

Spooler

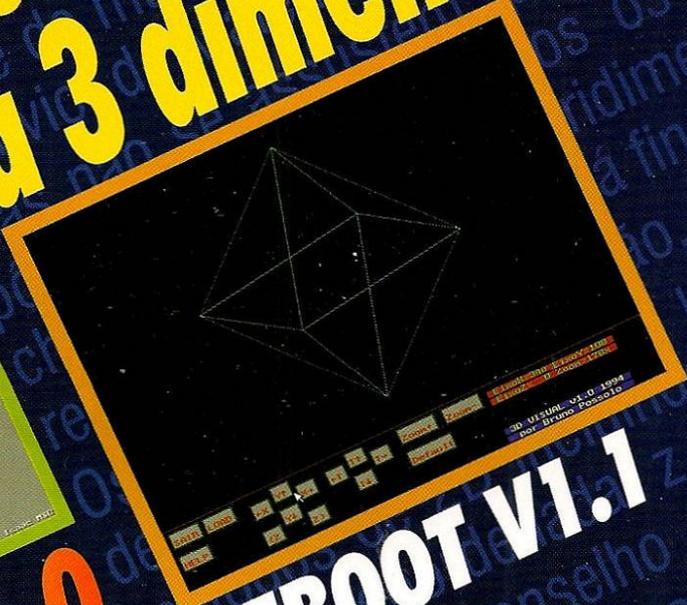
MAGAZINE DOS UTILIZADORES PC

A PROGRAMAÇÃO ORIENTADA POR OBJECTOS E O VISUAL BASIC (X)

Desenho a 3 dimensões

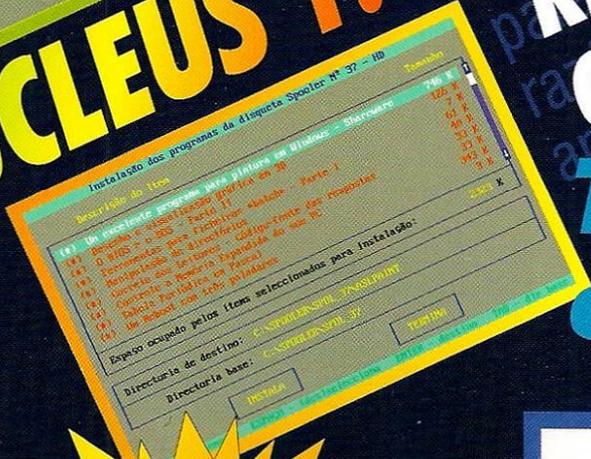
ANÁLISE E PROJECTO DE SISTEMAS

EMSTEST
SOBGELO
UNIDADES



NEWFUNCS.PRG
USERLIB V2.0
WEIRD 2

NUCLEUS 1.0



REBOOT V1.1 CDIR

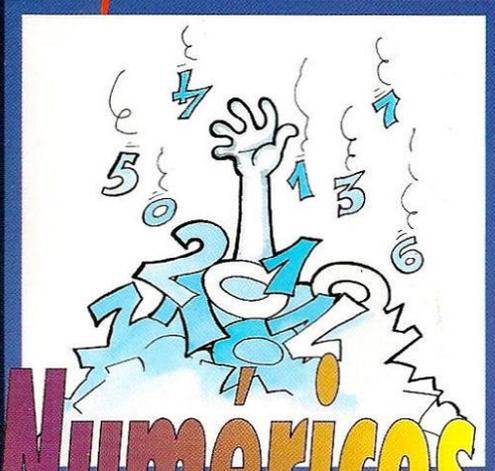
O BIOS E O DOS (II)
BATCH TOOLS (I)
SHAREWARE (IX)

Truques e Dicas para o Windows

Instala 2.0

EM BUSCA DO PC PERFEITO

Metodos Numericos

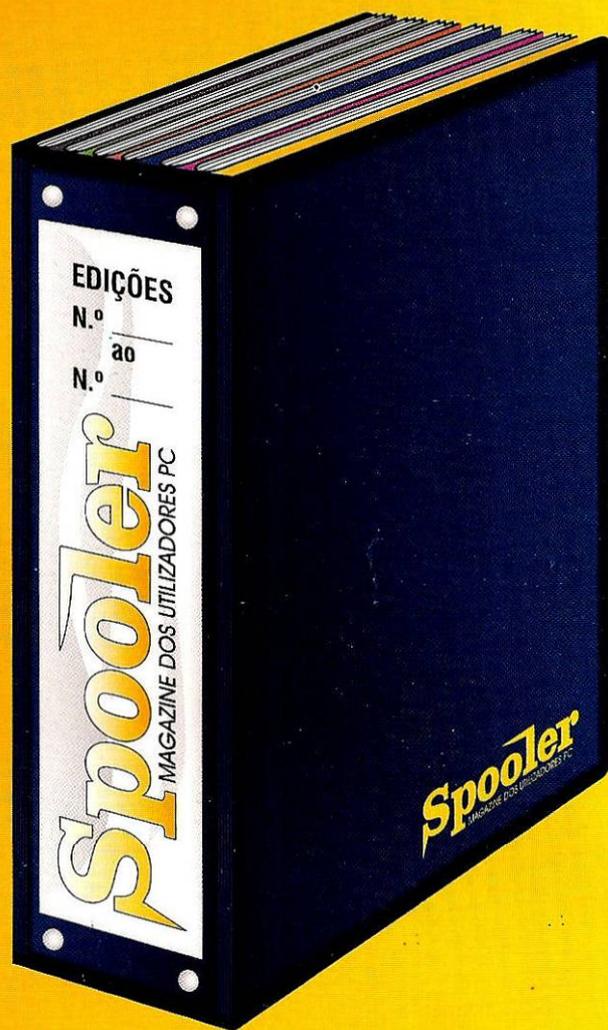


**Agora já pode
ter as suas revistas
arrumadinhas!**

Com as pastas

Spooler
MAGAZINE DOS UTILIZADORES PC

*poderá, finalmente,
arquivar de forma
simples e prática
as suas Revistas.*



2 800\$00
IVA INCLUIDO

**Para encomendar
utilize o cupão
da Montra Spooler**

Ref. CAPS_1

***O seu engenhoso sistema de arquivo
permitir-lhe-á colocá-las e retirá-las
sem as danificar.***

*As revistas não precisam
ser perfuradas*

Editorial

Novo INSTALA, e não só...

No número anterior da Spooler o tema deste espaço foi o «novo visual» que decidimos imprimir à nossa revista. Em boa hora o fizemos, pois obteve a generalizada aprovação dos nossos assinantes e leitores.

A Spooler e a sua disquete constituem um «todo» inseparável. Quem poderia imaginar a nossa revista sem a disquete sempre «recheada» de interessantes utilitários e, não menos importante, do código-fonte dos programas apresentados ao longo das suas páginas? Seria impensável!

O percurso da Spooler tem-se pautado pela permanente preocupação de preservar os principais valores que defendemos para uma «verdadeira» revista de informática. Para bem ou para mal (não dos nossos pecados, julgamos...), a ausência de publicidade, com todas as suas sequelas, não tem prejudicado tais objectivos. A nossa presença nas bancas e em casa dos fiéis assinantes sofrerá de uma irregularidade crónica; mas, mais tarde ou mais cedo, a Spooler (com a sua disquete) surge em toda a sua plenitude: orgulho para nós, que a fazemos, gáudio para os leitores, a quem a dedicamos.

Quando, no Nº 29, decidimos introduzir um utilitário para a instalação automática do material contido na disquete, fomos orientados por dois objectivos: primeiro, facilitar a vida aos leitores menos experientes; segundo, através do código-fonte do dito utilitário, proporcionar rotinas em *Assembly* para acesso aos discos e ao ecrã. O nosso objectivo didáctico foi cumprido.

Hoje, neste número, oferecemos aos nossos leitores uma actualização do INSTALA. Possui vários melhoramentos, nomeadamente na interface (onde o rato já actua) e na capacidade de escolha dos directórios para instalação dos diferentes programas. Mas o aspecto mais notório prende-se com a possibilidade de apresentar imagens e textos a elas associados: um verdadeiro *slide-show*.

Sabemos que muitos dos nossos leitores sofrem do «bichinho» da impaciência e alguns nem se dão ao trabalho de ler esta página. Esses, certamente já viram a apresentação das imagens que escolhemos para introdução do novo INSTALA. Aos restantes sugerimos que introduzam a disquete no respectivo leitor (vulgo *drive*) e digitem INSTALA. Deliciem-se! Aos aficionados do *Assembly*, recomendamos a leitura do artigo de Carlos Ladeira e a consulta do código-fonte presente na disquete.

Renato Casquilho

Propriedade: Graficria, Publicidade e Artes Gráficas, Lda.
Director de Edição: Franco Gomes
Director: Renato Casquilho
Colaboradores: Afonso Infante, Carlos Batista, Carlos Ladeira, José Iria, José Orlando Pereira, Luis Guerra, Paulo Bernardo, Pedro Andrade, Raquel Ferreira, Ricardo Barbosa, Ricardo Santos, Ruben Barbosa, Tomé Pessegueiro e Tiago Pascoal.
Ilustrações: Leonor Gomes.
Relações Públicas e Publicidade: Fátima Ribeiro.
Assinaturas e encomendas: Guiomar Russo.
Redacção, Publicidade e Serviços Administrativos:
Rua Duarte Pacheco Pereira, 69-A - Damaia
2720 AMADORA - Telfs.: 474 60 11/41 - Fax: 475 22 89
Produção Gráfica: Graficria, Publicidade e Artes Gráficas, Lda.
Impressão: Empresa Litográfica do Sul, S.A.
Distribuição: Electroliber
Revista bimestral ■ Preço: 800\$00 ■ SRIP N.º 213900
Tiragem: 12 500 exemplares ■ Dep. Legal N.º: 28587/89

CONTEÚDO DA DISQUETA	2
DESENHO A 3 DIMENSÕES	3
INSTALA 2.0	5
Actualização do Programa INSTALA.EXE	
EMSTEST	7
Contole a Memória Expandida	
SOBGELO	8
Um Joguinho para o Verão	
UNIDADES	9
Conheça as «Unidades» do seu Sistema	
CORREIO DOS LEITORES	10
MÉTODOS NUMÉRICOS	12
A PROGRAMAÇÃO ORIENTADA POR OBJECTOS E O VISUAL BASIC (X)	13
NEWFUNCS.PRG	17
Novas Funções para Clipper	
EM BUSCA DO PC PERFEITO	18
Placas Mãe (II)	
USERLIB V2.0	21
Actualização da USERLIB V1.1	
WEIRD 2	22
WEIRD? Continua o Mistério!	
O BIOS E O DOS (II)	24
TRUQUES E DICAS PARA O WINDOWS (IV)	28
ANÁLISE E PROJECTO DE SISTEMAS	29
O processo mais importante da Informática (VII)	
NUCLEUS 1.0	32
Tabela Periódica em Pascal	
BATCH TOOLS (I)	33
As Ferramentas para Ficheiros Batch	
SHAREWARE (IX)	34
Crítica e Divulgação do <i>Shareware</i> Nacional	
REBOOT v1.1	35
Um Reboot com três paladares	
CDIR	36
Manipulação de Directórios	
LIVROS	37
MONTRA SPOOLER	38

3DVISUAL.ZIP

Bruno Possolo traz-nos o resultado da aplicação dos seus conhecimentos. Para além do estudo do código-fonte em C++, o leitor pode disfrutar de interessantes efeitos gráficos.

ASLPAINT.ZIP

Um excelente programa para pintura em Windows. Versão *Shareware*.

BIOS2.ZIP

«Quase todas as linguagens de programação de alto nível possuem comandos (procedimentos ou funções) para escrever mensagens no ecrã, ler ou escrever em memória de massa (disco), etc. Para que o computador consiga «perceber» os programas criados nessas linguagens, é necessário compilá-los, ou seja, transformá-los em código-máquina. Tomé Pesseguero inicia, com estas palavras, a Parte II da sua série.

BT1.ZIP

Luís Guerra vai iniciar uma série de artigos sobre a chamada «linguagem Batch». Fiquem com uma ideia do que ele nos traz: «Todos os utilitários divulgados neste e próximos artigos destinam-se a melhorar a apresentação e construção de *batch files*. E, apesar de serem pouco flexíveis, as *batch files* são indispensáveis para necessidades do dia-a-dia: são fáceis de alterar e para as criar basta um programa vulgar de edição de texto.»

CDIR.ZIP

«Baseado no QCD (Quick Change Directory) da CBLong que, apesar de ser bastante prático não permite mudar quer de *drive* quer para a raiz, no PDIR (Paul's Directory) de Paul Askew, que permite listar o conteúdo de um directório com uma cor diferente para cada tipo de ficheiro e no NCD (Norton Change Directory) da Symantec...». Com estas palavras, Pedro Sinogas pretende despertar o apetite do leitor para o seu CDIR.

CORREIO.ZIP

Os ficheiros que complementam as respostas dadas na rubrica «Correio do Leitor».

EMSTEST.ZIP

Ruben Barbosa revela-nos um interessante utilitário dedicado ao «mapeamento da memória». E começa assim: «Na actualidade, a palavra memória tem-se revelado como palavra-chave. Assumi esta característica devido ao seu elevado preço e à grande necessidade da sua utilização. Aplicações de grande envergadura, sentiram assim a necessidade de utilizar toda a memória aparentemente disponível nas máquinas.»

INSTALA2.ZIP

O INSTALA da disquete Spooler foi actualizado! Leiam o que Carlos Ladeira tem para nos comunicar a respeito das melhorias introduzidas.

METNUM3.ZIP

«Em muitas aplicações que criamos somos confrontados com a necessidade do uso de matrizes. Foi essa a razão que me levou a desenvolver este módulo». Ricardo Barbosa continua com a terceira parte da série «Métodos Numéricos».

NEWFUNC.ZIP

Hugo Meneses falou: «Toda a gente gosta dos menus bonitos dos programas da Central Point, como por exemplo o PC Tools; então resolvi criar este pequeno programa que tenta duplicar os tipos de menus desses programas para usar no Clipper 5.01».

NUCLEUS.ZIP

«Estava na aula de Física-Química (chamada física química) sem nada p'ra fazer, quando abri o livro e vi a tabela periódica. Pensei: «Até podia fazer um programa sobre Química». Passei das palavras à acção (neste caso dos pensamentos à acção) e dois meses e meio depois, tinha o Nucleus completo!», disse Isaac Cadaxa. E, passados alguns meses, ele aqui está para deleite dos leitores e programadores de Pascal.

REBOOT.ZIP

Carlos Jorge Santos oferece-nos um «reboot com três paladares». O artigo é de leitura obrigatória, dada a «delicadeza» da matéria.

SOBGELO.ZIP

«Caso não conheçam, o Sob-Gelo é aquele jogo existente em alguns parques de diversões (como por exemplo na Feira Popular). Cada pessoa possui um *paddle* com o qual empurra um disco tentando introduzi-lo na baliza adversária. Esse disco desliza sobre uma mesa que emite ar quente.» Jorge Morgado falou sobre o seu «joguinho para o verão».

UNIDADES.ZIP

«Quem nunca teve a curiosidade de saber quais as unidades existentes num computador?» O António José satisfaz-nos esta natural curiosidade.

USERLIB.ZIP

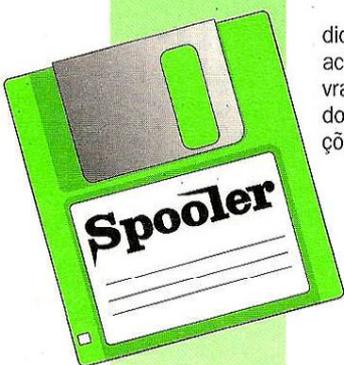
Nélio Abreu decidiu actualizar a sua anterior versão da USERLIB, publicada no Nº 31 da Spooler. Além da correcção de pequenos erros (vulgo *bugs*), também traz algo de novo.

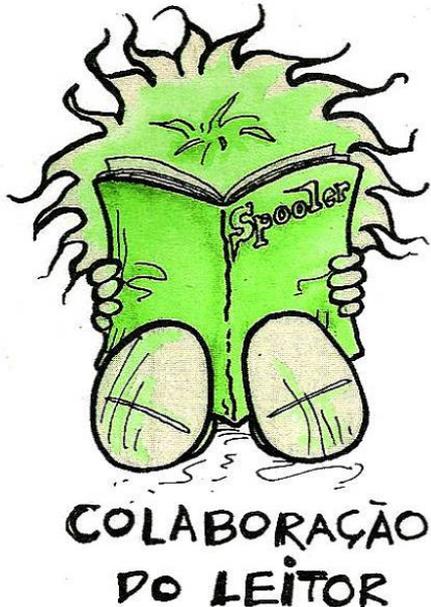
VIBASI10.ZIP

Com este décimo artigo, paulo Bernardo dá por terminada esta série dedicada à «Programação Orientada por Objectos», mais concretamente ao Visual Basic 3.0. Mas não suspirem de alívio... pois a versão 4.0 já por aí anda...

WEIRD2.ZIP

Ainda sem desvendar o mistério, Tiago Ribeiro continua: «Desta vez, após pesquisar os meus ancestrais arquivos *Top-Secret*, seleccionei sete funções que passo a explicar.» Bem, talvez seja para a próxima.





DESENHO A 3 DIMENSÕES

Linguagem: C++

por Bruno Shirley Possolo

Bruno Possolo traz-nos o resultado da aplicação dos seus conhecimentos. Para além do estudo do código-fonte em C++, o leitor pode disfrutar de interessantes efeitos gráficos.

3DVISUAL V1.0

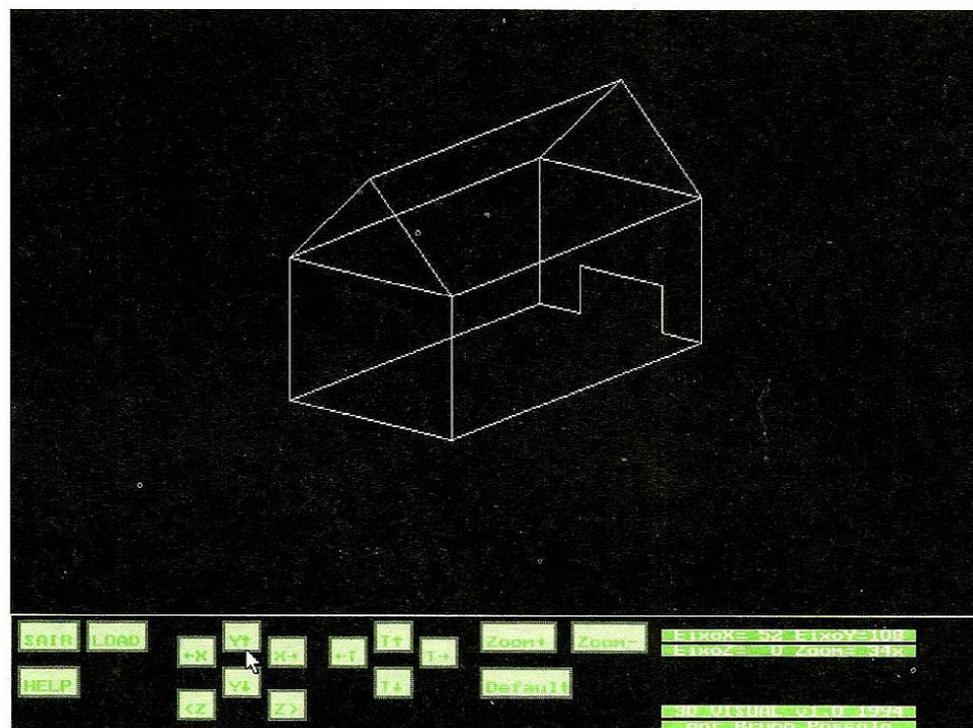
Programar em 3D requiere um vasto conhecimento de Trigonometria, Geometria Descritiva, Álgebra linear (especialmente a parte de multiplicação de matrizes) e, como é obvio, de Matemática em geral.

Mas não se assustem, porque não me vou pôr aqui a calcular todos os passos para chegar às matrizes tridimensionais. Os resultados dados serão já finais.

Os métodos de translação, escala e rotação de objectos em 3D são uma extensão dos métodos de 2D incluindo as considerações para a coordenada **z**. Por esta e outras razões, aconselho a quem queira aprofundar este assunto, que comece por estudar 2D e não passe, sem conhecer as bases, para o 3D.

TRANSLAÇÃO

Na representação de coordenadas homogêneas tridimensionais, um ponto é transladado de (x,y,z) para a posição (x',y',z') com o cálculo da seguinte matriz:



$$[x',y',z']=[x,y,z] \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ESCALA

A matriz operação para a escala em 3 dimensões relativa à coordenada da origem é:

$$[x',y',z']=[x,y,z] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

A matriz escala, relativa a um ponto arbitrário, não a vou escrever aqui visto que não a utilizo no programa.

ROTAÇÃO

E finalmente chegamos à rotação de imagens em 3D. Para especificarmos a transformação rotação de um objecto, temos que designar um eixo de rotação e o ângulo de rotação. Bom, não me vou pôr aqui com pormenores, porque para isso aconselho a quem tiver dinheiro, que compre um bom livro de computação gráfica em 3D, mas deixo o aviso que não contem com livros de qualidade abaixo dos 13 contos.

Rotação em torno do eixo **z** é definido pela matriz operação escrita a seguir, em que o parâmetro **A** será o ângulo de rotação.

$$[x'.y'.z']=[x.y.z] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos A & \sin A & 0 \\ 0 & -\sin A & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotação em torno do eixo **x** será:

$$[x'.y'.z']=[x.y.z] \begin{pmatrix} \cos A & 0 & -\sin A & 0 \\ 0 & 1 & 0 & 0 \\ \sin A & 0 & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

E finalmente a rotação em torno do eixo **y**:

$$[x'.y'.z']=[x.y.z] \begin{pmatrix} \cos A & \sin A & 0 & 0 \\ -\sin A & \cos A & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Vistas as bases da programação, passemos agora ao funcionamento dos programas MAKE e 3DVISUAL.

Conteúdo do pacote

- 1 - 3DVISUAL.EXE
- 2 - MAKE.EXE
- 3 - 3DVISUAL.CPP
- 4 - MAKE.CPP
- 5 - 3DBRUNO.H
- 6 - RATO.H
- 7 - CUBO.VIS
- 8 - OCTA.VIS
- 9 - CASA.VIS
- 10 - DODECA.VIS
- 11 - EGAVGA.OBJ

3DVISUAL

O 3DVISUAL é formado por um ecrã em que se vêem as transformações de imagem, e por um segundo ecrã que serve de controlador das transformações de imagens de 3D (ver FIG 1).

Todo o programa é controlado por rato, não havendo nenhum comando efectuado por teclado.

Este programa têm uma vantagem que na minha opinião é importante. Como usa um método de escrever

apenas uma linha de cada vez por ciclo, pode ler qualquer imagem feita com vértices e arestas, desde que não exceda a memória (esta pode ser aumentada no programa fonte, pelos interessados). Mas esta vantagem, como os que desenvolverem mais este assunto sabem, poderá trazer alguns dissabores, principalmente para o desenvolvimento de novas perspectivas e da remoção de linhas invisíveis.

PROBLEMAS

Se tiverem problemas ao correrem o programa eles poderão estar relacionados com:

1. O modo gráfico do seu monitor não aceitar EGA ou VGA.
2. Não ter o rato instalado.

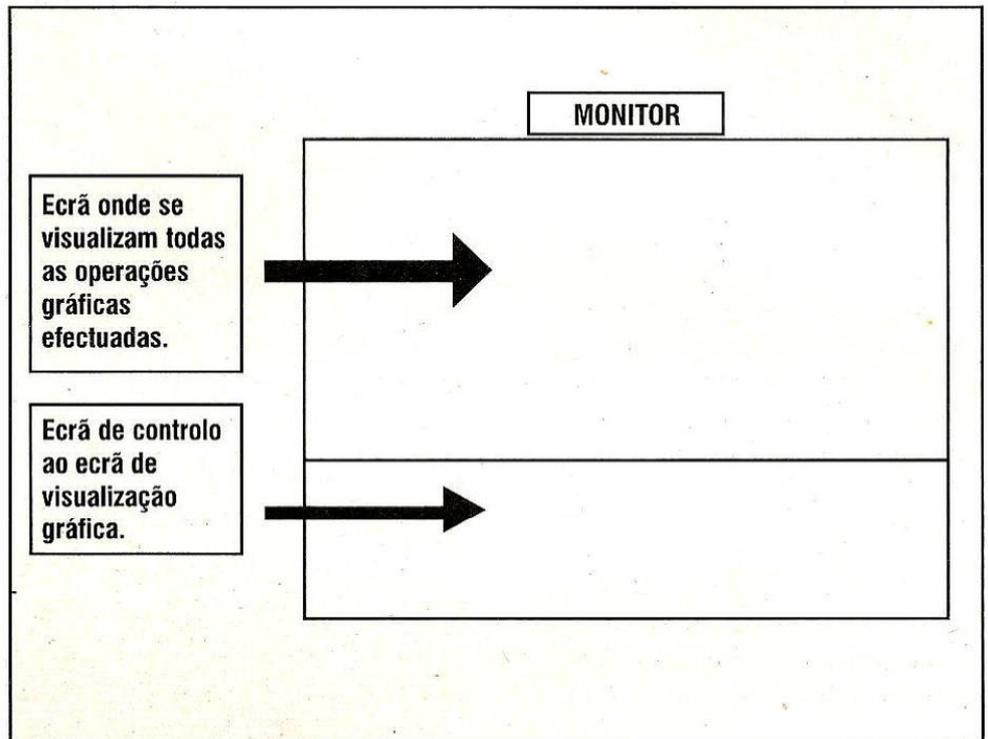


Figura 1

MAKE

MAKE é um programa que eu desenvolvi com o objectivo de criar novas imagens para correrem no 3DVISUAL. Aconselho às pessoas que quiserem criar novas imagens para o 3DVISUAL, para usarem ficheiros em extensão VIS, por exemplo AVIAO.VIS.

Antes de se utilizar este programa convém achar primeiro os vértices e as suas ligações num desenho em que se leve em conta os eixos **x y e z** (ver FIG 2).

