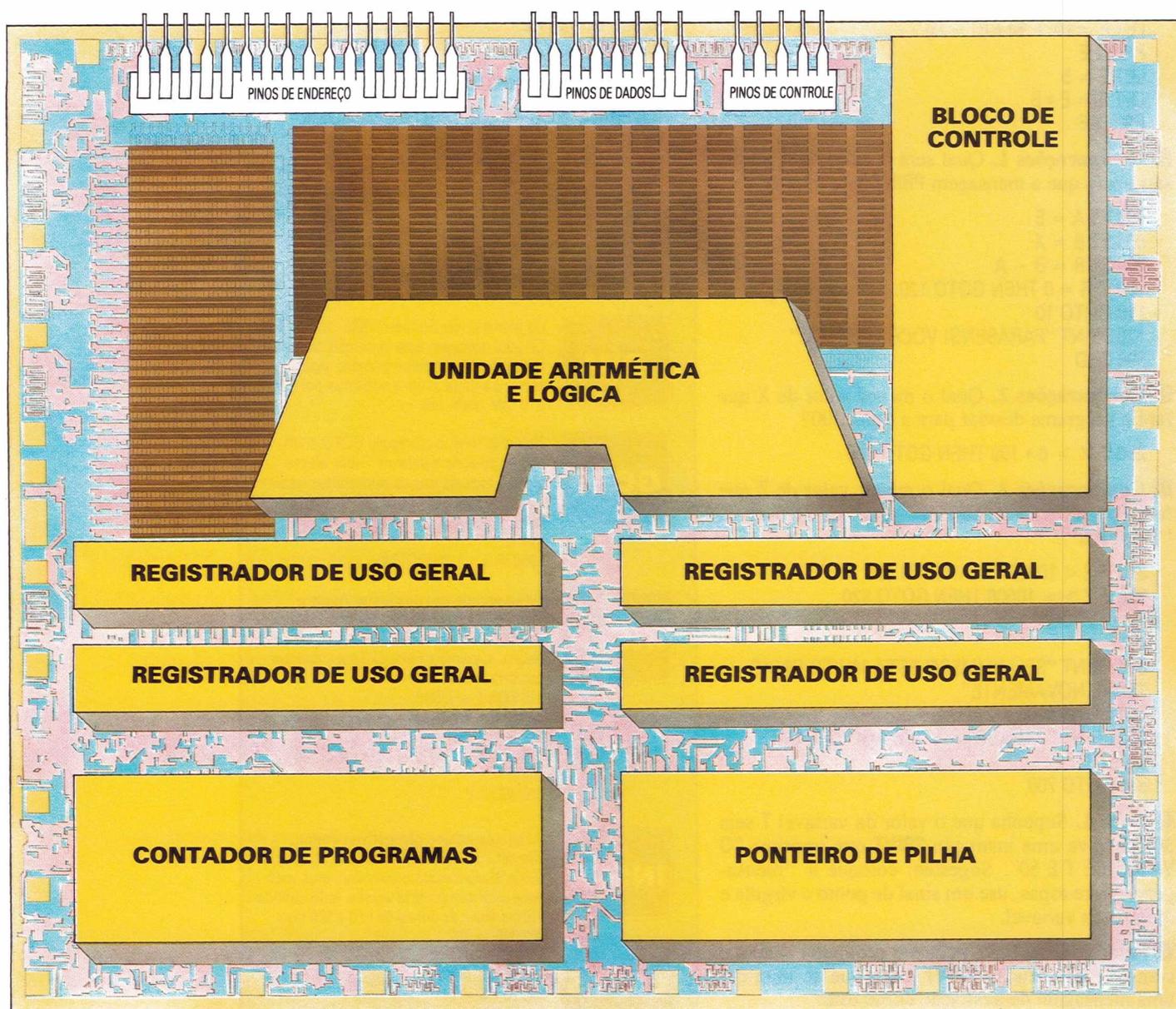
```
460 INPUT Z$
510 LET L = LEN(Z$)
520 FOR P = 1 TO L
530 LET S = 0
540 IF Z$(P) <>" " THEN GOTO 570
550 LET S = P
560 GOTO 580
570 NEXT P
580 LET C$ = Z$(TO S-1)
590 LET F$ = Z$(S+1 TO)
600 LET A$(X) = F$ + " " + C$
610 RETURN
```



Os equipamentos da linha Sinclair (TKs, CP 200 e Ringo) não possuem este comando; assim, substitua-o por STOP.

O centro nervoso

Todas as atividades de computação se dirigem para a Unidade Central de Processamento ou dela provêm.



As atividades do computador são controladas pela CPU (Central Processing Unit), Unidade Central de Processamento. Os termos significam: Unidade — auto-suficiente; Central — o coração do equipamento; Processamento — execução do trabalho. Um computador muito simples (ver ilustração) poderia consistir em apenas uma CPU, alguma capacidade de memória e um sistema de circuitos I/O (entrada e saída).

O computador necessita do circuito I/O para sua comunicação com o exterior. Um exemplo bem simples: o computador utilizado em uma máquina de la-

var roupa precisa de um circuito I/O para fazer funcionar o motor e os aquecedores. A memória é necessária para o armazenamento de instruções e de dados para processamento pela CPU. Os dados processados pela CPU podem incluir números e códigos binários que representam caracteres (letras, dígitos e sinais, como “/” e o sinal “!”).

Se algumas posições da memória contêm instruções para a CPU e outros dados, como pode ela saber quais são as instruções e quais os dados? Para responder a esta questão, devemos examinar o interior de um microcomputador.



As CPUs de microcomputadores de 8 bits (a maioria dos microcomputadores) estão geralmente contidas em um único chip com quarenta pinos, vinte de cada lado, nas laterais mais longas. Cada um deles (com exceção dos pinos de abastecimento de energia de 0 e de +5V) transporta sinais para dentro e para fora da CPU procedentes de outros dispositivos, como a memória e os circuitos I/O.

As CPUs comuns de 8 bits têm dezesseis pinos de “endereço”, que são conectados ao “address bus”. Cada um desses pinos transporta um sinal de saída, que representa um dígito 1 ou um dígito 0. Pela combinação dos dois, pode-se chegar a 65.536 combinações, usadas para selecionar posições específicas na memória.

Há igualmente oito pinos de “dados”, ligados ao “data bus”. Os pinos de dados transportam os dados da memória ou dos circuitos I/O para a CPU e vice-versa.

Uma série de outros pinos transporta sinais de “controle”. Alguns desses sinais são saídas da CPU e outros são entradas. Examinaremos resumidamente como os sinais de controle são usados.

No interior da CPU, existem algumas células de memória, de 1 ou 2 bytes, chamadas registradores, algumas delas reservadas para finalidades especiais. As demais são usadas para armazenamento temporário de informações e são denominadas registradores de finalidades gerais. Há dois outros importantes “blocos” de função na CPU, a ALU e o “bloco de controle”.

A sigla ALU (Arithmetic and Logic Unit) representa a Unidade Aritmética e Lógica. Esta parte da CPU efetua operações lógicas e aritméticas, inclusive (mas não só) a soma, as operações com os conectivos AND e OR, e a mudança dos bits no interior do byte, para a direita ou para a esquerda.

O bloco de controle é um circuito especial projetado para fazer com que o comportamento da CPU esteja de acordo com a instrução recebida da memória. Podemos tomar um exemplo específico, usando os códigos de instrução da conhecida CPU Z80. Se receber da memória a instrução em código 11000110, a CPU somará o conteúdo do byte se-

guinte da memória ao conteúdo de um dos registradores no interior da CPU. Se quisermos, então, armazenar o resultado dessa adição em uma posição específica da memória, a instrução que a CPU deverá receber a seguir será 00110010, seguida de 2 bytes que especificam a posição efetiva na memória, na qual desejamos armazenar o resultado.

Admitamos que o resultado da soma seja 37 (pela notação decimal) e que os 2 bytes que seguem a instrução especifiquem a posição de endereço 33126 (novamente usando a notação decimal). O código de instruções fará com que o bloco de controle ajuste os pinos de endereço para seu equivalente binário 33126 (ou seja, 1000000101100110). Os pinos de controle enviarão sinais indicando à memória que deve aguardar dados e armazená-los. Em seguida, os pinos de dados assumem a forma do dígito binário equivalente a 37 (00100101). Esses dados passam, a seguir, através do “data bus”, para a memória e são armazenados na posição especificada pelo “address bus”. Se, em um estágio posterior, a CPU necessitar desses dados para processá-los de alguma outra forma (mostrá-los na tela ou imprimi-los, por exemplo), uma instrução diferente poderá ser enviada à CPU. O bloco de controle interpretará este procedimento como “acessar o endereço de memória 33126, obter seu byte e armazená-lo temporariamente em um dos registradores internos”.

O total de registradores ou de células de memória temporária no interior da CPU depende da CPU. Podem ser registradores de 8 bits (1 byte) ou de 16 bits (2 bytes). Os registradores especializados geralmente recebem nomes especiais, tais como ponteiro de pilha (“stack pointer”), contador de programas ou acumulador. Os registradores gerais são geralmente denominados por “o registrador X”, “o registrador Y”, e assim sucessivamente.

Um dos registradores de 16 bits — um dos mais importantes — será o registrador “contador de programas”. É uma célula de memória interna que sempre contém o endereço (em código binário) da instrução da memória a ser executada a seguir. Quando chega o momento de obter a instrução seguinte para a CPU, o conteúdo do contador de programas será colocado no “address bus” e o byte nessa posição será transmitido (através do “data bus”) à CPU.

O mais importante dos registradores de 8 bits é o “acumulador”. É o registrador que geralmente armazena (temporariamente) o resultado das operações efetuadas pela ALU, os bytes trazidos da memória ou dos circuitos I/O, ou ainda o lugar em que os bytes são temporariamente armazenados, imediatamente antes de serem enviados para a memória ou para os circuitos I/O.

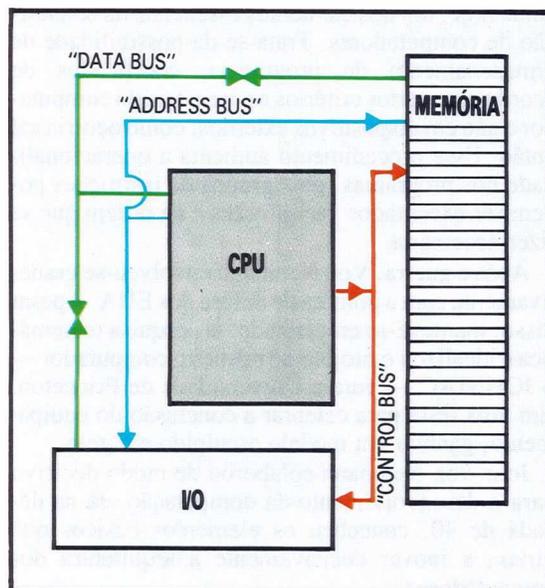
Esta apresentação da CPU foi muito genérica; veremos pontos específicos, em detalhe, mais tarde. O objetivo aqui foi demonstrar que a leitura de instruções especiais que estão na memória induz a CPU a efetuar operações específicas e a ajustar os pinos de endereço para acessar determinadas posições da memória. Os dados são recolhidos dessas locações ou a elas enviados através de um “data bus”. As instruções igualmente fazem com que os sinais de controle do bus informem à memória ou ao circuito I/O se os dados devem ser “lidos” ou “escritos”.

Sob controle

A ilustração mostra uma CPU, com “registradores” de memória, uma Unidade Aritmética e Lógica que inclui centenas de portas lógicas (que efetuam operações como a soma e a complementação de dígitos binários) e, ainda, um bloco de controle. Esse bloco admite a instrução codificada (em binário), interpreta-a e induz todas as outras partes da CPU a atuar de modo adequado. Por exemplo, se uma instrução indica que o conteúdo do acumulador deve ser armazenado em determinada posição da memória, o bloco de controle coloca o endereço nos pinos de endereço, envia sinais de controle, instruindo a memória a armazenar dados, e coloca o conteúdo do acumulador no “data bus”, para transmissão à memória.

Seguindo ordens

Um computador simples pode consistir apenas em uma CPU, alguma capacidade de memória e um circuito de I/O. A memória armazenará as instruções que permitirão à CPU executar atividades específicas. Também armazenará dados para a CPU processar, de acordo com as instruções. A CPU necessitará do circuito I/O para comunicação com o exterior. Se o computador estiver controlando uma máquina de lavar roupa, o circuito I/O receberá os sinais provenientes dos botões do painel e enviará sinais para ligar e desligar os motores e aquecedores. Cada modelo específico de CPU tem seu próprio conjunto de códigos de instrução.





John von Neumann

Autor de trabalhos de grande importância e repercussão, Von Neumann é um dos mais célebres matemáticos do século XX.



O par perfeito

John von Neumann, com sua segunda esposa Klara, que foi programadora dos primeiros computadores.

Só um húngaro poderia entrar por uma porta giratória atrás de você e depois surgir à sua frente. Assim disse John von Neumann, referindo-se ao espírito de competitividade de seus compatriotas.

Ele não era exceção. Pela sua capacidade e extraordinária inteligência, recebeu as mais altas incumbências científicas nos Estados Unidos.

Neumann nasceu em uma rica família judaica, no Império Austro-Húngaro. Suas habilidades matemáticas foram reconhecidas quando ainda era muito jovem: aos 25 anos tinha acumulado duas graduações e um doutoramento, e discutia problemas matemáticos em pé de igualdade com cientistas eminentes, como Einstein e David Hilbert.

Neumann sempre esteve atento aos problemas mundiais. Com a queda do Império Austro-Húngaro, após a Primeira Guerra Mundial, adotou o título de *von* e ingressou na vida acadêmica da Alemanha derrotada. Ao mesmo tempo, estabelecia seus contatos nos EUA, passando os invernos na Universidade de Princeton, em Nova Jérsei, e os verões na Europa, administrando as propriedades de seu pai.

Quando começou a Segunda Guerra Mundial, ele já se encontrava em segurança nos EUA.

Von Neumann tornou seu nome famoso na matemática: reabilitou a teoria dos conjuntos, que Bertrand Russell havia abalado com seus paradoxos lógicos. Era fascinado pela física quântica e também pela teoria dos jogos. Criou o método Monte Carlo, que utiliza números aleatórios para resolver equações matemáticas.

Com a entrada dos EUA na guerra, foi posto a par do Projeto Manhattan, ao qual aderiu, colaborando no estudo de produção da bomba atômica. Ainda se

encontrava ligado ao Manhattan, quando soube que se faziam tentativas de construção de um computador eletrônico e ele próprio foi convidado para o projeto ENIAC (Electronic Numeric Integrator And Calculator). O trabalho estava sob a orientação de engenheiros eletrônicos, mas, como primeiro matemático envolvido, analisou o problema de modo diferente e redigiu um relatório que estruturou e viabilizou o moderno computador.

Após o ENIAC, foi desenvolvido o projeto de outro computador, o EDVAC (Electronic Discrete Variable Computer). Neste, pela primeira vez foi aplicada a idéia de programação interna proposta por John von Neumann. Esse conceito inovador é, ainda hoje, um dos elementos essenciais na construção de computadores. Trata-se da possibilidade de armazenamento de programas, codificados de acordo com certos critérios na memória do computador e não em dispositivos externos, como ocorria até então. Esse procedimento aumenta a operacionalidade dos programas, pois grupos de instruções podem ser executados várias vezes e na ordem que se fizer necessária.

Após a guerra, Von Neumann envolveu-se gradativamente com a política de defesa dos EUA. Apesar disso, manteve-se empenhado na pesquisa matemática e idealizou o projeto do primeiro computador — o JOHNIAC — para a Universidade de Princeton. Em uma festa para celebrar a conclusão do equipamento, ganhou um modelo esculpido em gelo.

John von Neumann colaborou de modo decisivo para o desenvolvimento da computação. Já na década de 40, concebeu os elementos básicos que viriam a inovar decisivamente a arquitetura dos computadores.

1903

Nasce Janos Louis Neumann, em Budapeste, em 28 de dezembro.

1921

Primeiro jornal de matemática, publicado em colaboração com seu orientador.

1933

Torna-se professor da Universidade de Princeton, Nova Jérsei, o refúgio de Einstein e outros matemáticos europeus.

1942

Publica um livro sobre a aplicação da teoria dos jogos na economia.

1943

É nomeado consultor do Projeto Manhattan.

1944

Toma conhecimento do projeto secreto do ENIAC.

1945

Acompanha os testes da primeira bomba atômica.

1947

Relatório sobre o ENIAC e esboço de projetos para um novo computador, o EDVAC (Electronic Discrete Variable Computer), construído segundo suas sugestões.

1951

O JOHNIAC, o computador por ele idealizado, começa a funcionar na Universidade de Princeton.

1951-1953

Presidente da Sociedade Americana de Matemática. Trabalha na teoria dos autômatos.

1955

É nomeado membro da Comissão de Energia Atômica dos Estados Unidos.

1957

Morre de câncer em 8 de fevereiro.



Música eletrônica

A eletrônica é hoje parte integrante do cenário da música profissional. Os novos sons produzidos por pequenos sintetizadores tornam-se cada dia mais comuns.

Divertidos e sérios ao mesmo tempo, os computadores não se restringem aos jogos e ao processamento de informações; podem ser usados também como instrumentos musicais, sem dúvida uma atividade séria. O processo de fazer música artificialmente é conhecido como síntese musical.

O uso de computadores estende-se também às aulas de música, já que podem incentivar o aprendizado, ao mesmo tempo que diminuem o custo das aulas. Num próximo artigo, veremos como fazer esse tipo de música no seu microcomputador; por enquanto, vamos nos restringir ao uso que os profissionais fazem desse equipamento, o que é importante, pois a maioria das idéias que nascem de sintetizadores profissionais acabam sendo a base para os modelos domésticos.

Os instrumentos musicais automáticos sempre foram populares e têm muito em comum com os computadores. A pianola, tipo de piano automático usado na era vitoriana, funcionava por meio de um rolo de papel perfurado e de caixas de música semelhantes a um tambor de metal ou a um disco dentado que produziam sons em um pente de metal.

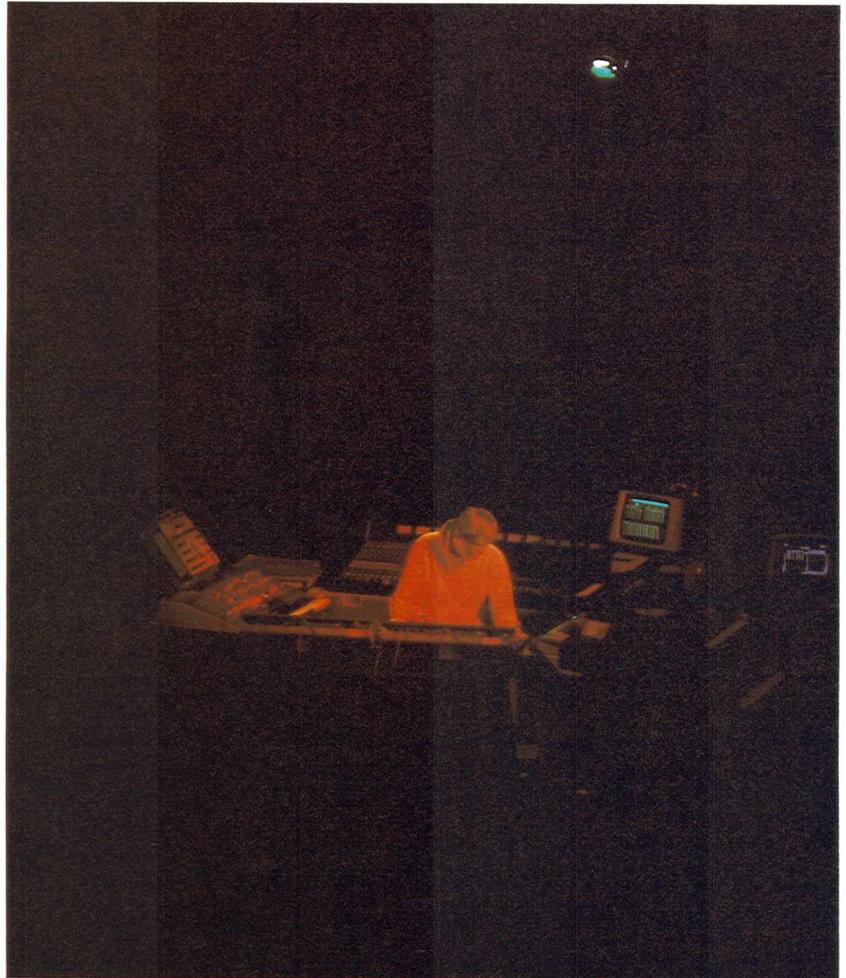
Até os realejos (órgão portátil movido a manivela) eram, de alguma forma, programáveis, já que os tons que produziam podiam ser alterados. Entretanto, não eram apreciados por Charles Babbage, um dos fundadores da computação, que não gostava de ver esse tipo de instrumento nas ruas da cidade.

Atualmente, na Inglaterra, a União dos Músicos tenta obter a proibição de aparelhos musicais programáveis em lugares públicos. Em 1982, foi recomendado o fim do seu uso em gravações e apresentações ao vivo. Preocupa os músicos o fato de que esses aparelhos imitam vários instrumentos ao mesmo tempo — e com isso os instrumentistas poderiam ser dispensáveis.

Os sintetizadores eletrônicos já existem há muito tempo, mas a introdução de técnicas digitais abriu novos caminhos para esse equipamento. Em vez de apertar botões para produzir um som, agora é também possível gravar qualquer som (de um instrumento convencional ou do estouro de um balão), analisá-lo em seus menores componentes e reproduzi-lo depois em qualquer volume.

O som digital é como uma foto no jornal — olhando-a com uma lupa, você poderá perceber que é formada por pequenos pontos separados, enquanto a foto original (análoga) apresenta tons que se encaixam e se completam. Coisa semelhante acontece com um som análogo comum, que pode ser dividido em uma seqüência de dígitos. Essa técnica é conhecida como amostragem.

Esses sistemas são bastante caros, e, dentre eles, o Fairlight e o Synclavier são os modelos mais sofis-



ticados e conhecidos. No entanto, como podem reproduzir os sons de vários instrumentos, tornam-se mais baratos do que o emprego de um músico para cada instrumento.

Com os preços dos computadores e da memória tornando-se menores a cada dia, as máquinas digitais estão ganhando popularidade; mesmo assim, vai demorar até que os sintetizadores analógicos desapareçam por completo. Eles usam uma técnica conhecida como “síntese de subtração”, que pode ser comparada ao modo como um escultor entalha a estátua num bloco de mármore. Todo o trabalho começa com um som básico criado eletronicamente; depois, esse efeito sonoro passa por uma série de processos eletrônicos. Cada processo altera ou divide o som, com o objetivo de adquirir a forma desejada. A síntese de subtração incentiva os músicos a experimentar diferentes processos de combinação e não dificulta o aprendizado do iniciante.

Banda de um só músico

Adeptos do sintetizador como Klaus Schultz (na foto) usam cada vez mais o tremendo poder de seus instrumentos baseados em microprocessadores, a fim de produzir música. Há vinte anos uma orquestra completa seria necessária para produzir a mesma variedade de sons.



Efeitos especiais

Os sintetizadores musicais controlados por um microcomputador ganham popularidade. Eles produzem efeitos musicais que há dez anos só eram possíveis com um dispendioso equipamento profissional. O aparelho da foto lê músicas representadas no papel sob a forma de código de barra. Uma vez na memória, essas músicas podem ser reproduzidas ou alteradas. Muitos dos sintetizadores domésticos são diretamente ligados a um microcomputador, a fim de fazerem uso da tela e da memória adicional.



Já em relação ao sintetizador digital, a criação de qualquer coisa precisa ser planejada cuidadosamente, como se um edifício estivesse sendo construído. O aparelho faz uso da "síntese de adição". O som final é produzido com a adição de componentes sobrepostos. Tem-se que chegar perto do fim do processo antes mesmo de o som ser reconhecível. No entanto, é possível tomar um som convencional, analisar seus componentes básicos, depositá-los na memória do computador, seja na RAM interna, em disco ou fita, e usá-los como um "tijolo" na construção do som desejado.

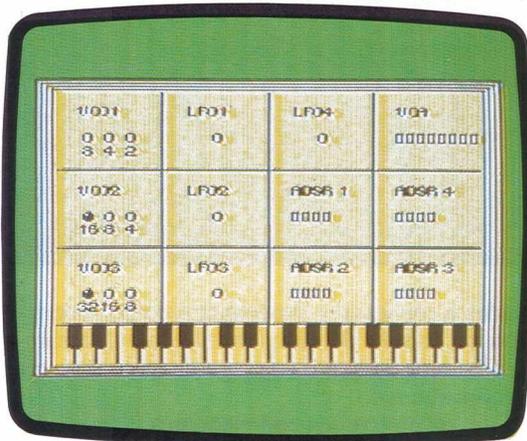
Além da habilidade de criar tamanha variedade de sons individuais, os computadores ainda podem ser usados para armazenar seqüências e composições

voz em rotação mais rápida (ou um disco de 33 rpm tocado em 45 rpm), perceberá que a altura dos sons aumenta muito. Um dos fatos intrigantes em relação aos sintetizadores computadorizados é a facilidade com que superam este efeito, tocando uma música em velocidade diferente da original sem apresentar mudanças de altura nos tons, ou mesmo transpondo um tom em diferentes chaves, mantendo a mesma velocidade.

Pode-se, inclusive, duplicar a seqüência de um pistão, ao mesmo tempo que esse som é transformado em som de trompa. Os dois instrumentos, então, podem ser tocados em uníssono ou, alternativamente, em harmonia.

Caixas de música

Atualmente, já estão disponíveis alguns programas para microcomputadores que exploram sua capacidade de fazer música. A tela pode ser usada para a interpretação visual do que está sendo tocado; e ainda ajuda o músico iniciante, estabelecendo associações entre o teclado QWERTY e o de um piano.



musicais. A maioria dos sintetizadores mais vendidos tem um seqüenciador (aparelho que armazena e se lembra da seqüência de sons) opcional ou integrante. O Fairlight, por exemplo, memoriza até 30 minutos de som em sua memória de disco e oito "vozes", que podem ser de instrumentos individuais. O Synclavier é capaz do dobro.

Percebe-se, portanto, que com os sintetizadores mais caros o compositor, os músicos e o maestro podem ser substituídos por uma só pessoa, que opera o equipamento e geralmente é também um programador de computador.

Se você ouvir uma fita gravada com sua própria

Milagres sonoros

Muitas dessas máquinas são alimentadas com uma linguagem de composição (a linguagem Synclavier é chamada SCRIPT), que não difere muito da linguagem BASIC — inclusive com a enumeração das linhas —, apesar de ser talvez um pouco mais complicada. O recurso mais importante é chamado compilador reverso; toca-se uma música no teclado, e o computador produz uma listagem do programa em SCRIPT, para sua composição. Este processo equivale a poder jogar um novo jogo que você inventou e, pressionando um botão, ter uma listagem completa do programa do jogo.

Caso você não obtenha o resultado desejado, é possível alterar a listagem em SCRIPT, com o uso de um teclado (QWERTY) e de uma tela convencionais, tal como em BASIC. Os sintetizadores Yamaha vêm acompanhados de um sistema visualmente mais atrativo, porém menos flexível, que imprime o que você acaba de tocar, em formato convencional de música, apresentando as notas numa pauta.

O cinema também já utilizou os sintetizadores. O filme *Tron*, de Walt Disney, recebeu críticas excelentes pelos maravilhosos efeitos gráficos produzidos por computadores. O fato de que microcomputadores foram usados para a música e os efeitos sonoros é pouco conhecido. Além da música "real" da Orquestra Filarmônica de Londres, o filme conta com uma grande variedade de sons (que incluem os



rúidos de um dirigível e de um refrigerador) que depois de gravados foram alterados por um Fairlight.

Tantos sons foram envolvidos nesse trabalho que um catálogo tinha de ser sempre atualizado, usando-se para armazená-lo um microcomputador Atari 800 e um pacote de banco de dados (File Manager 800). Algumas das vozes artificiais no filme foram produzidas usando-se um microcomputador e um aparelho sintetizador de voz, que permitiu a mixagem de vozes e música. Os produtores da película conseguiram da Atari um tipo de software que ainda não tinha sido utilizado fora da empresa. Eram programas semelhantes aos de síntese de adição, e estavam sendo usados para criar efeitos sonoros sofisticados em seus softwares de jogos e fliperamas. Pode-se dizer que esse sistema fez pela criação do som o mesmo que o processamento de palavras fez pelo texto.

Na verdade, existem vários elementos que podem transformar um microcomputador comum em um sintetizador musical sofisticado. O Apple é um modelo particularmente popular para esse fim, por ter tomadas na parte traseira onde equipamentos adicionais podem ser ligados. Alguns usam o computador para criar e produzir um som analógico de alta qualidade, enquanto outros incluem um teclado completo como o de um piano.

No mercado de sintetizadores computadorizados, destaca-se o Casio CT7000, que pode ser considerado um "seqüenciador polifônico", que funciona de modo semelhante a um sistema de gravação profissional através de um tape-deck e grande quantidade de memória RAM. Um cassete doméstico tem apenas uma pista por lado; um tape-deck estéreo tem duas. Mas uma unidade profissional pode ter mais de 24 pistas, para que cada instrumento seja gravado separadamente, e só depois agrupados para formar um som completo. Desse modo, com o CT7000 você pode fazer suas próprias sinfonias.

Seu predecessor, o CT701, utilizava uma leitora óptica para ler músicas em códigos de barra. Infelizmente, não é fácil para o usuário criar códigos de barra impressos; assim sendo, este método restringe-se ao uso com as músicas pré-impressas no Casio.

No entanto, você não precisa de um equipamento deste porte para ter êxito no mundo da música; o grupo alemão Trio teve mais sucesso com a música *Da Da Da*, usando um Casiotone VL1, do que Peter Gabriel, com seu Fairlight incomparavelmente mais sofisticado e caro.

Apesar de o VL1 ser um aparelho monofônico (só se pode tocar uma nota por vez), é capaz de simular os sons de diversos instrumentos musicais e memorizar uma seqüência de notas. Ele pode ainda mudar algumas das características nas notas, criando seu próprio som. Tal processo é conhecido como alteração do conjunto — "Attack, Decay, Sustain, Release" (ADSR). Os microcomputadores como o Commodore 64 e o inglês Oric-1 têm exatamente as mesmas características; e já que podem ser programados em BASIC, possuem também um seqüenciador embutido.

Softwares de composição musical estão disponíveis para vários microcomputadores, até mesmo os que têm pequena capacidade musical. Alguns mos-

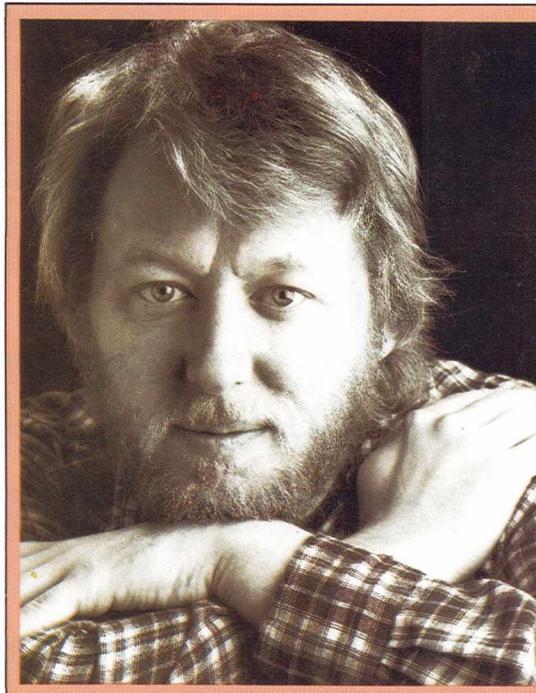
tram uma pauta na tela, e a música é composta através da seleção de notas musicais; é possível fazer isso com a ajuda de um joystick ou de uma caneta óptica que coloca as notas na pauta. Depois, basta que o botão disparador do joystick, ou qualquer outro comando, seja pressionado, e a música é tocada. Em outro tipo de método, a tela pode, também, representar um teclado de piano onde as notas são selecionadas pelo joystick, pela caneta óptica ou com o próprio teclado do computador.

Sem o auxílio desses programas, os efeitos musicais podem ser programados em BASIC. O método exato varia muito de uma máquina para outra, assim como varia a sofisticação dos comandos BASIC em se tratando de gráficos. O Dragon, por exemplo, utiliza apenas uma voz, mas possui um comando PLAY, que inicia uma seqüência de notas digitadas como as letras da escala musical (de A a G). Por outro lado, o Commodore 64 possui muitas facilidades musicais sofisticadas em seu próprio hardware, mas esse comando não está disponível. Mais adiante, você aprenderá a fazer música com seu computador.

Feito em casa

Um músico pode montar seu próprio sintetizador computadorizado, desde que disponha de alguns equipamentos básicos, como um microprocessador de desempenho médio e um conversor analógico-digital. Com um conjunto assim, é possível gerar até oito notas musicais simultaneamente.

Pensando em termos de equipamentos nacionais, pode-se construir um bom sintetizador programável usando como unidade de microprocessamento um CP 500, conectado a um conversor analógico-digital, que irá transformar em impulsos elétricos convencionais os sinais do computador. O CP 500 é um micro da linha TRS-80, baseada nos modelos da Radio Shack americana; esses modelos podem trabalhar com fita cassete ou com disco.



Martin Rushent

Produtor de bandas como Altered Images e Dexy's Midnight Runners, Martin Rushent é um dos precursores da música controlada por sintetizador. Seu estúdio conta com nove sistemas que refletem o estágio atual desse tipo de arte. Cada um deles usa um método de programação diferente e códigos variados para as notas da escala musical. Como não foi possível integrá-los em um único sistema apesar de muitas tentativas terem sido feitas, Rushent elaborou um programa que converte as músicas de um sistema para outro.

O endereço certo

A função da CPU é localizar instruções e dados armazenados na memória. Você sabe o que acontece no seu interior quando as instruções do programa são executadas?

Seqüência de eventos

São muitos os estágios envolvidos, mesmo na mais simples operação da CPU. As instruções, também chamadas "códigos de operação", são lidas no interior do computador. Essas instruções são decodificadas pelo bloco de controle, e levam à realização de operações específicas. Neste exemplo, a instrução 58 é lida da posição de memória 1053. Essa instrução específica induz à realização da seguinte seqüência de procedimentos: o byte na posição seguinte da memória (1054) será lido e armazenado em uma das metades do registrador de endereço de 16 bits da CPU. O byte da posição seguinte (1055) será lido e armazenado na outra metade. Esses 2 bytes agora representam o endereço (algum outro lugar da memória) em que alguns dados estão armazenados. O conteúdo do registrador de endereço é agora colocado no address bus de modo que a próxima posição de memória acessada seja 3071. O conteúdo desse endereço é colocado no data bus e lido pela CPU. Este byte (96, em nosso exemplo) é então colocado no acumulador da CPU, onde permanecerá até ser utilizado por alguma outra instrução. O address bus retornará então a seu endereço anterior +1, de modo que passará a endereçar agora a posição 1056. A CPU sabe que o que estiver contido nessa posição será uma instrução e repetirá uma seqüência análoga de operações. Neste exemplo, 84, a instrução que vem a seguir, é interpretada pelo bloco de controle como sendo para inverter os bits no acumulador. Uma vez que a instrução 84 consiste em "1 byte", a CPU sabe que o byte na próxima posição de memória, 1057, também será uma instrução.

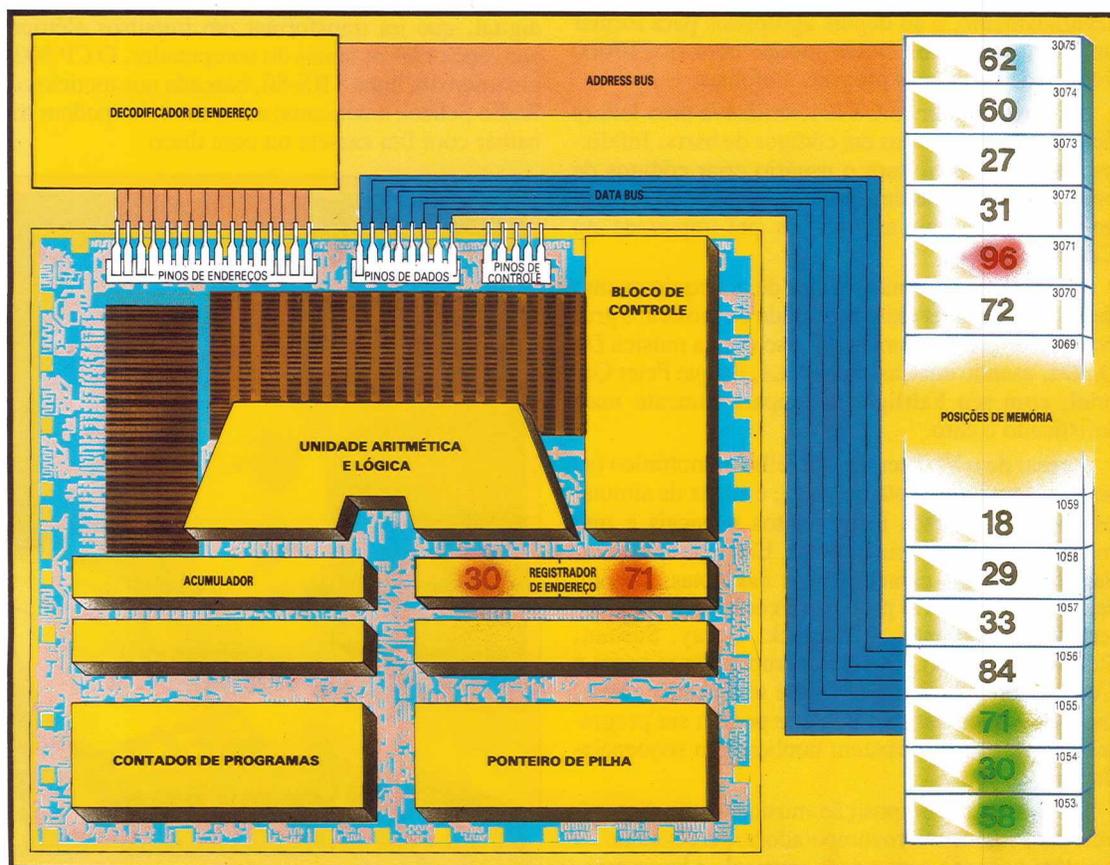
A CPU recebe suas instruções e dados das posições na memória do computador, atribuindo aos pinos de endereço o código binário adequado, permitindo à memória localizar e ler o conteúdo da posição no interior da CPU, através do data bus. Entretanto, essa operação é um pouco mais complexa. O problema está em que os bytes (códigos binários de 8 bits) em qualquer dos milhares de células da memória do computador podem ser instruções, informando à CPU que operações devem ser realizadas, ou dados que a CPU deve, de alguma forma, manipular. O que permite à CPU identificar quais dos bytes representam instruções e quais representam dados?

bytes seguintes da memória, interprete os dados neles contidos, coloque esses dados em um "registrador de endereços", ajuste os pinos de endereço para o mesmo número, dirija-se à posição de memória agora endereçada, obtenha os conteúdos dessa posição no data bus e carregue o acumulador da CPU com esses conteúdos.

Apresentado assim, pode parecer confuso, mas o que acabamos de descrever é um dos métodos de endereçamento da memória usado na CPU do conhecido Z80. Todo o processo de obtenção de 1 byte de dado no interior da CPU é apresentado no quadro abaixo. Suponha que a CPU saiba previamente que o byte a ser acessado em seguida, na memória, será uma instrução (e não dados) e que esse byte se situa na posição de memória 1053 (todos os números utilizados no quadro estão em notação decimal). Esse endereço, 1053, será colocado no address bus. Em código binário, esse endereço seria escrito como 0000010000011101. Os dezesseis pinos de endereço estão "ligados" ou "desligados" de modo a corresponder a esse número. Quando recebe esse endereço através do address bus, o "decodificador de

Reconhecimento de códigos

Primeiramente, verifiquemos o que é uma instrução. É um código em sistema binário que induz à realização de uma seqüência específica de operações no interior da CPU. Desse modo, o código 00111010, quando identificado pela CPU como uma instrução, em vez de simplesmente uma unidade de dados, fará com que a CPU enderece os 2





endereços” o “decodifica” e liga uma e apenas uma das linhas de saída — e esta será a linha correspondente à posição de memória 1053.

No próximo estágio, o conteúdo desse endereço, que corresponde a 58, ou 00111010, em binário, é colocado no data bus e carregado no interior da CPU. Neste ponto, como a CPU está aguardando uma instrução, o byte será interpretado pelo bloco de controle e induzirá à execução de uma seqüência muito precisa de operações.

Essa instrução em particular determina que os próximos bytes da memória conterão 16 bits a serem utilizados como posição de memória, e especifica também que o seu conteúdo deverá ser carregado no acumulador da CPU. Após identificar essa instrução, a CPU prevê que os próximos 2 bytes da memória serão especificações do endereço e que o conteúdo desse endereço deve ser carregado no acumulador. Conseqüentemente, a CPU pode prever que não receberá outra instrução da memória, até que essas operações tenham sido efetuadas; saberá também que a próxima instrução se situará na posição 1056.

A instrução que estamos utilizando como exemplo faz com que o address bus aumente em uma unidade, de modo que a próxima posição de memória endereçada será 1054. O conteúdo dessa posição é então colocado no data bus e carregado na CPU. Entretanto, desta vez, é colocado em uma metade do registrador de endereços. Tendo realizado isto, a CPU incrementa novamente o address bus para endereçar a posição 1055. O conteúdo dessa posição segue para o data bus e é carregado de modo análogo na CPU, sendo armazenado na outra metade do registrador de endereços.

Transferência de números

O estágio seguinte — e lembre-se de que todos estes procedimentos ocorrem automaticamente, como resultado da instrução original — é a transferência dos números do registrador de endereços para o address bus, neste caso 3071. Portanto, é a posição de memória 3071 que agora está sendo endereçada. Esse endereço (0000101111111111, pelo sistema binário) é decodificado pelo decodificador de endereços, que seleciona a célula de memória 3071. O conteúdo dessa posição, 96 (01100000, pelo sistema binário), é colocado no data bus e carregado na CPU. Desta vez, porém, os dados serão colocados no acumulador da CPU. Após isto, o address bus estará ajustado para 1056 e a CPU preparada para aí encontrar outra instrução.

Agora que a CPU tem uma unidade de dados em seu acumulador, que espécie de instrução pode-se encontrar a seguir? Poderia ser praticamente qualquer coisa — as CPUs são capazes de identificar de poucas dezenas a algumas centenas de instruções, dependendo da sua capacidade —, porém, vamos supor que queiramos inverter os dados no acumulador. Inverter significa converter cada dígito 0 em um dígito 1 e vice-versa. A instrução que permite realizar esse procedimento será encontrada no endereço 1056. Em nossa imaginária CPU, o código para essa instrução é 84. Quando este número for recebido pela CPU, os dados do acumulador serão invertidos.

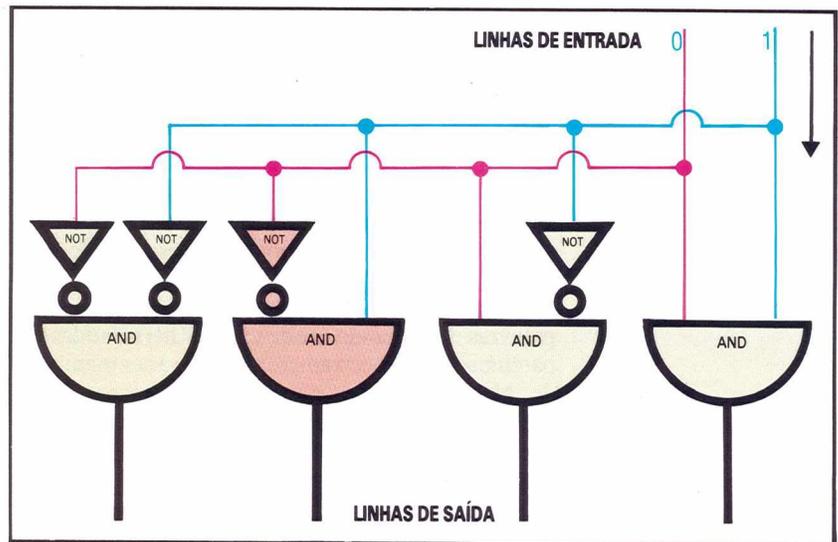
O número no acumulador era 96 (01100000, pelo sistema binário). Esta instrução ocasionará sua inversão para o número 10011111, em binário. A instrução para inversão de números no acumulador é de 1 byte; desse modo, a CPU prevê novamente que o conteúdo da próxima posição de memória, 1057, deverá ser uma instrução e não dados.

Esse método de endereçar uma posição de memória para recuperar uma unidade de dados é apenas um entre os diversos métodos que o programador pode utilizar. Os bytes de instruções utilizados no exemplo citado (58 para carregar o acumulador e 84 para inverter seu conteúdo) são as instruções de nossa CPU hipotética, mas o mesmo princípio se aplica a todos os outros chips microprocessadores. A única diferença é que códigos diferentes são utilizados para as várias instruções, e cada um dos tipos de CPU possui “conjuntos de instruções” próprios.

As posições I/O devem igualmente ter um endereço específico, mas os procedimentos pelos quais a CPU os endereça são os mesmos. Em geral, em microprocessadores de 8 bits, apenas oito das linhas de endereço estão disponíveis para endereçamento de posições I/O. Desse modo, o número máximo de endereços de I/O é 256 — mais do que suficiente para a maioria das aplicações de microcomputadores, mesmo se utilizados comercialmente.

Decodificação de endereços

As dezesseis linhas que constituem o address bus podem identificar qualquer uma das 65.536 posições isoladas da memória. A combinação de dígitos 1 e 0 no address bus é decodificada por decodificadores de endereço. Parte da decodificação é realizada por decodificadores de endereço constituídos de simples portas lógicas em chips montados no circuito; boa parte do procedimento de decodificação é realizada por circuitos equivalentes no interior dos próprios chips de memória. O quadro mostra o processo pelo qual duas linhas de endereço podem ser decodificadas de modo que selecionem um e apenas um entre quatro chips.



A decodificação de endereços é sempre necessária e faz com que o dispositivo escolhido pela CPU (seja ele uma posição de memória, seja uma posição de I/O) mantenha-se ativo enquanto todas as outras posições de memória ou I/O permanecem inativas. Esse processo é chamado “enabling”. Quando há pequena quantidade de linhas de endereço, é possível utilizar chips com portas lógicas simples a fim de realizar a decodificação. O princípio do decodificador de linhas dois-para-quatro é apresentado no quadro.

É comum utilizar esse tipo de decodificação simples para seleção de dispositivos de I/O. Entretanto, à medida que aumenta o total de endereços, também aumenta a complexidade do circuito de decodificação. Quando há 65.536 posições de memória a serem individual e exclusivamente selecionadas, é comum a maior parte da decodificação de endereço realizar-se no interior dos chips de memória.

Descubra as funções

O BASIC apresenta algumas funções incorporadas. Assim, boa parte do trabalho de programação já está pronta. Conhecer o uso dessas funções facilita a elaboração de seus programas.

Vamos admitir que em um de seus programas você queira calcular a raiz quadrada de um número. Há vários métodos para fazer isso. O mais simples e menos satisfatório seria organizar uma tabela dos valores de raiz quadrada, a ser consultada sempre que necessário. Provavelmente, você aprendeu a efetuar esse cálculo na escola. Um método alternativo seria utilizar a “função” de raiz quadrada, existente na maioria das versões da linguagem BASIC. Aqui, a parte aritmética da operação é assumida pelo programa BASIC — o programador não precisa ocupar-se disso. Vejamos como funciona:

```
10 REM ESTE PROGRAMA CALCULA A RAIZ
    QUADRADA
20 REM DE UM NUMERO
30 PRINT "FORNECER O NUMERO QUE VOCE
    DESEJA"
40 PRINT "CALCULAR A RAIZ QUADRADA"
50 INPUT N
60 LET A = SQR(N)
70 PRINT "A RAIZ QUADRADA DE ";N;" E ";A
80 END
```

Digite esse pequeno programa e observe como ele de fato lhe fornece a raiz quadrada de qualquer número digitado. Examinemos as regras do método de utilização dessa “função” de raiz quadrada.

“Funções” em linguagem BASIC são geralmente palavras de comando (neste caso, SQR) seguidas por parênteses que encerram a expressão a ser manipulada. Nesse programa, a variável N representa o número fornecido através do teclado, do qual queremos extrair a raiz quadrada. A linha 60 significa “atribuir a raiz quadrada de N à variável A”. A linha 70 imprime o valor de A.

A expressão entre parênteses chama-se “argumento” da função e não é necessariamente uma variável: também é possível utilizar números. Digite o seguinte programa e observe o que acontece quando processado:

```
10 PRINT SQR(25)
20 END
```

Você perceberá que este programa funciona tão bem quanto o primeiro. De modo análogo, podemos utilizar argumentos mais complexos entre os parênteses: Experimente este:

```
10 LET A = 10
20 LET B = 90
30 LET C = SQR(A+B)
40 PRINT C
50 END
```

Esse pequeno programa pode ser reduzido, pela combinação das linhas 30 e 40, assim:

```
10 LET A = 10
20 LET B = 90
30 PRINT SQR(A+B)
40 END
```

As funções devem ser entendidas como pequenos programas incorporados ao BASIC, que podem ser utilizados pelo programador a qualquer momento. A maioria das versões da linguagem BASIC possui grande quantidade de funções e ainda oferece ao programador a possibilidade de definir outras que poderão ser utilizadas em qualquer programa. Posteriormente, veremos como isso é realizado. Aqui, vamos examinar algumas das funções mais comuns. Há dois tipos: funções numéricas, nas quais o argumento (a parte entre parênteses) consiste em um número, variável numérica ou expressão numérica, e funções alfanuméricas, nas quais o argumento é uma série de caracteres, ou expressão constituída de séries de caracteres. Primeiramente, examinemos algumas das funções numéricas.

Anteriormente, na página 77, apresentamos um programa que calculava o total de azulejos necessários para o revestimento de uma sala. Um pequeno inconveniente nesse programa era que a resposta poderia resultar em frações de azulejos. O programa poderia ter como resultado o número 988,24, por exemplo. Em casos como esse, você necessita arredondar a resposta para o número inteiro mais próximo, e esse arredondamento é uma das funções encontradas na linguagem BASIC. Eis como se processa:

```
10 PRINT "FORNECER UM NUMERO QUE
    CONTENHA UMA FRACAO DECIMAL"
20 INPUT N
30 PRINT "A PARTE INTEIRA DO NUMERO E";
40 PRINT INT(N)
50 END
```

Se esse programa for processado e o número que você fornecer for 3,14, o programa apresentará na tela:

A PARTE INTEIRA DO NUMERO E 3

Naturalmente, lidando com azulejos, é preciso acrescentar 1 à resposta, para garantir a compra de maior quantidade do que o total necessário.

Podemos também querer encontrar o “sinal” de um número — determinar se é negativo, zero ou positivo. Para fazê-lo, a maioria das versões da linguagem BASIC tem incorporada a função SGN. Experimente este programa:

```
10 PRINT "FORNECER UM NUMERO"
20 INPUT N
30 LET S = SGN(N)
```



```
40 IF S = -1 THEN GOTO 100
50 IF S = 0 THEN GOTO 120
60 IF S = 1 THEN GOTO 140
100 PRINT "O NUMERO E NEGATIVO"
110 GOTO 999
120 PRINT "O NUMERO E ZERO"
130 GOTO 999
140 PRINT "O NUMERO E POSITIVO"
150 GOTO 999
999 END
```

Observando os valores para a variável S, apresentados na função SGN, na linha 30 (testados nas linhas 40, 50 e 60), você verifica que há três valores possíveis. Se o argumento entre parênteses for um número negativo, o número apresentado será -1; se for 0, o número apresentado será 0, e se for positivo, será 1. O uso da função SGN na linha 30 economiza várias linhas de programação. Poderíamos ter escrito da seguinte forma:

```
IF N < 0 THEN LET S = -1
IF N = 0 THEN LET S = 0
IF N > 0 THEN LET S = 1
```

É sempre possível obter as mesmas operações realizadas pelas funções BASIC através da programação normal; o recurso às funções apenas economiza tempo, espaço e trabalho.

Eis aqui mais algumas funções numéricas. A função ABS apresenta o valor "absoluto" de um número, isto é, seu valor real, mediante a eliminação do sinal. Deste modo, o valor absoluto de -6 é 6. Experimente:

```
10 LET X = -9
20 LET Y = ABS(X)
30 PRINT Y
40 END
```

A função MAX calcula, entre dois números, aquele que tem valor maior. Assim:

```
10 LET X = 9
20 LET Y = 7
30 LET Z = X MAX Y
40 PRINT Z
50 END
```

A função MIN é análoga à MAX, porém calcula entre dois números aquele que tem valor menor. Experimente:

```
10 PRINT "FORNECER UM NUMERO"
20 INPUT X
30 PRINT "FORNECER OUTRO NUMERO"
40 INPUT Y
50 LET Z = X MIN Y
60 PRINT Z
70 END
```

Observe que estas duas últimas funções possuem dois argumentos em vez de um, que não necessitam ser colocados entre parênteses. Estas funções provavelmente ainda não existem no seu microcomputador. Consulte o manual de instruções BASIC para certificar-se. A maioria das versões de BASIC tem também várias outras funções numéricas, inclusive a função LOG, que calcula o logaritmo do número fornecido; TAN calcula a tangente; COS calcula o cosseno; e SIN calcula o seno. Mais adiante, veremos como algumas dessas funções "trigonométricas" podem ser usadas.

Várias das funções incorporadas na linguagem BASIC operam com séries de caracteres. Utilizamos algumas em nosso programa para ordenar nomes (p. 135); porém, na ocasião, não vimos em detalhe como funcionavam, bem como algumas outras funções alfanuméricas.

Uma das funções alfanuméricas mais úteis é a LEN, que efetua a contagem do número de caracteres presentes em uma série colocada entre aspas ou, ainda, a contagem do número de caracteres atribuídos a uma variável alfanumérica. Experimente:

```
10 LET A$ = "COMPUTADOR"
20 LET N = LEN(A$)
30 PRINT "O NUMERO DE CARACTERES NA SERIE
E";N
40 END
```

Mas por que precisaríamos saber quantos caracteres estão incluídos em uma variável alfanumérica? Para entender por quê, digite o programa seguinte, elaborado para construir um "nome em triângulo". Ele vai imprimir a primeira letra da palavra, depois a primeira e a segunda, e assim sucessivamente, até que a palavra esteja impressa por inteiro.

```
5 REM IMPRIME UM "NOME EM TRIANGULO"
10 LET A$ = "MARIA"
20 FOR L = 1 TO 5
30 LET B$ = LEFT$(A$,L)
40 PRINT B$
50 NEXT L
60 END
```

Agora processe este programa. Você imagina qual será a disposição resultante? Será assim:

```
M
MA
MAR
MARI
MARIA
```

Este pequeno programa utiliza a função LEFT\$, que extrai os caracteres de uma determinada série. Esta função possui dois argumentos. O primeiro especifica o conjunto de caracteres e o segundo (que vem após a vírgula) especifica o número de caracteres a ser extraído da série, iniciando pela esquerda. A série "MARIA" foi atribuída à variável A\$; assim, LEFT\$(A\$,1) deve apresentar a letra M. A instrução LEFT\$(A\$,2) deverá representar as letras MA. O pequeno programa acima utiliza um índice, L, que varia de 1 a 5, de modo que o segundo argumento na função LEFT\$ varia de 1 a 5, a cada passagem pelo loop. Sabíamos exatamente quantos caracteres havia na palavra que queríamos imprimir (MARIA), e foi fácil prever que o limite máximo do loop FOR-NEXT seria 5. Mas o que faríamos se não soubéssemos o número de caracteres que estariam incluídos no loop?

É aqui que a função LEN entra em cena. Ela toma uma série (entre aspas) ou uma variável alfanumérica como seu argumento. Eis alguns exemplos de como ela procede:

```
10 REM PROGRAMA PARA TESTAR A FUNCAO
"LEN"
20 PRINT LEN("COMPUTADOR")
30 END
```

Ao ser processado, o programa deve imprimir o nú-

mero 10. Ele executa a contagem do número de caracteres que constituem a palavra COMPUTADOR e apresenta esse valor. Vamos fazer o mesmo, de forma ligeiramente diferente:

```
10 REM CALCULO DA EXTENSAO DE UMA SERIE
20 LET A$ = "CURSO DE COMPUTACAO"
30 LET L = LEN(A$)
40 PRINT L
50 END
```

Processado o programa, o computador deve imprimir o número 19 na tela. Há dezenove caracteres nesta série, não dezessete; não esqueça que para o computador os espaços entre palavras também são considerados caracteres. Agora utilizamos a função LEN, alterando nosso programa inicial, para impressão do "nome em triângulo"

```
10 REM ESTE PROGRAMA IMPRIME UM "NOME
EM TRIANGULO"
20 PRINT "DIGITE O NOME"
30 INPUT A$
40 LET N = LEN(A$)
50 FOR L = 1 TO N
60 LET B$ = LEFT$(A$,L)
70 PRINT B$
80 NEXT L
90 END
```

Toda vez que este loop se realiza, o valor da variável L aumenta de 1 até N (que representa a extensão da série). Se você fornecer o nome ALFREDO, a linha 40 equivalerá a LET N = LEN("ALFREDO"); assim, N assumirá o valor 7. Na primeira passagem pelo loop, a linha 50 atribuirá o valor 1 à variável L e a linha 60 corresponderá a LET B\$ = LEFT\$("ALFREDO",1); desse modo, um caractere da série será atribuído à variável B\$, iniciando pela esquerda — e este será o caractere A.

Na segunda passagem pelo loop, a variável L assumirá o valor 2; assim, a linha 60 se tornará equivalente a LET B\$ = LEFT\$("ALFREDO",2), o que fará com que os dois primeiros caracteres da série sejam atribuídos à variável B\$. Ela passará a ter como conteúdo os caracteres AL.

A função LEN encontra sete caracteres na série ALFREDO e atribui este valor à variável N; desse modo, na última passagem pelo loop, os sete caracteres da série serão atribuídos à variável B\$ e toda a série será impressa.

Observe que a função LEFT\$ tem uma correspondente, a função RIGHT\$, que toma os caracteres da variável alfanumérica, da direita para a esquerda, exatamente do mesmo modo.

Finalmente, examinaremos uma última função alfanumérica, também utilizada no programa para ordenação de nomes. É a função INSTR, utilizada para localizar a posição de uma série específica (chamada "subsérie") na primeira vez em que ela ocorre no interior de uma série. No programa para ordenação de nomes, a função INSTR foi utilizada para localizar a posição do espaço entre o nome e o sobrenome. Vejamos como ela funciona:

```
10 LET A$ = "CORREDOR"
20 LET P = INSTR(A$,"DOR")
30 PRINT P
40 END
```

Antes de fornecer e processar o programa, verifique se pode prever o valor que será impresso para a variável P. Lembre-se de que a função INSTR localiza a posição de início da "subsérie" dentro da série, na primeira vez em que ela ocorre. Se a série for CORREDOR, a posição inicial da subsérie DOR será 6 — o caractere D da série DOR é a sexta letra em CORREDOR. Algumas versões da linguagem BASIC não utilizam a função INSTR, mas possuem, em seu lugar, uma função semelhante, chamada INDEX. Eis como utilizar a função INSTR (ou INDEX) para localizar um espaço no interior de uma série:

```
10 REM CALCULAR A POSICAO DE UM ESPACO
EM UMA SERIE
20 LET A$ = "PROGRAMACAO BASIC"
30 LET P = INSTR(A$," ")
40 PRINT P
50 END
```

Observe que o segundo argumento na função INSTR (linha 30) é " ". As aspas contêm um espaço — o caractere a ser encontrado. O programa imprimirá o número 12 como valor de P, uma vez que o espaço é a décima segunda posição na série. Calcule o que será impresso se a linha 30 for alterada para:

```
LET P = INSTR(A$,"B")
```

Finalmente, uma função útil usada com a instrução PRINT. Veja o que acontece quando você processa o seguinte programa:

```
10 PRINT "ESTA LINHA NAO E DESEJADA"
20 PRINT TAB(5); "ESTA LINHA E DESEJADA"
30 END
```

Você percebe o que aconteceu? A segunda linha foi impressa a cinco espaços da margem esquerda. A função TAB procede de modo análogo ao tabulador de uma máquina de escrever. Eis aqui outro pequeno programa que utiliza a função TAB:

```
10 REM UTILIZACAO DA FUNCAO TAB
20 PRINT "FORNECER O VALOR TAB"
30 INPUT T
40 LET W$ = "TABULACAO"
50 PRINT TAB(T);W$
60 END
```

Agora você pode retornar ao programa de ordenação de nomes, na página 136, e veja como algumas destas funções foram nele usadas.

Exercícios

■ **Loop 1.** O que será impresso, quando este programa for processado?

```
10 LET A = 500
20 FOR L = 1 TO 50
30 LET A = A - 1
40 NEXT L
50 PRINT "O VALOR DE A E ";A
```

■ **Loop 2.** O que aparecerá na tela, quando este programa for processado?

```
10 REM
20 REM ESTE E UM LOOP DE CRONOMETRAGEM
30 REM VERIFIQUE SUA DURACAO
40 REM
```

A propósito...

TAB

Nos computadores compatíveis com o Sinclair (TKs, CP 200, Ringo), substituir o TAB (número) por TAB número.

LEFT\$

Os computadores compatíveis com o Sinclair não possuem quaisquer destes três comandos, mas você pode elaborar suas próprias versões deles com o uso de "Slicing" ou "partição de variáveis alfanuméricas". Isto pode ser feito com a seguinte equivalência de instruções:

RIGHT\$

LEFT\$(X\$,N) equivale a X\$(TO N)
 RIGHT\$(X\$,N) equivale a X\$(LEN(X\$)-N+1 TO)
 MID\$(X\$,P,N) equivale a X\$(P TO P+N-1)

MID\$

INSTR

Os computadores compatíveis com Sinclair Apple e TRS80 (TKs, CP 200, CP 500, Unitrón, Micro Engenho etc.) não possuem esta função, mas você pode desenvolver um programa substitutivo. Supondo que a linha original seja:

```
20 LET P = INSTR(A$, "DOR")
Substitua-a por:
20 LET X$ = A$ : LET Z$ = "DOR":
    GOSUB 9930: P = U
e acrescente as seguintes linhas:
9929 STOP
9930 LET U = 0: LET X = LEN(X$):
    LET V = LEN(Z$)
9940 FOR W = 1 TO L - V + 1:
    IF MID$(X$,W,V) = Z$ THEN LET
    U = W
9950 IF U <> 0 THEN LET W = L - V + 1
9960 NEXT W: RETURN
```

Nos computadores compatíveis com o Sinclair, substituir a instrução MID\$(X\$,W,V), na linha 9940 e seguir as instruções fornecidas acima.

```
50 PRINT "INICIO"
60 FOR X = 1 TO 5000
70 NEXT X
80 PRINT "FIM"
90 END
```

■ **Loop 3.** Que resultado será impresso se este programa for processado e você digitar o número 60, quando for solicitado?

```
10 PRINT "PENSE EM UM NUMERO E DIGITE-O"
20 INPUT N
30 LET A = 100
40 FOR L = 1 TO N
50 LET A = A + 1
60 NEXT L
70 PRINT "O VALOR DE A AGORA E ";A
80 END
```

■ **Loop 4.** O que acontecerá quando este programa for processado?

```
10 PRINT "GOSTO DE BASIC"
20 GOTO 10
30 END
```

■ **Loop 5.** O que aparecerá na tela quando este programa for processado?

```
10 FOR Q = 1 TO 15
20 PRINT "CHEGA DE LOOPS"
30 NEXT Q
40 END
```

■ **Read-Data 1.** Que resultado será impresso?

```
10 READ X
20 READ Y
30 READ Z
40 PRINT "ESTAMOS TESTANDO A INSTRUCAO
    READ"
50 DATA 50,100,20
60 PRINT X + Y + Z
```

■ **Read-Data 2.** O que será impresso na tela quando o seguinte programa for processado?

```
100 FOR L = 1 TO 10
110 READ X
120 PRINT "X = ";X
130 NEXT L
140 DATA 1,2,3,5,7,11,13,17,19,23
```

Respostas no próximo fascículo.

Respostas aos "Exercícios" das pp. 136-137.

Variáveis

(A) (B6) ~~DS~~ ~~XS~~ (A12) (D9) (Q81) (Q5) ~~DE~~ HS

Aritmética 1

```
10 LET B = 6
20 PRINT B
```

Aritmética 2

```
10 LET A = 5
20 LET B = 7
30 LET C = 9
40 LET D = A + B + C
50 PRINT D
```

Aritmética 3

17

Aritmética 4

25
25

Comparações 1

5

Comparações 2

601 (considerados só os números inteiros)

Comparações 3

10000

Print 1

PRINT "O VALOR DE T E "; T

Print 2

640 PRINT "SINTO, MAS SUA CLASSIFICACAO DE ";S;"FOI MUITO BAIXA"

Print 3

O QUE VOCE ACHA DO MICROCOMPUTADOR—CURSO BASICO?

Input 1

6

Input 2

POR FAVOR, DIGITE SEU NOME
 OLA (SEU NOME) SOU SEU COMPUTADOR

Observe que as respostas às "Variáveis" serão diferentes nos equipamentos que só admitem um caractere alfabético (isto é, sem sufixos numéricos).



Elppa Jr.

Recomendado para grande gama de aplicações, o Elppa Jr. dispõe de diversificado software já desenvolvido.

O Elppa Jr., fabricado pela Victor do Brasil Eletrônica Ltda., é um microcomputador totalmente compatível em hardware e software com o Apple II Plus. Sua Unidade Central de Processamento é o microprocessador 6502 de 8 bits. Encontra-se disponível em versões de 16 e 48 Kbytes de memória RAM e 12 Kbytes de memória ROM (EPROM's). Utilizando cartões de expansão, a memória RAM pode atingir até 320 Kbytes.