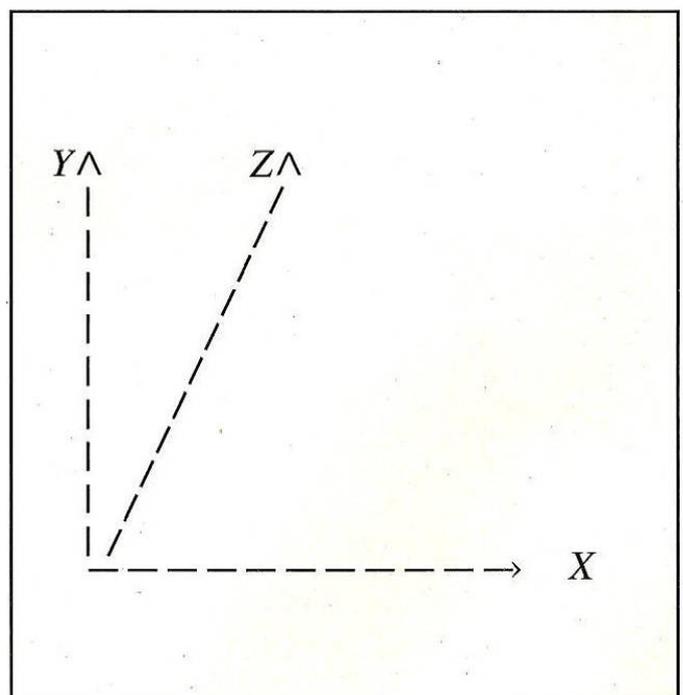
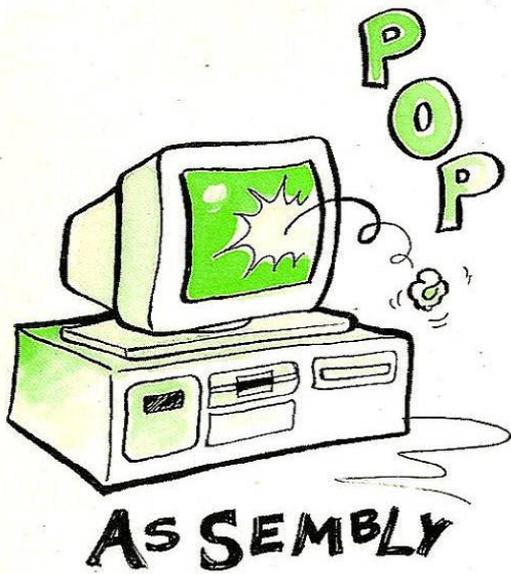


Figura 2



INSTALA 2.0

ACTUALIZAÇÃO DO PROGRAMA INSTALA.EXE

Linguagem: Assembly

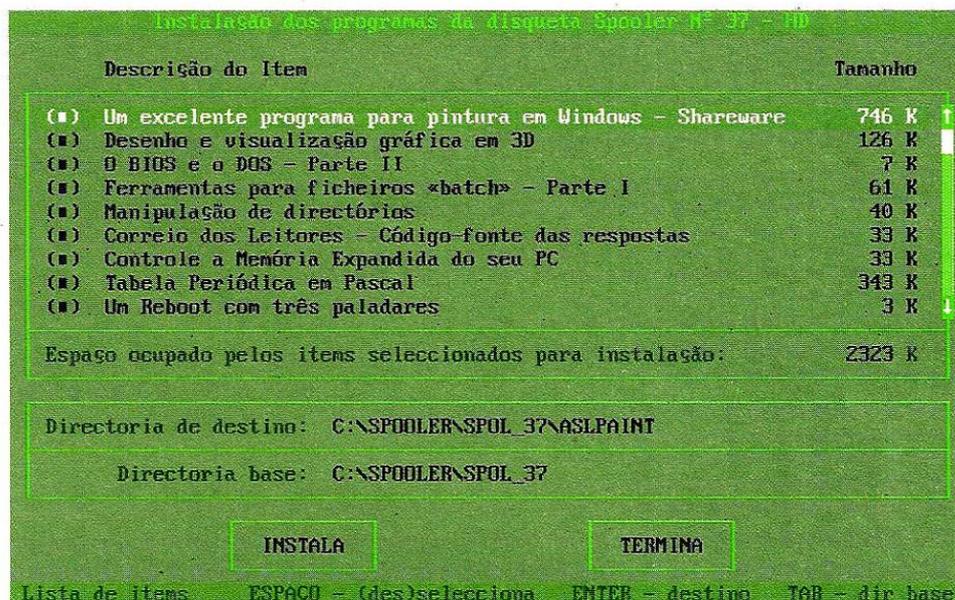
por Carlos Ladeira

O termo *upgrade* é hoje em dia sobejamente conhecido pela comunidade informática. É normalmente utilizado para referenciar o lançamento de uma actualização de *software*. Foi precisamente o que sucedeu com o programa INSTALA, que tem acompanhado a disquete dos últimos números da **Spooler**. Nesta nova versão destacam-se dois aspectos: as melhorias introduzidas na interface do utilizador e a implementação da possibilidade de fazer um *slide-show*.

Características

O programa destina-se a facilitar a tarefa de instalação de programas (ou outro tipo de dados) repartidos por várias partes. É especialmente indicado para programas ou dados de tamanho avantajado, pois tem como característica principal o armazenamento do «produto» a instalar de forma compactada. Para esta compactação foi utilizado o formato ZIP. Escolhi este formato por ser, sem qualquer dúvida, o mais divulgado e utilizado actualmente. Para além do que já foi referido, há ainda a acrescentar as seguintes características:

- Descrição das várias opções de instalação num ficheiro de texto;
- Tamanho reduzido, ocupando apenas 10Kb + 3Kb para o PKUNZJR;
- Simplicidade de operação com



- utilização do rato numa interface agradável;
- Capacidade para até 100 itens por instalação;
- Possibilidade de instalação apenas dos itens pretendidos pelo utilizador;
- Indicação permanente do espaço ocupado pelos itens seleccionados para instalação;
- Gestão cuidada da directoria de destino de cada item;
- Possibilidade de fazer um *slide-show* de imagens e texto antes da instalação.

Funcionamento

O funcionamento do programa

INSTALA é baseado num ficheiro de dados. Este, obrigatoriamente, deve chamar-se INSTALA.DAT e pode ser criado com qualquer editor de texto semelhante ao EDIT do MS-DOS. No conteúdo deste ficheiro de dados deve existir informação referente a: um título com o máximo de 80 caracteres, uma directoria **base** para a instalação também com o tamanho máximo de 80 caracteres e finalmente **n** grupos de quatro linhas referentes aos **n** itens a instalar. Todas as linhas em «branco» ou cujo primeiro carácter seja um «*», serão ignoradas, podendo ser utilizadas como linhas de comentário. Segue-se um pequeno exemplo:

- * Ficheiro de dados referente à instalação do programa ABC
- * criado em 09/04/95 por Carlos Ladeira
- * a linha que se segue contém o título (max. 80 caracteres)

Instalação do programa ABC

- * a linha que se segue contém a directoria base para a instalação
- C:\APLICS\ABC**
- * segue-se a descrição dos items
 - * cada item necessita de 4 linhas de informação:
 - * Descrição do item com o máximo 58 caracteres
 - * Nome do ficheiro ZIP que contém os «dados» deste item
 - * Directoria de destino para instalação do item
 - * Tamanho ocupado pelo item depois de descompactado, em KBytes

ABC – programa principal

ABC.ZIP

321

ABC – Dados adicionais

ABCDADOS.ZIP

DADOS

100

ABC – Tutor de ajuda

ABCTUTOR.ZIP

AJUDA

65

ABC – PIF's para o Windows

ABCPIFS.ZIP

C:\WINDOWS

12

De salientar que o 3º campo da descrição do item (directoria de destino) tem duas variantes. Se for constituído apenas pelo nome de uma directoria (sem *drive*), esta será criada «dentro» da directoria **base**. Se, pelo contrário, este campo contiver alguma referência a uma *drive*, a directoria base será ignorada. Para além do ficheiro INSTALA.DAT, é ainda necessária a presença do ficheiro PKUNZJR.COM, todos eles no mesmo local (directoria) onde se situa o programa INSTALA.EXE.

Criação de um *Slide-Show*

Com o intuito de permitir que seja feita alguma promoção ao produto distribuído ou ainda a outros, foi incluída a possibilidade de fazer um *slide-show*. Neste devem ser utilizadas imagens no formato **GIF 87a não interlaçado** com as dimensões máximas de 360x480, 256 cores. Juntamente com estas, podem ser mostrados textos com as dimensões máximas de: 45 caracteres por linha e 28 linhas. Para que seja possível fazer um *slide-show*, é necessário criar um ficheiro de texto com o nome INSTALA.SHW. Este contém em cada uma das suas linhas o nome (sem extensão) de um ficheiro GIF. Caso pretenda que seja mostrado algum texto sobreposto à imagem presente no ficheiro GIF, deverá criar um ficheiro com o mesmo nome, mas com a extensão TXT. O texto deste ficheiro será mostrado centrado horizontalmente e de cima para baixo. A apresentação deste texto no ecrã é feita depois da imagem ser escurecida cerca de 50%. Segue-se um pequeno exemplo:

```
LOGOTIPO
BBS_ABC
LIVRO
PROGS
```

INSTALA.INC	- Contém a definição de estruturas e de constantes utilizadas por todo o programa.
8X16FNT.INC	- Contém a definição binária de uma fonte portuguesa (8x16) com todos os caracteres ASCII entre o 33 e o 175.
STRING.ASM	- Rotinas várias para manipulação de <i>strings</i> ASCII.
FILES.ASM	- Rotinas para manipulação de ficheiros.
UTIL.ASM	- Rotinas várias de utilização geral.
VIDEO.ASM	- Rotinas várias para manipular registos da placa VGA. Na sua maioria, operações com a <i>palette</i> .
MOUSE.ASM	- Rotinas para tratamento dos vários eventos do rato.
UNZIP.ASM	- Rotinas utilizadas na descompressão dos ficheiros ZIP.
MENU.ASM	- <i>Loop</i> principal do programa.
SHOW.ASM	- Rotinas responsáveis pela execução do <i>slide-show</i> .
INPUT.ASM	- Rotinas de gestão e de tratamento dos <i>inputs</i> do utilizador.
MODE13X.ASM	- Rotinas para o modo de vídeo utilizado na apresentação do <i>slide-show</i> .
TEXTMODE.ASM	- Rotinas para escrita directa na memória de vídeo em modo de texto – 80 colunas.
GIF.ASM	- Rotinas para a descodificação de imagens GIF não interlaçadas, versão 87a.
LOADER.ASM	- Rotinas de leitura e interpretação do ficheiro INSTALA.DAT.
MAIN.ASM	- Inicialização de todo o programa.

Resta ainda salientar que, visto o modo de vídeo utilizado para o *slide-show* ser pouco vulgar (embora possível numa placa VGA ou SVGA), qualquer imagem que seja digitalizada para incluir neste programa deverá previamente ser tratada, nomea-

damente alterar as suas proporções de forma que esta fique mais estreita do que é na realidade.

Organização da disquete

Foram introduzidas algumas alterações à estrutura da disquete. Os ficheiros passam a estar distribuídos pelas seguintes directorias:

- os ficheiros INSTALA (EXE, DAT e SHW) e o ficheiro PKUNZJR.COM devem ser colocados na **raiz**;

- os ficheiros que contém o «produto» a instalar (com extensão *.ZIP) devem instalar-se na directoria \SPO-OLER;

- as imagens (quando existirem) e os respectivos textos (também facultativos) devem ser instalados na directoria \IMAGENS.

O código-fonte

Resta agora falar do código-fonte que foi feito integralmente em linguagem *Assembly*. Ao contrário da primeira versão do programa, este está distribuído por 16 ficheiros, um total de mais que 150Kb de código-fonte. Em cada ficheiro, encontra-se o código referente a cada uma das «partes» do programa. Segue-se uma descrição abreviada do conteúdo de cada um destes ficheiros:

Para além destes, é ainda incluído o ficheiro MAKEFILE que pode ser utilizado conjuntamente com o programa MAKE (distribuído juntamente com o Turbo Assembler) para criar o executável INSTALA.EXE.



PROGRAMAÇÃO EM C

Na actualidade, a palavra memória tem-se revelado como palavra-chave.

Assumi esta característica devido ao seu elevado preço e à grande necessidade da sua utilização. Aplicações de grande envergadura, sentiram assim a necessidade de utilizar toda a memória aparentemente disponível nas máquinas.

O primeiro passo foi dado com a construção dos *device-drivers*, embora a grande parte dos programadores/utilizadores não soubessem como os utilizar.

Decidi, então, apresentar algumas funções que permitem acessos às sub-funções dos *device-drivers*. Para isso, construí um módulo (EMS.*), e um programa de teste, fazendo chamadas às várias funções do mesmo. A sua utilização depende ainda da instalação dos *device-drivers* no CONFIG.SYS. Assim, caso não existam, devem ser acrescentadas as seguintes linhas:

```
DEVICE = C:\...\HIMEM.SYS  
DEVICE = C:\...\EMM386.EXE
```

O DOS, a partir da versão 6.0, optou por utilizar a memória expandida fundida com a estendida, embora esta última seja sempre em maior dimensão. Isto significa que ao reservar páginas de memória expandida estamos a diminuir a quantidade de memória estendida disponível.

EMSTEST CONTROLE A MEMÓRIA EXPANDIDA

Linguagem: C

por Ruben R. B. Barbosa

Mas como funciona o *device-driver*?

É extremamente simples a sua utilização. Apenas é necessário reservar páginas (e não blocos de memória de dimensão pretendida, como o *malloc*, *new*, etc).

A quantidade mínima que se pode reservar consiste numa página de dimensão igual a 16 KBytes.

Torna-se claro que não deve ser reservada uma página que use pelo menos metade desta quantidade. Para alguma coisa ainda serve a memória convencional.

O *device-driver* proporciona aos seus utilizadores um conjunto de páginas que permitem «mapear» informação (converter endereços lógicos em físicos).

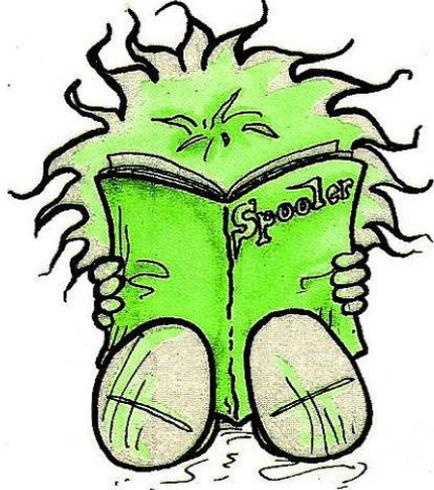
Esta zona começa no endereço base que é calculado através de uma interrupção.

O programa de demonstração (EMSTEST.C) reserva uma página e mapeia-a para memória convencional, escrevendo um conjunto de valores. Posteriormente verifica se estes são os que realmente foram escritos, acabando por libertar a respectiva página. É feita, também, uma descrição do conteúdo da memória expandida (lado direito do ecrã), somente até ao fundo. Isto significa que se existirem mais blocos do que os podem ser visionados no ecrã, estes não serão escritos. No entanto, no início desta janela podemos ver

qual o número total de blocos reservados. A tecla chave para continuar o teste é a barra de espaços e qualquer erro que surja, durante a execução do programa, dá lugar à escrita, na terceira linha do ecrã, da respectiva mensagem de erro.

Atenção ao Windows: ao correr este programa, através do MS-DOS Prompt, a quantidade de memória expandida disponível aumenta consideravelmente. Será que a memória estendida é que fica prejudicada? Parece-me que sim.

Se o leitor considerou interessante este artigo, então tomo a liberdade de sugerir que dê uma vista de olhos na «Montra Spooler», onde se encontra anunciado o meu livro «C - Avançado», editado pela Graficria e que no décimo capítulo foca a utilização da memória expandida e estendida.



COLABORAÇÃO DO LEITOR

Este é um jogo para as placas VGA (modo MCGA – *Multi Color Graphics Adapter*). Foi escrito em linguagem C, já há alguns meses, mas só agora resolvi enviá-lo.

Gostaria de oferecer alguns créditos ao João António Zigue pelo seu jogo «Tutti-Fruti» publicado na **Spooler** Nº 25 – foi ele quem me inspirou e introduzi na programação da placa VGA, e já agora fiquem a saber que o seu jogo possui boas ideias (algumas das quais utilizadas por mim no Sob-Gelo).

Caso não conheçam, o Sob-Gelo é aquele jogo existente em alguns parques de diversões (como por exemplo na Feira Popular). Cada pessoa possui um *paddle* com o qual empurra um disco tentando introduzi-lo na baliza adversária. Esse disco desliza sobre uma mesa que emite ar quente.

O Sob-Gelo pode ser jogado contra o computador, entre duas pessoas no mesmo PC, ou ainda entre duas pessoas cada uma no seu PC. Neste último caso, a comunicação é feita através da COM 2 (porta série RS232).

Comunicação via RS232

Se quiserem construir o cabo de transmissão deverão comprar duas fichas RS232 fêmeas de 25 pinos, assim como um cabo coaxial de 9 condutores. Agora começa a parte mais delicada: a soldadura.

Quando abrirem as fichas RS232

SOBGELO UM JOGUINHO PARA O VERÃO

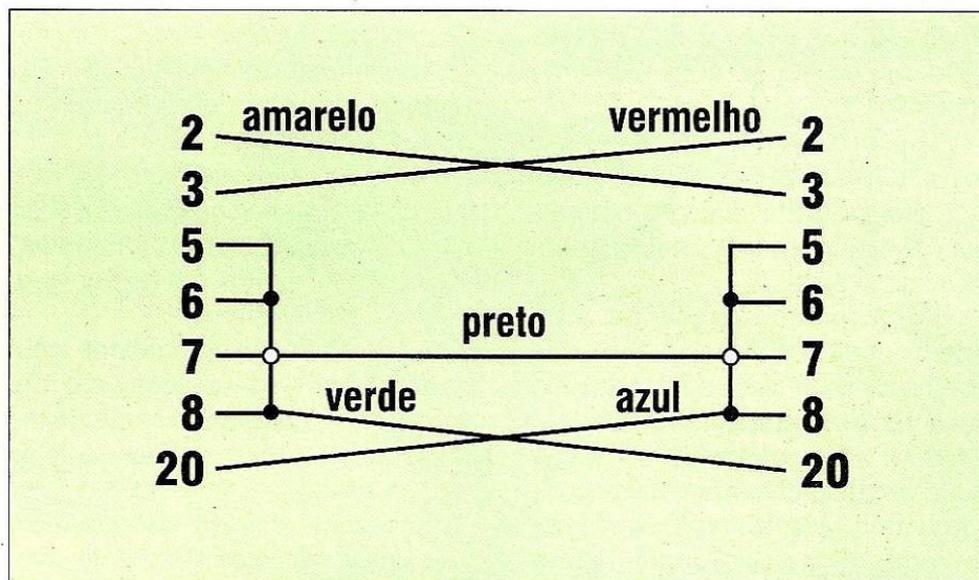
Linguagem: C

por Jorge Alexandre Morgado

irão reparar que os vários pinos estão numerados. Do mesmo modo cada um dos 9 condutores tem uma cor diferente. Isto evitará que nos enganemos na construção do cabo. Observem no seguinte esquema como efectuar as ligações:

que quando estão a jogar duas pessoas em dois computadores, o *paddle* superior não é exibido, o que significa que cada jogador movimentará o *paddle* inferior.

Ainda no que diz respeito à transmissão de dados, convém talvez ela-



Os números representam os pinos de cada ficha RS232 – somente 7 dos 25 pinos são realmente utilizados. Da mesma forma, as cores representam os possíveis condutores a utilizar na ligação – também neste caso, somente 7 dos 9 condutores do cabo coaxial são necessários. Finalmente, reparem que em cada ficha os pinos 5 e 6 deverão ser ligados ao pino 8.

Transmissão de dados

Tenham em atenção o facto de

borar um «mapa» dos dados enviados/recebidos. Então aqui vai:

[0..5] – Se o receptor receber um valor entre 0 e 5, ele sabe que o emissor lhe está a informar qual a direcção que a bola vai tomar no próximo deslocamento. Note-se que o receptor deve então transformar a direcção recebida visto que neste caso a bola vai descer, isto é, a direcção deve situar-se no intervalo [6..11];

[21..28] – Quando qualquer um do PC recebe um valor dentro deste intervalo, ambos sabem que estão a

receber a velocidade (*delay*) do jogo. O valor do *delay* situa-se entre 1 e 8, mas quando é enviado o emissor adiciona-lhe 20, de forma a que não seja confundido com as direcções da bola. Como tal, o receptor deve decrementar 20 e só então considerar o valor resultante;

30 SYN – este é o valor do *Synchronous Idle*. É enviado sucessivamente pelo PC emissor quando não tem mais dados para transmitir. Por seu lado, o PC receptor entende-o da mesma forma, ou seja, quando recebe o valor 30 (SYN) sabe que não há mais dados para receber, logo houve entendimento;

40 BALL_LOSTED – este valor é enviado por qualquer um dos PC quando a bola é perdida. Por exemplo, quando um PC recebe o valor 40 (BALL_LOSTED) ele sabe que o outro PC não defendeu a bola;

[51..99] – Os valores situados neste intervalo são enviados por ambos os PC no início do jogo para determinar de quem sai a bola. Na realidade, é gerado um número ale-

atório entre 1 e 49, mas somamos-lhe 50 para que não seja confundido com outros dados transmitidos. Por sua vez, o receptor executa a tarefa inversa, isto é, subtrai-lhe 50. Note que a bola sai do PC que gerar o menor número. Em caso de igualdade ambos os PC repetem o procedimento.

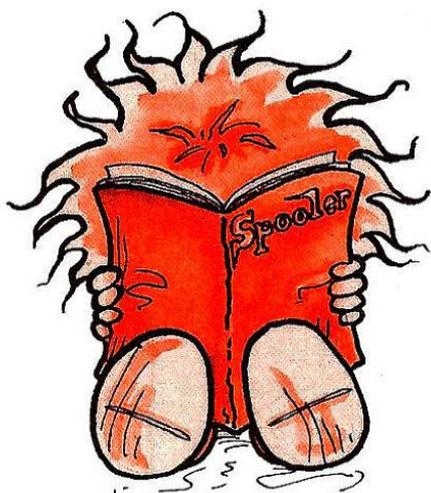
Teclas utilizadas

- No menu inicial utilizem as teclas UP, DOWN e ENTER;
- Para seleccionarem a dificuldade e velocidade utilizem as teclas UP, DOWN, LEFT, RIGHT e ENTER;
- Durante o jogo as teclas são LEFT, RIGHT e SPACE para lançar a bola. No caso de dois jogadores no mesmo PC, utilizem ainda as teclas Z e C para movimentar o *paddle* superior;
- Para escrever o nome no quadro das pontuações utilizem as teclas UP, DOWN, LEFT, RIGHT e ENTER ou SPACE.

Sobre o Programa

O código fonte está todo comentado, pelo que não me parece difícil percebê-lo depois de algum estudo. Os ficheiros *.BIN contêm as imagens do jogo em formato binário e o ficheiro SCORES.SG está em formato texto e serve para guardar a tabela das pontuações.

O executável SOB-GELO.EXE foi obtido através da compilação e posterior «linkagem» dos ficheiros *.C e *.H, utilizando o modelo de memória COMPACT. Note que a biblioteca GRAPHICS.H não deve ser incluída.



COLABORAÇÃO DO LEITOR

Quem nunca teve a curiosidade de saber quais as unidades existentes num computador? De facto, para além daquelas já habituais que todos nós conhecemos, também podem existir outras com letras pouco usuais.

Mas atenção, todas as letras que

UNIDADES CONHEÇA AS «UNIDADES» DO SEU SISTEMA

Linguagem: Pascal

por António José Monteiro Pereira

representam as ditas unidades podem não ser, obrigatoriamente, seguidas. Por exemplo, o facto de não existir o «D» não significa que não exista o «E».

Este utilitário vem apresentar ao utilizador todas as unidades activas na sua máquina (ou noutra qualquer,

claro) e, também, o número total das mesmas.

Modo de utilização: UNIDADES [/?]



CORREIO DOS LEITORES

CORREIO DOS LEITORES

MENSAGENS

Para Ricardo Peixoto, de Barcelos: Pedro Alexandre Nunes, autor do Master Mind publicado no Nº 32, pede que lhe volte a enviar as suas questões.

Programadores de Turbo Pascal: querem juntar-se num projecto a nível nacional tendo como objectivo a criação de programas e utilitários? Escrevam para a morada abaixo pedindo a vossa admissão no recente clube de programação, indicando nome, morada, telefone, data de nascimento, computador que possuem e a vossa auto-avaliação em termos de programação (de 0 a 20).

PProg
Av. Vitorino Nemésio, Lote 1
7300 PORTALEGRE

QUESTÕES TÉCNICAS

O leitor Luis Paulo Costa, do Porto, coloca várias perguntas sobre programação em C, respondidas por Carlos Ladeira:

Como incluir os drivers BGI nos executáveis ?

A inclusão quer dos *drivers* gráficos BGI quer das fontes CHR exige a execução de vários passos. Seguidamente vou dar o exemplo da inclusão do driver EGAVGA.BGI e da fonte TRIP.CHR num pequeno programa:

1º – É necessário converter o *drivers* e a fonte para o formato OBJ. Para tal é necessário recorrer ao utilitário BGI0BJ.EXE distribuído juntamente com o Turbo C, da seguinte forma:

```
C:\TURBOC\BGI> bgiobj egavga
BGI to OBJ Converter Version 4.0 Copyright (c) 1987, 1992 Borland
International
```

```
5554 bytes from 'EGAVGA.BGI' converted into 'EGAVGA.OBJ',
public name = '_EGAVGA_driver'.
```

```
C:\TURBOC\BGI> bgiobj trip
BGI to OBJ Converter Version 4.0 Copyright (c) 1987, 1992 Borland
International
```

```
16677 bytes from 'TRIPCHR' converted into 'TRIPOBJ',
public name = '_triplex_font'.
```

Deve-se ter em atenção os nomes pelo qual o *driver* e a fonte serão «conhecidos» no OBJ: **EGAVGA_driver** e **triplex_font**. Estes são de extrema importância para que depois o *driver* e a fonte sejam identificados no programa.

2º – Criação de um ficheiro de projecto que, para além do ficheiro com código fonte, deverá ainda incluir os ficheiros OBJ referentes ao *driver* e às fontes. Por exemplo:

```
EGAVGA.OBJ
TRIPOBJ
TESTEGFX.C
```

3º – Finalmente, resta a parte referente à programação. O reconhecimento do *driver* e da fonte é feito respectivamente através das funções **registerbgi_driver** e **registerbgi_font**. O parâmetro aceite por estas consiste nos nomes indicados pelo BGI0BJ durante a conversão. **EGAVGA_driver** para o *driver* e **triplex_font** para a fonte. Caso o valor de retorno de uma destas duas funções seja menor que 0, é porque ocorreu um erro e o programa deve ser abortado. A listagem do ficheiro encontra-se na disquete com o nome de TESTEGFX.C. Tal como foi feita a conversão do *driver* EGAVGA e da fonte TRIP, poderão ser feitos de todos os outros. É possível incluir mais que um *driver* e mais que uma fonte num mesmo programa.

Como controlar os erros críticos que podem ocorrer durante o acesso a uma drive não preparada?

É habitual, por distração, acontecer tentativas de acesso a uma *drive* não preparada ou

ainda tentativas de escrita numa *drive* protegida. Quando uma destas situações sucede, o DOS escreve no ecrã uma mensagem bem ao seu estilo. Que claro, nem sempre é vista de bom grado pelo programador que está a desenvolver o programa com uma interface «toda bonita». O ideal seria que o programador pudesse controlar este tipo de erros. E foi a pensar nessa situação que a Borland, ao criar o Turbo C, incluiu funções destinadas a tratar estes erros. São ao todo três funções, que passo a descrever:

void harderr (int (*handler)()) – Define um *handler* que vai ser chamado sempre que ocorrer um erro crítico. A estrutura deste *handler* será vista mais adiante. Este *handler* é uma função do nosso programa, cujo nome é passado como parâmetro na chamada a esta função.

void hardresume (int axret) – Retorna para o DOS o valor **axret**. Esta função deve ser chamada só pelo *handler*. O parâmetro **axret** pode conter os seguintes valores: 0 – Ignora o erro, 1 – tenta novamente, 2 – aborta para o DOS ou 3 – falha a operação.

void hardretn (int retn) – Retoma directamente para o programa o valor **retn**. Deve ser chamada pelo *handler*.

Resta agora falar do *handler* que deve ter a seguinte ordem de parâmetros:

int handler (int errval, int ax, int bp, int si)
errval corresponde ao código

de erro colocado no registo DI pelo DOS.

ax indica o tipo de erro: disco/*drive* se tiver um valor positivo correspondendo à *drive* onde se deu o erro (0 = A, 1 = B, etc.), *device* se tiver um valor negativo.

bp e **si** constituem um ponteiro *far* (bp:si) para o *header* do *device-driver* onde ocorreu o erro.

Caso as funções **hardresume** ou **hardretn** não sejam utilizadas, esta função deve retornar um dos seguintes valores: 0 – ignora, 1 – tenta novamente ou 2 – aborta. Dentro desta função só estão disponíveis os serviços do DOS entre 1 e 12, todos os outros têm consequências inesperadas. O que quer dizer que não podem ser feitas chamadas às funções de I/O da *standard library*. Um exemplo que demonstra tudo o que até aqui foi dito encontra-se na disquete com o nome HARDERR.C.

Quantas páginas tem o modo gráfico VGA 640x480 x16?

Este modo de vídeo do *driver* EGAVGA permite apenas uma página. No entanto, para mais pormenores acerca deste e dos outros *drivers*, consultar a tabela que se segue:

Drivers	Modos gráficos	Valor	Dimensões	Palette	Páginas
CGA	CGAC0	0	320 x 200	C0	1
	CGAC1	1	320 x 200	C1	1
	CGAC2	2	320 x 200	C2	1
	CGAC3	3	320 x 200	C3	1
	CGAHI	4	640 x 200	2 cor	1
MCGA	MCGAC0	0	320 x 200	C0	1
	MCGAC1	1	320 x 200	C1	1
	MCGAC2	2	320 x 200	C2	1
	MCGAC3	3	320 x 200	C3	1
	MCGAMED	4	640 x 200	2 cores	1
	MCGAHI	5	640 x 480	2 cores	1
EGA	EGALO	0	640 x 200	16 cores	4
	EGAHI	1	640 x 350	16 cores	2

EGA64	EGA64LO	0	640 x 200	16 cores	1
	EGA64HI	1	640 x 350	4 cores	1
EGA-MONO	EGAMONOH1	3	640 x 350	2 cores	1*
	EGAMONOH1	3	640 x 350	2 cores	2**
HERC	HERCMONOH1	0	720 x 348	2 cores	2
ATT400	ATT400C0	0	320 x 200	C0	1
	ATT400C1	1	320 x 200	C1	1
	ATT400C2	2	320 x 200	C2	1
	ATT400C3	3	320 x 200	C3	1
	ATT400MED	4	640 x 200	2 cores	1
	ATT400HI	5	640 x 400	2 cores	1
VGA	VGALO	0	640 x 200	16 cores	2
	VGAMED	1	640 x 350	16 cores	2
	VGAHI	2	640 x 480	16 cores	1
PC3270	PC3270HI	0	720 x 350	2 cores	1
IBM8514	IBM8514HI	1	1024 x 760	256 cores	
	IBM8514LO	0	640 x 480	256 cores	

* 64K numa placa EGA monocromática

** 256K numa placa EGA monocromática

O nosso colaborador José Fortunato Tomás traz-nos a sua participação a esta rubrica:

Há alguns dias atrás estava a ler as *Spooler* que tinha, pois andava à procura de uns textos até que por coincidência encontrei na rubrica «Correio dos leitores» um comentário que propunha a quem quisesse que poderia responder de uma forma mais completa ao leitor Hugo Filipe Cruz, que tá resposta seria publicada nessa mesma rubrica.

O assunto que tratava a pergunta do leitor tinha a ver com a possibilidade de fazer numa impressora matricial uma impressão *postscript*. Tal como era dito, tornar uma impressora matricial comum numa impressora *postscript* é coisa difícil se não impossível. Na verdade é bem difícil pois os comandos de cada impressora nem sempre respeitam as normas e as engrenagens dessas mesmas impressoras quase nunca estão preparadas para uma exactidão de movimentos de papel. No entanto eu fiz em C e PASCAL as rotinas para que qualquer um compreenda como se poderá simular o *postscript* numa impressora matricial comum.

Estas rotinas funcionam em princípio em todas as impressoras do tipo «proprinter» não funcionando à partida para impressoras matriciais de jacto de tinta. Penso que também é possível e com melhor resultado nas impressoras a jacto de tinta pois têm habitualmente boas engrenagens só que é necessário por parte dos leitores uma pequena investigação em busca dos comandos dessas impressoras.

O funcionamento da rotinas é bastante fácil de compreender,

no entanto aqui vai uma breve explicação:

Todos os comandos das impressoras tem quase sempre o caracter 27 como prefixo e de seguida um código de valores que dizem qual o comando. No meio dos comandos ditos *standart* há os «mais» *standart*. Para este caso usei o comando que faz a impressora regressar a folha de papel até ao ponto que ela considera o início da página e a partir daí dou os *Line Feeds* e espaços necessários para fazer o Gotoxy.

Penso que com este texto introdutório e resumido e com umas pequenas investigações nas rotinas, irão perceber por completo o funcionamento. No entanto chamo a atenção para o facto das impressoras terem *buffers* e perioridades de comandos (daí o modo 1 e modo 2; ver rotinas), portanto não se espantem de pensarem que já imprimiram tudo e quando carregam na tecla de *On Line* a impressora espetar uma quantidade enorme de caracteres no papel. Na disquete encontra-se o código-fonte, com os nomes de *PRINTER.C/PAS* e *TEST.C*.

Paulo Bernardo responde a Carlos Ribeiro, da Batalha:

O nosso leitor tem um problema relacionado com um programa em Pascal, onde o segmento de dados do programa excede 64k. Para poder usar mais do que 64k para variáveis, a solução passa por criar e usar variáveis cuja memória seja atribuída dinamicamente, estando então apenas limitadas pela memória convencional disponível no sistema quando o programa

estiver a correr (cada bloco de memória atribuído terá, no máximo, 64k, mas podem ser atribuídos tantos blocos quanto o permitido pela memória livre). Será necessário usar *pointers* e o procedimento *NEW* para atribuir a memória, ficando o *pointer* a apontar a zona de memória cedida; quando já não for necessária, a memória ocupada pela estrutura pode ser libertada com o procedimento *DISPOSE*. Apresentamos agora um pequeno programa que demonstra como se declara um *pointer*, atribui memória, afecta a estrutura e liberta (O programa segue na disquete com o nome *CARRIB.PAS*).

No nosso exemplo usa-se a função *MAXAVAIL*, que devolve o número de bytes do maior bloco disponível para atribuição dinâmica, e a função *MEMAVAIL*, que devolve a quantidade total de memória disponível para atribuição. No entanto, não há garantia que a quantidade de memória devolvida pela função *MEMAVAIL* se encontre num único bloco; quando é pedida memória ao sistema através da função *NEW*, apenas pode ser devolvido um bloco contínuo.

Um nosso leitor colocou-nos algumas questões sobre programação em Visual Basic. Eis as respostas de Paulo Bernardo:

«...será que é possível ler e transportar para uma variável as características de ficheiros situados em qualquer drive e directório, nomeadamente designação, extensão, dimensão, atributos, ...»

```
Global Const MF_BYPOSITION = &H400 'configura o acesso ao
item de menu pela posição SystemMenu%=getSystemMenu(hWnd,0)
'devolve um handle para o menu
Res%=removeMenu(SystemMenu%,6,MF_BYPOSITION) 'remove
o 6º item
Res%=removeMenu(SystemMenu%,6,MF_BYPOSITION) 'remove
também o separador (que é agora o 6º item)
```

É possível efectuar todas estas tarefas, não com uma única função mas recorrendo a um conjunto de funções, que passamos a indicar:

GETATTR(ficheiro\$) – devolve um código numérico onde estão representados os atributos do ficheiro especificado como argumento; *FILELEN(ficheiro\$)* – devolve o número de bytes do ficheiro; *FILEDATETIME(ficheiro\$)* – devolve uma *string* com a indicação da data e hora em que o

ficheiro foi criado. Todas estas funções recebem como argumento o nome do ficheiro sobre o qual vão operar. Para recolher os nomes de ficheiros existentes numa directoria o Visual Basic disponibiliza a função *DIR\$(especificação\$)*: se o argumento especificação contiver *wildcards* (* ou ?), a função *DIR\$* devolve o primeiro ficheiro da directoria que satisfaça a condição (devolve só o nome de ficheiro, não a directoria); sucessivas novas chamadas à função *DIR\$* sem argumentos devolverão os próximos ficheiros; quando não houver mais ficheiros, a função *DIR* devolverá uma *string* vazia.

«...em Visual Basic é possível alterar os comandos da Caixa de Controlo das aplicações?»

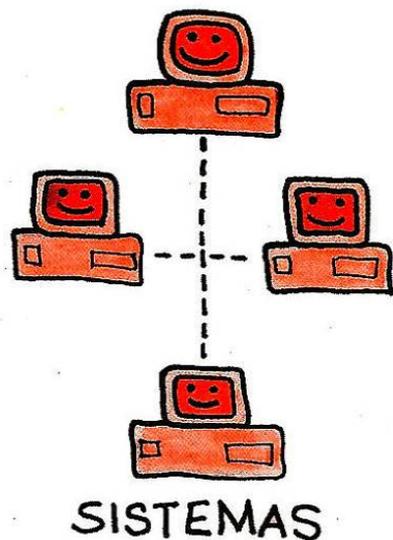
A resposta é não e sim. Não é possível alterar directamente por manipulação das propriedades dos forms; é possível se efectuarmos uma chamada às funções da API do Windows.

A API do Windows contém funções que permitem manipular todos os elementos constituintes de uma aplicação de Windows. As funções que nos interessam são *GETSYSTEMMENU* e *REMOVEMENU*. Para aceder a estas funções é necessário declará-las na secção general de um module: *Declare function GetSystemMenu Lib "User" (byval hWnd as integer, byval bRevert as integer) as integer; Declare function RemoveMenu Lib "User" (byval hMenu as integer, byval nPosition as integer, byval wFlags as integer) as integer*. Por exemplo, para remover a opção *CLOSE* do menu usar-se-ia o seguinte código:

Para uma explicação detalhada destas (e outras funções) os programadores poderão consultar o ficheiro *VB\WINAPI\WIN31WH.HLP* com descrição de todas as funções da API do Windows.

Existem outras funções interessantes da API do Windows relativas à manipulação de menus que também poderão interessar: *GETMENUID* e *INSERTMENU*.





MÉTODOS NUMÉRICOS

Operações com matrizes (Parte III)

por Ricardo M. B. Barbosa

Em muitas aplicações que criamos somos confrontados com a necessidade do uso de matrizes. Foi essa a razão que me levou a desenvolver este módulo.

As matrizes são de fácil tratamento computacional, se pensarmos que podem ser armazenadas em *array* bidimensional. Além desta representação convém saber por quantas linhas e colunas é formada a nossa matriz. Esta é a base da definição de matriz apresentada no ficheiro MATRIX.H. Aqui encontram-se também todas as funções visíveis ao utilizador (protótipos).

O ficheiro MATRIX.C contém a implementação das funções e está bastante comentado. Penso que não trará problemas ao leitor mais interessado neste tema.

Como demonstração do seu uso, implementei ainda um programa de teste (DEMO.C) ao módulo, fazendo este várias chamadas a todas as funções.

Um caso mais particular de implementação é o cálculo do determinante. Esta função faz uso dum *Gauss()*, que foi apresentada no número anterior da série «Métodos Numéricos». Assim sendo, não há motivo para explicar outra vez o seu funcionamento.

Um sistema diz-se mal condicionado se o seu determinante for um valor próximo de zero. Isto significa que se introduzirmos pequenas perturbações nos coeficientes da ma-



triz (ex: somar 0.0001 a um coeficiente), o resultado do sistema após se proceder à eliminação de Gauss, é bastante diferente do original (sem perturbação).

Para compilar estes ficheiros, basta apenas carregar o ficheiro DEMO.PRJ e seleccionar a opção

Build all. É ainda necessário activar a opção de *floating point: emulation* e de preferência, escolher o maior tamanho no modelo de memória (*huge*).



VISUAL BASIC

A PROGRAMAÇÃO ORIENTADA POR OBJECTOS E O VISUAL BASIC

Linguagem: Visual Basic (Parte X)

por Paulo Bernardo

Olá, caros leitores!

Com este décimo artigo terminamos a abordagem do Visual Basic como linguagem de carácter generalista para criação de aplicações em ambiente Windows.

Para encerrar de uma forma lógica esta série de artigos, abordaremos aqui a utilização de um conjunto variado de *statements* da linguagem e de controlos, que assim fecham o lote que considerámos fundamental para uma iniciação ao Visual Basic.

Ciclos de repetição

Os ciclos de repetição permitem a execução de uma instrução (ou conjunto de instruções) várias vezes consecutivas. O Visual Basic dispõe de três possibilidades nesta área: o ciclo incondicional FOR e os ciclos condicionais DO...LOOP e DO...WHILE.

forma decrescente, no caso de se utilizar um **step** negativo (neste caso, o **valor_final** deverá ser inferior ao **valor_inicial**. Se for executada uma instrução EXIT FOR dentro do ciclo FOR, a contagem é imediatamente interrompida e o programa «salta» para a instrução colocada a seguir ao NEXT.

```
Exemplo:
FOR I%=100 TO 1000
  PRINT I%
NEXT I%
```

Neste ciclo, a variável I% toma valores entre 100 e 1000 e, para cada um deles, executa a instrução PRINT, que vai escrever o valor da variável na janela.

Nos ciclos FOR têm de ser respeitadas algumas regras, das quais as mais importantes são as seguintes:

- as variáveis de controlo do ciclo têm de ser de tipo numérico;
- os limites inferior e superior do ciclo devem ser compatíveis com o tipo da variável de controlo do ciclo.

Instrução DO...LOOP
Sintaxe1: DO [{WHILE UNTIL} condição] bloco1 [EXIT DO] bloco2 LOOP
Sintaxe 2: DO bloco1 [EXIT DO] bloco2 LOOP [{WHILE UNTIL} condição]

A instrução DO...LOOP serve para implementar um ciclo de repetição condicional, i.e., onde o número de vezes que as instruções dentro do ciclo se executarão não é conhecido à partida mas depende de uma condição que, a seu tempo, fará terminar o ciclo.

A diferença entre as sintaxes 1 e 2 tem a ver apenas com a posição do teste da condição: no primeiro caso verifica-se que o bloco de acções dentro do ciclo pode nunca ser executado se a condição assim o determinar, pois a condição é testada ANTES da entrada no ciclo; no segundo caso, mesmo que a condi-

Instrução FOR...NEXT
Sintaxe: FOR <variável>= <valor_inicial> TO <valor_final> [STEP <salto>] bloco1 [EXIT FOR] bloco2 NEXT [<variável>]

No ciclo FOR, a variável de controlo do ciclo toma sucessivamente todos os valores entre o valor inicial e o valor final (inclusive). O ciclo pode ser executado de forma crescente, se for usado um **step** positivo, ou de

- não é boa estratégia afectar a variável de controlo do ciclo com qualquer valor, pois isso tornará o programa mais difícil de interpretar e corrigir (em caso de erro);

ção determine o fim do ciclo, o bloco de acções executou-se pelo menos uma vez.

A utilização da palavra WHILE ou UNTIL antecedendo a condição define implicitamente qual o valor lógico da condição que terminará o ciclo:

- se for usado WHILE, o ciclo efectuar-se-á ENQUANTO a avaliação da condição produzir o resultado lógico TRUE;
- se for usado UNTIL, o ciclo efectuar-se-á ATÉ a avaliação da condição produzir o resultado lógico TRUE (ou seja, enquanto for FALSE).

Uma última nota sobre os ciclos de repetição condicionais: o Visual Basic dispõe da instrução EXIT DO que o permite interromper imediatamente.

Controlo de Fluxo

O controlo de um programa é definido por instruções que testam condições e seleccionam para execução um ou outro bloco de instruções em alternativa. O Visual Basic coloca à disposição dos programadores as instruções IF e SELECT CASE para controlo de fluxo. A instrução IF já foi abordada no 4º artigo desta série; vamos agora tratar do SELECT CASE.

Instrução SELECT CASE
Sintaxe:
Select Case expressão
[Case valor1
[bloco1]]
[Case valor2
[bloco2]]
[...]
[Case Else
[blocoN]]
End Select

A instrução SELECT CASE possui uma **expressão** que é avaliada e uma lista de alternativas, das quais apenas uma será escolhida, em função do resultado. Se a expressão tiver o **valor1**, são executadas APENAS as instruções do **bloco1**; se a expressão tiver o **valor2**, são executadas APENAS as instruções do **bloco2**; etc. Se a expressão produzir um valor que não esteja previsto na lista,

será efectuada a acção associada ao ramo CASE ELSE.

```
Exemplo:
SELECT CASE A
CASE 0
PRINT «ZERO»
CASE 1
PRINT «UM»
CASE ELSE
PRINT «NÃO É ZERO NEM UM»
END IF
```

Existe ainda uma outra instrução que permite enviar o programa para uma localização determinada: é a instrução GOTO <label>. A *label* (etiqueta) tem de ser definida algures no procedimento, sendo constituída por um identificador válido seguido por dois pontos (:). Esta não é considerada uma verdadeira instrução de controlo de fluxo, porque não permite decisão, tratando-se apenas de um salto incondicional (mas que altera a sequência normal do programa!).

```
Exemplo:
IF A=0 THEN GOTO NULO
PRINT «DIFERENTE DE ZERO»
GOTO FIM
NULO:
PRINT «NULO»
FIM:
```

Regra geral, a instrução GOTO é usada para interromper o curso normal do programa e enviá-lo para um local pré-determinado (por exemplo, devido à ocorrência de um erro).

A função INPUTBOX

Como já foi sobejamente referido, em Visual Basic, todo o interface com o utilizador é feito através de controlos. Alguns controlos não permitem interacção do utilizador; outros, porém, servem exactamente para que o utilizador possa comunicar ao programa as suas ordens.

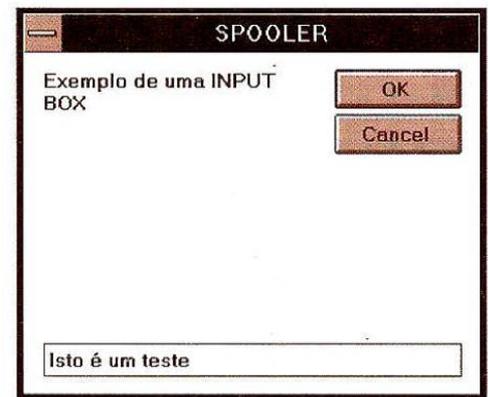
Além dos controlos, o Visual Basic oferece ainda a função INPUTBOX\$, com a seguinte sintaxe:

```
InputBox$(mensagem[, [título][, [padrão][, posiçãoX, posiçãoY]])
```

Esta função mostra uma mensagem no ecrã, numa *dialog-box*, e permite que o utilizador insira texto numa *text box* e/ou pressione um dos botões OK/CANCEL.

Se for omitido o argumento título, a janela é mostrada sem título; os argumentos **posiçãoX** e **posiçãoY** especificam a distância do canto superior esquerdo da *dialog-box* em relação ao canto superior esquerdo do ecrã (se forem omitidos, a janela é centrada).

A função devolve uma *string* com o conteúdo da *text box* ou uma *string* vazia, no caso do utilizador ter premido o botão OK. Se a função for evocada sem o cifrão no nome, o resultado é devolvido como um Variant, cujo conteúdo é do tipo *string*.



Esta INPUTBOX apresentada na figura foi configurada com a seguinte instrução:

```
dummy=InputBox("Exemplo de uma INPUT BOX", "SPOOLER")
```

O uso de elementos gráficos

O uso de elementos gráficos (pontos, linhas, caixas, círculos e outras formas geométricas) permite tornar mais apelativas as aplicações de Visual Basic.

Em Visual Basic podem usar-se duas técnicas para trabalhar com grafismos: através de controlos ou através dos métodos gráficos suportados pelos objectos.

O local onde o elemento gráfico é desenhado chama-se contentor (*container*). A posição e dimensão do desenho é expresso no sistema de coordenadas do contentor, com a

origem no canto superior esquerdo, ou seja, com as coordenadas (0,0).

As cores

O melhor método para o trabalho com cores é a sua especificação em termos de RGB (Red, Green, Blue) onde cada componente tem valores entre 0 e 255.

Por exemplo, o fundo de um *form* pode ser afectado com uma dada cor através da instrução:

```
form.backcolor =rgb(10,20,30)
```

Controlos para desenhar

Os elementos gráficos podem ser criados em *design-time*, com controlos, ou em *run-time*, com métodos. Os controlos são mais eficientes que os métodos (usam menos recursos do Windows e menos código para produzir as figuras).

Controlo Line – desenha segmentos de recta; as propriedades x1, y1, x2, y2 definem a posição; a propriedade DRAWSTYLE configura o tipo de linha. Este controlo não suporta o método MOVE.

Controlo Shape – desenha uma figura geométrica que pode ser um rectângulo (de cantos angulares ou arredondados), um quadrado (de cantos angulares ou arredondados), uma oval ou um círculo; este controlo permite definir a cor, tipo de transparência, padrão de preenchimento, cor, linha de contorno e dimensão da figura. Este controlo suporta o método MOVE.

Métodos para desenhar

Os métodos de desenho oferecem uma alternativa ao uso de controlos, permitindo criar efeitos vedados aos controlos (por exemplo, desenhar um ponto individual).

Os métodos gráficos produzem os desenhos num *form*, *picture box* ou no objecto PRINTER (o objecto PRINTER virtualiza uma página de papel da impressora).

Para desenhar usa-se a sintaxe **objecto.método**. Por exemplo, a instrução TESTE.PSET (10, 50) coloca um ponto no *form* TESTE, na posição de coordenadas (10, 50).

Os principais métodos gráficos são os seguintes:

Move left[,top[,width[,height]]] – aplica-se a *forms* ou controlos; move e, facultativamente, pode servir para ajustar dimensões dos objectos; o movimento produzido é suave.

Cls – aplica-se a *forms* e *picture boxes*; limpa a área de desenho.

Pset (x,y) [,cor] – desenha pontos individuais.

Line [(x1,y1) – (x2,y2)[,cor] – desenha uma linha entre dois pontos; se forem omitidas as primeiras coordenadas, usa a posição actual como ponto de origem.

Line (x1,y1)-(x2,y2)[,cor] B [F] – desenha um rectângulo ou quadrado entre cantos opostos; se for especificado o argumento “F”, enche a área com a cor de desenho da caixa; se for omitido, a caixa é preenchida com a cor e padrão especificadas pelas propriedades *fillcolor* e *fillstyle* do contentor.

Circle (x,y),raio[,cor] – este método desenha círculos; os argumentos (x,y) indicam o centro do círculo.

Circle (x,y),raio[,cor],[início],[fim],[aspecto] – desenha uma elipse; se o argumento “aspecto” for 1, desenha um círculo; se o argumento “aspecto” for maior que 1, desenha uma elipse; se o argumento “aspecto” for maior que 1, a elipse é vertical; se o argumento “aspecto” for menor que 1, a elipse é horizontal; os parâmetros “início” e “fim”, expressos em radianos ($360^\circ=2\text{Pi}$), permitem desenhar arcos; se os parâmetros “início” e “fim” forem negativos, além de delimitarem os arcos, produzem linhas a partir do centro, criando sectores de gráficos sectoriais (*pie slices*, ou seja, «fatias de tarte»).

Diferenças entre OLE (Object Linking and Embedding) e DDE (Dynamic Data Exchange)

O DDE é um protocolo definido para estabelecer ligações entre aplicações e permitir trocas de dados. Nem todas as aplicações suportam DDE. Das duas aplicações que estabeleceram a ligação DDE, a que estiver programada para receber os dados via *link* é designada por «aplicação-destino». De igual modo, a aplicação que estiver programada para enviar os dados via *link* é designada por «aplicação-fonte». O DDE é um mecanismo suportado pelo próprio ambiente Windows, e em ligações DDE, os dados são mostrados em controlos nativos do Visual Basic.

Com OLE, os dados são mostrados numa «fotografia», com a aparência que têm dentro da aplicação que os origina. Uma alteração aos dados implica que o controlo do programa passe para a aplicação que os criou (de forma automática); quando a aplicação termina o seu funcionamento, o controlo regressa automaticamente ao programa de Visual Basic.

A utilização do OLE dentro de um programa de Visual Basic implica a

Vejamos um exemplo em que, numa janela, se desenha aleatoriamente uma linha. Para que o funcionamento seja automático pode usar-se um *timer* e escrever este código dentro do *event procedure* TIMER:

```
line1.X1 = Int(form1.Width * Rnd)
line1.Y1 = Int(form1.Height * Rnd)
line1.X2 = Int(form1.Width * Rnd)
line1.Y2 = Int(form1.Height * Rnd)
```

O controlo OLE

A utilização deste controlo permite a uma aplicação de Visual Basic mostrar e manipular dados de outras aplicações de Windows.

colocação do *custom control* MSOLE2.VBX num *form*. Ao ser colocado o controlo no *form*, esse controlo vai guardar o nome da aplicação que contém os dados, os dados propriamente ditos (se for *embedded*) ou apenas uma referência (se for *link*) e a imagem dos dados no *form*.

tas alterações deverá ser feita programaticamente. Para guardar os dados (e, mais tarde, repô-los no *form*) é necessário colocar em ficheiro o conteúdo do objecto OLE. Para este fim usa-se a acção OLE1.ACTION=11 (*save to file*) ou =12 (*read from file*). O ficheiro a manipular é aberto em

A disquete de exercícios

Como exemplo prático ilustrativo do artigo deste mês propomo-vos um exercício (académico, dirão alguns) de manipulação de ficheiros de acesso sequencial e do uso de variáveis tipificadas: trata-se da implementação de uma pequena agenda pessoal.

E as novidades finais...

Pois fiquem a saber que o Visual Basic 4.0 já dá os seus primeiros passos, e aqui o escriba já teve oportunidade de «dar umas voltinhas» na versão BETA.

Um dos assuntos que fica por abordar nesta série de artigos acerca do Visual Basic (a criação de aplicações que manipulem bases de dados, acedidas através de *drivers* ODBC ou criadas com o *engine* do Access) será tratado nestas páginas mas já sobre a versão 4.0 do Visual Basic, que tem como uma das grandes novidades a utilização muito mais versátil dos controlos «ligados» a campos de tabelas de dados.

Até breve!

Modo de Utilização

1.

Inserir o controlo no *form* (cada controlo só pode conter um único tipo de dados).

2.

É mostrada uma janela *Insert Object* com a lista de todos os objectos registados no Windows; selecciona-se o objecto pretendido.

3.

A aplicação seleccionada entra em funcionamento e permite criar um novo documento (opção por omissão). Após a inserção dos dados e terminada a aplicação, os dados são mostrados no *form*. O objecto assim criado é EMBEDDED.

4.

Para criar um objecto LINKED, este deverá ser criado a partir de um ficheiro existente; na janela de especificação do ficheiro tem de seleccionar-se a opção *Link*.

5.

Em *run-time*, um duplo clic efectuado sobre o controlo abre a aplicação para edição dos dados.

6.

Em *design-time*, o botão direito do rato server para mostrar um conjunto de menus *pop-up* onde estão agrupados comandos que trabalham com o controlo OLE.

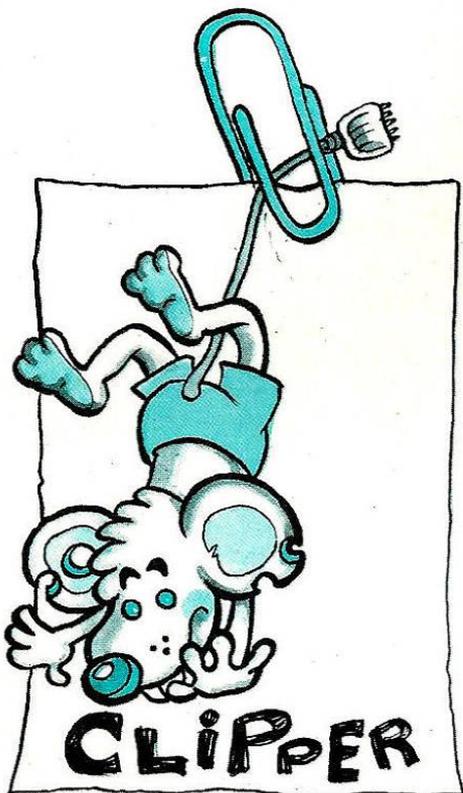
Uma das propriedades mais importantes do controlo OLE é a propriedade SOURCEITEM, que especifica a parcela dos dados a «ligar». A sintaxe desta propriedade varia de aplicação para aplicação. Por exemplo, para o Excel, poderia usar-se um bloco de células como «fonte»: R1C1:R15C3.

A (não) persistência dos dados

As alterações feitas aos dados de um objecto OLE não são guardadas quando o *form* é fechado. Isto significa que a eventual salvaguarda des-

binary mode. Vejamos um exemplo:

```
'para gravar
fich=freeFile
open «myole.ole» for binary as #fich
ole1.filenumbe=fich
ole1.action=11 'save to file
close #fich
'para ler
fich=freeFile
open «myole.ole» for binary as #fich
ole1.filenumbe=fich
ole1.action=12 'read from file
close #fich
```



NEWFUNCS.PRG

NOVAS FUNCÇÕES

PARA CLIPPER

Linguagem: Clipper

por Hugo Meneses

Toda a gente gosta dos menus bonitos dos programas da Central Point, como por exemplo o PC Tools; então resolvi criar este pequeno programa que tenta duplicar os tipos de menus desses programas para usar no Clipper 5.01.

Este programa contém várias funções que são:

- Function Fundo(cMensagem)
- Function TheMenu(aPrompts, cMsg)
- Procedure TerminarPrograma()

Para que servem? Boa pergunta. Como as fiz penso que posso responder a essa pergunta. Aqui vai...

Função: Fundo()

Propósito: O propósito desta função é criar um fundo para a aplicação. Este fundo contém na primeira linha o título que quisermos.

Modo de Usar:

Fundo("Titulo") // Mostra um título na 1ª linha do ecrã.