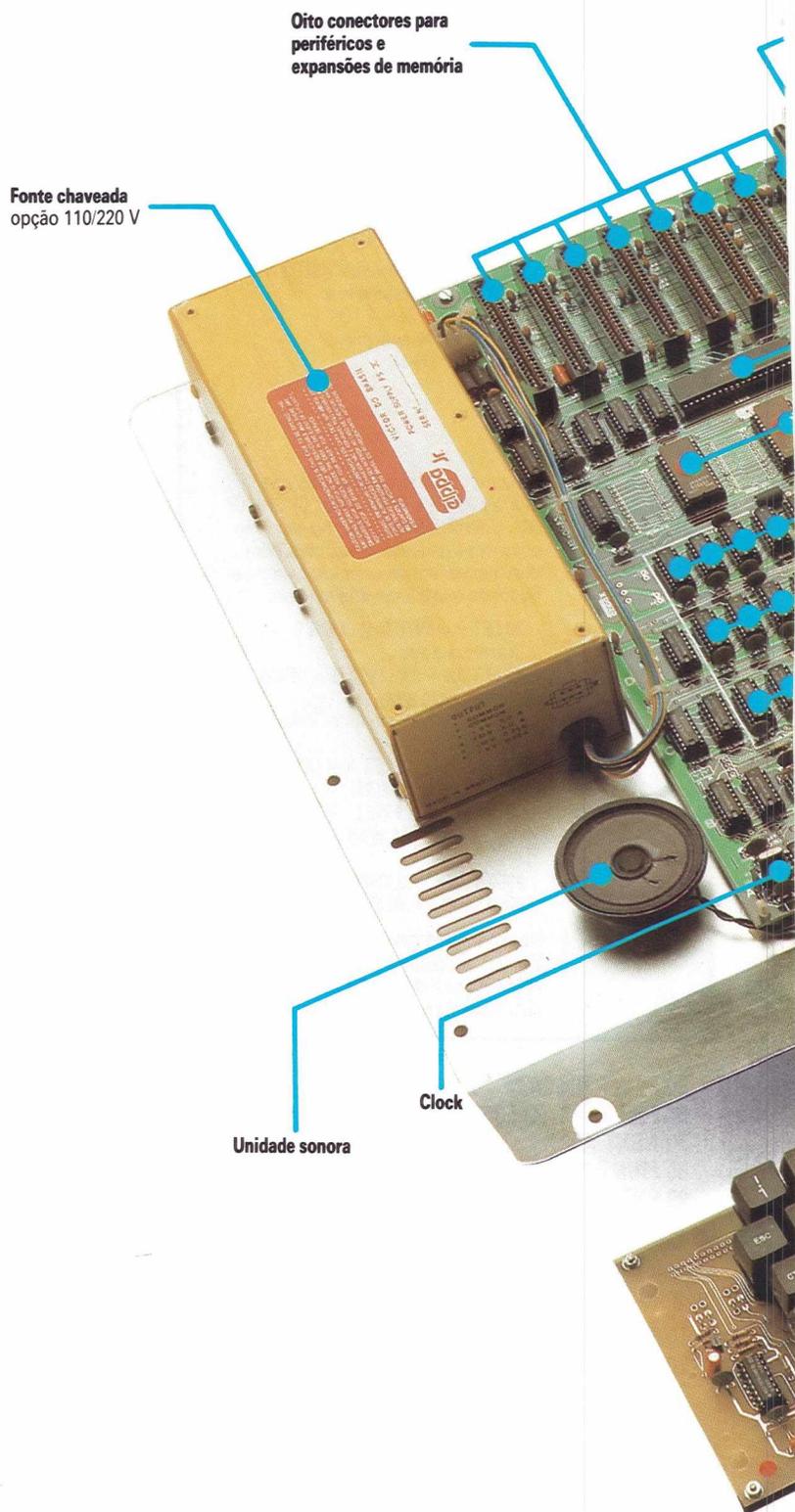
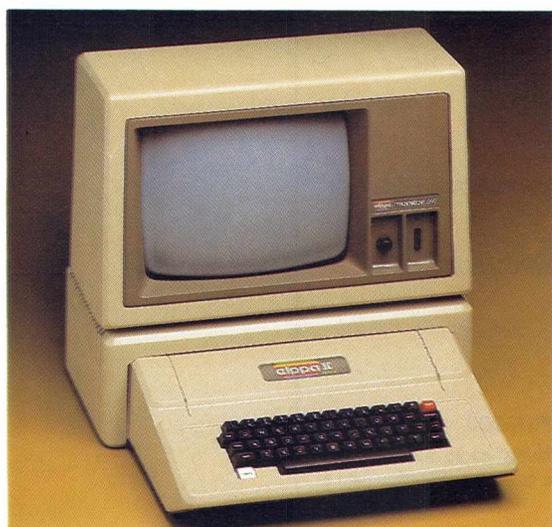
Possui uma fonte de alimentação do tipo chaveada, operando em 110/220 volts. O teclado é profissional, anatômico e de toque preciso. Compõe-se de 52 teclas e utiliza a codificação ASCII.

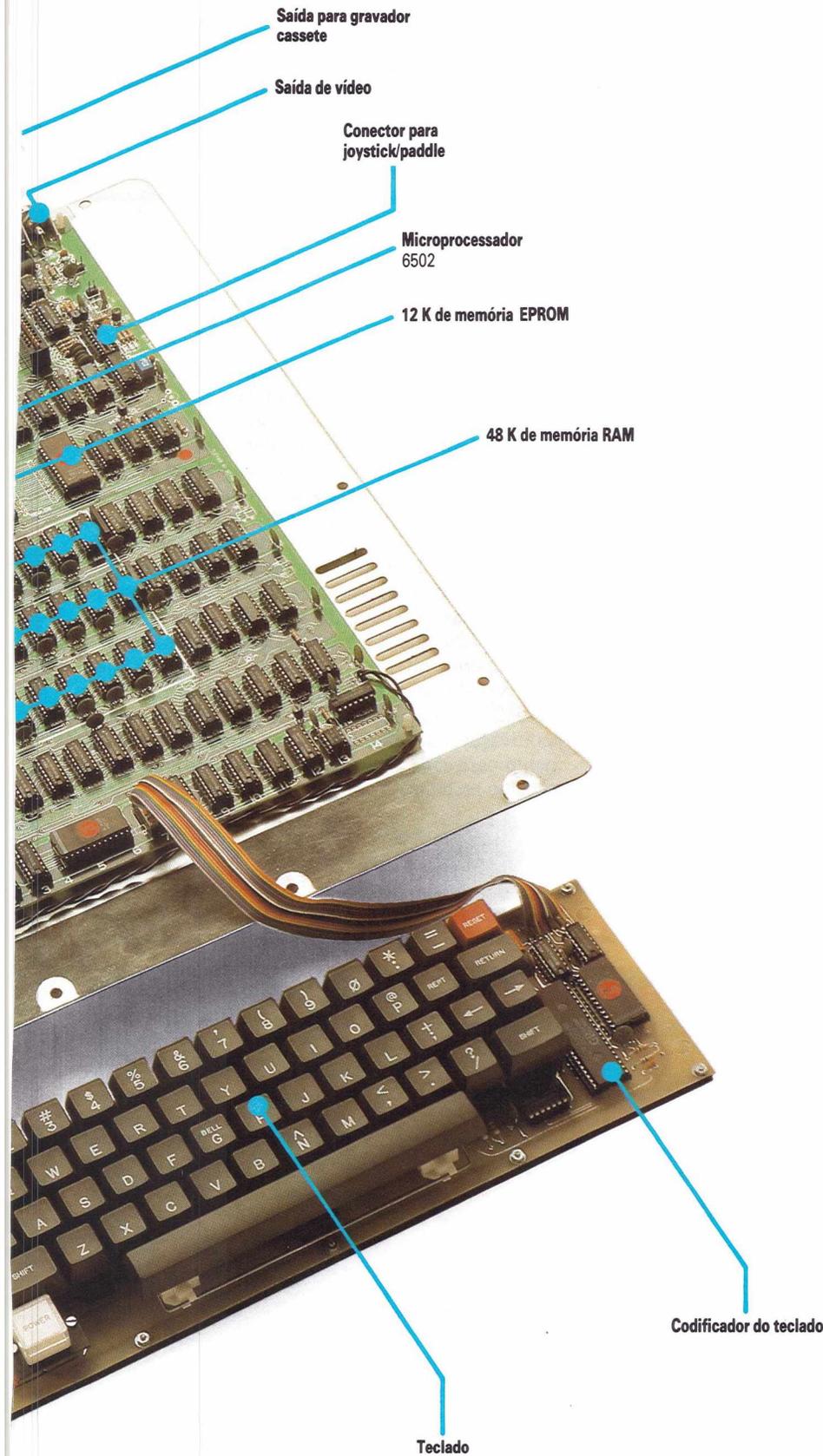
O monitor de 12 polegadas pode ser de fósforo branco, verde ou azul, operando com resolução de 80 colunas ou mais. A capacidade de texto do equipamento é de 960 caracteres (40 colunas por 24 linhas), a gráfica é de 1.920 blocos em baixa resolução e 53.760 pontos em alta resolução. Dispõe de 16 cores para gráficos de baixa resolução e 6 cores para os de alta resolução.

O software residente em EPROM é constituído do interpretador BASIC, do sistema monitor Elppa Soft e do sistema auto-start. As mensagens de erro são enviadas pelo equipamento em português.

Possui oito slots de expansão para interfaces paralela ou serial, impressora, controlador de discos, digitalizador de imagem, sintetizador de voz, cartões de linguagem, joystick.

Pela sua total compatibilidade com o Apple II Plus, o Elppa Jr. pode utilizar todo o software desenvolvido para esse equipamento, tais como editores de texto, bancos de dados, planilha eletrônica, contabilidade, controle de estoque, folha de pagamento, jogos etc.





Elppa Jr.

MICROPROCESSADOR

6502

CLOCK

1,0 MHz

MEMÓRIA

Em duas versões disponíveis: 16 K e 48 K de RAM e 12 K de ROM.

VÍDEO

Pode ser utilizado um monitor ou um televisor normal (preto e branco ou colorido) conectado através da placa PAL-M aos terminais da antena.

TECLADO

Profissional, tipo máquina de escrever, com 52 teclas, algumas com dupla função e auto-repetitivas.

LINGUAGENS

BASIC, residente. Opera também em CP/M, COBOL, FORTRAN, PASCAL, TASC, FORTH e PILOT.

PERIFÉRICOS

Cassete, monitor, até 44 unidades de disco (5 1/4 pol.), impressora, interfaces: gráfica/não-gráfica, paralela ou serial, controlador de discos, 80 colunas, SOFT SWITCH, RS232, expansões de memória, sintetizador de voz, codificador PAL-M, cartões de linguagem, cartões de genlock, digitalizador de imagem, tablet gráfico, ultraterm. Aceita joystick e controladores de jogos.

DOCUMENTAÇÃO

Completo e detalhado manual de instruções com inúmeras gravuras e fotos que bem ilustram a instalação, manuseio e operação do Elppa Jr.



Gráficos em dimensão

A grande capacidade de memória permite a alguns micros produzir imagens coloridas e com rápidos movimentos.

Um dos aspectos mais notáveis dos microcomputadores está na sua capacidade de criar gráficos e composições com movimento, o que é conhecido como animação. Em muitos microcomputadores, o usuário traça pontos individuais, desenha linhas e círculos e altera as cores do fundo e do primeiro plano.

Para jogos de simulação e de ação rápida, precisamos ter condições de simular movimentos. O meio mais fácil de fazer isso é produzir uma série de figuras imóveis, uma após outra, e com rapidez suficiente para dar a ilusão de movimento. As imagens de televisão são obtidas de modo semelhante.

Velocidade de ação

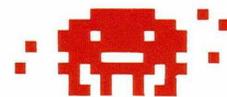
Outro recurso utilizado para criar a ilusão de movimento consiste em imprimir um caractere, apagá-lo e imprimi-lo de novo em posição ligeiramente afastada da inicial. A cada passo, a distância do afastamento deve ser mínima, para obtermos um fluxo contínuo de movimento. Analogamente, o tempo para produzir a forma e apagá-la deve ser o menor possível.

O uso da linguagem BASIC para a produção de ani-

cional de controle dos caracteres individuais, à medida que estes se acumulam na tela.

Muitos computadores, tais como o Commodore 64, o Sord M5, o Texas Instruments TI99/4A e os da linha Atari, superam o problema utilizando as mesmas técnicas empregadas nas máquinas de flipperama. A técnica é conhecida como "sprite graphics".

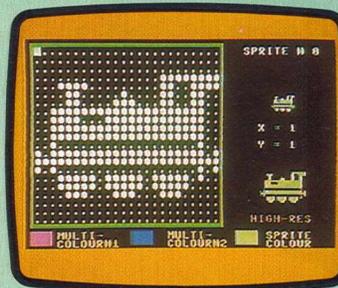
"Sprites" são "objetos", ou figuras, que se movimentam na tela de modo independente uns dos outros. Isso é feito pela simples alteração dos conteúdos de duas posições de memória especificadas pelas coordenadas X e Y (as posições direita-esquerda e superior-inferior). De modo geral, a coordenada X pode variar de 0 a 255 e a coordenada Y de 0 a 191. Alguns equipamentos chegam a admitir a especificação da velocidade e da direção do movimento para cada figura, deixando ao computador o trabalho restante.



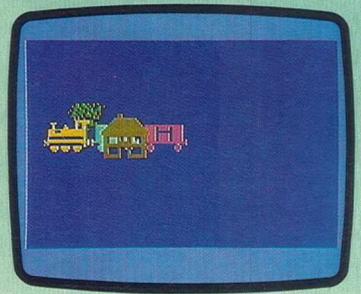
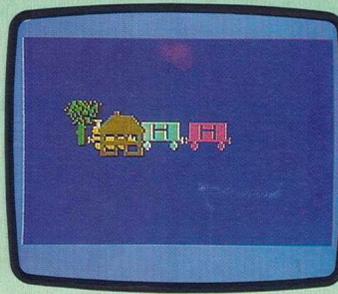
Para este tipo de gráfico, são necessários, em geral,

Percurso do trem

O trem aqui apresentado foi construído como três figuras (a máquina e dois vagões). A imagem foi criada em escala maior usando-se recursos de edição e, em seguida, armazenada em cassete, com as imagens da casa e da árvore.

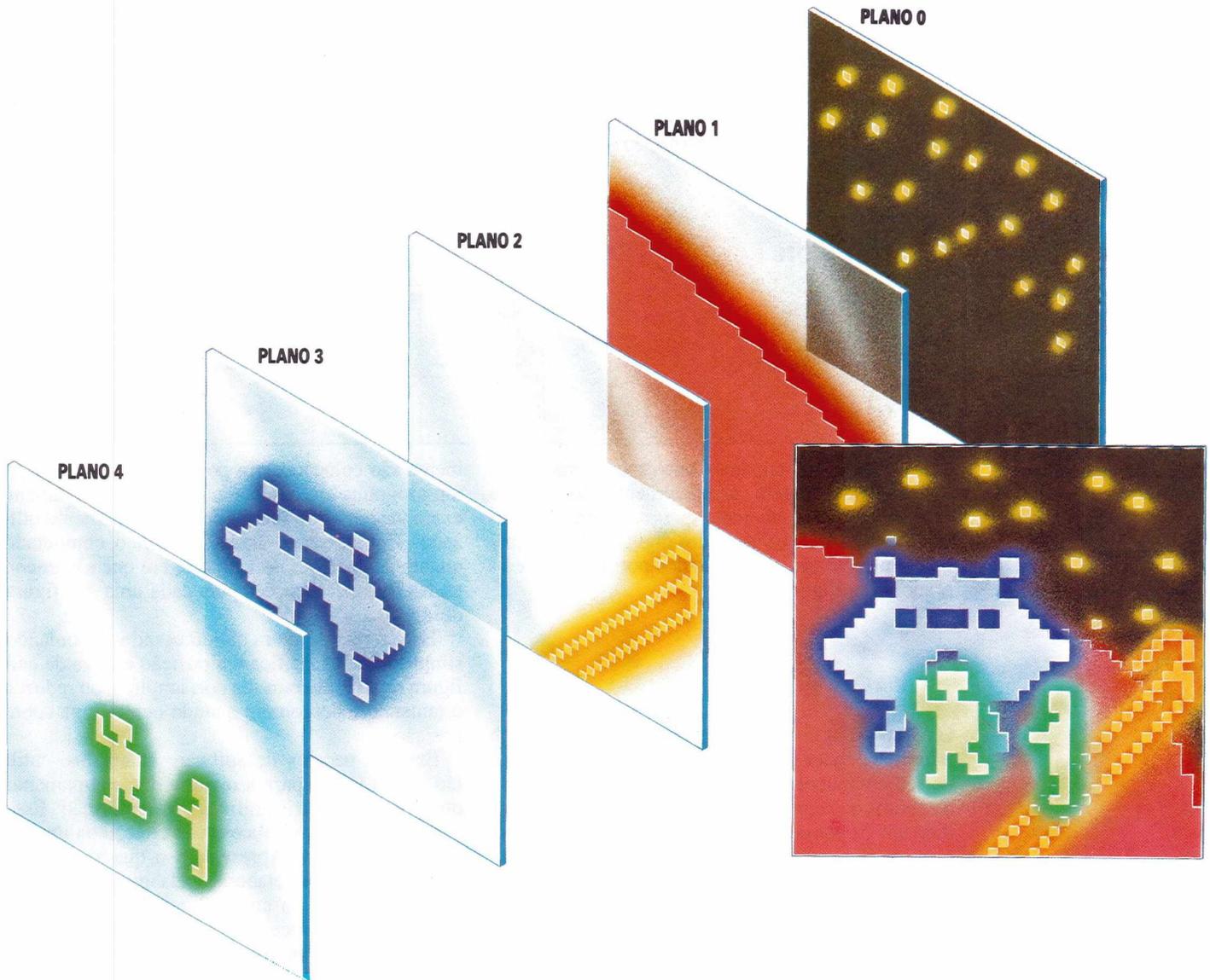


Novamente colocadas na memória, as figuras são a seguir controladas pelos comandos POKE para determinar as posições, a cor e a velocidade do trem na tela. A "prioridade" das figuras é especificada de modo que o trem passe por trás da casa e na frente da árvore.



mação resulta em movimentos lentos. Um modo de superar esse inconveniente é recorrer à linguagem Assembly, um procedimento que exige prática, cuidado e atenção, para obtermos composições isentas de oscilações. Além disso, temos o problema adi-

chips ou circuitos especiais pertencentes ao hardware do computador. É possível comprar software feito para outros computadores com a finalidade de obter efeitos semelhantes, mas, quase sempre, os resultados são menos satisfatórios.



A velocidade de cada um dos elementos varia, pois eles se situam em “planos” diferentes. Desse modo, uma tela repleta de gráficos é constituída de uma série de planos sobrepostos, embora perceptíveis aos olhos apenas como um único plano na tela.

Efeito tridimensional é obtido pela passagem da figura na frente ou atrás de outra. Os elementos são numerados de 0 até o maior número disponível, que varia de acordo com o equipamento. Se dois elementos se sobrepõem, o que tem número menor aparece na tela. Desse modo, pela cuidadosa ordenação das figuras, obtém-se facilmente efeito tridimensional, como o de um trem passando na frente de uma árvore. Outra possibilidade é encobrir parcialmente o trem, quando este passa por uma casa situada em um plano de numeração mais baixa.

Pelo uso da variedade de cores que seu computador proporciona, cada figura pode ser colorida individualmente. Em alguns casos, pode-se ampliá-la ou reduzi-la com a mudança do conteúdo de uma posição de memória.

Evidentemente, em programas de jogos é que os “sprites” revelam melhor suas possibilidades, e um dos efeitos especialmente interessantes é o conhecido como “detecção de colisão”. O equipamento é

programado para, na ocorrência da sobreposição de dois ou mais elementos (por exemplo, quando um míssil atinge uma espaçonave inimiga), saltar para uma outra parte do programa, criando gráficos que simulam uma explosão (e aumentam a contagem de pontos do jogador).



Para utilizar essas figuras é preciso criá-las previamente, de modo muito semelhante ao usado na elaboração de um novo caractere. As letras do alfabeto, os números e símbolos gráficos especiais estão armazenados no interior do computador em um chip chamado gerador de caracteres.

Os caracteres, como mencionamos anteriormente no curso, são em geral construídos por uma matriz de oito pontos por oito, ou “pixels”. O tamanho máximo dos elementos varia de um equipamento para outro, mas terá vários caracteres de largura e altura. No microcomputador Commodore 64, por exemplo, o tamanho máximo é de 24 pixels de largura por 21 de altura.

O melhor meio de criar uma figura é desenhar um

Em planos diferentes

Pelo uso de “sprite graphics”, imagens complexas podem ser construídas através da colocação dos elementos componentes em planos diferentes, que se sobrepõem uns aos outros. Embora os gráficos sejam vistos na tela do computador apenas como um plano único, a grande vantagem desse sistema está em fazer com que objetos em planos diferentes se movam de modo independente. O programador determina uma ordem de prioridade, e assim, quando dois deles se sobrepõem na tela, aquele com prioridade maior é visto na frente do outro, com um efeito tridimensional. Computadores com “sprite graphics” também prevêm a interrupção do programa sempre que duas figuras entrarem em contato, criando o efeito de explosão, para aumentar os pontos do jogador.



Dimensões da figura

Existem "sprite graphics" para microcomputadores como o Commodore 64, o Sord M5, o TI99/4A e a linha dos Atari.

As dezesseis cores normalmente existentes aumentam para 256 no Atari, pois cada uma se apresenta em dezesseis graus de "luminosidade".

Nos jogos, é interessante saber quando duas figuras se chocam — por exemplo, duas espaçonaves — e, para isso, é fornecido um dispositivo para detecção de colisão.

O efeito tridimensional é produzido pela colocação de cada figura em um plano diferente; quanto mais planos forem possíveis, melhor. O tamanho máximo de cada figura é expresso em pixels.

Elas são controladas de modo a expandir-se ou a reduzir-se e podem ser movimentadas por toda a tela.

Para simplificação do processo de elaboração de figuras, existem à venda softwares especializados.

MICROCOMPUTADOR	NÚMERO DE CORES	DETECÇÃO DE COLISÃO	PLANOS	TAMANHO DA FIGURA	SOFTWARE
Commodore 64	16	Sim	8	24×21	Sim
Atari 800	256	Sim	16	16×16	Sim
Texas TI99/4A	16	Sim	28	32×32	Sim
Sord M5	16	Sim	32	8×8	Não

quadriculado, no tamanho máximo, e produzir a forma desejada pelo preenchimento dos blocos apropriados. Já sabemos que os computadores operam pelo uso exclusivo dos dígitos 1 e 0: os quadros em preto são representados por dígitos 1 e os brancos por dígitos 0.



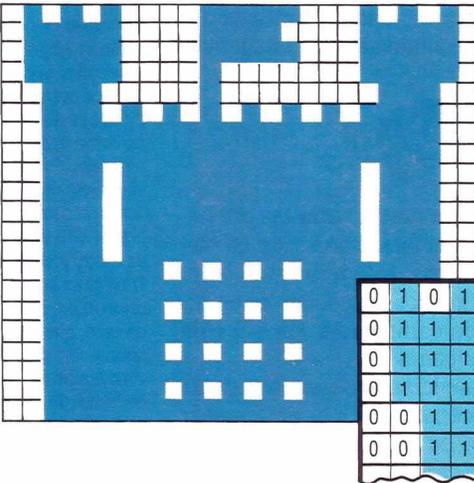
O computador controla a maior parte dos dados sob a forma de bytes (um conjunto de 8 bits). O manual de seu computador explicará como o quadriculado de pontos que constitui uma figura deve ser subdividido em grupos de oito. Cada byte, por sua vez, precisa ser convertido em um único número decimal, que varia de 0 a 255, para poder ser usado em um programa BASIC. Isto é feito multiplicando-se o dígito binário (bit) da extrema esquerda por 128, o seguinte por 64 e assim por diante. Os resultados são somados em seguida.

O conjunto resultante de números decimais define de modo completo o formato da figura. Esses nú-

meros são colocados em posições da memória por um programa em linguagem BASIC; o procedimento exato vai variar de acordo com o equipamento utilizado. É necessário, então, instruir o computador quanto às posições de memória em que são encontradas as especificações para cada uma das figuras desejadas.

Todo o resto é agora obtido usando-se comandos simples, que especificam a posição efetiva de cada figura na tela, alteram sua cor, ampliam ou reduzem o tamanho e descobrem quando ocorre uma sobreposição.

Existem pacotes de software chamados "utilitários", à venda para a maioria dos equipamentos que podem produzir gráficos com animação. Esses softwares tornam o processo de criação da imagem menos monótono. Apresentam o quadriculado na tela e possibilitam a elaboração da imagem pelo simples movimento do cursor luminoso pelo quadriculado. Todo o procedimento aritmético é controlado automaticamente e os resultados são então distribuídos em seus bytes correspondentes. Finalmente, o quadriculado desaparece e a figura sozinha surge na tela, pronta para utilização.



Nasce uma figura

O melhor modo de construir uma figura é começar com uma folha de papel quadriculado. O objeto é desenhado preenchendo-se diversos quadros.

Em outro quadriculado, os

quadros preenchidos são representados como dígitos 1 e os não preenchidos como dígitos 0 — as duas unidades com as quais os computadores operam. E, como a memória do computador é dividida em bytes (com 8 bits cada), todo o quadriculado é dividido em grupos de oito quadros.

Cada grupo deve ser convertido em um único número decimal. O bit da extrema esquerda será multiplicado por 128, o seguinte por 64 e assim sucessivamente. Os resultados somados devem fornecer uma resposta de 0 a 255, que é então utilizada no programa em BASIC.

0	1	0	1	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0	1	0	1	0
0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	0
0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0
0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	0
0	0	1	1																				
0	0	1	1																				

0	0	1	1	1	1	1	0
128	64	32	16	8	4	2	1
↓	↓	↓	↓	↓	↓	↓	↓
0	0	32	16	8	4	2	0
							= 62



Steve Wozniak



De passatempo a um negócio multimilionário: conheça um pouco da história dos famosos microcomputadores Apple.

Nos Estados Unidos, Steve Wozniak é mais conhecido pelo público por sua participação em concertos de rock do que pela construção do Apple I e II. Mas no mundo dos computadores ele é tido como o “gênio da eletrônica”, pois fez pela simplificação e difusão dos microcomputadores mais do que qualquer pessoa. As máquinas de Wozniak foram as primeiras a ter cores, gráficos, um teclado e um vídeo. O Apple II tornou-se tão popular que mais de 1 milhão de aparelhos já foram vendidos.

A ascensão meteórica de Steve Wozniak, de técnico numa oficina a bilionário da computação, parece um moderno conto de fadas. Ele nasceu e cresceu no agora famoso Vale do Silício, na Califórnia. Seu pai era engenheiro e lhe transmitiu os conhecimentos básicos de eletrônica.

Wozniak passou os primeiros anos de vida brincando com peças eletrônicas e usando sua habilidade técnica para pregar sustos nos colegas de escola. Uma vez ele construiu um dispositivo eletrônico chamado caixa azul, a partir da idéia que leu em uma revista. O dispositivo era capaz de imitar certos sons do sistema telefônico. Tais sons significavam que uma quantidade certa de fichas havia sido colocada no telefone público. Com isso, Wozniak fazia ligações telefônicas para qualquer lugar do mundo sem pagar. Certa vez ligou até para o papa.

Wozniak não chegou a fazer um curso de enge-

nharia. Suas notas de matemática e eletrônica eram excelentes, mas no colegial ele desistiu de estudar. Seu primeiro emprego foi como técnico na empresa Hewlett Packard, onde projetava calculadoras. Lá, todos diziam que Steve não tinha as condições necessárias para fazer o que ele realmente queria: projetar computadores.

Assim, ele começou a trabalhar sozinho, em geral à noite, e desenhou um microcomputador que a Hewlett Packard rejeitou. Chateado, Wozniak abandonou a empresa, e começou a trabalhar com o colega de escola, Steve Jobs. O projeto que fizeram vendeu cinquenta exemplares. O Apple I tinha nascido. A pequena companhia e o projeto receberam o nome de Apple, simplesmente porque Jobs já tinha trabalhado num pomar.



Entre 1975 e 1976, Wozniak fechou-se em sua oficina. Nela, desenvolvendo seu trabalho criativo dia e noite, produziu o Apple II. Tinha nessa época 26 anos de idade.

Os especialistas ainda vêem o Apple como uma façanha surpreendente de projeto e circuitos muito simples. Eles dizem que Steve Wozniak lê diagramas de circuito de um chip com a mesma facilidade com que as pessoas lêem uma história em quadradinhos. Uma das inovações mais importantes do Apple II foi a simplificação da unidade de disco. Antes de Wozniak, a unidade precisava de trinta chips, e seu novo design fez com que apenas cinco fossem necessários. Isso não foi algo totalmente inovador, mas, simplificando todas as peças e reunindo-as num só bloco, possibilitou a qualquer pessoa ter em casa um computador.

O projeto comercial

O comércio, entretanto, não era a meta principal de Wozniak. Seu companheiro Steve Jobs foi o responsável pela comercialização do Apple e pela consolidação da Apple Corporation (em 1984 contava com 3.300 empregados); a companhia, segundo consta, contribuiu para que cinquenta pessoas se tornassem milionárias. Wozniak possui apenas 4% da Apple e nunca se envolveu em assuntos administrativos. Ele ainda prefere divertir-se com seus micros e desenvolver novas idéias.

Em 1983, depois de estar apenas empenhado na organização de festivais de rock, Wozniak voltou à Apple para trabalhar em outros projetos, aparentemente novas propostas de software. Além disso, ele está envolvido na futura produção do Apple II com vídeos e sistemas de edição de alta qualidade gráfica. Mas Wozniak não se preocupa apenas com a criação de máquinas melhores e mais rápidas. Ele também acredita ser possível desenhar um microcomputador muito inteligente. Mediante um programa simples, tal micro seria capaz de aprender qualquer coisa. Vamos esperar a próxima invenção desse gênio da engenharia eletrônica.

O superintendente

Steve Jobs, superintendente da Apple Computer Inc., foi companheiro de escola de Wozniak. Se este era o gênio da eletrônica na turma, Jobs era o homem de negócios. Quando o Apple foi projetado em 1975, Jobs tinha 20 anos. Ele logo percebeu que a máquina tinha grande utilidade como diversão e também como computador pessoal. Wozniak criou o Apple, mas Jobs foi o responsável pela venda e produção em massa. O Apple II vinha numa caixa completa, pronto para ser usado em casa. Para conseguir o primeiro capital do empreendimento, Jobs vendeu seu carro, e Wozniak, sua calculadora programável. Logo depois o jovem milionário Mike Markkula financiou o projeto, que lhe pareceu muito interessante. Em 1980, o crescimento da empresa de computadores foi muito grande e em 1982 as vendas alcançaram mais de 500 milhões de dólares.

O crescimento fantástico da Apple foi alvo de prêmios nos Estados Unidos. Steve Jobs foi capa da revista *Time*, como um dos jovens mais bem-sucedidos do mundo.

Caneta mágica

A caneta óptica é um acessório que pode ser utilizado para desenhar ou fazer marcas diretamente na tela. Saiba como funciona este instrumento.

A “fobia do teclado” faz com que algumas pessoas evitem o computador, tanto em casa como no trabalho. Pelo fato de o teclado assemelhar-se ao de uma máquina de escrever e elas em geral não saberem datilografar, além de algumas teclas conterem sinais estranhos, o teclado pode parecer misterioso. A caneta óptica é uma solução para esse problema (juntamente com outros dispositivos, como a entrada de dados através da voz), embora tenha também muitas outras utilidades.

A caneta óptica é um instrumento cilíndrico bastante parecido com uma caneta (daí deriva seu nome), que tem um fio semelhante ao de um telefone em uma das extremidades. Nesse fio há um plugue que se encaixa na parte traseira do microcomputador. Quando a caneta óptica é apontada diretamente para a tela, o computador é capaz de detectar exatamente a posição apontada (em alguns sistemas, pressiona-se a caneta na tela, para ativar o interruptor interno da caneta).

O que ocorre, na realidade, é que a fotocélula na ponta da caneta responde com um sinal elétrico toda vez que passa por um dos pontos de luz que continuamente percorrem a tela formando as imagens. Os circuitos dentro do chip controlador de vídeo calculam o lugar exato onde se encontra o ponto de luz quando a caneta óptica emitiu o sinal.

Entretanto, a caneta óptica é usada mais comumente para escolher um dos itens mostrados na tela. Reconhecendo o ponto indicado pela caneta, o computador deduz o caractere ou símbolo a que o ponto se refere.

Muitos programas de aplicação utilizam “menus” como parte de sua operação. Trata-se de uma lista de opções mostradas na tela e, assim, o usuário faz a escolha. Num programa de finanças domésticas, por exemplo, essa lista pode ser:

- 1) Fazer pagamento
- 2) Examinar a conta bancária
- 3) Registrar um rendimento
- 4) Deduzir uma despesa
- 5) Verificar os estoques, e assim por diante

Normalmente, o usuário indicaria o curso de ação escolhido pressionando uma única tecla (1, 2, 3, 4, 5) ou digitando uma palavra de comando. Utilizando a caneta óptica, o usuário limita-se a apontar a opção desejada. O computador geralmente responde fazendo a opção escolhida piscar, indicando que a entrada foi aceita.

Alguns programas sofisticados são operados qua-

se totalmente por esses “menus” (chamados de “menu dirigido”); nesse caso, o usuário precisa tocar no teclado apenas quando um dado real é necessário, como o nome e o endereço de alguém.

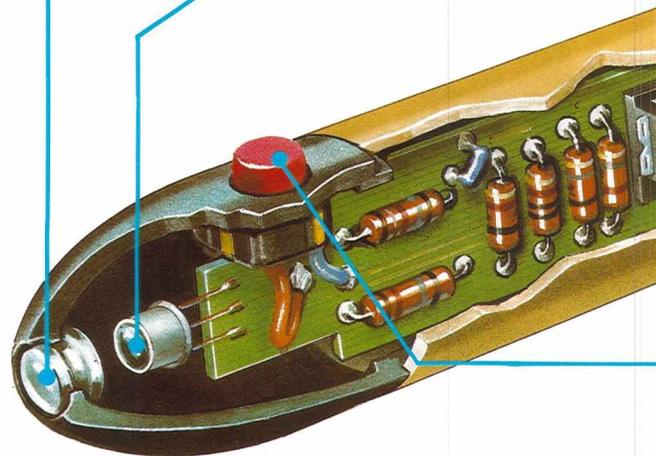
Rotinas especiais

Esse programa precisa ser escrito especialmente para trabalhar com a caneta óptica, pois ele é diferente do programa que utiliza o teclado. É necessário apenas escrever uma pequena rotina que verifique as coordenadas da posição da caneta óptica a partir do controlador de vídeo, e perceba que opção ocorre naquela posição da tela. Infelizmente, poucos fornecedores de software oferecem versões que trabalham com uma caneta óptica.

Entretanto, além de selecionar itens, a caneta óptica pode ser usada para criar imagens na tela. A maioria dos microcomputadores que utilizam a caneta óptica tem um software adequado para esse fim. O usuário recebe uma tela em branco para desenhar diagramas, figuras ou o que quiser e uma divisão separada (geralmente na parte inferior da tela) fornece uma série de funções especiais que ajudam na criação de imagens. Uma dessas funções será uma paleta de cores, que pode ser utilizada exatamente como a paleta de pintura a óleo. A caneta óptica é colocada na cor desejada pelo usuário, e para onde quer que ela se mova na tela principal deixará uma linha nessa cor.

Lente

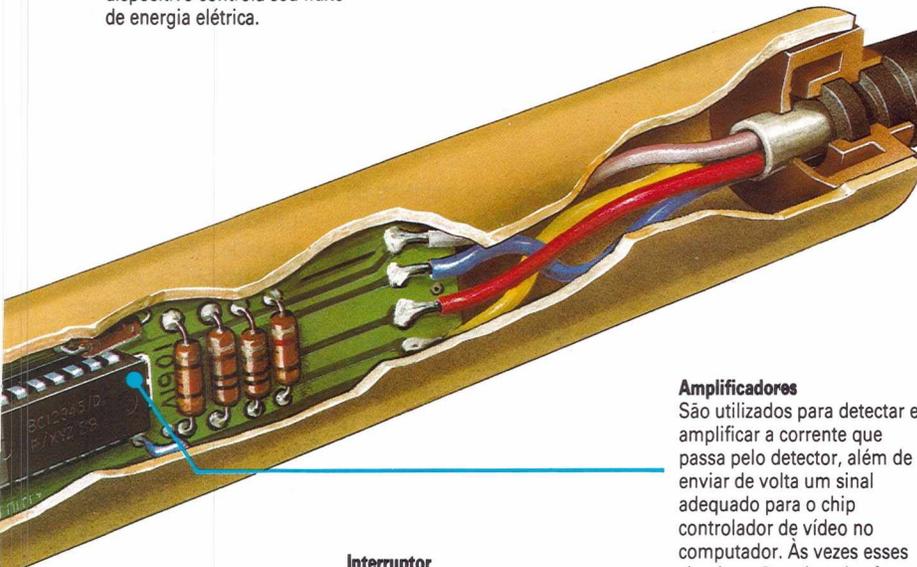
A quantidade de luz emitida por um único pixel ativado é tão pequena que o uso de lente é necessário para que essa luz se concentre na superfície da fotocélula.





Fotocélula

Dispositivo semicondutor que, grosso modo, é como um transistor ou um diodo com a extremidade ressaltada. A luz que vai para dentro desse dispositivo controla seu fluxo de energia elétrica.



Amplificadores

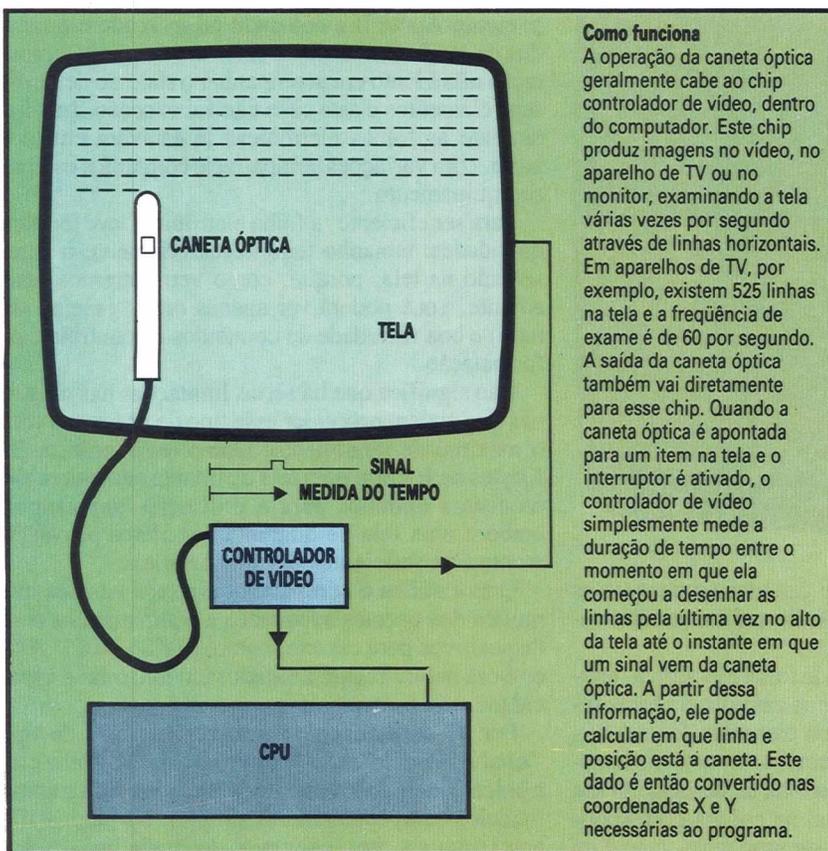
São utilizados para detectar e amplificar a corrente que passa pelo detector, além de enviar de volta um sinal adequado para o chip controlador de vídeo no computador. Às vezes esses circuitos são colocados fora da caneta.

Interruptor

A maioria das canetas ópticas possui algum tipo de interruptor, que pode ser controlado manualmente ou por pressão da caneta na tela. O interruptor é necessário para que a caneta óptica não reaja à luz ambiente, quando não está sendo usada para selecionar um item na tela.

Fio condutor

Um cabo especial parecido com o de um telefone é ligado à parte traseira do computador e, daí, ativa o chip controlador de vídeo.



Como funciona

A operação da caneta óptica geralmente cabe ao chip controlador de vídeo, dentro do computador. Este chip produz imagens no vídeo, no aparelho de TV ou no monitor, examinando a tela várias vezes por segundo através de linhas horizontais. Em aparelhos de TV, por exemplo, existem 525 linhas na tela e a frequência de exame é de 60 por segundo. A saída da caneta óptica também vai diretamente para esse chip. Quando a caneta óptica é apontada para um item na tela e o interruptor é ativado, o controlador de vídeo simplesmente mede a duração de tempo entre o momento em que ela começou a desenhar as linhas pela última vez no alto da tela até o instante em que um sinal vem da caneta óptica. A partir dessa informação, ele pode calcular em que linha e posição está a caneta. Este dado é então convertido nas coordenadas X e Y necessárias ao programa.

O usuário pode ainda escolher linhas diferentes, em largura e textura, na parte inferior da tela, e pode também desenhar círculos e quadrados. Resumindo, tudo que foi exposto em "O micro: um artista" (ver p.34) é realizado pela caneta óptica, sem dúvida, com maior rapidez e facilidade.

Outras aplicações

Os jogos que utilizam a caneta óptica também começam a aparecer no mercado e se tornam bem mais fáceis com essa caneta. Isso permite a criação de outros mais difíceis, como o xadrez — basta que você aponte para onde vai mover a peça e o computador se encarrega do resto.

Engenheiros, designers e projetistas são, talvez, os que mais utilizam a caneta óptica. Os sistemas de Computer Aided Design (CAD) são muito utilizados, por exemplo, em propaganda de carros novos; funcionam como qualquer outro sistema de computador, mas com um software especializado e gráficos de alta qualidade. Se um sistema CAD é usado para projetar um novo equipamento eletrônico, a tela apresentará todos os componentes disponíveis, bastando que o projetista escolha o adequado e coloque-o no lugar certo da tela.

A caneta óptica, pelas suas numerosas aplicações, é um dos melhores exemplos de um acessório de computador divertido e, ao mesmo tempo, de grande valor prático.

Faça suas previsões

As "folhas eletrônicas" são úteis em muitas situações. Elas podem ser empregadas, por exemplo, para elaborar planejamentos, orçamentos ou estimativas de custo.

Campos de dados

A folha eletrônica está dividida em linhas e colunas e a interseção de linha e coluna é chamada "campo" ou "célula". Cada célula pode conter um título (por exemplo, JANEIRO), um número (por exemplo, 149,89) ou uma fórmula. O campo N2 contém a fórmula "SUM (A2:M2)", que representa a soma dos números da linha superior, de janeiro a dezembro. O resultado desse cálculo é exibido: 388,4. Observe que a folha foi dividida em duas janelas, de modo que os meses de março a dezembro não estão visíveis na tela.

Calcula-se que gerentes de empresas gastam cerca de 30% de seu tempo preparando orçamentos — uma atividade que sempre exige respostas a muitas questões do tipo "o que aconteceria se...". Tradicionalmente é utilizada para esse trabalho uma folha de papel dupla, do tamanho destas duas páginas. Essa folha é dividida em doze colunas ou mais, cada qual com o nome ou número de um mês e com todas as despesas relacionadas de um lado. Em cada coluna, as despesas de várias categorias são especificadas. Somadas as colunas, obtém-se o total de despesas por mês e somam-se os resultados para determinar o total durante o ano, que é apresentado na coluna correspondente ao ano inteiro.

Surgem inconvenientes quando se planeja gastar muito acima — ou, pior ainda, muito abaixo — das despesas efetivas, sendo necessário, então, retornar

admirar que os pacotes de folha eletrônica sejam o software de maior venda no mundo.

Assim como a maioria das unidades de software comercial, os programas de folha eletrônica são desenvolvidos em "overlays", o que significa que nem todos os programas estão efetivamente presentes no equipamento durante todo o processo. Se você imaginar o programa dividido em sub-rotinas (ver p. 77), a sub-rotina não necessária à operação em andamento não será chamada do armazenamento auxiliar (disco, ou fita). Quando isto ocorrer, o sistema operacional sobreporá a nova sub-rotina à anterior (daí o nome overlay: over = sobre, lay = colocar). Como você pode imaginar, esse método de ampliação da memória disponível é muito útil, mas também significa que muitas vezes será necessário esperar pela transferência dos dados — da memória auxiliar de armazenamento de dados para a principal.

Os pacotes de folha eletrônica — que você facilmente reconhecerá pelo nome, com frequência terminado em "calo" — existem em versões compatíveis com uma grande variedade de microcomputadores para uso doméstico e empresarial. O mais vendido é o Visicalc, originalmente escrito para o equipamento Apple II e colocado no mercado em meados de 1979. O mundo do software de microcomputação não é lento em acompanhar o sucesso de um de seus elementos e este caso não foi exceção. Imediatamente as folhas eletrônicas já estavam em toda parte, com variações compatíveis com todos os tipos de equipamento.

Para ser eficiente, a folha eletrônica deve ter duas qualidades: tamanho (não necessariamente o apresentado na tela, porque, como verificaremos mais adiante, você poderá ver apenas uma "janela" do todo) e boa variedade de comandos de controle e de formatação.

Isso significa que há sérias limitações nas máquinas que podem processar esse tipo de programa com o máximo de rendimento. Como regra prática, 32 Kbytes de RAM e uma tela de oitenta caracteres são requisitos mínimos para a utilização empresarial, embora uma tela de quarenta caracteres provavelmente seja suficiente para fins pessoais.

Quem utiliza o computador em casa verifica que muitos dos pacotes adequados a equipamentos com dispositivos para cassete, como o TK85 ou CP 200, embora naturalmente limitados em tamanho e capacidade, são bastante úteis.

Por ter capacidade de resolver questões do tipo "qual o impacto de...", os programas de folha eletrônica podem, obviamente, ser usados para montar modelos computadorizados simples. Na página 101 apresentamos um exemplo do trabalho de um

JANEIRO	FEVEREIRO	TOTAIS
42.41	18.75	388.4
160.35	149.89	1732.7

FORM: SUM (A2:M2)

A linha inferior

O cursor é o bloco retangular que em geral ocupa o campo N2. O caractere digitado aparecerá no campo em que o cursor está posicionado. Os conteúdos totais desse campo serão igualmente apresentados na linha de comando do programa que, neste caso, se encontra na parte inferior da página.

e alterar diversas parcelas e calcular novamente os totais de linhas e de colunas, de acordo com os novos dados.

Utilizando-se programas de folha eletrônica, tem-se a possibilidade de refazer os cálculos da folha inteira, toda vez que um único elemento for modificado. Alterando-se, por exemplo, o custo de transportes em janeiro, altera-se o total de despesas nesse mês, bem como o total geral na coluna correspondente — tudo isso ao toque de uma tecla. Não é de



analista de sistemas e, caso você venha a usar um pacote de folha eletrônica, logo se tornará evidente a necessidade de um planejamento cuidadoso semelhante. Os programas de banco de dados, conforme já verificamos, consistem em quantidades maciças de dados, que são organizados de acordo com as exigências do usuário. Os programas processadores de palavras, outro software de grande vendagem, são projetados para permitir o deslocamento de palavras isoladas ou blocos inteiros de texto, ao serem processados. Porém, os programas de folha eletrônica são um pouco diferentes porque exigem efetivamente que o usuário realize um processo de planejamento.

Por exemplo, se você utilizar um programa de folha eletrônica para analisar suas despesas gerais, poderá agrupar prestações da casa ou hipoteca, impostos, seguros etc. e depois utilizar o resultado do cálculo na tabela maior, incluída na mesma folha. Você deve tomar o cuidado de somar primeiramente todas as despesas domésticas antes de transportar qualquer número para a tabela maior.

Cada posição individual na folha, chamada célula, é endereçada e localizada através de suas coordenadas X e Y. Horizontalmente são usadas as letras de A a Z, de AA a AZ e, em alguns casos, de BA a BM, o que permite dar conta da extensão total possível da folha — 65 colunas nas versões mais populares. Verticalmente, podem ser usados os números de 0 a 256. Cada célula pode ter um título (como “Vendas” ou “Lucros”), um valor, fornecido ou derivado de um cálculo, ou a fórmula desse cálculo, como $B4+B6*B5$. Muitas vezes, as fórmulas ultrapassam o tamanho do quadro visto na tela, e são, de modo geral, apresentadas em uma linha separada na parte superior da tela. Ao iniciar um novo programa, o tamanho da célula será preestabelecido, talvez em oito ou nove dígitos ou caracteres. Este é o chamado tamanho default. Geralmente, você terá a possibilidade de reduzir ou aumentar as células; para adaptá-las ao tipo de cálculo que vai realizar. Alguns pacotes permitirão que a coluna da extrema esquerda (títulos ou itens que você mesmo escolhe) seja maior que as demais. E não é necessário decidir de imediato o tamanho desejado. A maioria das versões admite ampliação ou redução, mesmo quando já houver dados nelas incluídos. Caso venha a reduzir para um tamanho menor que o do conteúdo, a parte não apresentada na tela não será apagada, apenas não será mostrada.

O último componente importante na folha é a linha de comando, que se apresenta na parte superior

ou inferior da tela em resposta à tecla de “comando”:/, por exemplo. Esses comandos são destinados ao uso em formatação e ao controle da disposição das unidades, e não dos dados, embora possam afetar o modo pelo qual estes aparecem. A maioria das versões populares desse software possibilita a realização de grande variedade de operações com banco de dados. Você pode, por exemplo, apagar, deslocar ou copiar linhas ou colunas inteiras; ou dividir a janela de modo que exiba conjuntamente as partes da folha que em geral estão longe demais para que a janela as inclua ao mesmo tempo; ou, ainda, deslocar essas folhas separadamente por toda a tela.

Geralmente, o deslocamento entre as células é realizado pela tecla de controle do cursor, mas há ainda outra tecla de comando que permite saltar para uma célula específica. Os procedimentos de armazenamento e retenção, limpeza e proteção são realizados por teclas de comando e, já que estamos falando nisso, salientamos novamente a importância de sempre guardar os próprios programas. A montagem de uma folha eletrônica toma um longo tempo, maior do que para fornecer os dados. Como regra geral, sempre retenha a folha, antes de começar a alimentá-la de informações. E, se cometer algum erro grave, você terá minimizado as perdas.

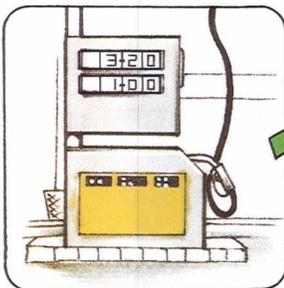
Os resultados são enviados à impressora por uma tecla de comando, e deve-se tomar cuidado em definir qual a parte da folha que se deseja imprimir, usando parâmetros. Assim como a tela é semelhante a uma janela, ou porção da folha inteira, assim também, evidentemente, será a página da impressora. Se você necessitar imprimir uma folha maior do que a da impressora, o procedimento recomendado é fazer duas impressões e, em seguida, colar as páginas de modo conveniente.

O uso de janelas, como observamos, permite que duas partes da folha sejam exibidas ao mesmo tempo. Uma folha com janela dividida também pode ser impressa. Isso é especialmente útil quando se fornecem dados, porque se pode fazer referência a itens anteriormente incluídos. A maioria dos pacotes de folha eletrônica permite ao usuário “manter” a parte superior ou as primeiras linhas, o que é conveniente pelas mesmas razões, pois estas costumam conter os títulos.

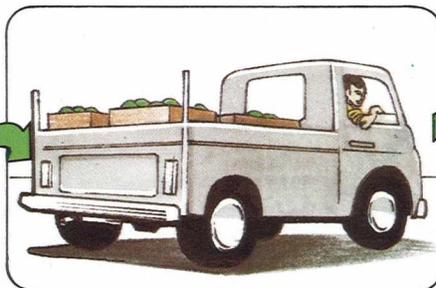
Até agora, consideramos a folha como uma tabela acessada unicamente de modo serial (um item após o outro, nas linhas e nas colunas); se quisermos, porém, abandonar a comodidade de efetuar as somas pela disposição em linhas e colunas, pode-se

E se...?

Se o conteúdo de um campo for alterado, o programa de folha eletrônica automaticamente refaz os cálculos dos outros campos que de algum modo dependem desse número. A rapidez e facilidade desse processo incentiva o usuário a testar suas estimativas e planejamentos, para verificar o que acontecerá, digamos, ao lucro global, caso certas condições se alterem. Veja, por exemplo, o vendedor de frutas...



Se o preço da gasolina aumentar X%...



Os custos mensais do transporte subirão Y%...



Isso acarretará o aumento do preço dos produtos por atacado...



Que será transferido ao consumidor, com preço final mais alto...

dar à folha qualquer outra disposição, se isso facilitar a linha de pensamento do usuário. Todavia, tenha em mente que essa modificação no interior do banco de dados exigirá um trabalho de análise de sistemas ainda mais completo.

As folhas eletrônicas com finalidade empresarial, como o Visicalc e o Supercalc, proporcionam ao usuário a possibilidade de transferir dados para os pacotes processadores de palavras ou de banco de dados, e existem muitos programas complementares que possibilitam a saída sob várias formas gráficas: gráficos em curvas ou barras, por exemplo.

Já tratamos de uma das utilizações da folha eletrônica: a elaboração de orçamentos e a análise de despesas domésticas. Outra aplicação muito interessante é a instalação de um sistema central de refrigeração, em que grande número de variáveis deve ser levado em consideração: energia a ser consumida, a quantidade e o tipo de equipamento, a área a ser refrigerada etc. De fato, qualquer tomada de decisões é consideravelmente auxiliada por um programa de folha eletrônica, pois o usuário é quase forçado a levar em conta qualquer possibilidade.

Talvez o aspecto mais notável do programa de folha eletrônica com finalidades empresariais, processado em equipamento adequado, seja a rapidez da operação. Esta é função direta da programação em código de máquina e não é surpreendente que a velocidade de um pacote desenvolvido em linguagem BASIC para alguns pequenos microcomputadores seja bem menor.

Será interessante examinar alguns problemas que poderíamos encontrar se tentássemos desenvolver tais programas em BASIC, apenas para dar uma idéia da complexidade da tarefa.

Para começar, cada célula deve ser definida de três modos: deve ser capaz de conter dados de "variáveis alfanuméricas", como "janeiro" ou "impostos"; deve ser capaz de conter dados numéricos para utilização em operações aritméticas; por exemplo, o total de impostos pagos em janeiro; e também deve poder conter fórmulas, que são basicamente linhas em código de programação, como "impostos anuais 12", para se obter a média mensal. Por isso, cada célula deve ter seu tamanho expandido ou reduzido, sem perder a menor parte de informação significativa; por esta razão, todas devem ser duplicadas: uma aparecerá na tela e outra, oculta, conterá todos os dados.

Como você vê, a simples manipulação dos dados é uma tarefa complexa, e lembre-se de que os mais sofisticados pacotes podem ter 16.000 células individuais! As técnicas usadas para desenvolver tal software são muito semelhantes às de desenvolvimento de interpretadores para linguagens, como BASIC e FORTH. Técnicas semelhantes são também utilizadas nos softwares de banco de dados.

Tudo isso leva à explicação do alto preço de softwares desenvolvidos com finalidades empresariais. Pacotes como o Visicalc ou o Supercalc podem ser considerados caros, mas deve-se ter em mente a economia que seu uso proporciona. O exemplo relativamente simples que examinamos anteriormente, da utilização desse tipo de software em substituição ao trabalho do gerente na elaboração de orçamentos, pode levar a uma economia de 15 a 20% ao ano. O custo do software será, então, insignificante, comparado a tal economia. Na verdade, o Visicalc talvez tenha sido o melhor amigo dos vendedores dos equipamentos da linha Apple.

Janelas do mundo

Podemos mover o cursor pela tela, para a direita ou a esquerda, para cima ou para baixo, pelo controle do teclado; assim também, em um programa de folha eletrônica movemos a tela por toda a folha. Isto permite que uma área muito maior do que a da tela seja examinada.

	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO
1: TÍTULOS								
2: ALUGUEL	180.00	180.00	180.00	192.00	192.00	192.00	192.00	192.00
3: IMPOSTO	42.00	42.00	42.00	51.00	51.00	51.00	51.00	51.00
4: SEGURO	.00	.00	.00	.00	.00	.00	.00	.00
5: TOTAL	-----							
6: DESPESA	222.00							243.00
7: TRANSP.	2.00							.00
8: REFEI.	6.00							.00
9: ROUPAS	14.00							.00
10: LIVROS	.00							.00
11: TOTAL	-----							
12: ESCOLA	22.00							.00
13: COMBUS.	42.00							.00
14: MANUT.	.00							92.00
15: OFICINA	26.00							.00
16: IMPOSTO	.00							18.00
17: SEGURO	.00							.00
18: TRU	100.00							.00
19: TOTAL	-----							
20: CARRO	68.00							110.00
21: PEDÁGIO	.00							.00
22: AVIÃO	190.00							74.00
23: HOTEL	86.00							340.00
24: REFEI.	22.00							190.00
25: DIVER.	12.00							38.00
26: TOTAL	-----							
27: LAZER	378.00							752.00





Quando o herói é você

Os jogos eletrônicos de aventura estão cada vez mais difundidos. Neles você não é só um espectador, mas um participante ativo.

Aventura: uma palavra que faz recordar um livro, um filme, uma viagem ou, talvez, uma experiência pessoal. Para muitas pessoas, usuárias de computadores, aventura também lembra um determinado tipo de jogo eletrônico.

Para caracterizar melhor o que são as aventuras de computadores, vamos compará-las a um livro. Quando você lê uma história de aventuras, pode empolgar-se com os perigos, os mistérios e uma série de episódios interessantes — mas, na verdade, tudo isso é “vivido” por outra pessoa. Numa aventura de computador, você não fica de fora da ação, mas participa dela! Como o herói ou heróis da história, você se liga nos acontecimentos, vive diretamente as experiências.

Num livro, o leitor não pode influenciar o curso do enredo. A ordem e o desfecho da trama são sempre iguais, não importa quantas leituras sejam feitas. Numa aventura de computador, suas decisões, julgamentos e ações é que determinam o desdobramento e a modificação de uma trama. Pode haver qualquer número de variações para a ordem dos acontecimentos, além de muitos finais diferentes, alguns alegres, outros tristes. O que existe de emocionante nos jogos de aventura é o fato de você participar da ação, apesar de estar sentado confortavelmente em sua casa.

Cada aventura ocorre num ambiente diferente, que pode ser um mundo subterrâneo, um labirinto, uma cidade fantasma, um outro planeta, uma terra encantada; também pode acontecer em qualquer época: passado, presente ou futuro.

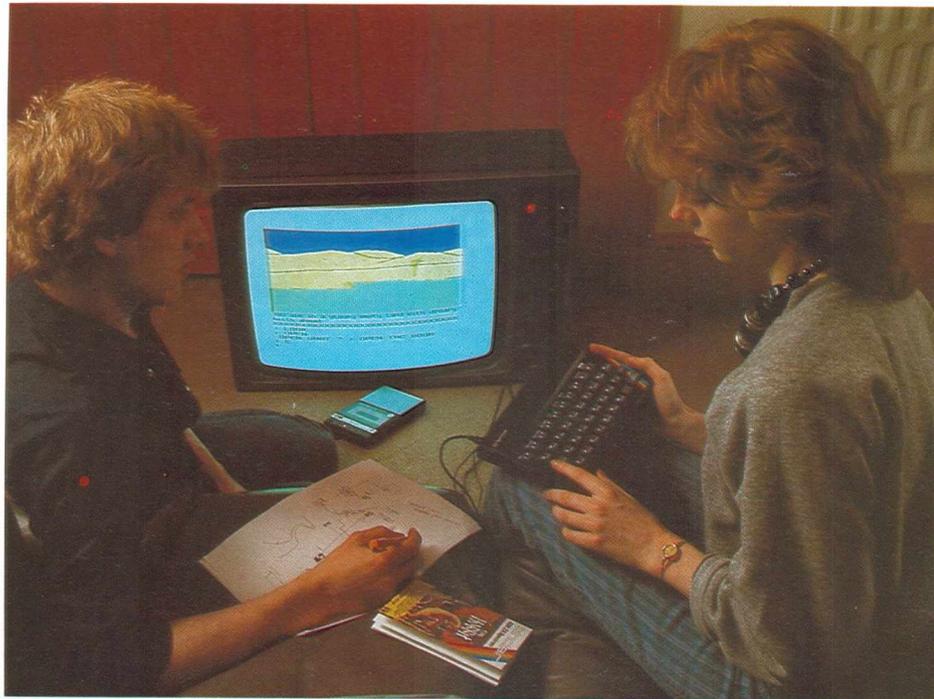
Geralmente, a aventura tem um tema constante que inclui um objetivo final. Por exemplo, você deve fugir de um planeta estranho, ou encontrar e destruir um feiticeiro malvado, salvar a princesa, pegar um tesouro, descobrir o motivo de um crime e prender o culpado.

Incidentes interessantes como esses fazem com que a aventura se torne um quebra-cabeça agradável e o enigma seja uma parte constante do jogo. Com frequência, tem de ser resolvido antes que a confusão aumente. Há também as situações imprevistas; por exemplo, depois de atravessar um desfiladeiro para agarrar a figura estranha que o observa há muito tempo, você descobre que era apenas um enorme espelho que refletia sua própria imagem.

No decorrer do jogo surgem pistas ocasionais que ajudam mas não são essenciais para a solução do mistério — a descoberta de uma passagem secreta pode levar a um lugar muito perigoso. É uma questão de vida ou morte — você está totalmente perdido, sem comida e sem água.

Esses percalços podem ser decifrados usando o bom senso, o que não exige nenhuma experiência ou conhecimento especializado. Entretanto, o aventureiro deve ficar atento, pois sempre surge uma nova pista. Os elementos casuais, exceto em doses pequenas, não aparecem num programa de aventura bem elaborado.

No decorrer de um jogo, é provável que você encontre objetos, mensagens ou personagens aparentemente insignificantes. Entretanto, tenha em mente que tudo é relevante numa história de aventura; para



cada detalhe há um propósito, mesmo que seja apenas tirar você da pista certa.

O que significam várias garrafas quebradas? O que pensar ao ouvir uma voz cavernosa que diz BAH? Como alguém pode utilizar um monte de lama suja? Como um tapete pode ser pregado no chão se não existe chão? Esses enigmas são componentes essenciais do enredo da aventura. Ao encontrar um objeto pela primeira vez, não importa que seja estranho ou inútil, dificilmente você descobrirá logo sua importância; mas com certeza precisará dele até o final da história.

Muitas aventuras têm um pequeno labirinto onde cada sala ou lugar é descrito em termos idênticos. O único modo seguro de traçar uma trilha nesses labirintos é fazer como João e Maria, deixando obje-

Você vive o papel

Jogos de aventura como os de masmorras e abismos existem há muito tempo; mas o microcomputador levou-os a uma camada maior da população. Na Europa e nos Estados Unidos, os jogos de aventura tornaram-se tão populares como os fliperamas. Apesar de poderem ser jogadas por uma só pessoa, as aventuras são um divertimento para toda a família. O personagem fictício ou histórico é assumido pelo jogador, que viaja em busca de um tesouro ou de outro objetivo.

Leitura do mapa

Num jogo de aventura, o jogador ou o personagem que ele representa tem de agir dentro de um vasto terreno ou outro ambiente — uma rede de passagens subterrâneas é algo bastante comum. Alguns fabricantes fornecem sugestões num manual de instruções para os que não têm muita experiência nesse tipo de jogo, mas o melhor meio de obter as soluções é a própria experiência na aventura. Esses jogos podem levar semanas para terminar; sendo assim, é importante traçar um resumo dos lugares e obstáculos que você já ultrapassou.

tos pelo caminho que percorrer. Este método tornou-se tão conhecido que alguns programas já acrescentaram outros problemas, tal como a presença de um bandido que persegue o jogador e vai apagando os vestígios.

Em algumas aventuras, apesar de você ter de solucionar os enigmas e alcançar todos os objetivos para completar o jogo com sucesso, a ordem em que os mistérios são desvendados e as metas alcançadas não tem grande importância. Isto contrasta com aquele tipo de aventura onde há apenas uma trilha para a conclusão triunfante.

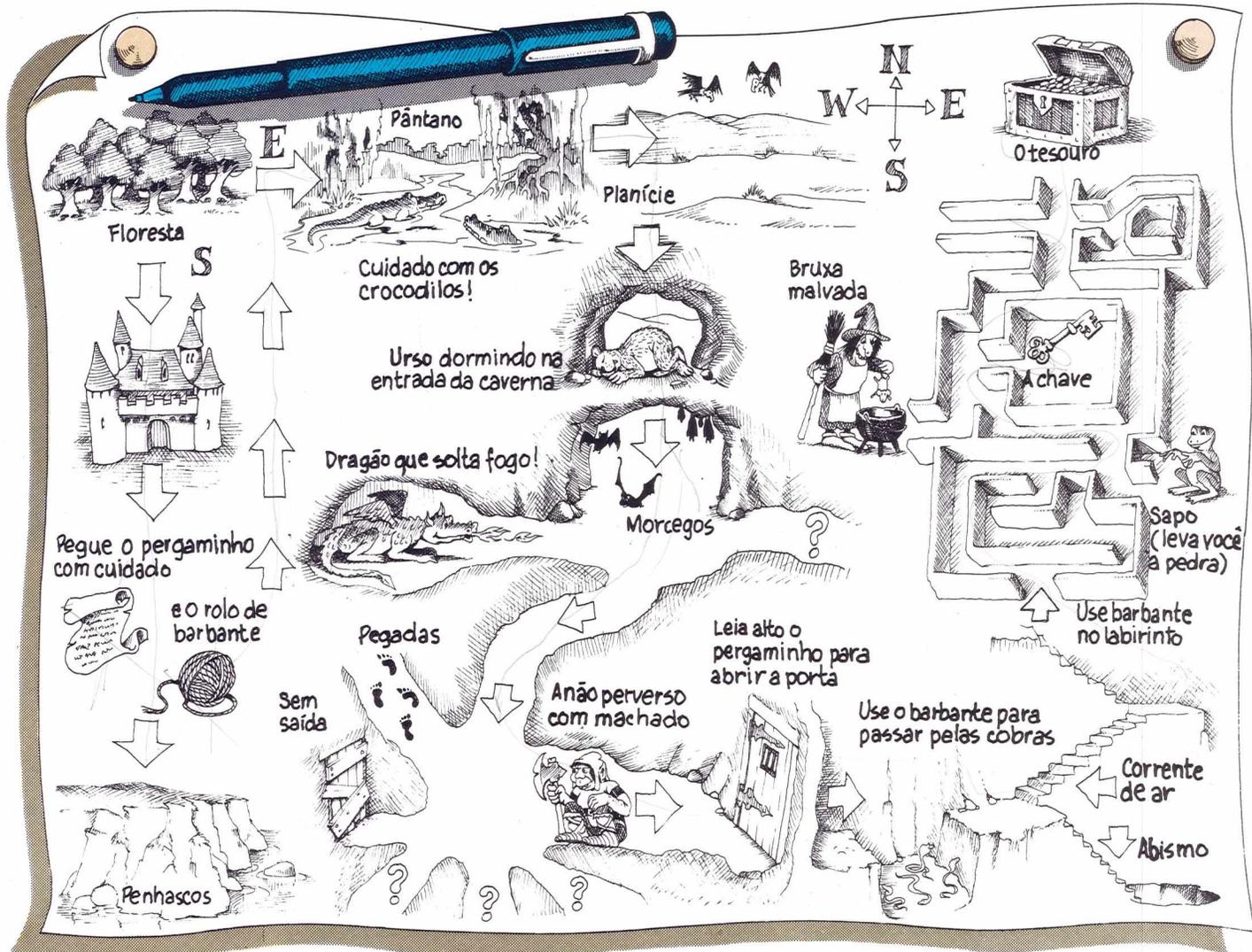
Uma boa aventura deve durar horas, talvez semanas de jogo, até que todos os segredos sejam revelados. Deve, ainda, possibilitar que você pare de jogar no ponto que quiser, no cassete ou no disco, e continue mais tarde. Isso também é bastante útil quando você atinge uma parte perigosa da aventura; por exemplo, indo corajosamente ao encontro de um exército, mesmo tendo apenas uma lanterna e um cantil. O jogador aventureiro mais prudente poupará o tempo de jogo antes de iniciar um diálogo com os soldados do exército. Então, caso o exército o convide para jantar, o jogo será recommençado e o aventureiro pode procurar um novo curso de ação; isto,

se ele não ficar bastante aborrecido por ser a refeição escolhida pelo inimigo.

A morte do aventureiro não significa necessariamente o fim do jogo. Alguns programas permitem que o perdedor “ressuscite”, em geral por meio de um sopro de fumaça amarela. Ao mesmo tempo provoca a perda de alguns pontos e posições conquistadas e coloca-o numa situação indesejável, digamos, num mundo inferior ou no meio de “lugar nenhum”.

Como você participa e se relaciona com o programa? Ele pode estabelecer uma comunicação direta com você, ou utilizar um “boneco”, um personagem controlado por comandos. O computador atua como intérprete e narrador de seus desejos. O jogador registra os comandos no teclado e as respostas do computador aparecem na tela.