Função: TheMenu()

Propósito: Criar um menu com um mínimo de 2 opções e um máximo de 5 e depois apresentar uma mensagem à direita deste menu.

```
Modo de Usar: cOpcao := ;  
TheMenu({" Opcao 1 ", " Opcao 2 ", " Opcao 3 " }, "Uma mensagem")
```

Do Case

```
Case cOpcao = " Opcao 1 "
```

```
...
```

```
Case cOpcao = " Opcao 2 "
```

```
...
```

```
...
```

```
EndCase
```

Função: TerminarPrograma()

Propósito: Apresentar uma opção para sair do programa quando premos determinadas teclas.

```
Modo de Usar: // Usar no início do  
programa
```

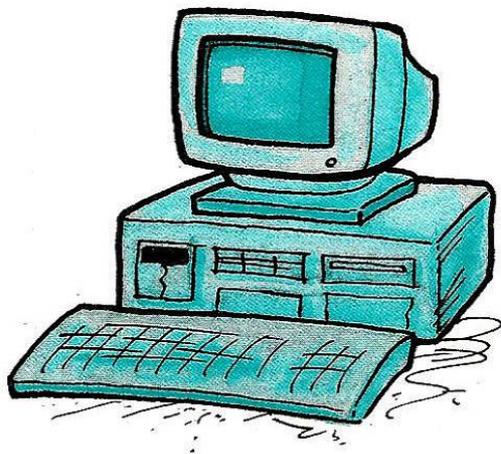
```
Set Key -33 To TerminarPrograma //  
Teclas ALT e F4
```

Agora devem estar a perguntar como podem usá-las. É fácil:

```
Clipper Programa  
RTLink File Programa, Funcs
```

Espero que gostem destas funções e as usem nos vossos programas e, já agora, se não gostarem façam críticas. Vai um exemplo do uso no programa chamado

EXEMPLO.PRG e uma *batch file* E.BAT para criá-lo.



EM BUSCA DO PC PERFEITO

Placas Mãe (Parte II)

por Afonso Infante

Na segunda parte desta série de artigos falaremos da placa-mãe do PC.

A placa-mãe (*motherboard*), é simplesmente a base de qualquer PC. Nela normalmente reside o processador, o co-processador (se este existir e for externo), o *chipset*, a memória (principal e *cache*), o BIOS e o *bus* de expansão.

Os sockets do processador e co-processador

Nos tempos do 8088/8086, 80286 e 80386, o processador estava simplesmente soldado à placa-mãe, havendo à parte um *socket* para encaixar o co-processador matemático 80x87 respectivo, caso se desejasse.

Já os 486 e Pentium vêm encaixados num *socket*, o que permite que o processador seja facilmente retirado e substituído por outro. Só os modelos sem co-processador matemático integrado (486SX, 486DLC, 486SLC, etc) é que vêm com um *socket* para um co-processador externo.

Apesar disso, modelos mais antigos de placas-mãe com o 486DX vinham com um *socket* para o co-processador matemático da Weitek, que era teoricamente muito mais rápido em cálculos de vírgula flutuante, desde que as aplicações estivessem preparadas para ele. Devido ao seu elevado preço, acabou por nunca ter grande aceitação, e hoje em dia as placas-mãe já não vêm preparadas

para ele, até porque nunca teve versões a mais de 33 Mhz.

Existe obviamente um tipo de *socket* para cada processador, existindo por vezes processadores que funcionam em vários *sockets* e *sockets* que aceitam vários processadores.

Há sempre quem defenda que sempre que se queira substituir o processador é simplesmente trocar a placa-mãe inteira. Afinal, o resto dos componentes custam uma fracção do que custa o processador. Uma placa-mãe completa custa cerca de 15% mais que o processador sozinho e, para todos os efeitos, obtém-se sempre melhores resultados do que trocando apenas o processador.

No entanto, nem sempre é possível trocar placas-mãe completas (isto é especialmente verdade nos computadores «de marca») e, nesses casos, os sistemas de substituição do processador por um *Overdrive* são a maneira mais económica de substituir o processador, já que a única outra opção é trocar o computador todo...

O chipset

O *chipset* é o conjunto de integrados que controla todo o sistema, desde o *bus* até à memória (principal e *cache*). Em geral são três, encarregando-se cada um da sua função. São facilmente identificáveis pois são sempre os integrados com mais pinos (a seguir ao processador).

Existem várias empresas a produzir *chipsets*, como a própria Intel, a Headland, a Micronics, a VLSI, a SIS, a IMS, a UMC e a OPTi (esta última tristemente famosa pela má qualidade dos seus integrados).

A RAM principal

A RAM (*Random Access Memory* – Memória de Acesso Aleatório) principal é simplesmente a memória normalmente utilizada para o carregamento e execução dos programas.

Hoje em dia os PC necessitam cada vez de mais memória. Apesar de os 386 e superiores serem capazes de utilizar os discos rígidos como memória, a sua lentidão relativa faz com que os sistemas equipados com RAM a menos acabem por ficar extremamente lentos.

Assim, convém ter bastante memória à partida e haver bastante espaço para expansão. Claro que também não convém exagerar: a RAM é cara e a partir de certo ponto um aumento de RAM não se traduz num aumento de desempenho do sistema.

Existem vários tipos de RAM:

DRAM – (*Dynamic RAM* – RAM dinâmica) Com tempos de acesso normais de 40, 50, 60, 70 ou 80 ns (nanosegundos – 1 ns = 10⁻⁹ segundos), este é o tipo utilizado na RAM principal dos PC pois, apesar de tudo, é a mais barata e com menor consumo de energia.

SRAM – (*Static RAM* – RAM estática) Com tempos de acesso normais de 12, 15, 20 ou 25 ns, esta é mais rápida que a DRAM, mas muito mais cara (mais de 6 vezes) e com muito maior consumo. Falaremos dela a propósito da RAM *cache*.

VRAM – (*Video RAM*) Especialmente otimizada para placas gráficas, é mais rápida e cara que a DRAM porque tem dois canais de acesso (*dual-port*).

A DRAM pode-se apresentar em várias «embalagens». Nos anos 80, cada fabricante usava o seu sistema, mas desde o princípio da década de 90 que se começou a utilizar o formato SIMM (*Single In-Line Memory Module*) como *standard* entre todos os fabricantes.

Existem 3 tipos de SIMM:

SIMM de 8 bits (30 pinos)

Estes SIMM, raramente utilizados em PC (apesar de serem vulgares nos CBM Amiga, por exemplo), não têm controlo de paridade. Enquanto que algumas pessoas são da opinião que a fiabilidade da DRAM actual elimina a necessidade de verificação da sua precisão (o controlo de paridade serve para detectar eventuais falhas físicas na memória), outras contrapõem dizendo que nunca se pode ser demasiado cuidadoso com os dados... A vantagem dos SIMM de 8 bits é que são obviamente mais baratos. Existem com capacidades de 256 kb, 1 Mb, 4 Mb e 16 Mb.

SIMM de 9 bits (30 pinos)

Estes SIMM já têm controlo de paridade e são os mais vulgares em PC. Devido ao controlo de paridade (1 bit extra para cada byte), são necessários 9 bits para armazenar um byte de informação. Como a memória dos 386DX e 486DX é acedida a 32 bits (4 bytes), estes SIMM têm que estar instalados em grupos (*banks*, ou bancos) de 4 SIMM absolutamente idênticos. Por isso é absolutamente obrigatório que o nú-

mero de SIMM instalados no computador seja um múltiplo de 4 (4, 8, 12, 16, etc), sendo os módulos iguais 4 a 4. Estes existem com as mesmas capacidades dos SIMMs de 8 bits:

- 256 kb (9 integrados de 256 kbits – 256 x 9)
- 1 Mb (2 integrados de 4 Megabits e 1 de 1 Megabit ou 9 integrados de 1 Megabit – 1 x 9)
- 4 Mb (9 integrados de 4 Megabits – 4 x 9)
- 16 Mb (9 integrados de 16 Megabits – 16 x 9).

Para manter as possibilidades de expansão abertas, convém sempre que de base sempre que possível se use apenas um banco (4 SIMM), mantendo o outro livre para acrescentar memória se necessário. Outra hipótese é procurar uma placa-mãe que tenha 4 bancos de SIMM (16 SIMM), o que permite uma muito maior flexibilidade na expansão.

No caso dos Pentium a situação ainda se complica mais, porque como este processador acede à memória a 64 bits, os SIMM de 9 bits têm que estar agrupados em bancos de 8... Por isso é que a esmagadora maioria das placas-mãe Pentium utilizam SIMM de 36 bits, de que falarei de seguida.

SIMM de 36 bits (72 pinos)

Também com controlo de paridade, estes SIMM diferem dos anteriores num ponto importante: como cada módulo já tem 4 bytes (9 bits x 4), não é necessário agrupá-los aos 4 (ou 8, com nos Pentium) de cada vez. Assim, há muito maior liberdade de configuração de memória: nos 386DX e 486DX pode-se colocar qualquer número de SIMM, até ao máximo que a placa-mãe permitir, e nos Pentium apenas têm que ser agrupados aos pares.

Aliás, aproveito para chamar a atenção para o facto de que as verdadeiras placas-mãe Pentium têm que obrigatoriamente ter bancos de memória de 64 bits (grupos de 8 SIMM de 9 bits ou 2 SIMM de 36 bits). Existem placas-mãe que anun-

ciam ser para Pentium mas têm bancos de memória 32 bits (4 SIMM de 9 bits ou 1 SIMM de 36 bits). Essas placas não tiram partido do *data path* de 64 bits do Pentium, o que se traduz numa diminuição considerável do desempenho.

Os SIMMs de 36 bits existem nas seguintes capacidades:

- 1 Mb (256 kbits x 36)
- 2 Mb (512 kbits x 36)
- 4 Mb (1 Mbit x 36)
- 8 Mb (2 Mbit x 36)
- 16 Mb (4 Mbit x 36)
- 32 Mb (8 Mbit x 36)
- 64 Mb (16 Mbit x 36)

Este sistema de memória é vulgar apenas nos sistemas com um *bus* EISA e nos Pentium, mas começa cada vez a tornar-se mais popular nos 486, até porque o seu custo é ligeiramente inferior ao dos SIMM de 9 bits. Infelizmente ainda não existem versões de 40 e 50 ns (só 60 ns e superior), mas estas devem estar prestes a sair.

Existem ainda SIMM de 32 bits (72 pinos), que são semelhantes aos SIMM de 36 bits, mas não têm controlo de paridade (daí os 4 bits – 1 bit por byte – a menos).

A RAM cache

A RAM *cache*, introduzida com os 386DX, é do tipo SRAM e como tal muito mais rápida que a DRAM usada na RAM principal.

O problema é que a DRAM é demasiado lenta para os processadores actuais, e a SRAM, essa suficientemente rápida, é demasiado cara para ser utilizada como RAM principal. 16 Mb de SRAM custariam centenas – se não milhares – de contos, para além de ocuparem mais espaço e consumirem mais do que a DRAM. Assim, foi necessário criar uma solução de compromisso, a RAM *cache*.

As *caches* vêm em todos os tamanhos e feitios, mas são todas desenhadas para acelerar o acesso do processador a dados frequentemente usados. As coisas processam-se da seguinte maneira: à medida

que o processador vai buscar dados à RAM principal, o controlador de *cache* vai copiar esses dados e, segundo o seu algoritmo de controlo, vai também copiar instruções próximas ou relacionadas. O resultado é que da próxima vez que o processador precisar da informação, ela provavelmente estará na *cache* (95% do tempo o processador repete operações sobre o mesmo conjunto de dados), e como esta é mais rápida que a RAM principal, os dados chegarão mais depressa à CPU.

Cache interna e externa

Um sistema pode ter dois níveis de *cache*. A de nível 1, que se encontra incorporada no processador, e a de nível 2, que é externa ao processador.

A *cache* interna é ultra-rápida, pois tem uma ligação ao processador muito larga (128 bits) e está fisi-

camente muito próximo deste (mais próximo, impossível). É extremamente útil para manter «à mão» os dados e instruções usados mais frequentemente. Mas como vimos, as *caches* não são todas iguais, nem sequer têm o mesmo tamanho de processador para processador.

A mais eficaz entre os compatíveis Intel 80x86 é a incorporada no *Pentium*, pois além de ser do tipo *write-back* (ao contrário de todos os outros processadores, à excepção dos Cyrix 486DX e DX2, que também têm uma *cache write-back*), está dividida em duas secções de 8 kb cada: uma para dados e a outra para instruções. Já os DX4 e os IBM *Blue Lightning* utilizam uma *cache* unificada de 16 kb.

A *cache* externa é também importante. Esta tem normalmente 128 kb nos 386DX, 256 kb nos 486 e 512 kb nos *Pentium*, pois estes são os valores recomendados para cada um

desses processadores.

Até há bem pouco tempo a *cache* externa (devido ao consumo elevado da SRAM) era utilizada em pouca quantidade nos computadores portáteis, mas com o aparecimento recente das SRAM de 3,3 V as coisas podem alterar-se.

No próximo artigo da série falaremos de uma outra parte importante das placas-mãe: o *bus* de expansão.

Graficaria

PUBLICIDADE E ARTES GRÁFICAS, LDA.

- PROJECTOS GRÁFICOS
- COMPOSIÇÃO
- ARTE-FINALIZAÇÃO
- SELECÇÃO DE CORES
- TRATAMENTO DIGITAL DE IMAGEM
- EDIÇÃO ELECTRÓNICA
- SERVICE BUREAU
- MONTAGEM
- IMPRESSÃO

QUALIDADE
EFICIÊNCIA
RAPIDEZ

Rua Duarte Pacheco Pereira, 69-A - Damaia-de-Baixo - 2720 AMADORA - Telef.: 474 60 11/41 - Fax: 475 22 89



USERLIB V2.0

ACTUALIZAÇÃO

DA USERLIB V1.1

Linguagem: QuickBasic

por Nélcio Abreu

Um olá em especial aos BASprogramadores...

Há quem diga que sou insistente, mas não é verdade... a razão do meu regresso deve-se simplesmente à publicação da VERSÃO 2.0 da USER LIB, que fora publicada na **Spooler Nº 31** como versão 1.1. Porquê uma nova versão? Pura e simplesmente para eliminar alguns *bugs* detectados na versão 1.1, esses «malditos» que sempre surgem durante o trabalho «duro» de um programador.

Convém ainda referir que na versão 1.1 o manual possuía a extensão MNL, vindo agora a possuir a extensão DOC (USERLIB.DOC).

Em relação à versão anterior, só me resta acrescentar as novidades e alterações na versão 2.0:

Instruções novas adicionadas à LIB:

Procedimentos:

Controlo das impressoras HP DeskJet PCL nível 3 (HP 5xx - 500, 510, 520, 500C, 550C, 560C, etc)

Controlo de ambiente

SAVESCR - Grava o conteúdo do ecrã (modo de texto) para dentro de uma variável
RESTORESCR - Restaura o conteúdo do ecrã (oposto do SAVESCR)
GETCOLOR - Obtém as cores numa determinada zona do ecrã

Funções:

GET.KEY\$() - Leitura do teclado
HP850\$() - Conversão de texto para imprimir em HP DeskJet em code-page 850
CP860\$() - Conversão de texto de 850 para 860
PRN.NUMBER%() - Obtém o Nº associado à impressora
PRN.STATUS%() - Obtém o estado da mesma
NET.STATUS%() - Obtém o estado da rede (*NetWork*)
NET.ERROR%() - Obtém o estado de erro

Instruções modificadas/corrigidas na LIB:

Procedimentos:

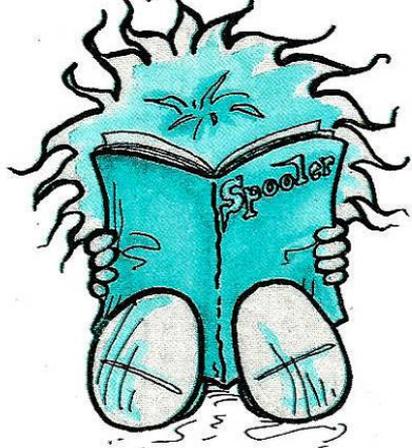
CURSOR, SAVE.SCREEN, RESTORE.SCREEN, COPY, BAK, MOVE, ADIR

Funções:

BIN2DEC&()	BIN2HEX\$()	BIN2OCT\$()
DEC2BIN\$()	DEC2HEX\$()	DEC2OCT\$()
HEX2BIN\$()	HEX2DEC&()	HEX2OCT\$()
OCT2DEC&()	OCT2BIN\$()	OCT2HEX\$()
ADD.BIN\$()	ADD.HEX\$()	ADD.OCT\$()
SUB.HEX\$()	SUB.OCT\$()	SUB.BIN\$()
MUL.HEX\$()	MUL.OCT\$()	MUL.BIN\$()
DIV.HEX\$()	DIV.OCT\$()	DIV.BIN\$()
TEMPFILE\$()		

HPRESET	Inicialização da impressora
HPEJECT	Ejecção do papel
HPNLQ	Seleção do modo de impressão NLQ
HPDRAFT	Seleção do modo de impressão <i>draft</i>
HPFONT	Seleção do tipo de fonte
HPSTYLE	Seleção do estilo
HPINTENSITY	Seleção da intensidade de impressão
HPFIXED	Seleção do modo de escrita fixo

HPPROP	Seleção do modo de escrita proporcional
HPPITCH	Seleção do <i>pitch</i> do carácter
HPMARGIN	Controlo das margens
HPCODEPAGE	Seleção do <i>code-page</i> da impressora
HPPAPER	Seleção do tipo de papel
HPORIENTATION	Seleção da orientação (Retrato/Paisagem)
HPSIZE	Seleção do tamanho da folha



COLABORAÇÃO
DO LEITOR

I'm back...

Desta vez, após pesquisar os meus ancestrais arquivos *Top-Secret*, selecionei sete funções que passo a explicar.

`int x_PrintScr()`

Imaginem que estão a fazer um pequeno programa de gestão de clientes (por exemplo) para uma empresa. Imaginem ainda que no meio das vossas «simples» rotinas de reconhecimento de voz, existe uma que serve para efectuar a consulta da ficha de cada um dos clientes. Imaginem também que quem vos encomendou o programa quer ter a hipótese de imprimir a ficha do cliente. Parem de imaginar e pensem como o fariam...

Aposto que grande parte de vocês pensou numa coisa do género: «É fácil! Escrevo na última linha `PRIMA SHIFT+PRINT SCREEN PARA IMPRIMIR A FICHA` e pronto!». Mas esquecem-se, pelo menos, de duas coisas. Primeiro, grande parte do pessoal que trabalha com esse género de programas não conhece lá muito bem um computador, e encontrar as teclas pedidas (ou melhor, perdidas!) pode ser uma tarefa árdua. E segundo, uma ficha de cliente com «`PRIMA SHIFT+PRINT SCREEN PARA IMPRIMIR A FICHA`» escrito na última linha não fica lá muito bem, pois não? A minha solução é esta: mostram a ficha, escrevem por exemplo «`PRIMA F2 PARA IMPRIMIR A FI-`

WEIRD 2

WEIRD?

CONTINUA O MISTÉRIO!

Linguagem: C

por Tiago Ribeiro



CHA», esperam pela tecla, apagam a mensagem e chamam a função `x_PrintScr`.

Já devem ter percebido o que isto faz, ou seja, envia um ecrã inteiro para a impressora. Esta função retorna ainda um valor para poderem testar o seu grau de sucesso. Embora isto tudo possa parecer complicado, bastam duas linhas de código para o fazer. E pronto, como agora já se devem estar a contorcer todos para saber como é que isto se faz, o melhor é correr o *Demo* ou irem ver o código...

`x_MotorOff(int)`

Sempre que acedem a uma *drive* a respectiva luz acende-se... Pois, provavelmente já tinham reparado... Mas o que não devem saber é que o DOS, depois de fazer o que lhe foi pedido, a deixa ligada mais um pouco. Não me perguntem para quê, mas eles lá devem ter alguma razão! O sistema operativo tem um registo próprio para guardar o tempo que a luz permanece acesa. Esse valor é decrementado 18.2 vezes por segundo e, quando chegar a zero, «puf», não há luz p'ra ninguém! Geralmente o valor que lá está é 37, o que significa que a luz continua ligada durante $37 / 18.2 = 2$ segundos,

mas este número pode ser alterado. Para isso basta chamarem esta função com um valor inteiro que representa o número de *ticks* (18.2 por segundo) durante os quais a luz ficará ligada.

int x_KeyLights(int)

Liguem lá o *led* do CAPS LOCK sem tocar na tecla respectiva! Não conseguem? Então pronto, liguem o do NUM LOCK... Também não? Pronto, pronto, liguem apenas o do SCROLL LOCK... O quê? Não? Ainda bem, porque se conseguissem esta função não servia para nada...

Esta função serve então para «imaginariamente» premir as teclas CAPS, NUM ou SCROLL LOCK, o que também acende ou apaga os *leds* respectivos. Para isso basta chamar a função com um valor inteiro que representa quais as teclas a ligar, ficando as restantes desligadas. A função retorna um valor que representa a combinação de teclas que estava ligada antes desta nova chamada.

int x_IsColor()

Ok, esta não tem muito que se lhe diga. Serve apenas para indicar se o monitor instalado é mono ou policromático. Isto nem sempre trabalha como devia, embora na maior parte dos casos funcione bem.

Isto é feito «olhando» para o CRT (*Controller Base Address*), que se situa na posição de memória 0:0463h. Este terá o valor 3D4h para um ecrã poli, e 3B4h para mono. Esta função retorna 0 ou 1, caso o ecrã não seja ou seja policromático, respectivamente.

int x_NumDrives()

Esta também não é difícil de explicar e muito menos de implementar, mas acho que é bastante útil. Serve para contar o número de *drives* reais existentes no sistema. É que mesmo existindo apenas uma *drive*, o computador cria a *drive* B: que fisicamente não existe! Como isto poderá induzir em erro alguns problemas complicados que vocês

façam, convém verificar antes as *drives* que realmente existem.

int x_Coprocessor()

Esta é a terceira e última das funções que embora pequenas (realmente minúsculas!) são bastante úteis (o que também é treta!). Esta é realmente espectacular (tanga!), pois serve para verificar se está presente no sistema um coprocessador numérico (*big deal!*). Para quem não saiba, esta coisa serve para efectuar cálculos numéricos aliviando assim o trabalho (trabalhão!) da UCP. UCP é, para quem não saiba, Unidade Central de Processamento, ou seja, é a parte mais importante do computador. Ooops, já nos desviámos um pouco do assunto!

Pois como eu dizia esta função retorna o valor 1 ou 0 caso exista (ou não) um coprocessador numérico. Para quem não saiba esta coisa serve para... Oh, não! Voltámos ao mesmo!

```
x_StartClock (int, int, int, int)
x_StopClock ( )
x_HideClock ( )
x_ShowClock ( )
x_ClockColor (int, int)
x_ClockPos (int, int)
```

Estas funções, embora sendo as últimas, são as primeiras (lá dizia o ditado!) em termos de importância e a nível didáctico.

Fica sempre bem um programa ter um relógio em tempo real num canto qualquer. Pois fica, mas metê-lo lá deu-me mais trabalho que Análise Matemática 2. Pensei em duas hipóteses: ou criar um TSR para ser executado antes do programa, ou de algum modo iniciar e parar o relógio dentro do programa.

Como todas as minhas tentativas de criar um TSR saíram frustradas com *crashes* mais ou menos estranhos, optei pela segunda hipótese. Pensei então em capturar uma interrupção, e como os meus conhecimentos nesta área são praticamente nulos, as primeiras tentativas foram uma autêntica desgraça. Mas, finalmente, lá consegui instalar uma rotina na interrupção 1Ch (*USER TI-*

MER TICK). Como dizia o outro, foi sol de pouca dura, pois por vezes, por motivos mais ou menos sinistros, o computador ficava pendurado. Pensei, pensei, pensei, e tive que ir dormir pois estava cansado de tanto pensar... No dia seguinte abriu novamente a época de caça e lá capturei, a custo, a interrupção 8 (*SYSTEM TIMER*). Essa já não me deu problemas, e lá consegui instalar a minha rotina (*Yesss!*). Mas como numa interrupção não posso (eu pelos menos não consegui!) usar as funções *standard* de C, tive que ir estudar um pouco melhor a escrita directa no ecrã.

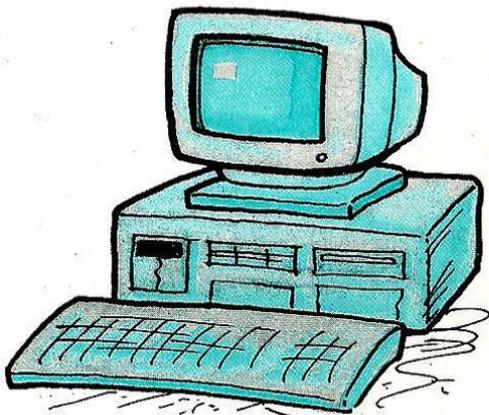
Neste momento para inicializar o relógio basta chamar a função **x_StartClock** com as coordenadas e atributos que o relógio irá ter no ecrã. Quando já não necessitarem do relógio chamem a função **x_StopClock** e pronto. Esta função TEM que ser executada pelo menos uma vez. Após terem iniciado o relógio podem ainda alterar os seus atributos e/ou coordenadas, com as funções **x_ClockColor** e **x_ClockPos**, respectivamente.

Para pararem o relógio momentaneamente podem usar a função **x_Hide** que deixará de actualizar o relógio no ecrã. Para voltarem ao normal façam **x_Show**. Notem que esta função está preparada para apresentar o relógio num ecrã policromático. Ainda estive para alterar isto de modo a funcionar independentemente do ecrã, mas muito sinceramente, já estava farto! Por isso deixo essa alteração para quem a quiser efectuar. Para isso basta saberem que a linha (...) MK_FP (0xB800, 0), ficaria, num ecrã monocromático, (...) MK_FP (0xB000, 0).

E pronto, são estas as funções deste volume. Para perceberem melhor o funcionamento de cada uma delas sugiro que corram o Demo e consultem o código. E já agora observem bem o final e... não se assustem... AH, AH, AH (gargalhadas sinistras)...

Enquanto aguardo notícias *daí*, vou trabalhando no WEIRD 3 *aquí*...

I'll be back ...



BIOS

Quase todas as linguagens de programação de alto nível possuem comandos (procedimentos ou funções) para escrever mensagens no ecrã, ler ou escrever em memória de massa (disco), etc. Para que o computador consiga «perceber» os programas criados nessas linguagens, é necessários compilá-los, ou seja, transformá-los em código-máquina. Este código é equivalente às instruções em *assembly*, uma linguagem de baixo nível. No entanto, o *assembly* não tem instruções que permitam escrever no ecrã ou em disco, tendo então que se utilizar os serviços do BIOS e do DOS.

Este raciocínio, que à primeira vista parece um tanto básico e inútil, demonstra-nos porque é que o Turbo Pascal teve tanta aceitação e sucesso no meio dos programadores profissionais e não-profissionais. Foram criados muitos comandos que não pertenciam ao Pascal *standard*, de maneira a aproveitar melhor as capacidades do PC. Assim, outras potencialidades do BIOS e DOS estão ao dispor através de um procedimento, em vez de deixar ao programador a tarefa de perceber e utilizar directamente esses recursos.

O aumento de poder da linguagem teve como contrapartida uma perda de compatibilidade, pois esses comandos não são *standard* – mas, para a grande maioria dos utilizadores, isso não tem qualquer importância.

Noutras linguagens, como o Basic, esse tipo de instruções não foi desenvolvido deixando para o programador todo o trabalho de pesquisa. Mas isso não quer dizer que o trabalho esteja todo feito em linguagens como o Turbo Pascal ou Turbo C; ainda existem alguns «truques» bem úteis por fazer!

O BIOS E O DOS

(Parte II)

BIOS & Assembly

Como já referi (neste artigo e no anterior), o BIOS e o DOS possuem uma variedade enorme de serviços que são usados para tirar partido de certos dispositivos e periféricos. O que talvez ainda seja pouco perceptível, é a quantidade e minuciosidade destes. Ora, como foi dito acima, os comandos das linguagens de alto nível são transformadas em instruções mais pequenas, algumas utilizando esses serviços fornecidos pelo BIOS.

Escrever uma letra no ecrã envolve a utilização de um serviço específico que faz todo o trabalho por nós. Mas e ele? Sim, ele, o serviço! Como é que ele faz para escrever uma letra no ecrã? Até porque é feito a partir de *software*, pois os serviços do BIOS não são nada mais que programas, não é verdade?

Pois bem, através da utilização dos portos internos de comunicação (ligações dos vários dispositivos ao microprocessador), torna-se possível «falar» directamente com a placa de vídeo e programá-la de maneira a que seja escrita uma letra no ecrã. E este processo até é mais rápido que aceder ao BIOS!

Existem duas razões para não o fazer: primeiro é mais complicadito – é como estar a reescrever um programa que está já pronto a ser utilizado; segundo, porque pode ser problemático utilizar esse processo quando estão a correr outros programas (TSR, Windows e coisas do género).

Aos programas que utilizam este tipo de processos, o Windows chama-lhes «mal comportados» e como usam a placa de vídeo e/ou o disco directamente, não podem ser usados em multi-tarefa nem metidos dentro de janelas.

por Tomé Pesseguero

Todos os serviços do BIOS/DOS podem ser acedidos directa e facilmente a partir do *Assembly*. Para o fazer, é necessário não esquecer os parâmetros essenciais ao funcionamento do serviço chamado.

Para escrever uma letra no ecrã, poder ser utilizado o serviço 09h da interrupção 10h (BIOS) ou a função 02h do DOS (interrupção 21h). Em muitas ocasiões existem funções do DOS que executam um subserviço do BIOS, o que a torna mais fácil de utilizar; por essa razão o exemplo será feito com a chamada ao DOS:

```
mov ah, 02h ; escolhe a função 02h
mov dl, 'x' ; coloca a letra a
              escrever no registo DL
int 21h      ; activa ou chama a
              interrupção 21h
```

Este é um exemplo, em *Assembly*, de como se pode escrever um carácter no ecrã através de uma função do DOS. O valor em AH irá activar o serviço correspondente e a instrução INT irá chamar a interrupção. Até aqui, funciona tudo da mesma forma, o resto dos parâmetros irão variar em tipo e de registo, de acordo com o serviço escolhido. Mais informações sobre *Assembly* e a arquitectura interna do computador, poderão ser encontradas no minicurso de *Assembly* publicado na **Spooler** (números 18 a 22) ou, mais desenvolvidamente, no livro «*Assembly*, Puro e Simples» da autoria de Vasco Correia Veloso, editado pela Graficria.

Basic BIOS-lógico

Nas linguagens de alto nível, o processo de chamada a uma interrupção é quase tão simples como em

Assembly. Primeiro, tem que ser definida a estrutura onde irão ficar armazenados os dados referentes aos registos – em Quick Basic deverá ser feito da seguinte maneira:

```

TYPE RegType
  AX AS INTEGER
  BX AS INTEGER
  CX AS INTEGER
  DX AS INTEGER
  BP AS INTEGER
  SI AS INTEGER
  DI AS INTEGER
  FLAGS AS INTEGER
END TYPE

```

Este tipo também se encontra definido no ficheiro QB.BI, o qual pode ser utilizado com um \$INCLUDE. Depois, basta declarar uma variável do tipo registos:

DIM regs AS registos

É agora possível escrever em Basic o exemplo anterior (em *Assembly*):

```

regs.ax=&h0200 ' carrega o registo AX com o valor 200h – AH=02 AL=00
regs.dx=asc ("x") ' o registo DL é carregado com o valor ASCII da letra «x»
CALL INTERRUPT (&h21, regs, regs) ' é chamada a interrupção 21h

```

Existe uma série de considerações a fazer relativamente a estas linhas de código. Primeiro, foi preciso definir uma estrutura de dados, para que o estado (valores) dos registos possa ser estabelecido. Depois definiu-se uma variável desse tipo e deram-se valores aos registos AX e DX. Note-se que cada registo recebe um valor de 16 bits e não é possível aceder directamente a cada uma das suas partes (mais e menos significativa) separadamente.

Portanto, o valor 0200h em AX, é o mesmo que colocar 02h em AH e 00 em AL; no caso do registo DX, o valor em ASCII da letra 'x' só tem oito bits – isso significa que DL será carregado com esse valor, enquanto DH receberá o valor 00h (zero).

O comando CALL INTERRUPT necessita de três parâmetros, o número da interrupção, os registos a serem modificados e o resultado final dos registos (depois de executada a interrupção).

Como os registos de entrada só são necessários antes da execução da interrupção e os de saída só são modificados no fim, eles podem estar na mesma variável, embora o exemplo dado não tenha registos de saída. Nos casos em que isso aconteça, colocam-se os valores nos registos correspondentes e depois vai-se lá buscar os resultados – tudo na mesma variável!

Antes de finalizar o comentário ao exemplo, devo referir que a função INTERRUPT não está embutida no Basic. Em vez disso, está presente na biblioteca QB.QLB e para poder ser usada deve introduzir-se a seguinte linha de comando, ao carregar o compilador:

```
QB /L QB.QLB
```

Pascal e C

A partir do Pascal e do C, também se pode aceder ao BIOS. Para isso, é necessário seguir os mesmos passos que efectuámos no Basic.

Por exemplo, no Turbo Pascal já existe uma estrutura de dados definida na unit DOS; portanto não é preciso estar a defini-la novamente, basta fazer o seguinte:

```

PROGRAM XXX;
Uses Dos;

Var regs: Registers;

Begin
  regs.ah := $02;
  regs.dl := ord ('x');
  intr ($21, regs)
End.

```

O Pascal possui também duas vantagens no acesso aos serviços do BIOS/DOS. Em primeiro lugar, pode-se utilizar os registos na forma de 16 bits ou de 8 bits, em que são divididos em duas partes (AH e AL, por exemplo). A outra vantagem (pouco significativa) é de poder usar uma instrução mais confortável para ac-

der às funções do DOS (int 21h) – o procedimento MSDOS, que necessita de um único parâmetro – a variável que contém os registos. Assim, no exemplo, em vez de se utilizar intr (\$21, regs), poderia escrever-se msdos (regs). Recomenda-se também a leitura do artigo «Aceder aos serviços do BIOS através do Pascal» (**Spooler** Nº 25) para os interessados em saber mais sobre os serviços vídeo do BIOS.

Em Turbo C, o caso é idêntico, como se poderia esperar. Existe uma *union* já definida no ficheiro DOS.H, com o nome de REGS. Segue-se o exemplo:

```

#include <dos.h>

void main (void)
{
  union REGS regs;

  regs.h.ah = 0x21;
  regs.h.dl = 'x';

  int86 (0x21, &regs, &regs);
}

```

Tal como no Turbo Pascal, também existe uma estrutura pré-definida para este fim. Podem-se utilizar os registos como conjuntos de 8 ou 16 bits, tendo em atenção que existem duas *unions* encadeadas; assim, para se poderem usar os registos de 8 bits, deve-se usar **regs.h** e para os de 16 bits deve ser **regs.x**. Além disso, existe uma outra função (intdos) que permite aceder única e exclusivamente à int 21h.

Como os programadores de C já devem ter alguma experiência com Pascal ou Basic, não serão feitas muito mais referências a essa linguagem. Todos os exemplos podem ser facilmente transportados de Turbo Pascal para Turbo C.

O mistério das teclas especiais

A compreensão do funcionamento do BIOS/DOS pode ser muito útil, mesmo que não se utilizem os seus serviços directamente. Por exemplo, a detecção de teclas especiais permanece ainda um mistério para muita gente.

As teclas especiais produzem caracteres que não estão definidos no código ASCII, que é utilizado pelo resto do teclado (a parte convencional). Por isso, elas têm que ser detectadas de uma maneira especial, pois os 255 caracteres do código ASCII já estão tomados, não havendo lugar para eles.

Para estudarmos o assunto, temos de ir ao fundo da questão – o BIOS! O serviço 00h da interrupção 16h (serviços de teclado do BIOS) é o responsável pela leitura dos caracteres do teclado. Ao ler um carácter, emite dois bytes (nos registos AL e AH) – um referente ao código ASCII e outro que indica o código de varrimento (*scan code*). O *scan code* é um código único para cada tecla, ao contrário do ASCII que tem um número para cada símbolo.

Quando uma tecla especial é premiada, é emitido o código ASCII 0 (zero). Portanto, a tecla só é distinguível das outras através do seu código de varrimento. Cada compilador tem a sua maneira própria de resolver este problema; o Quick Basic envia uma *string* com dois elementos – o primeiro é o código ASCII zero e o segundo é o *scan code* da tecla. Por outro lado, o Turbo Pascal e Turbo C, emitem esses dois caracteres também, mas independentemente, ou seja, funciona como se valesse por dois toques na tecla. Assim, se for emitido um carácter zero, quer dizer que o próximo carácter é o código de varrimento da tecla premiada – é a nova moda do 2 em 1!

O Turbo C possui também uma outra função (*bioskey()*) que dá o estado actual de algumas teclas especiais, como CTRL, ALT e SHIFT.

Fazer um *reset*

Em alguns casos, é necessário fazer um *reset* ao sistema (em programas que tenham *passwords*, por exemplo). Normalmente, uma coisa deste tipo só seria possível em *assembly*, pois não existe nenhuma interrupção que permita fazê-lo. A interrupção 19h carrega o *Bootstrap*, ou seja, carrega novamente o sistema operativo, mas normalmente isso não permite uma inicialização do

computador, até porque esse é o último passo do POST (*Power-On Self Test*) – o programa em ROM que faz os testes à memória e aos vários periféricos.

O POST está situado na ROM (como seria de esperar) e a única maneira de lhe chegar é através do endereço FFFF:0000h, onde existe uma instrução de salto (*jump*) que passa o controlo para o POST. Este endereço não foi escolhido ao acaso; quando o computador é ligado, os registos CS (*Code Segment*) e IP (*Instruction Pointer*) ficam com os valores FFFFh (os 16 bits todos no estado 1) e 0000h (todos os bits a 0) respectivamente. Portanto, a primeira coisa que o computador faz, é executar a instrução neste endereço; depois o controlo é passado ao POST.

Nas linguagens de alto nível, não é possível saltar para um determinado endereço. A única maneira de o fazer é através da chamada a uma interrupção. Uma maneira de chamar o POST é, então, modificar um vector de interrupção – fazendo-o apontar para o endereço FFFF:0h e chamar a respectiva interrupção. Pode-se utilizar a interrupção 19h, uma vez que ela não é chamada pelo BIOS nem pelo DOS.

Para proceder à toca de endereços, deve ser utilizado o serviço 25h das funções do DOS, que permite substituir o endereço do vector de uma interrupção. Os parâmetros a fornecer são os seguintes:

AH = 25h (número da função)
AL = 19h (número da interrupção)
DS = FFFFh (segmento do endereço)
DX = 0 (offset do endereço)

De seguida, deve-se chamar interrupção 21h (serviços do DOS), para que o endereço seja substituído e por fim, chama-se a interrupção 19h para fazer o *reset*. Esta rotina está presente na disquete, em Basic e Pascal, com o nome *ResetCold*.

É ainda necessário fazer três comentários às listagens deste exemplo. Em primeiro lugar, quando o POST está a correr, ele lê o conteúdo da *word* no endereço 0:472h, onde está um código. Este indica-lhe qual o tipo de *reboot* a executar. Exis-

tem dois tipos diferentes de *reboot* (ou *reset*, que neste caso significam a mesma coisa) – a quente (*warm reboot*) e a frio (*cold reboot*). No primeiro caso, o efeito é o mesmo que premir as teclas Ctrl-Alt-Del, enquanto o outro produz o efeito de *reset* em *hardware*, isto é, é feito um teste à memória e outro mais extenso ao disco rígido. Para o *reboot* a quente, é necessário colocar no endereço 0:472h, o valor 1234h. Portanto, nestes exemplos o *reboot* faz-se a frio!

Para que se possa fazer um *reboot* a quente, criei uma outra rotina, cuja única diferença é a colocação da palavra 1234h no endereço necessário para evitar os testes de memória e disco prolongados. Esta é chamada *ResetWarm* e também está disponível nas duas linguagens.

No Quick Basic existem dois casos especiais: como neste exemplo foi usado o registo de segmento DS, torna-se necessário usar o comando *INTERRUPTX()*, em vez do *INTERRUPT()*. Por essa razão, é também obrigatório mudar a estrutura de dados onde se guardam os valores dos registos; usando neste caso, o tipo *RegTypeX*, que se encontra definido no programa-fonte.

É também no Quick Basic que existe mais um problema. Uma vez que ele permite que sejam usados os valores originais dos registos (colocando no registo o valor -1), leva ao aparecimento de uma questão: se existe um valor que é usado para efeitos de controlo, então não pode ser colocado no registo como um verdadeiro valor! É o que acontece aqui, pois o valor -1, corresponde ao hexadecimal FFFFh; quando este valor é atribuído a um registo, o Basic pensa que se trata de um código de controlo, informando-o para não modificar esse registo, usando-o com o valor actual. Para contornar isto, basta usar um par segmento: *offset* diferente, mas que aponte para o mesmo endereço físico – foi utilizado o segmento F000h e o *offset* FFF0h, que correspondem ao mesmo endereço que FFFF:0000h.

Lista de equipamento

A interrupção 11h permite saber

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
X	X	nº de impressoras (paralelo) (reservado)
.	.	X	adaptador de jogos
.	.	.	X	X	X	nº de portas série (não usado)
.	X	nº de drives de disquete - 1 (sem interesse)
.	X	X	(sem interesse)
.	X	X	coprocessador instalado
.	X	X	.	.	X inexistência de drives

Figura 1

se determinados dispositivos estão ou não instalados. Quando é chamada, devolve uma *word* (dois bytes), que é a lista de dispositivos ou equipamento. A *word* está assim distribuída (ver figura 1).

Através da figura, pode-se ver que cada conjunto de bits tem um significado específico; por exemplo, os bits 14 e 15 indicam o número de impressoras – normalmente isto equivale ao número de portas de comunicação em paralelo. No caso do adaptador de jogos, que não é mais do que uma placa com uma interface para joystick, o único bit (12) simplesmente indica a sua existência ou inexistência, conforme for 1 ou 0, respectivamente.

Os bits 9, 10 e 11 fornecem o número de portas de comunicação em série (0-7), sendo usado um só bit para a existência de coprocessador aritmético. Para o número de drives de disquete, o caso é diferente: os bits 6 e 7 formam um número de 0 a 3 (4 valores distintos), indicando o número de drives, de 1 a 4; isto é, quando o valor for igual a zero, significa que existe uma drive. Para o caso de não existir qualquer drive, pode ler-se o bit 0, que contém o valor 1 se existirem drives de disquetes (devendo portanto verificar-se os bits 6 e 7).

Para utilizar as rotinas em Basic e Pascal (com o nome Equip), devem consultar-se os seus cabeçalhos que contêm vários parâmetros de diversos tipos. A compreensão do seu código, no entanto, será um pouco mais difícil para quem não está habituado a lidar com operações a nível do bit.

Para se conseguir ler os valores de apenas alguns bits de uma *word* ou byte, é necessário efectuar duas operações: a primeira, consiste em

colocar a zeros todos os bits de ordem superior aos que queremos consultar; depois, basta dividir o resultado pelo valor do bit menos significativo do conjunto que queremos extrair o conteúdo.

Talvez um exemplo seja mais elucidativo: se quisermos saber qual o número de portas série existentes numa máquina, a partir daquela *word*, basta fazer a operação lógica AND da *word* com o valor FFFFh. Isto irá colocar a zero todos os bits acima do 11. A intenção desta operação é fazer dos bits que queremos ler, os mais significativos de toda a *word*; isto, para que depois se possa dividir por 512 (2^9), o que irá resultar no valor colocado nesses 3 bits. Se a divisão não for inteira (resultando num real), o número irá ter uma parte decimal, devido ao resto dos bits abaixo daqueles que queremos ler.

O Turbo C tem incorporada a função `bioequip()`, que faz o acesso a esta função do DOS e retorna os valores tal como são apresentados aqui.

Além destes serviços, as rotinas dão também informações sobre a memória do computador. A variável **membase** indica o tamanho da memória base ou convencional do computador (até 640 KB), **memext** contém o valor da memória estendida e por fim, **memexp** dá a memória expandida.

Para saber o tamanho da memória convencional do computador, basta utilizar o serviço 18h da interrupção 12h – o valor requisitado será encontrado em AX. No caso da memória estendida, o caso é basicamente o mesmo, basta chamar o serviço 88h da interrupção 15h e o valor é devolvido em AX. Caso haja erro, a *flag* de *carry* é colocada a 1.

O mesmo acontece ao ser usado o serviço 42h da interrupção 67h – memória expandida – no entanto, o erro é detectado no caso de no registo AH existir um valor diferente de zero. Aqui consegue-se saber a quantidade de páginas de memória expandida através de DX, e também a quantidade de páginas disponíveis, em BX (que não é utilizada nas rotinas).

Embora o serviço de memória expandida devolva o valor em páginas (de 16 KB), as rotinas fazem-no sempre em Bytes. No caso do coprocessador e joystick, a rotina envia um zero no caso de não existência e um em caso contrário.

Note-se que a interrupção 67h não faz parte do BIOS/DOS; é uma interrupção criada pelo gestor de memória expandida. Por isso, é necessário verificar se ele está instalado em memória ou não. Quando um gestor de memória expandida se instala em memória, muda o vector de interrupção 67h para o início do seu código. Mas, para ser reconhecido, deixa a assinatura 'EMMXXX0' no offset 0Ah (no mesmo segmento indicado no vector de interrupção). Se esta assinatura não for encontrada, então não foi carregado o gestor de memória expandida e o valor devolvido pela rotina é zero.

Nas versões mais recentes do DOS não é necessário executar esta verificação, pois este vector é inicializado (é reconhecido como *standard*, para EMM) e aponta para uma zona inofensiva da memória, retornando ao ponto inicial, sem mexer no valor de AH (42h), o que no caso do gestor estar instalado, indicaria um erro. Portanto, pode ser chamado à vontade, quer o gestor esteja ou não em memória.

Cenas do próximo capítulo

Até agora, usei os exemplos mais simples para demonstrar a utilidade das funções do BIOS e DOS. No próximo artigo irei falar um pouco mais aprofundadamente sobre elas, dando alguns exemplos mais complicados ou menos convencionais.



TRUQUES E DICAS PARA O WINDOWS

(Parte IV)

por Pedro Andrade

Introdução

Neste número vou, como prometi da última vez, falar sobre a segurança no ambiente Windows.

Segurança no Windows

Se não é o único a utilizar o seu computador e começarem a aparecer algumas anomalias (falta de dados, ícones desaparecidos, etc.), então só podem ter acontecido duas coisas: ou vírus ou alguém andou a mexer...

Para a primeira, o remédio é um anti-vírus, mas para a segunda, o problema é mais complicado: ou se mete um programa de *password* no *boot*, para impedir o total acesso ao computador, ou restringem-se apenas partes.

No Windows, a solução são as restrições. Para criar as restrições, acrescente a secção **[Restrictions]** no seu PROGMAN.INI, e crie as seguintes linhas:

NoRun	= 0
NoFileMenu	= 0
NoClose	= 0
NoSaveSettings	= 0
EditLevel	= 0

A linha **NoRun**, quando alterada para 1, impede a utilização do comando **File. Run** do Program Manager.

NoFileMenu, como o nome indica, impede o acesso ao menu **File**, quando tem o valor 1.

NoClose, quando tem o valor 1, impede o utilizador de sair do Win-

dows. Este parâmetro pode ser útil quando se trabalha num sistema que arranca automaticamente o Windows e onde não se pretende possibilitar a utilização da *prompt* (não esquecer retirar o ícone do MS-DOS Prompt).

A linha **NoSaveSettings**, se alterada para 1, impede a gravação das alterações ao *desktop* (ambiente de trabalho).

EditLevel pode assumir vários valores:

- 0 - Utilização normal do Program Manager
- 1 - Impede a utilização dos comandos **New** e **Delete** do menu **File**, evitando assim a criação e a eliminação de ícones e janelas.
- 2 - Impede a utilização dos comandos **New**, **Copy**, **Move** e **Delete** do menu **File**.
- 3 - Não permite alterações às propriedades **Name** e **Command Line** dos ícones.
- 4 - Impede qualquer alteração às propriedades dos ícones.

Nota: os níveis superiores têm as restrições de todos os anteriores.

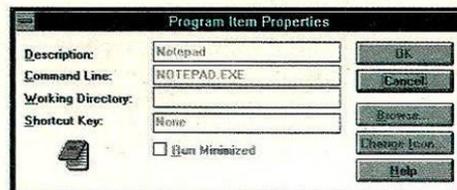


Fig. 1 - EditLevel = 4

- Impede qualquer alteração às propriedades dos ícones.

Ao utilizar um ScreenSaver do Windows, pode configurá-lo com *password*. Assim, ao deixar o seu computador por algum tempo, impede que outros vão olhar o seu trabalho. Para isso, utilize a opção **Desktop** do **Control Panel**.

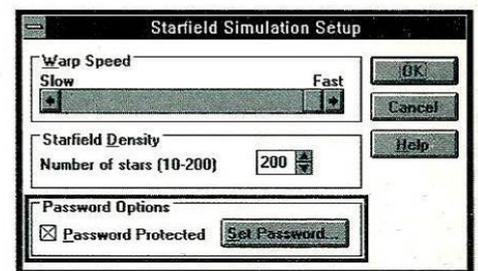
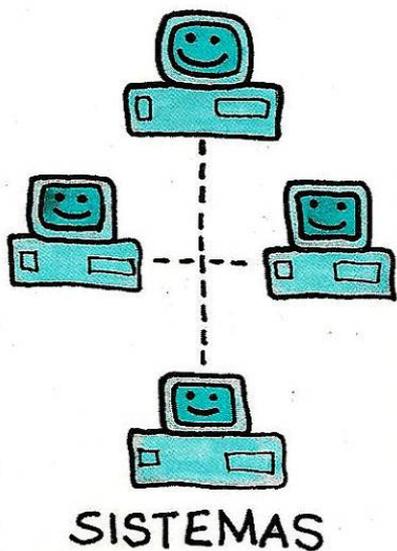


Fig. 2 - Screen Saver protegido por password

No próximo número...

No próximo número vou falar de como resolver os principais problemas que surgem ao utilizar o Windows. Até lá...



ANÁLISE E PROJECTO DE SISTEMAS

O processo mais importante da Informática (Parte VII)

por Ricardo Jorge Santos

Como referi no artigo anterior, vou falar-lhes agora sobre as técnicas de elaboração das Estruturas de Processamento do Projecto Lógico, acabando assim a série de artigos acerca do Projecto Lógico.

As etapas do Projecto de Sistemas: Estruturas de Processamento

Em que consistem as Estruturas de Processamento?

O analista, nesta etapa de desenvolvimento do sistema, deve definir a estruturação do projecto final do sistema, ou seja, a divisão em módulos – modularização – de todo o sistema, incluindo o modo como interagem os seus algoritmos, validações e esquemas de processamento e cálculo de dados, idealizando assim a passagem ao Projecto Físico – programação.

Quais os passos a seguir ao efectuar a Estrutura do Sistema?

Os passos da estruturação do sistema são:

1) Definir a estrutura geral do sistema, isto é, a definição das opções e subopções de tarefas existentes no Menu Principal e con-

seqüentes submenus do sistema. Deve ser tido em conta que o nome da opção deve ser relevante em relação aos procedimentos que executa e ser assim facilmente interpretada pelos utilizadores do sistema. Da mesma forma, os submenus devem ser sequenciados de maneira lógica em relação à sua origem.

2) Devem ser referenciados os *layouts* a utilizar em cada uma das opções do sistema.

3) Devem igualmente ser referenciados os ficheiros a utilizar em cada uma das opções do sistema.

4) Devem ser elaborados os algoritmos com todos os procedimentos-base do sistema, incluindo as validações, controlos e acessos.

5) São enumeradas e esquematizadas as interdependências do sistema.

Um exemplo prático

Suponhamos que existe uma empresa que está a desenvolver um projecto de sistema informático para a sua secção de facturação onde, após ter sido feita a respectiva análise do sistema, vamos supor que resultaram os seguintes dados, resumidos no Projecto Lógico (ver edições anteriores da **Spooler**):

LAYOUTS

Layout Nº	Descrição
1	Layout de <i>output</i> dos menus de opções do sistema
2	Layout de <i>input/output</i> das fichas de dados de clientes
3	Layout de <i>input</i> da introdução de dados para uma factura
4	Layout de <i>output</i> de uma factura
5	Layout de <i>output</i> do mapa de facturação a clientes
6	Layout de <i>output</i> do mapa de clientes

FICHEIROS

1º Ficheiro

Nome do Ficheiro – CLIENTES.DBF

Tipo de Ficheiro – Mestre

Tipo de Organização – Indexada

Nº de Registos – Cerca de 5000

Estrutura do Registo

Campo	Tipo	Tamanho	Posição
COD_CLI	Carácter	06	001-006
NOME	Carácter	30	007-036
ENDERECO	Carácter	60	037-096
CODPOSTAL	Carácter	40	097-136
TOT_FT_MES	Numérico	12(2)	137-148
TOT_FT_ANO	Numérico	12(2)	149-160
NC_CLI	Numérico	10	161-170

Descrição dos Campos

COD_CLI	– Código de cliente
NOME	– Nome do cliente da factura
ENDERECO	– Endereço do cliente
CODPOSTAL	– Código postal do endereço do cliente
TOT_FT_MES	– Valor total facturado ao cliente no corrente mês
TOT_FT_ANO	– Valor total facturado ao cliente no corrente ano
NC_CLI	– Nº de contribuinte do cliente

Significado do registo – Dados da ficha de um cliente.

Comprimento do registo – 170

Índices do ficheiro:

- 1) Ordenado por COD_CLI
- 2) Ordenado por NOME

Validações:

O campo COD_CLI nunca pode ficar vazio e não pode haver 2 registos com o mesmo COD_CLI.

2º Ficheiro

Nome do Ficheiro – FACT_TIT.DBF

Tipo de Ficheiro – Mestre

Tipo de Organização – Indexada

Nº de Registos – Cerca de 50 por dia de trabalho

Estrutura do Registo

Campo	Tipo	Tamanho	Posição
NUM_FACT	Numérico	07	001-007
DATA	Data	08	008-015
NOME_CLI	Carácter	30	016-045
ENDE_CLI	Carácter	60	046-105
CODP_CLI	Carácter	40	106-145

NC_CLI	Numérico	10	146-155
TOT_LIQ	Numérico	10	156-165
TOT_IVA	Numérico	10	166-175
TOT_ADI	Numérico	10	176-185
COD_CLI	Carácter	06	186-191

Descrição dos Campos

NUM_FACT	– Número da factura
DATA	– Data da factura
NOME_CLI	– Nome do cliente em que fica a factura
ENDE_CLI	– Endereço do cliente em que fica a factura
CODP_CLI	– Código postal do endereço do cliente em que fica a factura
NC_CLI	– Nº de contribuinte do cliente em que fica a factura
TOT_LIQ	– Valor total líquido da factura
TOT_IVA	– Valor total do I.V.A. da factura
TOT_ADI	– Valor total dos custos adicionais da factura
COD_CLI	– Código do cliente que dá origem à factura

Significado do registo – Dados de cabeçalho e totais de cada factura.

Comprimento do registo – 191

Índices do ficheiro:

- 1) Ordenado por NUM_FACT
- 2) Ordenado por NOME_CLI + str(NUM_FACT)
- 3) Ordenado por COD_CLI + str(NUM_FACT)

Validações:

O campo NUM_FACT nunca pode ser 0, deve tomar valores sequenciais e não podem haver 2 registos com o mesmo NUM_FACT.

O campo DATA nunca pode ficar vazio, tem sempre de conter uma data válida.

3º Ficheiro

Nome do Ficheiro – FACT_DET.DBF

Tipo de Ficheiro – Mestre

Tipo de Organização – Indexada

Nº de Registos – Cerca de 500 por dia de trabalho

Estrutura do Registo

Campo	Tipo	Tamanho	Posição
NUM_FACT	Numérico	07	001-007
COD_PROD	Carácter	10	008-017
NOME_PROD	Carácter	40	018-057
PREC_UNIT	Numérico	05	058-062
QUANT	Numérico	06	063-068

Descrição dos Campos

NUM_FACT	– Número da factura
----------	---------------------

COD_PROD – Código do produto pertencente à factura
 NOME_PROD – Nome do produto pertencente à factura
 PREC_UNIT – Preço unitário líquido do produto pertencente à factura
 QUANT – Quantidade do produto pertencente à factura

Significado do registo – Dados de venda por produto pertencente à factura.