Algumas aventuras mostram apenas textos na tela do computador, outras só apresentam gráficos, e há as que combinam os dois. Efeitos sonoros são utilizados nas aventuras elaboradas por meio de gráfico. Os jogos de aventura que só trazem um texto são como um livro de histórias não ilustrado; as palavras descrevem locais, objetos e acontecimentos. Os gráficos, na maioria dos jogos combinados, servem





para complementar as descrições textuais, e geralmente são constituídos de cenas estáticas de locais e objetos. Variam de uma simples linha desenhada até figuras bem detalhadas. Nas aventuras baseadas em gráficos, estes normalmente são usados em mapas estilizados, figuras de camadas da terra ou em representações de interiores de edifícios. Os personagens e os objetos aparecem sob a forma de símbolos ou figuras em miniatura. Neste caso, o jogador restringe-se a um pequeno número de comandos que, na maior parte, se realizam por meio de pressões no teclado do computador, a fim de mover e controlar o personagem.

O texto numa aventura geralmente abrange três elementos: onde você está, o que você pode ver e aonde pode ir. Por exemplo, o texto que aparece na tela pode ser: "Você se encontra numa floresta escura. O céu está encoberto por densa folhagem. Existe uma trilha que leva a leste e a oeste. Mais à frente, para o norte, há uma armadilha perigosa. Pode-se ver uma espada no chão: em volta dela, uma cobra verde". Você recebe uma descrição do que está mais próximo, algumas direções que pode seguir e os objetos que aparecem no caminho.

Os comandos quase sempre consistem em duas palavras: um verbo seguido de um substantivo, embora as aventuras mais sofisticadas compreendam frases inteiras. Alguns verbos são: PEGAR, SOLTAR, EMPURRAR, PUXAR, JOGAR, ACENDER, MATAR, COMER e BEBER, IR PARA O NORTE. Cada um deles pode especificar um movimento, apesar de algumas aventuras utilizarem apenas a abreviatura desses comandos — por exemplo, N para NORTE.

EXAMINAR é um verbo essencial — geralmente significa que o aventureiro deve buscar mais informações. EXAMINAR COBRA na ilustração (à esquerda) pode resultar numa mensagem como "É uma cobra de jardim", ou talvez "A cobra percebe seu movimento na direção dela e vai atacá-lo". Alguns verbos não precisam de substantivo.

A palavra INVENTORY (listagem) é empregada para informar o que você está carregando. Alguns objetos podem ser colocados dentro de outros — água em garrafa e machado numa sacola; enquanto outros devem ser usados diretamente no corpo — um anel, talvez, ou uma capa.

A palavra SCORE é utilizada para que o jogador saiba o quanto ele já progrediu em relação aos objetivos que deseja alcançar. A palavra HELP pode fazer com que você supere mais facilmente uma dificuldade, e, ao mesmo tempo, funciona como um incentivo para que continue tentando. Por vezes, como resposta a um comando, o aventureiro pode receber um "Não faça isso ainda!", o que mostra que a combinação de um verbo com um substantivo dará algum resultado, mas não naquele momento nem, talvez, naquele local. Grande parte do desafio está em descobrir os verbos e os substantivos que são importantes para a aventura. Palavras e combinações que a aventura não reconhece recebem uma resposta do tipo "Eu não compreendi você".

A maior parte dos produtores de aventuras fornece folhas que contêm dicas para aqueles que se confundem no jogo e também incluem algumas sugestões sutis.

Algumas aventuras são muito longas para a me-



Caça ao tesouro

O Hobbit é uma aventura gráfica para o Sinclair Spectrum e leva você à história de J. R. R. Tolkien. Uma cópia dessa história acompanha a fita cassete. Você é Bilbo, e faz uma viagem ao centro da Terra, onde encontra muitos dos personagens e acontecimentos do livro, enquanto procura o dragão e o tesouro.

mória do computador. Para que isso não represente um problema, elas são muitas vezes fornecidas em discos, onde o programa principal é carregado na memória no início, e partes do texto ou dos gráficos são colocadas ou tiradas do disco, de acordo com a necessidade.

Mas, graças às técnicas de condensação de grandes textos em memórias restritas e ao uso da programação em código de máquina, uma aventura longa pode ser armazenada na memória do computador, e, conseqüentemente, gravada numa fita cassete ou num disco flexível.

Existem aventuras para quase todos os tipos de micro. E qualquer uma delas proporcionará uma experiência agradável e será até mesmo o começo de um passatempo habitual. Por isso teste seu raciocínio e divirta-se. Feliz aventura!



Na pista certa

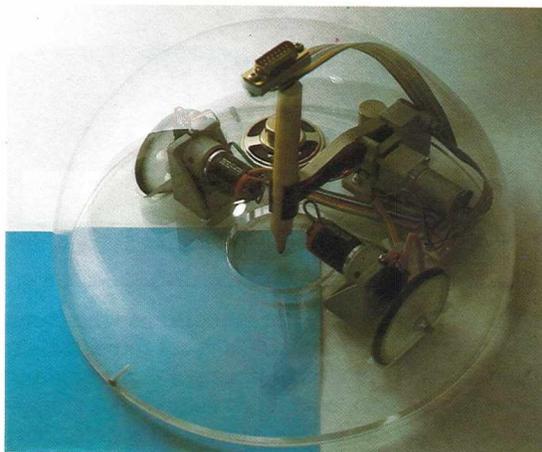
Deadline é uma variação dos temas de aventura. Você faz o papel de um detetive que vai solucionar o mistério de um assassinato. Seu arquivo contém o resultado de uma autópsia, algumas pílulas encontradas perto do cadáver e outras anotações sobre o caso. O jogo não ocorre num só dia (o espaço de algumas semanas é normal), mas cada nova ação, tal como movimentar-se de um local para outro, ou a entrevista com um novo suspeito, faz você perder minutos reais do seu prazo limite de doze horas.

Os micromundos

Muitos programas didáticos equivalem a meros livros de textos eletrônicos. Já a linguagem LOGO utiliza o computador para criar um novo “ambiente” de ensino.

Tartaruga ambulante

A tartaruga foi projetada para ser um auxílio ao raciocínio — especialmente no aprendizado da mais elementar geometria e das relações de espaço. Quando as crianças se sentem inseguras por não saber exatamente como comandar a tartaruga numa determinada manobra, elas tendem a seguir as instruções do LOGO, e andam como se fossem a tartaruga. Assim, entendem os comandos e fazem do aprendizado uma experiência real.



Desde que o micro apareceu em 1977, os educadores mais avançados logo perceberam seu potencial de ensino nas escolas. Há escolas que têm, atualmente, pelo menos um micro, e outras oferecem cursos de computação como parte do currículo. Entretanto, os micros registravam poucos avanços em relação aos métodos de ensino tradicionais.

Esta afirmação baseia-se na série de programas educacionais normalmente disponíveis para micro-computadores que demonstram a total falta de imaginação existente nesse sentido. Esses programas podem ser descritos como “livros eletrônicos”, onde o computador apresenta ao aluno uma série de “formas” na tela (equivalentes a páginas de um livro) e, a seguir, testa se a informação foi assimilada, utilizando uma série de questões de múltipla escolha, que o computador marca automaticamente.

Esses conjuntos são facilmente escritos num computador e apresentam gráficos coloridos, às vezes figuras animadas, como complemento do texto. Este processo é, no entanto, apenas a automatização de um processo já existente e não utiliza o poder do microcomputador de maneira inovadora.

A linguagem LOGO é diferente. De acordo com o trabalho do professor Seymour Papert, do Instituto de Tecnologia de Massachusetts, LOGO define-se como “uma filosofia educacional e um grupo de linguagens de programação de computadores feito para ajudar a complementar essa filosofia”.

Muitas pessoas se enganam ao pensar que LOGO é simplesmente uma linguagem de programação e, ao comparar seus comandos e sua estrutura com a linguagem BASIC, concluem que LOGO é a melhor linguagem para iniciantes. Acontece que não é esse, em absoluto, seu objetivo. Papert nunca pretendeu destinar seu método ao ensino de computação. Ele o concebeu como um meio de as crianças compreenderem várias matérias; um recurso pelo qual elas realmente aprendessem a aprender.

Grande parte dessa filosofia deriva do eminente filósofo e educador suíço Jean Piaget: quando a criança encontra recursos e ambientes corretos, ela aprende qualquer coisa sozinha, do mesmo modo como aprende a andar e a falar. Entretanto, o trabalho realizado por Piaget foi sobretudo teórico; Papert empenhou-se na produção de um meio prático de ensino piagetiano.

Os métodos de educação tradicionais não foram bem-sucedidos; isto se evidencia no fato de que muitos adultos têm medo de aprender, não gostam da idéia de adquirir novas habilidades ou entrar em novas áreas de conhecimento. O mais comum sintoma disso, diz Papert no livro *Mindstorms — Children, Computers, and Powerful Ideas*, é as pessoas terem aversão pela matemática; a isso ele atribui o nome de “mathophobia”.

Uma das razões para esse tipo de comportamento é que a maioria das matérias é ensinada do mesmo modo, ao passo que suas aplicações são completamente diferentes. As crianças aprendem, por exemplo, a multiplicar do mesmo modo que aprendem as capitais dos países ou a análise sintática, e tudo é feito de modo extremamente automático. O processo de ensino encontra-se muito distante daquilo que está sendo aprendido, quando, na verdade, os dois processos deveriam ser inseparáveis.

O próprio Papert vê o aprendizado de uma nova habilidade como uma distração, não importa que o assunto seja culinária, vôo livre ou uma língua estrangeira. Ele relaciona essa atitude à sua própria infância, quando descobriu como funcionavam as rodas de engrenagens, e aplica esse conceito toda vez que enfrenta um problema novo. Albert Einstein também afirmava que, ao encontrar algo que não entendia, recorria aos conceitos que aprendera antes dos cinco anos de idade.

A linguagem LOGO

Essas idéias interessantes foram incorporadas ao LOGO, como se pode ver no exemplo do nosso LOGO em ação. A primeira atração importante do LOGO é a “tartaruga”, projetada como um “auxílio ao raciocínio”, do mesmo modo como Papert brincou com engrenagens quando pequeno. Para as crianças, a tartaruga assume a forma de um robô (ver p. 176) que é ligado ao micro e se move no chão, por comandos do LOGO. A tartaruga leva, geralmente, uma pena com tinta para desenhar linhas no piso (ou numa folha grande de cartolina) e pode também ter um pequeno alto-falante e detectores de colisão — livrando-se desse modo dos obstáculos.

As crianças usam as tartarugas de solo e as tartarugas de tela — formas que se movem na tela do computador. A tartaruga é um instrumento eficiente



Curva de aprendizado

AP CURVA
REPITA 80

FRENTE 1
DIREITA 1

FIM
CURVA



Este exemplo fictício mostra como a linguagem MLOGO (LOGO adaptado para o português) incentiva um grupo de crianças a solucionar um problema novo.

"Se vamos fazer pétalas, primeiro temos de desenhar uma curva."

"Mas a tartaruga sempre se move em linhas retas."

"Se fizermos com que ela se mova uma pequena distância, virarmos um pouco, depois movermos para a frente novamente, e assim por diante, teremos uma curva."

"Está certo, a menor distância é 1 e o menor ângulo é 1. Vamos tentar fazer isso oito vezes."

CURVA
CURVA



"Isso é justamente o que queremos."

"Duas dessas farão uma pétala. Experimente."

AP PETALA
CURVA

DIREITA 90
CURVA

FIM
PETALA



"Isso não parece uma pétala. O que aconteceu?"

"Ela devia ter continuado a partir da última curva. Acho que nós falamos para ela mudar de direção."

"Mas como faremos para que ela vire?"

"Vamos tentar com 90, muitas vezes dá certo."

"E vamos criar uma nova palavra — PÉTALA. Assim ganhamos tempo."

AP PETALA
CURVA

DIREITA 100
CURVA

FIM
PETALA



"Está melhor, mas 90 ainda não é o bastante. O que vamos experimentar agora?"

"Vamos tentar fazer, em vez de ficar imaginando."

"Certo, e já aprendemos que se a tartaruga dá um giro completo ao redor de alguma coisa, ela gira uma total de 360."

"Bem, sabemos que ela vira 80 na primeira curva; assim, ela deve virar mais 80 na volta — e teremos 160."

"Deixando 200 para ser usado no momento de realizar a pétala."

"Não, porque, para retornar ao ponto de partida, ela teria de girar na outra extremidade também."

"Que tal se tentarmos a metade de 200?"

PETALA
PETALA
PETALA
PETALA



"Ótimo. Quatro dessas farão uma flor."

AP FLOR
REPITA 4

PETALA
DIREITA 10

FIM
FLOR



"Ainda não está muito bom. Ela está torta de um lado."

"Nós esquecemos de deixar um espaço entre as pétalas."

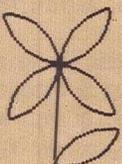
"Mas por que não fazemos o desenho delas começando na ponta de cada uma?"

"Porque, quando desenhamos uma pétala, a tartaruga está situada 100 à esquerda de onde ela começou."

"Assim, as pétalas são viradas em 100 à esquerda a cada vez."

"Parece que agora está certo — o que queremos é 90, vamos acrescentar um giro de 10 entre cada pétala."

FLOR
DIREITA 180
FRENTE 100
DIREITA 180
PETALA



"Finalmente! Basta virar a tartaruga e desenhamos o talo."

"Acho que 100 é um bom comprimento."

"Vamos fazer uma folha na ponta — pode ser igual a uma pétala."

"Mas veja o ângulo desta vez — a tartaruga tem de girar à direita."

para as crianças aprenderem conceitos básicos de relações de espaço, que as conduzem à geometria um pouco mais avançada.

Entretanto, o controle da tartaruga é apenas uma das operações do LOGO, e, por ser o aspecto visualmente mais interessante, tornou-se, sem dúvida, o mais popular. De grande importância também é o conceito de construção de idéias simples em termos mais sofisticados, e, ao contrário, o desdobramento de problemas mais complexos em problemas simples.

Esses processos podem ser vistos na conversação imaginária de um grupo de crianças aprendendo a fazer com que uma tartaruga desenhe uma flor na tela de um microcomputador usando uma versão em português chamada MLOGO desenvolvida pela Microarte (ver quadro ao lado). O início é de apenas três comandos básicos: FRENTE — que move a tartaruga para a frente mediante um valor específico; DIREITA — que vira a tartaruga em um ângulo determinado; e REPITA — que repete as linhas definidas no programa um número de vezes especificado.

Com essas idéias fundamentais, primeiramente as crianças constroem um "instrumento" — um programa — para desenhar a curva (AP CURVA... FIM). Para que essa seqüência completa seja conseguida, basta que se digite CURVA. Do mesmo modo, depois da experimentação e do posterior aprendizado, o comando PETALA, que faz uso do comando CURVA, é definido. Eventualmente, desenvolve-se um comando FLOR, e este desenha toda a flor.

A linguagem LOGO não é a única que incorpora essas estruturas (a outra é a FORTH), mas é a única especialmente projetada para crianças pequenas. Ela acaba com muitas das formalidades e procedimentos associados à programação em outras linguagens. Na verdade, o que se quer é que a criança não perceba que está programando um computador — ela está apenas solucionando um problema.

Em certas situações de aprendizado, o aluno nem chega a se envolver tanto. O professor determina, usando o LOGO, uma série de "instrumentos" que dizem respeito a uma matéria ou área específica de conhecimento. A criança pode então explorar a matéria, usando os "instrumentos" e fazendo suas próprias descobertas. Tais áreas recebem o nome de "micromundos", que são ambientes limitados onde o computador é utilizado para simular algo no mundo real e numa área específica.

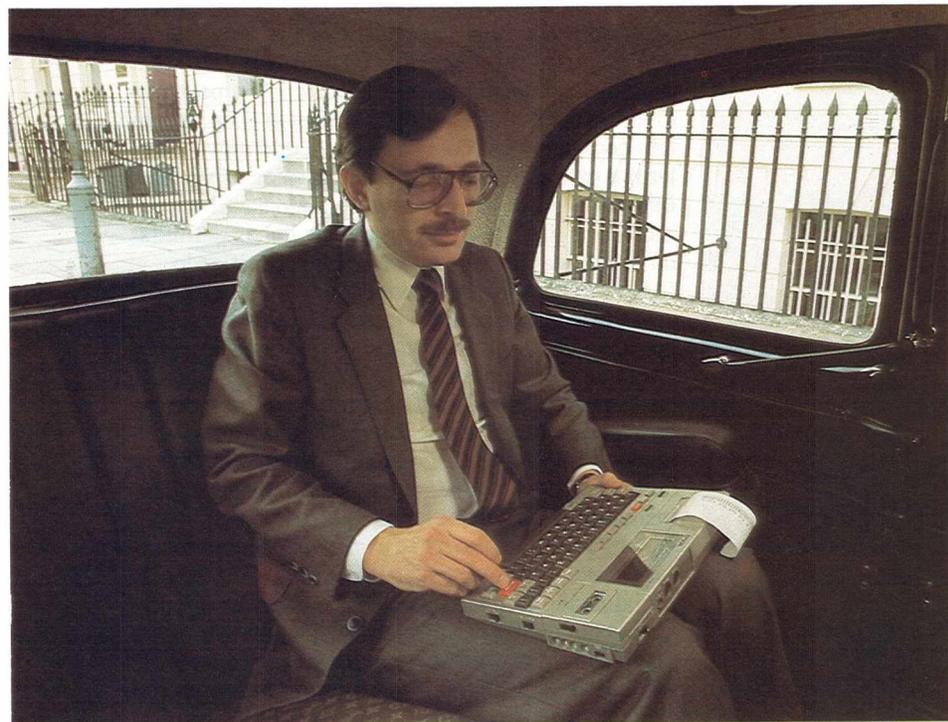
O melhor exemplo de um micromundo é talvez o modelo LOGO da física newtoniana. Apesar de a primeira lei de Newton dizer que sem a influência de forças externas um corpo continuaria se movendo em linha reta e velocidade constante, os jovens observadores percebem que no mundo real tudo perde a velocidade inicial. Isso causa um bloqueio no aprendizado. No entanto, com o uso do LOGO, um micromundo pode ser criado exatamente como diz a lei de Newton; e, com a ajuda de instrumentos para empurrar os objetos na tela, as crianças aprendem sozinhas as três leis de Newton.

A linguagem LOGO é de grande valor para o aprendizado com a utilização do microcomputador. As tartarugas de chão têm alto preço, e versões de LOGO com tartarugas de tela já estão disponíveis para alguns computadores pessoais.



A era dos portáteis

A uma maior evolução tecnológica correspondem equipamentos cada vez menores: os micros portáteis são hoje uma realidade.



Um brinquedo sério

Os computadores portáteis como o Epson HX-20, que possui até 32 Kbytes de memória e grande variedade de softwares, proporcionam ao executivo, onde ele estiver, uma quantidade de informações cada vez maior. Cada microcassete pode conter 220 Kbytes (cerca de 40.000 palavras, o que equivale a um banco de dados bastante amplo).

Processadores de palavras e folhas eletrônicas podem ser utilizados, permitindo que muitas atividades se realizem em qualquer lugar, até mesmo num táxi.

O desenvolvimento do microcomputador portátil que conhecemos hoje vem de duas direções. Uma delas foi a calculadora de bolso, como a Sharp PC1251 e a Casio FX700P. A outra é o processo evolutivo da miniaturização, que resultou em aparelhos como o Tandy TRS80 Model 100 Personal Computer e o Epson HX-20.

Esses avanços derivam do desenvolvimento de chips mais compactos, que permitem maior número de informações contidas num mesmo espaço físico.

Com o advento do microcomputador de um só chip, em 1972, seria possível teoricamente a construção de um computador numa caixa não maior que um maço de cigarros. Entretanto, o tamanho da tela e os meios de acesso, como o teclado, trouxeram limitações práticas à miniaturização.

É verdade que as calculadoras de bolso ficaram menores, e hoje os relógios digitais podem ser usados como calculadoras, mediante um tipo de "teclas" especialmente feitas para sua operação. Mas é difícil realizar uma simples operação aritmética nessas calculadoras, e seria desanimador dar entrada, por exemplo, a um programa de cinquenta linhas em BASIC. Sendo assim, parece improvável que os computadores se tornem tão pequenos como os relógios de pulso; chegarão, no máximo, a ser apresentados com o mesmo tamanho das calculadoras portáteis.

O fato de as calculadoras tornarem-se programáveis, embora se restrinjam a suas próprias notações de programação, foi só a primeira etapa para a incorporação de uma linguagem de alto nível, e a escolha óbvia foi o BASIC.

Quase ao mesmo tempo, os fabricantes começaram a utilizar uma RAM permanente — um tipo de memória que, pela retenção de pequena carga elétrica, não perde o conteúdo quando a máquina é desligada. Começou-se também a empregar um gerador de caractere mais amplo, que permitia que os caracteres alfabéticos fossem mostrados junto com os numerais.

Tais aparelhos, de preço razoável, como os computadores de bolso da Sharp e da Casio, possuem uma série de comandos BASIC comparáveis aos dos computadores pessoais, com capacidade de memória semelhante, e estão entrando rapidamente no mercado das calculadoras portáteis. Compactos a ponto de caberem até num bolso de camisa, alguns oferecem a possibilidade de ser ligados a uma impressora ou a um gravador cassete.

Sem dúvida, as possibilidades de comunicação continuarão a crescer, caso a procura seja suficiente, permitindo a transferência de informação por linhas telefônicas. O equipamento atraente para cientistas e engenheiros é o da Casio, pois retém grande quantidade de funções matemáticas e científicas, o que torna essas calculadoras bastante úteis.

Modelos com impressora acoplada e equipamento cassete completo estão à venda em muitos países. Nessa linha inclui-se o Sharp 1251.

Modelos como o HX-20 da Epson, o Tandy 100 e o NEC 8220, representam um grande passo da indústria de micros. Estes aparelhos oferecem implementação BASIC completa (Microsoft, em todos os casos), 16 a 64 K de RAM útil, um mostrador de cristal líquido (LCD, Liquid Crystal Display) de tamanho razoável (20 x 4 caracteres no da Epson, 40 x 8 nos outros dois) e podem ser conectados a um grande número de periféricos.

Ligando o aparelho a um monitor comum (talvez seja preciso comprar uma interface especial), você terá um micro comparável, em capacidade e desempenho, a um microcomputador convencional, de preço razoável. A característica mais importante dessas máquinas é que mantêm seu próprio suprimento de energia, permitindo, assim, fácil transporte.

O Epson, por exemplo, tem baterias de níquel-cádmio, com carga elétrica de cinquenta horas e um sistema projetado para que, quando as baterias estiverem fracas, ele se feche e preserve os dados essenciais.



O NEC e o Tandy, por sua vez, usam um sistema complementar de metal óxido semiconductor, que exige muito menos energia e os capacita a operar com pequenas baterias comuns.

O Epson, o primeiro exposto a uma campanha publicitária maciça no mercado dos micros, era visto a princípio como um “auxiliar de executivos”, mas na verdade seu uso é bem mais adequado fora do escritório. Ele é perfeitamente apropriado para captar dados (obtenção de informação) na fábrica, em locais distantes ou até mesmo na rua. Esses dados, depois, serão processados no escritório.

Na prática, o Epson oferece várias maneiras de transmitir dados. Acessórios apropriados permitem que seja usado como terminal de telex; através de um modem ele é ligado a outra máquina, pela linha telefônica; ou ainda, de modo mais simples e barato, os dados podem ser armazenados numa fita cassete e enviados pelo correio.

Tal versatilidade faz com que essas máquinas sejam atraentes até para pessoas que nunca tinham visto o computador como um aparelho de uso diário. Um vendedor, por exemplo, que faz vinte visitas ou mais por dia, apenas para anotar pedidos, pode usar uma máquina portátil que grave todos os pedidos normais que ele recebe. O programa processador de pedidos faria com que ele tivesse uma lista dos produtos à mão e, logo que o cliente localizasse um item na lista, ele iria apenas registrar a quantidade. Após cada chamada, os dados dos pedidos seriam depositados numa fita e no final do dia o vendedor a enviaria para uma agência ou transmitiria a informação diretamente ao sistema principal de computação da empresa.

A própria impressora do micro forneceria ao vendedor uma cópia dos pedidos do dia e ao consumidor uma confirmação imediata.

Versões um pouco mais sofisticadas também podem contabilizar o estoque dos produtos à venda e avisar quando o depósito está em falta; nos locais onde há mais de um vendedor, o micro estaria ligado a um equipamento central. Esse processo não apresentaria problemas e, além do mais, o custo e o peso do acoplador acústico seriam menores. Neste ponto, o micro portátil transforma-se num terminal interativo, permitindo que todo o banco de dados da empresa seja interrogado, com resposta imediata, sempre que necessário.

Além dos óbvios benefícios do acesso imediato a informações atualizadas, o tempo poupado e a economia trazida por esse tipo de registro de dados online podem aumentar substancialmente o lucro da empresa. Desse modo esses equipamentos acabam em pouco tempo pagando a si próprios.

Assim como os computadores de bolso estão tornando obsoletas as calculadoras portáteis, os computadores portáteis estão superando os pequenos terminais que recolhem dados. Esses terminais movidos a bateria — que não podem ser programados — estiveram no mercado europeu por algum tempo, mas não foram bem-sucedidos devido ao alto custo e à dificuldade de operação.

Com o software adequado, a habilidade de se ligar monitores e impressoras para escritório (e talvez unidades de disco ou fita) faz com que os computadores portáteis tornem-se cada vez mais comuns.

As máquinas portáteis desse tipo favorecem o primeiro mercado real para memória rápida — uma memória permanente, baseada num tipo diferente de chip, que oferece enorme capacidade de armazenamento numa pequena embalagem. Exemplos típicos, como o PC5000 da Sharp, fornecem 128 K de memória em cartucho, com um tempo de acesso mais rápido que o do disco. A perspectiva de máquinas pequenas com 1 milhão de bytes de memória incorporada apresenta-se como uma possibilidade bastante próxima.

Tal capacidade de memória está normalmente disponível em disco, no terceiro tipo de máquinas portáteis, como o Osborne Executive, o Ajile Hyperion e o Portico Miracle. De preço bem superior ao do HX-20 ou ao do TRS100, essas máquinas podem ser descritas como transportáveis e não portáteis, pois precisam de grande quantidade de energia. Grupos de baterias são usados para algumas máqui-

Para todos os gostos

O HX-20 da Epson (primeiro equipamento à direita) oferece até 32 Kbytes de RAM e grande variedade de periféricos.

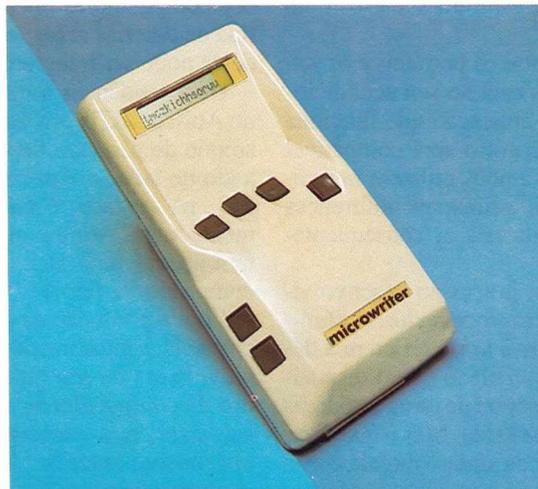
Dentre tantos computadores portáteis, o modelo PC1251, da Sharp (na foto aparecem a impressora e a unidade microcassete), não é tão potente mas, em compensação, de preço menor que muitos outros desse gênero. Já o Casio FX700P utiliza um gravador cassete normal.

Depois de fazer parte do mercado de computadores “transportáveis”, a Osborne lança um segundo modelo, o Executive, com tela maior e outros aperfeiçoamentos.



nas, mas muito raramente conseguem possibilitar mais que algumas poucas horas de uso.

As especificações dessas máquinas geralmente incluem duas unidades de discos flexíveis, um monitor de TV acoplado, um teclado separado e um sistema operacional como o CP/M.



Escrita por computador

O conceito de um “caderno de anotações eletrônicas” deu um passo à frente com a introdução do Microwriter, um processador de palavras feito exatamente para operações normais. As seis teclas são pressionadas em combinações diferentes a fim de abranger todo o alfabeto. A saída pode ir direto para a impressora, com alguns comandos que permitam a organização do texto, ou então para um microcomputador de mesa, para armazenamento.



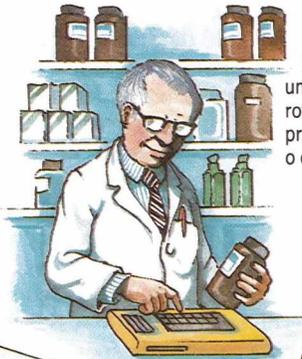
Ferramentas de trabalho

Computadores portáteis e baratos estão sendo usados em áreas que anteriormente não tinham acesso aos sistemas de processamento de informação.

Os jornalistas registram no computador as perguntas que desejam fazer numa entrevista. Depois, as respostas são



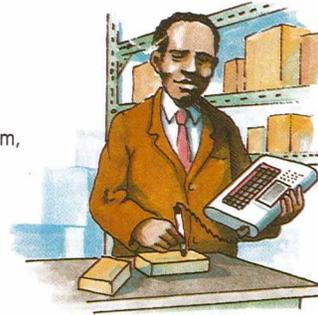
também registradas; e eles editam a notícia na forma que desejam.



Farmacêuticos têm agora um modo fácil e barato de rotular legivelmente seus produtos, facilitando, também, o controle do estoque.



Os vendedores fazem estimativas corretas em cálculos complicados como os da instalação de um sistema de refrigeração central ou os de seguros de vida.



O controle de estoque é importante, mesmo que a empresa seja pequena. Em grandes indústrias, onde o valor do estoque é muito elevado, convém que se tenha um registro exato. Os micros portáteis, com 32 ou 64 Kbytes de RAM, armazenam os dados numa fita cassete.

O primeiro a ser lançado no mercado foi o modelo 1 de Adam Osborne, que oferecia uma série completa de software, incluindo BASIC, Wordstar e Supercalc, além do CP/M.

Sua única falha era ter uma tela de tamanho reduzido — 100 x 85 mm. Isso produzia caracteres pequenos demais, que eram lidos linha por linha. No entanto, os usuários logo se acostumaram com esse problema, e rapidamente o Osborne 1 tornou-se um grande líder de vendagem.

Competidores no mercado trouxeram facilidades para o usuário, principalmente no que se referia a consumo de energia, velocidade do microprocessador e capacidade de armazenamento do disco. Por fim, esses tipos de máquina alcançaram o mesmo nível. Todos se adaptam bem a sofisticados sistemas de escritórios. Em muitos casos, eles oferecem total compatibilidade com as máquinas específicas, principalmente o IBM PC, o que estimula seu uso como um “segundo computador”.

A verdadeira diferença entre os portáteis e os sistemas usados em escritórios está no seu acondicionamento. Um programa de aplicação que funciona num sistema normal no escritório será compatível com um portátil, desde que ambos utilizem um microprocessador similar. As principais diferenças encontram-se no tamanho da tela e, conseqüentemente, dos caracteres.

Isso tudo é possível hoje, graças ao desenvolvimento de um único chip microprocessador. As limitações no acesso à entrada ou à saída refletem a dependência que temos de um teclado como dispositivo de entrada principal e do tubo de raios catódicos como dispositivo principal de saída. Mas o que dizer do reconhecimento da voz e da síntese da fala como formas alternativas?

Não há razão para que um micro de poder semelhante ao do Ajile Hyperion, por exemplo, ocupe espaço maior do que o Walkman da Sony. Quando tivermos a tecnologia adequada, o Ajile Hyperion poderá ser semelhante a um walkman, com um microfone em miniatura embutido.

Numa época como essa, em que a tecnologia se desenvolve com muita rapidez, os avanços nesse setor provavelmente acontecerão em breve. Acesso à entrada e à saída dos micros pela fala logo deixarão de pertencer ao mundo da fantasia. Imaginemos, por exemplo, a combinação do Sinclair ZX Spectrum com uma tela plana de TV, o microdrive da Sinclair, e a energia do conjunto de níquel-cádmio recarregável. Não há razão real para que esses aparelhos não se combinem entre si.

Por ter um tamanho adequado, o ZX Spectrum é a máquina mais indicada para este fim, mas a maioria dos computadores opera num tipo de voltagem obtida de baterias secas, e por essa razão qualquer um deles pode ser transformado em portátil.

Uma tendência muito interessante manifesta-se no Apricot ACT, um sistema normal para escritório, só que de pequeno tamanho. Se o monitor for deixado de lado, o Apricot torna-se portátil. (“Portátil”, na verdade, é uma caracterização excessivamente vaga. Muitos micros considerados portáteis têm apenas uma alça, mas não possuem unidades compactas de fato.)

Com a produção de grande quantidade de micros portáteis, a indústria deu um passo à frente no desenvolvimento de seu potencial, que é o de colocar o uso do computador ao alcance de todos, a um preço acessível, sem necessidade de treinamento ou conhecimento anterior. A cada dia esse objetivo está mais próximo.



I-7000

Os microcomputadores Itautec são bastante versáteis, flexíveis, e apresentam uma tecnologia totalmente própria, tanto em hardware como em software.

Atualmente, a Itautec despende grande parte do seu esforço em pesquisa, formação de seus técnicos e desenvolvimento do setor de informática. Devido a este fato, excelentes resultados têm sido obtidos, com o lançamento de diversos produtos, entre eles a família de microcomputadores I-7000.

Os microcomputadores Itautec possuem grande aptidão para ambientes on-line, interagindo com diversos equipamentos de grande porte, seja como terminal ou na modalidade de processamento distribuído. Mas isto não os impede de atuar em ambientes off-line, como microssistemas.

A Itautec coloca à disposição do usuário uma estrutura, com uma série de serviços, descritos abaixo:

Centro de Atendimento aos Usuários (CAU)

O usuário poderá resolver alguns problemas e dúvi-

das durante a operação dos softwares e equipamentos, através de uma ligação telefônica ou telex ao CAU. Uma equipe de técnicos dará uma solução ao problema levantado num intervalo sempre inferior a 24 horas.

Suporte técnico

Formado por uma equipe de técnicos, que atuam na instalação e posterior acompanhamento dos equipamentos em campo.

Centro Educacional

Estruturado para atender, treinar e ensinar os atuais e futuros usuários.

Documentação técnica

Tem como preocupação fundamental fornecer informações técnicas de forma altamente profissional, abrangendo todos os produtos Itautec do mercado.





Assistência técnica

Estrutura para efetuar reparos em campo, com a substituição de peças defeituosas, e laboratório para manutenções mais complexas.

A família de microcomputadores I-7000 abrange basicamente três faixas distintas do mercado: uso pessoal e profissionais liberais, empresas pequenas, médias e de grande porte.

A mesma tecnologia (hardware e software) é empregada no Júnior, Júnior-E e I-7000. A diferença básica refere-se à possibilidade de expansão: no Júnior, menor que no Júnior-E e neste inferior ao I-7000.

As características técnicas são: CPU NSC 800 D4 NATIONAL, 4 K de memória de vídeo, 64 K RAM da placa básica, clock 4 MHz, vídeo multitonal, teclado com caracteres em português, interface paralela Centronics ou Dataproducts, saída serial, duas portas analógicas com três interfaces A/D, saída de áudio programável, entrada de cartucho de até 32 K EPROM e 4 K EPROM BOOT.

Placas de expansão

Placa de controle de até 4 unidades de disco de 8 pol., placa de controle de até 4 unidades de disco de 5 1/4 pol. e uma unidade Winchester, placa de expansão de memória para 128 K, além de duas saídas seriais RS232 C, placa de interface telex, placa concentradora de três micros, placa de interface co-axial — IBM 3274 emulando vídeo 3278/2 STD e placa de interface co-axial — IBM 3274, emulando impressora 3287 STD.

O sistema operacional SIM/M é compatível com a versão CP/M 2.2, por isto utiliza qualquer tipo de software, desenvolvido para esta versão. Além das facilidades-padrão do CP/M 2.2, o SIM/M oferece um conjunto de 27 funções adicionais nos BDOS, que permitem comunicação, tratamento de vídeo, tratamento das entradas analógicas etc.

A Itautec oferece, ainda, o Redator (processador de textos), Calcotec (planilha eletrônica de cálculos), BASIC, utilitários assembler, utilitários do sistema operacional e de uma série de 150 softwares, desenvolvidos por software houses cadastradas pela Itautec.



Impressora matricial 100 cps, 132 colunas, bidirecional

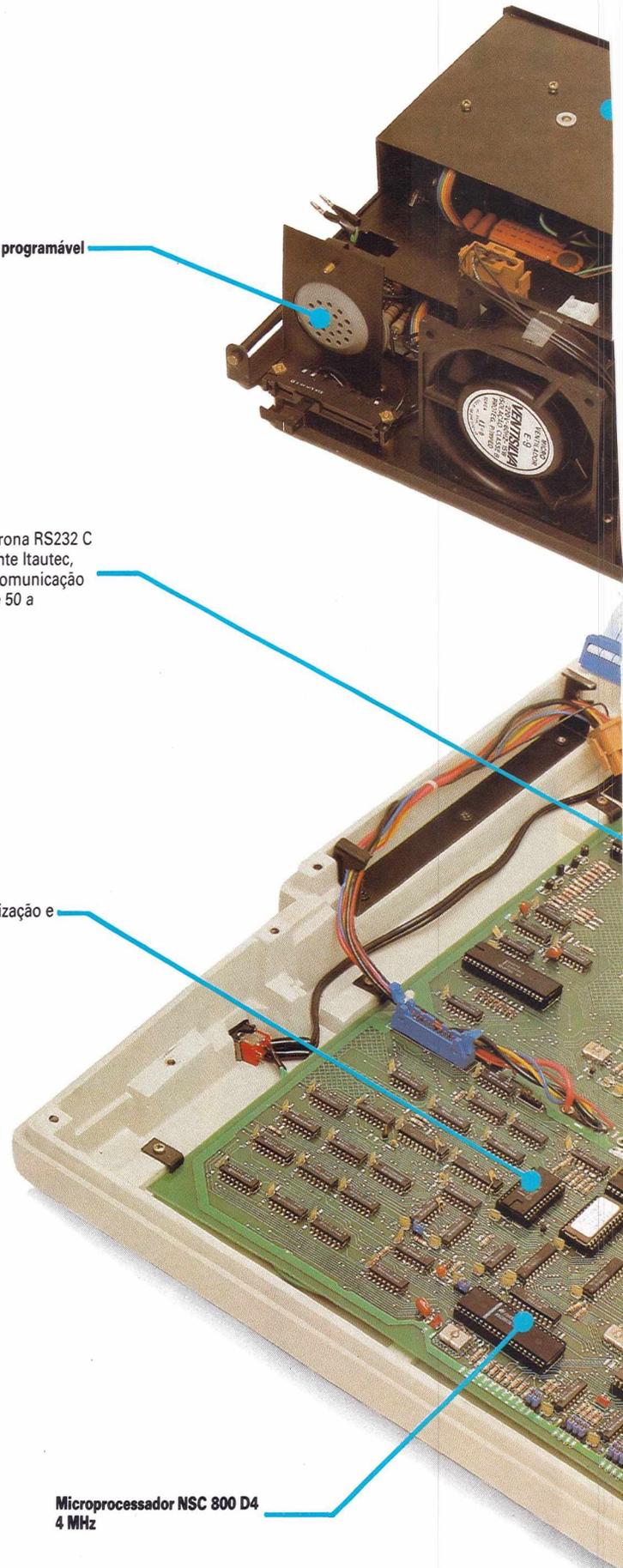
Saída de áudio programável

Interface serial

assíncrona/síncrona RS232 C ou Elo de corrente Itautec, velocidade de comunicação programável de 50 a 19.200 bps

4 K de EPROM utilizada para inicialização e testes

Microprocessador NSC 800 D4
4 MHz





Fonte com opção 110/220 V

I-7000

MICROPROCESSADOR

NSC 800 D4

CLOCK

4 MHz

MEMÓRIA

64 K de RAM, 4 K de memória de vídeo e 4 K de EPROM.

VÍDEO

Monitor de fósforo verde (12 polegadas) com resolução de 640 x 200 pontos, faixa de passagem de 20 MHz, tela anti-reflexiva e base giratória.

TECLADO

Tipo máquina de escrever com 79 teclas, sendo 12 de função, teclado numérico separado.

LINGUAGENS

BASIC ITAUTEC, COBOL MB

PERIFÉRICOS

Duas unidades de disco 5 1/4 pol., face simples/dupla, 1 unidade de disco Winchester, 1 unidade de disco 8 pol., face simples/dupla, impressora matricial ou tipo margarida, plotter, concentrador de micros I-2080, set co-axial, set VT-52, RJE.

Interface paralela padrão Centronics ou Dataproducts

Duas portas analógicas com 3 interfaces conversoras A/D cada -- entrada de 0 V a +5 V, 256 níveis de conversão e frequência de amostragem máxima de 10 KHz

4 K de memória de vídeo

Controlador de vídeo colorido 40 ou 80 colunas por 25 linhas ou ainda em modo semigráfico, com 160 x 100 pontos endereçáveis na tela

64 K de RAM

Tentando a sorte

Continuando o estudo das funções em BASIC, chegamos à função RND, que produz números aleatórios (ou quase aleatórios) para uso em jogos ou programas estatísticos.

Já vimos como várias das funções em linguagem BASIC operam, e agora vamos examinar uma das mais freqüentemente empregadas — a função RND, usada para gerar números aleatórios. É utilizada também em jogos, sempre que haja algum componente aleatório.

Infelizmente, RND é uma das “siglas” de uso mais variável em BASIC: nossa descrição dela poderá diferir do modo como essa função está adaptada ao seu microcomputador. Vamos esclarecer algumas das diferenças entre a linguagem BASIC usada no curso de programação BASIC e a que existe em seu computador.

A maioria de nossos programas segue a linguagem BASIC da Microsoft (ou MBASIC). A Microsoft é uma empresa dos Estados Unidos e sua linguagem BASIC foi uma das primeiras a se tornar amplamente acessível ao público. A linguagem BASIC não possui padrão oficial, mas sua versão Microsoft é a que está mais próxima do que se poderia chamar de um padrão: muitas outras marcas seguem o modelo da Microsoft, e essa empresa foi encarregada de produzir versões para diversos computadores mais populares.

A principal diferença entre o MBASIC e a maioria das versões mais recentes está em que os microcomputadores agora possuem alta capacidade para a elaboração de gráficos, não existente quando o MBASIC foi desenvolvido. As versões mais recentes de BASIC em geral incluem uma série de comandos e instruções para gráficos. Para obter o máximo rendimento do computador, é necessário explorar totalmente sua capacidade de elaboração de gráficos, o que exige um estudo cuidadoso do manual de instruções do equipamento.

Entre as várias versões de BASIC adaptadas a microcomputadores, a da Sinclair (usada no TK83, TK85, CP 200, Ringo) é provavelmente a que mais difere do MBASIC. A versão da Texas Instruments (usada no TI99/4A) tem igualmente diferenças significativas. Até onde for possível, indicamos como fazer determinadas modificações, nos quadros “A propósito...”, que você deve observar, sempre que encontrar algum problema no processamento de programas.



Como dissemos, a função RND difere de uma versão para outra. Verifique no manual de instruções de seu equipamento como ela funciona. Aqui explicamos o modo de usá-la em um jogo de dados muito simples. Conforme programas anteriores, realiza-

mos a maior parte do trabalho em sub-rotinas. Esta técnica tem a vantagem de tornar o programa mais legível, mais fácil de escrever e de corrigir.

O programa principal começa com a instrução RANDOMIZE, na linha 20. A maioria das versões de BASIC, mas não todas, precisa dessa instrução para restaurar a função RND. Na verdade, é muito difícil obter dos computadores a apresentação de números genuinamente aleatórios. Sem esta operação de restauração, a mesma seqüência de números supostamente aleatórios será produzida, toda vez, pela função RND. A linha 50, a seguir, chama uma sub-rotina que utiliza a função RND para atribuir um número aleatório à variável D. Escreveremos assim:

```
320 LET D = INT(10 * RND)
```

Esta é a linha que mais provavelmente exigirá alterações para que você possa processar o programa. Os detalhes de como operam diferentes versões da função RND estão no quadro “A propósito...”. Assim, vejamos o que acontece no BASIC da Microsoft. A instrução RND utiliza uma expressão (entre parênteses, usual em funções) como alternativa para alterar levemente a seqüência dos números gerados. Sem essa expressão — por exemplo, LET A = RND —, o valor da variável A será um número entre 0 e 1. Não queremos um número menor que 1 e, assim, multiplicamos o número por 10. Isto é feito da seguinte forma: LET A = 10 * RND. Se, por exemplo, a função RND apresentar o valor 0,125455, o valor de A será, então, 1,25455.

Para eliminar a parte fracionária do número e reter apenas a parte inteira, usamos a função INT (inteiro) da seguinte forma: LET A = INT(10 * RND).

Para obter números inteiros aleatórios que variem entre 1 e 6 com o BASIC Microsoft, temos de verificar se os números apresentados foram maiores que 6 ou menores que 1, já que tais números não seriam apropriados para o jogo de dados. Isto é feito nas linhas 330 e 340:

```
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
```

Se a variável D estiver fora dos limites de 1 a 6, as instruções GOTO farão com que o programa retorne e proceda a nova tentativa.

Tendo escolhido um valor aleatório entre 1 e 6 para a variável D, a sub-rotina de lançamento de dados retorna ao programa principal. Este imprime a mensagem: SUA CONTAGEM DE PONTOS E, seguida da figura de um dado. Observe como o desenho correspondente é escolhido e feito na sub-rotina ESCOLHER. Por exemplo, se o dado (e, desse modo, a variável D) tiver o número 1, a linha 410 chamará a sub-rotina que se inicia na linha 530, assim:



410 IF D = 1 THEN GOSUB 530

Esta sub-rotina é simplesmente uma série de instruções PRINT, elaboradas para produzir gráficos simplificados na tela. Seu BASIC pode dispor de gráficos de tela muito melhores e, se este for o caso, será melhor substituir as nossas sub-rotinas pelas instruções apropriadas de gráficos.

Tendo escolhido aleatoriamente um dado para você, o programa repete o procedimento para selecionar e apresentar um dado para o computador. A parte do programa que decide quem venceu foi incorporada ao programa principal; poderia igualmente ter sido escrita como uma sub-rotina, mas isto dificilmente valeria a pena, já que consiste apenas em quatro linhas. A linha 200 compara a variável M (meu dado) com a variável C (o dado do computador) para verificar se são iguais. Se forem, a palavra EMPATADO será atribuída à variável alfanumérica S\$. A linha 210 verifica se a variável M é maior que a variável C. Se for, ela atribuirá as palavras VOCE GANHOU à variável S\$. A linha 220 verifica se M é menor que C. Se for, ela atribuirá as palavras O COMPUTADOR GANHOU à variável S\$. A linha 240 apenas imprime o resultado e o jogo termina aí. Embora bastante longo, este programa é muito simples: utiliza apenas uma função, RND, e não recorre a loops ou variáveis indexadas.

Uma vez que a função RND pode variar, e algumas versões do BASIC requerem a instrução RANDOMIZE para dar origem a seqüências de números aleatórios, há algum meio de gerar números efetivamente aleatórios (isto é, que não podem ser previstos), sem recorrer a essas funções? Há várias técnicas possíveis.

Uma das funções que até agora não examinamos foi a INKEY\$ (dita "inkey-string"). Toda vez que o termo INKEY\$ é encontrado, o programa examina o teclado e verifica se alguma tecla foi pressionada. O programa não aguarda que se dê entrada a um caractere, como acontece quando é usado o comando INPUT. Por isso, o comando INKEY\$ é em geral colocado em um loop. Desse modo, o programa examina continuamente o teclado, aguardando que algo seja introduzido. Geralmente, há um teste incluído no loop, para concluí-lo, quando o caractere adequado for fornecido. Isto torna possível escrever um programa para formar um loop que se encerrará quando um caractere específico for digitado. O que acontecerá se usarmos o seguinte programa?

```

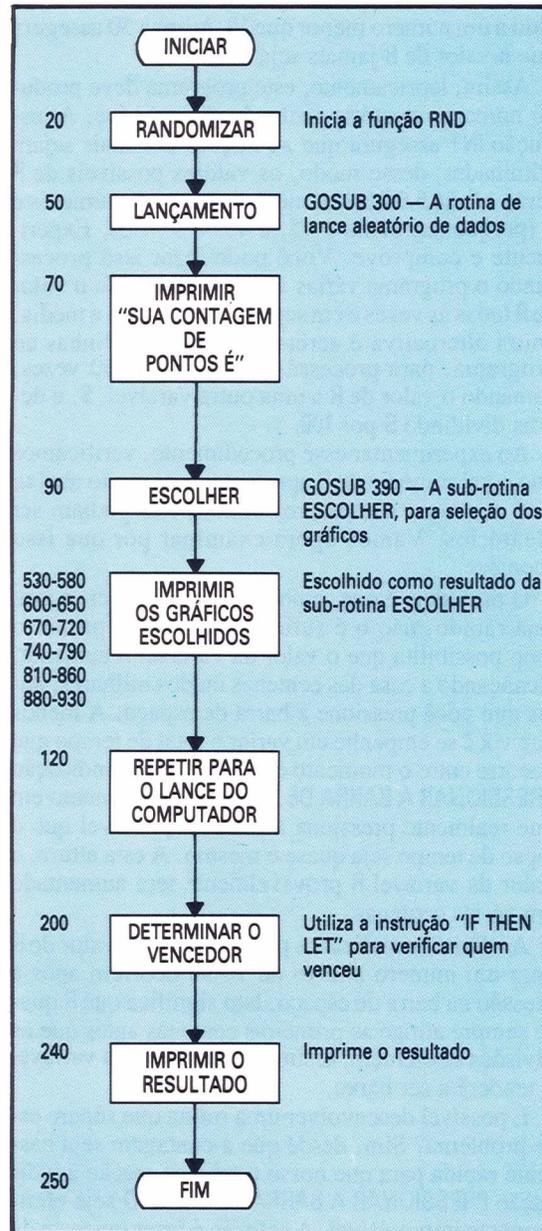
10 PRINT "PRESSIONAR A BARRA DE ESPACO"
20 FOR X = 0 TO 1
30 LET R = R + 1
40 LET A$ = INKEY$
50 IF A$ = " " THEN GOTO 80
60 LET X = 0
70 NEXT X
80 FOR Q = 0 TO 1
90 IF R < 10 THEN GOTO 130
100 LET Q = 0
110 LET R = R/10
120 NEXT Q
130 PRINT INT (R)
140 END
    
```

Neste caso, a variável R é um número aleatório? Provavelmente, sim; examinemos por quê.

A linha 10 imprime a indicação PRESSONAR A

BARRA DE ESPACO. Antes de termos tempo de atender a esta indicação, o programa terá iniciado o loop FOR X = 0 TO 1, na linha 20. Os números 0 e 1 parecem ser limites estranhos para o loop, mas vejamos resumidamente como esta estrutura é usada. A linha 30 atribui o valor 1 à variável R na primeira passagem pelo loop. A linha 40 atribui à variável A\$ qualquer caractere que for digitado no teclado. Isto é realizado pelo uso da função INKEY\$. Se você digitar a letra R, esta será a letra atribuída à variável A\$. A linha 50 testa esta variável a fim de verificar se corresponde a um espaço (o que é representado em BASIC como um espaço entre aspas, assim: " "). Se A\$ for um espaço, o programa se desviará, usando a instrução GOTO, mas, se não o for, seguirá para a linha seguinte.

Esta será a linha 60, que contém a instrução LET X = 0, e a variável X é o contador do loop. A instrução NEXT X, na linha 70, faz com que o programa retorne ao início do loop, na linha 20. Uma vez que X assume novamente o valor 0, o loop repete o procedimento. Deste modo, o loop FOR X = 0 TO 1 será repe-



Fluxo do programa

O fluxograma ao lado mostra, de forma simplificada, as principais atividades realizadas pelo programa. À esquerda, são apresentados os números das linhas correspondentes e à direita estão pequenas notas de esclarecimento. Este fluxograma não está completo, pois muitas das "decisões" e desvios do programa não são apresentados.

tido indefinidamente, enquanto o teste `IF A$ = " "` não for verdadeiro.

Se em algum estágio a barra de espaço for pressionada, o caractere que representa um espaço será atribuído à variável `A$` e, assim, o programa se desviará para a linha 80 e o loop não será repetido.

O que acontecerá, porém, enquanto o loop se repetir? A linha 30 aumenta o valor de `R` em cada repetição do loop. Na primeira passagem, a variável `R` assumirá o valor 1, na segunda o valor $1 + 1$ e assim sucessivamente. Quando o loop for interrompido pelo teste da variável `A$`, poderemos ler a variável `R` e ver até onde chegou nossa contagem.

Todavia, o computador opera muito rapidamente, atingindo a casa das centenas, quando você pressionar a barra de espaço. O que faremos se necessitarmos de valores de `R` entre 1 e 10? A linha 80 monta outro loop que nos permite testar a variável `R` e dividi-la por 10 se for maior que este número. Sendo `R` maior que 10, o teste na linha 90 falha, a variável `Q` assume novamente o valor 0 e o loop é repetido. A linha 110 divide o valor da variável `R` por 10 e o resultado só será impresso quando seu valor for reduzido a um número menor que 10. A linha 30 assegura que o valor de `R` jamais seja 0.

Assim, teoricamente, este programa deve produzir números aleatórios entre 1 e 9, inclusive. A instrução `INT` assegura que as frações decimais sejam eliminadas; desse modo, os valores possíveis de `R` serão 1,2,3,4,5,6,7,8,9. A média entre esses números é 5 (porque sua soma é 45, e $45 \div 9 = 5$). Experimente e comprove. Você pode fazer isso processando o programa várias vezes, registrando o valor de `R` todas as vezes e em seguida calculando a média. Outra alternativa é acrescentar algumas linhas ao programa, para processá-lo, digamos, 100 vezes, somando o valor de `R` a uma outra variável, `S`, e depois dividindo `S` por 100.

Ao experimentar esse procedimento, verificamos que o valor médio de `R` apresentava-se muito abaixo de 5 e, assim, os números obtidos não podiam ser aleatórios. Vamos agora examinar por que isso acontece.

O problema é que, embora o programa em BASIC seja rápido, não o é suficientemente. O primeiro loop possibilita que o valor da variável `R` aumente, alcançando a casa das centenas ou dos milhares, antes que você pressione a barra de espaço. A menos que você se empenhe em variar o total de tempo que decorre entre o momento em que recebe a indicação **PRESSIONAR A BARRA DE ESPACO** e o momento em que realmente pressiona a barra, é provável que o lapso de tempo seja quase o mesmo. A esta altura, o valor da variável `R` provavelmente terá aumentado em várias centenas.

As divisões realizadas para redução do valor de `R` para um número abaixo de 10 só ocorrem após a pressão na barra de espaço. Isto significa que `R` quase sempre atinge as primeiras centenas antes que as divisões se efetuem; assim, o valor final da variável `R` tenderá a ser baixo.

É possível desenvolver uma rotina que supere esse problema? Sim, desde que a contagem seja bastante rápida para que nosso tempo de reação à indicação **PRESSIONAR A BARRA DE ESPACO** seja efetivamente imprevisível. A solução é fazer um teste de

```

10 REM JOGO DE DADOS — PROGRAMA PRINCIPAL
20 RANDOMIZE
30 REM SEU LANCE
40 REM GOSUB ROTINA "LANCE"
50 GOSUB 300
60 LET M = D
70 PRINT "SUA CONTAGEM DE PONTOS E"
80 REM GOSUB ROTINA "ESCOLHER"
90 GOSUB 390
100 PRINT
110 REM LANCE DO COMPUTADOR
120 REM GOSUB ROTINA "LANCE"
130 GOSUB 300
140 LET C = D
150 PRINT "A CONTAGEM DE PONTOS DO COMPUTADOR E"
160 REM GOSUB ROTINA "ESCOLHER"
170 GOSUB 390
180 PRINT
190 REM QUEM GANHOU?
200 IF M = C THEN LET SS = "EMPATADO"
210 IF M > C THEN LET SS = "VOCE GANHOU"
220 IF M < C THEN LET SS = "O COMPUTADOR GANHOU"
230 REM IMPRIMIR O RESULTADO
240 PRINT SS
250 END
260 REM
270 REM
280 REM
290 REM
300 REM SUB-ROTINA LANCE ALEATORIO DE DADOS
310 REM
320 LET D = INT(10*RND)
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
350 RETURN
360 REM
370 REM
380 REM
390 REM SUB-ROTINA ESCOLHER
400 REM
410 IF D = 1 THEN GOSUB 530
420 IF D = 2 THEN GOSUB 600
430 IF D = 3 THEN GOSUB 670
440 IF D = 4 THEN GOSUB 740
450 IF D = 5 THEN GOSUB 810
460 IF D = 6 THEN GOSUB 880
470 RETURN
480 REM
490 REM
500 REM
510 REM SUB-ROTINA "GRAFICOS"
520 REM
530 PRINT " "
540 PRINT "-----"
550 PRINT "| |"
560 PRINT "| * |"
570 PRINT "| |"
580 PRINT "-----"
590 RETURN
600 PRINT " "
610 PRINT "-----"
620 PRINT "| *|"
630 PRINT "| |"
640 PRINT "|* |"
650 PRINT "-----"
660 RETURN
670 PRINT " "
680 PRINT "-----"
690 PRINT "| *|"
700 PRINT "| * |"
710 PRINT "|* |"
720 PRINT "-----"
730 RETURN
740 PRINT " "
750 PRINT "-----"
760 PRINT "|* *|"
770 PRINT "| |"
780 PRINT "|* *|"
790 PRINT "-----"
800 RETURN
810 PRINT " "
820 PRINT "-----"
830 PRINT "|* *|"
840 PRINT "| * |"
850 PRINT "|* *|"
860 PRINT "-----"
870 RETURN
880 PRINT " "
890 PRINT "-----"
900 PRINT "|* *|"
910 PRINT "|* *|"
920 PRINT "|* *|"
930 PRINT "-----"
940 RETURN

```



limite superior, no primeiro loop. Observe este programa:

```
10 PRINT "PRESSIONAR A BARRA DE ESPACO"
20 FOR X = 0 TO 1
30 LET R = R + 1
40 IF R > 9 THEN LET R = 1
50 IF INKEY$ = " " THEN GOTO 80
60 LET X = 0
70 NEXT X
80 PRINT R
90 END
```

Neste programa, a variável R jamais poderá ser menor que 1 ou maior que 9. No momento em que a barra for pressionada (e reconhecida pela função INKEY\$ na linha 50), a variável R terá um valor situado entre 1 e 9, inclusive.

Este programa foi testado mil vezes e o valor médio da variável R foi 5,014. O fato de a média perfeita ser 5 e o erro aqui ter sido da ordem de apenas 0,28% sugere que o programa efetivamente gera números aleatórios muito próximos à média teórica. Evidentemente, a questão aqui está no fato de que, mesmo quando o programa parece razoável no papel, nele podem surgir falhas imprevisíveis. Assim, é interessante a realização de um teste.

Alguns leitores terão notado que esses programas de números aleatórios poderiam ser reduzidos pelo recurso às instruções GOTO em lugar do loop FOR-NEXT. O motivo por que evitamos as instruções GOTO se evidenciará mais adiante no curso de programação em linguagem BASIC.

Exercícios

■ **Função RND.** Alterar o último programa do texto, para obter um número aleatório que varie de 1 a 6 (inclusive).

A propósito...

RANDOMIZE

Nos computadores compatíveis com o Apple II Plus (Unitron, Maxxi, Micro Engenho etc.), a linha 20 deve ser eliminada e a linha 320 alterada para:
320 LET D = INT(10*RND(1))

RND

Nos computadores compatíveis com o TRS-80 (CP 500, Digitus, etc.), pode-se eliminar a linha 20 e substituir a linha 320 por:
320 LET D = RND(6)

Nos computadores compatíveis com o Sinclair (TK83, TK85, CP 200, Ringo etc.), o termo RANDOMIZE está abreviado no teclado como RAND.

INKEYS

Nos computadores compatíveis com o Apple II Plus (Unitron, Micro Engenho, Maxxi etc.), deve-se substituir o comando INKEY\$ por GET. Assim, no primeiro programa do texto, substituir a linha 40 por:
40 GET A\$
A linha 50 do mesmo programa pode ser substituída por:
50 GET A\$: IF A\$ = " " THEN GOTO 80

■ **Loop e média.** Acrescentar linhas ao último programa do texto, de modo a fazê-lo repetir 100 vezes e apresentar o valor médio dos 100 resultados.

■ **Substituir com sub-rotina.** Substituir as linhas 50 e 130 do programa principal (a sub-rotina lançamento de dados) por uma instrução GOSUB que chame seu "gerador de números aleatórios" no primeiro exercício.

■ **INKEY\$.** Usando esta função, escreva um programa que leia qualquer tecla digitada e imprima: A TECLA PRESSIONADA FOI: * (* representa a tecla que você pressionou).

■ **Loop de controle de tempo.** Escreva um loop de controle de tempo (loop de cronometragem) e utilize a função INKEY\$ para saber o valor alcançado por uma variável após 10 segundos (observe no relógio). Desenvolva o programa de modo que a leitura final seja:
VALOR DE R APOS 10 SEGUNDOS E: * (* representa o valor de R).

■ **Testes IF-THEN.** Escreva um programa de jogo simples em que o computador dê origem a um número aleatório entre 1 e 100 (inclusive) e o jogador tenha de adivinhar o número. O jogador faz cinco tentativas. Em cada vez, o programa responde com as mensagens: SEU PALPITE ESTA MUITO ALTO, SEU PALPITE ESTA MUITO BAIXO, VOCE ACERTOU, ou SEM DIREITO A OUTRA TENTATIVA. VOCE PERDEU.

Respostas no próximo fascículo.

Respostas aos "Exercícios" das pp. 148-149

Loop 1
O VALOR DE A E 450

Loop 2
INICIO
FIM

Loop 3
O VALOR DE A AGORA E 160

Loop 4
GOSTO DE BASIC
GOSTO DE BASIC
GOSTO DE BASIC
:
:
:
:
Até que as teclas RESET ou BREAK sejam pressionadas.

Loop 5
CHEGA DE LOOPS
(15 vezes)

Read-Data 1
ESTAMOS TESTANDO A INSTRUCAO READ
170

Read-Data 2
X = 1
X = 2
:
:
:
X = 23

Sobre duas rodas

A tecnologia dos robôs apresenta rápido desenvolvimento. Alguns podem até ser programados para detectar obstáculos.

Robôs e tartarugas que se movem não são apenas educativos — são divertidos. Os princípios envolvidos no controle de dispositivos como o equipamento inglês BBC Buggy são os mesmos necessários aos robôs industriais. Embora efetivamente ainda não executem trabalho caseiro, podem vir a tornar-se a próxima geração de auxiliares domésticos.

Os robôs devem se localizar de modo preciso com relação ao seu ambiente e, por esta razão, são geralmente controlados por motores graduais. Ao contrário dos motores convencionais, os motores graduais não têm rotação completa ao receber energia de alguma fonte. Para cada impulso de energia, seu eixo gira em uma fração predeterminada da rotação completa. A quantidade de impulsos necessária para realizar uma volta completa depende de cada motor em particular. É possível também controlar a direção da rotação, fazendo com que robôs ou tartarugas se desloquem por distâncias determinadas de modo preciso, ao deixar o computador exercer o controle dos motores separadamente. Robôs podem, ainda, rodar no lugar pelo simples girar das duas rodas em direções opostas.

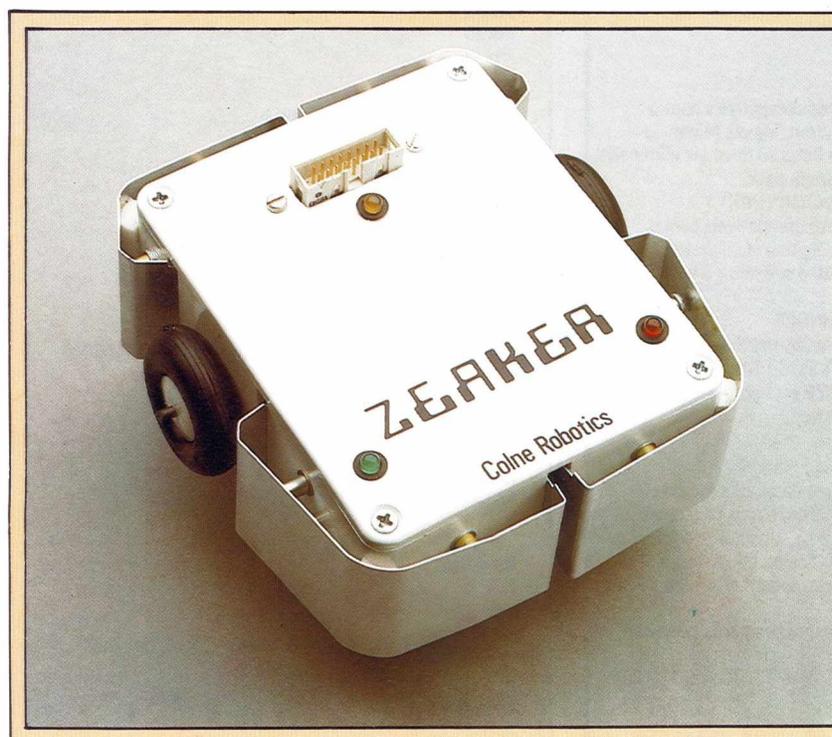
Todavia também é importante que o robô informe ao computador quando encontra um obstáculo. As colisões são geralmente detectadas por pára-choques montados no corpo do veículo e conectados a microinterruptores instalados em torno do equipamento. Os microinterruptores são, por sua vez, liga-

dos ao canal de entrada do computador e o fechamento e a abertura de cada interruptor fazem com que um dos bits mude para "0" ou para "1".

O robô muitas vezes precisa de outros tipos de entrada. Por exemplo, pode ser conveniente que ele tenha a capacidade de seguir linhas brancas sobre um piso negro. Isto é obtido pela adaptação ao robô de uma fonte luminosa que se dirija para baixo com um detector instalado junto a ela, para medir a quantidade de luz refletida. Essa quantidade varia de acordo com a superfície sobre a qual o robô está se deslocando e é analógica e não digital. O BBC Micro é equipado com uma saída analógica que possibilita o uso direto deste tipo de detector. A maioria dos outros equipamentos precisará primeiramente converter o sinal em digital, antes de enviá-lo ao computador.