Comprimento do registo – 68

Índices do ficheiro:

1) Ordenado por str(NUM_FACT) + COD_PROD

Validações:

O campo NUM_FACT nunca pode ser 0 e deve ter a respectiva correspondência com o campo NUM_FACT do ficheiro FACT_TIT.DBF.

Layouts a utilizar em cada opção do sistema

OPÇÃO

Layout Nº

Menu Principal	1
Fichas de Clientes	2
Criação de Facturas	3
Emissão de Facturas	4
Emissão do Mapa de Facturação a Clientes ...	5
Emissão do Mapa de Clientes	6

Ficheiros necessários em cada opção do sistema

Fichas de Clientes – CLIENTES.DBF
 Criação de Facturas
 – CLIENTES.DBF, FACT_TIT.DBF, FACT_DET.DBF
 Emissão de Facturas
 – FACT_TIT.DBF, FACT_DET.DBF
 Emissão do Mapa de Facturação a Clientes
 – FACT_TIT.DBF
 Emissão do Mapa de Clientes – CLIENTES.DBF

Assim sendo, poderemos elaborar as Estruturas de Processamento do Sistema, que ficarão da seguinte forma:

Estrutura Geral do Sistema

Menu Principal

Fichas de Clientes
 Criação de Facturas
 Emissão de Facturas
 Emissão do Mapa de Facturação de Clientes
 Emissão do Mapa de Clientes

Sub-menu da opção Mapa de Facturação de Clientes

Ordenado por Cliente
 Ordenado por Nº de Factura

Sub-Menu da opção Mapa de Clientes

Ordenado por Código de Cliente
 Ordenado por Nome de Cliente

Algoritmos-base do sistema

Aqui deverão ser incluídos os algoritmos de todas as opções com os seus procedimentos e funções importantes e relevantes para o bom funcionamento do sistema. Não escreverei aqui nenhum algoritmo, por questão de espaço. Definirei apenas Algoritmo como sendo uma estrutura de linguagem próxima da Programação em si, mas que contém como instruções pseudocódigo em português para a resolução de problemas. Trata-se de um conjunto de directivas indicando ao(s) programador(es) encarregado(s) do Projecto Físico do sistema como deve(m) dirigir a programação. Esta etapa do Projecto Lógico deve ser elaborada por um Analista-Programador, ou então por um Analista que detenha pelo menos conhecimentos fundamentais de programação, preferencialmente em linguagem de médio ou baixo nível.

Interdependências do sistema

1. O campo TOT_LIQ do ficheiro FACT_TIT.DBF deve corresponder ao somatório de todos os campos PREC_UNIT * QUANT do ficheiro FACT_DET.DBF, para o campo NUM_FACT comum a ambos os ficheiros.

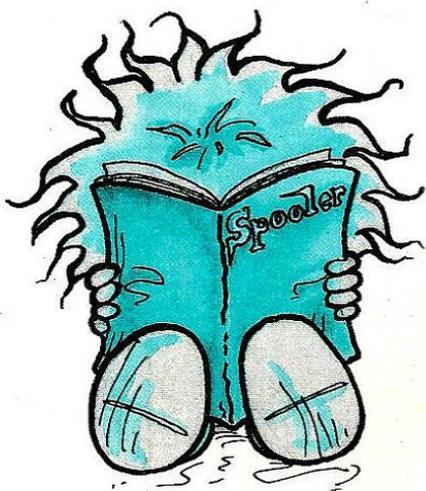
2. O campo TOT_IVA do ficheiro FACT_TIT.DBF corresponde a 17 % do valor registado no campo TOT_LIQ do mesmo ficheiro, ou seja, $0.17 * TOT_LIQ$.

3. O campo TOT_FT_MES do ficheiro CLIENTES.DBF deve corresponder ao somatório de todos os campos TOT_LIQ + TOT_IVA + TOT_ADI do ficheiro FACT_TIT.DBF que dizem respeito a facturas do corrente mês, para o campo COD_CLI comum a ambos os ficheiros.

4. O campo TOT_FT_ANO do ficheiro CLIENTES.DBF deve corresponder ao somatório de todos os campos TOT_LIQ + TOT_IVA + TOT_ADI do ficheiro FACT_TIT.DBF que dizem respeito a facturas do corrente ano, para o campo COD_CLI comum a ambos os ficheiros.

5. Os campos COD_CLI registados no ficheiro FACT_TIT.DBF têm de existir no ficheiro CLIENTES.DBF.

E assim este artigo está terminado. No próximo número da **Spooler** entrarei na série de artigos acerca do Projecto Físico.



**COLABORAÇÃO
DO LEITOR**

NUCLEUS 1.0

TABELA PERIÓDICA

EM PASCAL

Linguagem: Pascal

por Isaac MSC

Estava na aula de Física-Química (chamada físic'ô química) sem nada p'ra fazer, quando abri o livro e vi a tabela periódica. Pensei: «Até podia fazer um programa sobre Química». Passei das palavras à acção (neste caso dos pensamentos à acção) e dois meses e meio depois, tinha o Nucleus completo!

De facto é um programa que pode ser utilizado por todos os amantes da matéria.

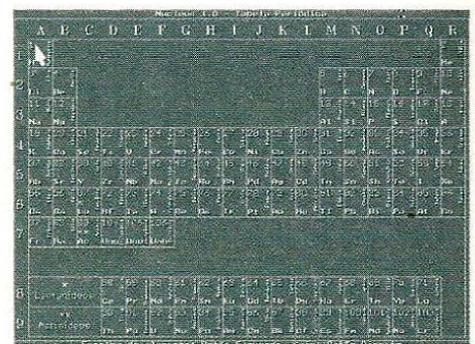
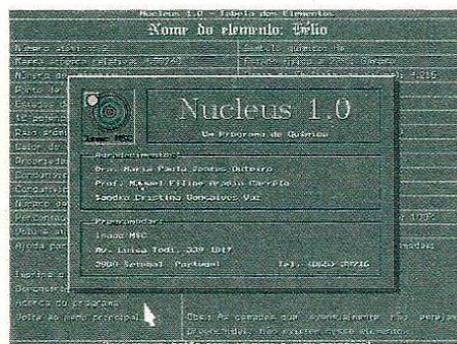
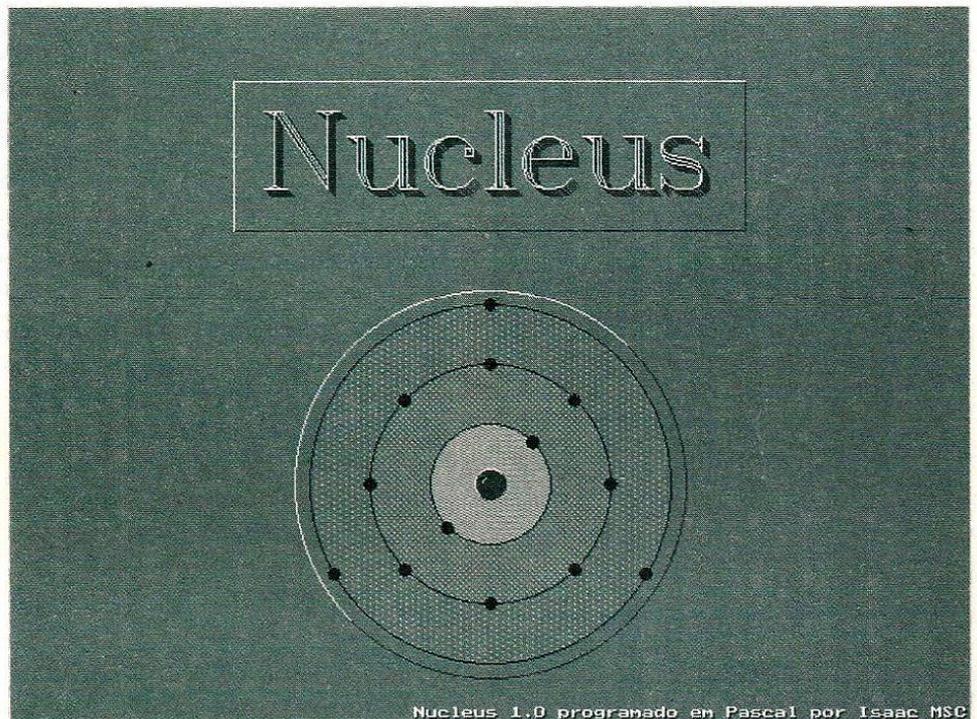
Foi feito em Turbo Pascal 7.0 (TM Borland) e funciona em modo gráfico VGA 640x480x16.

Está, como o Nutritus (programa já abordado nesta rubrica), com os *drivers* e fontes compilados no executável (é aquela coisa com o BINOBJ).

Ao correr o programa aparece o ecrã de apresentação, depois a tabela periódica com os 106 elementos mais conhecidos. Para escolher um elemento, move-se o rato até ao elemento escolhido e prime-se o botão esquerdo, que vai listar mais de 24 dados diferentes sobre esse elemento. A partir de aí, é fácil. Ah, também tem o *Help*, que pode ser chamado através do ecrã «Nucleus 1.0 - Tabela Periódica» com a tecla F1.

Também acompanhando o Nucleus vem o EXEMPLO.PAS que, desta vez, é um programa com um menu seleccionável através do rato. Está mais uma vez comentado.

Espero que dêem bastante traba-

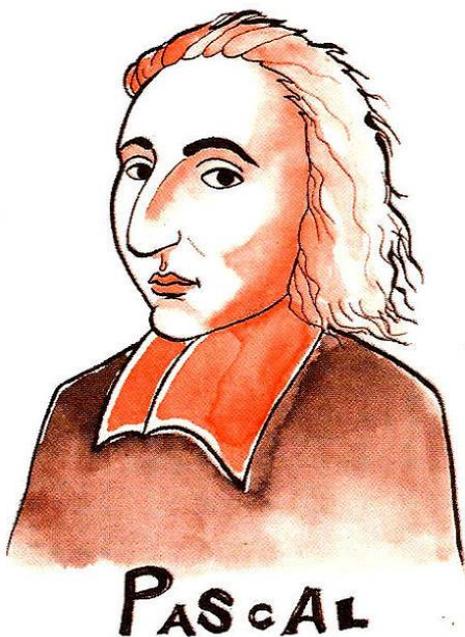


lho ao programa: ele não se cansa, Ok?

Numa próxima oportunidade:

MINÉRIUS: um programa sobre mineralogia, com os dados relativos ao reconhecimento de mais de 150 minerais diferentes e várias outras

funções: técnicas de criação de cores (mais de 260 mil), biblioteca de estruturas cristalinas tri-dimensionais, etc.



BATCH TOOLS (Parte I)

AS FERRAMENTAS

PARA FICHEIROS BATCH

Linguagem: Pascal *por Luis Manuel Santos Guerra*

Este é meu primeiro contributo para a revista **Spooler**, à qual agradeço desde já pela oportunidade que oferece aos leitores de participarem na sua elaboração. Desta vez, com o meu contributo, pretendo apresentar programas que sejam mais úteis do ponto de vista do utilizador, e não tanto na complexidade das rotinas elaboradas em Pascal.

Todos os utilitários divulgados neste e próximos artigos destinam-se a melhorar a apresentação e construção de *batch files*. E apesar de serem pouco flexíveis, as *batch files* são indispensáveis para necessidades do dia-a-dia: são fáceis de alterar e para as criar basta um programa vulgar de edição de texto. Pena que a «linguagem» *batch* nem sempre tenha a atenção que merece, apesar de utilitários como BAT2EXEX e BE terem em muito alargado as aplicações possíveis com *batch files*.

Com os programas em seguida apresentados, a minha meta é preencher uma lacuna. Se conseguir motivar ao uso de *batch files*, os meus objectivos serão alcançados. Neste momento muitos de vós estarão a torcer o nariz e a pensar: «Para que preciso de rotinas de *batch*?». Só peço que vejam o que tenho para dizer e experimentem os exemplos mostrados.

Apesar da sua extrema simplicidade de construção, o seguinte utilitário é (o meu objectivo principal)

muito útil e fácil de usar: falo do Batch Tools, cuja versão beta fora concluída no final de Março de 1994 e do qual fiz uma revisão para apresentar na **Spooler**. O nome diz tudo; espero que com ele consiga ganhar a preferência dos utilizadores ferrenhos do BE. Com essa convicção, embora mantendo um código simples de programação, apresentarei no próximo artigo o BT2, que trará mais algumas funções. No total, são 29 funções do BT para rotinas *batch*.

Para iniciar o seu uso, basta tecer BT ou BT2 e apresenta-se a lista de todos os comandos disponíveis. Depois, ao acrescentar o nome de um comando qualquer, apresenta-se a sintaxe e a descrição dos parâmetros.

Antes de se conseguir assimilar todas as funções, aconselho os leitores a correrem as rotinas *batch* de demonstração, para terem uma ideia das capacidades de Batch Tools. Mesmo assim a leitura completa dos artigos afigura-se indispensável para a compreensão dos utilitários.

Construí o BT para agrupar vários utilitários que já tinha criado. Ao ver que o Batch Enhancer era semelhante, decidi melhorar o BT e fazer o BT2 e BT3. Só para poder livrar-me do estigma BE. Os utilizadores de Batch Tools verificarão se assim é ou não. À data da publicação deste artigo o BT3 já deverá estar isento de *bugs*. Posso garantir que não se tra-

ta de «mais uma sequela» e espero incluir novidades em relação a versões anteriores. De qualquer modo, 12 funções estão já completas. Com esta versão devo fechar a trilogia... Talvez...

Embora não tenha ficado documentado, nem tenha feito qualquer espécie de referência nos manuais, o BTv1.1 e BT2v1.0 só podem ser utilizados correctamente em modos de texto com 80 colunas (80x25, 80x43, 80x50).

Para terminar, lembro que devem editar o ficheiro BT1.TXT incluído na disquete **Spooler**, onde é feita uma descrição pormenorizada de todos os ficheiros.

Até a uma próxima oportunidade!



SHAREWARE (Parte IX)

CRÍTICA E DIVULGAÇÃO

DO SHAREWARE

NACIONAL

por Carlos Batista e Tiago Pascoal

Neste número vamos apreciar mais um programa do já sobejamente conhecido nestes páginas, o Vasco Anjos. Desta feita, trazemos aqui um editor de texto.

O editor tem uma apresentação gráfica bastante cuidada, possuindo menus *pull-down* e um ambiente com janelas ao estilo do Windows.

Se quiser pode modificar as cores do ambiente, contudo só pode gravar as cores que seleccionou na versão registada. Como seria de esperar no ambiente orientado por janelas, o rato é totalmente utilizável, quer seja para executar funções existentes nos menus, ou para marcar, recortar, colar ou copiar texto; contudo este não pode ser usado ao estilo de «Drag & Drop».

Ao contrário do que seria de esperar, este editor não suporta combinações de teclas Wordstar. Embora esta combinação de teclas seja considerada pelos novatos ou por pessoas que entraram no mundo da informática já no tempo das interfaces gráficas e dos menus *pull-down*, totalmente desprovidas de lógica (na verdade até o são) muita gente habituou-se a estas e são bastante eficientes. Estranhámos a sua inexistência, porque são suportadas pela grande maioria de editores existentes no mercado.

Este editor permite que se desenhem molduras ou linhas, contudo não as pode desenhar com os cursores como na maioria dos editores, terá que escolher o tamanho da caixa (ou do número de caracteres que compõem a linha) e o tipo de caracteres que constituem a caixa (ou linha) e o editor fará o resto. Se para

caixas simples estas funções servem perfeitamente, para molduras menos regulares ou até para quadros, esta função torna-se totalmente inútil.

Se desejar pode incluir automaticamente no seu texto várias coisas, tais como a data do dia, a hora actual ou até mesmo um calendário do mês e ano que desejar; pode ainda introduzir macros que durante a impressão serão substituídos pelos respectivos códigos da impressora (são suportados três tipos de impressoras: IBM, HP com PCL, e Epson com ESC/P). Estes macros permitem atribuir ao texto a imprimir atributos tais como o *Bold*, o *Itálico*, o *Superscript*, etc.

Existe também a opção de paginar os textos que estiver a editar, sendo possível optar por páginas entre 10 a 66 linhas caso escolha uma paginação automática ou pode ainda utilizar uma paginação manual, podendo definir o fim de página sempre que quiser. Ao optar por um tipo de paginação, pode depois pedir ao editor que o coloque numa página à sua escolha.

O editor possui ainda uma função nada usual neste tipo de programas: uma opção para encriptar ficheiros. Não percebemos muito bem qual a utilidade desta opção, mas enfim, a sua utilização e utilidade fica ao critério do utilizador do programa.

Como é habitual num editor de texto, este possui funções de procura e de substituição.

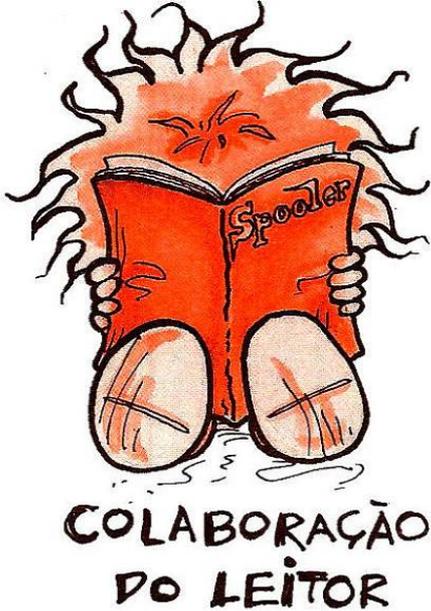
Pouco mais há a dizer. Durante os ensaios foram encontrados alguns *bugs*, e o programa tem uma limitação muito grande e totalmente incompreensível: o tamanho dos fi-

cheiros a editar está limitado a 32Kb! Ora isto nos dias que correm é totalmente ridículo; o executável é também bastante grande e o editor é um bocado lento para as características que possui. É caso para dizer, «muita parra e pouca uva», pois o programador concentrou-se demais na interface e desleixou-se no número de coisas que o programa faz. Hoje em dia qualquer editor que se preze possui uma linguagem de macros por mais simples que esta seja, a não ser que seja um editor para trabalhos casuais.

Portanto, se necessita de um editor para algo mais do que editar um ficheiro casualmente, achamos que este programa não é adequado para tal; contudo, se apenas necessita de um editor para editar um ficheiro uma vez por outra, também não necessita deste editor, pois o MS-DOS é fornecido com algo que chega perfeitamente para isso.

O autor menciona o facto deste editor ir integrar num futuro próximo o programa SECRETARIO; aí já estamos de acordo que será útil para editar textos de pequenas dimensões, mas achamos que registar este editor individualmente não vale a pena, dado existirem já no mercado muitos editores (até *freeware*) que lhe são muito superiores.

Terminamos, mais uma vez, com um apelo. Enviem-nos os vossos programas para que estes possam ser divulgados. Esta coluna é dedicada ao *shareware* português e existe para que este seja divulgado condignamente.



REBOOT v1.1

UM REBOOT COM TRÊS

PALADARES

Linguagem: Assembly por Carlos Jorge Santos

Para começar um abraço de parabéns a quantos na **Spooler** contribuem para fazer uma revista espectacular.

Olá a todos, chamo-me Carlos Jorge Santos, vivo em Faro, tenho 22 anos e ando no 3º Ano de Informática de Sistemas na Universidade do Algarve.

Envio aqui um pequeno utilitário para fazer REBOOT. Já devem estar a pensar, «este tipo pensa que descobriu a pólvora, basta carregar no RESET ou fazer CTRL-ALT-DEL». Pois é, mas e se não estivermos ao pé do computador para fazer um *reboot*?, pensem por exemplo nas BBS, que passam noites inteiras a trabalhar, sem ninguém ao pé para fazer um *reboot* caso algo corra mal. Este é apenas um exemplo.

Assim sendo, aqui entra o REBOOT v1.1 em acção. Possui 3 (três) sabores diferentes. Um *reboot* a frio, a quente e inicia o BOOTSTRAP LOADER.

O REBOOT v1.1 foi feito em *assembly* como exercício de programação, devido a isso ocupa somente 300 e poucos bytes. Utilizei o A86 v3.22, um excelente assembler *shareware* (que adquiri através da **Spooler**), o qual dispensa todas aquelas «chatas» directivas de compilação, tornando mais fácil programar em *assembly*, como podem ver pela listagem.

Utilização

Muito fácil, basta indicar na linha de comando o tipo de *reboot* desejado.

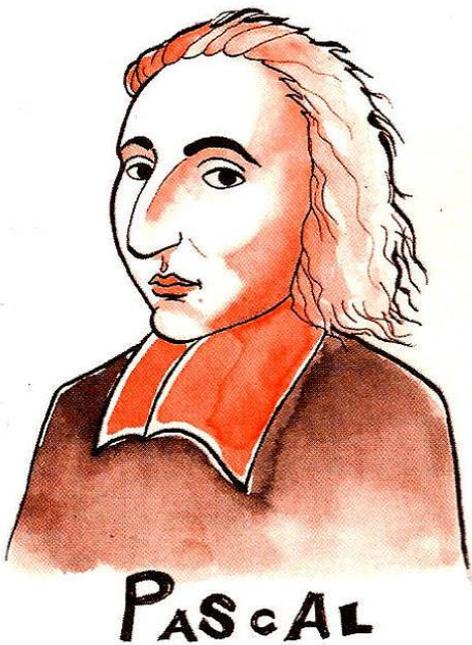
REBOOT /F *Reboot* a frio, limpa a memória, restaura os vectores de interrupção e faz todos os testes ao *hardware*. (Equivalente ao botão de RESET).

REBOOT /Q *Reboot* a quente, limpa a memória, restaura os vectores de interrupção, mas não faz testes ao *hardware*. (Equivalente a CTRL-ALT-DEL).

REBOOT /B Inicia o BOOTSTRAP LOADER, ou seja, lê o sector de *boot* da disquete ou do disco para a memória em 0000:7C00h e executa. Assim são contornados os testes ao *hardware* e não são restaurados os vectores de interrupção (o que por vezes provoca o CRASH do sistema).

Acho que é tudo. Quaisquer dúvidas, críticas, sugestões, ou se simplesmente vos apetecer escrever, façam-no para:

Carlos Jorge Viegas dos Santos
R. Dr. José Neves Júnior, Lote 14 – 1º Esq.
8000 Faro
Telf. (089) 29674



PASCAL

CDIR MANIPULAÇÃO DE DIRECTÓRIOS

Linguagem: Pascal

por Pedro Sinogas

Criei o CDIR para responder às minhas necessidades como utilizador da linha de comando do DOS.

Baseado no QCD (*Quick Change Directory*) da CBLong que, apesar de ser bastante prático não permite mudar quer de *drive* quer para a raiz, no PDIR (*Paul's Directory*) de Paul Askew que permite listar o conteúdo de um directório com uma cor diferente para cada tipo de ficheiro e no NCD (*Norton Change Directory*) da Symantec, que permite através da linha de comando mudar para qualquer subdirectório, dadas as suas primeiras letras, implementei no CDIR variantes das propriedades citadas, à medida das necessidades.

O CDIR funciona com as teclas de controlo (Setas, PageUP, PageDown, Home e End) que permitem navegar pelos directórios e *drives*, bem como listar o seu conteúdo, através do Space.

Ao listar o conteúdo de um directório, o CDIR dá, com as cores escolhidas, as seguintes informações sobre cada ficheiro: nome, tamanho, data, hora e atributos (*Archive*, *System*, *Hidden* e *Read-Only*).

Enquanto navega pelos directórios, se carregar em alguma tecla, o CDIR seleccionará o primeiro directório por ela começado.

Durante a utilização do CDIR este pode abandonar-se para o directório seleccionado com a tecla <Enter> ou



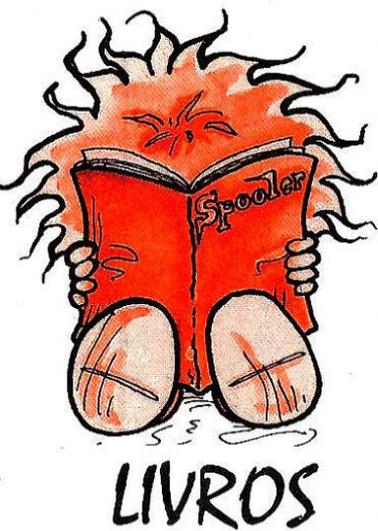
<Escape> para voltar ao directório inicial.

Com o parâmetro /C o CDIR permite ao utilizador seleccionar as cores em que serão mostradas as informações, os ficheiros EXE, COM, BAT, TXT, outros ficheiros e subdirectórios. Para este parâmetro e com o objectivo do CDIR não necessitar de qualquer ficheiro externo para configuração, o utilizador tem de se

posicionar no directório em que se encontra o ficheiro CDIR.EXE, onde será gravada a configuração.

Por fim com o parâmetro /? é mostrado ao utilizador um ecrã com todas as informações necessárias para melhor utilizar o CDIR.

Espero que o CDIR vos seja tão útil quanto o é a mim.



LIVROS

Estes e outros livros sobre informática editados pela Sybex, QUE, Ziff, Editorial O Livro, FCA, Edições Cetop, McGRAW-HILL e Graficria, podem ser encomendados através da Montra Spooler. Para o efeito utilize o cupão respectivo.

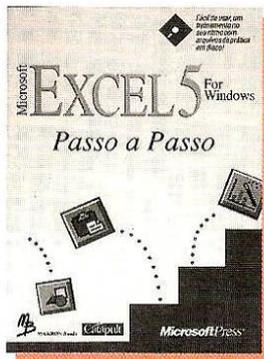
Os assinantes da Spooler beneficiam de 10% de desconto.

Os preços aqui indicados poderão sofrer alterações por parte dos editores/distribuidores.

Microsoft Excel 5 for Windows

As lições acessíveis deste livro fornecem uma introdução perfeita aos novos recursos da versão 5 do Excel. Permitem que aprenda as habilidades do computador fora do ambiente tradicional da escola.

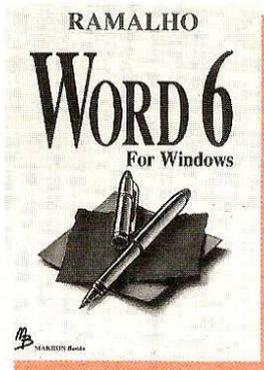
Título: Microsoft Excel 5 for Windows
Autores: Catapult, Inc
Idioma: Tradução brasileira
Editores: MAKRON Books
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 7 600\$00 + 5% de IVA
ISBN: 3460260-3



Word 6 for Windows

Dividido em 21 capítulos, o livro tem a finalidade de ensinar o leitor a utilizar os principais recursos deste processador de texto.

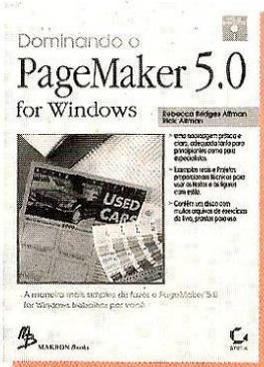
Título: Word 6 for Windows
Autor: José Antônio Ramalho
Idioma: Português
Editores: MAKRON Books
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 4 600\$00 + 5% de IVA
ISBN: 3460249-2



Dominando o PageMaker 5.0 for Windows

Esta obra constitui um guia prático e único, combinando lições passo-a-passo para os principiantes com conceitos avançados para os experientes.

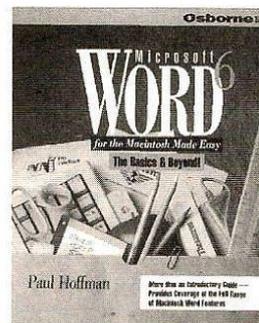
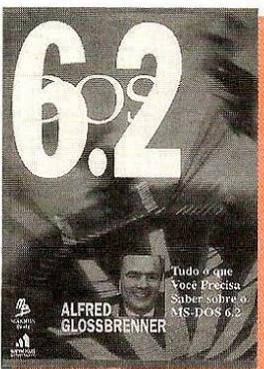
Título: Dominando o PageMaker 5.0 for Windows
Autores: Rebecca Bridges Altman e Rick Altman
Idioma: Tradução brasileira
Editores: MAKRON Books
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 8 400\$00 + 5% de IVA
ISBN: 3460279-4



DOS 6.2

Numa linguagem clara e não técnica, o autor apresenta todos os recursos desta nova versão, desde os comandos básicos até aos aperfeiçoamentos mais sofisticados.

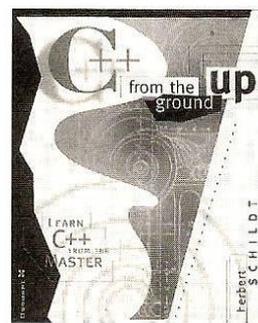
Título: DOS 6.2
Autor: Alfred Glossbrenner
Idioma: Tradução brasileira
Editores: MAKRON Books
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 7 600\$00 + 5% de IVA
ISBN: 3460235-0



Microsoft Word 6 for the Macintosh Made Easy

Mais do que um guia introdutório, este livro facultava uma cobertura de todas as características do Word for Macintosh.

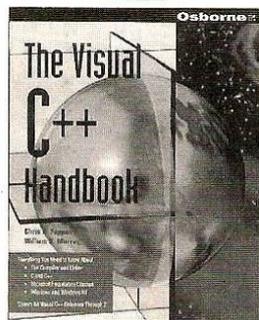
Título: Microsoft Word 6 for the Macintosh Made Easy
Autor: Paul Hiffman
Idioma: Inglês
Editores: Osborne McGRAW-HILL
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 3 990\$00 (IVA incluído)
ISBN: 882065-0



C++ from the Ground

Tudo o que é necessário aprender sobre o C++ encontra-se nesta obra, escrita pelo famoso especialista Herb Schildt.

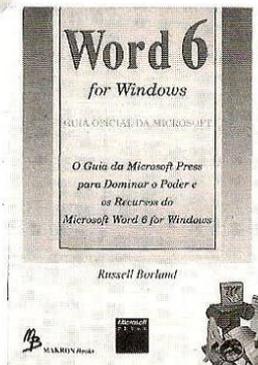
Título: C++ from the Ground
Autor: Herb Schildt
Idioma: Inglês
Editores: Osborne McGRAW-HILL
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 4 260\$00 (IVA incluído)
ISBN: 881969-5



The Visual C++ Handbook

Se o leitor é programador, ou deseja programar em Visual C++, este é o livro indicado. Os especialistas Chris Pappas e William Murray construíram uma abordagem modular para o ensino desta linguagem aos programadores de todos os níveis, desejando programar para Windows, Windows NT ou Win32.

Título: The Visual C++ Handbook
Autores: Chris Pappas e William H. Murray
Idioma: Inglês
Editores: Osborne McGRAW-HILL
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 6 720\$00 (IVA incluído)
ISBN: 882056-1



Word 6 for Windows - Guia oficial da Microsoft

Eis as informações que o leitor precisa para obter o máximo proveito das capacidades da nova versão do Word.

Título: Word 6 for Windows - Guia oficial da Microsoft
Autor: Russel Borland
Idioma: Tradução brasileira
Editores: MAKRON Books
Distribuidora: McGRAW-HILL de Portugal, Lda.
Preço: 10 080\$00 (IVA incluído)
ISBN: 3460244-1

Montra Spooler

IMPORTANTE

A partir deste número, a «Montra Spooler» sofre algumas alterações, a saber:

- 1 - As encomendas deixam de ser enviadas à cobrança, o pedido deve ser acompanhado do respectivo pagamento em cheque. Com isto o leitor poupa os encargos da taxa de cobrança e a Spooler não suporta os prejuízos de encomendas não reclamadas e que nos são devolvidas.
- 2 - Para cada Cupão de Encomenda foi estipulada uma taxa média de portes no valor de 450\$00. Desta forma o leitor saberá exactamente a quantia que irá dispendir.
- 3 - Para facilitar ainda mais, cada revista será acompanhada de um sobrescrito que não precisa de selo, onde o leitor enviará o seu cupão de encomenda e respectivo cheque. Este sobrescrito só pode ser utilizado para este efeito ou para a inscrição como assinante da revista, ou renovação da assinatura.
- 4 - Se por ruptura de stocks ou outro motivo, não pudermos satisfazer alguma das encomendas, devolveremos, de imediato o respectivo valor.
- 5 - Os assinantes da Spooler passam a usufruir do desconto especial de 20% sobre os livros «Edições Graficria» e sobre todo o «Shareware». Como foi diminuída a margem de lucro dos restantes produtos, não nos é possível praticar qualquer desconto.

TODOS OS PREÇOS INCLUEM IVA.

DRIVE IOMEGA ZIP

NOVO



Acabou-se a «tragédia» que constituía transportar e arquivar grandes quantidades de informação.

As drives Zip, completamente compatíveis com sistemas DOS/WINDOWS e Macintosh, apenas com 450 g de peso, utilizam discos amovíveis de 100 Mb ou 25 Mb.

A instalação é extremamente simples. As ferramentas Zip tornam fáceis todas as tarefas de arquivo, organização e pesquisa de dados. Pode executar os seus programas a partir da drive ZIP.

Seek time: 29 milissegundos. Vel. Transf. dados 1,25 Mb/seg.

O Kit contém: Drive ZIP e disco de 100 Mb. Ligação à porta paralela. Opcional: Placa ZIP SCSI-2.

Ref. NOU01	Cada disco ZIP suplementar de 100 Mb	4 224\$00
Ref. NOU02	Placa ZIP SCSI-2	13 923\$00
Ref. NOU03	Drive ZIP + 1 disco de 100 Mb	39 900\$00

PASTA ARQUIVADORA SPOOLER

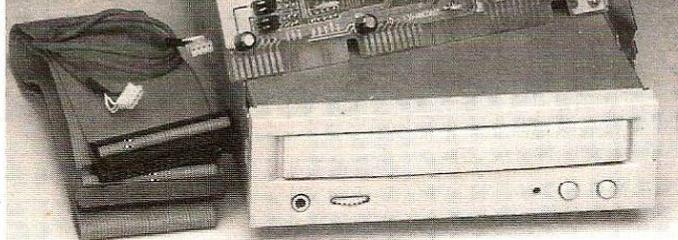
Com as pastas Spooler poderá, finalmente, arquivar de forma simples e prática as suas revistas.

O seu engenhoso sistema de arquivo permitir-lhe-á colocá-las e retirá-las sem as danificar. As revistas não precisam ser perfuradas.

CD-ROM ZETA STARTER PACK

NOVO

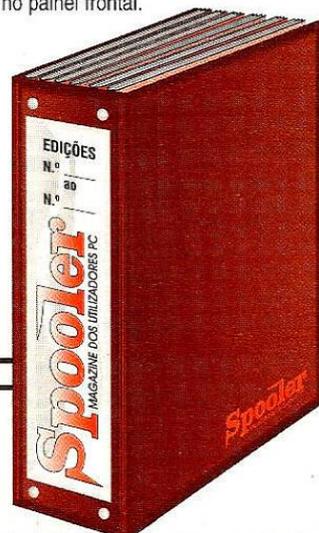
A última palavra em drives de CD-ROM.



- Compatibilidade total com Kodak Photo CD, sessões simples ou múltiplas.
- Verdadeiro desempenho MPCII
- Botões para reprodução rápida de CDs no painel frontal.
- Compatibilidade CD-i para reprodução MPEG.
- Compatibilidade com CD Áudio e CD Datastream.
- Mecanismo motorizado de abertura, silencioso e suave.
- Velocidade de transferência dupla: 300 Kb/seg.
- Saída para auscultadores e controlo de volume de som no painel frontal.

Ref. ZET01 24 716\$00

Ref. CAPS_1
Preço: 2 800\$00



DISQUETAS 3,5" PRÉ-FORMATADAS

Ref. DF62	Disq. 3,5" HD - Cx. 10 unid. com etiquetas	863\$00
Ref. DF52	Disq. 3,5" DD - Cx. 10 unid. com etiquetas	669\$00
Ref. DF67	Disq. 3,5" HD - Cx. 50 unid. sem etiquetas	3 993\$00

ETIQUETAS P/ DISQUETAS 3,5"

Ref. DF04	Conjunto 50 etiquetas	366\$00
-----------	-----------------------	---------

OCTPUS VTD 2000

NOVO

Drive para computador utilizando gravador de vídeo.

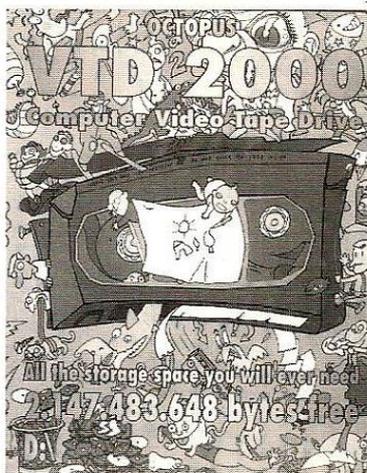
Cá está um meio económico, simples e eficaz de armazenar os seus dados. Basta possuir um gravador de vídeo normal e esta revolucionária drive.

O VTD 2000 toma o lugar do próximo drive livre do seu PC, funcionando no MS-DOS ou no Windows como se fosse um disco de alta capacidade. A capacidade de armazenamento é de 600 a 2000 Mb, dependendo da duração da cassete de vídeo utilizada.

A gravação e leitura é feita usando os comandos ou programas já conhecidos do MS-DOS ou Windows tais como COPY, XCOPY, BACKUP, RESTORE, DOS-SHELL, PKZIP, PKUNZIP, WINDOWS FILE MANAGER, WINDOWS BACKUP, etc.

O gravador de vídeo é comandado por um emissor de infra-vermelhos, sendo a busca, leitura e escrita dos dados feitas de forma automática.

Os programas armazenados na fita podem ser executados directamente sem necessidade de os copiar para o disco rígido. A velocidade de transferência é bastante superior à das disquetes.

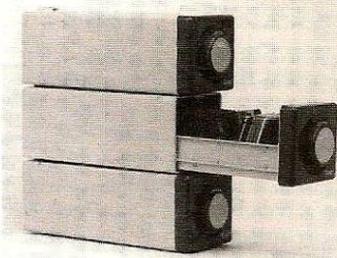


Ref.^a. OCT01

31 520\$00

CAIXA ARQ. P/ 37 DISQUETAS DE 3,5"

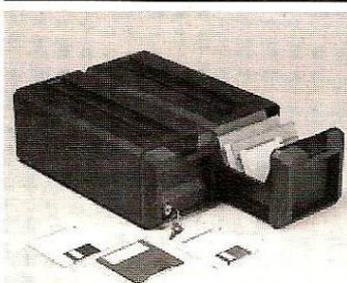
Esta funcional caixa arquivadora com capacidade para 37 disquetes de 3,5" vem resolver muitos dos problemas com o acondicionamento de software. A caixa é do género gaveta e podem sobrepor-se várias caixas, criando uma espécie de estante arquivadora. O preço é incrivelmente baixo.



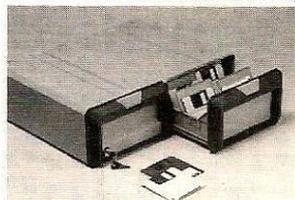
Ref. CX37

1 463\$00

ARQUIVOS MEDIA BOX



Se já possui uma quantidade apreciável de disquetes, estas caixas arquivadoras dão a melhor e a mais económica solução.



Ref. POS1	75 disquetes 3,5"	2 950\$00
Ref. POS2	150 disquetes 3,5"	5 135\$00



COLUNAS FX 20

Colunas estereofónicas AZTECH para PC. 4 W potência de saída, freq. de resposta de 100-15.000 Hz, 8 ohms impedância, 4 pilhas formato C, 6 volts - não incluídas. Pode usar adaptador AC/DC. Cabo com 1,2m.

Ref.^a. FX20

3 100\$00

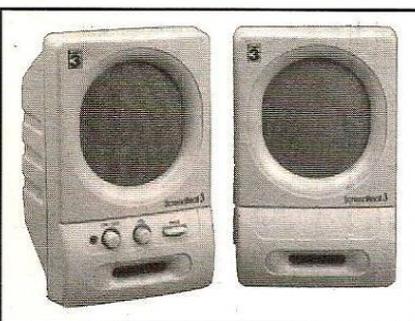
PORTA RASCUNHOS

Útil dispositivo para segurar documentos perto do seu campo de visão.



Ref.^a. PR02
Preço: 432\$00

COLUNAS SB 203

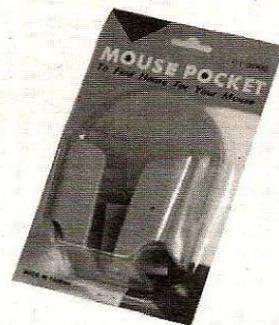


Colunas estereofónicas LOGIC 3 para PC. Circuito «booster». 10 W saída, freq. resposta de 50 - 15 000 Hz, 4 ohms impedância. 4 pilhas formato AA, 1,5 volts - não incluídas. Pode usar adaptador AC/DC.

Ref.^a. SB203
Preço: 4 183\$00

SUPOORTE PARA RATO

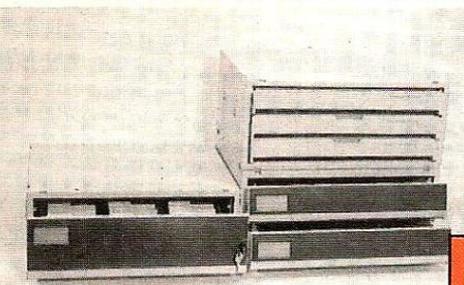
Não abandone o seu rato quando não precisa dele. Esta caixa foi especialmente concebida para acondicionar a «fera». Colocação com pastilha auto-colante.



Ref.^a. PR01
Preço: 132\$00

GAVETAS PARA PAPEL E DISQUETAS

NOVO



Gaveta de arquivo para 260 disquetes de 3,5" com chave.

Ref.^a. CHA01
Preço: 5 777\$00

FILTRO PARA MONITOR

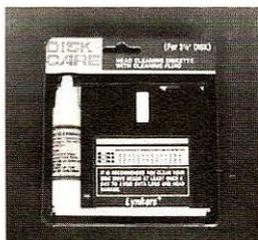
NOVO



Filtro de vidro anti-radiações, anti-reflexo e anti-estático. Para monitor 12"/14".

Ref.^o VIDR1
Preço: 9 510\$00

KITS PARA LIMPEZA DE DRIVES



Kits de limpeza de drives incluindo disquete e líquido de limpeza. Qualidade comprovada.

Ref.^o G001 Disquete 3 1/2 220\$00

RATO MITSUMI



Melhor e mais barato, é impossível!

- 400 dpi
- Dois botões

Ref.^o MIT1 1 536\$00

JOYSTICK

NOVO

Joystick para IBM e compatíveis.

- 2 botões de disparo
- 2 botões de afinação
- Tiro automático



Ref.^o ARO2
Preço: 2 195\$00

QUICKJOY PC-PACK (SV-203) - PC



Joystick para compatíveis PC/XT/AT e placa. Ótimo desempenho e *design* atractivo. Um novo dinamismo nos seus jogos predilectos. Um preço incrível!

Ref.^o G003
Preço: 3 200\$00

TAPETE PARA MOUSE

Tapete para Mouse de elevada qualidade. Anti-estático. Permite-lhe uma maior precisão e facilidade no seu trabalho.

Ref.^o G05F TAPETE FLEXÍVEL 145\$00

JOYSTICK GAME CARD - PC

Placa para joystick, para os computadores que não disponham de porta própria.

Ref.^o G04A 2 194\$00

PAPEL CONTÍNUO

500 folhas A4 (zebra americana) 12x9,5"

Ref.^o P020 761\$00

PROGRAMA FACTURA 2.0 (nova versão)

Este programa de fácil utilização, instalação simples, textos de ajuda claros e precisos, menus *pull-down*, obedece a todos os requisitos legais em vigor e proporciona os melhores resultados para a sua Empresa, Negócio ou actividade profissional. Dispõe de rotinas de impressão para os tipos mais comuns de impressoras, incluindo impressoras Laser. Não precisa de utilizar formulários impressos; o programa gera todo o desenho dos documentos a imprimir.

- Criar e manter actualizados dados sobre clientes, fornecedores e artigos;
- Gestão de stocks;
- Produção de propostas, proformas e documentos de transporte de materiais;
- Mailing personalizado;
- Gestão de contas bancárias;
- Visualização ou impressão de listagens organizadas pelo utilizador;
- Utilitários diversos tais como: Etiquetas, Planeamento, Seguranças, Reposições e Formatação de disquetes.

OBEDECE A TODAS AS DISPOSIÇÕES LEGAIS EM VIGOR

Com garantia e assistência técnica ONLINE. Produzido por XT-Software, Lda.

Ref.^o FT20 23 400\$00

PROGRAMA AGENDA

AGENDA é a solução para quem pretender organizar e rentabilizar o seu tempo.

PRINCIPAIS CARACTERÍSTICAS:

- **Lista de endereços e telefones:** Número ilimitado de fichas contendo informações diversas sobre empresas ou personalidades.
- **Agenda:** Marcação e visualização de entrevistas, chamadas telefónicas e tarefas.
- **Contas bancárias:** Simples mas eficiente controlo de todas as suas contas bancárias, incluindo gestão de cheques pré-datados e letras.
- **Editor de texto:** Faça os seus próprios documentos, sem necessidade de processadores de texto convencionais. Ideal para mailings, cartas, etc.
- **Etiquetas:** Emissão de etiquetas a partir da base de dados de endereço e com seleções variadas.
- **Plano de tesouraria:** Controlo de pagamentos e recebimentos.
- **Utilitários:** Calculadora, calendário, bases de dados de utilidade pública.

Com garantia e assistência técnica ONLINE. Produzido por XT-Software, Lda.

Ref.^o AG01 4 387\$00

MODEMS / FAX ZOOM

Ref. ^o Z001	Modem/Fax 2400/9600 interno	14 391\$00
Ref. ^o Z002	Modem/Fax 14400/14400 interno	24 863\$00
Ref. ^o Z003	Modem/Fax 14400/14400 externo	35 100\$00
Ref. ^o Z004	Modem/Fax 28800/14400 Interno	43 875\$00

PROTECTOR DE TENSÃO CA-365



Caixa com 8 tomadas, 4 com ligação à terra. Filtro para eliminar os «picos» temporários da alimentação e ruídos dos circuitos de luz eléctricos. Disjuntor 10A e interruptor.

Ref.^a. CA36
Preço: 5 412\$00

RATO MIC - MI-260



O rato MIC é um *best-seller!* Tão bom como os melhores e o mais barato de todos!

Características: 260-2600 DPI, Compatível Microsoft e Mouse System, com comutador no corpo do rato. 3 teclas.

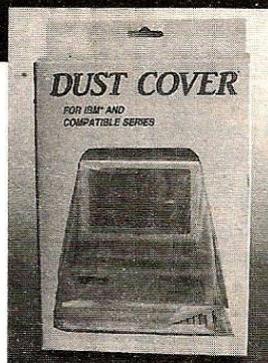
Inclui: Software de instalação e teste, ficheiros de ajuda (em inglês).

Ref.^a. MIC1
Preço: 2 194\$00

CAPA PROTECTORA PARA PCs E IMPRESSORAS

Capa protectora para computadores em plástico especial reforçado para a completa protecção do seu equipamento contra fumos, poeiras, humidade e sujidades de toda a espécie.

Adaptável a qualquer tipo de PC e impressoras.



Ref. ^a . G006	Capa para PC	629\$00
Ref. ^a . G007	Capa p/ imp. 80 c.	293\$00
Ref. ^a . G008	Capa p/ imp. 132 c.	336\$00
Ref. ^a . HP109	Capa p/ imp. HP Deskjet	475\$00

FITAS, TONNERS E TINTEIROS PARA IMPRESSORAS

PARA TODOS OS TIPOS DE IMPRESSORAS (E MÁQ. DE ESCREVER)

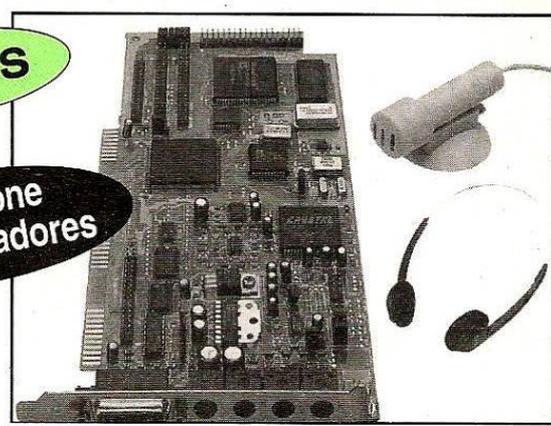
MARCAS ORIGINAIS E COMPATÍVEIS.

Indique no cupão de encomenda a referência que pretende (ou a marca e modelo da impressora)

TEMOS OS MELHORES PREÇOS.

SOUND GALAXY WAVERIDER 32+ NOVO

GRÁTIS
Microfone e Auscultadores



- Processador Wave para a reprodução realística dos sons sintéticos.
- 128 instrumentos e 69 sons de percussão.
- Reproduz 32 vozes em simultâneo + 20 vozes Yamaha OPL3
- Compatível com os principais standars: SoundBlaster™, SoundBlaster™ Pro, Microsoft™ Windows Sound System™, AdLib™ + Drives CD-ROM IDE.
- Compatibilidade MIDI: interface Roland MPU 401, MT-32 standard, MIDI standard.
- Suporta todos os tipos de fontes sonoras: Audio Digital, Síntese FM, Microfone, Entrada de linha CD-Audio, Entrada de linha.
- Porta para conexão de joystick

Software incluído: COMVOICE - Programa para reconhecimento de voz; MONOLOGUE - Para o computador ler os textos que vão passando no ecrã; CLEF STUDIO - colecção completa de programas para controlar a sua placa de som; CLEF EDIT - Editor de Audio Digital, para manipulação de Waves; CLEF OLE - Para «embeber» objectos sonoros em documentos compatíveis OLE; UTILITIES - Programa para diagnóstico/ teste, configuração, controlo de volume, mistura áudio e device drivers; WINDOWS DRIVERS.

Ref.^a. SGAL1

29 400\$00

FAXMATE 3000



O FAXMATE 3000 permite operar Fax/modem, Telefone e Atendedor Automático de Chamadas usando uma única linha telefónica.

O FAXMATE reconhece automaticamente a natureza do sinal telefónico e remete-o para o Fax/Modem, Telefone ou Atendedor de Chamadas.

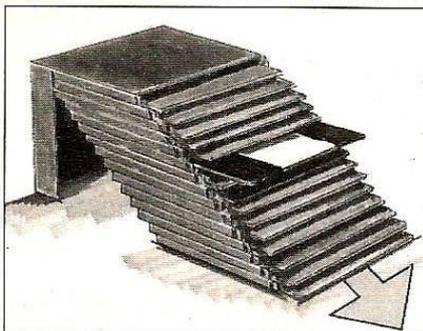
Para instalar basta ligar, na parte posterior do aparelho, as diversas fichas: entrada de linha telefónica, telefone, modem/fax e Atendedor de chamadas.

Ref.^a. FX01

Faxmate 3000

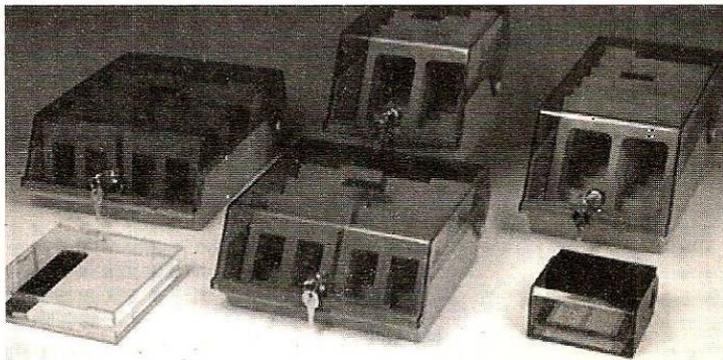
24 863\$00

CAIXA «ELÁSTICA» PARA 15 DISQ. 3,5 "



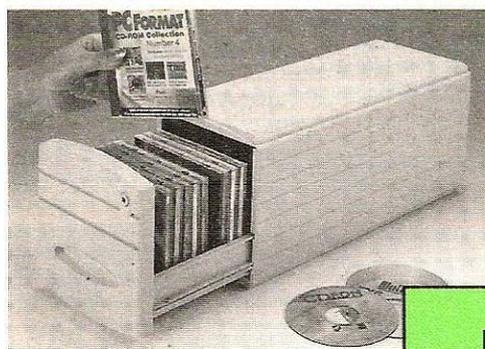
Ref.^a. CE01
Preço: 499\$00

CAIXAS ARQUIVADORAS PARA DISQUETAS



10 disq. 3,5"	Refª. H004	147\$00
50 disq. 5,25" c/ chave	Refª. H006	870\$00
100 disq. 5,25" c/ chave	Refª. H007	1 038\$00
50 disq. 3,5" c/ chave	Refª. H010	570\$00
100 disq. 3,5" c/ chave	Refª. H012	730\$00

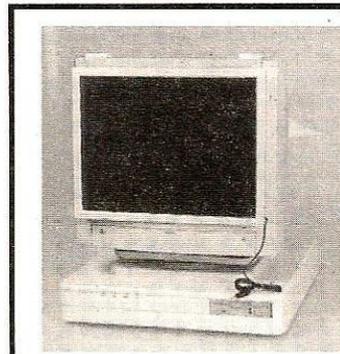
DISK BOX CD



- Para arquivar os seus CD-ROM ou CD Audio, ou quaisquer outros tipos de CDs.
- Capacidade 30 CDs.
- Concepção atractiva, funcional e robusta.

Refª. BOX01
Preço: 2 635\$00

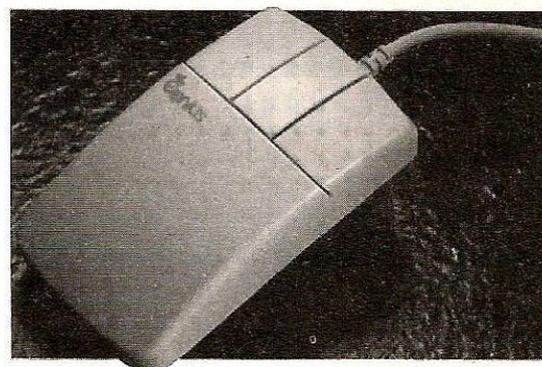
FILTRO PARA MONITOR



Filtro de vidro anti-radiações e anti-estático.
Para monitor 12"/14".

Refª. VIDR2
Preço: 1 450\$00

EASY MOUSE



Bom e Barato!

- 3 botões
- 300 a 1800 dpi

Refª. EAM01

2 487\$00

Tornado JT 251



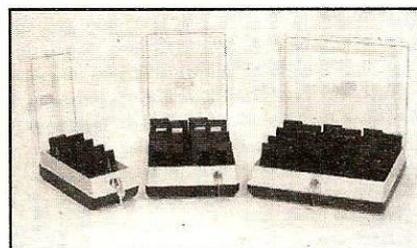
Joystick analógico

- Desenho ergonómico
- Switches sensíveis ao tacto
- Ajustamento fácil dos eixos x e y.
- Autodisparo
- Botões de disparo tipo gatilho e para o polegar
- Centralização automática
- Cabo com 1,8 metros com ficha de 15 pinos.

Refª. JT251

3 291\$00

CAIXAS DE ARQUIVO ESPECIAIS

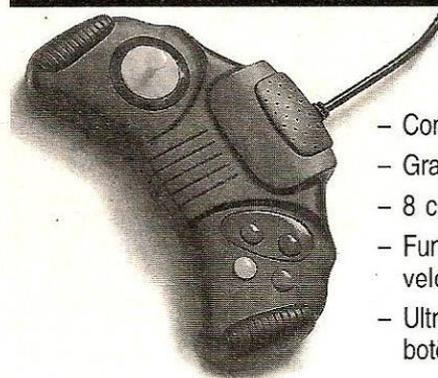


NOVO

Caixas de arquivo especiais com chave.

Caixa 50 Disquetes	Refª. ESP01	876\$00
Caixa 100 Disquetes	Refª. ESP02	1 316\$00
Caixa 150 Disquetes	Refª. ESP03	1 682\$00

GAME PAD J-05



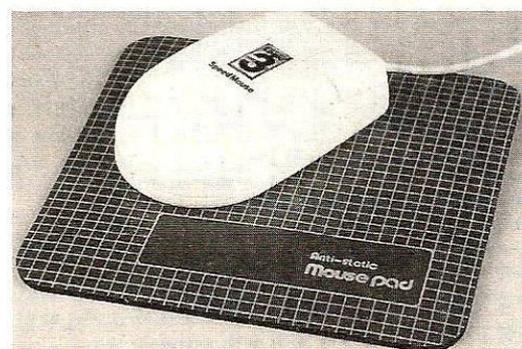
- Concepção ergonómica
- Grande precisão
- 8 controlos direccionais
- Função turbo de alta velocidade
- Ultra-sensíveis, mas robustos botões de disparo.