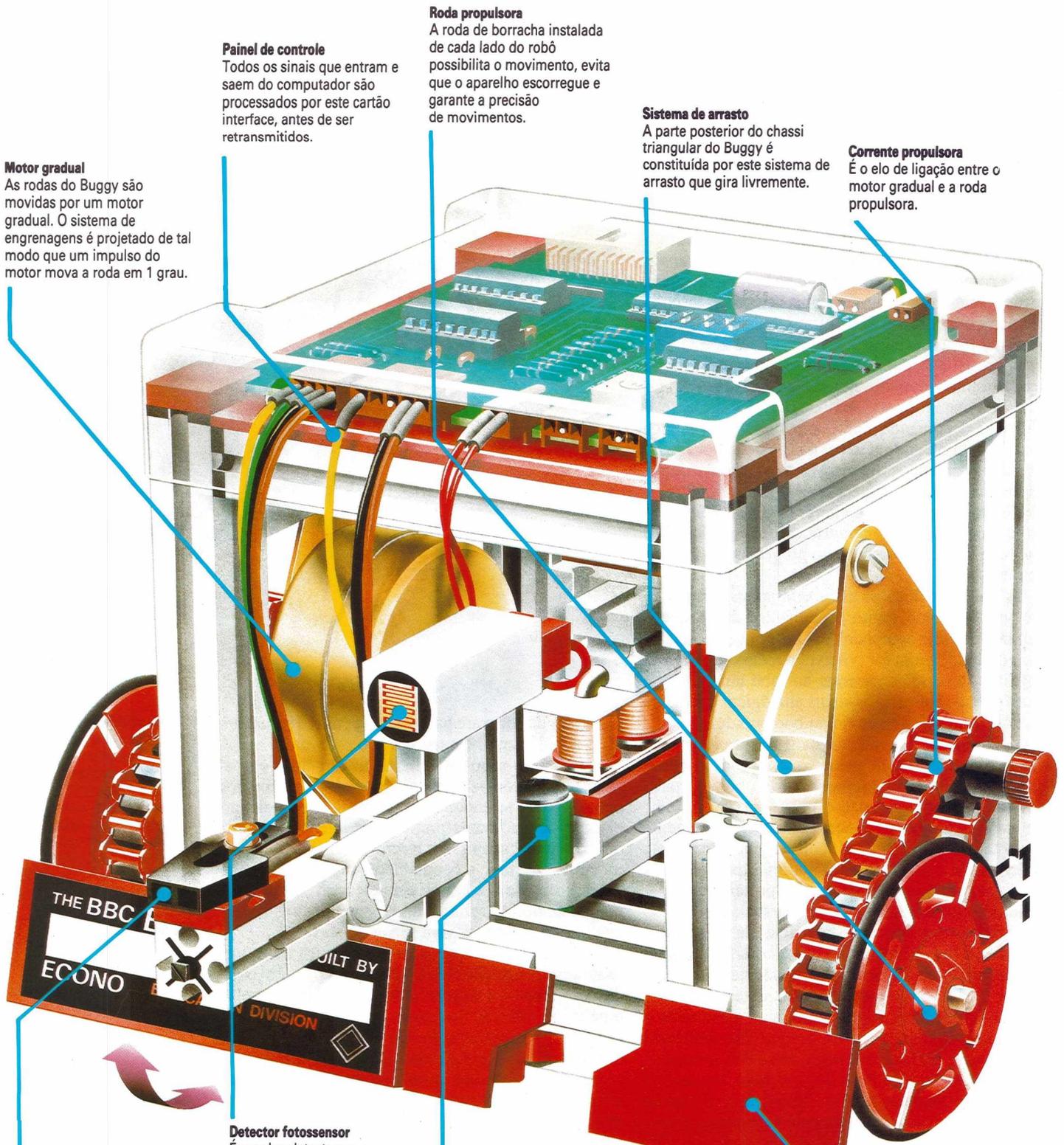
Esse tipo de detector também pode ser utilizado como leitor de código de barras. Os códigos que informam as características dos produtos armazenados em um depósito são, desse modo, verificados quando o robô busca o item correto. O BBC Buggy é equipado com software para demonstração que utiliza a leitora de código de barras para tocar música: os procedimentos são os mesmos.

Outros sinais analógicos que o robô é capaz de seguir são a luz, o som e os campos magnéticos. Esses campos são freqüentemente utilizados na indústria, onde os robôs têm de se deslocar por um trajeto fixo



A caminho

Essa "microtartaruga" está efetivamente a meio caminho entre o robô e a tartaruga, pois possui detectores de colisão e eventualmente, também, um certo grau de "inteligência". Este modelo pode mover-se para a frente, para trás, para a esquerda e para a direita, subir e abaixar a caneta. O controle torna-se muito simples pelo recurso a uma versão da linguagem LOGO, chamada SNAIL LOGO; desse modo, comandos como FORWARD e BACKWARD podem ser diretamente utilizados. Há computadores que necessitam de uma interface. Esta interface possui uma unidade de força e, deste modo, não sobrecarrega o suprimento energético do computador.



Motor gradual

As rodas do Buggy são movidas por um motor gradual. O sistema de engrenagens é projetado de tal modo que um impulso do motor mova a roda em 1 grau.

Painel de controle

Todos os sinais que entram e saem do computador são processados por este cartão interface, antes de ser retransmitidos.

Roda propulsora

A roda de borracha instalada de cada lado do robô possibilita o movimento, evita que o aparelho escorregue e garante a precisão de movimentos.

Sistema de arrasto

A parte posterior do chassi triangular do Buggy é constituída por este sistema de arrasto que gira livremente.

Corrente propulsora

É o elo de ligação entre o motor gradual e a roda propulsora.

Leitora de código de barras

É o segundo detector analógico. Consiste em um diodo emissor de luz (LED, Light Emitting Diode) e em um diodo detector de luz (LDD, Light Detecting Diode). A luz do LED é refletida na superfície debaixo do robô e detectada pelo LDD. A quantidade de luz refletida depende do material refletor e de sua cor; desse modo, o detector, além da leitura de código de barras, pode acompanhar linhas no solo.

Detector fotossensor

É um dos detectores analógicos do BBC Buggy. Ele envia ao canal de entrada analógico um sinal proporcional à quantidade de luz detectada; quanto maior a intensidade da luz que atinge o Buggy, maior será o sinal recebido.

Suporte da caneta

O BBC Buggy é equipado com um suporte de caneta adicional, entre as rodas, que o capacita a desenhar linhas na superfície que percorre. O suporte é movido para cima ou para baixo por controles do computador.

Detectores frontais de colisão

Cada lado deste pára-choque frontal está equipado com um microinterruptor que transmite ao computador um sinal positivo ou negativo de colisão.

no piso de um depósito ou de uma fábrica. Cabos especiais são instalados no chão, formando uma pista que o robô deve acompanhar.

O controle do robô é em geral realizado por um conjunto especialmente desenvolvido de rotinas de programa, que controlam o envio e a recepção de dados, através do cabo de ligação, entre o usuário do computador e o dispositivo. No caso do BBC Buggy, há necessidade de 4 bits para controle dos motores. Os dados são enviados pelo Buggy através do mesmo cabo. As saídas analógicas, tanto do sensor de luz como da leitora de código de barras, dirigem-se para o canal analógico e os dois detectores de colisão são conectados a duas outras linhas de entrada no canal para o usuário.

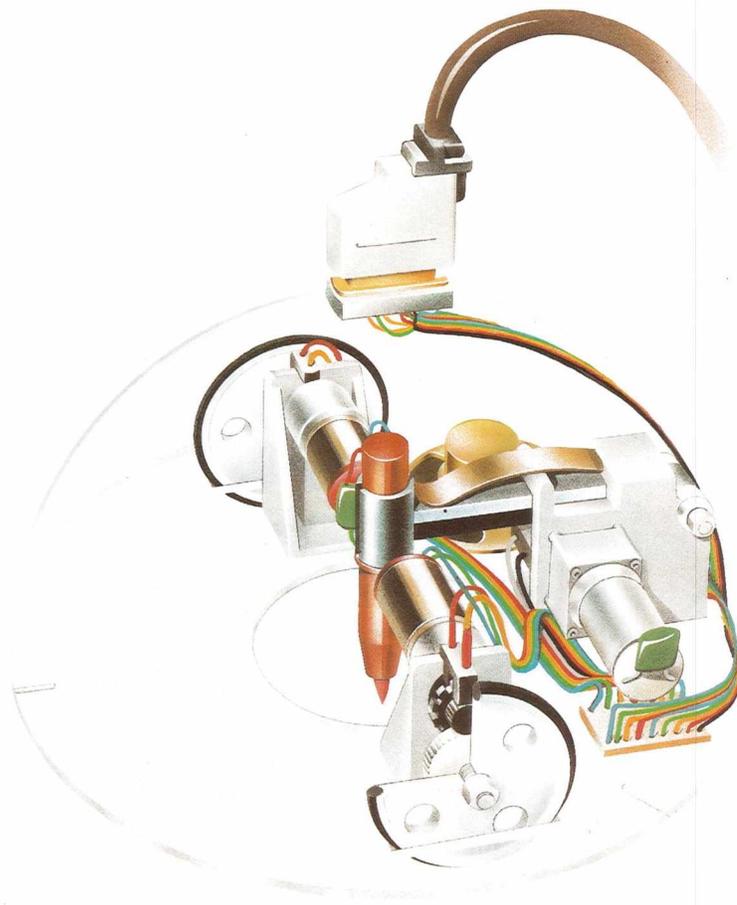
Um canal I/O, como o canal para o usuário, pode ser examinado pela observação de certa posição no mapa da memória do computador. Geralmente é utilizado o comando PEEK, em linguagem BASIC, para ler os conteúdos efetivos das posições. A fim de alterar os conteúdos, de modo a modificar a direção de um dos motores e fazer o robô girar, por exemplo, o programador deve alterar o valor do bit apropriado na posição da memória. É o comando POKE que realiza estas operações no programa em linguagem BASIC.

Os dados analógicos são examinados através de um procedimento muito semelhante, desde que o computador seja equipado com um conversor analógico/digital. Se não for possível o recurso a este dispositivo, deve-se acrescentar ao robô uma unidade interface que converta os sinais analógicos em dados digitais, antes de seu envio ao computador.

As tartarugas são, na realidade, um tipo de robô que se move, projetado para uso junto com a linguagem LOGO, embora a diferença entre robôs e tartarugas esteja se tornando menor a cada dia (ver p. 164): muitas das tartarugas lançadas mais recentemente são dotadas de detectores de colisão, enquanto robôs estão sendo equipados com suportes para canetas que os capacitam a funcionar como tartarugas. A idéia de criar desenhos em larga escala através do deslocamento, sobre uma superfície, de uma caneta adaptada a um equipamento com rodas tem origem no ensino das relações entre distância, ângulo e forma. Se uma pessoa avança 10 unidades, vira à esquerda e anda mais 10 unidades, vira à esquerda e anda outras 10 unidades, vira à esquerda e anda mais 10 unidades, ela terá executado um movimento que forma um quadrado. Para ilustrar essas formas e a relação que mantêm com o movimento, podemos equipar o robô ou a tartaruga com uma caneta e executar a figura no papel.

O BBC Buggy vem na forma de um kit para montagem, de modo que é preciso montá-lo antes de poder explorar o mundo dos robôs. O Buggy funciona acoplado ao equipamento Fischer Technik, que é encontrado à venda em muitos países e pode ser ampliado ou aprimorado com muita facilidade.

A montagem de um dispositivo como o Buggy é por si só um processo educativo. Aprende-se muito examinando o modo como as várias peças se encaixam. Todavia, o real aprendizado inicia-se quando o usuário experimenta tomar o controle de seu novo "brinquedo". Muitos dos robôs à venda vêm acom-



panhados de software para controle, mas é bem mais divertido desenvolver seu próprio software.

Enquanto o robô está em funcionamento, o programa deve constantemente controlar os sensores para verificar se encontrou, digamos, uma linha no piso, se detectou uma fonte de luz intensa, ou colidiu com uma cadeira. No momento em que o detector emite um sinal, o computador reage a fim de proteger o robô de um perigo em potencial. Os programas deste tipo são chamados "de tempo real", porque suas respostas devem ser imediatas.

Teoricamente, quase não há diferença entre um programa que possibilita ao robô mover-se sem colidir com objetos e aqueles projetados para controlar uma usina elétrica. As técnicas aprendidas ao brincar com robôs também ajudam a compreender o funcionamento da inteligência artificial. Podemos escrever programas que permitam ao robô desempenhar tarefas predeterminadas até que o sensor detecte que as baterias estão se esgotando; quando isso acontece, o robô deve buscar uma fonte de energia adequada para se recarregar, continuando assim a funcionar.

A próxima geração de robôs terá recursos ainda mais notáveis: provavelmente serão equipados com braços que lhes permitirão pegar e transportar pequenas cargas. Os dispositivos fotossensores também poderão ser substituídos por câmaras transistorizadas que permitirão ao robô "ver" o caminho que percorre. As unidades sintetizadoras de voz possibilitarão ao robô comunicar-se com seu operador e o reconhecimento da voz abrirá um novo canal de controle das atividades do robô.

Um robô dedicado

A tartaruga é um tipo de robô eficiente que executa desenhos no solo através de uma caneta hidrográfica retrátil, de acordo com as instruções do computador. Tartarugas geralmente estão vinculadas à linguagem pedagógica LOGO (ver p. 164), embora possam ser controladas também pela linguagem BASIC.



Números hexadecimais

Os computadores utilizam o sistema binário, pois admitem apenas dois estados. Por que, então, os programadores às vezes recorrem ao sistema hexadecimal, de base 16?

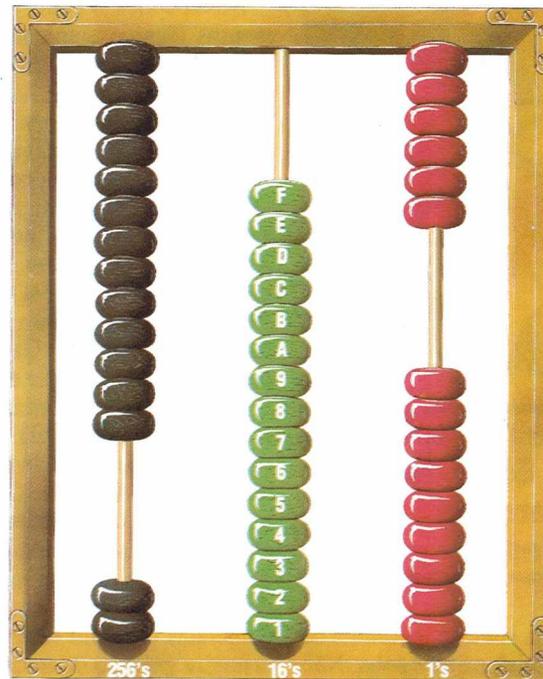
É fácil perceber por que se usa o sistema binário nas operações internas do computador — a representação dos números que utiliza unicamente os dígitos 0 e 1 presta-se bem aos sinais elétricos ligado/desligado com os quais o computador funciona. É igualmente fácil entender por que o sistema decimal é usado quase universalmente pelas sociedades humanas — o fato de termos dez dedos faz dele o nosso número base.

Que utilidade, porém, pode ter o sistema hexadecimal (de base 16)? A resposta está no fato de os números hexadecimais serem convertidos em números binários, e vice-versa, muito mais facilmente do que os decimais. Uma vez que os computadores “compreendem” os números binários, por que os programadores não fazem seus programas em dígitos binários? Há duas razões para isso. Primeiramente, os números binários são muito mais extensos do que seus correspondentes na base 10 ou 16. Por exemplo, o número 356 em decimal é 0000000101100100 no sistema binário. Em segundo lugar, é fácil cometer erros quando se usam apenas os dígitos 0 e 1.

O sistema hexadecimal utiliza 16 dígitos diferentes, inclusive o zero, de modo que o maior número que pode ser representado na coluna das unidades corresponde ao decimal 15. Os matemáticos poderiam ter adotado seis novos símbolos para os números entre 10 e 15, mas convencionou-se utilizar as seis primeiras letras do alfabeto. Ao realizar uma contagem pelo acréscimo de dígitos 1, recorremos a dígitos numéricos padrão juntamente com os dígitos alfabéticos até a letra F — o hexadecimal equivalente a 15. Uma vez esgotados os símbolos, temos de passar para uma outra coluna, a coluna dos “dezesesseis”. Aqui, igualmente, utilizamos os mesmos símbolos disponíveis até F. Após o preenchimento também desta segunda coluna, o acréscimo de mais um dígito significa iniciar uma terceira coluna, a dos “duzentos e cinquenta e seis” (256). O quadro apresenta alguns números decimais com seus equivalentes binários e hexadecimais.

O número 65.535 é representado por FFFF no sistema hexadecimal, e isto evidencia uma das vantagens do sistema hexadecimal: um número que no sistema decimal utiliza cinco dígitos, e no sistema binário precisa de dezesseis, pelo sistema hexadecimal necessita de apenas quatro dígitos. Além dessa economia de espaço, há outra vantagem mais importante. Quatro dígitos binários podem ser representados por apenas um dígito hexadecimal. Isto torna a conversão do sistema binário em hexadecimal, e vice-versa, uma operação relativamente simples.

A conversão de números binários em seus equivalentes hexadecimais consiste em dividir o número binário em grupos de quatro dígitos, iniciando pela



O ábaco hexadecimal

Um ábaco feito para contagem pelo sistema hexadecimal tem 15 contas em cada coluna. Na primeira coluna à direita, as contas são puxadas, uma por vez, até alcançar F, a última. O sistema hexadecimal é um sistema de numeração de base 16 que utiliza letras do alfabeto para números maiores que 9. Como mostra a coluna do meio, a letra A corresponde ao decimal 10, B ao 11, C ao 12, D ao 13, E ao 14 e F ao 15. Após o “número” F, os dígitos se esgotam e temos de passar para a outra coluna, a dos “dezesesseis”.

A coluna das unidades, no quadro, mostra uma contagem de 9 contas, a coluna dos “dezesesseis”, uma contagem de F e a coluna dos “duzentos e cinquenta e seis”, uma contagem de 2. Este número é lido como “2F9 hex” e equivale ao número decimal 761 ($256 \times 2 + 16 \times 15 + 1 \times 9$).

direita, e em seguida realizar a conversão, um grupo de cada vez. Eis aqui alguns exemplos:

	1110	1001	binário
	=E	9	hex
	=233		decimal
	1111	1000	binário
	=F	8	hex
	=3.980		decimal
0111	1110	0010	binário
=7	E	2	hex
=32.301			decimal

Quando seria realmente interessante utilizar a notação hexadecimal? Embora linguagens de alto nível como o BASIC não exijam a utilização do sistema hexadecimal (ou mesmo do binário), as linguagens de baixo nível, como código de máquina e assembly, o fazem.

Costuma-se colocar o símbolo H após o número hexadecimal para distingui-lo dos números decimais. Desse modo, o número 256 (decimal) seria escrito da seguinte forma: 100H e sua leitura seria “um, zero, zero, hex”.

Não teremos necessidade prática dos números hexadecimais em nosso curso. Entretanto, poderão ser mencionados em algum apêndice do manual de instruções de seu computador e, com estes esclarecimentos, você pode ao menos calcular os equivalentes no sistema binário ou decimal.

Binário	Decimal	Hex
00000001	1	1
00000010	2	2
00000011	3	3
00000100	4	4
00000101	5	5
00000110	6	6
00000111	7	7
00001000	8	8
00001001	9	9
00001010	10	A
00001011	11	B
00001100	12	C
00001101	13	D
00001110	14	E
00001111	15	F
00010000	16	10
00010001	17	11
00010010	18	12
00010011	19	13
00010100	20	14
00010101	21	15

Chuck Peddle



O cérebro que está por trás do microprocessador 6502, hoje utilizado em numerosos microcomputadores.

Chuck Peddle faz parte de uma geração anterior à de empresários como Steve Wozniak e Steve Jobs. Não obstante, só teve seu primeiro envolvimento com os microprocessadores em 1973, quando foi contratado pela Motorola para trabalhar no projeto do microprocessador 6800.

Como o 6800 foi um dos primeiros microprocessadores a disputarem o mercado, a Motorola estabeleceu por ele um preço alto. Esse é o procedimento econômico normal, quando a procura é maior que a oferta. Descontente com o preço demasiadamente elevado do produto, Chuck Peddle abandonou a Motorola e passou a trabalhar para a MOS Technology.

Nessa empresa, Peddle empenhou-se no projeto de outro microprocessador, o conhecido MPU 6502, sem dúvida o mais bem-sucedido microprocessador da primeira década da microcomputação. No entanto, nessa época ninguém poderia perceber que o produto então em desenvolvimento estava destinado a servir de esteio a toda uma indústria, contribuindo em larga escala para uma revolução social como não ocorria já há cerca de duzentos anos.

Uma das poucas pessoas capazes de reconhecer o significado dos microprocessadores em geral e o potencial do 6502 da MOS Technology em particular foi Jack Tramiel, presidente da Commodore Business Machines. Até esse momento, a Commodore estava envolvida com uma série de produtos para escritório e calculadoras de bolso, não tendo ainda obtido sucesso significativo em qualquer dessas áreas.

A Commodore era a principal cliente da MOS Technology, e comprava regularmente grande quanti-

dade de chips com quatro funções calculadoras. Apesar das dificuldades financeiras que enfrentava para manter a Commodore, Tramiel realmente acreditava no 6502. Tanto assim que encontrou recursos para adquirir a MOS Technology. Juntamente com a empresa, ele obteve os serviços de seu funcionário Charles Peddle, então alocado como engenheiro no desenvolvimento de microprocessadores.

Nessa época, Peddle percebeu que seus conhecimentos técnicos, somados a sua imaginação, poderiam dar origem a um grande acontecimento — o computador pessoal. A mesma idéia estava sendo desenvolvida por Wozniak e Jobs, na Apple Computer, Inc. Peddle estava tão preocupado com o uso correto dessa nova tecnologia, que se reuniu a Bill Gates, fundador da Microsoft — famosa por seu interpretador BASIC —, numa tentativa de comprar a Apple, que coincidentemente estava à venda ao mesmo tempo que a MOS Technology. Entretanto, Wozniak e Jobs pediam 150.000 dólares pela companhia, mas Peddle e Gates só tinham condições de pagar dois terços dessa quantia.

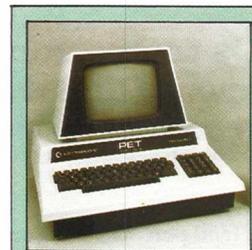
Peddle ficou na Commodore e comprometeu-se a criar o Commodore PET (Personal Electronic Transactor), que foi lançado em 1977, quase ao mesmo tempo que o Apple II. O PET era diferente por ter um monitor e uma unidade de fita acoplados, e o teclado era mais parecido com o de uma calculadora do que com o de uma máquina de escrever. Pouco depois de seu lançamento, a Commodore tinha encomendas para mil unidades do PET. Começava a primeira geração de microcomputadores projetados especialmente para uso pessoal.

Isso aconteceu três anos antes que Peddle realizasse sua segunda e maior ambição — controlar sua própria empresa de computadores. Juntamente com Chris Fish, um dos responsáveis pelo notável sucesso financeiro da Commodore, Peddle uniu-se à Victor United, uma subsidiária da Walter Kidde Corporation, e deu início à Sirius Systems Technology.

O trabalho de desenvolvimento na indústria do computador pessoal concentrava-se em chips de 16 bits como o 8088 da Intel. A IBM também estava trabalhando num computador pessoal baseado no mesmo chip, mas a Sirius conseguiu apresentar o resultado de seu trabalho algumas semanas antes. A máquina recebeu amplos elogios e estabeleceu-se firmemente no mercado. Foi a primeira produzida em massa, pois o preço mais baixo do microcomputador constituía vantagem apreciável em relação a muitos de seus concorrentes.

O Sirius I era relativamente barato e fácil de usar. Possuía um teclado separado, gráficos de alta resolução e tela anti-reflexo. Com tudo isso, ele colocava em cena novos padrões de microsistemas para escritório. Os usuários descobriram o grande aumento de velocidade e da capacidade de empregos do microprocessador de 16 bits. Era, sob qualquer ponto de vista, um aparelho muito eficiente.

Em tudo que fez, Chuck Peddle sempre procurou satisfazer sua aspiração — colocar o computador ao alcance de todos. E, de algum modo, ele incentivou outros fabricantes a que também o fizessem.



Primeiro computador pessoal
Depois de desenvolver o microprocessador 6502, Chuck Peddle empenhou-se em projetar algo chamado "computador pessoal", inteiramente autônomo, que pudesse ser ligado e usado para qualquer propósito, onde quer que o usuário estivesse. O Commodore PET foi o resultado de seu trabalho, lançado quase ao mesmo tempo que o Apple II de Wozniak. Tinha tela incorporada, gravador cassete e um interpretador BASIC Microsoft. Embora tenha passado por várias modificações, como o aspecto do teclado, o PET ainda é bastante popular. Uma das características da máquina é seu aspecto físico, que sugere uma cabeça e dois ombros. Atualmente, o 6502 é um dos microprocessadores mais utilizados em computadores pessoais.



Imagens animadas

Multiplique por mil os recursos gráficos de seu micro e você terá um sistema de animação por computador.

A retina do olho humano é incapaz de perceber os intervalos de fração de segundo existentes entre os quadros fotográficos fixos (fotogramas) que compõem um filme de cinema ou de televisão. A impressão de movimento contínuo pode assim ser criada com a exibição de uma rápida sucessão de imagens.

As primeiras tentativas para produzir essa ilusão de movimento foram feitas com a introdução de tiras de desenhos nas fendas de um cilindro e a posterior rotação desse cilindro. Olhando pelas fendas, via-se uma representação grotesca de uma “figura” depois da outra.

O praxioscópio, como era chamado, antecedeu a descoberta da fotografia, e, naturalmente, fotos logo substituíram os desenhos no interior do cilindro. Na etapa seguinte, o cinematógrafo requeria uma relativa rapidez de ação das emulsões fotográficas, capazes de registrar uma imagem em fração de segundo, pois os primeiros filmes foram projetados com dezesseis imagens por segundo.

A simulação do movimento

Curiosamente, a idéia do desenho animado de cada imagem foi concebida bem antes do filme industrial; os desenhos eram fotografados e projetados, quadro por quadro, proporcionando desse modo a simulação “animada”.

Levando em conta que a visão em cada segundo requer a criação de 24 desenhos separados (a velocidade de produção do filme moderno), é claro que a produção de até mesmo um filme de 5 minutos exige bastante trabalho — 7.200 imagens, neste caso. Não é de estranhar que o estilo da ilustração seja formalizado — a exigência mais importante é a repetição precisa. Não adiantaria haver um Pato Donald diferente a cada segundo.

Tarefas repetitivas e cansativas como estas são facilmente realizadas por máquinas. Quando o computador cuida da animação — isto é, ajuste da velocidade do movimento, mudanças de perspectivas e de formatos, luz e sombra, mudanças de volume e ritmo —, o artista fica livre para se concentrar na qualidade da imagem. Deste modo, a animação passa a ser uma verdadeira arte gráfica, pois o artista ocupa seu tempo criando a imagem a ser trabalhada pelo computador.

Basicamente, esse processo usa “sprites” (ver p. 152) para criar o “elenco de personagens”, que, transferidos para a tela e movimentados, produzem a animação dos jogos. Para simular mudança e movimento (por exemplo, alguém andando), é necessário o deslocamento de um desenho para outro.

Como já vimos, a criação dos desenhos é um processo relativamente lento, dada a qualidade gráfica dos resultados, e a imagem nada mais é do que uma simples representação bidimensional.

A etapa seguinte da animação requer que o animador/programador construa um algoritmo que dê à imagem sensação de profundidade, de acordo com as regras da perspectiva. Os objetos podem ser definidos na tela em termos de suas coordenadas X, Y e Z. Nesta etapa, é extremamente importante que o

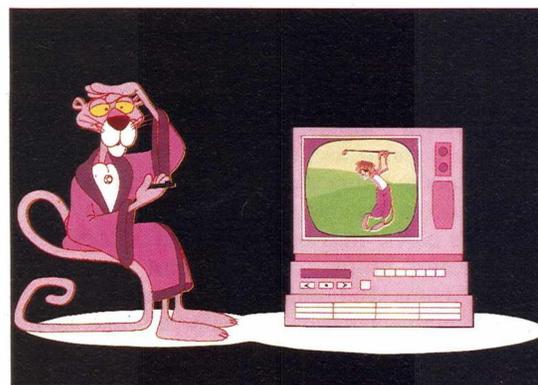


Imagem por imagem

A animação convencional, como estas imagens da “Pantera cor-de-rosa”, requer que o artista desenhe cada figura separadamente; as características comuns não precisam ser redesenhadas, a menos que haja uma mudança de posição ou de aspecto. O filme transparente é usado para que uma imagem completa seja feita através de uma série de sobreposições. O artista irá concentrar-se nas imagens principais da seqüência, deixando as seções secundárias para serem realizadas por assistentes conhecidos como “intermediários”. Fotografam-se os desenhos prontos com uma câmara na ordem em que serão vistos.

programa não reproduza linhas “escondidas” e gere áreas opacas no que era até então um esqueleto de modelo ou representação.

A suavidade das curvas também constitui um refinamento na produção de uma seqüência. Uma linha curva é especificada por três pontos: as duas extremidades e o ponto mais distante da linha reta que se pode traçar entre eles.

É claro que numa curva menos simples (um “S”, por exemplo) torna-se necessário separar seus componentes para que se possa trabalhar adequadamente; e é importante haver um modo simples de indicar à máquina que se trata de uma curva que requer suavidade, e não uma linha angulosa.

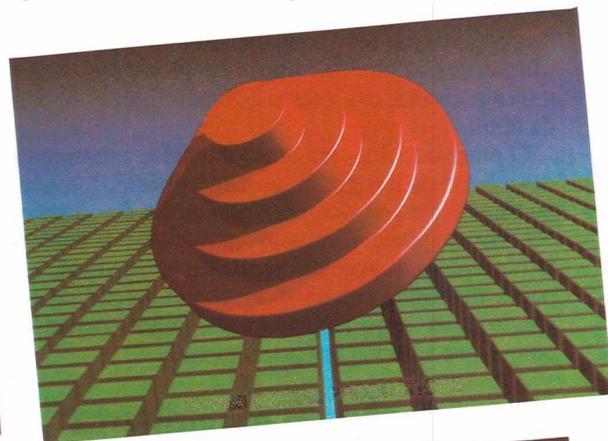
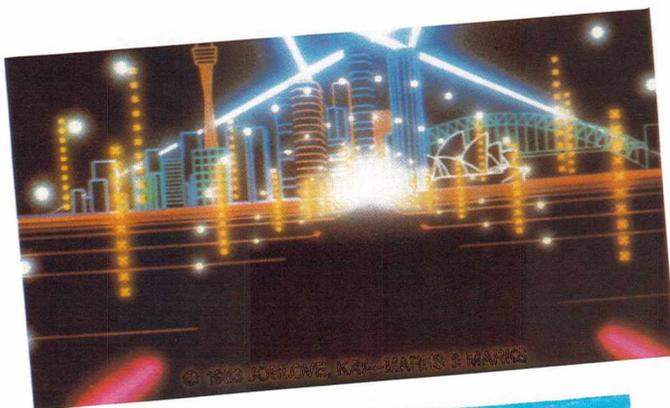
Cor e luz

A seguir vem a habilidade de se introduzir luz e sombra no desenho. Primeiramente, é necessário que se especifique a posição da fonte de luz. A parte do objeto desenhado que fica de frente para a fonte será mais iluminada, e uma sombra progressiva ajudará a

cores. O sombreado e o colorido combinam-se num único aspecto.

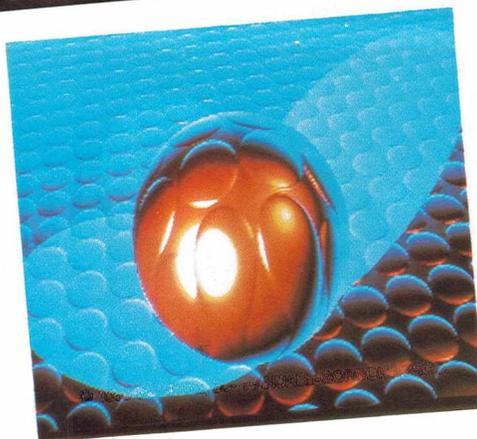
Vamos agora examinar a simulação do movimento. É relativamente fácil reduzir um movimento a seus diversos componentes, se pensarmos na figura como um problema de geometria sólida contínua, mesmo quando o objeto representado é tão complexo como o corpo humano.

O fator determinante será o tamanho e a capacidade do computador utilizado. Entretanto, é preciso ter em mente que para produzir uma imagem de boa qualidade precisamos de um monitor que apresente nitidez de 1.000 x 1.200 “pixels”. Cada um desses pixels necessita de, ao menos, 1 byte de 8 bits para reter a informação que define sua cor e brilho. Isso significa que há necessidade de mais de 1 megabyte por tela. Sendo assim, a criação de filmes de alta qualidade não é viável em computador doméstico. Na verdade, os animadores profissionais usam alguns dos maiores e mais potentes computadores do



Caixa de cores mágicas

Até recentemente, o trabalho artístico de cenas paradas do cinema e TV e das seqüências animadas seguia o sistema usado em revistas: o desenho era feito num papel de filme transparente, e só então fotografado. O sistema Pintura de Caixa da Quantel, entretanto, elimina o uso do papel, compondo o trabalho artístico de modo digital no computador; a seguir, esse trabalho é gravado diretamente em vídeo-tape.



definir a forma do objeto. Softwares mais sofisticados permitem o uso de outras fontes de luz e superam o problema do reflexo de luz produzido por um objeto em outro.

Assim como o uso da sombra, o emprego das cores também é importante. Até mesmo os microcomputadores bastante simples já oferecem oito ou dezesseis cores, mas os computadores com qualidade gráfica profissional permitem pelo menos 4.096. Alguns são limitados simplesmente pelo número de dígitos binários da “palavra” do computador. Se esse número é de 24 bits, por exemplo, o computador tem cerca de 16,7 milhões de opções de

mundo, com altos honorários por segundo de filme produzido.

Se tomarmos um simples objeto, um cubo, por exemplo, compreende-se facilmente como é possível movê-lo na tela, rolando-o, talvez. Um cubo pode ser definido pelas coordenadas dos oito ângulos, e este mesmo princípio se aplica a objetos mais complexos. As únicas diferenças são a quantidade de memória necessária para armazenar todas as coordenadas e a capacidade do processador de manipular a informação com rapidez suficiente para gerar o movimento no tempo exato.

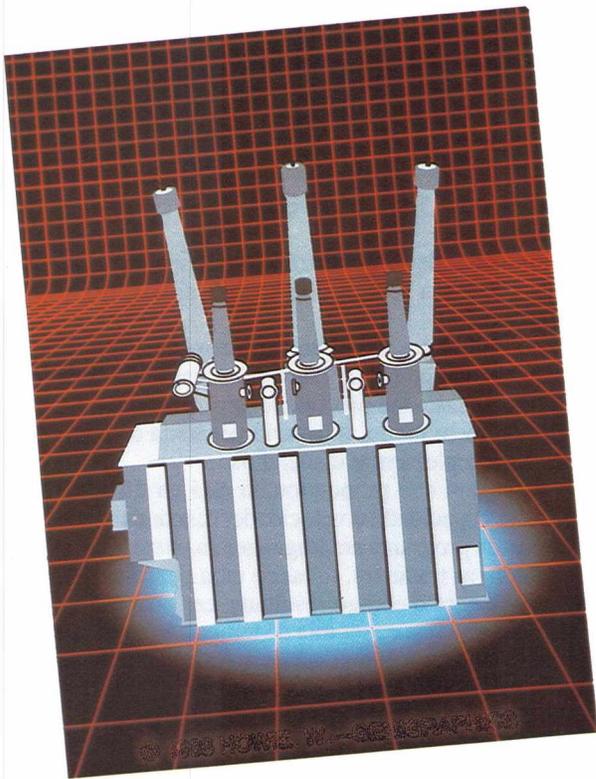
Nesta aplicação, como em todas as outras, deve haver uma combinação entre a qualidade da imagem e a quantidade disponível de espaço. Quanto menor a unidade desenhada, maior será a necessidade de



memória. Mas é essencial trabalhar nos menores detalhes para se alcançar alta qualidade de reprodução.

Um aparelho de TV — com cerca de 525 x 1.000 pixels — é bem menos refinado se comparado com o potente monitor de que falamos. Assim, podemos estar certos de que um trabalho detalhado parecerá tão real quanto qualquer imagem de televisão. Mesmo usando apenas as técnicas de que dispomos, criamos uma imagem precisa da realidade através de figuras animadas.

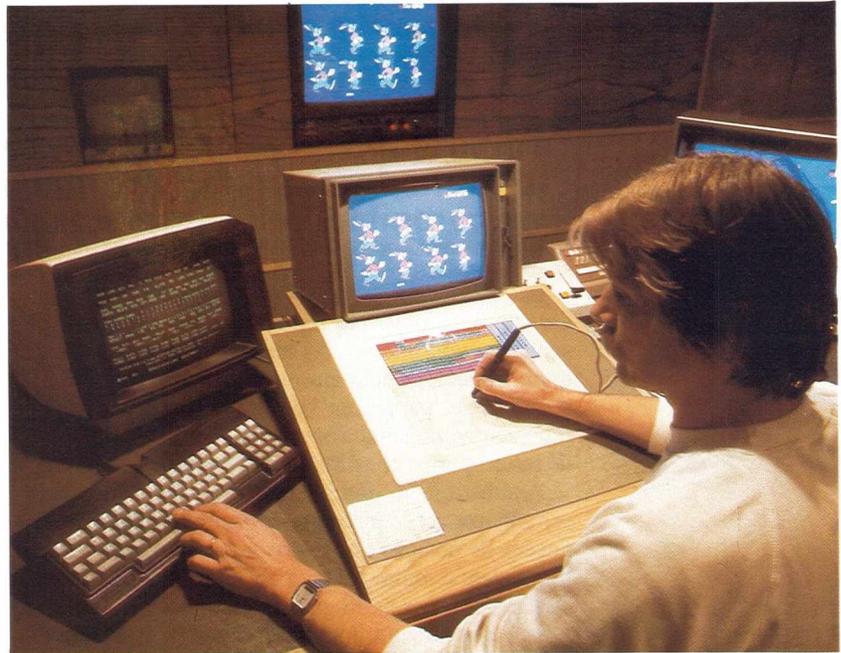
Para elaborar a imagem com esses detalhes, é necessário que se tenha um software sofisticado e um hardware essencialmente feito para este fim. O método mais conhecido utiliza um aparelho denominado digitalizador “bit-pad”, semelhante a um grande quadro de desenhos que contém uma matriz



de fios. Esta malha é usada para perceber a posição do ponteiro que a cruzou. O computador então mostra na tela do monitor a linha ou ponto resultante. É possível traçar através de um trabalho artístico já existente, desenhar a mão livre ou usar instrumentos convencionais de desenho.

A imagem é digitalizada (com as coordenadas X-Y ativadas), escrita na memória e mostrada pelo computador. A característica do traço que aparece na tela pode ser definida pelo usuário, como se ele escolhesse usar um lápis, uma caneta ou um pincel. Do mesmo modo, define-se a cor com a ajuda da paleta — um conjunto de cores que aparece na parte de baixo da tela, semelhante à paleta de um pintor. Se alguém não quiser uma cor homogênea, é possível fazer misturas, exatamente como se usasse tinta. As imagens podem ser apagadas, e desse modo os “desenhos” sobrepõem-se.

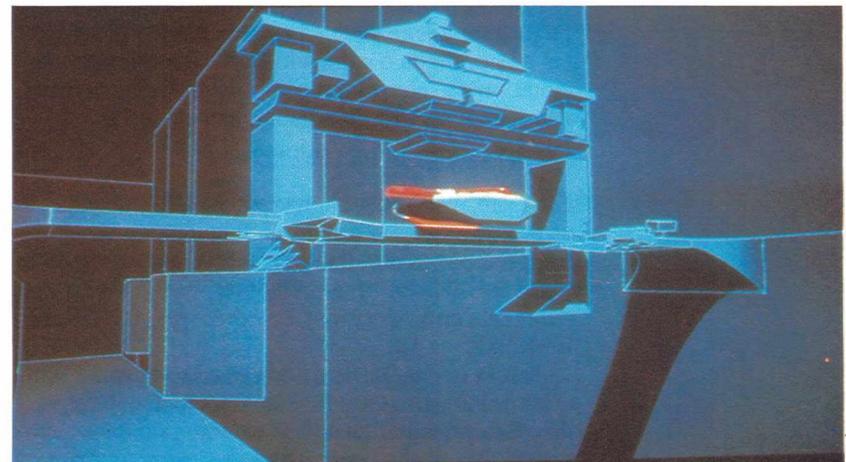
Depois da criação de uma simples imagem, como fazer para que ela se mova? Basta que se utilize o



processo convencional, usando o sistema de computador para arquivar material, colorir as imagens e talvez mostrar, grosso modo, as seqüências prontas. Este método irá acelerar a tarefa, mas técnicas de programação mais avançadas tornarão o desenvolvimento ainda maior. Do mesmo modo que as curvas podem ser automaticamente amenizadas, blocos inteiros de ação são fabricados; basta apenas que a primeira e a última imagem da seqüência sejam especificadas.

Este processo, conhecido como “tweening”, é realizado pelos assistentes de um estúdio convencional de animação, cujo trabalho o computador substitui. Como dissemos anteriormente, a introdução dos computadores no processo da animação dá liberdade ao artista para se concentrar na qualidade da imagem. A maior parte do esforço feito pela mão dos animadores fica na criação da ilusão de movimento, e esta tarefa pode ser realizada de forma precisa pelo computador.

Uma vez que as regras sejam corretamente estabelecidas e observadas, obtém-se o resultado desejado. Eis aí a indicação de que este é um trabalho adequado ao computador.



Ver para crer

Com o processamento ultra-rápido e a enorme capacidade de armazenamento dos computadores modernos, é possível criar no filme ou na tela de TV uma imagem totalmente indistinguível da fotografia. Usando técnicas de programação para solucionar problemas numéricos, pode-se manipular as imagens, de tal modo que o espectador acredite que elas são reais.

Mistura bem dosada

A Walt Disney Productions foi a primeira a utilizar no cinema técnicos de computador e de imagens programadas, com o filme “Tron” (de TRace ON). Tendo parte de sua realização baseada na realidade e parte desenvolvida em um enorme computador, “Tron” mistura imagens animadas por computador e fotografias de efeitos especiais, criando um estonteante mundo de fantasia.

Tradução alternativa

Computadores “pensam” em código de máquina. Programadores preferem linguagens como o BASIC ou PASCAL. E os compiladores e interpretadores fazem a tradução.

Os computadores não tinham teclado no início de seu desenvolvimento. As instruções dos programas eram fornecidas pelo ajuste de cada um dos oito interruptores “para cima” ou “para baixo”, a fim de representar uma única operação. Essas combinações de interruptores constituíam um código de máquina.

A substituição dos interruptores por um teclado semelhante ao de máquina de escrever e das combinações de posição dos interruptores por palavras da linguagem comum foi uma consequência lógica, da qual resultaram as linguagens de alto nível, como BASIC, que substituíram os códigos de máquina de baixo nível.

Como processadores, todavia, os computadores não se modificaram; ao contrário, continuaram a funcionar no antigo sistema de interruptores (e ainda o fazem); desse modo, os programadores tinham de desenvolver programas na notação original de baixo nível, a fim de traduzir os de alto nível em combinações que os processadores poderiam operar. Esses programas de baixo nível tornaram-se conhecidos como interpretadores e compiladores, conforme seu método de tradução.

Em computação (como em tudo o mais), qualquer aumento em capacidade ou velocidade tem um custo — em dinheiro, tempo ou liberdade de ação. O mesmo acontece com os interpretadores e compiladores. Juntos, proporcionam ao programador todos os recursos para a tradução de programas. Os interpretadores oferecem vantagens em alguns aspectos e os compiladores em outros, mas também apresentam desvantagens.

Os interpretadores, geralmente incorporados ao microcomputador, são um método barato de tradução de programas de linguagem de alto nível para uma linguagem que o computador possa compreender. Não utilizam muita memória, o que deixa mais espaço para os programas.

A maioria dos micros possui um interpretador em linguagem BASIC: você dá entrada em um programa em BASIC, digita o comando RUN e o programa funciona ou se interrompe pela apresentação de uma mensagem de erro do sistema, semelhante a:

```
SYNTAX ERROR ON LINE 123
```

Com isto, você deve digitar a tecla LIST, detectar o erro, corrigi-lo e o programa funciona ou se interrompe, e assim por diante. Observe que alguns dos mais sofisticados interpretadores BASIC verificam realmente erros sintáticos à medida que cada linha é fornecida.

Talvez isso tenha ocorrido centenas de vezes sem que você imaginasse o funcionamento do interpretador. Sua principal qualidade está no fato de ser um dispositivo oculto, que permite ao usuário operar o

programa sem se preocupar com sua localização na memória, ou como executá-lo: o programa encontra-se ao alcance de seus dedos e você pode imediatamente teclar RUN, LIST ou EDIT.

O interpretador é de uso fácil e não muito sofisticado: quando você digita o comando RUN, o interpretador deve localizar o programa em linguagem BASIC na memória, traduzi-lo e executá-lo, linha por linha. Se seu programa contiver o seguinte loop:

```
400 LET N = 0
500 PRINT N
600 LET N = N + 1
700 IF N < 100 THEN GOTO 500
```

o interpretador deverá traduzir e executar cem vezes as linhas de 500 a 700, como se jamais as tivesse encontrado antes.

Os compiladores são diferentes: caros, difíceis de ser escritos, ocupam e utilizam muita capacidade de memória. Quase sempre funcionam com software em disco; desse modo, o usuário necessita de equipamento complementar dispendioso.

A vantagem que proporcionam está na flexibilidade, alta capacidade e velocidade; diante das quatro linhas de programa em linguagem BASIC acima, o compilador as traduzirá, todas de uma vez, e em seguida executará o código resultante cem vezes.

Isto permite boa economia de tempo. Admitamos que você tenha um compilador em linguagem BASIC e queira processar um programa em BASIC.

Primeiramente, você deve carregar um programa de criação de arquivos (chamado Editor), que lhe permite digitar o programa e retê-lo em disco, como um “arquivo fonte”.

Os arquivos devem receber denominações que permitam sua localização (exatamente como os arquivos de um escritório); assim, o Editor solicita a digitação do nome do arquivo fonte. Os nomes de arquivo têm em geral duas partes: a primeira consiste em um nome qualquer que você escolhe — digamos, MEUPROG — e a segunda, em um código de três letras, que indica o conteúdo do arquivo; este código é a “extensão”. Um arquivo em BASIC possui o código BAS como extensão. Seu arquivo fonte está agora no disco, sob o nome MEUPROG.BAS. Agora a digitação de

```
COMPILE MEUPROG.BAS
```

induz o computador a carregar e processar o compilador, utilizando um arquivo fonte em BASIC chamado MEUPROG.BAS.

Você aguarda alguns segundos, de acordo com a extensão de seu programa, enquanto o compilador o traduz em um “arquivo objeto”, que é retido em



Compilação de programas

Escrever programas compilados não é tão simples como utilizar o interpretador. Todavia, o processamento será muitas vezes mais rápido. Esta é a seqüência principal de procedimentos.

Carregar o programa "Editor"

Digitar seu programa em linguagem BASIC.

Refer seu programa como um arquivo fonte em cassete ou em disco.

Carregar o programa compilador e dar o nome do arquivo fonte.

O compilador produz um arquivo objeto — uma versão do seu arquivo fonte em código de máquina.

Se houver erros no programa, serão indicados, e o arquivo fonte deve ser reeditado.

Quando finalmente não houver erros, o arquivo objeto pode ser carregado e processado diretamente no equipamento.

disco, sob o nome de MEUPROG.OBJ — a extensão OBJ indica que este é o arquivo objeto, uma tradução para o código de máquina de um arquivo fonte.

Enquanto traduz o arquivo, o compilador verifica possíveis erros sintáticos. Ao encontrar algum, emite uma mensagem semelhante às seguintes:

```
100 REED X:IF X=3(N+2) LET P=Q
      1           2     3
```

FATAL ERROR:-

- 1) //REED// UNRECOGNISED COMMAND
- 2) /// ILLEGAL OPERATOR HERE
- 3) ?? "THEN" OR "GOTO" EXPECTED HERE

Você receberá este tipo de mensagem para cada linha que contiver erro. Ou seja, a indicação de erro é muito mais abrangente do que em um interpretador BASIC. Então você deve carregar e processar novamente o Editor, chamar outra vez o arquivo fonte retido no disco, efetuar as alterações e tentar nova compilação. Se não mais restarem erros, digite

RUN MEUPROG

e ele funcionará ou não como o esperado. Nesta fase, corrigidos os erros sintáticos, você pode, ainda, querer alterar o programa; deverá então carregar e processar o Editor, modificar o arquivo fonte, recompilá-lo, e assim sucessivamente.

O bom desempenho do compilador não é percebido de modo claro no estágio de desenvolvimento do programa, embora a informação de erros tenha muita utilidade. O valor dos compiladores se evidencia quando você está com um programa sem mais correções a fazer e digita o comando RUN.

Os programas compilados são rápidos — cinco a cinquenta vezes mais que programas interpretados, conforme a eficiência do compilador; mas a maior velocidade de execução de programas compilados obtém-se à custa da menor velocidade no desenvolvimento do programa.

A comparação entre compiladores e interpretadores pelo contraste de seqüências habituais de comandos para o usuário, como as acima, não faz justiça aos compiladores, já que são projetados visando principalmente a equipamentos menos especializados e de maior porte, nos quais os usuários podem ter necessidade de escrever e processar programas em muitas linguagens diferentes.

A linguagem COBOL (para escrever programas comerciais, folhas de pagamento e estoques), por exemplo, foi inventada tendo em vista a compilação, enquanto o BASIC efetivamente foi desenvolvido para ser interpretado. Se quiser comparar dois veículos diferentes, você deverá testá-los tanto em estrada de terra como na asfaltada.

Uma vez elaborado e compilado o programa, você não precisará do arquivo fonte, a não ser para referência. Assim, o programa fonte pode ser comentado e desenvolvido objetivando a facilidade de leitura, enquanto o arquivo objeto é muito menor, ocupando menos espaço em disco e memória.

O fato de um arquivo objeto criado pelo compilador consistir em código de máquina ilegível é, surpreendentemente, uma vantagem. Se você for um vendedor de software, não venderá o arquivo fonte, apenas o arquivo objeto, mais difícil de plagiar, copiar ou alterar.

CHARACTER NAME	CLASS	AC	HITS	STATU
EINSTEIN	N-FIG	1	60	65
FUKUDD	G-PRY	1	47	45
DASH	N-FIS	1	50	50
BASH	G-MAG	2	24	24
ARLON	N-FTH	1	46	46

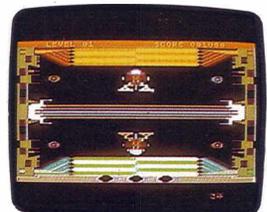
Lento

Em jogos de aventuras, a velocidade não é fundamental e a maior parte do programa consiste na manipulação de texto. Por isso, o programa pode ser desenvolvido em BASIC e interpretado à medida que é processado.

Command	Action
Print	Print
Input	Input
Open	Open
Close	Close
Read	Read
Write	Write
Append	Append
Seek	Seek
Locate	Locate
EOF	EOF
End	End

Mais rápido

Muitos programas contábeis são difíceis de desenvolver em código de máquina pois implicam grande quantidade de procedimento. Todavia, linguagens interpretadas seriam muito lentas; desse modo, esses programas geralmente são desenvolvidos em BASIC e então compilados.



Rapidíssimo

Para jogos de ação rápida, tipo fliperama, que envolvem a manipulação de gráficos, mesmo programas compilados não seriam suficientemente rápidos. Tais pacotes devem ser escritos diretamente em código de máquina — uma tarefa lenta e cansativa.

Conversa de amigo

Há algum tempo isso não passaria de ficção científica. Utilizando-se um sintetizador de voz e um alto-falante, seu micro responderá a muitas de suas perguntas.

Até 1984, a técnica do reconhecimento da fala não estava totalmente desenvolvida, mas sua reprodução eletrônica tornou-se possível graças à ajuda de acessórios adequados. Muitos microcomputadores e brinquedos eletrônicos já são capazes de falar. Os rápidos avanços da tecnologia e o menor custo dos componentes do aparelho fizeram do computador que fala algo bastante comum.

Quando as pessoas falam, três tipos distintos de sons são produzidos. O primeiro é o sonoro ou o som das vogais — *a, e, o*, e assim por diante. Estes sons produzem-se pela vibração das cordas vocais; a frequência dessa vibração determina o som da vogal. O segundo tipo de som é o das fricativas, como o *ch, f*, e o *s*. Neste caso, o ar vem dos pulmões, passa pelas

cordas vocais e a frequência do som é controlada pela posição dos lábios e da língua. O terceiro som é o de “explosão” ou, para ser mais exato, o som oclusivo que ocorre quando se interrompe um fluxo de ar. Desse modo emitem-se sons como *p, t, q*.

A criação de blocos de sons

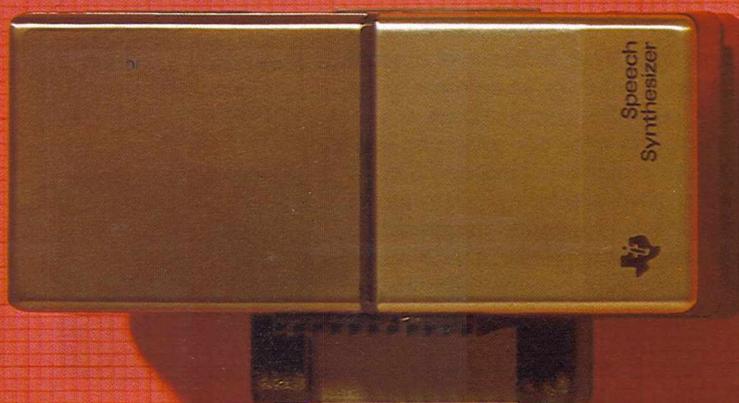
Existem dois modos de emitir os sons da fala eletronicamente. O primeiro, até há pouco o mais comum, é o processo da síntese. Através da análise das frequências da fala, é possível estabelecer regras que permitam a produção de um som pelos seus componentes. Por exemplo, a palavra *tio* poderia ser definida como muitos milissegundos da mistura de frequências que compõem o som *t*, seguido imediatamente das frequências *io*.

Esses blocos estruturados recebem o nome de fonemas, e, reunidos em várias combinações, qualquer palavra pode ser formada. As características individuais da fala do ser humano tendem a desaparecer quando se enuncia uma oração desse modo, por fonemas, mas os vocábulos são reconhecidos e compreendidos.

Pelo fato de as regras da criação de fonemas constituírem parte integrante do equipamento eletrônico, o usuário tem condições de registrar uma lista de fonemas no sistema. Estes se reproduzem por um pequeno alto-falante. Com um pouco de treino, é possível criar sentenças completas instantaneamente, usando-se seqüências de fonemas, que podem ser armazenadas com facilidade utilizando-se a linguagem BASIC.

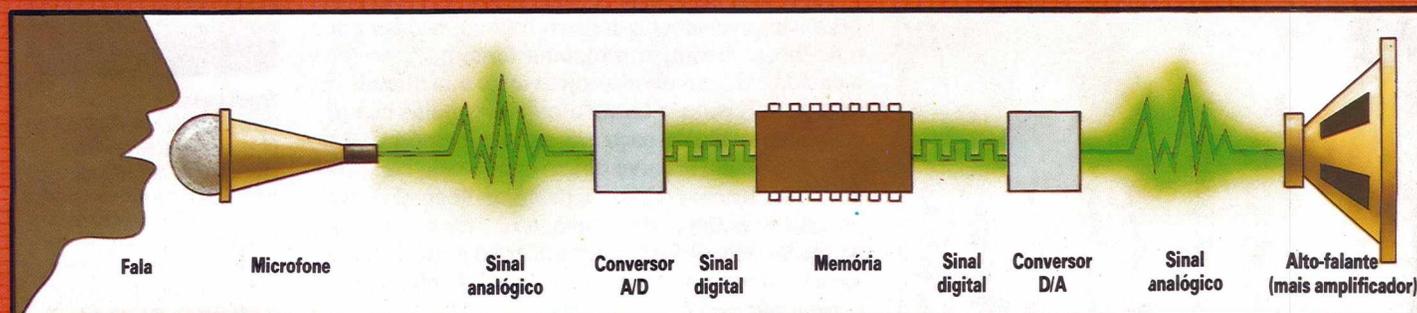
O segundo método de síntese recai sobre o ouvido e o cérebro humano, que preencherão os espaços. Por exemplo, a escala de frequências que pode ser transmitida pela linha telefônica oferece apenas a quinta parte da qualidade que se espera de um bom sistema de som; mesmo assim, a voz que escutamos no telefone é totalmente compreensível. Isto ocorre porque nossa mente completa os espaços que ficam.

O segundo método de síntese, que recebe o nome de “fala digitalizada”, utiliza o mesmo fenômeno.



O fluxo do som

A fala pode ser digitalizada e armazenada na memória RAM ou ROM. A saída elétrica de um microfone passa por um conversor analógico-digital. A saída deste chip é um padrão de dígitos 0 e 1. A fala é reconstituída por um conversor digital-analógico, um amplificador e um alto-falante.





Com a redução de custo da memória do computador, é possível agora converter a fala em informação digital através de um conversor analógico-digital. Os dados resultantes são então reduzidos centenas de vezes e armazenados numa ROM — criando assim espaços que o ouvido humano tem capacidade de preencher.

Para que qualquer palavra armazenada seja pronunciada, temos apenas de dar ao computador o endereço da palavra na ROM, e a informação digital é encontrada e rapidamente convertida em som. Como as palavras originais do falante estão armazenadas, as características pessoais permanecem as mesmas.

Alguns computadores possuem um hardware e um software em disco que permitem ao usuário digitalizar sua própria voz usando um microfone. Os dados resultantes são armazenados em disco (1 segundo de fala ocupa cerca de 1 Kbyte) e utilizados como mensagens verbais e avisos.

As utilidades dos sintetizadores de voz são tantas que se torna quase impossível enumerá-las. Para começar, a síntese da voz pode substituir os avisos gravados em estações ferroviárias, aeroportos e outros terminais. Nos Estados Unidos, a síntese é bastante usada no sistema telefônico, para informar quando a pessoa discou um número errado, e também em ligações de serviços privados. Muitos sistemas de discagem automática contam agora com a resposta sintetizada. O computador pode também informar o nível de estoque de uma mercadoria ou o período provável de entrega, para que o pedido seja modificado a tempo, se for o caso.

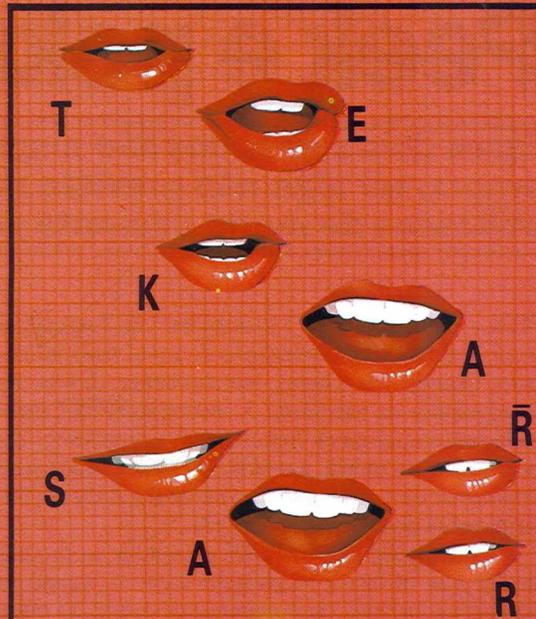
As unidades de síntese da voz também já estão presentes em alguns automóveis, como parte integrante de seu equipamento. Muito mais do que um mero apelo comercial, o sintetizador informa o motorista sobre eventuais problemas sem que sua atenção seja desviada do volante.

Certos fabricantes de câmaras fotográficas estão igualmente utilizando a síntese da voz. São equipamentos de revelação instantânea que fornecem instruções do tipo: "tire a tampa da máquina", "use o flash" ou "o filme acabou". E uma melodia é tocada enquanto se espera o processamento das fotos.

No mercado de microcomputadores e jogos eletrônicos, emprega-se a síntese da voz para realçar os jogos: o placar é falado e os avisos de ataques inimigos são dados verbalmente, deixando o jogador livre para concentrar-se nas táticas do jogo, em vez de ler mensagens na tela.

Por fim, existem dicionários de língua estrangeira que articulam as palavras à medida que elas aparecem na tela do computador. Vários instrumentos educacionais foram desenvolvidos pela Texas Instruments. O Speak 'n' Spell está entre os mais conhecidos e poderia ser descrito como um soletrador. Cada módulo de 128 Kbits armazena duzentas palavras diferentes, que, depois de enunciadas pelo equipamento, devem ser soletradas no teclado.

O Magic Wand, outro brinquedo educacional da Texas, ensina crianças a ler passando uma leitora óptica sobre códigos de barras que constituem as legendas de um livro de imagens. O aparelho então enuncia as palavras que podem ser associadas a sua forma impressa, auxiliando assim a alfabetização.

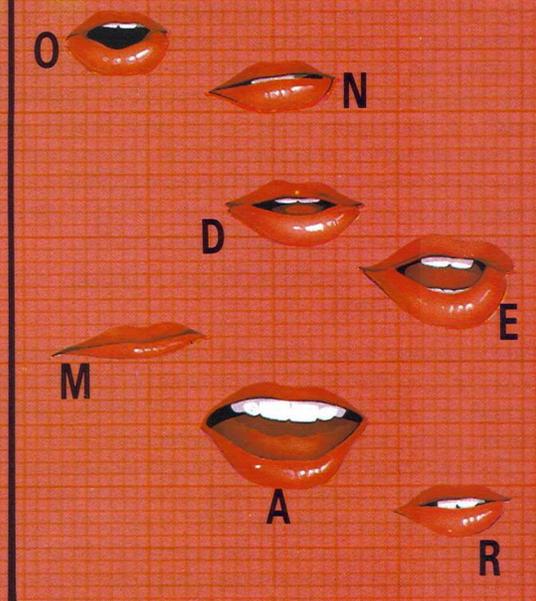


Figuras da fala

Tomemos como exemplo algumas palavras como: carro, quarto, girassol. Estas palavras podem ser divididas em vários fonemas, como:

K/A/R/U — K/W/A/R/T/U — Ž/I/R/A/S/O/W

Diferentes chips requerem que estas palavras sejam divididas e especificadas de várias maneiras. Para a mesma palavra, um outro chip pode exigir fonemas diferentes. O chip Votrax, por exemplo, contém cerca de sessenta fonemas e regras para seu uso. Os fonemas podem ser acessados facilmente por um simples número. A fim de tornar o sistema mais prático, um conjunto de programas é fornecido, permitindo que o usuário digite a pronúncia desejada na forma de fonemas, como no exemplo acima. Entretanto, com aparelhos como o Braid Speech Synthesiser ou Personal Speech System, da Votrax, que incluem microprocessador e software adequados, o usuário pode digitar um texto simples e receber o equivalente falado.



Peek e poke

Esses dois comandos são empregados toda vez que se quer programar algo não previsto na linguagem BASIC. Eles vão ser utilizados de modo diferente pelos diversos equipamentos.

Usando o POKE

A instrução POKE deve ser usada com cuidado, pois modifica os conteúdos das posições de memória, o que pode afetar o funcionamento do computador. Disto não advirá nenhum dano ao equipamento, mas o programa poderá perder-se. Eis algumas instruções POKE que são "seguras", para você experimentar.

No Atari 400 ou 800, o armazenamento do número 1 pela instrução POKE na posição 751 desligará o cursor na tela; experimentalmente a posição 751,1.

Nos computadores da linha Apple, experimente a posição 1024,1; o número 1024 é o endereço da primeira posição na tela. Nos computadores compatíveis com o Sinclair (TK83, TK85, CP 200 etc.), as instruções PEEK e POKE são utilizadas como se segue:

```
POKE 17300,57
PRINT PEEK 17300
```

Com estas instruções, o resultado será o número 57 que foi atribuído ao endereço 17300 pela instrução POKE.

PEEK e POKE são duas "instruções" da linguagem BASIC usadas em programação de nível mais adiantado, quando bits e bytes, isoladamente, devem ser manipulados na memória. A instrução PEEK é utilizada para examinar os conteúdos de um endereço específico na memória; a instrução POKE é usada para armazenar números (que variam de 0 a 255) em determinadas posições da memória.

As instruções PEEK e POKE permitem ao programador que utiliza a linguagem BASIC o acesso às operações internas do computador, o que de outro modo seria impossível. Geralmente, a linguagem BASIC incorporada em seu computador se encarrega das posições efetivas em que se encontram armazenadas as variáveis e os dados que definem os caracteres a serem apresentados na tela. Embora não precisemos nos preocupar com os locais da memória em que estão esses elementos, ocasionalmente teremos de localizá-los. É a instrução PEEK que nos permite realizar isso.

Escreve-se desse modo um pequeno programa para verificar qualquer posição na memória:

```
10 REM EXAMINAR POSICOES DE MEMORIA
20 PRINT "FORNECER A POSICAO DA MEMORIA EM
NUMEROS DECIMAIS"
30 INPUT M
40 P = PEEK(M)
50 PRINT "OS CONTEUDOS DA POSICAO "; M;"
SAO ";P
60 GOTO 20
70 END
```

Este programa imprimirá o conteúdo do endereço especificado, através de um número decimal (evidentemente, o computador o armazena em código binário). Se você quiser verificar os equivalentes dos conteúdos em termos de caracteres "imprimíveis", a linguagem BASIC inclui uma função que permite converter números decimais em seus caracteres correspondentes. Essa função é a CHR\$ e a alteração da linha 50 imprimirá, em seu lugar, caracteres equivalentes às posições da memória:

```
50 PRINT "OS CONTEUDOS DA POSICAO ";M;"
SAO "; CHR$(P)
```

Para examinar a memória por completo, acrescenta-se um loop FOR-NEXT, suprimindo a linha 30 e alterando a linha 20 para FOR X = 0 TO 65535; substitui-se, ainda, a linha 60 pela instrução NEXT X.

Para dar tempo suficiente ao exame de cada caractere enquanto é impresso, você acrescenta um loop de retardo após a instrução PRINT e antes da instrução NEXT X. Note que o limite superior do loop FOR-NEXT supõe que você tenha um equipamento com 64 Kbytes de memória. Este número pode ser alterado

no caso de memórias de menor capacidade: para memórias de 16 Kbytes é necessário o número 16383, em notação decimal; para memórias de 32 Kbytes, 32767; e para as de 48 Kbytes, 49151. A listagem completa deste programa é a seguinte:

```
10 REM VERIFICACAO E IMPRESSAO DE TODAS AS
POSICOES DE MEMORIA
20 FOR X = 0 TO 65535
30 LET Y = PEEK(X)
40 PRINT "POSICAO ";X;"=";Y;"=";
50 PRINT CHR$(Y)
60 FOR D = 1 TO 200
70 NEXT D
80 NEXT X
90 END
```

Embora a função CHR\$ converta números decimais em seus caracteres equivalentes, os caracteres que podem ser impressos representam-se pelos números 32 a 127. A maioria dos computadores utiliza números entre 128 e 255 (o maior número que pode ser representado através de um único byte) para caracteres gráficos especiais. Muitos dos números entre 0 e 31 têm funções especiais de controle de tela. Quando o programa, ao ser processado, encontra esses números na memória, a função CHR\$ converte esses números em interessantes efeitos de tela.

A instrução POKE é, basicamente, o procedimento oposto à instrução PEEK: permite que você "escreva" 1 byte de dados (qualquer número entre 0 e 255) em qualquer posição de memória. Esta instrução deve ser utilizada com cuidado: se, pela instrução POKE, você fizer o armazenamento de um número na posição errada da memória, poderá danificar uma parte de um programa essencial. O único modo de corrigir isso será desligar e ligar o computador novamente, a menos que tenha um botão RESET. De qualquer modo, você vai perder o que está na memória naquele momento. Assim, antes de utilizar a instrução POKE, verifique em seu manual qual é a área, no mapa da memória, designada como "área para o usuário".

A maioria dos microcomputadores possui memória de vídeo (a memória utilizada para armazenamento dos caracteres que serão apresentados na tela) acessível ao usuário. Normalmente, o computador obtém o formato desses caracteres da memória ROM especial, chamada gerador de caracteres, que armazena os padrões de pontos para cada caractere. Mas é igualmente possível utilizar a memória RAM. Quando os códigos de padrão para os caracteres estão armazenados na memória RAM, novos padrões podem ser armazenados pelo comando POKE, na posição correspondente da memória RAM e utilizados para definição de novos caracteres.



Commodore 64

O mais recente lançamento da Commodore possibilita a elaboração de gráficos animados e utiliza seu televisor ou alto-falantes para produzir excelente som.

O Commodore 64 é realmente um dos primeiros de uma nova geração de microcomputadores adequados tanto para jogos como para auxiliar a administração de uma pequena empresa.

Um padrão de 64 Kbytes de memória é suficiente para lidar com representações gráficas bem elaboradas, ou programas contábeis, tais como folhas eletrônicas, processadores de palavras e bancos de dados. Uma certa compatibilidade que possui com os sistemas contábeis do VIC-20 e do PET, da Commodore, aumenta ainda mais a variedade de software disponível.

As numerosas interfaces do Commodore 64 permitem que ele opere com muitos tipos de equipamentos periféricos, inclusive a unidade de discos VIC e impressoras.

Duas notáveis características do Commodore 64 são os gráficos sprites (ver p. 45) e todos os recursos de síntese musical. Os sprites são figuras criadas na tela por meio de gráficos de alta resolução, que podem ser movimentados mediante comandos simples, o que é ideal para efeitos de entretenimento, como invasores espaciais, aeronaves, explosões etc. Tais efeitos são possíveis sem necessidade de recur-

so a gráficos sprites, mas isto exige um esforço maior de programação. No Commodore 64, pode-se fazer com que os sprites se movam, aumentem e diminuam de tamanho, ou mudem de cor e passem na frente ou atrás de outros sprites ou figuras gráficas imóveis, tais como cenários de fundo — proporcionando desse modo profundidade à imagem.

A síntese sonora é igualmente bem elaborada, ao contrário dos rudimentares “beeps” e “squawks” de máquinas mais baratas. Além de possuir três vozes distintas (possibilitando acordes e harmonias, em vez de apenas notas simples), o Commodore 64 permite controle completo dos vários parâmetros que dirigem o som ou timbre da nota em questão. Em outras palavras, o Commodore 64 pode simular uma ampla variedade de instrumentos musicais e ruídos diversos.

O ponto fraco do equipamento é sua linguagem BASIC — praticamente a mesma versão que a Commodore usava nos seus primeiros computadores. Operações mais complexas exigem o comando POKE, de difícil utilização, em lugar de uma boa variedade de comandos fáceis que aproveitem as características excelentes do 64



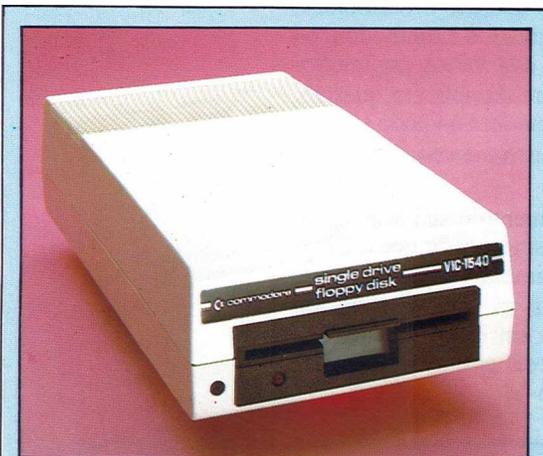
O Commodore 64 tem excelentes teclas “esculpidas” de modo a facilitar a digitação. Além dos caracteres comuns, há uma grande série de caracteres gráficos em bloco. Uma tecla de várias finalidades, marcada com o logotipo da empresa Commodore, efetua as mudanças entre os vários conjuntos de caracteres. As cores são alteradas por meio da tecla CONTROL(CTRL) e de uma das teclas da fila superior.

Pode-se atribuir às quatro teclas de função, à direita do teclado, funções determinadas dentro de um programa.



Cassete 1530 C2N

Esta é a unidade cassete standard da própria Commodore, para ser ligada diretamente à entrada para gravador cassete do 64. Recebe energia elétrica do computador, que pode também ligar e desligar o motor do gravador.



Unidade de disco 1540

Esta unidade simples de disco conecta-se ao 64 por meio de uma entrada serial. Um disco de 5 1/4" possibilita 170 Kbytes de armazenamento, e o tempo máximo de acesso aos dados armazenados é de 2 segundos.



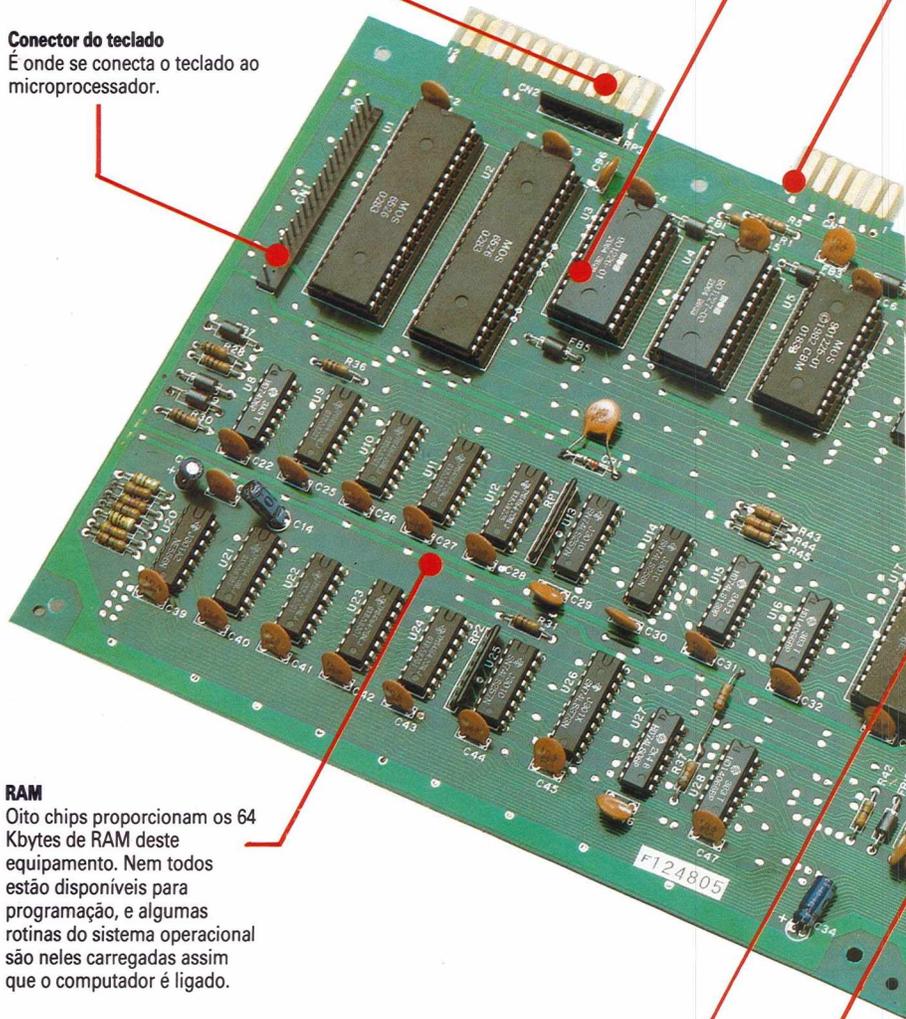
Impressora 1525

Esta impressora matricial pode imprimir todos os caracteres que o 64 apresenta na tela, inclusive caracteres gráficos. Conecta-se por meio da entrada serial, e imprime trinta caracteres por segundo.

Utilizando uma tomada para 24 pinos, é possível ligar uma série de dispositivos ao computador. Vários pinos podem ser colocados como linhas de entrada ou saída.

Conector do teclado

É onde se conecta o teclado ao microprocessador.



RAM

Oito chips proporcionam os 64 Kbytes de RAM deste equipamento. Nem todos estão disponíveis para programação, e algumas rotinas do sistema operacional são neles carregadas assim que o computador é ligado.

Microprocessador

O 6510 é o principal processador. Controla todas as operações do computador, bem como a manipulação de uma boa parte da entrada/saída de dados.

Chip de som

O dispositivo de interface de som 6581 gera todos os sons que o Commodore 64 pode produzir.



COMMODORE 64

MICROPROCESSADOR

6510

CLOCK

1 MHz

MEMÓRIA

Total de 64 Kbytes. 20 Kbytes de ROM utilizados para o sistema operacional e linguagem BASIC. 54 Kbytes de RAM estarão disponíveis para o usuário se o interpretador BASIC não for usado.

VÍDEO

25 linhas com 40 posições. Alta resolução com 320 x 200 pontos, 16 cores.

TECLADO

Tipo máquina de escrever, 62 teclas e quatro de função.

LINGUAGENS

BASIC, FORTH, COMAL, PILOT, LOGO, UCSD, PASCAL

PERIFÉRICOS

Gravador cassete, TV, monitor, unidade de disco (5¼"), cartuchos, interface RS232, caneta óptica, paddle, joystick.

DOCUMENTAÇÃO

O Guia do Usuário descreve o equipamento, seu funcionamento e linguagem, mas não a um nível que seja completamente apropriado a principiantes. Não proporciona uma orientação abrangente dos endereços dos registros especiais. Isto é particularmente lamentável, porque o som e os gráficos do 64 são programados por meio destes registros especiais.

O ideal seria ignorar o Guia do Usuário e adquirir o Guia de Referência do Programador do Commodore 64, publicado pela Commodore Business Machines Inc. e Howard W. Sams, de Londres.

ROMs de BASIC

Estes três chips contêm a linguagem BASIC do computador.

Tomada para fornecimento de energia

Entrada para gravador cassete

É uma entrada especial para conectar um gravador standard Commodore ao computador. Não aceita outras marcas de gravadores, sem modificações especiais.

Entrada serial

Esta é uma tomada para comunicação serial com dispositivos adicionais. Em comunicação serial, enviam-se dígitos binários um após o outro.

Conector para áudio/vídeo

O computador pode ser ligado a um equipamento de som ou a um monitor por meio desta tomada.

Abertura para cartucho

Cartuchos que contêm linguagens ou programas para jogos são encaixados aqui.

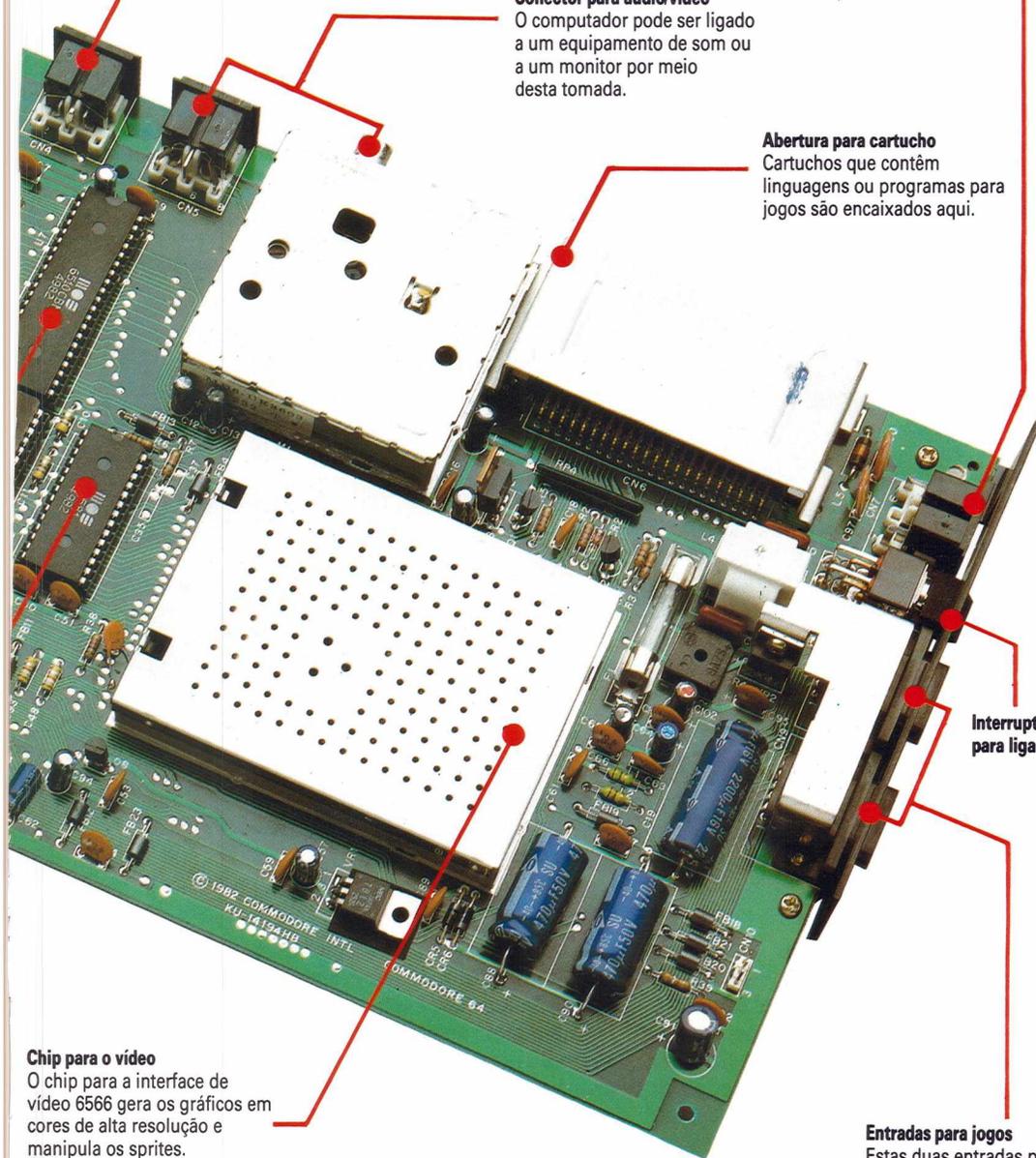
Interruptor para ligar/desligar

Entradas para jogos

Estas duas entradas permitem que se liguem ao computador canetas ópticas, joysticks e paddles.

Chip para o vídeo

O chip para a interface de vídeo 6566 gera os gráficos em cores de alta resolução e manipula os sprites. É recoberto por uma proteção contra o calor.



Piratas à vista

A pirataria na indústria do software é um sério problema. E muitas são as tentativas dos fornecedores para solucioná-lo.



Gravador "escravo"

Cassetes de software, assim como cassetes de música, são duplicados usando-se uma copiadora de alta velocidade. Este aparelho consiste em um deck principal, onde é colocado o original, e algumas unidades "escravas", que gravam simultaneamente. Copiar os dois lados de um programa em cassette é uma questão de segundos, ao passo que discos têm de ser copiados individualmente, usando-se unidades de disco normais.

Cem por um

Assim como é ilegal reproduzir gravações musicais em cassette, a reprodução de programas é chamada de pirataria de software. Infelizmente, ela é difícil de ser evitada, detectada ou legalmente processada. Alguns fornecedores afirmam que para cada exemplar de programa comprado legalmente são feitas cerca de cem cópias ilegais.

Pirataria em software é, por definição, a atividade de reproduzir programas sem a devida autorização. Assim como acontece com as gravações musicais, ela se processa em diversos níveis e modos. Em um nível mais elementar, comete-se pirataria toda vez que o usuário de um microcomputador faz a cópia de um programa emprestado por um amigo. O fato de alguns programas (principalmente os escritos em linguagem de máquina) não poderem ser reproduzidos através de comandos normais de BASIC não constitui empecilho, uma vez que basta conectar dois gravadores cassette e copiar o programa de um para o outro, sem haver necessidade do uso de computador.

Fornecedores de jogos eletrônicos afirmam que para cada título de programa vendido são feitas aproximadamente cem cópias ilegais. Embora alguns possam suportar esse prejuízo, a verdade é que há muita gente que vive de royalties de programas e nem por isso ganha uma fortuna.

Existem controvérsias em torno de comerciantes que alugam ou emprestam software por um período de experiência, pois se acredita que estes esquemas facilitam as coisas para os que copiam programas. Comerciantes menos escrupulosos vão além, distribuindo cópias piratas de programas conhecidos aos compradores de microcomputadores, a fim de aumentar o valor real destes aparelhos.

Há ainda os casos em que distribuidores reproduzem programas em larga escala, vendendo-os a outros comerciantes. Estes produtos equivalem a reproduções clandestinas de gravações de rocks famosos, o que não chega a ser muito arriscado quando os fornecedores originais dos pacotes são de outro país.

A pirataria passa a ser mais sofisticada e mais difícil de se detectar quando, por exemplo, alguém pega um programa já existente, faz nele algumas modificações e comercializa-o como sendo seu próprio. A nova versão às vezes apresenta um desempenho melhor ou algumas funções adicionais, mas, por outro lado, pode apenas ter sofrido modificações nos caracteres de comandos do programa original e no layout da informação que aparece na tela, a fim de que o pacote não seja reconhecido de imediato. Esta prática é mais usada com programas de aplicação comercial do que com jogos eletrônicos.

Se este processo constitui ou não um caso de pirataria de software, da mesma forma que uma simples cópia, é um ponto bastante discutível, e por essa razão tanta gente escapa ilesa. A lei pouco defende os editores de software e as disposições sobre direitos autorais não protegem os programas contra pirataria ou plágios.

Aparentemente, os direitos autorais só dizem respeito a material impresso (com exceção da música); portanto, os programas de computação armazenados em RAM, disco, ou cassete não são resguardados. Como acontece com a maioria das questões de ordem legal, é necessário estabelecer precedentes, o que custa tempo e dinheiro.

Quando uma empresa copia a idéia de um programa conhecido e lança sua própria versão, surge uma questão bastante controversa. Neste caso, eles não copiam os códigos do programa, mas tomam nota detalhadamente de como ele aparece no vídeo e reage aos dados fornecidos pelo usuário; então, a partir dessas anotações, escrevem um programa com o mesmo efeito. Um exemplo característico é o do PacMan, um jogo eletrônico que foi lançado em máquinas operadas por fichas, posteriormente distribuído pela Atari para ser usado em microcomputadores e em aparelhos de videogames, e que acabou sendo lançado, com pequenas alterações, por uma série de editores de software.

Todos estes jogos, com pequenas variações, apresentavam a conhecida criaturinha devorando pontos em seu caminho através de um labirinto. Após alguns meses, a Atari conseguiu eliminar a maior parte desses concorrentes por meio de ações judiciais e, em casos de operações menores, pela simples ameaça de processá-los.

Geralmente, autores e fornecedores de software recorrem a certos meios extrajudiciais a fim de proteger seus interesses. Alguns argumentam que vendendo seus programas a um preço bastante baixo, desestimularão aqueles que fazem cópias clandestinas. No caso de programas mais sofisticados, um manual bem produzido e uma embalagem atraente já proporcionam certo grau de proteção.

Um outro meio de proteger um software caro é o "registro do usuário": só recebe ajuda e apoio por telefone o usuário que devolver o cartão que acompanha o manual do proprietário.

Os assim chamados meios de proteção "hard"

consistem quase sempre em um dispositivo de tamanho de uma caixa de fósforos chamado "dongle", que deve ser ligado a uma das interfaces do computador, a fim de permitir que o programa funcione. Os circuitos do dongle têm um código eletrônico, geralmente uma combinação dos dígitos 1 e 0, gravados em ROM. A intervalos freqüentes, o programa solicita o dongle, e, se não receber de volta o código correto, recusa-se a continuar. O código pode ser diferente para cada dongle, significando que cada cópia do pacote deverá ser combinada com o dongle que a acompanha.

Neste caso, só se consegue fazer cópias ilegais falsificando o dongle ou reescrevendo o código do programa para remover as seções que o solicitam. Sem dúvida, este procedimento está além da capacidade da maioria dos programadores amadores, o que se não evita a pirataria, ao menos a dificulta.

Intensas pesquisas vêm sendo desenvolvidas para obtenção de métodos que ofereçam essa mesma proteção, sem o auxílio de hardware adicional. Um recurso, semelhante à linha-d'água do papel, consiste em um código magnético sobreposto no cassete ou disco, "atrás" da gravação do programa propriamente dito, o qual não se transfere para a cópia, de modo que o programa só funciona no cassete ou disco originais.

A única proteção "hard" economicamente viável para os fornecedores de jogos eletrônicos é o cartucho ROM, que em geral alcança preços mais elevados, pois não necessita de longo tempo para carregamento, como no caso dos cassetes. Contudo, nem mesmo os cartuchos são inexpugnáveis: atualmente existem dispositivos que copiam um cartucho para um cassete ou para um outro cartucho, que pode ser programado ou reprogramado pelo usuário.

A pirataria em software é uma guerra de "bandidos e mocinhos", em que protagonistas tentam superar uns aos outros em engenhosidade. Nesta brincadeira, ninguém é eliminado — na melhor das hipóteses, pode-se dificultá-la o bastante para que se torne apenas uma atividade marginal.

"Dongles"

São pequenos dispositivos usados para proteger certos programas contra reproduções ilícitas. Os programas não funcionam enquanto o dongle certo não for ligado a uma interface do computador. A parte eletrônica é geralmente envolta em resina sólida, dificultando qualquer acesso aos circuitos.



Segunda dimensão

Matrizes unidimensionais armazenam conjuntos de dados que apresentam características comuns. Matrizes bidimensionais, por sua vez, são utilizadas para tabelas de dados e diagramas.

Até agora, examinamos duas espécies de variáveis: as simples e as indexadas. As variáveis simples são semelhantes às posições de memória em que os números (ou séries de caracteres) são armazenados e manipulados pela referência ao "nome" da variável. Também podem armazenar apenas um valor ou série e ter nomes "simples" de variáveis — N, B2, X, Y3 são exemplos. As variáveis indexadas, às vezes chamadas matrizes unidimensionais, podem armazenar uma lista completa de valores ou séries. A quantidade de valores ou séries que o programa comporta é especificada no início do programa, com a instrução DIM. Por exemplo, a instrução DIM A(16) determina que a matriz denominada A pode conter dezesseis valores separados. Deve-se notar, entretanto, que muitas versões do BASIC admitem a variável A(0) como o primeiro elemento, de modo que a instrução DIM A(16) efetivamente define dezessete elementos. A referência a essas posições é feita usando-se o indexador apropriado. A instrução PRINT A(1) imprimirá o primeiro elemento da matriz; a instrução LET B = A(12) atribuirá o valor representado pelo décimo segundo elemento da matriz à variável B; a instrução LET A(3) = A(5) atribuirá o valor do quinto elemento ao terceiro elemento.

Todavia, às vezes precisamos manipular dados que se apresentariam melhor sob a forma de tabelas de dados. Observe como este procedimento é muito semelhante ao das folhas eletrônicas (ver p. 158). Tais dados podem variar desde tabelas para resultados de futebol a desdobramentos de vendas por itens e por seções em uma loja. Examine o desdobramento de despesas domésticas durante um ano, como exemplo típico de uma tabela de dados:

	ALUGUEL	TELEFONE	LUZ	ALIMENT.	CARRO
JAN	200000	25000	30000	160000	80000
FEV	200000	25000	30000	160000	80000
MAR	200000	25000	35000	200000	80000
ABR	200000	35000	35000	200000	130000
MAI	200000	35000	35000	230000	130000
JUN	200000	35000	45000	230000	130000
JUL	200000	40000	45000	260000	130000
AGO	450000	40000	45000	260000	180000
SET	450000	40000	60000	300000	180000
OUT	450000	50000	60000	300000	180000
NOV	450000	50000	60000	350000	220000
DEZ	450000	50000	70000	350000	220000

Tal organização de dados permite que estes sejam manipulados de vários modos. É fácil, por exemplo, saber as despesas de março pela simples adição de todos os números da linha desse mês. É fácil, também, obter as despesas de telefone e carro, somando-se as respectivas colunas. As médias anuais e mensais são igualmente fáceis de ser calculadas.

Esta tabela de dados é chamada matriz bidimensional. Tem doze linhas e cinco colunas.

Matrizes bidimensionais como esta podem também ser representadas, em BASIC, de modo muito semelhante ao das matrizes unidimensionais; a diferença é que a variável, neste caso, necessita de dois indexadores para remeter a qualquer posição.

Se escrevermos um programa em BASIC utilizando essa tabela de dados, o procedimento mais simples será lidar com a tabela como se fosse uma matriz bidimensional única. Assim, como no caso de matrizes indexadas comuns, nós lhe atribuímos um nome de variável. Vamos chamá-la de A. Novamente, como acontece com as variáveis indexadas comuns, ela precisará ser DIMensionada. Como são doze linhas e cinco colunas, a variável deve ser dimensionada assim: DIM A(12,5). A ordem de colocação dos dois indexadores é importante: está convencionalizado que as linhas são especificadas em primeiro lugar e as colunas em segundo. Nossa tabela tem doze linhas (uma para cada mês) e cinco colunas (para as cinco categorias de despesas) e, portanto, é uma matriz 12-por-5.

A instrução DIM atende a duas funções básicas: reserva posições de memória suficientes para a matriz na memória do computador e permite que cada posição seja indicada pelo nome da variável, seguido pelas posições de linha e de coluna entre parênteses. A instrução DIM, por exemplo DIM X(3,5), cria uma variável X, que representa uma matriz com três linhas e cinco colunas.

Observe o quadro, admitindo que os dados tenham sido fornecidos como elementos de uma matriz bidimensional, denominada A. Calcule os valores de A(1,1), A(1,5), A(2,1), A(3,3) e A(12,3).

Pode-se fornecer uma tabela de dados sob a forma de matriz em uma parte do programa, usando-se, por exemplo, instruções LET.

```
30 LET A(1,2) = 25000
40 LET A(1,3) = 30000
50 LET A(1,4) = 160000
.
.
.
610 LET A(12,5) = 220000
```

Evidentemente, porém, este procedimento é trabalhoso. Muito mais simples são as instruções READ e DATA ou a instrução INPUT com loops FOR-NEXT incluídos. Vejamos como isto é feito com a instrução READ:

```
10 DIM A(12,5)
20 FOR R = 1 TO 12
30 FOR C = 1 TO 5
```



```

40 READ A(R,C)
50 NEXT C
60 NEXT R
70 DATA 200000, 25000, 30000, 160000, 80000,
200000, 25000, 30000
80 DATA 160000, 80000, 200000, 25000, 35000,
200000, 80000, 200000
90 DATA 35000, 35000, 200000, 130000, 200000,
35000, 35000, 230000
100 DATA 130000, 200000, 35000, 45000, 230000,
130000, 200000, 40000
110 DATA 45000, 260000, 130000, 450000, 40000,
45000, 260000, 180000, 450000
120 DATA 40000, 60000, 300000, 180000, 450000,
50000, 60000
130 DATA 300000, 180000, 450000, 50000, 60000,
350000, 220000
140 DATA 450000, 50000, 70000, 350000, 220000
150 END
    
```

Há detalhes importantes a notar neste programa. O primeiro deles é que a instrução DIM inicia o programa. A instrução DIM deve ser executada só uma vez no programa e costuma-se colocá-la logo no início ou antes da execução dos loops. O segundo ponto a observar é que há dois loops FOR-NEXT, um para determinar a “linha” do indexador e outro para determinar a “coluna”. Esses dois loops não se seguem um após o outro, mas, sim, um está “alojado” no outro. Observe os limites escolhidos. A instrução FOR R = 1 TO 12 aumentará o valor da linha de 1 até 12; a instrução FOR C = 1 TO 5 aumentará o valor de 1 até 5.

Exatamente na metade do loop incluído encontra-se a instrução READ. A parte fundamental do programa é:

```

20 FOR R = 1 TO 12
30 FOR C = 1 TO 5
40 READ A(R,C)
50 NEXT C
60 NEXT R
    
```

Na primeira passagem após as instruções das linhas 20 e 30 terem sido executadas, os valores das variáveis R e C serão, ambos, 1. Assim, a linha 40 equivalerá a READ A(1,1). O primeiro item de dados na instrução DATA é 200000; portanto, este valor será atribuído à primeira linha e coluna da tabela. A escolha de oito elementos para cada instrução DATA faz-se de maneira arbitrária.

Executadas essas operações, a instrução NEXT C remete o programa outra vez à linha 30 e o valor da variável C aumenta para 2. A linha 40 passa então a equivaler a READ A(1,2) e o item seguinte de dados, 25000, será atribuído à primeira linha e segunda coluna da matriz. Este processo se repete até que a variável C tenha alcançado o valor 5. Após isso, a instrução NEXT R, na linha 60, faz o programa retornar à linha 20 e o valor de R aumenta para 2. A linha 30 atribuirá novamente o valor 1 para a variável C e, desse modo, a nova linha 40 equivalerá a READ A(2,1).

A inclusão de loops por este procedimento é muito conveniente, mas preste atenção! Cada loop deve ser alojado de modo completo no interior de outro loop e a ordem das instruções NEXT deve ser cuida-

dosamente observada. Repare que o primeiro loop, FOR R, tem a segunda instrução NEXT. Quando há dois loops, um alojado no outro, o primeiro é chamado loop interior, e o segundo, loop exterior. O loop interior deverá estar inteiramente concluído antes de o loop exterior ser desenvolvido. É possível efetuar o alojamento de loops em tantos “níveis” quantos o programa exigir, mas tais programas se tornam complexos, difíceis de acompanhar e corrigir. É um mau hábito de programação incluir instruções de desvio no interior de loops; igualmente, o uso de comandos GOTO deve ser evitado.

Examinemos as instruções DATA. Observe que as vírgulas são usadas para separar itens de dados, mas não deve haver vírgulas antes do primeiro item de dados ou após o último. Inserimos espaços entre cada item de dados; este procedimento, entretanto, não é comum. Com frequência cometem-se erros ao fornecer os dados, e é difícil detectá-los posteriormente. Pode-se recorrer a tantas instruções DATA quantas forem necessárias. Cada nova linha deve começar pela instrução DATA. Os dados são lidos, um de cada vez, a partir do início da primeira instrução DATA, prosseguindo até que todos os itens tenham sido lidos. Assegure-se de que esteja correta a numeração de itens de dados, senão você receberá uma mensagem de erro, quando o programa for processado.

O programa apresentado, até aqui, nada realiza, exceto a conversão dos dados correspondentes para a matriz bidimensional. Após a entrada e o processamento do programa, nada aparentemente acontece. A fim de testar se os dados estão distribuídos de modo correto, experimente usar alguns comandos PRINT (comando, em BASIC, é uma palavra-chave que pode ser imediatamente executada, sem estar incluída em um programa e, portanto, não necessita de um número de linha. São exemplos de comandos: LIST, RUN, SAVE, AUTO, EDIT e PRINT). O comando PRINT A(1,1) <CR> deverá fazer com que o número 200000 apareça na tela. O que será impresso pelos seguintes comandos?

```

PRINT A(12,1)
PRINT A(1,5)
PRINT A(5,1)
PRINT A(5,5)
    
```

Para que o programa realize uma operação que tenha utilidade efetiva, será necessário ampliá-lo. Do modo como se apresenta, ele forma uma base adequada para um “programa principal”. A fim de utilizá-lo como parte de um programa mais amplo e de maior utilidade, pode-se desenvolver módulos sob a forma de sub-rotinas a serem chamadas por instruções GOSUB inseridas em pontos convenientes, antes da instrução END.

Nos primeiros estágios de elaboração de um programa de contabilidade doméstica, é melhor iniciar com uma simples descrição, escrita, das várias exigências. Possivelmente, desejamos obter totais e médias das despesas mensais ou de uma categoria (luz, por exemplo). Calcularemos os detalhes para obtenção desses resultados em um estágio posterior. Se, no interior do programa, houver uma escolha das sub-rotinas que devem ser executadas, receberemos a indicação, talvez por um “menu”, que diri-

girá o controle para as sub-rotinas correspondentes em resposta aos dados que fornecemos. Um esboço inicial do programa neste estágio seria semelhante ao seguinte:

```
PROGRAMA PRINCIPAL
(ENTRADA DE DADOS)

MENU
(SUB-ROTINAS PARA ESCOLHA)

END
```

Um pequeno trabalho de aprimoramento deste programa nos mostrará que necessitaremos de sub-rotinas para calcular os totais dos meses e das categorias (TOTALMES e TOTALCAT), calcular a média de despesas mensais (MEDMENSAL) e a média de despesas anuais por categoria (MEDCAT). O motivo da utilização, nessas sub-rotinas, de nomes constituídos por uma só palavra está em que isto nos ajuda a planejar o programa sem preocupar-nos, nesta etapa, com detalhes como números de linha. Pensando melhor, chegaremos à conclusão de que mesmo a escolha do principal "menu" do programa deve ser executada como uma sub-rotina, a fim de manter a parte principal do programa como um módulo separado. O próximo estágio de elaboração será semelhante ao seguinte:

```
PROGRAMA PRINCIPAL (ENTRADA DE DADOS)
  MENU (CHAMAR SUB-ROTINA)
END
**SUB-ROTINAS**

1 MENU
2 TOTAIS
3 MEDIAS

(2) TOTAIS
4 TOTALMES
5 TOTALCAT

(3) MEDIAS
6 MEDMES
7 MEDCAT
```

Este esboço do programa mostra como a sub-rotina MENU nos permitirá a escolha entre TOTAIS e MEDIAS. Estas mesmas serão, ambas, sub-rotinas. A sub-rotina TOTAIS permitirá uma escolha ulterior entre TOTALMES e TOTALCAT. Estas operações se constituirão de sub-rotinas que realizarão os cálculos efetivos.

A sub-rotina MEDIAS permitirá a escolha das operações MEDMES ou MEDCAT, que serão também sub-rotinas para realização dos cálculos correspondentes. Nesta etapa, é possível perceber se nosso "programa" realizará as operações que desejamos, sem precisar recorrer a alguma codificação efetiva (detalhado programa escrito em BASIC). Se ficarmos satisfeitos com um "até aqui, tudo bem", estaremos em condição de enfrentar o desenvolvimento dos próprios módulos (sub-rotinas). A única alteração exigida pelo programa principal será uma chamada de sub-rotina, antes da instrução END; assim, podemos acrescentar:

```
145 GOSUB **MENU**
```

Observe que ainda estamos utilizando "nomes" para as sub-rotinas, em vez de números de linha.

Outras linguagens admitem que os subprogramas sejam chamados pelo nome, mas a maioria das versões da linguagem BASIC não; assim, é necessário substituí-los pelos números das linhas.

Examinemos o modo pelo qual a sub-rotina MENU poderia ser escrita.

```
REM A SUB-ROTINA **MENU**
PRINT "VOCE DESEJA T(TOTAIS) OU M(MEDIAS)?"
PRINT "DIGITE T OU M"
INPUT L$
IF L$ = "T" THEN GOSUB *TOTAIS*
IF L$ = "M" THEN GOSUB *MEDIAS*
RETURN
```

Observações: estamos assinalando as sub-rotinas chamadas, encerrando-as entre os sinais *—*, os quais você deverá substituir por números de linha, quando for possível saber quais são eles.

Admitamos que você digite T, para TOTAIS. O programa, então, chamará a sub-rotina TOTAIS, que apresentará um outro menu, semelhante ao seguinte:

```
REM A SUB-ROTINA **TOTAIS**
PRINT "VOCE DESEJA OS TOTAIS DE"
PRINT "M(ES) OU C(ATEGORIA)?"
PRINT "DIGITE M OU C"
INPUT L$
IF L$ = "M" THEN GOSUB *TOTALMES*
IF L$ = "C" THEN GOSUB *TOTALCAT*
RETURN
```

Admitamos que você escolha M para representar o TOTALMES. Vejamos como escrever um módulo para calcular a despesa total de qualquer mês no ano.

```
REM A SUB-ROTINA **TOTALMES**
REM ESTA CALCULA A DESPESA TOTAL EM
REM QUALQUER MES
PRINT "ESCOLHER MES"
PRINT "1-JAN 2-FEV 3-MAR 4-ABR 5-MAI"
PRINT "6-JUN 7-JUL 8-AGO 9-SET"
PRINT "10-OUT 11-NOV 12-DEZ"
PRINT "DIGITE O NUMERO CORRESPONDENTE AO
MES"
LET T = 0
INPUT M
FOR C = 1 TO 5
LET T = T + A(M,C)
NEXT C
PRINT "A DESPESA TOTAL NO MES "
PRINT "NUMERO ";M;" E ";T
RETURN
```

O número que representa o mês é digitado e a instrução INPUT o atribui à variável M(MES). A variável M é utilizada para especificar o indexador de "linha" da matriz bidimensional A. O loop FOR-NEXT aumenta o valor da variável C (que representa a coluna) de 1 a 5; assim, na primeira passagem pelo loop, se escolhermos o número 3 — que corresponde a março, a instrução LET equivalerá a LET T = T + A(3,1). Na passagem seguinte pelo loop, ela equivalerá a LET T = T + A(3,2) e assim sucessivamente.

Desta vez, deixaremos a você a tarefa de desenvolver as outras sub-rotinas ou experimentar outros exercícios. As matrizes bidimensionais são convenientes para programas que incluem tabelas de dados estatísticos, financeiros ou de outra natureza.



Respostas aos Exercícios da p. 175

Função RND

```
40 IF R > 6 THEN LET R = 1
```

Loop e média

```
5 FOR L = 1 TO 100
:
:
80 LET T = T + R
90 NEXT L
100 LET A = T/100
110.END
```

Substituir com sub-rotina

Elimine as linhas 5, 80, 90, 100 e 110 da resposta acima. Altere as linhas de 10 a 70 para (digamos) 1000 a 1070. Certifique-se de que a linha 40 está igual à da resposta ao exercício de Função RND, acima. A seguir, acrescente a instrução 1080 RETURN. Incorpore o resultado ao programa principal. Altere as linhas 50 e 130 do programa principal, de modo que se apresente da seguinte forma: 50 GOSUB 1000 e 130 GOSUB 1000.

INKEY\$

```
10 PRINT "DIGITE QUALQUER TECLA"
20 LET A$ = INKEY$
30 IF A$ = " " THEN GOTO 20
40 PRINT "A TECLA PRESSIONADA FOI ";A$
50 END
```

Loop de controle de tempo

```
5 PRINT "PRESSIONAR A BARRA DE ESPACO APOS 10 SEGUNDOS"
```

```
10 FOR L = 0 TO 1
20 LET R = R + 1
30 IF INKEY$ = " " THEN GOTO 60
40 LET L = 0
50 NEXT L
60 PRINT "O VALOR DE R APOS 10 SEGUNDOS E ";R
70 END
```

IF-THEN

```
10 GOSUB 1000
20 PRINT "SUGIRA UM NUMERO"
30 FOR G = 1 TO 5
40 INPUT N
50 IF N > R THEN GOTO 110
60 IF N < R THEN GOTO 130
70 IF N = R THEN GOTO 150
80 NEXT G
90 PRINT "SEM DIREITO A OUTRA TENTATIVA. VOCE PERDEU"
100 GOTO 500
110 PRINT "SEU PALPITE ESTA MUITO ALTO"
120 GOTO 80
130 PRINT "SEU PALPITE ESTA MUITO BAIXO"
140 GOTO 80
150 PRINT "VOCE ACERTOU"
500 END
1000 REM **SUB-ROTINA DO NUMERO ALEATORIO**
(introduza sua sub-rotina aqui)
1020 RETURN
```

Exercícios

■ **Atribuição de valores.** Desenvolva um programa que atribua valores aos elementos ("gasolina", "manutenção" etc.) da matriz (ver ilustração abaixo). A seguir, escreva uma sub-rotina que solicite o nome de um mês e um título para despesas e imprima os conteúdos do quadro assim especificado. Finalmente, escreva uma sub-rotina que calcule a soma de cada coluna, coloque o resultado no quadro abaixo, realize o mesmo procedimento com relação às linhas, e calcule, em seguida, o total geral, a ser armazenado no quadro inferior da direita.

■ **Erros.** O programa apresentado a seguir não poderá ser processado convenientemente e produzirá

uma mensagem de erro. Localize o erro e efetue a correção necessária.

```
10 DIM A(3,4)
20 FOR R = 1 TO 3
30 FOR C = 1 TO 4
40 READ A(R,C)
50 NEXT C
60 NEXT R
70 FOR X = 1 TO 3
90 FOR Y = 1 TO 4
100 PRINT A(Y,X)
110 NEXT Y
120 NEXT X
130 DATA 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22
140 END
```

	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ	TOTAL
GASOLINA													
MANUTENÇÃO													
PEÇAS													
LAVAGEM													
SEGURO													
IMPOSTO													
OUTROS													
TOTAL													

Despesas com o carro

A ilustração apresenta oito quadriculados verticais e treze horizontais, com os diversos itens de despesas de um carro, durante um ano. Acompanhe os dados do exercício "Atribuição de valores", para fazer os cálculos.

Os traços perfeitos

O plotter é o melhor instrumento para se fazer um gráfico impresso de boa qualidade. Usando-se canetas de ponta porosa pode-se mudar a cor automaticamente.

A capacidade de criar gráficos impressos dos diagramas que aparecem na tela do computador é uma necessidade essencial em diversas atividades profissionais. Engenheiros, cientistas, artistas técnicos e homens de negócios precisam de diagramas, mapas, plantas, esquemas precisos e bem tracejados, que as impressoras convencionais não podem produzir. O único aparelho capaz de demarcar tais imagens com precisão é o plotter.

Até recentemente, esse instrumento era caro demais para o usuário de um microcomputador. Entretanto, com a introdução do dispositivo impressor com quatro canetas, conjugado ao mecanismo de plotter, esse tipo de gráfico está se tornando mais popular. Os vários modelos de plotters lançados no exterior e mesmo no Brasil oferecem características que anteriormente só eram encontradas em máquinas de elevado preço.

A necessidade de um plotter depende do tipo de saída gerada pelo computador. Projetistas precisam de desenhos exatos de equipamentos e instalações; um homem de negócios utiliza mapas estatísticos de vendas. Produzi-los em impressoras convencionais é trabalhoso, e o resultado se apresenta apenas em preto e branco.

Neste caso, a única opção de baixo custo é tirar uma foto colorida da tela. Isto talvez seja suficiente para quadros ou mapas comerciais, mas com certeza não terá exatidão satisfatória para um desenhista ou um arquiteto.

Os plotters funcionam de modo completamente diferente das impressoras: desenharam linhas entre dois pontos, em vez de criar sua saída por meio de caracteres pré-formados ou configurações de pontos. O princípio básico dos vários sistemas é o das coordenadas X, Y.

Assim como um gráfico é traçado definindo-se as coordenadas pelas quais a linha deve passar, a representação de uma forma qualquer pode ser dividida em várias coordenadas. Para uni-las e recriar a forma, deve haver algum tipo de movimento. Desse modo, a caneta é fixada num suporte transportador que se desloca na direção X (esquerda e direita), enquanto a caneta se move no prendedor na direção Y (para cima e para baixo).

O tipo tradicional de plotter é conhecido como plotter "achatado", porque fixa-se o papel numa chapa plana com um cavalet transportador no topo (ver a ilustração). A desvantagem é que o plotter deve ter, pelo menos, o tamanho do papel.

Para reduzir o tamanho adota-se uma versão de grande escala do plotter de quatro canetas (ver ilustração menor), onde o papel se move em uma direção e a caneta em outra. Exemplos disso são os plot-

Depósito de canetas

Com três canetas, a troca é feita mecanicamente. O suporte retorna ao depósito e muda a caneta em uso por uma de outra cor. Cores adicionais são trocadas manualmente.

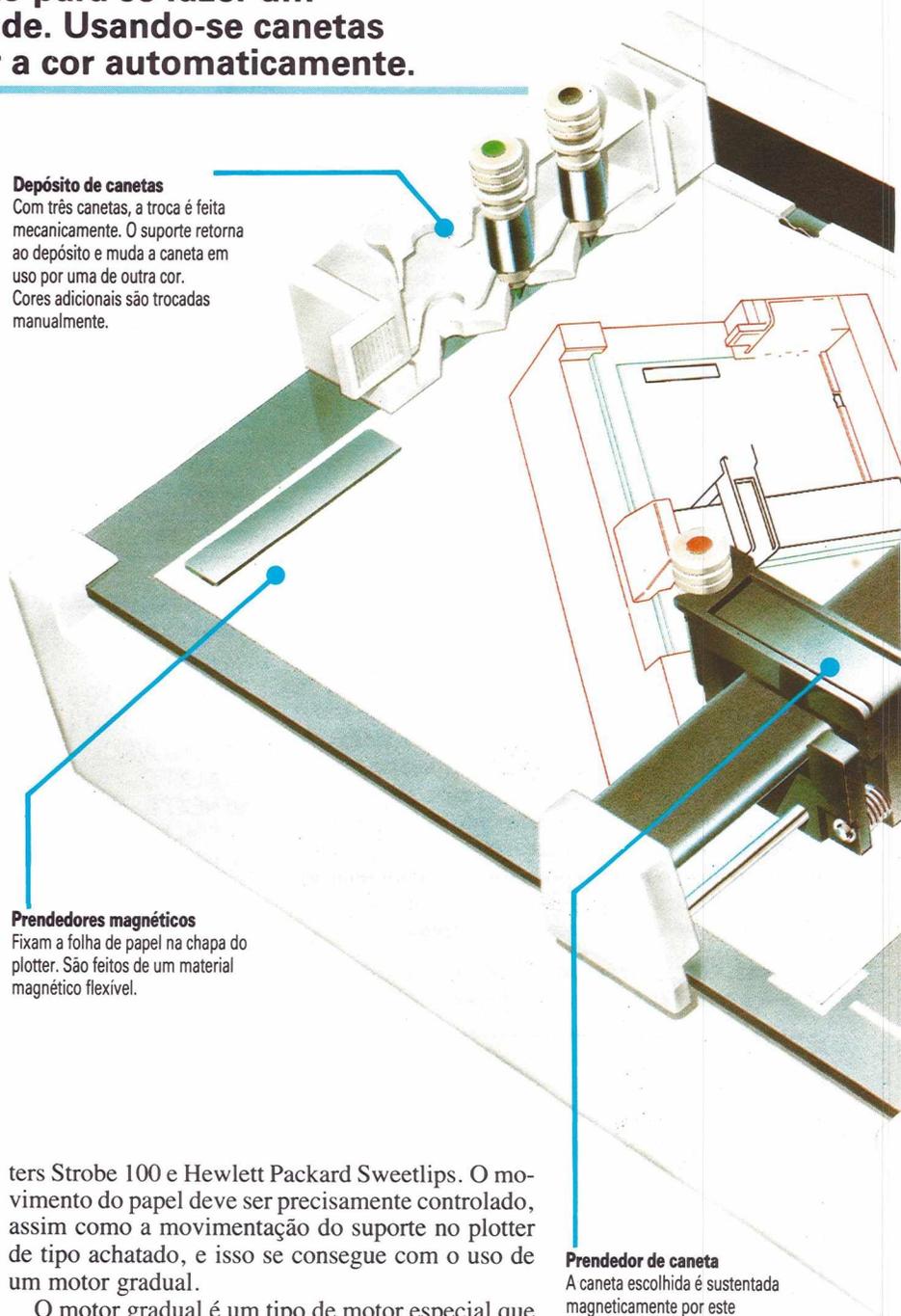
Prendedores magnéticos

Fixam a folha de papel na chapa do plotter. São feitos de um material magnético flexível.

ters Strobe 100 e Hewlett Packard Sweetlips. O movimento do papel deve ser precisamente controlado, assim como a movimentação do suporte no plotter de tipo achatado, e isso se consegue com o uso de um motor gradual.

O motor gradual é um tipo de motor especial que dá apenas uma fração de giro para cada impulso de energia utilizado. Esse motor encontra-se em unidades de disco, onde controla o posicionamento da cabeça de leitura/gravação na superfície do disco, e também em robôs (ver p. 176).

A ligação do plotter ao computador faz-se, geralmente, como a de uma impressora, pelo menos quanto à interface. Os plotters vêm com interfaces seriais (RS232) ou paralelas (Centronics ou IEEE488), que são conectadas à porta utilizada por uma impressora. A programação é, muitas vezes, um pouco mais complicada neste caso; em vez de



Prendedor de caneta

A caneta escolhida é sustentada magneticamente por este prendedor, que se move para baixo e coloca a pena em contato com o papel.



Suporte de canetas

O suporte pode ser posicionado em qualquer ponto da página (o eixo X) e o prendedor de caneta é movido ao longo do suporte (o eixo Y). As combinações esquerda/direita e os movimentos acima/abaixo permitem alcançar qualquer ponto da página.

apenas enviar os resultados do programa que serão impressos, a informação sobre o modo como esses resultados se apresentarão também deve ser enviada. Este processo ocorre de modo semelhante à elaboração de um diagrama na tela.

Pela maneira complexa de elaborar sua saída, os plotters são considerados "inteligentes". Isso significa que eles têm microprocessadores acoplados, que convertem os caracteres e instruções do computador em uma série de coordenadas, que a seguir são desenhadas pelos plotters. Os mais aperfeiçoados plotters possibilitam que formas mais complexas, como os círculos e as curvas, sejam desenhadas mediante o fornecimento apenas dos pontos iniciais — o plotter faz o resto. Os letreiros dos gráficos e diagramas e a coloração de mapas e gráficos de barra

são processos automáticos que simplificam a programação.

Muitos plotters vêm acompanhados de um software que permite que eles sejam usados diretamente dentro do programa, e não como simples cópias em papel do que aparece na tela. Caso esse tipo de software não seja fornecido, o usuário deverá elaborar as rotinas necessárias para transferir a informação da tela para códigos apropriados que guiem o plotter. Alguns plotters não possuem grupos de caracteres predefinidos e, por essa razão, mesmo os códigos para letras e números têm de ser criados. Isto permite que o usuário crie o estilo e o formato dos seus próprios caracteres. Uma vez gerada determinada forma, é possível tracejá-la em qualquer posição, orientação ou tamanho; assim, uma série de formas pode ser arquivada para facilitar o uso repetido. As rotinas para a representação de círculos, curvas e outras formas geométricas em seções de gráficos são muito úteis, sobretudo em gráficos comerciais que precisem ser criados. Os princípios da elaboração de um desenho a partir de coordenadas na tela assemelham-se aos necessários para a criação da forma geométrica no papel; portanto, a programação é, geralmente, bastante simples.

Motores graduais

Esses motores giram apenas alguns graus a cada impulso elétrico aplicado. Com transmissão adequada, eles dão movimento preciso à caneta e ao suporte.

Cartão de circuito

O plotter é um aparelho "inteligente" — ele pode receber comandos de alto nível como "desenhe um círculo com centro e raio especificados", e a caneta trabalha de modo adequado. O cartão de circuito contém seu próprio microprocessador, ROM e RAM.

Interface de conexão

O plotter é ligado ao computador através de uma interface padrão como RS232 (serial) ou Centronics (paralela). Para o computador é como se fosse uma impressora, apesar dos diferentes comandos que irão guiá-lo.

Controle de suspensão da caneta

Este permite que a caneta seja manualmente colocada em contato com o papel ou erguida.

Controles de movimento da caneta

A caneta pode ser manualmente posicionada na página por esses controles.

Plotter/ impressora de quatro canetas

Este mecanismo despertou a atenção da microindústria quando apareceu na impressora Sharp CE-150. Seus irmãos mais velhos, o CGP-115 da Tandy e o Oric MCP-40, ajudaram a trazer o baixo custo das impressoras coloridas para o usuário do microcomputador.

Como todas as boas idéias o sistema é simples. Um rolo de papel, puxado pelo mecanismo através de um cilindro, move-se para diante e para trás em etapas muito exatas, enquanto o condutor de caneta que segura quatro pequenas canetas esféricas move-se na superfície, da esquerda para a direita e vice-versa.

Para criar uma saída que pode ser de texto ou gráfico, o condutor de caneta gira até que a cor certa esteja em posição e, aí, a pena é pressionada no papel. As linhas horizontais são criadas quando a caneta se move no papel parado; as linhas verticais utilizam o movimento do papel com a caneta fixada num lugar. A combinação dos dois movimentos produz curvas e diagonais. A qualidade da impressão é de alto nível, embora as dimensões do papel não se prestem a muitas aplicações.



Alan Turing

O nome deste matemático inglês não é muito conhecido, pois grande parte de seu trabalho foi desenvolvido no serviço de espionagem, durante a Segunda Guerra Mundial.

Proeza matemática

Alan Turing (1912-1954) encontrou inspiração e relaxamento nas corridas de longa distância. Ele considerava excelente o efeito dos exercícios físicos em sua criatividade e agilidade mental.



As máquinas podem pensar?

Para responder a esta pergunta, Turing propôs seu famoso teste chamado jogo da imitação, mais conhecido como Teste de Turing. Um homem fica numa sala no controle de uma teleimpressora (teclado e impressora). Esta é ligada a uma teleimpressora (em outra sala e operada por outro homem) e também ligada ao computador em teste. Ao primeiro homem é permitido fazer qualquer pergunta ao segundo ou à máquina. Se ele for incapaz de saber quando está em comunicação com o homem e quando com o computador, então a máquina pode ser considerada inteligente. Afinal, prossegue o argumento, não podemos dizer ao certo se outras pessoas estão pensando e conscientes, a não ser observando suas reações às circunstâncias e comparando-as com as nossas.

O jovem Alan Turing demonstrou um enorme interesse pela ciência. Enquanto estava na escola, escreveu a sua mãe: “Parece que eu sempre quero extrair coisas novas até mesmo do que é mais comum na natureza”. Com frequência os matemáticos mostram desde cedo o seu talento; logo que Turing aprendeu a ler e escrever, ele fatorava números de hinos religiosos e desenhava bicicletas anfíbias.

Enquanto seu pai estava em Madras no Serviço Civil da Índia, Turing era homenageado na escola com prêmios, e a seguir com uma bolsa de estudos que o levou ao King's College, em Cambridge, na Inglaterra. E foi aí, primeiramente como aluno e depois como membro do Conselho do King's College, que ele começou a interessar-se por problemas de lógica matemática.

Em 1931, o matemático tcheco Kurt Gödel surpreendeu o mundo científico com a descoberta de que havia numerosos teoremas matemáticos que, embora verdadeiros, nunca poderiam ser provados. Alan Turing dedicou-se a analisar os teoremas que podiam ser comprovados.

Ele imaginou uma máquina, cuja construção não foi concretizada, que poderia efetuar de forma automática os processos geralmente desenvolvidos por um matemático. Para cada processo, haveria uma máquina: uma para somar, outra para dividir, uma terceira para calcular integrais, e assim por dian-

te. Mais tarde, foram denominadas máquinas de Turing.

Raciocinando sobre o funcionamento dessas máquinas imaginárias, Turing chegou a uma brilhante conclusão. Em vez de utilizar uma máquina específica para cada processo matemático, era possível desenhar um aparelho “universal” que tivesse condições de realizar tudo o que as máquinas especializadas podiam fazer, desde que fossem programadas para tal. Turing tinha elaborado por acaso a teoria dos computadores programáveis.

Quando começou a Segunda Guerra, Turing foi imediatamente recrutado do mundo acadêmico para a Escola de Códigos e Criptogramas do Governo em Bletchley Park, Buckinghamshire. Se não fosse a guerra, as máquinas de Turing não passariam de frutos de sua imaginação, mas o Bletchley Park foi incumbido do trabalho urgente e secreto de decifrar os códigos militares alemães.

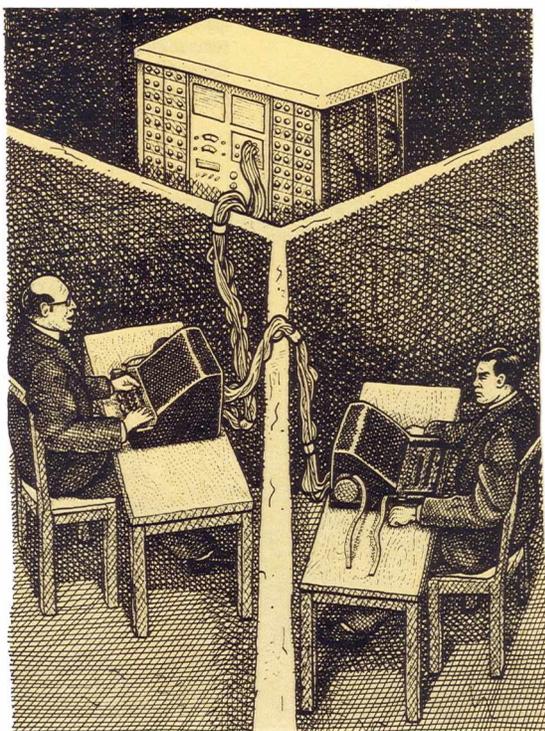
Como esses códigos podiam ser alterados diariamente, as máquinas deveriam decifrá-los antes que os novos fossem introduzidos. Bletchley Park tornou-se em pouco tempo grande centro de processamento de informações. No meio da guerra, Turing foi enviado aos Estados Unidos a fim de estabelecer códigos seguros para as comunicações transatlânticas entre os Aliados.

Como seu trabalho nessa época era altamente secreto, ainda hoje pouco se sabe sobre suas atividades. Entretanto, supõe-se que ele conheceu Von Neumann em Princeton, Nova Jérsei. No final da guerra, Turing foi solicitado a projetar um computador totalmente inglês, para o Laboratório Nacional de Física, e que seria chamado ACE (Automatic Computing Engine).

Esse nome foi dado, em parte, em homenagem a Charles Babbage e à sua máquina analítica. Assim como essa máquina pioneira, o ACE demorou muito a ser construído, mas pode-se dizer que era mais avançado que o ENIAC (ver p. 46).

Decepcionado com a morosidade da construção, Turing mudou-se para Manchester, onde tomou parte no projeto do computador da universidade local. Ao mesmo tempo, tornou-se consultor da empresa Ferranti, e mais tarde participou da construção dos primeiros computadores ingleses.

Turing era um excêntrico que perseguia o que considerava importante, sem convenções sociais ou impedimentos legais. Um amigo seu afirmou que ele era “divinamente louco” no que dizia respeito a encontrar defeitos nos outros, mas possuía indiscutível gênio científico. Em 1952, ele foi processado por homossexualidade, e suicidou-se dois anos depois.





O vôo simulado

Um moderno simulador de vôo imita perfeitamente o desempenho de um avião. Com isso, pode-se treinar pilotos em terra firme, com segurança e eficiência.



A partir de 1940, os pilotos de avião começaram a receber parte de seu treinamento inicial em terra, usando simuladores de vôo. Esses aparelhos, chamados Link Trainers, representaram a primeira geração de simuladores de vôo; eram máquinas bastante rudimentares, que não transmitiam ao aluno uma idéia real dos efeitos dos controles de um avião. Mesmo assim, davam os conhecimentos básicos da pilotagem.

Com o aparecimento dos primeiros jatos do tipo multimotor, como o Comet da De Havilland, evidenciou-se a necessidade de encontrar um método mais seguro e econômico para ensinar os pilotos a conduzir os mais diversos tipos de aeronaves, sem recorrer ao vôo real. A moderna indústria eletrônica criou então simuladores de vôo controlados por computador, que utilizam técnicas analógicas e digitais.

Armados sobre suportes hidráulicos, esses simuladores eram uma perfeita réplica da cabina dos aviões em que os pilotos deveriam voar. O sistema hidráulico, controlado por computador, possibilitava ao aparelho responder à manipulação dos instrumentos de vôo como se fosse realmente um avião. Permitia ainda que o instrutor simulasse situações de emergência para treinar os alunos.

O interior da cabina do simulador era exatamente igual ao original, com todos os instrumentos e controles. O treinamento incluía toda a tripulação, num trabalho conjunto, para garantir total segurança nos futuros vôos.

Para que a simulação fosse completa, faltava ape-

nas dar ao piloto a visão da pista, ao decolar ou pousar, se estivesse na cabina de uma aeronave. As primeiras tentativas para resolver esse problema envolveram a projeção de filmes, mas a necessidade de constante reorientação impediu que esse método tivesse sucesso.

Finalmente, partindo de experiências com imagens geradas por computador, que revolucionaram



Pronto para decolar

"Bravo, Alpha, três, dois, cinco, tudo pronto para a decolagem na pista um, dois. Velocidade do vento, um, cinco nós, rumo um, zero, cinco. Chame a torre quando ultrapassar os mil metros."

Mas a pista que se descortina à frente desse Boeing 737 existe apenas no "cérebro" do Novoview SP3, um simulador de vôo computadorizado, de alta precisão.

Aviões com pernas

Vistos de fora, os simuladores de vôo parecem caixas enormes, montadas sobre "pernas" hidráulicas.

Lá dentro, no entanto, a impressão é a de uma cabina de vôo equipada com os mais modernos instrumentos.

os desenhos animados (ver p.181), os técnicos acabaram escolhendo esse recurso para uso nos simuladores de voo. Com tal artifício, era possível finalmente oferecer ao piloto imagens "reais" das pistas de pouso e decolagem de um aeroporto. Essas imagens, passíveis de gravação em software de disco, podiam mesmo propiciar ao piloto a visualização de vários aeroportos, nos quais ele provavelmente iria decolar e aterrissar.

Outras aplicações

Mas os efeitos mais impressionantes dos simuladores foram desenvolvidos para treinar aviadores militares. Nesse caso, o objetivo principal não era habilitar o piloto de avião comercial a comandar uma aeronave de outro tipo, em que os sistemas eletrônicos modernos são tão avançados que o aparelho voa virtualmente sem maior interferência do homem; o treinamento visava a habilitar os tripulantes no reconhecimento de alvos militares e na técnica de

muladores de voo, para que os instrumentos da cabina respondam aos controles e também às influências do "ambiente externo", exatamente como num avião. Quando o piloto faz uma curva em mergulho, por exemplo, o cenário ao redor do "avião" deve girar também.

Na simulação de um combate aéreo, tudo fica ainda mais complicado, pois as manobras do suposto "adversário" são aleatórias (ou reproduzidas de uma vasta quantidade de manobras possíveis) para evitar que o piloto padronize suas reações e possa responder com criatividade e rapidez, de acordo com a ocasião.

Muitas escolas militares fazem treinamento por simulação, utilizando imagens de alta resolução, fotografias tiradas por satélites a grandes altitudes e diversos outros métodos sofisticados para conseguir filmes com imagens que se aproximem o mais possível da visão que um piloto teria da terra, se voasse baixo.

Usando-se simultaneamente um desses filmes e um simulador de voo, pode-se conseguir sensações e efeitos idênticos aos de uma missão real. Com esse treinamento, o piloto aprende a decolar e a aterrissar em sua base de origem, sem ter deixado a terra. Da mesma forma, aprende a decolar e a pousar nas grandes pistas de vários aeroportos e "vive" uma grande variedade de experiências quanto a diferentes alvos e missões.

Assumindo o controle

Um videogame da Microsoft coloca o jogador no comando de um Cessna 182, "livre" para decolar de qualquer uma das vinte representações de aeroportos americanos reais. Uma réplica do painel de controle do Cessna ocupa a parte inferior da tela; em cima, o "ambiente externo" está representado em perspectiva total. Esse videogame funciona sob instruções semelhantes às que são dadas aos pilotos em aprendizado.



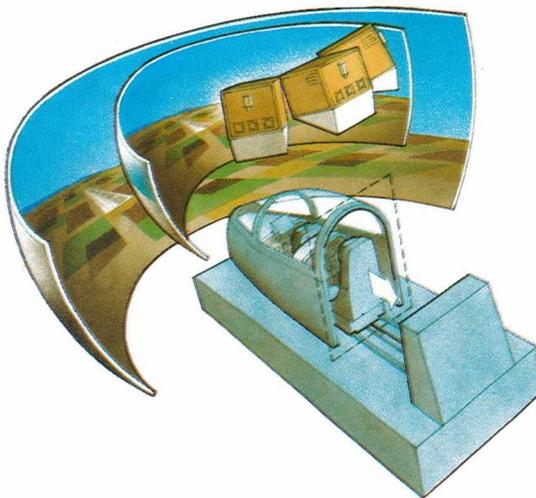
disparo de projéteis ar-terra ou ar-ar. Mesmo nesses casos, porém, a principal função do simulador de voo continuava sendo aumentar a eficiência da tripulação.

Quando tratamos das folhas eletrônicas (ver p.158), descrevemos uma técnica chamada "janela"; com ela, o operador pode verificar na tela do monitor parcelas de uma folha eletrônica. Um método semelhante é empregado nos mais modernos si-



Ilusão perfeita

Para projetar na tela do simulador de voo uma imagem convincente do ambiente que cerca o avião em condições reais, são necessários três projetores (cada um com três canhões: vermelho, azul e verde). Um espelho corretor ajusta a distorção de imagem causada pela curvatura da tela.



Depois de aprovadas no treinamento auxiliar de tripulações aéreas, essas técnicas de simulação agora são utilizadas pela marinha de muitos países para familiarizar seus oficiais com as particularidades de canais e portos; por mergulhadores, para o reconhecimento de objetos em más condições de visibilidade; e por astronautas, a fim de simular situações espaciais.

A qualidade técnica das imagens geradas por computador, junto com o controle da máquina pelo usuário, permite virtualmente reproduzir qualquer experiência visual; isso tornou possível também a criação de uma nova gama de videogames e diversões eletrônicas.

Alguns programas de jogos atuam como simples simuladores de voo, desafiando o jogador a realizar um voo bem-sucedido, sob vários tipos de dificuldades: pouco combustível, visibilidade precária, falhas mecânicas etc. Mas esses jogos não são iguais



aos sistemas usados para treinamento profissional; o que mais se aproxima de um sistema desses é o simulador de vôo da Microsoft.

No entanto, qualquer videogame que pretenda representar uma situação da “vida real” no monitor de computador é considerado um simulador de condições. A única diferença entre eles está na complexidade da representação e, talvez, na qualidade da imagem gerada pelo computador, muitas vezes comprometida pela qualidade do monitor que a reproduz.

Muitas vezes, as situações simuladas podem ser totalmente imaginárias, como no caso do Space Invaders ou do PacMan. Nesses jogos, o projetista imagina situações de jogo e as transfere para possíveis ações e condutas. Às vezes, eles constituem uma simulação direta da realidade, como os que imitam corridas de automóveis.

Jogos sérios

Nem todo software de simulação de vôo baseado em microcomputadores destina-se a jogadores. Existem programas de utilidade prática, como o Computer Assisted Learning (CAL), elaborados para auxiliar a navegação, o controle do tráfego aéreo e o planejamento de vôo, por exemplo. No entanto, o simulador de vôo da Microsoft, disponível para o IBM PC, caracteriza-se por ser um jogo muito divertido e, ao mesmo tempo, um bom exercício para quem deseja um dia ser piloto. Logo no início, o manual avisa que estão no jogo todos os instrumentos e equipamentos exigidos pelas normas federais de aviação dos Estados Unidos, para vôos diurnos e noturnos.

A parte inferior da tela reproduz de modo bastante aproximado o painel de instrumentos de um Cessna 182 (avião monomotor, em que se baseia o simulador), enquanto na parte superior aparecem imagens



tro áreas principais: os de Nova Iorque/Boston, Chicago, Los Angeles e Seattle.

Seria possível teoricamente “voar” de uma ponta a outra do país, de Nova Iorque a Los Angeles, via Chicago. Mas só existem pontos de abastecimento nessas áreas específicas; com isso, o “piloto” é forçado a manter-se dentro delas, sob pena de ver cair o “avião” por falta de combustível. Mesmo que fosse possível abastecer seguidamente a “aeronave”, o simulador trabalha em tempo real, a uma velocidade máxima de 240 km/h, o que significaria uma “viagem” de cerca de vinte horas — e quase toda em vôo direto por todo o país. Seguindo as regras do jogo, no entanto, o “piloto” pode “saltar” de uma área para outra facilmente.

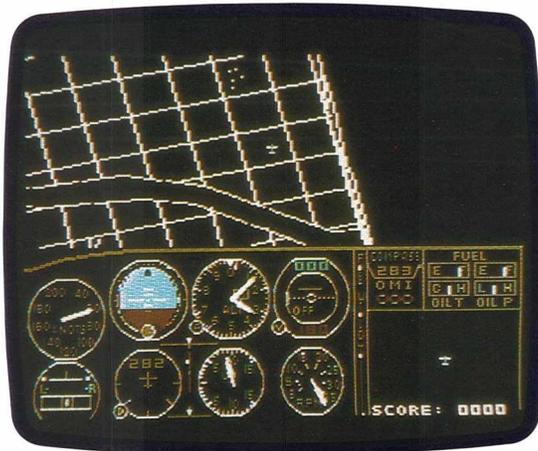
Mecanismos especiais permitem ao jogador escolher a hora do dia, a estação do ano (o que altera o horário do amanhecer e do anoitecer, as condições climáticas, velocidade e direção dos ventos, turbulência etc.).

O simulador de vôo da Microsoft baseia-se num avião simples. Mesmo assim, só foi possível construí-lo usando um código de máquina extremamente eficiente, de modo a permitir colocar o programa em 64 Kbytes.

Para simular com precisão a cabina de um jato, incluindo sensibilidade do aparelho aos movimentos dos controles, é preciso fazer muitas pesquisas.

“Avião inimigo... Direção 030A...”

A imagem acima foi produzida pelo sistema Novoview para simulação de vôo. Representa um ataque a alvo terrestre por um avião American A10. Por esse método, as tripulações podem aprender, “na prática”, a reagir corretamente às mais variadas situações de emergência.



coloridas e animadas, de média resolução, do que seria visto através do pára-brisa do avião. Com isso, obtêm-se efeitos pseudotrídimensionais, ou seja, há uma razoável sensação de movimento graças às constantes mudanças de perspectiva.

O simulador de vôo lúdico inclui seu próprio mapa de terra, cobrindo uma área de aproximadamente 2,5 milhões de km² — o território dos EUA, parte do Canadá, México e Antilhas — e abrangendo cerca de vinte aeroportos americanos em qua-



Colocando em ordem

Quando se projeta um programa que envolve a armazenagem ou a manipulação de informação, é importante ficar atento à estrutura, a fim de que os dados não precisem ser duplicados.

Trajeto reduzido

Uma firma tem interesses em seis cidades, e precisa enviar um caminhão para todas elas mensalmente. O computador da firma organiza o trajeto mais eficiente:

Cidade\$(1)	"Campinas"
Cidade\$(2)	"Cubatão"
Cidade\$(3)	"Limeira"
Cidade\$(4)	"Santos"
Cidade\$(5)	"Ubatuba"
Cidade\$(6)	"Valinhos"

Os nomes das cidades estão num arquivo seqüencial na fita, em ordem alfabética. São lidos do arquivo nessa ordem, através da matriz Cidade\$(). O computador estabeleceu que a melhor ordem é:

Limeira	Cidade\$(3)
Campinas	Cidade\$(1)
Valinhos	Cidade\$(6)
Cubatão	Cidade\$(2)
Santos	Cidade\$(4)
Ubatuba	Cidade\$(5)

Em vez de armazenar os nomes das cidades novamente nesta ordem, utilizando mais memória, o computador deposita apenas os números de posição na matriz Cidade\$(). Assim:

3	Campinas
1	Cubatão
6	Limeira
2	Santos
4	Ubatuba
5	Valinhos

A lista de números é um índice para a matriz Cidade\$(). Outros índices para a matriz poderiam ser criados, cada um dando uma ordem diferente para os nomes. A vantagem de se utilizar índices deste modo é que o arquivo original não precisa ser separado ou duplicado; basta separar e duplicar o índice.

Qualquer pessoa que tenha utilizado um sistema de índice por cartões (o catálogo de uma biblioteca, por exemplo) sabe o quanto ele pode ser útil. Se alguma vez você deixou que esse arquivo caísse no chão e saísse da ordem, notou que os cartões perderam toda a utilidade. Um catálogo de biblioteca contém muitas informações, mas, a menos que estas estejam ordenadamente arrumadas, o valor desse sistema de informação é nulo.

A essência de um sistema de dados não é a informação em si, mas sim sua organização. Considere, por exemplo, este item fictício numa lista de endereços comerciais:

Silva, J., rua Alta 15, Centro

Tomada isoladamente, essa informação é limitada. No entanto, se você sabe que ela procede de certa seção da lista de endereços, a informação ganha novo significado, relativo à estrutura do local onde se insere.

A mais simples estrutura de dados é denominada arquivo: um grupo de dados com características comuns. O nome do arquivo revela algo sobre a informação nele contida, e o fato de todas as informações estarem reunidas sob o mesmo nome torna mais fácil sua compreensão. O arquivo pode ser visto como uma grande unidade de informação, ou como grupos específicos de pequenas unidades. Um livro é um arquivo; lê-se uma autobiografia geralmente como um conjunto, ao passo que, num livro de culinária, as receitas formam uma coleção de dados individuais.

Se o arquivo é grande, a busca de uma informação específica significa partir do primeiro item e observar todos os seguintes até que se encontre a informação desejada. A isso atribui-se o nome de busca seqüencial ou acesso em série. Um arquivo organizado desse modo denomina-se arquivo seqüencial de informação. Um programa de televisão ou a fala humana constituem arquivos seqüenciais de informação.

Os arquivos seqüenciais são comuns por poderem ser alterados facilmente, e por espelharem, de certo modo, os métodos do pensamento humano. Mas muitas vezes revelam-se vagarosos e não abrangentes. Por isso, é comum sua divisão interna em subarquivos, que podem ser encontrados facilmente sem que para isso se precise pesquisar todo o arquivo. Os livros eram simples arquivos seqüenciais até que a criação de capítulos, páginas numeradas e índices transformou-os totalmente. Os capítulos funcionam como um subarquivo do livro, e as páginas são subarquivos do capítulo.

Um arquivo que não exige pesquisa seqüencial é chamado arquivo de acesso direto. Um conjunto de músicas numa fita magnética pode ser considerado

um arquivo seqüencial, já o mesmo conjunto em forma de disco transforma-se em um arquivo de acesso direto. Para se encontrar determinada música na fita é preciso tocá-la desde o começo e avançá-la gradualmente, enquanto, num LP, qualquer faixa pode ser encontrada se experimentarmos colocar o braço do toca-discos em todas elas.

O acesso direto depende de saber-se onde as coisas se localizam. Num livro, o índice faz esse trabalho para você. Saber onde as coisas estão significa trabalho — e custa dinheiro. A criação do índice de um livro é uma tarefa extra para o editor, e o tipo de informação contida no livro pode não justificar esta despesa: os romances, por exemplo, não são indexados, como em geral ocorre com livros de poesias.

Os computadores processam grandes quantidades de informação em alta velocidade, usando várias estruturas de dados. Estes devem ser armazenados permanentemente em fita magnética ou disco, de modo estruturado — em geral são utilizados arquivos seqüenciais —, e de forma bem diferente na memória central do computador.

Suponhamos que uma padaria tem o endereço de todos os seus clientes num arquivo seqüencial em fita, e deseja que o computador imprima roteiros para os motoristas que entregam pão. O arquivo na fita será mais ou menos assim:

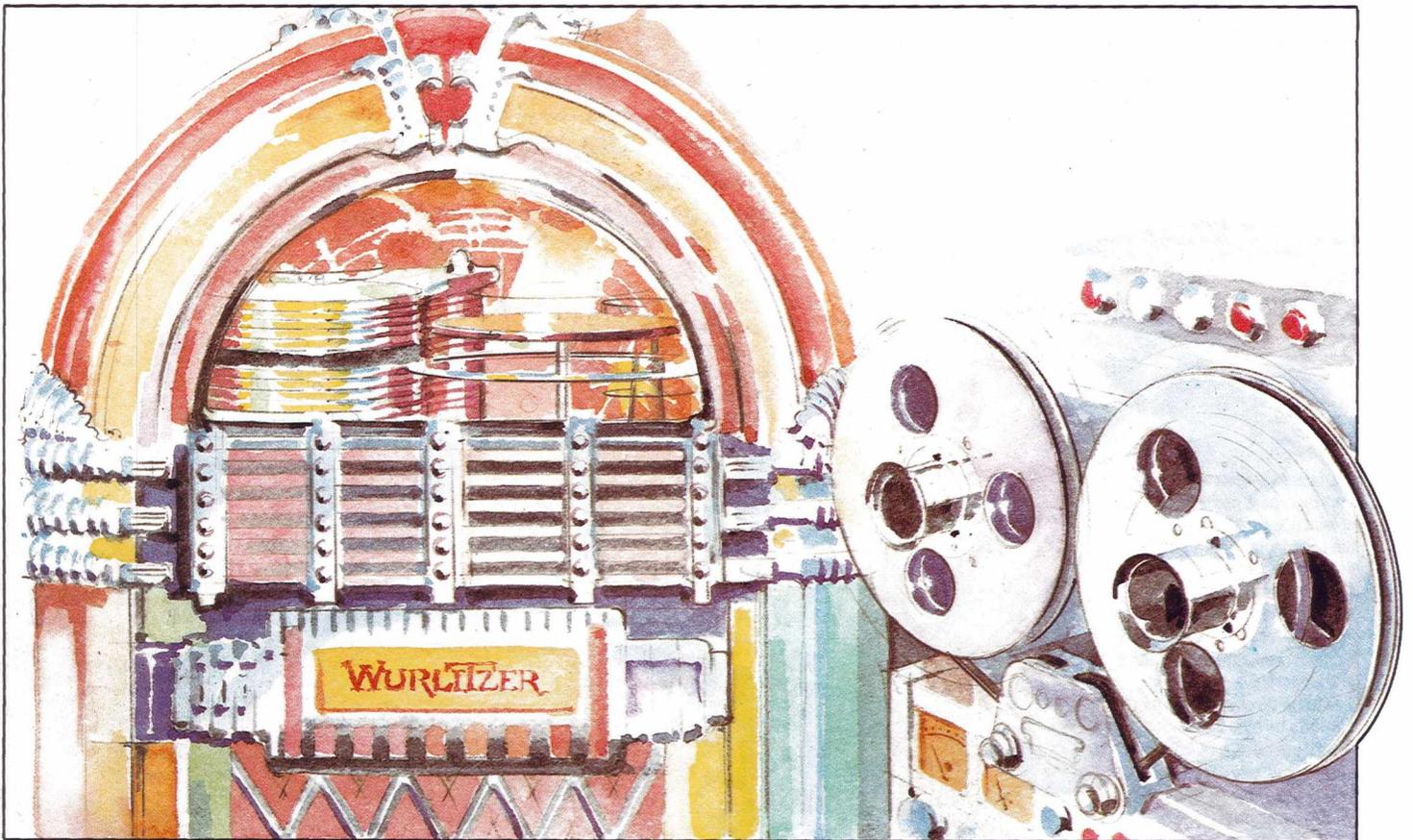
Almeida rua Alta 22
Arantes & Cia. av. Itabira 108
Duarte alameda da Saudade 49
Irmãos Wilson rua Alta 7
Soares estrada Paulistana 65
Vieira av. Atlântica 31

Quando o arquivo é lido da fita para a memória central, cada nome e endereço será depositado numa posição numerada e todas as posições reunidas formam um bloco de memória com seu próprio nome. Deste modo, o arquivo em memória fica assim:

NOME DE BLOCO: Estabelecimentos

- 1) Almeida rua Alta 22
- 2) Arantes & Cia. av. Itabira 108
- 3) Duarte alameda da Saudade 49
- 4) Irmãos Wilson rua Alta 7
- 5) Soares estrada Paulistana 65
- 6) Vieira av. Atlântica 31

Agora os itens de dados podem ser encontrados individualmente, bastando apenas que o bloco e a posição sejam nomeados. Estabelecimentos(4), por exemplo, contém Irmãos Wilson rua Alta 7. Essa estrutura, chamada matriz (ver p. 194), é mais comumente utilizada por computadores para processamento interno de dados. Assim como um livro traz



uma informação em cada página. Essa estrutura simples modifica o modo como vemos um bloco de dados anônimos. Não é necessário que os computadores conheçam o conteúdo de cada item; precisam apenas localizá-lo e saber o que fazer com ele.

Os dados na matriz Estabelecimentos() estão em ordem alfabética, mas não é certo que esta seja a ordem mais econômica para que se visitem os locais. Suponhamos que o computador descubra que o melhor programa de entrega é:

- 1) Irmãos Wilson rua Alta 7
- 2) Almeida rua Alta 22
- 3) Duarte alameda da Saudade 49
- 4) Arantes & Cia. av. Itabira 108
- 5) Vieira av. Atlântica 31
- 6) Soares estrada Paulistana 65

Essa ordem deve ser armazenada em outra matriz, mas isso significa que a mesma informação é armazenada na memória duas vezes. As pessoas que possuem micros sabem que a RAM é limitada, e talvez seja inconveniente ou impossível duplicarem-se dados desse modo. Portanto, um outro método deverá ser utilizado.

Caso os dados reais sejam substituídos por seus números de posição na matriz Estabelecimentos(), o programa de entrega fica assim:

NOME DE BLOCO: Entregas

- 1) 4
- 2) 1
- 3) 3
- 4) 2
- 5) 6
- 6) 5

As instruções para o motorista, portanto, são estas: "Primeiramente vá ao estabelecimento cujos detalhes estão armazenados em Estabelecimentos(4), a seguir vá para Estabelecimentos(1), depois para Estabelecimentos(3), e assim por diante. A única informação significativa no programa é a ordem em que os locais devem ser visitados; assim, isso é armazenado na nova matriz, Entregas.

Entregas() constitui agora um índice para a matriz Estabelecimentos() com o objetivo de entregas. Quando imprime esses dados, o computador usa os números da matriz Entregas() para imprimir na ordem apropriada os nomes e os endereços da matriz Estabelecimentos().

Nesse exercício simples, a informação — os nomes e os endereços — foi manipulada mas não modificada pelas diferentes estruturas de dados impostas. Uma estrutura de dados não deve modificar o conteúdo dos dados, mas sim dar significado a eles.

Do mesmo modo que podemos reorganizar a matriz Estabelecimentos() criando um novo índice de acordo com a matriz Entregas(), é possível produzir outros índices para diferentes fins. Quando estudamos bancos de dados (ver p. 124), percebemos que certa informação poderia ser selecionada utilizando-se ponteiros incluídos em cada registro individual. É o que ocorrerá se inserirmos em cada registro do arquivo Estabelecimentos() um ponteiro que mostre sua localização no programa de entregas. Poderíamos até mesmo aumentar o registro para incluir, por exemplo, um ponteiro que mostrasse os pedidos permanentes. O departamento de produção teria condições de percorrer o arquivo, extraíndo somente a informação relacionada com o número de pães solicitado por cada cliente.

Na pista certa

Um toca-discos automático contém duzentas músicas em cem discos arquivados. Para se escolher qualquer música, pressionam-se três teclas. O tempo médio entre a escolha e o início da música é de 15 segundos. Já para que uma música seja encontrada em uma fita, contendo as mesmas duzentas músicas, o tempo médio será de 1.500 segundos ou 25 minutos. O toca-discos é um aparelho de acesso direto, rápido, bastante especializado e caro. O toca-fitas é um aparelho de acesso seqüencial: lento, muito menos especializado, mas razoavelmente barato. O toca-fitas cassete ligado a um microcomputador é um aparelho de acesso seqüencial, enquanto uma unidade de disco flexível é um aparelho de acesso direto, mesmo quando utilizado para armazenar arquivos seqüenciais.

Entradas e saídas

Você não consegue encaixar uma peça quadrada em um orifício redondo. Do mesmo modo, dois dispositivos de computador não podem ser ligados, a menos que tenham interfaces compatíveis.

Interface para cassete

A entrada para cassete, ou interface, existente na maioria dos microcomputadores é efetivamente um tipo de interface serial. Uma vez que os dados têm de ser gravados em fita cassete comum, não são possíveis altas velocidades de transferência de dados. O sistema de circuitos toma da memória os bytes de dados a serem gravados e os converte em um conjunto de bits. Os bits 0 e 1 transformam-se em dois tons diferentes de som. Carregadas as fitas na memória, os tons são decodificados pelos circuitos da interface e os bits 1 e 0 resultantes, combinados em bytes de 8 bits para armazenamento na memória. As interfaces cassete da maioria dos microcomputadores são universais, ou seja, pode-se utilizar qualquer gravador comum. O conector empregado não é padrão, mas as tomadas DIN e os mini-jacks são os mais comuns.

Entrada analógica

Encontrada apenas nos computadores mais caros e de finalidade educacional, as entradas analógicas ligam o computador a dispositivos de laboratório, como medidores elétricos de temperatura ou fotossensores. A interface apenas apresentará uma ou mais linhas que poderão receber e ler a voltagem dentro de um nível específico. Cabe ao usuário assegurar-se de que o computador não seja ligado a uma voltagem maior que seu limite, porque danificaria seriamente o equipamento.

Porta de expansão da memória

Conecta a maioria das linhas, senão todas, que vêm diretamente do microprocessador, isto é, os buses de endereço, de dados e de controle. É aí que a memória suplementar será conectada e, em alguns computadores, até mesmo os periféricos produzidos pelo fabricante do equipamento. O conector é só um terminal PCB, embora, em alguns casos, seja uma tomada que admite um conector terminal, como os cartuchos de jogos (na verdade, uma forma de complemento da ROM).

Porta paralela

É uma interface paralela para finalidades diversas, que liga o micro a dispositivos periféricos. Os 8 bits de cada byte transmitido são enviados juntamente (em transmissão paralela), através de oito fios. Outros sinais são fornecidos para sincronizar o envio dos dados e garantir que estes sejam transmitidos apenas quando o dispositivo receptor estiver pronto para recebê-los.

Interface para unidade de disco

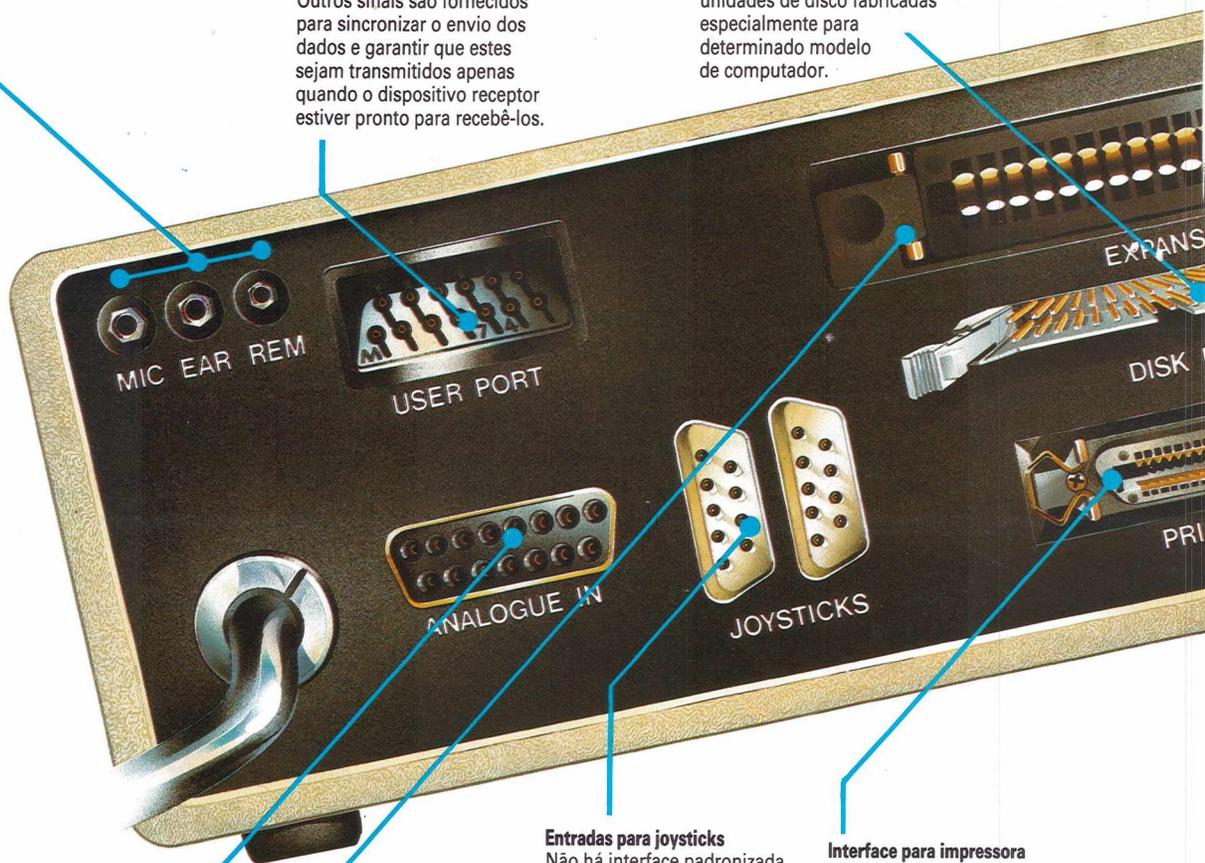
As unidades de disco são geralmente ligadas ao computador através de uma interface paralela. Não têm padronização e, com frequência, só é possível ligar unidades de disco fabricadas especialmente para determinado modelo de computador.

Entradas para joysticks

Não há interface padronizada para os joysticks. Muitos fabricantes adotam o padrão da Atari, para não produzir seus próprios joysticks. A maioria dessas interfaces tem apenas cinco linhas ativas: as provenientes de cada interruptor, nas quatro extremidades do movimento do joystick, e uma para o botão detonador. Os joysticks analógicos, todavia, exigem uma interface que admita uma série completa de voltagens, indicadoras da posição exata do comando. A maioria dos computadores tem mais de uma porta para joystick; às vezes, a mesma tomada é utilizada por mais de um dispositivo.

Interface para impressora

As interfaces para impressoras são bastante padronizadas, de acordo com o sistema desenvolvido pela Centronics Corporation; assim, há pouca dificuldade em obter uma impressora com interface Centronics que possa ser conjugada à interface impressora da maioria dos computadores. Os níveis de sinais, bem como as funções de sinais, também são padronizados: 0 e 5 volts para os binários 0 e 1, respectivamente. Os conectores utilizados e a atribuição dos sinais aos diferentes pinos não são padronizados; pode ser que você tenha de montar um fio conector especial para ligar a impressora ao computador.





Interface para vídeo (RF)

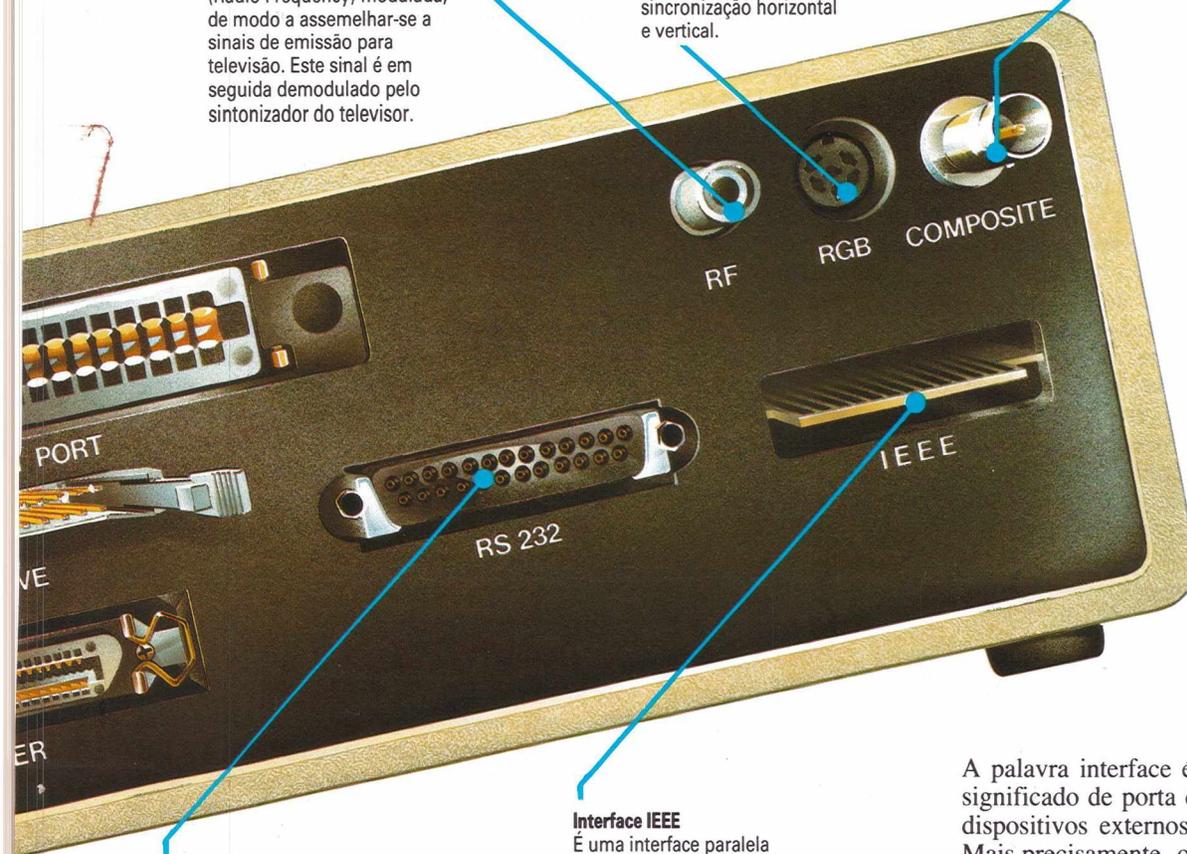
Todos os microcomputadores são projetados visando à sua conexão com uma unidade de vídeo, e, para a maioria deles, isto corresponde a um aparelho de televisão. Se for usada a tomada comum para antena, o sinal de vídeo deverá estar em RF (Radio Frequency) modulada, de modo a assemelhar-se a sinais de emissão para televisão. Este sinal é em seguida demodulado pelo sintonizador do televisor.

Interface RGB

Resultados de imagem colorida ainda melhores serão possíveis se os elementos necessários ao monitor do vídeo forem mantidos separados. A interface RGB possibilita sinais de vídeo vermelhos, verdes e azuis separados, além dos sinais de sincronização horizontal e vertical.

Interface para vídeo (composto)

Alguns televisores e muitos monitores de vídeo incorporam uma entrada para sinais compostos ou "vídeo" que evita o estágio de demodulação e possibilita a produção de imagens de melhor qualidade. Todos os elementos do sinal padrão de vídeo estão presentes na saída do computador, mas este sinal "composto" não necessita de modulação complementar para processamento pelo sintonizador de televisão.



Interface serial

Ao contrário de muitas outras, a interface serial RS232 é teoricamente definida de modo preciso pela padronização da Electrical Industries Association. Entretanto, poucos fabricantes se atêm ao padrão, o que dificulta o funcionamento de dispositivos seriais em certos computadores. Dos 25 pinos presentes na interface padronizada, geralmente apenas três são utilizados: o pino 2, para transmissão de dados seriais entre o computador e o dispositivo periférico; o pino 3, para recepção de dados provenientes do dispositivo periférico; e o pino 7, para o sinal terra. Tanto os dispositivos de emissão como os de recepção necessitam ser ajustados de modo que a velocidade da transmissão e o formato dos dados transmitidos sejam os mesmos.

Interface IEEE

É uma interface paralela universal, baseada no Hewlett-Packard Interface Bus, agora adotado como padrão pelo Institute of Electrical and Electronics Engineers. O padrão é bem definido, tanto física como eletronicamente. Ao contrário de outras interfaces de dados (como a interface serial RS232 e a paralela Centronics), que são conectadas a um único dispositivo periférico por vez, o bus IEEE pode ser ligado a quinze instrumentos (inclusive o computador) ao mesmo tempo. Dispositivos que incorporam interfaces IEEE abrangem impressoras, unidades de discos flexíveis, plotters, geradores de sinais, voltímetros e outros equipamentos para testes. Por se prestar tão bem ao uso em equipamentos de teste e medição, o bus IEEE é preferido em laboratórios e estabelecimentos industriais. Atualmente, apenas alguns microcomputadores apresentam interface IEEE.

A palavra interface é utilizada livremente, com o significado de porta de entrada ou tomada em que dispositivos externos são ligados ao computador. Mais precisamente, o termo se refere ao sistema de circuitos e ao software complementar, que tornam possível a conexão entre dois dispositivos quaisquer, vinculados ao computador.

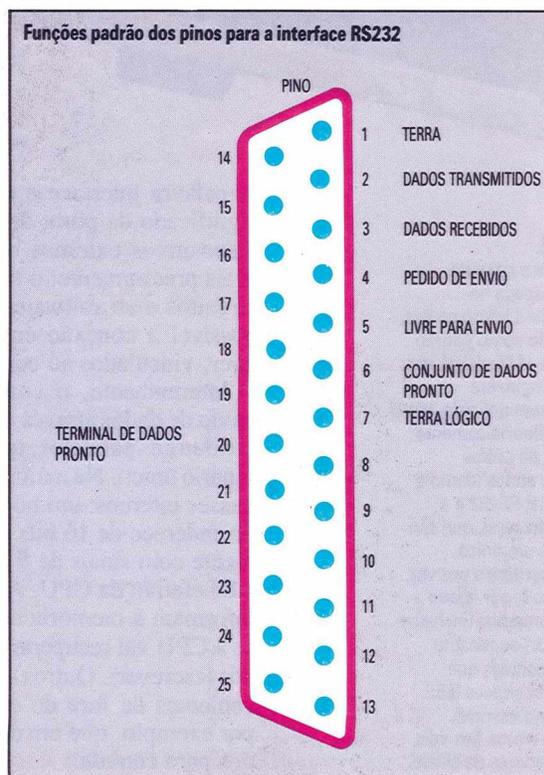
Internamente, o computador comunica-se pelo envio de dados através de "busses" — conjuntos de condutores paralelos, que levam, cada um, um sinal binário único. Na maioria dos computadores, há três busses internos: um bus de dados de 8 bits, um bus de endereço de 16 bits e um bus de controle, geralmente com sinais de 5 a 12 bits, que indicam o estado efetivo da CPU. Alguns dos sinais de controle informam à memória e aos dispositivos periféricos se a CPU vai recuperar dados (ler) ou depositar dados (escrever). Outros transmitem à CPU dados provenientes de fora do equipamento, informando-a, por exemplo, que um dispositivo periférico tem dados para entrada.

Ainda internamente, o computador manipula informação que consiste, em geral, em 8 ou 16 bits de cada vez. Assim, se for recuperar os dados na posição de memória 65535 (ou FFFF, pelo sistema hexadecimal), a CPU atribuirá o valor 1 a todos os dezesseis fios do bus de endereço, de modo a identificar essa posição. Se os conteúdos dessa posição de memória forem, por exemplo, 182 (B6, pelo sistema hexadecimal), esses dados serão colocados no bus de dados como oito dígitos binários: 10110110.

Quando se transferem os dados desse modo, 8 ou 16 bits por vez, a transferência chama-se paralela. Muitos periféricos são projetados para transmissão ou recepção paralelas de dados. As interfaces providas de dispositivos periféricos deste tipo denominam-se interfaces paralelas e a maioria dos computadores apresenta pelo menos uma tomada ou conector especiais para dispositivos paralelos.

Nem todos os dispositivos periféricos são projetados para recepção ou transmissão paralelas de dados. Alguns têm um fio único para comunicação com o computador, 1 bit por vez. Internamente, utilizam ainda os dados como bits de 8 ou 16 bytes, mas os bits de cada byte serão transmitidos ou recebidos, um de cada vez, a partir do "último bit significativo" do byte, concluindo com o bit "mais significativo". Cada byte é dividido em uma série de bits, enviados um após o outro, e reconstituídos em 1 byte na outra extremidade, mediante circuitos especiais de conversão de paralelo para serial e de serial para paralelo.

Tanto as interfaces seriais como as paralelas podem ser induzidas a transmitir dados, quer para fora quer para dentro do computador. Os computadores, de modo geral, incluem outras interfaces, que ou sempre enviam dados para fora (como o circuito de saída da televisão) ou sempre aceitam dados que chegam (por exemplo, portas para joysticks).



Interface serial

Existe uma interface serial padrão chamada RS232, para a qual os detalhes dos níveis de sinal e do ajuste de pinos são definidos. Mesmo o tipo de conector é também especificado. Infelizmente, não se costuma adotar esse padrão de modo rigoroso, e a execução de trabalho por conexões seriais pode tornar-se difícil. A interface RS232 é bidirecional serial, com um

pino (terminal) para transmissão e outro para recepção de dados do computador.

Os dados são enviados, 1 bit de cada vez, pelo terminal de "dados transmitidos" e, em sentido contrário, através do terminal de dados recebidos. O conjunto de bits pode exigir uma série de formatos standard, mas sem distinguir qual está sendo usado, desde que tanto o dispositivo emissor como o receptor utilizem o mesmo formato.

Uma vez que cada byte de informação é enviado como um conjunto em série de bits, o software que controla a interface deve ter um modo de indicar quando o primeiro bit de dados se inicia e quando o último se encerra. A convenção mais comumente usada tem um único "bit de início" (0, em lógica booleana), seguido de 8 bits de dados, e com um único "bit de encerramento" (dígito lógico 1).

A velocidade com que os dados são transmitidos deve ser especificada antecipadamente; caso contrário, o padrão de impulsos que representa os 8 bits de dados em dígitos 0 e 1 poderá ser interpretado de maneira errônea. Essa velocidade de transmissão de dados é conhecida como velocidade baud, nome que faz referência ao inventor francês Baudot. A velocidade baud varia de 75 a 9.600 bauds. Esses números correspondem aos 75 e 9.600 bits transmitidos por segundo, e como há, geralmente, 10 bits para cada caractere, a transmissão do caractere corresponde a um décimo da velocidade baud.

Interfaces paralelas

A interface paralela transmite ou recebe dados, 1 byte de cada vez, mas, além das oito "linhas de dados", será necessário fornecer outros sinais, de modo que o computador e o equipamento periférico tenham a indicação de quando é possível, ou não, a transmissão de dados. O tipo mais comum de interface paralela é a Centronics (da fabricante de impressoras Centronics Corporation, dos Estados Unidos). O conector utilizado e a atribuição de sinais a pinos específicos diferem bastante de um fabricante para outro. A maioria das interfaces Centronics possibilita pelo menos os seguintes sinais:

DADOS 0 a DADOS 7	
	Oito fios para transportar os 8 bits do byte que está sendo transmitido.
ADK	Sinal de entrada para o computador; indica que o dispositivo receptor está preparado para receber dados.
GND	O fio terra que fornece uma referência comum de 0 volt, tanto para o computador como para o dispositivo periférico.
BUSY	Sinal do dispositivo periférico para o computador; indica que o periférico não pode aceitar dados.
STROBE	Sinal de saída do computador; indica ao periférico que os dados estão prontos para leitura.

Muitos outros dispositivos, além das impressoras, adotam a interface paralela Centronics, e a conexão com o computador pode exigir apenas a aquisição de um fio especial ou a instalação elétrica feita pelo usuário. De modo geral, não haverá alterações no software necessário para "controle" dos periféricos.



Números ao acaso

Os computadores são inteiramente racionais, e desempenham certas tarefas melhor do que o ser humano. Entretanto, é impossível para eles escolher um número ao acaso.

Para um ser humano, escolher um número ao acaso é muito fácil, e até mesmo divertido. Mas, para um computador, a tarefa é simplesmente impossível.

Como os computadores obedecem apenas aos comandos de seu operador e precisam de uma razão para toda operação que fazem, qualquer número criado por eles é resultante de instruções recebidas. Não importa que essas instruções sejam complexas; o número é teoricamente previsível e, para obtê-lo outra vez, precisamos apenas repetir a seqüência de operações feitas pelo computador. Dessa forma, vê-se que os números "escolhidos" pela máquina são na verdade calculados por ela mediante processos bastante precisos.

No entanto, pode-se obter do computador números realmente aleatórios mediante certos processos. O ERNIE (Electronic Random Number Indicator Equipment, equipamento eletrônico para indicação de números aleatórios) sorteia números partindo do movimento casual dos elétrons livres.

Von Neumann foi o primeiro estudioso a considerar a possibilidade de obter do computador números pseudocasuais, criados matematicamente. Como primeiro passo, ele elevava ao quadrado um número qualquer de quatro algarismos — 4321, por exemplo. Do resultado de oito dígitos, 18671041, considerou os quatro centrais (6710) como um número casual. A partir dele, criou o seguinte. Ou seja, elevando ao quadrado 6710, obteve 45024100; os algarismos centrais são 0241, que originariam o próximo número casual, e assim por diante. As seqüências numéricas obtidas por esse método só voltariam a repetir-se depois de esgotadas todas as combinações possíveis envolvendo quatro algarismos (no caso, 9999).

Os computadores modernos partem de outros métodos e conseguem números ainda mais aleatórios, podendo gerar, por exemplo, um número "casual" aproximadamente a cada 1,5 milésimo de segundo e a repetição da seqüência de números só ocorreria depois de 150 dias de funcionamento contínuo.



O método Monte Carlo

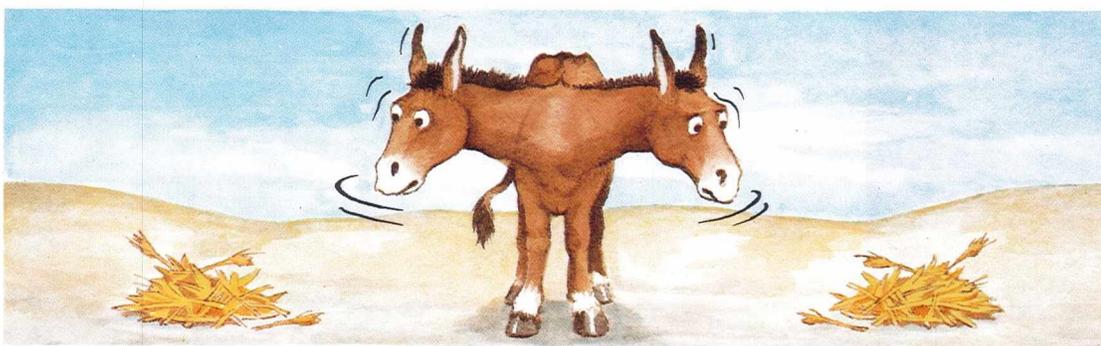
Não é difícil calcular a área territorial de uma ilha. Pode-se conseguir um resultado aproximado tendo-se somente um mapa do litoral: basta empregar o método idealizado por John von Neumann. Na figura ao lado, o contorno da Grã-Bretanha foi posto dentro de um retângulo de área conhecida, pontilhado aleatoriamente. Alguns pontos caíram sobre a área terrestre que se quer determinar. A soma deles todos vai ser proporcional à área da ilha. Vê-se que, dos 40 pontos aleatórios, 24 caíram sobre o mar e 16 ficaram sobre terra. De posse desses dados, arma-se o seguinte cálculo: $16/40 \times (1.050 \times 550) = 16/40 \times 577.500 = 231.000 \text{ km}^2$. Maior número de pontos casuais resultaria em melhor aproximação: 229.523. A ilha tem a superfície real de 229.890 km^2 .

Números aleatórios foram, a princípio, gerados eletronicamente para uso de técnicos em telefonia, numa experiência para simular as flutuações da demanda nas centrais telefônicas.

Hoje em dia, há usos diversos para os números aleatórios, não só nos jogos de computador para simulação de processos acidentalmente variáveis, como no cálculo de difíceis funções matemáticas.

O burrinho racional

Imagine que um burrinho totalmente racional foi posto entre duas pilhas idênticas de feno, para que escolhesse uma delas. Mas ele não pode preferir uma que seja maior porque as duas têm o mesmo volume de feno; não pode optar por uma mais próxima porque as duas estão equidistantes dele. Poderia escolher a pilha que olhasse em primeiro lugar. Mas por que essa pilha e não a outra chamaria mais sua atenção? Sem razão para decidir-se, o animal ficaria imóvel e acabaria por morrer de fome. De modo parecido, os computadores não são capazes de gerar números ao acaso, porque fazem tudo estritamente de acordo com processos racionais.





Micro Engenho 2

Edita textos com todos os caracteres ortográficos da língua portuguesa, realiza cálculos, arquiva dados, desenha gráficos e figuras.

O Micro Engenho 2 da Spectrum é utilizado por pequenas, médias e grandes empresas, atuando como uma eficiente ferramenta de trabalho na elaboração de orçamentos, previsões de vendas, cálculos de custos, controle de estoque e fichários, além de editar cartas, documentos ou material jornalístico.

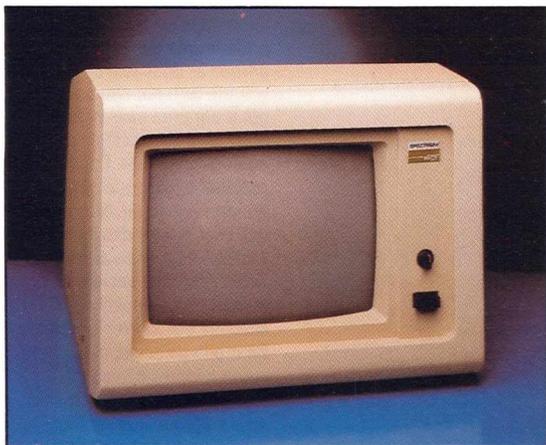
Sendo compatível com dois dos mais famosos microcomputadores estrangeiros (o Apple II Plus e o Apple IIe, fabricados pela Apple Computer Inc.), o Micro Engenho 2 tem acesso a vasta quantidade de programas desenvolvidos para os mais diversos ramos de atividade.

A Spectrum fabrica, além do Micro Engenho 2, o monitor de vídeo com tela de fósforo verde, modelo VD 121, e toda a família de interfaces e cartões: interface para impressora RS232 C, interface paralela, interface de comunicação assíncrona, cartão 80 colunas, cartão Z80 para CP/M, cartões de expansão de memória para 128 e 192 K. Fabrica também acionadores de jogos e modulador de RF.

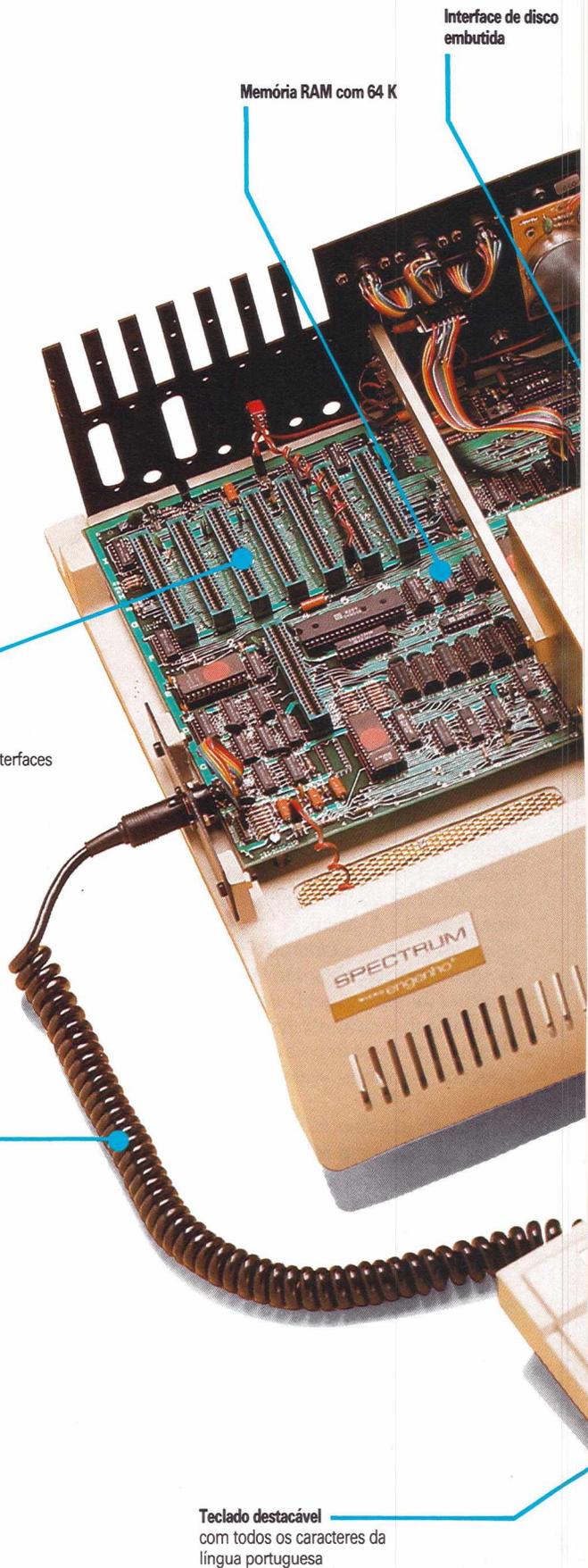
O Micro Engenho 2 vem com espaço para embutir duas unidades de disco (5 1/4 pol.) e possui a interface de disco residente na placa principal. Nesta placa encontram-se ainda sete conectores de interfaces (slots). Produzindo sinal de vídeo padrão Pal-M, o Micro Engenho 2 dispensa o uso de uma interface, quando utilizado com televisor em cores nacional.

Um teclado numérico reduzido permite que valores sejam digitados com maior rapidez. Seu teclado alfanumérico é o primeiro a oferecer uma disposição de teclas idêntica à de uma máquina de escrever, com cedilha e acentos (inclusive trema), permitindo acesso fácil a maiúsculas e minúsculas.

O Micro Engenho 2 oferece ainda minuciosa definição gráfica, possibilitando a manipulação de mais de 50.000 pontos na tela de vídeo.



Monitor de vídeo VD 121 com tela de fósforo verde



Teclado destacável com todos os caracteres da língua portuguesa



Micro Engenho 2

MICROPROCESSADOR

6502

CLOCK

1,00 MHz

MEMÓRIA

64 K de RAM
16 K de PROM

VÍDEO

Opera com monitores de vídeo profissional, além de poder ser conectado a qualquer televisor nacional, colorido e preto e branco.

TECLADO

Em módulo separado, possui todos os caracteres da língua portuguesa, com 80 teclas auto-repetitivas, e teclado numérico independente.

LINGUAGEM

BASIC, PASCAL, Assembler, FORTH, BASIC compilado.

PERIFÉRICOS

Monitor de vídeo, cassete, unidades de disco (5 1/4 pol.), impressora, modem, paddle, joystick.

DOCUMENTAÇÃO

Todos os equipamentos fabricados pela Spectrum, desde a CPU até as menores das interfaces, são acompanhados por um completo manual de instalação e operação.

Fonte de alimentação
com opção para 110 e 220 volts

Espaço previsto para até
duas unidades de disco

Teclado numérico
independente

Novas estruturas

Todas as versões da linguagem BASIC apresentam estruturas chamadas de controle. E alguns equipamentos possibilitam grande variedade de alternativas, com diferenças sutis.

As primeiras dez unidades do curso de programação BASIC abrangeram quase todos os aspectos mais importantes desta linguagem. Agora apresentamos um apanhado geral dos tópicos já vistos, focalizando algumas questões interessantes, com indicações do que faremos mais adiante.

Lembramos em primeiro lugar: linguagens de alto nível como o BASIC proporcionam ao usuário um conjunto de instruções traduzidas internamente para uma forma compatível com o computador. Qualquer programa para o computador pode ser escrito usando-se apenas duas construções simples: as seqüenciais e as estruturas de controle. Em BASIC, são construções essenciais: IF-THEN-ELSE e WHILE-DO. A maioria das outras linguagens de computação apresenta muitas outras estruturas.

A construção seqüencial permite que uma tarefa seja dividida em um conjunto de subtarefas que realizam a tarefa principal, quando executadas em seqüência. O tamanho das subtarefas depende da linguagem; em BASIC, as subtarefas são representadas pelas instruções escritas em cada linha e a seqüência, por números de linha. Desse modo, se a tarefa for multiplicar por 10 o valor atribuído a uma variável, a seqüência poderá ser a seguinte:

```
110 INPUT N
120 LET N = N * 10
130 PRINT N
```

Além das construções seqüenciais, também necessitamos de estruturas de controle. Estas são construções que alteram a ordem de execução das instruções no programa.

A estrutura de controle mais simples da linguagem BASIC é a instrução GOTO, que provoca um salto (ou desvio) incondicional, ou seja, sem condições determinadas. A execução do programa é desviada para um número específico de linha, sem necessidade de satisfazer a testes ou condições. A instrução GOSUB é também um desvio sem condições determinadas, mas, com ela, o programa sempre retorna ao ponto imediatamente seguinte à instrução GOSUB e admite-se seu uso em programação estruturada.

A estrutura de controle IF-THEN-ELSE também existe em BASIC, sob a forma da instrução IF-THEN, e apresenta a seguinte sintaxe ("sintaxe" é o jargão de computação que corresponde à "forma"):

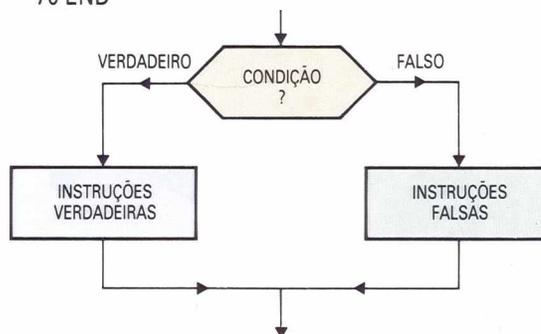
SE (IF) a condição especificada for verdadeira, ENTÃO (THEN) executar a instrução especificada; CASO CONTRÁRIO (ELSE), executar a próxima instrução

Observe que em linguagem BASIC standard, o elemento ELSE da instrução IF-THEN-ELSE está implícito. Em algumas versões ou adaptações de BASIC e

em outras linguagens, como PASCAL, o elemento ELSE faz parte da instrução.

A instrução IF-THEN-ELSE (IF-THEN, em BASIC) executa uma das duas subtarefas, dependendo de ser verdadeira ou não certa condição. Observe o seguinte programa, elaborado para determinar a raiz quadrada de números fornecidos através do teclado, a menos que se forneça um valor "flag" de -9999 (para encerrar o programa):

```
10 PRINT "FORNECER UM NUMERO"
20 INPUT N
30 IF N = -9999 THEN GOTO 70
40 LET S = SQR(N)
50 PRINT "A RAIZ QUADRADA DE "; N; " E "; S
60 GOTO 10
70 END
```



Estrutura de controle IF-THEN-ELSE

Se a condição for Verdadeira, as instruções Verdadeiras serão executadas. Se a condição for Falsa, as instruções Falsas serão executadas.

A linha 30 está efetivamente indicando: "SE (IF) é verdade que a variável N = -9999, ENTÃO (THEN) vá ao final do programa; CASO CONTRÁRIO (ELSE) (se não é verdade que N = -9999), execute a linha seguinte para calcular a raiz quadrada".

Outra estrutura de controle fundamental (WHILE-DO) não está diretamente disponível em BASIC, mas pode ser simulada com facilidade. WHILE-DO é um tipo de "loop de execução" e significa "repetir uma instrução ou conjunto de instruções, ENQUANTO (WHILE) uma determinada condição for verdadeira"; ou seja, "ENQUANTO (WHILE) uma condição for verdadeira, EXECUTAR (DO) determinado procedimento".

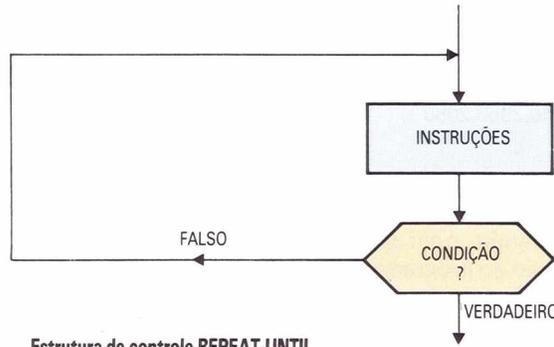
WHILE-DO sempre testa a condição antes de executar as instruções; assim, se o teste for negativo na primeira passagem, as instruções (chamadas corpo do loop) não serão executadas. Como exemplo, observe um programa de jogos que indica ao jogador "PRESSIONAR A BARRA DE ESPACO, QUANDO PRONTO". Esta parte do programa poderia ser desenvolvida (em "pseudolinguagem", ou inglês ou português simplificado) como:



ENQUANTO (WHILE) a barra de espaço não for pressionada, FAÇA (DO) a sondagem do teclado começar o jogo

Em BASIC, isto seria escrito:

```
250 PRINT "PRESSIONAR A BARRA DE ESPACO
    QUANDO ESTIVER PRONTO"
260 IF INKEY$ <> " " THEN GOTO 260
270 GOSUB *INICIAR*
```



Estrutura de controle REPEAT-UNTIL

O loop é repetido até que a condição se torne Verdadeira. As instruções terão de ser executadas, pelo menos uma vez.

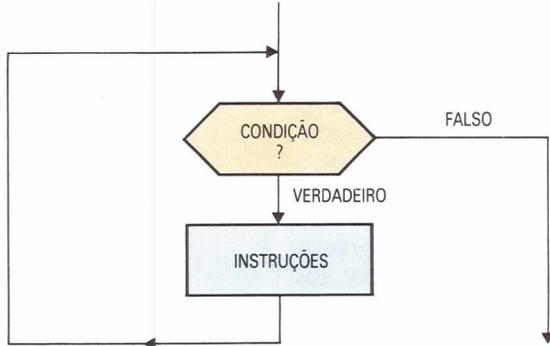
não encontrada diretamente em BASIC, mas simulável com facilidade, é a instrução REPEAT-UNTIL. Aqui, o teste de condição se apresenta após a parte principal do loop; assim, a instrução ou instruções no corpo principal do loop serão repetidas, pelo menos uma vez. Observe o seguinte programa "gerador de números aleatórios":

```
10 PRINT "PRESSIONAR A BARRA DE ESPACO"
20 FOR X = 0 TO 1 STEP 0
30 LET R = R + 1
40 IF R > 9 THEN LET R = 1
50 IF INKEY$ = " " THEN LET X = 2
60 NEXT X
70 PRINT "O VALOR DE R E ";R
```

Neste caso, o corpo principal (aumento do valor da variável R) tem de ser executado, pelo menos uma vez, já que o teste para desvio do loop (IF INKEY\$ = " ") só se apresenta após a instrução para aumento (LET R = R + 1).

Outra estrutura de controle, não essencial, mas útil, é a geralmente denominada CASE. Em BASIC a estrutura CASE realiza-se mediante a instrução ON-GOTO ou a instrução ON-GOSUB. Eis como funciona: ON-GOTO é uma instrução de desvios múltiplos que incorpora vários testes condicionais IF-THEN em uma única instrução. Observe um fragmento de programa que converte os números de 1 a 7 nas palavras que representam os sete dias da semana:

```
1050 IF D = 1 THEN GOTO 2020
1060 IF D = 2 THEN GOTO 2040
1070 IF D = 3 THEN GOTO 2060
1080 IF D = 4 THEN GOTO 2080
1090 IF D = 5 THEN GOTO 3000
2000 IF D = 6 THEN GOTO 3020
2010 IF D = 7 THEN GOTO 3040
2020 PRINT "SEGUNDA-FEIRA"
2030 GOTO *END*
2040 PRINT "TERÇA-FEIRA"
2050 GOTO *END*
2060 PRINT "QUARTA-FEIRA"
2070 GOTO *END*
2080 PRINT "QUINTA-FEIRA"
2090 GOTO *END*
3000 PRINT "SEXTA-FEIRA"
3010 GOTO *END*
3020 PRINT "SABADO"
3030 GOTO *END*
3040 PRINT "DOMINGO"
3050 GOTO *END*
```



Estrutura de controle DO-WHILE

O loop é repetido enquanto a condição for Verdadeira. As instruções podem não ser executadas (se a condição inicial for Falsa).

A linha 260 indica que SE (IF) INKEY\$ não é igual (<>) a um espaço (" "), ENTÃO (THEN) retorne e verifique o teclado novamente. Um modo um pouco mais elegante de desenvolver esse procedimento seria:

```
250 PRINT "PRESSIONAR A BARRA DE ESPACO
    QUANDO ESTIVER PRONTO"
260 FOR X = 0 TO 1 STEP 0
270 IF INKEY$ = " " THEN LET X = 2
280 NEXT X
290 GOSUB *INICIAR*
```

Neste fragmento de programa, o loop (para sondagem do teclado) é executado somente se a barra de espaço não for pressionada. Se a barra tiver sido pressionada (i.é, INKEY\$ = " "), então o programa vai do loop FOR-NEXT para a linha 290, que é a chamada para a sub-rotina INICIAR. (Obs.: Estamos utilizando "títulos" ou nomes para as sub-rotinas. Muitas versões de BASIC não admitem a chamada de sub-rotinas pelo nome; assim, você terá de utilizar os números de linha das sub-rotinas.)

Não apresentamos antes a instrução STEP, e o exemplo dado é uma aplicação pouco comum. Quando usada em um loop FOR-NEXT, a instrução STEP permite que o "contador" seja aumentado em unidades diferentes de um. A instrução FOR I = 1 TO 10 STEP 2 fará com que a variável I tenha o valor 1 na primeira passagem pelo loop, seguido dos valores 3, 5, 7 e 9. O aumento seguinte (para 11) ultrapassará o limite de 10 e, desse modo, o loop estará encerrado. Pode-se fazer com que o contador execute a contagem regressiva. A instrução FOR I = 10 TO 1 STEP -1 fará a variável I executar regressivamente a contagem de 10 para 1. A utilização da instrução STEP 0 é um artifício engenhoso para assegurar que o loop nunca termine, a menos que a variável X seja "aumentada artificialmente", como no caso da instrução IF-THEN.

Outra estrutura de controle muito útil, também

DECIMAL	BINÁRIO	CARACTERES
32	0 0 1 0 0 0 0 0	=(espaço)
33	0 0 1 0 0 0 0 1	= !
34	0 0 1 0 0 0 1 0	= "
35	0 0 1 0 0 0 1 1	= #
36	0 0 1 0 0 1 0 0	= \$
37	0 0 1 0 0 1 0 1	= %
38	0 0 1 0 0 1 1 0	= &
39	0 0 1 0 0 1 1 1	= '
40	0 0 1 0 1 0 0 0	= (
41	0 0 1 0 1 0 0 1	=)
42	0 0 1 0 1 0 1 0	= *
43	0 0 1 0 1 0 1 1	= +
44	0 0 1 0 1 1 0 0	= ,
45	0 0 1 0 1 1 0 1	= -
46	0 0 1 0 1 1 1 0	= /
47	0 0 1 0 1 1 1 1	= 0
48	0 0 1 1 0 0 0 0	= 1
49	0 0 1 1 0 0 0 1	= 2
50	0 0 1 1 0 0 1 0	= 3
51	0 0 1 1 0 0 1 1	= 4
52	0 0 1 1 0 1 0 0	= 5
53	0 0 1 1 0 1 0 1	= 6
54	0 0 1 1 0 1 1 0	= 7
55	0 0 1 1 0 1 1 1	= 8
56	0 0 1 1 1 0 0 0	= 9
57	0 0 1 1 1 0 0 1	= :
58	0 0 1 1 1 0 1 0	= ;
59	0 0 1 1 1 0 1 1	= /
60	0 0 1 1 1 1 0 0	= =
61	0 0 1 1 1 1 0 1	= >
62	0 0 1 1 1 1 1 0	= <
63	0 0 1 1 1 1 1 1	= @
64	0 1 0 0 0 0 0 0	= A
65	0 1 0 0 0 0 0 1	= B
66	0 1 0 0 0 0 1 0	= C
67	0 1 0 0 0 0 1 1	= D
68	0 1 0 0 0 1 0 0	= E
69	0 1 0 0 0 1 0 1	= F
70	0 1 0 0 0 1 1 0	= G
71	0 1 0 0 0 1 1 1	= H
72	0 1 0 0 1 0 0 0	= I
73	0 1 0 0 1 0 0 1	= J
74	0 1 0 0 1 0 1 0	= K
75	0 1 0 0 1 0 1 1	= L
76	0 1 0 0 1 1 0 0	= M
77	0 1 0 0 1 1 0 1	= N
78	0 1 0 0 1 1 1 0	= O
79	0 1 0 0 1 1 1 1	= P
80	0 1 0 1 0 0 0 0	= Q
81	0 1 0 1 0 0 0 1	= R
82	0 1 0 1 0 0 1 0	= S
83	0 1 0 1 0 0 1 1	= T
84	0 1 0 1 0 1 0 0	= U
85	0 1 0 1 0 1 0 1	= V
86	0 1 0 1 0 1 1 0	= W
87	0 1 0 1 0 1 1 1	= X
88	0 1 0 1 1 0 0 0	= Y
89	0 1 0 1 1 0 0 1	= Z
90	0 1 0 1 1 0 1 0	= [
91	0 1 0 1 1 0 1 1	= \
92	0 1 0 1 1 1 0 0	=]
93	0 1 0 1 1 1 0 1	= ^
94	0 1 0 1 1 1 1 0	= _
95	0 1 0 1 1 1 1 1	= `
96	0 1 1 0 0 0 0 0	= a
97	0 1 1 0 0 0 0 1	= b
98	0 1 1 0 0 0 1 0	= c
99	0 1 1 0 0 0 1 1	= d
100	0 1 1 0 0 1 0 0	= e
101	0 1 1 0 0 1 0 1	= f
102	0 1 1 0 0 1 1 0	= g
103	0 1 1 0 0 1 1 1	= h
104	0 1 1 0 1 0 0 0	= i
105	0 1 1 0 1 0 0 1	= j
106	0 1 1 0 1 0 1 0	= k
107	0 1 1 0 1 0 1 1	= l
108	0 1 1 0 1 1 0 0	= m
109	0 1 1 0 1 1 0 1	= n
110	0 1 1 0 1 1 1 0	= o
111	0 1 1 0 1 1 1 1	= p
112	0 1 1 1 0 0 0 0	= q
113	0 1 1 1 0 0 0 1	= r
114	0 1 1 1 0 0 1 0	= s
115	0 1 1 1 0 0 1 1	= t
116	0 1 1 1 0 1 0 0	= u
117	0 1 1 1 0 1 0 1	= v
118	0 1 1 1 0 1 1 0	= w
119	0 1 1 1 0 1 1 1	= x
120	0 1 1 1 1 0 0 0	= y
121	0 1 1 1 1 0 0 1	= z
122	0 1 1 1 1 0 1 0	= [
123	0 1 1 1 1 0 1 1	= \
124	0 1 1 1 1 1 0 0	=]
125	0 1 1 1 1 1 0 1	= ^
126	0 1 1 1 1 1 1 0	= _

ASCII

Eis aqui uma lista completa dos valores ASCII entre 32 e 126, seus equivalentes binários e os caracteres que representam. O significado dos valores fora destes limites varia de uma máquina para outra.

Um modo mais conciso de atingir o mesmo objetivo em BASIC é usar a instrução ON-GOTO, da seguinte forma:

```
1050 ON D GOTO 2020,2040,2060,2080
      3000,3020,3040
```

A instrução ON-GOSUB opera do mesmo modo, com exceção de que o valor da variável determina para qual sub-rotina o desvio é dirigido. Aqui apresentamos uma pequena modificação do programa de lançamento de dados (ver p. 174), usando a instrução ON-GOSUB, que seleciona os gráficos apropriados para os dados escolhidos pela função RND:

```
390 REM SUB-ROTINA ESCOLHER
400 REM UTILIZANDO A INSTRUCAO ON-GOSUB
410 ON D GOSUB 530,600,670,740,810,880
470 RETURN
```

Embora a versão de BASIC de seu equipamento possa, provavelmente, muitas instruções e funções que ainda não apresentamos, a maioria delas serão extensões da linguagem BASIC “padrão”. Muitas estarão vinculadas a finalidades gráficas incorporadas ao hardware — instruções como PAINT, PAPER, INK, BEEP e CIRCLE. Estas instruções, de modo geral, são “específicas da máquina” e, por isso, não as incluímos em nosso curso, mas delas daremos detalhes em artigos futuros.

Antes de encerrar a parte fundamental de nosso curso de BASIC, há alguns outros pontos a considerar: um exame do conjunto de caracteres ASCII, juntamente com as funções para auxiliar a manipulação de caracteres, e um método de definição de novas funções (ou funções não incluídas na versão de BASIC de seu computador).

Elaboraram-se vários métodos de representação das letras do alfabeto e de outros caracteres, como números e sinais de pontuação em forma digital, no decorrer dos anos. Um dos primeiros foi o código Morse, que utiliza combinações de pontos e traços para representar caracteres. Para o computador, o código Morse tem a desvantagem de utilizar diferentes quantidades de bits para letras diferentes — entre um e seis pontos e traços para cada caractere. Outras tentativas de formar um código de caracteres mais regular e sistemático (por exemplo, o código Baudot, que utiliza 5 bits para representar até 32 caracteres) foram superadas, e o sistema quase universal agora em uso é o código ASCII (American Standard Code for Information Interchange).

O código ASCII utiliza 1 byte para representar os 94 caracteres imprimíveis, o “espaço” e uma série de “caracteres” de controle. 8 bits podem fornecer 256 combinações diferentes (2⁸), mas isto é muito mais do que o necessário para representar os caracteres de uma máquina de escrever comum ou de um teclado de computador; desse modo, apenas 7 desses bits são usados, permitindo 128 combinações diferentes. (O oitavo bit, se utilizado, especifica um conjunto alternativo de caracteres gráficos ou de línguas estrangeiras.) No quadro da página anterior apresentamos os códigos ASCII, em decimal e binário, para o conjunto standard de caracteres.

Como você pode verificar no quadro, o código ASCII para a letra A é 65 e, para a letra B, 66. Os códigos para as minúsculas a e b são 97 e 98. Cada

letra minúscula tem um valor, em código ASCII, 32 números acima de seu correspondente em maiúscula. Essa diferença constante torna fácil a conversão de minúsculas para maiúsculas e vice-versa. Para realizar esta conversão, necessitaremos de outras duas funções não utilizadas até agora no curso de programação em linguagem BASIC — as funções ASC e CHR\$.

A função ASC toma o caractere imprimível e o transforma em código ASCII; desse modo, a instrução PRINT ASC("A") imprime o número 65 na tela; a instrução PRINT ASC("b") imprime o número 98.

A função CHR\$ faz o inverso: toma um número, supondo-o como código ASCII e o transforma no caractere que ele representa. Desse modo, a instrução PRINT CHR\$(65) imprime a letra A, enquanto a instrução PRINT CHR\$(98) imprime a letra b. As funções CHR\$ e ASC são amplamente empregadas com as instruções LEFT\$, RIGHT\$ e MID\$ em programas que fazem uso freqüente de séries de caracteres. Eis aqui um programa que recebe um caractere do teclado, verifica se está em maiúscula e, se não estiver, converte-o em maiúscula:

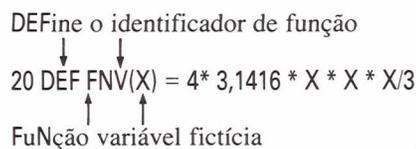
```
10 REM CONVERSION DE MINUSCULAS
   PARA MAIUSCULAS
20 PRINT "FORNECER UM CARACTERE"
30 INPUT C$
40 LET C = ASC(C$)
50 IF C > 90 THEN LET C = C-32
60 PRINT CHR$(C)
```

Veremos outros exemplos deste tipo de manipulação de séries em próximas explanações deste curso.

Examinemos agora funções que talvez não existam na versão BASIC de seu microcomputador. Quase todas as versões desta linguagem permitem ao programador criar novas funções e isto é quase tão fácil como utilizar funções incorporadas. A instrução DEF indica, em linguagem BASIC, que uma nova função está sendo definida. Eis aqui como definir uma função para cálculo do volume de uma esfera (a fórmula é $V = \frac{4}{3}\pi r^3$, onde r é o raio da esfera e π (pi) a constante aproximadamente igual a 3,1416):

```
10 REM FUNCAO PARA CALCULO DO VOLUME DE
   UMA ESFERA
20 DEF FNV(X) = 4 * 3,1416 * X * X * X/3
30 PRINT "FORNECER O RAO DA ESFERA"
40 INPUT R
50 PRINT "O VOLUME DA ESFERA DE RAO ";
   R;" E "
60 PRINT FNV(R)
70 END
```

Este modo de definir uma função é bastante simples, mas vejamos a linha detalhadamente:



Quando a função é definida, as letras FN são seguidas de uma letra identificadora — V, no caso da função acima —, que deve ser seguida de uma “variável fictícia”. Esta deve também ser usada na defini-



ção da função à direita do sinal de igualdade. Utilizada a função em um programa, qualquer variável numérica pode substituir a variável fictícia da definição.

Em um ponto posterior do programa acima, seria igualmente possível empregar a função "volume de uma esfera" da seguinte forma:

```
999 LET A = 66
1000 LET B = FNV(A)
1010 PRINT B
1020 LET C = 5
1030 LET D = B + FNV(C)
1040 PRINT D
1050 LET G = FNV(16)
1060 PRINT G
```

Algumas versões de BASIC permitem a utilização de múltiplas variáveis na função definida. Desse modo, uma função para calcular a média aritmética de dois números poderia ser escrita assim:

```
100 DEF FNA(B,C) = (B+C)/2
110 INPUT "FORNECER DOIS NUMEROS";B,C
120 LET A = FNA(B,C): REM A FUNCAO "MEDIA"
130 PRINT "A MEDIA DE "; B; " E "; C; " E "; A
```

Observe que a linha 110, acima, combina o equivalente a duas instruções separadas em uma única. A maioria das versões de BASIC imprime automática-

mente as palavras que aparecem entre aspas após a instrução INPUT; assim, esta linha equivale a:

```
110 PRINT "FORNECER DOIS NUMEROS"
115 INPUT A,B
```

A linha 120 também consegue o equivalente a duas instruções em uma linha, usando como separador dois pontos (:). Instruções que pertenceriam a linhas separadas podem ser escritas em uma única, desde que cada instrução "independente" seja separada da anterior pelo sinal de dois pontos. Isto economiza espaço em programas extensos, mas seu uso deve ser limitado, pois torna mais difícil a leitura e aumenta a probabilidade de erros.

Respostas aos exercícios da p. 197

Erros

Resultará em uma mensagem "OUT OF DATA ERROR", pois deveria haver doze valores na instrução DATA. Em segundo lugar, surgirá um erro na linha 100, quando o programa tentar endereçar um elemento A(4,1). A linha 100 deveria ser:

```
100 PRINT A(X, Y)
```

Atribuição de valores

Abaixo, um programa que executará o solicitado. Seu programa poderá estar desenvolvido de modo diferente.

```
10 DIM A(8,13)
20 FOR R=1 TO 7
30 FOR C=1 TO 12
40 READ A(R,C)
50 NEXT C
60 NEXT R
70 REM SOMAR TOTAIS
80 GOSUB 300
90 REM IMPRIMIR DADOS SOLICITADOS
100 GOSUB 200
110 PRINT "OUTROS DADOS?"
120 PRINT "S OU N"
130 INPUT A$
140 IF A$ = "N" THEN GOTO 160
150 GOTO 100
160 END
200 PRINT "QUAL MES?"
210 PRINT "1-PARA JANEIRO,"
220 PRINT "13-PARA TOTAL, ETC"
230 INPUT M
240 PRINT "QUAL DESPESA?"
250 PRINT "1-PARA GASOLINA"
260 PRINT "8-PARA TOTAL, ETC"
270 INPUT X
280 PRINT "O VALOR E ";A(X,M)
290 RETURN
300 FOR R=1 TO 7
310 LET T=0
320 FOR C=1 TO 12
330 LET T=T+A(R,C)
340 NEXT C
350 LET A(R,13)=T
360 NEXT R
370 FOR C=1 TO 13
380 LET T=0
390 FOR R=1 TO 7
400 LET T=T+A(R,C)
410 NEXT R
420 LET A(8,C)=T
430 NEXT C
440 RETURN
500 REM AQUI SEGUEM SEUS DADOS
510 REM OITENTA E QUATRO VALORES
520 REM "DADOS 45000, 50000" ETC
```

A propósito...



Os computadores tipo TK83, CP 200 e compatíveis possuem uma versão não standard do código ASCII, que não permite caracteres em minúsculas; assim, o programa de conversão de minúsculas em maiúsculas não executará essa operação; entretanto, experimente-o e procure em seu manual mais informações sobre o conjunto de caracteres destes equipamentos.



Nos micros VIC-20, Commodore 64 e Apple você não pode escrever mais de uma variável dentro do parênteses.



Nos micros tipo Sinclair (TK83, TK85, CP 200, Ringo), substituir:

```
ASC(A$) por CODE A$
```

e substituir:

```
CHR$(65) por CHR$ 65
```



Se o argumento for uma expressão, deverá ficar entre parênteses. Mas argumento simples — como A\$ e 65 — não precisam de parênteses.



Estas instruções não existem nos micros compatíveis com Sinclair (TKs, CP 200, Ringo).



Em alguns computadores, não há ";" após a mensagem. Neste caso, veja o manual de seu micro, e troque por ";" ou mesmo elimine o ";".



Esta instrução não existe em micros do tipo Sinclair (TKs, CP 200, Ringo) e compatíveis TRS-80 (CP 500, Digitus etc.). No entanto, pode-se contornar o problema com instruções que atribuam a uma função os valores a serem calculados.



Em alguns computadores convém alterar IF INKEY\$ = "" THEN LET X=2 por IF INKEY\$ = "" THEN LET X=1. Nos micros tipo VIC-20 e Commodore 64, por exemplo, deve-se alterar IF INKEY\$ = "" THEN LET X=2 por GET A\$: IF A\$ = "" THEN LET X=1.

Diálogo a distância

Um acoplador acústico converte dados digitais em tons audíveis; ligado a um telefone, estabelece comunicação entre dois computadores a distância.

A ligação de seu microcomputador a uma impressora ou a um grupo de unidades de disco é relativamente fácil, já que esses aparelhos em geral estão localizados na mesma sala, talvez na mesma mesa. No entanto, ligar o micro ao computador de um escritório, em algum lugar do mundo, é algo bastante diverso. Felizmente, já temos um meio de comunicação global — o sistema telefônico. Tudo de que precisamos é encontrar um meio de conectar nosso micro ao telefone.

Pelo fato de a rede telefônica ser tão usada pelos sistemas de computadores (para reserva de passagens aéreas, por exemplo), a tecnologia já está bem definida nesse sentido. Assim, o usuário do micro precisa apenas de uma versão simples e barata desse sistema. O meio convencional de efetuar essa conexão é através de um aparelho chamado modem. Este nome estranho resultou da junção das palavras modulador e demodulador.

O aparelho funciona quase como a interface de conexão cassete/micro. As configurações dos binários 1 e 0 são convertidas em sinais elétricos em duas frequências audíveis diferentes e depois enviadas para a linha telefônica (este é o processo de modulação). Na outra extremidade, elas são demoduladas, retornando das frequências audíveis para 1 e 0. O modem emite uma frequência constante (chamada tom transportador), indicando se os dados estão sendo enviados ou não; é assim que o computador receptor reconhece se a linha ainda está ligada.

A maior desvantagem do modem é estar permanentemente ligado por fios à rede telefônica, monopolizando seu uso, o que representa uma inconveniência para o usuário doméstico.

O acoplador acústico representa um método alternativo de comunicação. Como o sistema utiliza sons audíveis, nada impede que eles sejam gerados acusticamente através de um alto-falante. Este poderia então ser acoplado ao fone, possibilitando a transmissão. Na outra extremidade, um microfone colocado em contato com o receptor captaria os sinais transmitidos. É para isso que foi projetado um acoplador acústico. Ao contrário do modem, ele não precisa estar permanentemente ligado ao telefone.

Há vários tipos de acopladores acústicos no mercado, que variam dos aparelhos mostrados nas ilustrações (compactos o suficiente para serem usados em um microcomputador portátil), até as unidades próprias para ocupar lugar fixo em mesa. Unidades sofisticadas podem ser utilizadas para responder a chamadas automaticamente, sem necessidade da presença de um operador da máquina.

Assim como as interfaces de cassete podem variar as velocidades com que armazenam e recuperam a

informação, os acopladores acústicos possuem esta mesma característica. Entretanto, sua gama de velocidades é bastante limitada. As características de transmissão de um cabo telefônico não permitem que um sinal mais rápido que 1.200 caracteres por segundo (cps) seja transmitido com relativa segurança.

Unidades de menor custo podem trabalhar, a 30 cps, enquanto modelos mais caros contêm até mesmo um dispositivo de ajustamento para várias velocidades. É importante lembrar que os aparelhos que estão em ambas as extremidades da linha telefônica devem operar na mesma velocidade, caso contrário a transmissão não ocorre.

O enorme crescimento do uso dos microcomputadores em negócios tem resultado no desenvolvimento de uma grande variedade de produtos novos. Aparelhos como o Sendata e seus similares mais próximos permitem que computadores portáteis, como o Model 100, da Tandy, e o HX-20, da Epson, sejam usados como terminais remotos de computadores por profissionais tão diversos como jornalistas e representantes de vendas. Qualquer matéria escrita pode ser registrada na memória do computador e a seguir enviada para os escritórios centrais, através das linhas telefônicas.

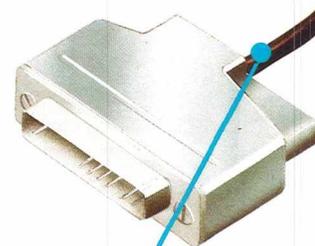


Microfone

Capta o sinal do alto-falante do telefone e o envia à placa do circuito.

Soquete de energia

Fornece energia ao acoplador através de um transformador adequado. É utilizado também para recarregar as baterias internas de níquel-cádmio.

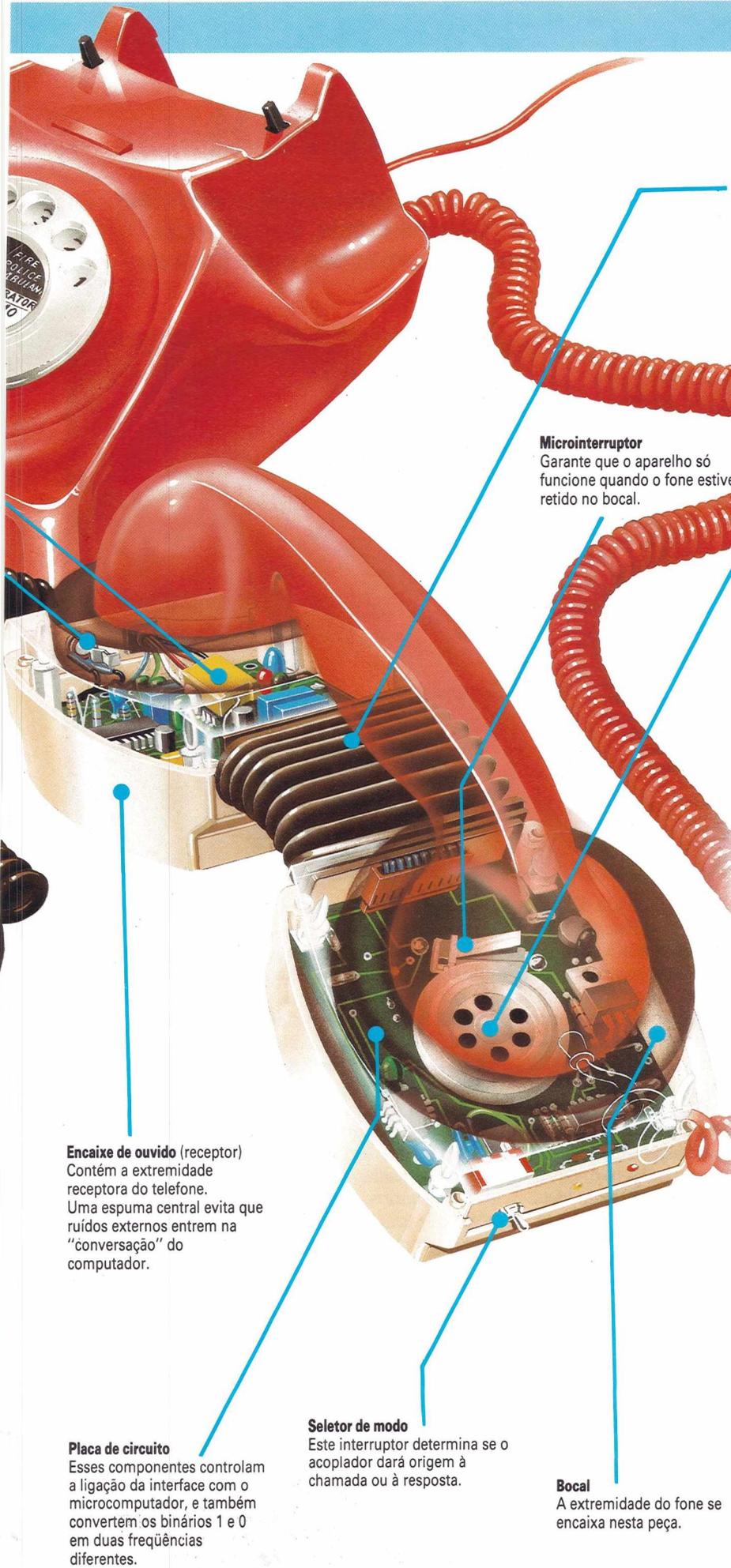


Cabo da interface

Faz a conexão ao soquete da interface do computador RS232 (em série).

Cabines telefônicas

Acopladores acústicos leves permitem que o usuário de um computador portátil se comunique com outro computador em qualquer lugar do mundo, através da rede de telefones públicos.



Ligação flexível
Permite que o acoplador se ajuste à maioria dos telefones existentes.

Microinterruptor
Garante que o aparelho só funcione quando o fone estiver retido no bocal.

Alto-falante (emissor)
O acoplador acústico é realmente o oposto do fone. Este dispositivo emite dados em tons audíveis.

No comércio, um terminal de computador e um acoplador acústico permitem o acesso instantâneo a uma grande variedade de serviços de informação e centros de computação. Atualmente, o pedido de estoque numa cadeia de lojas pode ser feito totalmente por computador; os funcionários registram os itens e as quantidades, e a seguir transmitem essas informações ao sistema de computador principal, encarregado da estocagem de mercadorias. Em casa, o acoplador acústico leva grande vantagem sobre o modem convencional por não precisar estar permanentemente ligado à linha telefônica. Trabalhando em casa com um computador pessoal, um executivo mantém contato com seu escritório, enviando ou recebendo informações, sem que para isso tenha de bloquear constantemente a linha telefônica.

No mercado do microcomputador, o acoplador acústico é uma alternativa conveniente e de baixo custo, substituindo o modem convencional, pois permite o acesso a bancos públicos de dados. Além disso, eles representam um modo de enviar programas a amigos, pois é mais seguro que as fitas cassete remetidas pelo correio. Serviços de correspondência eletrônica oferecem ao usuário do microcomputador acopladores acústicos rápidos e eficientes, aos quais anteriormente apenas as grandes companhias tinham acesso.

Existem problemas de compatibilidade para comunicação internacional: ainda não é possível para um acoplador acústico europeu entrar em contato com um acoplador americano. A indústria americana de computadores utiliza um sistema chamado Bell 103, e a européia usa um sistema chamado CCITT V21. Desnecessário dizer que esses dois sistemas são incompatíveis.

Uma outra complicação é que os telefones produzidos hoje não parecem adequar-se aos muitos acopladores acústicos (ver ilustração). Como o sistema funciona pela transmissão de sons, é importante que se evite a possibilidade de ruídos externos entrarem no fone. Caso o telefone não se adapte, ou haja um excesso de ruído externo, os dados enviados por você podem ficar confusos, e de difícil interpretação.

Encaixe de ouvido (receptor)
Contém a extremidade receptora do telefone. Uma espuma central evita que ruídos externos entrem na "conversação" do computador.

Placa de circuito
Esses componentes controlam a ligação da interface com o microcomputador, e também convertem os binários 1 e 0 em duas frequências diferentes.

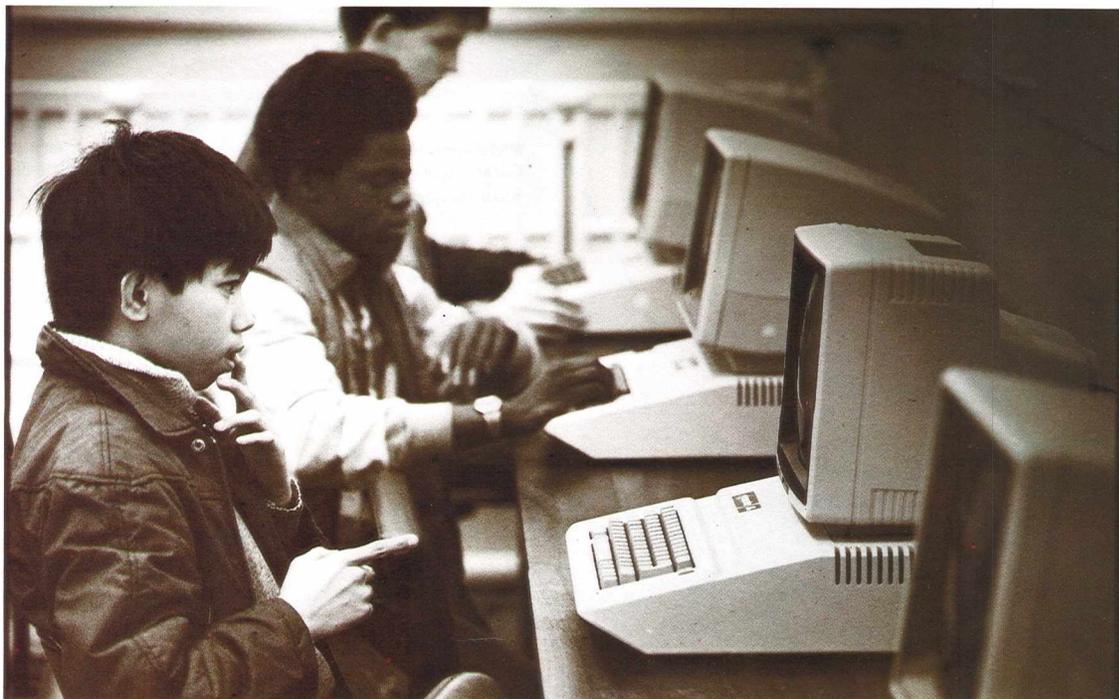
Seletor de modo
Este interruptor determina se o acoplador dará origem à chamada ou à resposta.

Bocal
A extremidade do fone se encaixa nesta peça.



Informação dividida

Instalados num mesmo edifício, microcomputadores podem ser interligados em sistemas chamados redes locais, permitindo que diversos usuários compartilhem informações e periféricos.



Apple para professores

Nos centros de tecnologia da informação, onde jovens aprendem a lidar com computadores, as redes de micros são usadas para levar o mesmo treinamento aos alunos.

Os microcomputadores tornam-se a cada dia mais comuns em escolas, escritórios e residências. É provável que, com o tempo, um mesmo edifício abrigue certa quantidade de máquinas compatíveis. Se isso acontecer, os micros poderão ser conectados entre si e, dessa maneira, tornar comunitário o uso de periféricos dispendiosos, como unidades de disco ou impressoras de boa qualidade.

Mas a ligação dos periféricos não é, de modo algum, o único proveito que se pode tirar da interconexão de vários micros. Eles podem estabelecer, também, uma rede de intercomunicação. Quando esse circuito só inclui máquinas localizadas num mesmo prédio, diz-se que se formou uma “rede local” (LAN — Local Area Network).

Já se falou muito sobre o fim do uso do papel nos escritórios, como se isso fosse uma possibilidade remota. Com os recursos atuais, porém, pode-se substituir um memorando escrito em local distante por uma mensagem que aparece na tela do monitor. Uma “caixa de correio” desse tipo pode fazer parte da maioria das “redes locais”. Para não perturbar o trabalho do destinatário, a chegada da mensagem seria anunciada por um aviso escrito na parte inferior da tela.

Em salas de aula, uma rede local é de grande utilidade. Os textos para estudo seriam “espelhados” do micro do professor para a central de trabalho dos estudantes. Ou então o professor poderia empregar seu equipamento para acompanhar as tarefas de seus

alunos, auxiliando-os com comentários e sugestões, e corrigindo seus eventuais erros.

Essas mesmas facilidades possibilitam também a utilização de um centro de informação compartilhado por diversos usuários. Talvez a aplicação mais freqüente de um sistema desse tipo sejam os bancos de dados, que contêm informações públicas ou privadas — é o que ocorre com sistemas do tipo Videotexto e outros.

Quando muitas pessoas utilizam o mesmo “arquivo” de informações, é essencial tomar precauções para que as informações armazenadas não sejam modificadas sem o conhecimento prévio de todos os usuários. Uma empresa industrial, por exemplo, pode usar a rede de computadores para fornecer a seus gerentes informações sobre a disponibilidade de componentes e matéria-prima. Cada usuário do sistema deve receber os mesmos dados, sempre atualizados, ou pode acontecer que estoques esgotados apareçam como existentes para um ou mais departamentos da empresa.

Utilizando-se microcomputadores com capacidade de 64 Kbytes de RAM ou superior, a tendência é carregar apenas uma parte do banco de dados para trabalho, e não mais recorrer ao arquivo “mãe”, até que alguém precise de outro bloco de informações. A menos que o software de controle seja suficientemente abrangente para detectar e passar as mudanças de informação ao usuário do banco de dados, corre-se o risco de consultar dados desatualizados.



De modo geral, o custo do hardware de computadores está diminuindo. Mas os periféricos mais aperfeiçoados podem ser tão caros quanto os próprios microcomputadores. Um bom exemplo disso é o Winchester, um disco rígido, hermeticamente fechado, que funciona com velocidade e capacidade de memória iguais às das unidades de disco principais. Os discos Winchester (assim chamados por causa do fuzil de repetição Winchester 30/30), originalmente constituídos por dois discos iguais, cada um com capacidade de 30 megabytes, são indispensáveis para assegurar o desempenho de software comercial, quando o volume de transações torna-se muito grande. Sua capacidade de armazenamento é suficiente para diversos usuários ao mesmo tempo.

De modo similar, uma impressora matricial de pontos (ver p.74) é adequada para muitas finalidades, mas não será boa para editar textos — para isso você terá necessidade de uma impressora tipo margarida, que apresenta melhor qualidade gráfica, e preço bem mais elevado. Poucos usuários precisam manter um acessório desses em funcionamento constante; assim, será interessante compartilhá-la com outras centrais de operação.

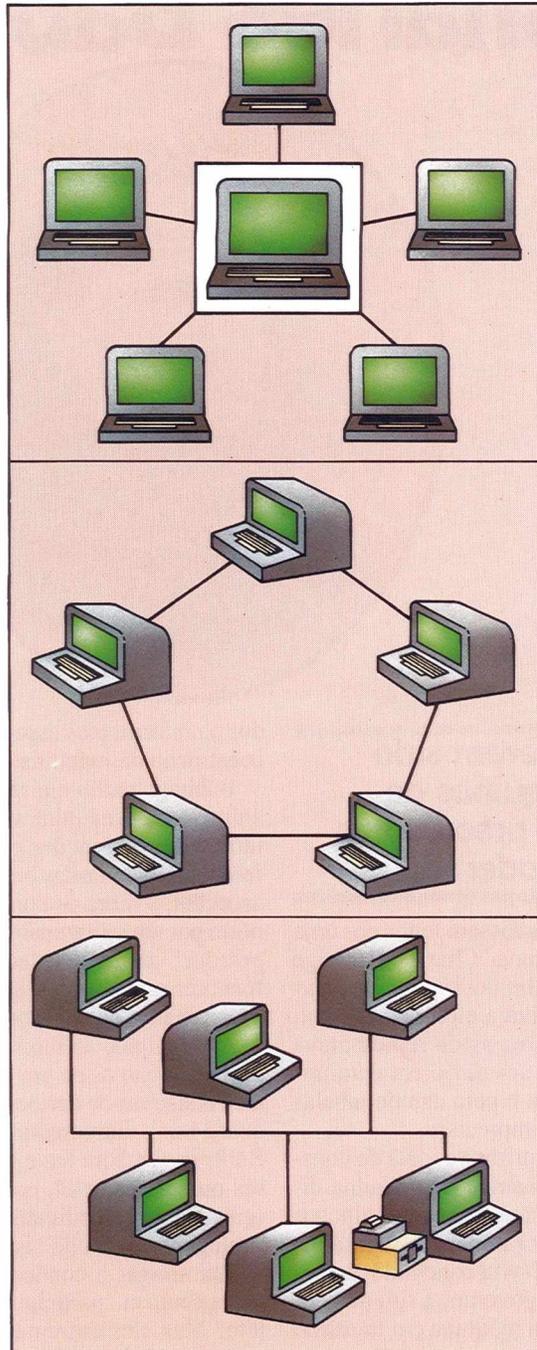
As redes de micros oferecem ainda outras possibilidades. Documentos que devem ser examinados por várias pessoas são transmitidos de uma para outra, mas não escritos em papel. É o que ocorre, por exemplo, em redações de grandes jornais e revistas: um redator envia seu texto pelo micro ao editor, para que faça os acertos necessários. Daí, pela mesma via, o texto segue para a preparação, onde sofre eventuais correções ortográficas e gramaticais, e recebe as especificações para composição e impressão. Sem a rede local, o texto precisaria ser redatilografado toda vez que sofresse modificações, atrasando e encarecendo todo o processo.

Em funcionamento

A maioria dos microcomputadores pode ser posta em sistema de rede. Um dos sistemas mais usados para isso na Europa é o Econet da Acorn. Esse equipamento designa uma das máquinas da rede para agir como “fornecedora de arquivo”, cuidando da unidade de disco central e atendendo aos pedidos de informação. A máquina pode ser usada para esse fim exclusivo ou ficar disponível para outras tarefas, sempre que a rede não estiver em ação. Quando a rede dispõe de impressora, esta precisa do controle constante e exclusivo de um micro.

O Econet tem condição de comandar até 254 centrais de operação e ainda suas estações “fornecedoras”. No entanto, a distância entre as estações e a unidade central de processamento das redes locais não pode ser superior a 500 metros; tal limitação impede o Econet de funcionar com capacidade plena. A central de processamento abriga um relógio especial, que controla o fluxo de entrada e saída de dados do sistema. O Econet usa dois pares de cabos, exatamente como um sistema telefônico. Um dos pares carrega os dados, o outro conduz os impulsos do relógio necessários para a sincronização.

O sofisticado software de comunicações usado pelo Econet consiste num EPROM (Erasable Programmable Read-Only Memory) de 8 Kbytes em



Tipos de rede

Estrela

A rede tipo estrela liga o operador de cada máquina ao controle central, que também comanda os periféricos de uso comum.

Anel

Nesse tipo de rede os computadores devem dispor-se em seqüência contínua. O sistema não é muito usado porque os dados só chegam ao destino depois de passar por mais da metade dos microcomputadores da rede.

Bus

Essa disposição, do mais eficiente sistema em rede, como o Econet, é similar à estrutura de um microcomputador moderno: os dados e as mensagens de controle são passados diretamente de um usuário para outro.

cada central de operação. A mais complexa função do sistema é prevenir a colisão de transmissões, ou seja, garantir que apenas um membro da rede por vez faça transmissões. Outros sistemas em rede baseados em microcomputadores funcionam de modo semelhante, embora não ofereçam tantas vantagens.

As redes de microcomputadores tornaram-se bastante difundidas nas últimas décadas, sendo utilizadas por todos os tipos de empresa, em todo o mundo. Muitas companhias aéreas, por exemplo, usam sistemas desse tipo para fazer reservas de passagens locais ou internacionais, com o auxílio de ligações telefônicas ou de satélites. É possível que, com a televisão a cabo, o emprego de redes aumente ainda mais e provavelmente se utilizem sistemas similares aos do Videotexto permitindo a transmissão de programas por via telefônica.

Charles Babbage



Apesar de nunca terem sido concluídas, as máquinas de Babbage foram as precursoras do computador moderno.

“Quisera Deus esses cálculos fossem feitos por uma máquina a vapor!”, exclamava Charles Babbage enquanto trabalhava nas tabelas do *Almanaque náutico*. O século XIX desenvolvera a propulsão a vapor, mas a navegação marítima ainda representava um problema. A posição de um navio era determinada pela observação da Lua e pelo uso de tabelas matemáticas, quase sempre imprecisas.

Em 1812, Babbage teve a primeira idéia de construir uma máquina, à qual chamou de máquina diferencial. Ela seria capaz de realizar os trabalhosos cálculos necessários à confecção das tabelas náuticas. Por volta de 1823, ele já havia concluído um pequeno modelo e recorreu ao governo, a fim de conseguir verbas para construir a máquina em tamanho funcional. O ministro da Fazenda da Grã-Bretanha concedeu-lhe 1.500 libras esterlinas, e ele empenhou-se na construção de um aparelho que eliminaria os erros mediante a impressão automática dos resultados de seus próprios cálculos.

A linha de trabalho de toda a vida de Babbage estava, a partir de então, determinada. O projeto consumiu altas somas em dinheiro, pois ele estava trabalhando com o que havia de mais avançado na engenharia da época. Para obter o dinheiro, ele contou com a ajuda do primeiro-ministro, seu amigo, o duque de Wellington.

Apesar da segurança de Babbage (“O que quer que a máquina faça, estará correto”, dizia ele), o governo acabou desistindo da idéia, depois de ter investido 17.000 libras no empreendimento. O engenheiro contratado por Babbage, Joseph Clement, demitiu-se algum tempo depois, e levou consigo to-

dos os instrumentos especialmente elaborados para a construção da máquina.

Babbage partiu em seguida para um projeto mais ambicioso, a máquina analítica. Esse novo engenho teria todas as funções previstas para a máquina diferencial — e muitas outras. Seu projeto, em vários aspectos, lembra os computadores atuais. Era composto por um armazenamento de memória, um “manipulador” aritmético (equivalente à CPU), e podia fornecer saídas impressas. A máquina podia até ser programada, usando ramificações condicionais.

A princípio, as instruções eram controladas por cravos, como os de um realejo; mais tarde, foi adotado o sistema de cartões perfurados que Joseph Jacquard havia introduzido na indústria de tecelagem. Babbage também fez experiências com diversas bases numéricas, mas, como todas as suas máquinas utilizavam procedimentos mecânicos, não havia vantagem no uso do sistema binário.

Sua amiga, a condessa Ada Lovelace, matemática talentosa, participou da elaboração desse projeto. Mas eles enfrentaram muitos problemas. Ela perdeu sua fortuna jogando num “infalível” sistema de apostas em corridas de cavalo. Depois de sua morte, aos 36 anos, Babbage prosseguiu sozinho.

Homem de energia prodigiosa, ele também inventou o oftalmoscópio médico, coreografou um balé, criou um sistema de iluminação de palco e inventou uma técnica de sinalização marítima.

Nos últimos anos de vida, ele tornou-se irritável ao extremo. Embora desejasse a nobreza, recusou o título de barão que lhe foi concedido em reconhecimento pelo seu trabalho.

As experiências de Babbage anteciparam a estrutura do moderno computador eletrônico, mas ele não teve condições de desenvolver seus projetos inteiramente. A máquina analítica ficou para sempre inacabada, pois as limitações técnicas do século XIX impediram sua conclusão.

1792

Nasce em Totnes, Devon, em 26 de dezembro.

1810

Ingressa no Trinity College, em Cambridge, para estudar matemática.

1814

Casa-se com Georgina Whitmore.

1822

Publica um artigo intitulado “Observações sobre os usos de máquinas para o cálculo de tabelas matemáticas”. Recebe a primeira medalha de ouro da Associação Astronômica, que ele ajudou a fundar.

1827

Cambridge indica-o como professor para a cátedra que pertencera a Newton, apesar de nunca ter lecionado lá.

1833

Candidato ao Parlamento em Finsbury.

1834

Projeto da máquina diferencial interrompido após a desistência de Joseph Clement.

1862

A máquina diferencial, parcialmente acabada, é exibida em Londres.

1871

Morre em 18 de outubro.



Inimigo eletrônico

Alguns dos mais potentes microcomputadores já criados não se encontram em escritórios ou fábricas de alta tecnologia, mas em fliperamas, bares e restaurantes.



Invasores alienígenas

“Você precisa atacá-lo, antes que atinja o ponto zero, se não ele vai se multiplicar... Atenção! A nave-mãe está lançando uma esquadrilha... é melhor você usar a bomba Smart...”

Não, não se trata de um diálogo de filme de ficção científica, é apenas uma conversa ouvida por acaso em um fliperama.

Em 1971, um jovem chamado Nolan Bushnell gastou tempo e esforço tentando convencer proprietários de bares e restaurantes nas cercanias de Sunnyvale, Califórnia, onde vivia, a experimentar um novo tipo de jogo por ele inventado. Funcionava por meio de moedas, o que significava dinheiro em caixa para o arrendatário, mas baseava-se em uma idéia totalmente nova.

Finalmente, o dono de um bar concordou em experimentar. Dois dias depois, telefonava para o inventor, reclamando que o jogo havia quebrado. Bushnell logo descobriu que o defeito era bem simples — o compartimento para moedas estava repleto, e a fenda para colocá-las, obstruída. Resolveu o problema instalando uma caixa maior para as moedas.

Pong — uma variante do pingue-pongue — foi o precursor de todos os emocionantes e originais jogos eletrônicos, que funcionam baseados em microcomputação e se encontram agora por todo o mundo. Talvez seja interessante notar que, apesar de ter sido

o primeiro a sugerir que jogos como pingue-pongue podem ser simulados por um computador, Bushnell ainda não tinha atingido o ponto essencial da questão: seu jogo requeria duas pessoas, para competir uma com a outra, e não permitia que um jogador sozinho experimentasse sua habilidade e destreza em confronto com a máquina. Longo tempo se passou até o surgimento da geração seguinte de jogos eletrônicos. Só em 1977 é que uma companhia japonesa, a Taito, entrou em cena com o jogo Space Invaders, que alcançou enorme sucesso.

Na terminologia moderna, Space Invaders é conhecido como um jogo no qual um “alienígena ataca usando escudos”. O jogador move sua base de bombardeio ao longo da parte inferior da tela, ora protegendo-se atrás de escudos, ao sentir-se ameaçado, ora atirando contra uma fila de alienígenas estilizados, que avançam continuamente e respondem ao fogo inimigo a intervalos irregulares. O ritmo do avanço alienígena é previsível e seguido de um ruído eletrônico bitonal adequado, que acompanha sua



lenta aceleração. Descrito assim, esse jogo realmente parece bem simples, mas o efeito conjunto é fascinante e irresistível.

Os primeiros jogos Space Invaders utilizavam uma tela de televisão monocromática e gráficos de sondagem exploratória. Apresentavam poucas inovações em hardware e software, mas, quando surgiram, ocasionaram uma revolução social como há muito não se via. O hardware que sustentava Space Invaders e todos os seus similares diferia muito pouco do utilizado pelos micros da época. Os fabricantes desses microcomputadores logo perceberam que havia grande mercado inexplorado entre as classes de maior poder aquisitivo. Esse mercado teve seu foco rapidamente transferido do ensino de programação para o uso do computador como um meio de entretenimento, ao que se seguiu a elaboração de numerosos projetos de máquinas. Ainda hoje são vendidos muitos equipamentos devido ao fato de executarem jogos, apesar de os mais recentes jogos eletrônicos ultrapassarem a capacidade de quase todos os mais avançados microcomputadores. Não é raro encontrar até 1 milhão de bytes de memória em um jogo eletrônico atual. Seus recursos são semelhantes

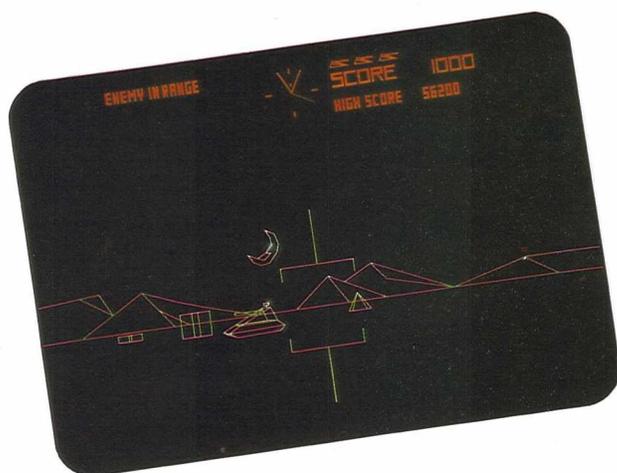
A Atari, cujo êxito está fundado no jogo original de Nolan Bushnell, é um exemplo particularmente adequado do intercâmbio entre jogos eletrônicos e microcomputadores. Essa empresa há muito tinha descoberto no micro um meio de lazer antes de tudo original (um reflexo, talvez, do maior interesse de sua proprietária, a Warner Bros) e, além de sua série de micros, também oferece um eficiente computador para jogos, o VCS (Video Cartridge System), que leva para casa, praticamente sem alteração, muitos dos jogos encontrados em equipamentos de fliperamas. No Brasil, a Atari, através da Gradiente/Polyvox leva grande vantagem, é claro, por se tratar de importante produtora de equipamentos eletrônicos. Entretanto, outras empresas dedicadas ao lazer e entretenimento, como Philips e Sharp, estão também envolvidas na mesma atividade. Adquiriram os direitos de produção de muitos jogos eletrônicos para o mercado doméstico, que podem ser utilizados em suas consoles Odissey e Intellivision, respectivamente.

Todo o mercado, seja apenas para jogos eletrônicos, ou software de jogos para microcomputadores, está rapidamente se tornando uma indústria autônoma. Em termos de estratégia de marketing, lembra a indústria fonográfica, apresentando também as listas dos vinte jogos mais vendidos no momento. Os dois fenômenos são similares em outros aspectos: os softwares de jogos são objeto de troca entre crianças em idade escolar, e as atividades de pirataria de software estão proliferando dia a dia (ver p.192).

Mencionamos antes que o grau de desenvolvimento do hardware encontrado nos jogos eletrônicos coloca-os numa classe diferente da dos micro-

Campos de batalha

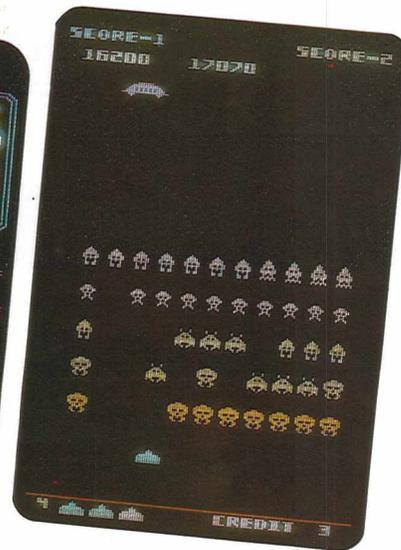
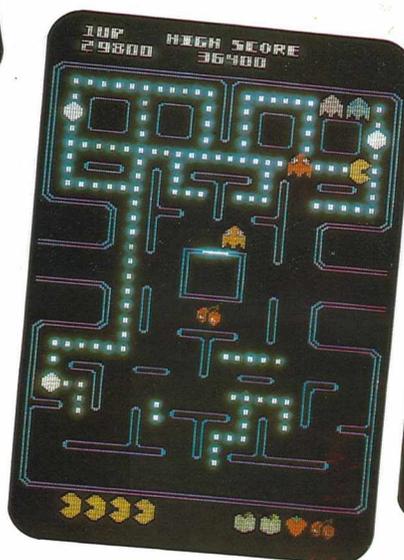
Jogos eletrônicos, como Space Invaders, em que o jogador pode mover seu sinal apenas em uma linha fixa, têm sido iguais em popularidade aos jogos de labirinto e perseguição, como PacMan. Ambos utilizam gráficos sprite e, por isso, têm qualidade visual inferior. Ultimamente, os projetistas de jogos têm produzido representações bem mais abstratas, como Battlezone, jogo de tanque e míssil contra um jogador do futuro, ou Tempest, que utiliza bela composição gráfica produzindo uma ilusão de expressiva profundidade.



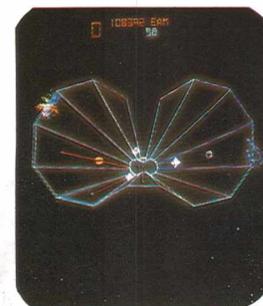
aos encontrados em terminais gráficos utilizados em computadores de grande porte.

Muitas das conquistas no campo dos microcomputadores podem remontar aos jogos eletrônicos. A mudança para processadores de 16 e até 32 bits, por exemplo, deu-se como resultado da necessidade de endereçar mais de 64 Kbytes e de obter velocidades maiores de processamento. Os fabricantes de jogos eletrônicos estavam à frente desse movimento e entre os primeiros clientes de microprocessadores de 16 bits, como o Motorola 68000 e o Intel 8086. Sua necessidade de processamento rápido e memória de maior capacidade foi anterior à dos usuários de microcomputadores de escritórios.

Derivaram daí vantagens para os usuários de microcomputadores, mesmo para os de menor poder aquisitivo. Os gráficos sprite, por exemplo, foram desenvolvidos para os jogos eletrônicos e, posteriormente, se tornaram acessíveis aos microcomputadores. Os eficientes chips para gráficos e produção de som, como o Video Interface Chip e o Sound Interface Device da Commodore, tiveram a mesma origem, assim como os três chips empregados pela Atari com a mesma finalidade em seus microcomputadores das séries 400 e 800.



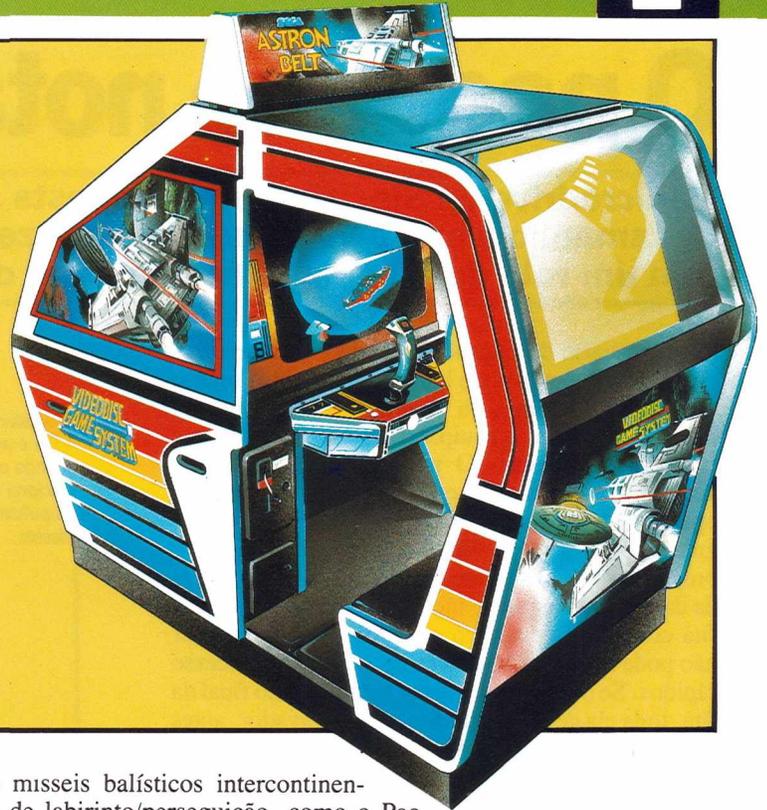
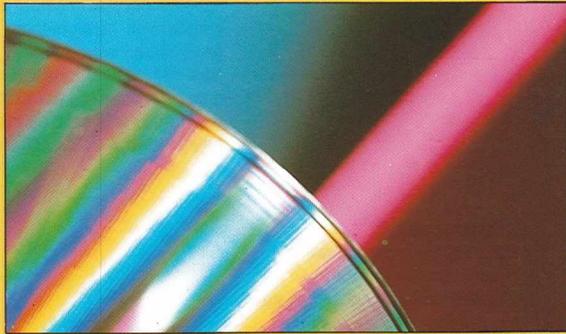
computadores. O mais recente destes desenvolvimentos requer o uso de videodiscos para proporcionar um cenário de fundo no monitor de televisão, contra o qual os jogos são realizados. Ainda não se sabe, porém, se esses refinamentos chegarão a alcançar o mercado doméstico. A tecnologia certamente está a preços acessíveis ao consumidor de produtos eletrônicos, embora o software de jogos produzido para consoles e microcomputadores não tenha realmente atingido o padrão dos jogos eletrônicos em sua forma original.





Cápsula espacial

Astron Belt é um jogo característico da nova geração de jogos eletrônicos, que utiliza discos laser (ver abaixo) para proporcionar um fundo que muda, no jogo em andamento. Por serem discos de acesso aleatório, é possível a passagem direta de uma cena a outra — de um confronto com uma nave alienígena, para uma explosão, por exemplo. Para o jogador, é um grande passo em direção ao realismo. As imagens no disco laser podem ser filmes da vida real ou desenhos animados feitos pelo computador. O Astron Belt também oferece som estéreo e um assento trepidante, que reage a baixas frequências no circuito sonoro.



Quando tratamos dos efeitos de simulação de voo (ver p. 201), observamos que todos os jogos eletrônicos são simulações de um tipo ou de outro, quer de uma situação da vida real, como o jogo de pingue-pongue ou uma corrida de carros, quer de uma fantasia, como Space Invaders, PacMan ou Frogger. Durante os anos que se seguiram à invenção dos jogos eletrônicos, estes dois grandes ramos têm se expandido em correntes paralelas de desenvolvimento, apesar de os jogos de fantasia/espaço serem muito mais comuns.

A partir desses dois jogos originais — Pong e Space Invaders — muitas alterações foram sendo introduzidas.

O primeiro jogo de bola e bastão a colocar o homem contra a máquina foi o Breakout e suas variantes, no qual o jogador atira uma bola contra um muro de tijolos. Cada tijolo desaparece da tela assim que é atingido, e o objetivo do jogo é removê-los todos, sem perder a bola. Como resultante deste jogo, temos simulações de golfe, snooker/bilhar e pebolim. De modo geral, quanto maior a atenção do programador em reproduzir as forças físicas (por exemplo, a gravidade, a resistência do ar, o atrito, o choque), melhor será o jogo.

Mas tais critérios não se aplicam aos jogos de fantasia. Aqui, o jogador está na verdade competindo com a pessoa que programou a máquina para o jogo em questão, inteiramente sob suas regras e no universo por ela criado. O primeiro estágio após Space Invaders foi introduzir diferentes espécies de alienígenas, atacando de formas diversas e a intervalos irregulares. Em seguida, foi necessário acrescentar movimento aos sinais executados pelo jogador na tela, o que deu origem a jogos como Defender, considerado pelos conhecedores como o melhor de todos os jogos deste tipo.

Jogos de linhas fixas, como o Space Invaders inicial, continuaram a se desenvolver, originando outros como Missile Command e seus derivados, nos quais o objetivo do jogador é defender sua base dos

ataques de mísseis balísticos intercontinentais. Jogos de labirinto/perseguição, como o Pac-Man, evoluíram dos primeiros jogos de corrida, em que o objetivo era dirigir um elemento em forma de carro em um trajeto, sempre que possível no tempo permitido, sem colidir com as muretas. Não havia nenhum elemento de antagonismo nesses jogos — mesmo considerando as manchas de óleo ocasionais que apareciam como por mágica — e, assim, o passo seguinte foi transformar o que era antes “uma competição contra o tempo” em uma perseguição. A pista inicial transformou-se então em um labirinto, e os sinais, por sua vez, em frutas, lâmpadas ou algo parecido.

Jogos de corridas de automóveis tornaram-se representações pseudotrídimensionais do trajeto, visto do carro, ou atrás deste, com a estrada sempre se modificando e surgindo à frente do jogador. Um método muito semelhante é utilizado nos jogos de simulação de voo mais realísticos.

Por fim, vêm os tradicionais jogos de tabuleiro, como dama, xadrez e gamão. Seu uso restringe-se aos microcomputadores porque, em geral, é preciso um tempo muito longo para jogá-los e a representação gráfica está em segundo plano no programa, em relação aos próprios algoritmos dos jogos.

A única diferença técnica real está no método de geração de gráficos. Os primeiros jogos e a maior parte dos atuais apresentam gráficos de sondagem exploratória, mas alguns, em especial Asteroids, usam métodos de sondagem vetorial, por meio dos quais somente as imagens na tela, não as áreas escuras, são sondadas pelo feixe de elétrons.

Assim, a próxima vez em que você passar por um fliperama, lotado de jogos complicados, ou apoiar-se em um deles num bar, lembre-se de que o computador que está no interior da máquina é, provavelmente, muito mais complexo e potente que aqueles usados em casa ou nos pequenos escritórios, utiliza muitas de suas técnicas e é programado por alguns dos mais talentosos profissionais do mundo de hoje.

O pequeno notável

O microdrive da Sinclair é uma resposta aos problemas de tamanho e custo vinculados à armazenagem de dados. Ele tem um rolo de fita magnética no lugar de um disco flexível.

Para o usuário do microcomputador, o cassete comum oferece um método barato e geralmente seguro de gravar programas ou carregar um software comercial. Todavia, o equipamento cassete apresenta várias desvantagens. O maior problema está na velocidade: mesmo os equipamentos cassete que operam em alta velocidade, a 1.200 bits por segundo, podem levar vários minutos para carregar programas longos ou localizar determinada parcela de dados. Outro grande inconveniente está em que a fita corre em apenas um sentido — o computador não pode operar os controles de avanço e retrocesso rápidos. Se o programa estiver localizado no final da fita, toda ela deverá ser bobinada no gravador, antes de iniciar o carregamento.

O sistema de disco resolve esses problemas, porém a um preço mais elevado. O que os usuários do microcomputador efetivamente necessitam é de um dispositivo que seja muitas vezes mais rápido

Bobinador da fita

A fita é puxada pelo cartucho, girando o bobinador, que funciona como o cilindro propulsor de gravadores cassete.

Cabeçote da fita

Um cabeçote reproduzidor e gravador miniaturizado, semelhante aos existentes nos gravadores cassete comuns.

Interface 1

Além de possibilitar as ligações necessárias para os microdrives, esta unidade proporciona entrada serial para conexão de impressoras e de uma interface para rede, permitindo a ligação de até 64 equipamentos ZX Spectrum.

Microinterruptor de proteção

Quando o cartucho estiver protegido, pela remoção da aba de proteção dos dados, este microinterruptor é ligado, evitando que o microdrive grave no cartucho.

Conector para expansão

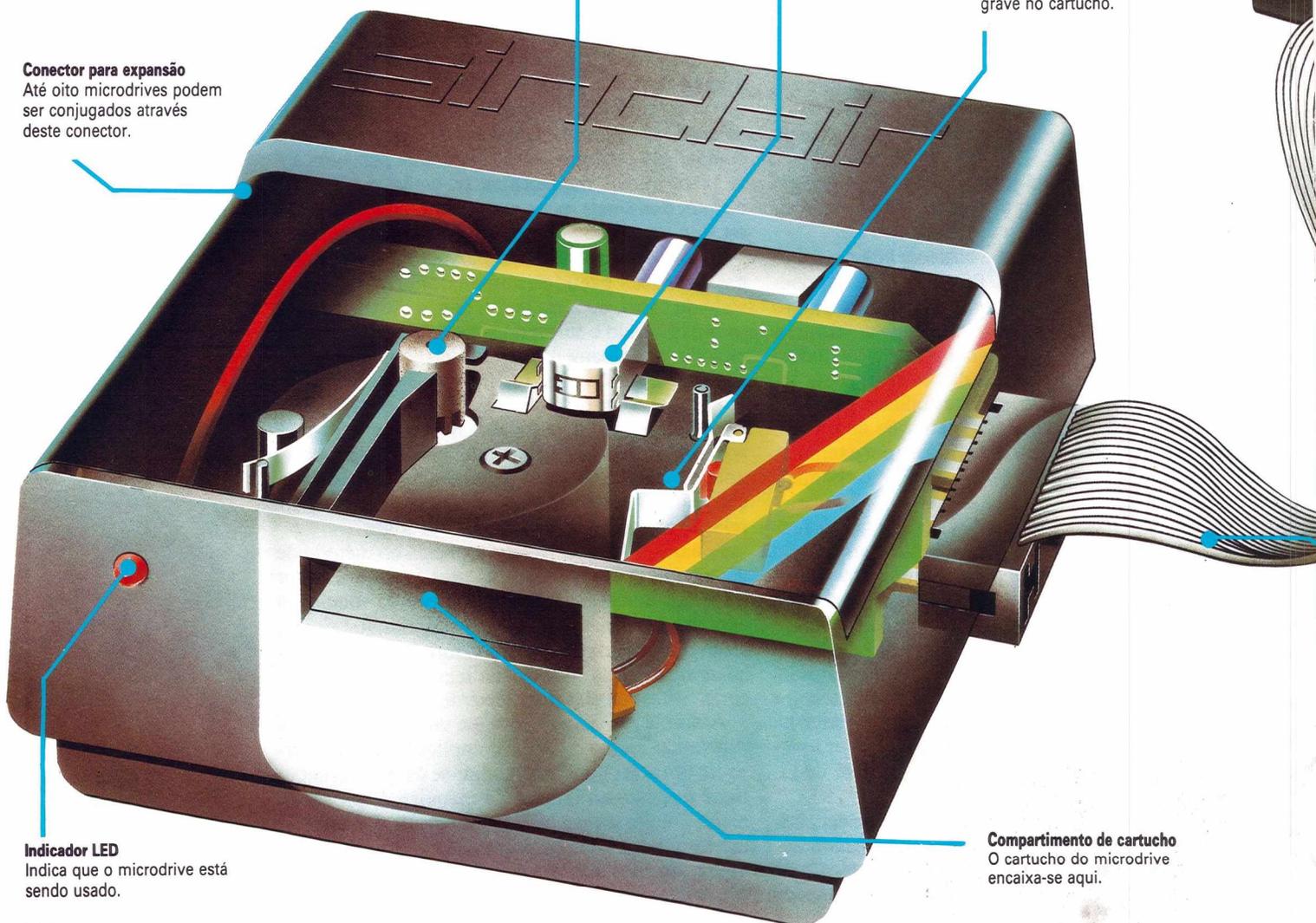
Até oito microdrives podem ser conjugados através deste conector.

Indicador LED

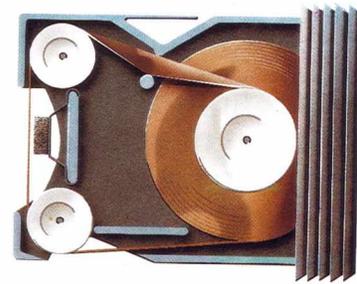
Indica que o microdrive está sendo usado.

Compartimento de cartucho

O cartucho do microdrive encaixa-se aqui.



que o cassete e mais barato que o disco. Tal dispositivo existe e chama-se "floppy tape" ou "stringy-floppy". Inicialmente desenvolvido nos Estados Unidos para o equipamento TRS-80 Model 1, da Tandy, pela Exactron, o primeiro stringy-floppy utilizava um rolo de fita acondicionado em um cartucho; a idéia baseava-se no sistema para fitas so-



O cartucho do microdrive

A fita do microdrive consiste em um rolo de fita contínua de vídeo-tape, que mede cerca de 2 mm de largura. A fita para vídeo é utilizada, em vez da cassete comum, devido à resistência e durabilidade. É mais estreita que a fita para gravação sonora, o que a torna mais sujeita a danos.

Para funcionar, o rolo de fita corre no interior do microdrive em aproximadamente 7 segundos e os dados são transferidos a cerca de 6 Kbytes por segundo, uma melhora significativa em relação a 1,5 Kbyte por segundo da interface cassete do Spectrum. O programa armazenado no microdrive é localizado e carregado em cerca de 15 segundos. A fita pode reter até 100 Kbytes de dados, mas a Sinclair garante a disponibilidade de somente 85 Kbytes para sua utilização. O equipamento deve ser preparado para uso através da "formatação" da fita, processo que é iniciado por um único comando BASIC. A formatação verifica as partes da fita em condição de uso, evitando os trechos com defeito.

dos stringy-floppies apresentavam problemas de baixa qualidade das fitas. Até o lançamento do microdrive, da Sinclair, o equipamento de floppy tape não era considerado com seriedade.

O sistema do microdrive é exatamente o mesmo: um rolo de fita passa de forma contínua por um cabeçote gravador e reproduzidor. Além disso, a fita ficou bem mais estreita, cerca de 2 mm de largura, menos da metade da do floppy tape original. Sua confiabilidade, todavia, ainda não foi comprovada.

Um sistema de armazenamento, que oferece alta velocidade e confiabilidade, é a fita cassete digital. Unidades para uso profissional existem há longo tempo, mas a preços altos. Com a introdução do equipamento cassete microdigital, da Philips, dispositivos como o Hobbit (ver p. 94) começaram a surgir. Embora não se apresente sob forma de um rolo contínuo, a fita proporciona velocidade surpreendente. O diretório é mantido no meio da fita, que pode ser bobinada para diante e para trás, sob controle do equipamento operador. Os computadores portáteis PC-1251, da Sharp, e HX-20, da Epson, recentemente começaram a empregar este tipo de microcassete. O único problema destes três tipos de armazenamento é que o software para seus respectivos formatos existe à venda em número limitado. Os computadores PC-1251 e HX-20 permitem carregar programas retirados de um cassete comum e depois retê-los no microcassete interno. O Hobbit e o microdrive podem ser ligados simultaneamente ao computador, como acontece com os gravadores cassete comuns, o que torna o procedimento de copiar ainda mais fácil.

Quanto à confiabilidade comprovada, o sistema cassete microdigital leva vantagem em relação ao stringy-floppy e aos equipamentos cassete convencionais. Entretanto, se estes serão substituídos ou não, permanece uma questão especulativa.



noras de oito pistas, que esteve muito em uso alguns anos atrás. O princípio de funcionamento é simples: o rolo de fita circula constantemente, de modo que os programas são encontrados com maior rapidez. Também é mantido um índice de todos os programas e arquivos armazenados na fita (exatamente como o diretório dos discos); dessa forma, uma lista do conteúdo está sempre à disposição.

Uma vez que a informação é gravada de modo digital e não por métodos sonoros, sua transferência pode ser muito mais rápida — no mínimo cinco vezes mais, ou em proporção ainda maior. Não é necessária uma interface complexa e cara: a unidade utiliza uma entrada paralela normal e todo software operacional está incorporado na unidade ou vem como uma ROM a ser ligada em uma tomada de reserva no interior do computador. O nome dado a essa unidade indica sua posição entre as fitas e os discos flexíveis — utiliza fita, mas funciona como disco.

Infelizmente, as primeiras unidades fabricadas apresentavam muitos problemas. Os mecanismos funcionavam bem, mas as fitas não eram confiáveis. A maior falha estava no fato de que a fita não podia agüentar a tensão ao ser constantemente puxada a partir da metade do rolo e rebobinada de fora. Este inconveniente não acontecia com as fitas sonoras de oito pistas, porque eram muito mais largas e rodavam muito mais lentamente. As primeiras versões

Cabo de interface
Este cabo de fita flexível liga o primeiro microdrive à unidade Interface 1.



Como escolher?

Que micro adquirir e como aumentar seu desempenho com acessórios e periféricos? Aqui estão algumas sugestões para facilitar sua tarefa.

Periféricos

Quando um novo computador é lançado, o manual que o acompanha às vezes descreve uma série de periféricos que talvez ainda estejam em projeto. É importante distinguir entre o que existe e o que está apenas planejado. O equipamento que tem unidade de disco e impressoras disponíveis contará também com o desenvolvimento de software comercial. Se for destinado à diversão, o micro deve ter entradas para joysticks e paddles.

Tela

Dê-se muita atenção aos gráficos do computador, quando na prática a maior parte do tempo se examina um texto ou listagens de programas. Dois fatores são essenciais: o número de caracteres que podem ser projetados de uma só vez (25 linhas de 40 caracteres é a quantidade média) e a legibilidade dos próprios caracteres. A facilidade de alteração e correção na tela também é importante. Pode-se modificar um programa de duas maneiras: ou pela simples movimentação do cursor para a posição a ser corrigida, ou pelo emprego de comandos especiais.

Teclado

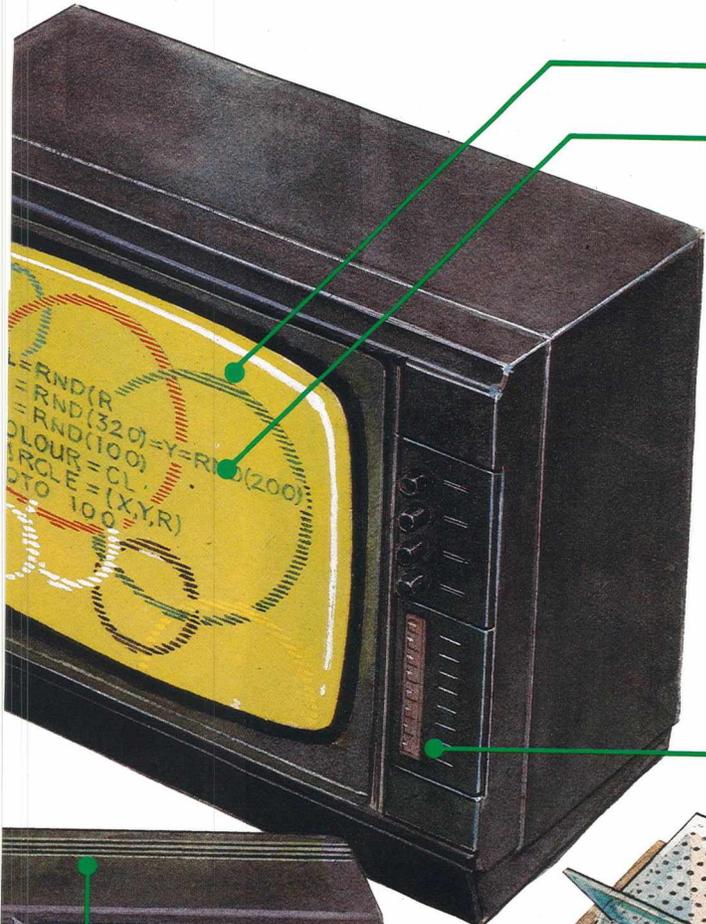
A menos que você queira fazer jogos que utilizem só o joystick, poderá muitas vezes usar o teclado. O importante é que fique satisfeito com sua escolha. Há duas características a considerar quanto a qualidade de um teclado. A primeira é o bom design, que deve apresentar sensibilidade ao toque. Alguns computadores de baixo custo possuem teclas que não oferecem "retorno tátil". A segunda é o layout do teclado, e o ideal seria o de teclas distintas para cada função, já que uma tecla com várias aplicações pode causar confusão. É interessante haver teclas com função programável, específica, como DISPARAR ou COMEÇAR NOVAMENTE.

Software comercial

Se a razão para a compra de um microcomputador é a ajuda que ele presta à pequena empresa, deve-se considerar um computador projetado para esse fim. Mas os microcomputadores mais caros podem ainda ter outros usos, se você estiver preparado para investir numa unidade de disco e numa impressora. As aplicações, então, incluem processamento de palavras, contabilidade, bancos de dados e folhas eletrônicas. Assim como ocorre com os jogos, algumas máquinas possuem grande quantidade de software disponível, e outras não têm praticamente nada.

Interfaces

Um computador que apresenta diversas interfaces sugere que seu projeto visou a uma grande expansão, e essa característica é um ponto positivo do seu design. Se ele possui portas standard, como a RS232 e a Centronics, você terá boa gama de dispositivos para seu computador, além dos oferecidos pelo fabricante.

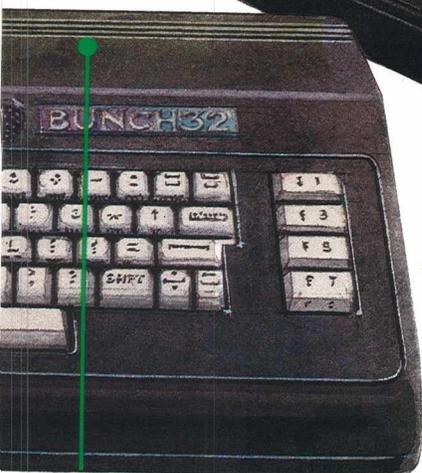


A qualidade do BASIC

A maioria dos micros traz um interpretador BASIC acoplado em ROM, mas, como você percebe nos quadros "A propósito...", os comandos existentes em cada máquina variam muito. Todas as versões de BASIC da Microsoft funcionam de modo semelhante; outros podem usar estruturas diferentes para variáveis, matrizes, tabelas etc. Um bom BASIC é o que tem comandos de "alto nível", como CIRCLE, DRAW e PAINT, para aproveitar os gráficos e sons.

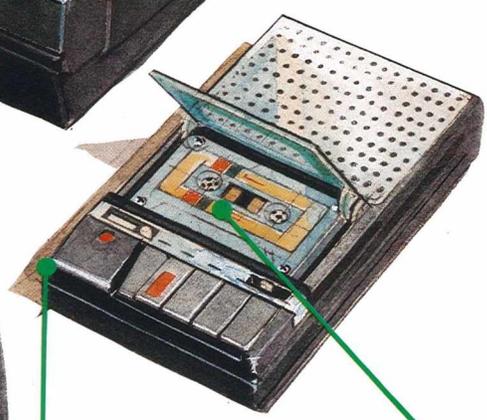
Gráficos

A dificuldade em avaliar gráficos nos micros está em que a maioria deles oferece agora mais de um tipo de gráfico. Digitando-se MODE 1, por exemplo, tem-se acesso a dezesseis cores numa matriz de 40 x 25 posições, enquanto o MODE 7 pode oferecer a resolução máxima de 320 x 200 pixels, mas com a escolha de apenas duas cores. Deve-se observar que em alguns equipamentos só é possível a resolução máxima com o auxílio de um monitor, no lugar de uma tela de televisor comum.



Memória

Quanto maior a capacidade de RAM do computador, mais sofisticados serão os programas que ele pode operar, ou seja, trabalha com mais dados de uma só vez. Os microcomputadores vêm com 1 K a 64 K como standard, apesar de muitos deles terem condição de expandir-se com módulos conectados. Entretanto, o computador pode necessitar de parte dessa memória para seu próprio uso. Gráficos coloridos de alta resolução, especialmente, consomem grande quantidade de RAM.



Gravador cassette

A menos que esteja preparado para restringir-se exclusivamente ao cartucho ROM ou tenha optado por unidades de disco, você vai precisar de um gravador cassette. Algumas máquinas requerem cassette da mesma marca. A maioria dos micros, entretanto, funciona com gravador comum, embora a velocidade e segurança com que os programas são guardados e recuperados variem consideravelmente. Os melhores sistemas permitem, ainda, o controle do motor do cassette.

Software de jogos

A maioria dos melhores softwares de jogos não provém do fabricante do computador, mas de fornecedores especializados. O que influi na qualidade e quantidade dos jogos produzidos para as máquinas é o número de usuários de cada uma delas (portanto, o potencial de vendas). Por isso, alguns computadores são mais adequados para jogos, enquanto outros voltam-se para programação doméstica ou uso comercial.

Som

Os computadores que utilizam o alto-falante da televisão geralmente produzem melhor qualidade de som do que os que têm alto-falante próprio. Mas isso nem sempre ocorre — peça demonstração na loja antes da compra. Um sistema de uma voz permite apenas notas únicas (como se alguém tocasse piano com um só dedo); três ou quatro vozes permitem acordes. Os geradores de som encontrados em alguns computadores são usados para criar explosões e outros efeitos, enquanto aqueles com minissintetizadores imitam diversos instrumentos musicais.



Se você vai comprar um microcomputador pela primeira vez, ou já tem um e quer aproveitar todas as suas possibilidades técnicas, a variedade de computadores disponíveis no mercado pode deixá-lo desorientado. Até agora, neste curso, descrevemos detalhadamente alguns dos micros à venda, explicamos os princípios básicos de operação, vimos diversas aplicações do microcomputador em nossa vida diária e esclarecemos alguns jargões próprios dessa tecnologia.

Há muitos fatores a considerar quando da primeira aquisição de um microcomputador. Antes de tudo, o interessado talvez queira saber o preço do equipamento. Pode-se encontrar nas boas lojas um modelo como o TK83 a um custo acessível. A maior desvantagem, neste caso, é a qualidade ou a "sensibilidade" do teclado do TK83. A fim de reduzir seu tamanho o máximo possível, a Microdigital criou um design de membrana para o teclado. O resultado é a ausência de sensibilidade ou "resposta tátil", que pode deixar o usuário insatisfeito.

O menor custo relaciona-se, ainda, ao fato de um equipamento, como o TK83, não ter condições de produzir cores e sons, além de possibilitar a conexão com pequeno número de periféricos. Muitas destas observações são válidas para outros computadores de baixo preço.

Diversas aplicações

Em geral, espera-se que o microcomputador apresente uma série de recursos, já detalhados antes, e é

vital que estes integrem uma lista personalizada, em ordem de prioridade. Se você, como muitos usuários, prefere jogos eletrônicos, naturalmente irá querer um amplo software, periféricos apropriados, como joysticks, ou paddles, bons gráficos e capacidade de geração de som.

Caso você se interesse pela administração comercial ou doméstica, dará maior importância ao número de colunas mostradas na tela, à qualidade do teclado, às possibilidades de armazenamento de dados, à fácil conexão com a impressora e à disponibilidade de software.

Embora os computadores sejam muito versáteis, alguns se adaptam melhor a certas tarefas do que a outras. É essencial, pois, que você faça uma lista que abranja suas preferências e necessidades, antes de escolher o primeiro equipamento. Relacione todas as coisas que lhe ocorrerem, para maior segurança quanto à qualidade e desempenho na hora da compra.

Se sua escolha recair sobre uma máquina de uso genérico, verifique se tem uma boa versão da linguagem BASIC, teclado adequado, condições de expansão e razoável capacidade de memória.

Vale a pena visitar várias lojas pesquisando a qualidade e o preço dos equipamentos. Você talvez realize, ou não, uma boa compra numa loja sofisticada, mas, de qualquer modo, peça ao vendedor desse e de outros estabelecimentos que lhe mostre os vários modelos que possam responder a suas necessidades.

Não se esqueça também de dar uma olhada nas máquinas de segunda mão, pois inúmeras pessoas já compraram microcomputadores e muitas querem adquirir um novo equipamento mais rápido ou com um maior número de periféricos, e aí pode surgir um

Aviso ao comprador

Novos computadores aparecem mensalmente nas lojas, aumentando a indecisão do comprador em potencial na escolha do melhor aparelho. Você deve considerar diversos fatores para efetuar uma compra adequada.



bom negócio para você que está começando agora. Nas revistas especializadas ou nos classificados de jornais aparecem, muitas vezes, oportunidades excelentes de compra de um equipamento usado.

Ampliando horizontes

Quando você pensar em ampliar os recursos de seu computador e comprar periféricos, o estudo atento do mercado será ainda mais importante. Se você possui um microcomputador de uma determinada marca, muitas vezes não terá outra saída, senão utilizar os periféricos do próprio fabricante original, que apesar de em geral eficientes, limitam eventualmente seu campo de ação.

É possível que você queira aumentar a capacidade de memória de seu micro e que, ao verificar as opções do próprio fabricante do computador, acabe constatando que existem aparelhos similares de outras marcas. Não raro, estes oferecem o mesmo produto por um preço menor, ou potência ligeiramente maior pelo mesmo preço.

Talvez a área que disponha de maior possibilidade de escolha seja a das impressoras. Há numerosos fabricantes e vários tipos desse equipamento à venda. A matricial, a impressora margarida e a tipo térmica constituem as modalidades mais comuns. Se você se interessa pelo processamento de palavras, provavelmente dará preferência à impressora tipo margarida, que produz letras de alta qualidade, parecidas com as de uma máquina de escrever elétrica. Caso você esteja fazendo orçamentos com um programa de folha eletrônica, o primeiro critério que deverá levar em consideração será o comprimento máximo da linha permitido pela impressora.

O próximo passo nesse difícil processo de escolha será, sem dúvida, a análise do volume de armazenamento. Conforme o caso, é preferível investir em unidades de discos flexíveis mais caras, ou se contentar com um gravador cassete. Provavelmente, o equipamento mais dispendioso lhe trará maior satisfação.



Há uma pergunta que parece assustar o comprador em potencial: "E se eu comprar agora e o preço baixar na próxima semana?" Tudo indica que esta situação poderá ocorrer, mas, de qualquer modo, essa não é uma razão suficiente para adiar a compra de um computador.

Você deve adquirir o equipamento adequado a suas necessidades e, como já observamos, o modo mais seguro de conseguir isso é pensar nas aplicações específicas que você dará a seu computador. Lembre-se sempre de que seu computador não consiste só em uma caixa com teclado ligada ao aparelho de televisão, nem tampouco em periféricos e software.

O computador é um instrumento que, utilizado de maneira correta e com sua plena capacidade, pode lhe oferecer horas de entretenimento, além de solucionar problemas e armazenar grande número de registros acessíveis a qualquer momento. Potencialmente, são infinitas as aplicações de seu micro.

Primeiro o software

A menos que você pretenda usar seu computador apenas para aprender a programar, a existência de bons softwares será tão importante quanto as especificações do computador. Quanto mais vende um computador, maior a probabilidade de novos softwares serem lançados para ele e quanto mais software o computador possui, mais ele é vendido — parece que temos aqui situação análoga à do ovo e da galinha.

Olhando as vitrines

Lojas em geral

As lojas que oferecem descontos em aparelhos de som e televisores geralmente têm bons preços para os micros, mas você deve ter certeza do modelo que quer comprar.

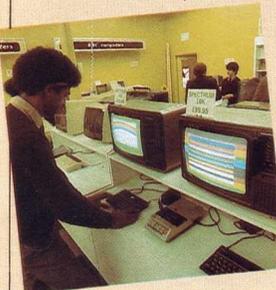


As grandes lojas

Cada vez mais são adquiridos computadores nas grandes lojas de departamentos, que oferecem a vantagem de variado estoque. Até há pouco tempo, os balconistas para a venda de computadores eram poucos. Agora, muitas lojas estão organizando departamentos só de computadores.

Lojas especializadas

Se você comprar seu computador de um comerciante especializado, estará seguro quanto a manutenção do aparelho e as recomendações mais viáveis relativas ao software; entretanto, talvez você não obtenha o melhor preço.



Pedido pelo correio

Em muitos países, este é um sistema comum de venda de microcomputadores. Entretanto, a longa espera do comprador é um dos inconvenientes desta forma de compra.



Sinclair QL

O Quantum Leap oferece o mais avançado microprocessador existente em qualquer microcomputador e um potencial de meio megabyte de memória.

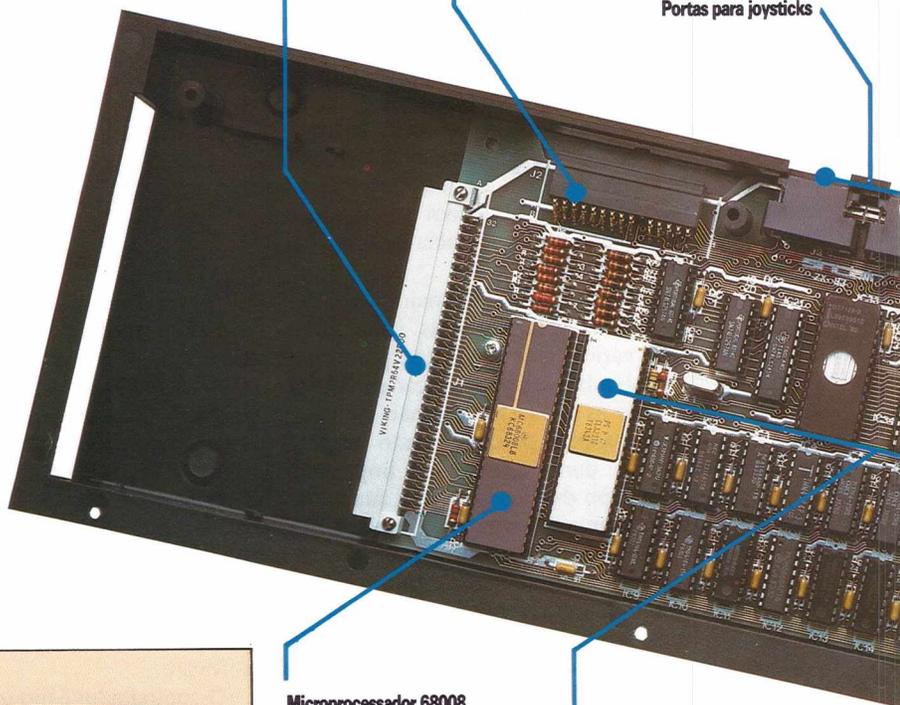
Todas as inovações criadas por Sir Clive Sinclair no campo de microcomputadores representam uma transição brusca (ou salto quântico, *quantum leap*, em inglês), tanto em termos de tecnologia quanto em valor monetário, mas seu microcomputador mais recente é o primeiro a levar esta característica como nome: Sinclair Quantum Leap (QL). Destinase a um número crescente de usuários que ou são grandes entusiastas de computadores, ou têm em mente não só aplicações pessoais como também comerciais. Assim sendo, ele representa uma concorrência muito séria para máquinas como o Commodore 64 e equivalentes, embora seja extraordinariamente superior em especificações técnicas.

É evidente que o projeto do QL é resultado da reunião de todos os componentes e características que significam o que há de melhor em matéria de computadores. Deixando de lado a escolha usual do Z80 ou 6502, a CPU faz parte da família Motorola 68000, que é o mais sofisticado dos microprocessa-

Interface para expansão
Periféricos em geral e até 0,5 megabyte de RAM podem ser aqui acoplados.

Abertura para cartucho ROM
Até 32 K de ROM adicional podem ser ligados aqui.

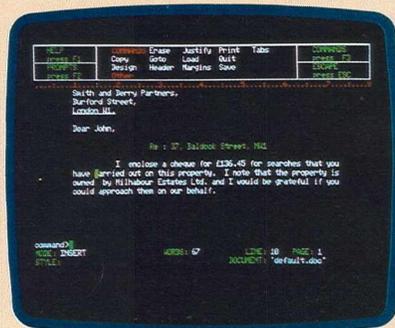
Portas para joysticks



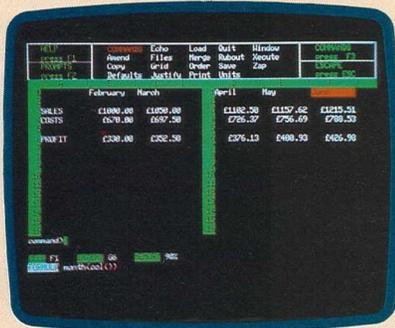
Microprocessador 68008
Este processador apresenta registros internos de 16 e 32 bits, com um data bus externo de 8 bits.

Chips "customizados"
Um número crescente de novos computadores apresenta um chip especialmente projetado. O QL tem dois, para controlar o vídeo e as diversas interfaces.

Software do QL



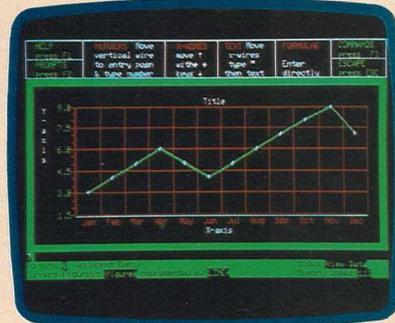
QL Quill é um pacote processador de palavras que exhibe o texto na tela no mesmo formato em que ele será impresso.



QL Archive é um pacote de banco de dados. Os layouts dos registros são desenhados com a ajuda de um editor de tela.



QL Abacus é uma folha eletrônica com a característica incomum de que os elementos podem ser consultados pelo nome.



QL Easel é um pacote projetado para produzir gráficos e mapas, controlando aspectos como a representação automática em escala.

encontrados em microcomputadores e é usado em máquinas como o Lisa, da Apple (ver p. 261). A CPU é uma 68008, o que significa que, embora seus registros internos sejam de 16 bits (podendo também desempenhar muitas funções de 32 bits), seu data bus externo tem a extensão de apenas 8 bits. Isto reduz ligeiramente a velocidade operacional da CPU, pois os registros têm de ser carregados e armazenados em metades. Mas isto também permite manter o baixo custo dos chips, já que economia é um fator fundamental na escolha que a Sinclair faz dos componentes.

O QL vem com 128 Kbytes de RAM, mas, com os futuros acessórios, será expansível até 512 Kbytes (ou "meio mega", como se costuma dizer). Esta memória ampla é particularmente útil para aplicações de uso comercial, pois reduz a frequência com que o programa recorre ao armazenamento off-line. Este armazenamento consiste em dois microdrives embutidos no gabinete, cada um oferecendo aproxi-

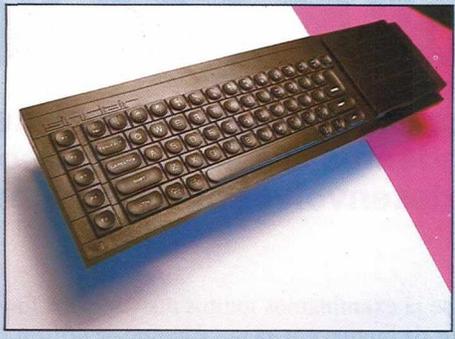


Portas seriais

Duas entradas RS232 estão incorporadas e são adequadas para ligar uma impressora e um modem. A interface mais comum para impressora (Centronics) deve ser adquirida como acessório.

Teclado

Embora o teclado seja construído em membrana (dessa maneira resguardado contra cafezinhos derramados etc.), ele apresenta 65 teclas móveis e o "toque" é tão bom quanto o de algumas das máquinas mais caras de uso comercial. Há quatro teclas para controle do cursor e cinco teclas para funções programáveis.



Tomada para TV

O QL funciona com um televisor, mas só exibe de 40 a 60 colunas; conseguem-se 85 com o uso de um monitor.

Porta para monitor

Ao contrário do Spectrum, o QL pode ser ligado a um monitor RGB, o que realmente é necessário para tirar proveito da resolução máxima de 512 x 256 pixels em quatro cores.

Interface para rede

Até 64 aparelhos QL e Spectrum (este último com uma interface 1) podem ser interligados, formando uma rede local.

Abertura para extensão de microdrive

Como o Spectrum, o QL pode controlar até oito microdrives.

Microdrives

Cada um deles usa um minúsculo cartucho do tipo wafer, contendo uma fita de loop contínuo para armazenar até 100 K cada.

Segundo microprocessador

Este Intel 8049 controla o teclado, o som e as entradas seriais, deixando o 68008 livre para executar o programa do usuário.

madamente 100 Kbytes. Embora isso faça do QL um sistema completo de uso comercial, os microdrives podem ser considerados um ponto fraco quando comparados com o processador extraordinariamente eficiente. Leva em média 3,5 segundos para localizar um item de dados no microdrive, ao passo que na nova geração de miniunidades de discos flexíveis leva em média meio segundo.

A Sinclair pretende produzir uma interface para uma unidade de disco rígido (Winchester), mas não tem planos para discos flexíveis, embora alguns fabricantes independentes os ofereçam. Infelizmente, sem discos o QL não pode executar o sistema operacional Unix, que é considerado uma das principais razões para a escolha de uma CPU Motorola 68000, e que dizem ser o substituto do CP/M como sistema operacional padrão para software comercial.

O QL é acompanhado de quatro pacotes comerciais, todos desenvolvidos pela software-house Pision. Quill é um processador de palavras; Abacus,

Sinclair QL

TAMANHO

472 x 138 x 46 mm

CPU

Motorola 68008

CLOCK

7,5 MHz

MEMÓRIA

128 K de RAM, expansível para 512 K e 32 K de ROM, expansível para 64 K.

VÍDEO

25 linhas de 85 caracteres (com monitor), alta resolução gráfica: 512 x 256 pixels (4 cores), 256 x 256 (8 cores).

INTERFACES

Serial RS232 (2), joysticks (2), microdrives, TV, monitor RGB.

LINGUAGENS

BASIC. Várias outras estão sendo planejadas, principalmente a linguagem "C".

DOCUMENTAÇÃO

O manual de instruções é de alta qualidade, apresentado em uma pasta com espiral, e inclui manuais para o software padrão.

um pacote para folha eletrônica; Archive, um banco de dados; e Easel, um pacote para gráficos. Todos são executados pelo sistema operacional que faz parte da máquina que a Sinclair chamou de QDOS. A popularidade que esta máquina deve atingir significa que uma grande quantidade de software será desenvolvida para ela, embora não seja fácil para as software-houses transferirem os pacotes já existentes para o QL. Além do mais, pode-se argumentar que se a Sinclair adotasse os padrões industriais, seus produtos não estariam liderando o mercado, como realmente estão.

O BASIC da máquina é uma versão aperfeiçoada da linguagem do Spectrum e, como se Quantum Leap já não fosse um nome bastante imodesto, a Sinclair chamou-o de SuperBASIC. Estão incluídos recursos para manipulação de procedimentos (estimulando dessa forma a programação estruturada) e de acesso ao sistema operacional a partir de um programa em BASIC. Tanto o BASIC quanto o QDOS estão contidos nos 32 Kbytes de ROM como padrão.

O Sinclair QL é uma máquina impressionante e, o que talvez seja mais importante, tem possibilidades de expansão suficientes para prevenir-se contra a obsolescência. É mais um acréscimo marcante à longa linha de sucessos da Sinclair: o ZX80, o ZX81 e o Spectrum.

Soluções reais

Utilizando todas as técnicas de programação BASIC que aprendemos até aqui, iniciamos agora os primeiros estudos para desenvolver um programa de banco de dados.

Agora que já examinamos muitos dos aspectos fundamentais da linguagem BASIC, é tempo de utilizar o que aprendemos, desenvolvendo um programa verdadeiro. Evidentemente, todos os programas que vimos até agora são “verdadeiros”, no sentido de que cumprem uma determinada tarefa, mas constituem apenas ilustrações ou exemplos de como funcionam os vários elementos da linguagem BASIC, e não tipos de programa que você poderá querer utilizar diariamente. Eles ilustram como as “engrenagens” do BASIC são ajustadas de modo a formar um mecanismo simples. Agora passaremos a construir todo o mecanismo, ou seja, a desenvolver um programa completo.

Uma pergunta que o não-usuário de computador faz ao usuário é a seguinte: “Em que o computador pode ser realmente útil?” A questão não é tão simplista quanto parece. A resposta a uma dona-de-casa, por exemplo, tende a ser do tipo: “Bem, você pode colocar suas receitas no computador”; ou “Você pode fazer uma agenda telefônica ou de endereços computadorizada”. Esse tipo de resposta não é satisfatório porque geralmente o interlocutor faz a seguinte objeção: “Eu consulto meu livro de receitas quando quero cozinhar e procuro em minha agenda o endereço, sem me dar ao trabalho de escrever um programa que faça isso”. Por conseguinte, precisamos reconhecer quando um problema se torna merecedor de uma solução pelo computador, em vez de lhe darmos uma solução convencional. Vejamos o que ocorre no exemplo específico de uma lista de endereços computadorizada.

Agendas comuns de endereços trazem um índice alfabético que possibilita ao usuário encontrar um determinado nome. Os nomes vão sendo acrescentados à medida do necessário e não se pode obedecer à estrita ordem alfabética. A primeira anotação na letra D talvez seja Dimas. Posteriormente, você acrescenta Dulce e Daniela. Embora não colocados em ordem alfabética, estão agrupados na letra D; assim, é fácil encontrar determinado nome iniciado por essa letra. Nem é preciso dizer que, se não for utilizado um índice, será muito trabalhoso tentar localizar o nome desejado.

As outras anotações num livrinho ou caderneta são o endereço e o número do telefone, e às vezes alguns dados pessoais. Entretanto, uma agenda convencional não nos daria uma lista separada de todas as pessoas com as quais nos interessa manter contato em certa cidade ou que podem atender em determinado número telefônico.

É verdade que isso talvez não represente um sério contratempo, mas, se você dirigir uma pequena empresa que atende pedidos pelo correio, disporá de um recurso valioso se tiver informação específica

sobre as pessoas que constam de sua lista de endereçamento e correspondência. Por exemplo, se a empresa lançar uma nova coleção de roupas infantis e enviar anúncio a seus clientes, poderá economizar em despesas postais, porque provavelmente não será interessante remeter a publicidade a clientes que não têm filhos. Isso serve apenas para exemplificar detalhes que devem ser levados em conta ao decidir se um problema merece ser tratado por um computador ou ter uma solução convencional.

Sendo o recurso ao computador a solução conveniente, a nova questão é saber se existe ou não para ele software à venda. Observando os anúncios de revistas de computação talvez você conclua que todas as possibilidades em programação já foram cobertas pelos programadores. Entretanto, um exame mais atento poderá mostrar que o programa encontrado não realiza exatamente o que você pretende, ou não é compatível com o seu computador, ou, ainda, é muito caro. O preço dos programas, em geral, reflete o custo do trabalho exigido em seu desenvolvimento. Um pacote processador de palavras pode parecer caro à primeira vista, mas escrever seu próprio programa representará um custo ainda maior, se lhe tomar seis meses de trabalho intensivo.

Um aspecto positivo do desenvolvimento de seu próprio software é que ele vai executar exatamente aquilo que você deseja. Outro fator de relevo está na absoluta satisfação de escrever com êxito um programa de importância para você mesmo.

A elaboração de um programa abrange vários estágios. O primeiro é a compreensão completa do assunto, o que requer a colocação clara do problema — o estágio da Definição do Problema.

O segundo consiste em encontrar um método adequado para a solução. Isto envolve a descrição da forma esperada de entrada e de saída, como uma “descrição de primeiro nível” do problema. Os problemas e as soluções devem ser apresentados nos termos mais amplos possíveis, os quais serão gradualmente elaborados até atingir um estágio em que se possa codificá-los em uma determinada linguagem.

O terceiro estágio é a própria codificação, quando usaremos o BASIC como a nossa linguagem de alto nível, embora possamos utilizar qualquer outra linguagem. Até o estágio final da codificação em BASIC, usaremos uma pseudolinguagem, intermediária entre a linguagem livre e flexível com que cotidianamente nos exprimimos e as estruturas rígidas da linguagem real do computador, como o BASIC.

O método de programação que acabamos de expor é em geral chamado de programação top-down. Ela opera do nível mais elevado — uma apresentação genérica dos objetivos mais globais, passando



por vários níveis de elaboração — até atingir o nível de elaboração detalhada do programa necessário para o início da codificação em linguagem de alto nível. Procuraremos, também, ater-nos aos princípios da chamada “programação estruturada”. Estes princípios ficarão claros no decurso deste projeto.

Os estágios incluídos no desenvolvimento do programa podem ser assim resumidos:

1. Definição clara do problema.
2. A forma da entrada e da saída (descrição em primeiro nível).
 - 2.1 Elaboração (descrição em segundo nível).
 - 2.2 Elaboração complementar (descrição de terceiro a enésimo nível).
3. Codificação em linguagem de alto nível.

Antes de iniciar a elaboração de um software de porte, há necessidade de apresentar o problema com clareza — o que não é um trabalho insignificante. Examinemos algumas idéias possíveis para a nossa relação de endereços computadorizada.

Primeiramente, começaremos com uma lista dos aspectos desejados; depois, decidiremos quais deles podem ser realizados com um trabalho moderado de programação. Desejamos, portanto:

1. Encontrar o endereço, número telefônico e anotações, por meio da digitação de um nome.
2. Obter uma lista de nomes, endereços e números de telefone, com a entrada de apenas parte do nome (talvez o sobrenome ou o primeiro nome).
3. Encontrar uma lista de nomes, com endereço e telefone, de determinada área ou cidade.
4. Obter uma relação de todos os nomes que comecem por determinada letra.
5. Conseguir a relação completa de todos os nomes, ordenados alfabeticamente.
6. Acrescentar novos itens, quando necessário.
7. Alterar itens, quando quiser.
8. Eliminar itens, quando desejar.

Vamos supor que o programa da relação de endereços tenha sido escrito. Que forma deveriam tomar as entradas e saídas? Como você gostaria, do ponto de vista de usuário, que o programa funcionasse? Em termos gerais, os programas são dirigidos por menu ou dirigidos por comandos, ou por uma combinação de ambos. No programa dirigido por menu, o usuário recebe sugestões de uma lista de opções (menu), em cada ponto onde uma decisão tiver de ser tomada. Geralmente, a escolha é feita pressionando-se uma única tecla. Em programa dirigido por comandos, o usuário deve digitar palavras ou expressões específicas, sem necessidade de receber sugestões. Alguns programas combinam ambas as técnicas. A vantagem de um programa dirigido por menu é que este será fácil para o iniciante, não familiarizado com seu uso — o que torna o programa um “amigo do usuário” (user friendly). O programa dirigido por comandos apresenta maior rapidez, no caso de usuários mais experientes. Vamos optar por um método dirigido por menu, embora você possa preferir realizar o programa usando rotinas por comandos — a escolha é sua.

Uma vez que o programa se concentrará em uma lista de nomes, a primeira coisa a considerar é a forma que esses nomes deverão assumir. O compu-

tador poderá compreender, por exemplo, todos os formatos que damos a seguir?

A. J. P. Pereira
Leonardo da Vinci
Clara NUNES
Liz T.
A. Barroso
o. v. filho
P Pontes
Susana
CHARLIE CHAPLIN
José da Silva

Talvez pareça um detalhe supérfluo, mas imagine o que acontecerá se você fornecer o nome sob a forma P Pontes e depois solicitar ao programa a localização de P. Pontes. A menos que você tenha previsto esta possibilidade, o computador provavelmente responderá com NOME NAO ENCONTRADO.

Há duas maneiras de evitar o problema: fornecer os dados de modo “impreciso”, permitindo a entrada dos nomes sob qualquer forma com rotinas bem elaboradas que possam lidar com isso quando as buscas forem feitas; ou fazer a entrada dos nomes de forma estritamente definida. Qualquer nome que não estiver na forma convencionada resultará, neste último caso, em uma mensagem de erro, como NOME EM FORMATO NAO ACEITAVEL, ou algo semelhante. A escolha é arbitrária, mas optaremos por uma entrada de modo “impreciso” e deixaremos ao programa a tarefa de realizar a conversão dos nomes para a forma padrão.

Para uma busca em ordem alfabética, consideremos os nomes como constituídos de duas partes — o sobrenome e o restante. É relativamente fácil definir um sobrenome: qualquer série de caracteres alfabéticos em maiúsculas ou minúsculas terminada por um RETURN e precedida por um espaço (ASCII 32). Aqui se apresenta de imediato um problema: o que acontecerá se dermos entrada ao nome Susana, sem deixar um espaço antes? Provavelmente, o programa o rejeitará por apresentar um formato não previsto. Assim, será melhor mudar nossa definição.

Os nomes incluem uma ou mais partes: o sobrenome ou o sobrenome e o nome. O nome consiste em caracteres maiúsculos ou minúsculos e os sinais de ponto, apóstrofo e hífen. Será sempre iniciado por um caractere alfabético e terminado por um RETURN (ponto final para encerramento da linha não será admitido). Se houver um espaço, o último grupo de caracteres — incluindo apóstrofo e hífen — será considerado sobrenome, e outros elementos, inclusive o espaço, nome. Se não houver espaço, o nome completo será considerado sobrenome.

O sobrenome exige cuidado especial, pois em qualquer busca em ordem alfabética sempre terá precedência em relação ao prenome; desse modo, o nome Alberto Pereira será colocado após Tiago Pavan. Se o nome tiver apenas um único grupo de caracteres, como Soares, Susana, ou for um apelido, como Zeca, para os objetivos de nosso programa será considerado um sobrenome.

Em qualquer pesquisa ou colocação em ordem alfabética, que nome deverá vir primeiro: A. J. P. Pereira ou Alfredo Pereira? A escolha é arbitrária, mas a solução mais simples consistirá em desprezar

os pontos de abreviação e os espaços que precedem o último espaço, e apresentar os nomes como AJP PEREIRA e ALFREDO PEREIRA. Se fizermos assim, AJP e ALFREDO serão considerados prenomes e, desse modo, AJP será colocado antes de ALFREDO.

Parte de nosso programa admite nomes como entrada e apresenta como saída o nome, endereço e número telefônico (note que ainda não examinamos o significado dos termos "endereço" e "número telefônico"). Se fôssemos aceitar nomes fornecidos em um formato impreciso, com conversão interna para um formato padronizado, poderíamos esperar na saída a forma padronizada ou a mesma forma assumida na entrada inicial? A saída mais "amiga do usuário" seria o nome na forma inicial, mas, como veremos, isto tornará o programa complicado.

Como primeira tarefa de programação, suponha que um nome tenha sido atribuído à variável alfanumérica NOME\$ e que temos outras duas variáveis, PRENOME\$ e SOBRENOME\$. Como atribuiremos as partes correspondentes de NOME\$ a PRENOME\$ e SOBRENOME\$? Deixando de lado, por enquanto, o problema de manter um registro na forma inicial em que o nome foi fornecido (para que possa ser recuperado mais tarde, quando necessário). Uma apresentação simples do programa seria:

Converter todos os caracteres em maiúsculas.

Eliminar todos os caracteres não-alfabéticos, exceto o espaço final.

Atribuir à variável SOBRENOME\$ todos os caracteres que seguem o último espaço.

Atribuir à variável PRENOME\$ todos os caracteres que precedem o último espaço.

Antes de examinar o modo como este problema pode ser codificado em BASIC, veremos como o processo de programação top-down pode nos levar de uma apresentação muito ampla de nosso objetivo ao ponto em que a codificação em determinada linguagem de programação se torna possível. Você verificará que utilizamos não apenas nomes extensos de variáveis como SOBRENOME\$, mas também palavras de comando como BEGIN, LOOP e ENDLOOP. Há estruturas criadas para auxiliar-nos a definir nosso programa; no último estágio de desenvolvimento serão substituídas por comandos equivalentes em BASIC. Entraremos em maiores detalhes quanto a esses comandos e explicaremos por que deslocamos a posição de algumas linhas no próximo fascículo do curso.

1.ª APRESENTAÇÃO DOS OBJETIVOS

INPUT (ENTRADA)

Um nome (em qualquer formato)

OUTPUT (SAÍDA)

1. Um prenome

2. Um sobrenome

1.ª ELABORAÇÃO

1. Read (ler) NOME\$

2. Converter todas as letras em maiúsculas

3. Encontrar o último espaço

4. Read (ler) SOBRENOME\$

5. Read (ler) PRENOME\$

6. Eliminar caracteres não alfabéticos da variável PRENOME\$

2.ª ELABORAÇÃO

1. Read (ler) NOME\$

2. (Converter todas as letras em maiúsculas)

BEGIN (INICIAR)

Executar um LOOP enquanto os caracteres não examinados permanecerem na variável NOME\$

Read (ler) os caracteres de NOME\$ a seguir

IF (SE) o caractere estiver em minúscula,

THEN (ENTÃO) convertê-lo para maiúscula

ELSE (CASO CONTRÁRIO) nada executar

ENDIF (ENCERRAR A SEQUÊNCIA)

Atribuir o caractere à variável alfanumérica auxiliar

ENDLOOP (ENCERRAR O LOOP)

LET NOME\$ = variável alfanumérica auxiliar

END (ENCERRAR)

3. (Encontrar o último espaço)

BEGIN (INICIAR)

Executar um LOOP enquanto os caracteres não examinados permanecerem na variável NOME\$

IF (SE) o caractere = " "

THEN (ENTÃO) anotar a posição em uma variável

ELSE (CASO CONTRÁRIO) nada fazer

ENDIF (ENCERRAR A SEQUÊNCIA SE-ENTÃO)

ENDLOOP (ENCERRAR O LOOP)

END (ENCERRAR)

4. (Ler SOBRENOME\$)

BEGIN (INICIAR)

Atribuir a SOBRENOME\$ os caracteres à direita do último espaço em NOME\$

END (ENCERRAR)

5. (Ler PRENOME\$)

BEGIN (INICIAR)

Executar um LOOP enquanto os caracteres não examinados permanecerem em NOME\$ até o último espaço

SCAN (EXAMINAR) os caracteres

IF (SE) o caractere não corresponder a uma letra do alfabeto

THEN (ENTÃO) nada fazer

ELSE (CASO CONTRÁRIO) atribuir o caractere a PRENOME\$

ENDIF (ENCERRAR A SEQUÊNCIA SE-ENTÃO)

ENDLOOP (ENCERRAR O LOOP)

END (ENCERRAR)

6. (Eliminar caracteres não alfabéticos de PRENOME\$)

(Isto já foi realizado no item 5)

Este segundo nível de elaboração está agora muito próximo do estágio em que poderá ser codificado em linguagem de programação. Desenvolveremos o segundo item (conversão de letras em maiúsculas) em um terceiro nível de elaboração e em seguida o codificaremos em BASIC. Encontramos, anteriormente, um algoritmo para fazer isso (ver p. 212).

3.ª ELABORAÇÃO

2. (Converter todas as letras em maiúsculas)

BEGIN (INICIAR)

READ (LER) NOME\$

LOOP

FOR = 1 TO comprimento da série de caracteres

READ (LER) caractere L

IF (SE) o caractere estiver em minúscula

THEN (ENTÃO) subtrair 32 do valor ASCII do caractere



```
ELSE (CASO CONTRÁRIO) nada fazer
ENDIF (ENCERRAR SEQUÊNCIA SE-ENTÃO)
LET|SÉRIE TEMPORÁRIA$ = SÉRIE
TEMPORÁRIA$ + caractere
ENDLOOP (ENCERRAR O LOOP)
LET NOME$ = SÉRIE TEMPORÁRIA$
END (ENCERRAR)
```

Este fragmento de programa em pseudolinguagem está bastante próximo da linguagem de programação, e pode ser codificado. Nossa versão de BASIC da Microsoft não admite que nomes de variáveis apareçam por inteiro, por isso usaremos abreviaturas. Assim, NOME\$ passa a ser N\$.

```
1000 REM SUB-ROTINA PARA CONVERSAO EM
    MAIUSCULAS
1010 INPUT "ENTRAR NOME"; N$: REM APENAS
    PARA TESTE
1020 LET P$ = " ": REM ASSEGURAR-SE DE QUE A
    SERIE DE CARACTERES NAO ESTA
    PREENCHIDA
1030 FOR L = 1 TO LEN(N$): REM INDICE DO LOOP
1040 LET T$ = MID$(N$,L,1): REM OBTEM
    CARACTERE
1050 LET T = ASC(T$): REM DETERMINA O VALOR
    ASCII DO CARACTERE
1060 IF T >= 97 THEN LET T = T - 32: REM
    CONVERTER EM MAIUSCULAS
1070 LET T$ = CHR$(T)
1080 LET P$ = P$ + T$: REM P$ E SERIE
    TEMPORARIA
1090 NEXT L: REM ENCERRAMENTO DO LOOP
1100 LET N$ = P$: REM N$ ESTA AGORA TODA EM
    MAIUSCULAS
2000 PRINT N$: REM APENAS PARA TESTE
2010 END: REM APENAS PARA TESTE. SUBSTITUIR
2020 REM "RETORNAR" PARA O PROGRAMA
    PRINCIPAL
```

Este fragmento de programa seria normalmente utilizado como uma sub-rotina a ser chamada de um programa principal. Nós o escrevemos, para finalidade de teste, com uma instrução INPUT, uma instrução PRINT, muitas instruções REM e uma instrução END, as quais deverão ser eliminadas antes que a sub-rotina seja incorporada em um programa completo.

A propósito...

Apesar de não se utilizar letras minúsculas nos equipamentos compatíveis com o Sinclair (TK83 e TK85, CP 200, Ringo, etc.), o programa em questão nestes equipamentos seria:

```
1000 PRINT "FORNECER O NOME"
1010 INPUT N$
1020 LET P$ = " "
1030 FOR L=1 TO LEN N$
1040 LET T$=N$(L)
1050 LET T=CODE T$
1060 IF T >= 97 THEN LET T = T-32
1070 LET T$ = CHR$ T
1080 LET P$ = P$ + T$
1090 NEXT L
2000 LET N$=P$
2010 PRINT N$
2020 REM RETORNAR PARA O
    PROGRAMA REAL
```

Exercícios

■ Elaborar todos os estágios acima até o ponto adequado à conversão para um programa em BASIC. A "pseudolinguagem" que você utiliza não precisa ser igual à nossa, mas será interessante seguir a convenção de usar letras maiúsculas para palavras-chaves que provavelmente corresponderão às palavras de instrução na linguagem final (por exemplo, LOOP, IF, LET etc.). Use letras minúsculas em operações que precisarão ser apresentadas mais explicitamente, quando forem codificadas. Estas poderão ficar em linguagem comum.

■ Estando os programas desenvolvidos até um nível satisfatório de elaboração, convertê-los em módulos de programas (sub-rotinas) em BASIC. Testá-los individualmente, usando entradas fictícias e imprimir as instruções que puderem ser eliminadas depois, se funcionarem adequadamente.

Exercícios de revisão

Estes exercícios trazem exemplos da maioria das instruções e funções usadas em BASIC. Não há questões muito difíceis nem aspectos não vistos anteriormente. Se fizer todos ou a maioria destes exercícios sem dificuldade, considere-se a caminho de tornar-se um bom programador em BASIC.

1. Escrever um programa que admita uma entrada de dois números pelo teclado, somar e imprimir o resultado.

2. Atribuir duas palavras (séries de caracteres) a duas variáveis alfanuméricas e a seguir criar uma terceira variável alfanumérica que concatene (unifique) as duas palavras iniciais. Imprimir a terceira variável.

3. Escrever um programa que possibilite digitar qualquer palavra e que imprima a seguir a extensão da série na mensagem: A PALAVRA QUE VOCE DIGITOU TEM * CARACTERES (* representa o número).

4. Escrever um programa que admita um único caractere fornecido pelo teclado e que a seguir informe o valor ASCII do caractere (em decimal).

5. Escrever um programa que apresente a mensagem DIGITE UMA PALAVRA e em seguida responda A ULTIMA LETRA DA PALAVRA FOI * (* representa a letra).

6. Escrever um programa que lhe sugira DIGITE O NOME DE UMA PESSOA e depois responda O ESPACO CORRESPONDE AO *.º CARACTERE.

7. Alterar o programa acima de modo a imprimir QUARTO ou QUINTO ao invés de 4.º ou 5.º quando o espaço se apresentar nestas posições.

8. Desenvolver um programa que indique ao usuário DIGITAR UMA SENTENÇA e depois responda com a mensagem A SENTENÇA QUE VOCE DIGITOU TEM * PALAVRAS (supondo que o número de palavras será uma unidade maior que o número de espaços na sentença).

9. Teste seu computador a fim de verificar se os caracteres (ou gráficos especiais) foram atribuídos aos valores ASCII de 128 a 255 (utilizar um loop e a função CHR\$(X)).

Sala de espera

O computador transfere informação com tal rapidez que um dispositivo como a impressora não pode acompanhá-lo. A solução é dada pelo buffer, ou memória intermediária.

Os pára-choques (buffers) dos vagões ferroviários são projetados para reduzir um eventual impacto, graças à absorção de energia pelas molas e pistões amortecedores. Os computadores também possuem os chamados buffers, que, de certo modo, funcionam como os amortecedores dos trens, ou seja, como isolantes, possibilitando que as diferentes partes do sistema do computador funcionem juntas.

Esse termo estranho é usado um tanto livremente no mundo dos computadores e abrange dois aspectos distintos. Para o programador, buffer significa um uso especializado da memória do computador; já para o projetista do circuito, quer dizer um tipo de amplificador de um sinal elétrico. O segundo tipo, a que chamaremos buffers de sinal, é operado no próprio painel.

Buffers de memória

Imagine um programa processador de palavras que, entre outras coisas, pode transpor um bloco de texto de uma parte do "documento" na memória do computador para outra parte. O texto consiste em caracteres imprimíveis e espaços, e caracteres não imprimíveis, como RETURN, por exemplo. Todos estes são representados na memória do computador em código binário ASCII. Um byte de memória é necessário para cada caractere. A fim de que os blocos sejam transferidos de sua posição na memória para novas localizações, uma outra parte da memória do computador tem de ser reservada ao armazenamento temporário do texto. Esta área de memória deixada proposadamente de lado para uma tarefa específica recebe o nome de buffer.

Consideremos, por exemplo, a tarefa de imprimir um "documento" ou texto redigido num processador de palavras. O documento consiste, digamos, em 15.000 caracteres, e é claro que eles não podem ser colocados na impressora de uma só vez — a maioria das impressoras não funciona com velocidade superior a 80 caracteres por segundo. Para contornar

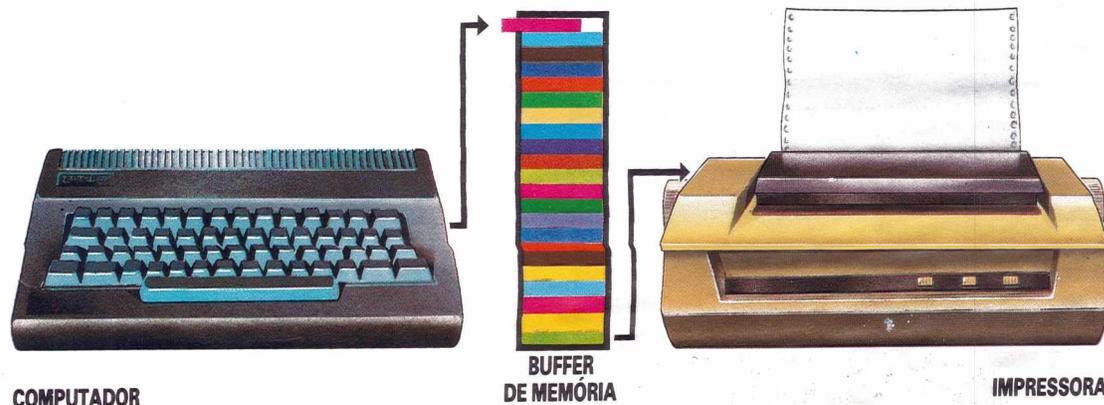
esse problema, uma parte da memória do computador é reservada e mantida sob o controle do software processador de palavra, funcionando como um "amortecedor" de impressão. Primeiramente, o software alimenta este buffer com os caracteres a serem impressos; a seguir, envia-os numa velocidade adequada à impressora.

O buffer de impressão pode não ser muito grande, talvez apenas 128 ou 256 bytes de capacidade, mas o princípio é o mesmo, não importando seu tamanho. Primeiro um "bloco" de caracteres em ASCII é escrito nele, depois os caracteres são enviados novamente, 1 byte de cada vez. O primeiro byte a ser escrito no buffer será também o primeiro a ser lido (é óbvio que nosso objetivo é ter os caracteres impressos do mesmo modo em que foram digitados). Este tipo de buffer é chamado "Primeiro a Entrar, Primeiro a Sair", ou buffer FIFO (First In First Out). Quando todos os caracteres do buffer tiverem sido lidos, o software fará novo preenchimento com os blocos seguintes de caracteres destinados à impressora.

Os buffers FIFO deste tipo constituem um recurso extremamente comum na maioria dos softwares. São utilizados todas as vezes em que há incompatibilidade de velocidade, não só entre computador e impressora mas também entre computador e unidade de disco flexível, e entre computador e teclado. Apesar de a extraordinária velocidade de processamento dos computadores permitir que eles localizem instantaneamente as teclas pressionadas, há momentos em que o computador não pode identificá-las nem apresentar os caracteres correspondentes tão rapidamente. Isso acontece quando o computador está, em dado momento, ocupado em outra função (acessando um disco, por exemplo). Neste caso, é comum a incorporação de um "buffer de digitação" ao sistema de operação do computador. O buffer "lembra" quais teclas foram pressionadas e o computador apresenta os caracteres tão logo seja possível. Normalmente, o usuário não percebe isso,

Parada temporária

Um dos usos mais comuns do buffer é entre o computador e uma impressora, porque esta não pode liberar caracteres com a mesma velocidade que o computador os envia. Os caracteres são, portanto, armazenados na memória temporária até que o buffer esteja completo; um sinal de "ocupado" é enviado ao computador para que este pare de transmitir. O conteúdo do buffer é então enviado à impressora na mesma ordem em que foi recebido, mas num andamento bem mais lento. Quando essa parte termina, o processo recomeça, até que todo o texto seja impresso.





Se derrarmos a água de um balde num tanque, como o que aparece acima, perceberemos que a torneira continua com a saída de água com vazão constante. Do mesmo modo, o computador "derrama" dados no buffer, que os libera numa velocidade bem mais lenta, adequada aos diversos periféricos.

mas em certos sistemas operacionais de disco, ou em certos tipos de software de aplicação (que envolvem muito processamento de informação), pode haver um pequeno atraso entre o momento em que a tecla é pressionada e o aparecimento do sinal na tela. Alguns sistemas operacionais permitem que o buffer de digitação seja ligado e desligado, e que sua capacidade seja alterada pelo usuário.

A maneira exata com que o software é organizado para manejar buffers varia, dependendo do que o buffer deve fazer; mas, geralmente, é necessário deixar de lado alguns bytes, que serão usados como contadores e sinalizadores (flags). É preciso saber quantos bytes foram lidos de um buffer completo antes que mais bytes sejam escritos; caso contrário, dados importantes podem ser destruídos antes de serem usados.

Hardware de buffers de memória

Quem possui uma impressora deve ter notado que ela parece ser lenta, principalmente quando imprime uma listagem ou texto longo. De modo geral, os sistemas operacionais do computador não conseguem desempenhar nenhuma outra atividade enquanto a impressora está em funcionamento; assim, se ela demora muito para terminar a impressão, o usuário tem de esperar a conclusão do processo. Atualmente, muitos fabricantes oferecem buffers de impressão como acessório, em geral na forma de uma caixa que liga o computador à impressora. Essa caixa acelera o desempenho da impressora mas só em relação ao computador. Apesar de, na verdade, a impressora não estar realmente imprimindo com maior rapidez, a caixa contém memória extra, específica (às vezes de até 16 Kbytes), com seu próprio software incorporado. Para o computador, parece que a impressora está trabalhando com maior velocidade. Quando o computador tem de imprimir um arquivo de dados, ele envia bytes para a impressora até receber um sinal de "ocupado" — isso significa que a impressora não pode aceitar mais bytes. O computador, então, espera que o sinal "ocupado"

se torne "falso", indicando que a impressora está livre novamente para aceitar dados. As impressoras têm, geralmente, um pequeno buffer de memória incorporado, mas isso não passa de 2 Kbytes e não permite que o computador envie mais dados até que o buffer esteja vazio. O hardware acessório de buffers contém maior quantidade de memória; assim sendo, pode aceitar mais dados antes de enviar um sinal de "ocupado" ao computador. Se o buffer for suficientemente grande, será capaz de reter todos os dados a serem impressos em uma única vez, e o computador poderá prosseguir com outras tarefas, enquanto o buffer envia os dados em menor velocidade para a impressora.

A memória é mais frequentemente utilizada como um "pombal" que armazena programas e dados, mas ela pode também ser organizada em pilhas ou buffers. Pilha significa "Último a Entrar, Primeiro a Sair" ou estrutura LIFO (Last In First Out), ao passo que os buffers são estruturas FIFO. Pode-se fazer uma analogia com uma pilha de pratos: o último prato empilhado será o primeiro a ser retirado. Como os buffers, as pilhas são áreas de memória temporária e diferem dos primeiros apenas na ordem em que os dados entram e são recuperados. As pilhas são usadas "internamente" em linguagens de alto nível (em interpretadores BASIC, por exemplo) quando a informação precisa ser armazenada temporariamente e recuperada mais tarde. Considere este fragmento de programa BASIC:

```
FOR X = 1 TO 10
PRINT "X = ";X
FOR Y = 1 TO 10
GOSUB SCAN
NEXT Y
PRINT "C$ = ";C$
NEXT X
```

Este é um exemplo de ninhos de loops FOR-NEXT. Quando o interpretador BASIC chega à segunda instrução FOR, ele deve lembrar-se da variável utilizada no primeiro FOR (X, neste caso), e assim "empurra" a informação com o primeiro FOR para cima da pilha. Quando o loop interior está completo, ele toma a informação do alto da pilha e sabe que o FOR em questão usa a variável X. Já que os loops FOR-NEXT podem ser muito bem guardados, ele talvez precise empurrar informações para vários FOR sobre a pilha. Ao tirar a informação da pilha, o interpretador BASIC precisa obviamente ter a informação na ordem inversa da que ele empurrou.

Os buffers, entretanto, organizam a memória de tal modo que a primeira informação que entrou é a primeira a sair. Os buffers são frequentemente utilizados em rotinas de entrada/saída e também como "interfaces" entre as rotinas e os dispositivos que funcionam em diversas unidades ou em diferentes velocidades. Por exemplo, uma rotina de entrada em BASIC poderá trabalhar em unidades de linhas, terminadas por um RETURN <CR>, mas o interpretador trabalharia nas linhas em unidades de um caractere. Os buffers geralmente necessitam de um "indicador" para apontar onde o próximo caractere deve ser escrito. O indicador seria um byte ou vários bytes que contenham o endereço desse caractere. O endereço seria incrementado após o armazenamento de cada caractere.

Sinais poderosos

A lógica interna de seu computador funciona em nível TTL (Transistor-Transistor-Logic). Esta lógica significa o binário 1 com 5 volts, e o 0 com zero volt. Entretanto, apesar de dispositivos como a CPU serem capazes de produzir essas voltagens, não podem gerar corrente suficiente para levar todos os outros chips que devem ser ligados a cada pino. Os buffers de sinal são, assim, ligados às linhas de output da CPU, que aumenta a quantidade de corrente existente. Os buffers de sinal são, em si, pequenos chips, cada um tipicamente capaz de agir como um buffer para seis sinais.

Dados contínuos

Computadores analógicos, utilizados no controle de máquinas e processos, reagem imediatamente a mudanças no mundo real, sem necessidade de converter informação para a forma digital.

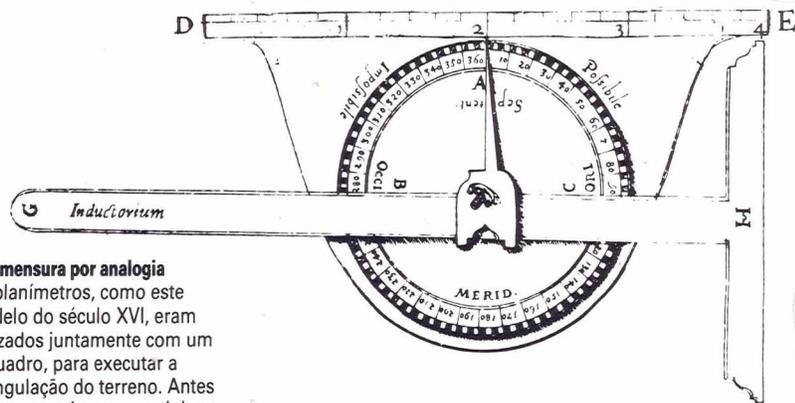


Painel de ligações

Os computadores analógicos não utilizam linguagens como o BASIC. São "programados" pela instalação elétrica de vários componentes. Estes são fixados na parte posterior do painel de circuitos. Na parte anterior encontram-se os pinos e tomadas que permitem a ligação dos componentes escolhidos.

Há dois tipos bastante diferentes de computadores e, até agora, examinamos apenas um deles: o computador digital, assim chamado porque todas as instruções dos programas e todos os dados são representados pela utilização de dígitos binários. O outro tipo, mais antigo, compreende os computadores analógicos.

Os velocímetros são exemplos simples de um computador analógico; este nome é derivado da re-



Agrimensura por analogia

Os planímetros, como este modelo do século XVI, eram utilizados juntamente com um esquadro, para executar a triangulação do terreno. Antes disso, os agrimensores tinham de fazer medições e depois efetuar cálculos matemáticos, a fim de determinar o comprimento do lado mais afastado do triângulo; mas este instrumento analógico simples tornou o trabalho consideravelmente mais fácil e preciso. Instrumentos com nomes semelhantes ainda são usados hoje, para medição de área de planos irregulares.

lação análoga da velocidade do carro com a posição do ponteiro no mostrador, a segunda diretamente proporcional à primeira. Os modernos computadores analógicos desempenham muitas tarefas e são baseados nos componentes comumente empregados em mecanismos eletrodomésticos — transistores, capacitores, resistores e indutores magnéticos.

No início do desenvolvimento da eletrônica, verificou-se que o comportamento dos componentes elétricos assemelhava-se ao de dispositivos mecâni-

cos. Por exemplo, os engenheiros eletrônicos descobriram que as oscilações da corrente elétrica que resultam da interligação de um indutor magnético e um capacitor produzem oscilações extremamente semelhantes às de um peso suspenso por uma mola. A formulação matemática dos dois sistemas é idêntica e constitui a base da computação por analogia.

Alguns dispositivos analógicos são perfeitamente similares aos sistemas dos quais são modelo — por exemplo, o modelo de avião usado experimentalmente em um túnel de vento ou aerodinâmico é a cópia, em escala menor, da estrutura de um avião. Outros modelos são muito diferentes. Um "modelo" de uma situação da vida real pode ser simplesmente uma relação de fórmulas matemáticas, ou então, um circuito elétrico que imita o fluxo da água em uma represa, por exemplo.

Máquinas calculadoras que utilizam princípios analógicos mostraram sua importância pela primeira vez com a invenção da régua de cálculo, por William Oughtred, em 1630. Os números eram dispostos em duas réguas (distribuídos de forma logarítmica), de tal modo que o deslizar de uma ao longo da outra equivalia à multiplicação, e a resposta era simplesmente lida na graduação da régua.

No final do século XIX, Lord Kelvin projetou um engenhoso dispositivo mecânico, movido a manivela, que podia ser utilizado para cálculo das marés alta e baixa em um porto, durante todo o ano.

Em 1930, foi construída uma máquina eletromecânica nos Estados Unidos que resolvia de modo geral equações diferenciais (que surgem em quase toda representação matemática do mundo concreto), em vez do conjunto específico de equações diferenciais que Kelvin conseguira resolver com seu engenho. Foi inventada por Vannevar Bush e chamada analisador diferencial.

Os primeiros analógicos

Logo após o surgimento dos computadores digitais, começaram a operar, em 1947, os primeiros computadores analógicos inteiramente eletrônicos. ~~Vimos como os computadores digitais efetuam cálculos aritméticos usando uma combinação de portas lógicas (ver p. 68).~~ O computador analógico efetua cálculos matemáticos simplesmente pela utilização das propriedades da eletricidade. Se houver, digamos, uma corrente elétrica de 5 ampères passando por um fio e uma corrente de 4 ampères passando por outro, e os dois fios forem unidos de modo a formar um só, a corrente do fio resultante terá 9 ampères — a soma das duas correntes. Assim, o comportamento e as propriedades do fluxo da corrente criarão automaticamente um "adicionador".



Mesmo as funções matemáticas mais complexas têm geralmente soluções simples em circuitos elementares (um exemplo é a integração, o cálculo da área sob uma curva). Os computadores analógicos não são “programados” como máquinas digitais; ao contrário, é preciso formar um circuito que modele o programa a ser desenvolvido. Os componentes necessários são montados na parte posterior de um painel de ligações, e na parte anterior situam-se pinos e tomadas para a ligação conjunta dos componentes. Os painéis de ligação assemelham-se aos antigos quadros de distribuição operados por telefonista.

No computador analógico as variações de voltagem ou corrente são usadas para representar quantidades físicas, como força ou velocidade, e “valores” dos componentes elétricos, símbolos de elementos como a massa do carro ou a força de suas molas. Mas, no computador digital, todos os dados são representados por séries de impulsos, 5 volts para o binário 1 e zero volt para o binário 0. Aqui se verifica outra diferença entre o computador analógico e o digital: no primeiro, a informação pode ser armazenada de modo a corresponder a intensidades de impulso que continuamente variam, enquanto, no segundo, os dados são armazenados em unidades “discretas” ou isoladas.

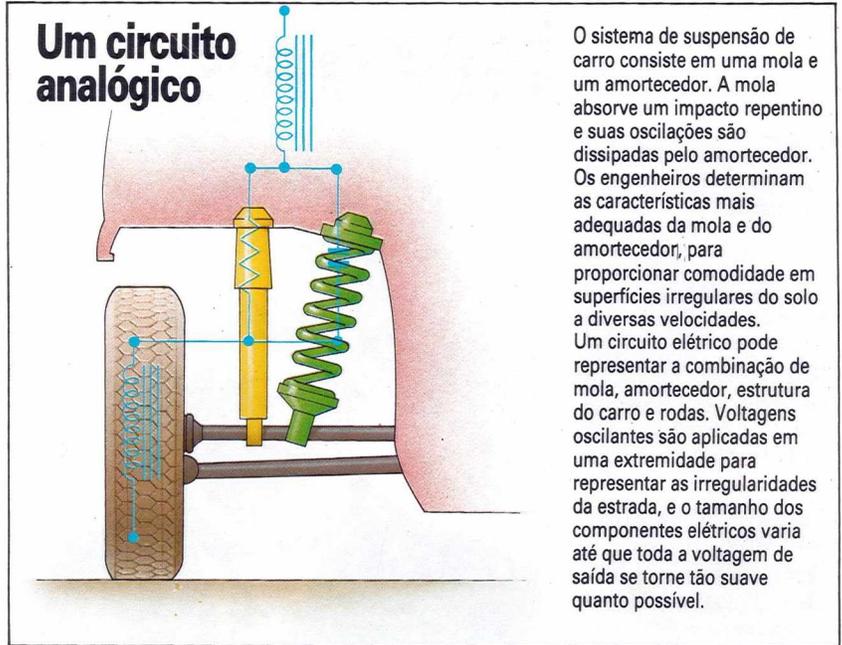
Alguns microcomputadores têm uma tomada para entrada analógica. Em geral, quando computadores digitais trabalham com dados analógicos, tais como temperatura ou força, primeiramente convertem os dados em forma digital.

A grande vantagem dos sistemas digitais está em que a informação é processada ou transmitida sem perda de qualidade. Um impulso de 5 volts fornecido pelo circuito elétrico pode ser afetado por distorção inerente a qualquer circuito e se apresentar, por exemplo, como 4,9 volts. No sistema analógico, em que as variações de voltagem representam alterações na informação, isto poderá significar a diferença entre a emissão de uma nota lá e um lá bemol na voz de uma cantora. Mas no sistema digital, em que são admitidos apenas dois sinais, 5 volts ou zero, qualquer sinal próximo a 5 (como 4,9) será automaticamente reconhecido e restaurado como um sinal de 5 volts. Assim, as falhas são corrigidas imediatamente e não se acumulam. Em contrapartida, o acúmulo de falhas ou erros à medida que o sinal passa por sucessivos circuitos é uma das desvantagens do computador analógico.

Todavia, o computador analógico obtém uma vantagem na representação de quantidades, como valores variáveis de uma corrente ou voltagem. Essa vantagem está no fato de que uma condição de entrada pode mudar subitamente e o sistema corresponderá às alterações de modo imediato. Não é preciso tempo algum para codificar os dados em impulsos binários, processá-los e finalmente decodificá-los para a saída.

Essa característica é muito importante em operações em que respostas rápidas são fundamentais. Por exemplo, um piloto automático deve responder a uma súbita rajada de vento durante uma aterrissagem, porquanto não há tempo de fazer cálculos, mesmo na velocidade em que operam os modernos computadores digitais. Os sensores detectam a ra-

Um circuito analógico



O sistema de suspensão de carro consiste em uma mola e um amortecedor. A mola absorve um impacto repentino e suas oscilações são dissipadas pelo amortecedor. Os engenheiros determinam as características mais adequadas da mola e do amortecedor, para proporcionar comodidade em superfícies irregulares do solo a diversas velocidades. Um circuito elétrico pode representar a combinação de mola, amortecedor, estrutura do carro e rodas. Voltagens oscilantes são aplicadas em uma extremidade para representar as irregularidades da estrada, e o tamanho dos componentes elétricos varia até que toda a voltagem de saída se torne tão suave quanto possível.

jada repentina pela geração de uma voltagem de saída relativamente pequena. O circuito do piloto automático responde de imediato com uma mudança alta da voltagem de saída, que aciona de maneira automática os freios aerodinâmicos das asas, para manter o avião na posição correta de equilíbrio.

Os computadores analógicos são utilizados em muitas áreas de controle industrial, onde equipamentos complexos devem ser manejados com precisão e ajustes minuciosos e contínuos, como numa fábrica de produtos químicos. Mas permanecem menos conhecidos que seus equivalentes digitais. Embora computadores analógicos simples sejam às vezes usados em escolas com propósitos didáticos, as aplicações para as quais são adequados indicam que provavelmente jamais teremos microcomputadores analógicos. Há trinta anos parecia haver acirrada competição entre o computador analógico e o digital. Computadores analógicos sempre terão sua utilidade, mas o mercado será dominado pelos computadores digitais, que estão se tornando a cada dia mais poderosos e rápidos.



Cochichos chineses

O modo pelo qual uma mensagem é distorcida em um jogo de “cochichos chineses” tem um paralelo muito grande com os erros cumulativos em um circuito analógico.

Em um circuito digital, apenas uma quantidade limitada de mensagens (efetivamente 1 ou 0) pode ser transmitida. Desse modo, se qualquer distorção surgir, poderá ser eliminada no estágio seguinte.

Herman Hollerith

1860

Nasce em Buffalo, Estado de Nova Iorque.

1879

Forma-se na Universidade de Columbia. Trabalha no National Census Office dos Estados Unidos.

1882

Faz pesquisas no Instituto de Tecnologia, em Massachusetts.

1883

Trabalha em Washington no Patent Office.

1884

Primeiras patentes da fita contínua de papel perfurado para representação de informações.

1887

Seu sistema é adotado para processar estatísticas de mortalidade, em Baltimore, Maryland.

1889

Seu sistema é instalado no Hospital Militar para fazer estatísticas médicas do exército. Concedida patente do cartão individual.

1890

Vence a concorrência para fornecer equipamento para o censo de 1890. Ganha título de PhD da Universidade de Columbia por trabalho sobre processamento de informação.

1900

Introduz nova geração de equipamento aperfeiçoado.

1901

Novo equipamento é usado no censo de agricultura.

1905

Suas primeiras patentes começam a expirar e há ameaças ao seu monopólio.

1911

Abre uma empresa holding, a Companhia de Registro de Tabulação.

1914

O famoso empresário Thomas J. Watson assume a direção da companhia.

1924

O nome da empresa é mudado para International Business Machines (IBM).

1929

Morre em Washington, DC.



O inventor que registrou os números da população americana em cartões e fundou a maior empresa de computadores do mundo.

Nascido nos Estados Unidos em 1860, Hollerith graduou-se na Universidade de Columbia, tornando-se depois assistente do National Census Office (NCO), onde ajudou a compilar as estatísticas demográficas de 1880. Nesse tempo, o trabalho era todo feito a mão, meticuloso e demorado — tão lento, de fato, que chegada a ocasião do novo recenseamento, dez anos depois, o departamento americano ainda tabulava os resultados anteriores. Hollerith estava consciente de suas aptidões, e, pretendendo desenvolver sua capacidade para inovações, deixou o NCO e passou a trabalhar no Departamento de Patentes, em Washington.

A primeira idéia de Hollerith foi codificar informação numa fita de papel, marcada a tinta e dividida em “campos”. Cada campo representava grupos diferentes — por exemplo, masculino ou feminino, negro ou branco. Um furo no campo masculino/feminino representava indivíduo do sexo masculino, ao passo que a ausência de furo significava sexo feminino, e assim por diante. Esses furos seriam mais tarde “lidos” por uma máquina. Suas primeiras patentes foram obtidas em 1884, e durante os anos seguintes ele aperfeiçoou seu sistema. Começou processando dados estatísticos de saúde pública, trabalho resultante do rápido desenvolvimento das cidades americanas, e fez levantamentos para a administração do exército.

Decorridos cinco anos, em 1889, ele aperfeiçoou a idéia da fita de papel perfurada, utilizando cartões para cada indivíduo. Os cartões tinham o tamanho da cédula de 1 dólar — em parte, segundo se diz, porque o único equipamento que poderia ser adaptado tinha sido criado para manejo de dinheiro. Os furos eram originalmente redondos e feitos com perfuradoras usadas por cobradores de ônibus; mais tarde foram produzidas perfuradoras especiais para um furo quadrado de 6 mm. Por este meio, muita informação podia ser contida num único cartão.

A vantagem dos cartões individuais sobre a fita contínua é que a informação podia ser específica ou genérica. Por exemplo, para encontrar o número de mulheres brancas com 80 anos que viviam em Nova Iorque, todos os cartões seriam classificados, e aqueles com furos perfurados nos campos que não interessavam seriam mecanicamente separados do resto. As primitivas máquinas apenas produziam um total geral, mas com o tempo Hollerith introduziu a adição e outras operações aritméticas simples.

O sucesso comercial veio em 1889, quando o Bureau of Censuses anunciou uma concorrência para instalar um sistema de equipamento processador do censo do ano seguinte. Os sistemas foram testados com nova tabulação dos números anteriores. O equipamento de Hollerith venceu. Suas máquinas estavam protegidas por patente de invenção e por isso ele cobrou a taxa de 65 cents do governo, no processamento de cada conjunto de mil cartões. Apesar de haver um cartão para cada indivíduo, Hollerith levou apenas dois anos para concluir a apuração. Ele anunciou que a população era de 56 milhões, e cobrou do governo o valor equivalente.

Na época do censo de 1900, ele tinha desenvolvido um equipamento ainda mais eficiente, mas recusou-se a baixar o preço estipulado pelo trabalho. Quando se esgotou o prazo de validade de sua patente, o governo tentou negociar com algumas companhias, mas Hollerith ganhou a luta criando sua própria empresa, que mais tarde se tornou a International Business Machines. Hoje, a IBM é a maior fabricante de computadores do mundo, com um movimento comercial de 20 bilhões de dólares por ano.

Jogos de cartões

O método original de Hollerith de representar informação é usado até hoje, um século mais tarde, embora o formato dos cartões tenha mudado. Os cartões perfurados atuais têm 12 linhas de 80 colunas. As máquinas de tabular

usavam o sistema decimal, e cada cartão podia acumular 80 números. Os caracteres alfabéticos eram criados por multiplicação, fazendo-se mais de um furo em cada coluna. Os computadores também podem aceitar cartões perfurados em sistema binário.



Chips & bytes

Como sobreviver às tentações do micro	13
Perguntas e respostas	24
O futuro chegou	28
Perguntas e respostas	48
Quando 1 + 1 = 10	54
Tudo sob controle	60
Perguntas e respostas	64
Mensagem recebida	66
Menos igual a mais?	79
Código decifrado	84
Quem é o quê?	101
O grupo dois	119
Microeletrônica	121
Números ao acaso	209

Conexões

Rumo à expansão	20
Fechando contato	36
Ação rápida	56
Pronta para imprimir	74
Grave e archive	94
A ligação que faltava	108
Memória permanente	114
Mantendo o foco	132
Caneta mágica	156
Sobre duas rodas	176
Conversa de amigo	186
Os traços perfeitos	198
Diálogo a distância	216
O pequeno notável	224

Fundamentos

Bits e bytes	32
Memória infalível	58
Verdadeiro ou falso?	68
Caixa-forte	92
Lógica misteriosa	96
Diálogo digital	112
Leis do pensamento	128

O centro nervoso	138
O endereço certo	144
Números hexadecimais	179
Peek e poke	188
Entradas e saídas	206
Sala de espera	236

Hardware

O que é computador?	1
Qual deles?	14
A ficha técnica	12
Micros em movimento	65
A casa automática	106
A era dos portáteis	166
Como escolher?	226
Dados contínuos	238

Os precursores

Contato!	46
Do ábaco ao micro	86
Sir Clive Sinclair	120
John von Neumann	140
Steve Wozniak	155
Chuck Peddle	180
Alan Turing	200
Charles Babbage	220
Herman Hollerith	240

Perspectivas

O enigma das barras	21
O professor eletrônico	25
Nos bastidores	41
Um novo aluno	81
Micros na medicina	126
Música eletrônica	141
Os micromundos	164
Imagens animadas	181
O voo simulado	201
Informação dividida	218

VOLUME 1

Por dentro do hardware

CP 500.....	9
TK85.....	30
CP 300.....	49
Unitron AP II.....	70
Nexus 1600.....	89
TK2000.....	109
D-8100.....	130
Elppa Jr.....	150
I-7000.....	169
Commodore 64.....	189
Micro Engenho 2.....	210
Sinclair QL.....	230

Programação Basic

Às suas ordens.....	16
Loops sob controle.....	38
Direto ao ponto.....	52
Problemas de rotina.....	77
À espera do Natal.....	98
Desafie os elementos.....	116

Organize seus dados.....	134
Descubra as funções.....	146
Tentando a sorte.....	172
Segunda dimensão.....	194
Novas estruturas.....	212
Soluções reais.....	232

Software

Domine seu micro.....	5
Jogos e brincadeiras.....	22
O micro: um artista.....	34
Pintando com números.....	44
O texto perfeito.....	61
Consulte o chip.....	72
O mapa lógico.....	104
Siga as pistas.....	124
Gráficos em dimensão.....	152
Faça suas previsões.....	158
Quando o herói é você.....	161
Tradução alternativa.....	184
Piratas à vista.....	192
Colocando em ordem.....	204
Inimigo eletrônico.....	221