Refª. GPJ05

3 218\$00

SPEED MOUSE LG 3PC

NOVO



Cá está um «Speed Gonzalez» da família dos camundongos informáticos.

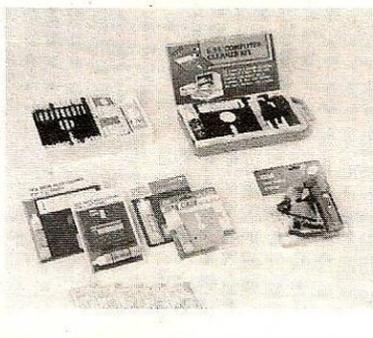
- 3 botões supersensíveis: 800 dpi
- Deslocação suave e rigorosa
- Software de instalação, disq. 3,5"
- Inclui: Rato, Suporte para o rato e tapete.

Refª. SPM01

3 290\$00

PRODUTOS DE LIMPEZA

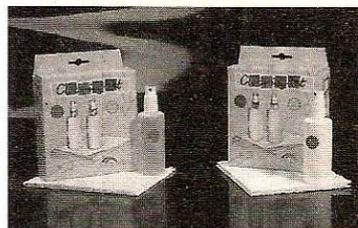
NOVO



Kit Limpeza c/ Aspirador*	Refª. KIT01	2 780\$00
Kit Limpeza c/ Spray	Refª. KIT02	2 635\$00
Mini Aspirador*	Refª. KIT03	735\$00

* Pilhas não incluídas.

PRODUTOS DE LIMPEZA SUNNYLINE



CLEANING-KIT PLÁSTICO

Frasco de 125 ml líquido anti-estática + 15 Panos.

Refª. SUN01	1 126\$00
-------------	-----------

CLEANING-KIT VIDRO

Frasco de 125 ml líquido anti-estática + 15 Panos.

Refª. SUN02	1 126\$00
-------------	-----------

CLEANING-KIT PLÁSTICO E VIDRO

2 Frascos de 125 ml + 15 Panos.

Refª. SUN03	1 455\$00
-------------	-----------

AIR SPRAY

Refª. SUN04	1 390\$00
-------------	-----------



CLEANING-BOX PLÁSTICO

Embalagem recarregável com 100 Panos de limpeza.

Ref. SUN05	915\$00
------------	---------

CLEANING-BOX VIDRO

Embalagem recarregável com 100 Panos de limpeza.

Ref. SUN06	915\$00
------------	---------

CLEANING-BOX PLÁSTICO RECARGA

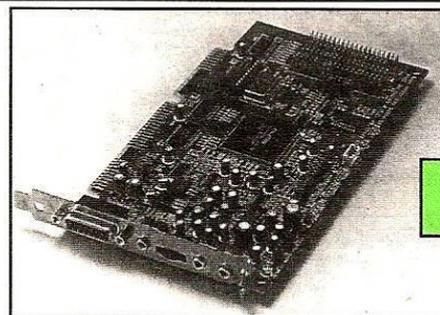
Ref. SUN07	658\$00
------------	---------

CLEANING-BOX VIDRO RECARGA

Ref. SUN08	658\$00
------------	---------

PLACA DE SOM ACERMAGIC S30

NOVO



Refª. ACER2
Preço: 16 015\$00

- Compatível: Sound Blaster, Sound Blaster Pro, AdLib, Sound Blaster 16 Basic, Microsoft Windows Sound System, Microsoft Windows 3.1, MPC nível 2.
- Sintetizador FM: Yamaha OPL3 FM. 20 vozes, 4 operadores.
- Wave Audio: Gravação e reprodução até 44,1 KHz, 16-bit DAC e ADC, 16-bit e 8-bit em estéreo e mono, gravação e reprodução simultâneos.
- MIDI Audio: Interface MIDI para device MIDI externo e Placa Wave interna, suporta Roland MPU401 e MIDI standard.
- Interface CD-ROM para Sony, Panasonic e IDE.
- Saída para altifalante, entrada/Saída de linha, entrada para microfone e porta para joystick.
- Suporte para carta de síntese Wave, não incluída.
- Fácil instalação
- Software «Music Center» para transformar o seu PC num estúdio de som.

LIVROS

EDITORA MCGRAW-HILL

OBRAS EM PORTUGUÊS

Obras	Autor	Refª	Preço
WINDOWS 95 - NÃO AUTORIZADO	Shulmann	Refª MCG01	8 400\$00
ACCESS 2 F/WINDOWS	Ramalho	Refª MCG02	6 720\$00
GUIA INCRÍVEL DO ACCESS 2.0	McFedries	Refª MCG03	5 250\$00
ORACLE	Ceriola	Refª MCG04	6 825\$00
3D STUDIO-MATERIAL E ILUMINAÇÃO	Bousquet	Refª MCG05	3 990\$00
3D STUDIO-DICAS TÉCNICAS	Bousquet	Refª MCG06	4 200\$00
3D STUDIO-VIDRO E REFLEXÃO	Bousquet	Refª MCG07	3 990\$00
3D STUDIO-ANIMAIS E PESSOAS	Bousquet	Refª MCG08	4 200\$00
3D STUDIO-CRIAÇÃO DE FORMAS E EXTRUSÃO	Bousquet	Refª MCG09	3 990\$00
3D STUDIO-TRANSMUTAÇÃO E ANIMAÇÃO	Bousquet	Refª MCG10	3 990\$00
SIMCITY 2000	Tauber	Refª MCG11	5 040\$00
LIVRO OFICIAL DO SOUND BLASTER	Ridge	Refª MCG12	8 400\$00
QUALIDADE EM SOFTWARE	Antonioni	Refª MCG13	3 150\$00
INFORMATION HIGHWAY	Sullivan	Refª MCG14	5 985\$00
PROGRAMANDO EM MS VISUAL BASIC	Young	Refª MCG15	9 240\$00
MS PROJECT 4 F/WIN PASSO/PASSO	Microsoft	Refª MCG16	8 190\$00
MULTIMÍDIA-PROGRAMAÇÃO F/WINDOWS	Rimmer	Refª MCG17	10 290\$00
MS POWERPOINT 4 F/ WINDOWS	Sagman	Refª MCG18	9 345\$00
DECLÍNIO E QUEDA DOS ANALISTAS E PROGRAMADORES	Yordon	Refª MCG19	7 350\$00
FALCON 3	Hart-Davies	Refª MCG20	7 350\$00
GUIA PRÁTICO WINDOWS 3.1, WORD 6.0, EXCEL 5.0, ACCESS 2.0	Computerland	Refª MCG21	5 985\$00
GUIA PRÁTICO WINDOWS 3.1	Computerland	Refª MCG22	2 415\$00
GUIA PRÁTICO WORD 6.0	Computerland	Refª MCG23	2 415\$00
GUIA PRÁTICO EXCEL 5.0	Computerland	Refª MCG24	2 415\$00
GUIA PRÁTICO ACCESS 2.0	Computerland	Refª MCG25	2 415\$00

OBRAS EM INGLÊS

Obras	Autor	Refª	Preço
SOFTWARE EVALUATION FOR CERTIFICATION	Rac	Refª MCG26	18 900\$00
THE ESSENTIAL INTERNET STARTER KIT	McGraw-Hill	Refª MCG27	18 018\$00
ENGINEERING DATA MANAGEMENT	Mcintosh	Refª MCG28	14 154\$00
MULTIMEDIA LITERACY	Hofstetter	Refª MCG29	14 490\$00
OBJECT ORIENTED/CLIENT SERVER APPLIC DEV.	Ayer	Refª MCG30	12 600\$00
THE INTERNET FOR EVERYONE	Wiggins	Refª MCG31	9 434\$00
REPAIRING PCS BEYOND BASICS2ED	Hordeski	Refª MCG32	5 040\$00
TROUBLESHOOT & REPAIR PCS	Margolis	Refª MCG33	5 250\$00
EASY PC MAINTENANCE AND REPAIR 2ED	Laplant	Refª MCG34	3 990\$00
OSBORNE WINDOWS PROGRAMM SERIES I	Schildt	Refª MCG35	7 560\$00
ADVANCED MULTIMEDIA PROGRAM	Rimmer	Refª MCG36	8 610\$00
THE INTERNET INSIDER	Prevost	Refª MCG37	2 310\$00
PLANET INTERNET	Rimmer	Refª MCG38	5 460\$00
INTERNET 101: COLLEGE STD'S GDE	Glossbrenner	Refª MCG39	4 200\$00
EASY AUTOCAD LT F/WINDOWS	Hood	Refª MCG40	5 040\$00
STEP INTO VIRTUAL REALITY	Iovine	Refª MCG41	7 140\$00
CA-CLIPPER DEVELOPER'S REALITY	Ochiogrosso	Refª MCG42	9 660\$00
THAT'S EDUTAINMENT	Brown	Refª MCG43	4 410\$00

LIVROS EDITORIAL O LIVRO (Em português)

INTRODUÇÃO À INFORMÁTICA E COMPUTADORES	Refª L001	2 200\$00
TÉCNICAS DE PROGRAMAÇÃO	Refª L002	2 188\$00
INTRODUÇÃO AO BASIC	Refª L003	3 500\$00

LIVROS

edições **CETOP**

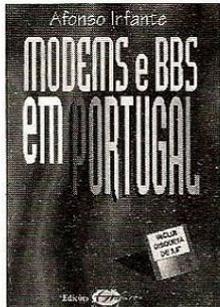
INVISUAIS: Quando solicitado forneceremos em disquete suplementar os textos integrais destes livros



DBase III Plus PROGRAMAÇÃO AVANÇADA

Renato Prista Casquilho

O dBase III Plus é uma poderosa base de dados e a sua linguagem de programação, quando devidamente estruturada e implementada, permite a criação de aplicações eficientes, podendo resolver praticamente todas as necessidades do utilizador individual e de muitas pequenas e médias empresas. Os programas-exemplo - listados ao longo do livro e fazendo parte da disquete - são discutidos e analisados com grande pormenor. Alguns programas têm utilização prática imediata (BANCOS, TITULOS, etc.).



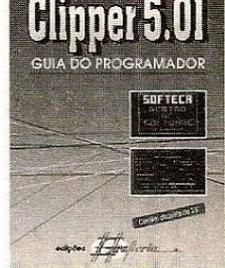
MODEMS E BBS EM PORTUGAL

Afonso Infante

O objetivo deste livro é levar ao conhecimento de todos o extraordinário mundo das BBS, com especial relevo para a situação portuguesa. BBS? Modem? Programas de Comunicação? Estes são os temas que o autor desenvolve ao longo das 128 páginas do livro. O livro é acompanhado por uma disquete de 3,5" - 720Kb com o Programa de Comunicações TELEDATE 4.00 (shareware).

Ref.^o GR04

P. V. P. 2 415\$00
P. V. A. 1 932\$00



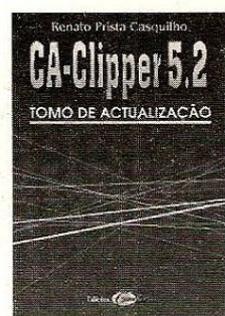
CLIPPER 5.01 Guia do Programador

Renato Prista Casquilho

Este livro destina-se, essencialmente, aos programadores de dBASE que pretendem dar o «salto» qualitativo para o poderoso ambiente facultado pelo Clipper 5.01, capaz de gerar aplicações sofisticadas, elegantes, rápidas e autónomas. Nele são analisados e exemplificados todos os comandos, funções e declarações, incluindo as novidades e correções introduzidas pela versão 5.01. Abra é acompanhado por uma disquete de 3 1/2", cujo conteúdo constitui uma demonstração das inúmeras capacidades desta linguagem.

Ref.^o GR03

P. V. P. 3 600\$00
P. V. A. 2 880\$00



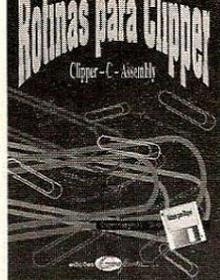
CA-CLIPPER 5.2 Tomo de Atualização

Renato Prista Casquilho

Esta edição constitui um complemento ao «Clipper 5.01 - Guia do Programador». O lançamento da versão 5.2 do Clipper pela Computer Associates veio criar uma imperiosa necessidade de atualização motivada pelas novas capacidades introduzidas, das quais se destaca a arquitetura RDD (Repeatable Database Driver) destinada a permitir o acesso a bases de dados de outros formatos, nomeadamente do dBASE IV, do FoxPro e do Paradox.

Ref.^o GR05

P. V. P. 1 470\$00
P. V. A. 1 176\$00



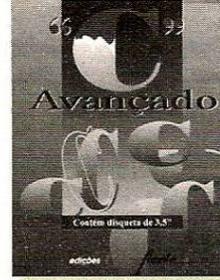
ROTINAS PARA CLIPPER Clipper - C - Assembly

Nelson Pires

Para além de um conjunto de rotinas escritas em Clipper que proporcionam um melhor aproveitamento das capacidades da linguagem, o autor explica com todo o pormenor como criar a sua própria biblioteca de funções, alterando-a ou alargando-a à medida das necessidades. O livro facultar-lhe 10 funções para o ecrã, 15 funções para data e hora e 29 funções ou procedimentos para controlo de menus, matrizes, caixas, estilos, ecrãs de ajuda, etc. O livro é completado por uma disquete contendo todas as rotinas em Clipper, C e Assembly.

Ref.^o GR06

P. V. P. 3 570\$00
P. V. A. 2 856\$00



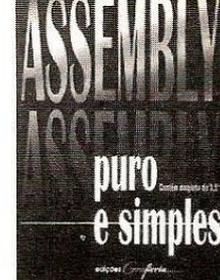
«C» AVANÇADO

Ruben Romano Borges Barboza

O mundo alucante da «C» - a linguagem com que foram escritos alguns dos mais célebres programas e compiladores - é tratado nesta obra de forma particularmente atraente e sugestiva. Os temas são desenvolvidos com a necessária profundidade ao longo dos doze capítulos do livro. Nele poderá o leitor encontrar um módulo repleto de funções para tratamento de ecrã: estruturas de dados, tabelas de Hashing, listas ligadas e árvores binárias de decisão e balanceadas completam o terceiro capítulo. O sempre actual e algo misterioso tema do TSR é totalmente desmistiado no quarto capítulo; o quinto versa uma não menos empolgante matéria: compressão de dados. O tratamento de impressoras e seu controlo é objecto de grande desenvolvimento no sexto capítulo. O sétimo aborda uma matéria de grande actualidade: programação de placas de som. A comunicação assíncrona entre computadores via portas série é desvendada no oitavo capítulo. No nono, de forma mais relaxante, a memória estendida é expandida, as interrupções mais conhecidas e os device-drivers completam o décimo capítulo. O perfeitico «ratio» não foi esquecido, dele se ocupando o décimo primeiro capítulo. O último capítulo é totalmente dedicado à programação gráfica para os modos 640x480, 320x200 e 320x240. Todos os capítulos incluem pelo menos um programa de demonstração, com a respectiva listagem e as explicações apropriadas. Finalmente, todo o código apresentado ao longo do livro, bem como os executáveis, são facultados numa disquete, cujo conteúdo é sistematizado no primeiro capítulo.

Ref.^o GR08

P. V. P. 3 276\$00
P. V. A. 2 621\$00



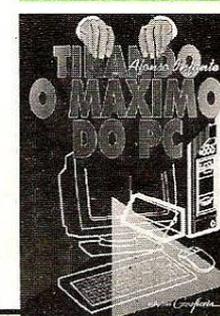
ASSEMBLY - Puro e Simples

Vasco Correia Veloso

Por muitos encarada como uma linguagem «hermética» e «elitista», o Assembly revela-se ao longo das 215 páginas do livro na sua maior simplicidade e pureza. O autor desmistifica todas estas crenças injustificadas, iniciando o leitor pelos conceitos elementares de memória e seus endereços e pelas diferentes bases numéricas, nomeadamente a hexadecimária, utilizada pelas mnemónicas do Assembly. Do livro faz parte integrante uma disquete contendo todo o código apresentado nos diferentes capítulos, o qual é compatível com o assembler MASM (Macro Assembler da Microsoft) e com todos os restantes dílios compatíveis, incluindo o AS6 de Eric Isaacson. Este último é distribuído na disquete em regime de Shareware, pelo que obedece às normas internacionais para este tipo de software.

Ref.^o GR09

P. V. P. 3 159\$00
P. V. A. 2 527\$00



NOVO TIRANDO O MÁXIMO DO PC

Afonso Infante

Uma das grandes características que distingue os compatíveis IBM PC dos outros computadores é a quantidade e variedade de hardware e software disponível no mercado. Os computadores são máquinas em constante evolução, saindo novidades quase todos os dias que permitem «tirar um pouco mais» de cada máquina. O problema é a confusão gerada precisamente por este fenómeno. A certa altura começa a tornar-se impossível conhecer tudo o que se encontra disponível para «Tirar o Máximo do seu PC». É aqui a razão de ser deste livro: tenta servir de guia ou referência para tudo o que existiu ou existe em termos de hardware e software para PC, ajudando o leitor a decidir qual deverá ser o próximo passo para conseguir aquele «algo mais» que ambiciona do seu computador. Os seus dezasseis capítulos distribuídos por 184 páginas amplamente ilustradas, cobrem uma completa descrição de todos os processadores conhecidos à data da publicação, as placas-mãe, os discos rígidos e seus controladores, drives de disquetes, sistemas de backup em fita magnética, drives de CD-ROM, impressoras, placas gráficas, monitores, teclados, ratos e joysticks, placas de som, conversão vídeo/VGA a vice-versa, modems e Fax/Modems e Scanners. No respeitante ao Software, são desenvolvidos os ADD para MS-DOS, os sistemas operativos e o software de utilização geral: processadores de texto, folhas de cálculo, bases de dados, compressores, desenho vectorial e BitMap, compiladores e anti-virus.

COMPÊNDIO DE INFORMÁTICA

José A. de Matos

Um livro que se pretende acessível, mas não simplista; didático, mas não académico. Dirige-se ao utilizador comum de computadores, facultando uma perspectiva global da sua constituição e evolução, mas também aos mais experientes, revelando não segredos, mas informações indispensáveis para o dia-a-dia de trabalho na empresa ou no lar. Trata-se de um verdadeiro «Compêndio» e, como tal, de um guia para os estudantes de cursos de informática.

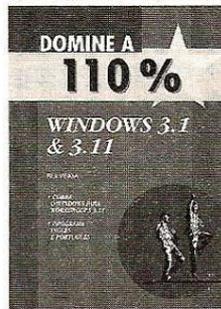
Ref.^o GR07

P. V. P. 2 415\$00
P. V. A. 1 932\$00

Ref.^o GR10

P. V. P. 2 940\$00
P. V. A. 2 352\$00

FCA - EDITORA DE INFORMÁTICA



- AUTOCAD 12 - CURSO COMPLETO COMPUTADORES PARA TODOS NÓS
- DOMINE A 110% EXCEL 5 PARA WINDOWS
- DOMINE A 110% MICROSOFT ACCESS 2
- DOMINE A 110% WINDOWS 3.1/3.11
- EXCEL FOR WINDOWS - VERSÃO 4 - DEPRESSA E BEM
- FUNDAMENTAL DO AMI PRO 3
- FUNDAMENTAL DO CLIPPER 5.2
- FUNDAMENTAL DO CORELDRAW 3.0
- FUNDAMENTAL DO DBASE 5 FOR WINDOWS
- FUNDAMENTAL DO DOS 5
- FUNDAMENTAL DO EXCEL 5 FOR WINDOWS (3ª edição)
- FUNDAMENTAL DO EXCEL 5 FOR WINDOWS
- FUNDAMENTAL DO FOX PRO 2.5 FOR WINDOWS/DOS
- FUNDAMENTAL DO HARVARD GRAPHICS VERSÃO 1.0/1.01
- FUNDAMENTAL DO LOTUS 1 2 3 VERSÃO 4 FOR WINDOWS
- FUNDAMENTAL DO MICROSOFT ACCESS 2
- FUNDAMENTAL DO MICROSOFT PROJECT 4/WINDOWS
- FUNDAMENTAL DO MS DOS 6 (4ª edição)
- FUNDAMENTAL DO MS DOS 6/6.2 (5ª edição)
- FUNDAMENTAL DO MS DOS 6/6.2/6.22
- FUNDAMENTAL DO POWER POINT 4 WINDOWS
- FUNDAMENTAL DO QUATTRO PRO 4.0/5.0 (3ª edição)
- FUNDAMENTAL DO TURBO PASCAL 6.0 E 7.0 (2ª edição)
- FUNDAMENTAL DO WINDOWS 3.0 E 3.1 (5ª edição)
- FUNDAMENTAL DO WORD 6 PARA WINDOWS
- GUIA PRÁTICO DO WINDOWS 3.1
- GUIA PRÁTICO MICROSOFT ACCESS 2
- MICROSOFT OFFICE PARA TODOS NOS
- SEGREDOS DO SOM NO PC
- VISUAL BASIC FOR WINDOWS 3.0 - CURSO COMPLETO
- WINDOWS 3.1 - DEPRESSA E BEM
- WINDOWS 3.1 - GUIA DE APRENDIZAGEM VISUAL
- WORD 6 FOR WINDOWS - DEPRESSA E BEM
- WORD FOR WINDOWS VERSÃO 2.0 - CURSO COMPLETO (2ª edição)
- WORD FOR WINDOWS VERSÃO 2.0 - DEPRESSA E BEM

- Ref.^o FC01 4 450\$00
- Ref.^o FC02 2 550\$00
- Ref.^o FC03 4 750\$00
- Ref.^o FC04 4 550\$00
- Ref.^o FC05 4 350\$00
- Ref.^o FC06 3 240\$00
- Ref.^o FC07 2 350\$00
- Ref.^o FC08 2 960\$00
- Ref.^o FC09 1 750\$00
- Ref.^o FC10 2 960\$00
- Ref.^o FC11 1 750\$00
- Ref.^o FC12 2 750\$00
- Ref.^o FC13 2 350\$00
- Ref.^o FC14 1 850\$00
- Ref.^o FC15 1 750\$00
- Ref.^o FC16 1 950\$00
- Ref.^o FC17 2 550\$00
- Ref.^o FC18 3 300\$00
- Ref.^o FC19 1 840\$00
- Ref.^o FC20 1 950\$00
- Ref.^o FC21 2 450\$00
- Ref.^o FC22 2 550\$00
- Ref.^o FC23 1 850\$00
- Ref.^o FC24 1 950\$00
- Ref.^o FC25 2 100\$00
- Ref.^o FC26 2 750\$00
- Ref.^o FC27 2 970\$00
- Ref.^o FC28 3 950\$00
- Ref.^o FC29 4 850\$00
- Ref.^o FC30 4 250\$00
- Ref.^o FC31 4 250\$00
- Ref.^o FC32 4 950\$00
- Ref.^o FC33 2 400\$00
- Ref.^o FC34 3 450\$00
- Ref.^o FC35 2 850\$00
- Ref.^o FC36 2 980\$00

EDIÇÕES CETOP

- COMO CRIAR TABELAS COM WORD 2
- COMO CRIAR MAILINGS COM WORD 2
- COMO CRIAR GRÁFICOS COM EXCEL 4
- APRESENTAÇÃO DE DADOS COM EXCEL PARA WINDOWS
- NETWORK PARA WINDOWS - LIVRO I
- NETWORK PARA WINDOWS - LIVRO II
- NETWORK PARA WINDOWS - LIVRO III
- NETWORK PARA WINDOWS - LIVRO IV
- INICIAÇÃO AOS COMPUTADORES
- MS DOS 6.0 E 6.2 - GUIA DE TREINO
- WINDOWS 3.1 - GUIA DE TREINO
- EXCEL 4.0 E 5.0 PARA WINDOWS - GUIA DE TREINO
- INICIAÇÃO À MULTIMÉDIA
- QUARKXPRESS PARA WINDOWS - GUIA DE TREINO
- FOX PRO 2.0 - GUIA DE TREINO
- WINDOWS NT
- EXCEL 5 PARA WINDOWS - GUIA FORMAÇÃO RÁPIDA
- WORD 2 E 6 PARA WINDOWS - GUIA FORMAÇÃO RÁPIDA
- REVELANDO O WINDOWS 95
- WORD 6 PARA WINDOWS
- FOLHAS DE CÁLCULO
- INTRODUÇÃO À PROGRAMAÇÃO EM PASCAL
- INTRODUÇÃO À PROGRAMAÇÃO EM COBOL
- MANUAL CONCISO APLICAÇÕES DE SOFTWARE
- PROGRAMAÇÃO CRIATIVA
- O ESSENCIAL WINDOWS 3.1 - VERSÃO INGLÊS
- ESSENCIAL DO DOS 6
- ESSENCIAL DO DBASE III PLUS
- ESSENCIAL WORDPERFECT - VERSÃO 5.1
- DICIONÁRIO MICROINFORMÁTICA
- TECNOLOGIAS DA INFORMAÇÃO

- Ref.^o CET01 1 880\$00
- Ref.^o CET02 1 880\$00
- Ref.^o CET03 1 880\$00
- Ref.^o CET04 1 880\$00
- Ref.^o CET05 1 880\$00
- Ref.^o CET06 1 880\$00
- Ref.^o CET07 1 880\$00
- Ref.^o CET08 1 880\$00
- Ref.^o CET09 1 985\$00
- Ref.^o CET10 1 985\$00
- Ref.^o CET11 1 985\$00
- Ref.^o CET12 1 985\$00
- Ref.^o CET13 1 985\$00
- Ref.^o CET14 1 985\$00
- Ref.^o CET15 1 985\$00
- Ref.^o CET16 4 463\$00
- Ref.^o CET17 1 985\$00
- Ref.^o CET18 2 090\$00
- Ref.^o CET19 2 290\$00
- Ref.^o CET20 1 890\$00
- Ref.^o CET21 2 678\$00
- Ref.^o CET22 3 833\$00
- Ref.^o CET23 3 770\$00
- Ref.^o CET24 2 415\$00
- Ref.^o CET25 4 190\$00
- Ref.^o CET26 2 678\$00
- Ref.^o CET27 2 478\$00
- Ref.^o CET28 2 478\$00
- Ref.^o CET29 1 900\$00
- Ref.^o CET30 2 478\$00
- Ref.^o CET31 2 394\$00

P. V. P. - Preço de venda ao público; P. V. A. - Preço especial para assinante.

BOOT125
BOOT.SYS V1.25 - Correrários
CONFIG.SYS
CCP10B
Proteção do computador da
MkAte.

COPY122
Um copy mais poderoso que o
MS-DOS.
PM05
Múltiplas paths seleccionadas
por menu.

Automóveis.
SH197
Natureza.
SH198
Alcomotiva - "Chalenger 3995".
Michele Pfeifer.
Chete tribal da Nova Guiné.

Manipulação da impressora em
Assembly.
REBOOT16
Um reboot diferente.
SVGABG31
Driver para 256 cores em C e
Pascal.

FIRE
Simulação de fogo de artifício
FISH
Simulação de um aquário
SH074 - BG06
Jogo do Gamão para Windows.

MICROWIN
Jogo de arcaça para Windows
3.1.
WDBBRK2
Jogo de códigos com SRC em C.

Converte GIF para ICON.
METZ30
METZ - Conjunto de utilitários
para Windows.

Programa de desenho.
W-ENVEL
Inversão de envelopes.
WNTIME
A hora em qualquer ponto do
globo.

MULTIMEDIA

SH171 - PIANOMAN
Editor de música com exemplos.
SH222 - MUSICAMI
Coleção de ficheiros MIDI.

VISUAL BASIC

SH243 - BFVBDDL
Biblioteca Add-On.
GBLB1A
Rolinas e bibliotecas.

PASCAL

SH048 - TECHNOS
Turbo Toolkit V5.0 Fontes e manual.
SH186 - ALIAS
Definição de macros no DOS.

PROGRAMAÇÃO - GERAL

SH064 - DOSMAP2
Apresenta o mapa da memória.
EMSTOOL2
Toolkit da Intel para programadores.

WINDOWS PROGRAMAS

SH020 - AIRFORCE
Jogo de arcaça.
ALM104
Almanaque para Windows 3.0.



Montra Spooler

Recorte ou fotocopie o cupão e remeta-o juntamente com o cheque no envelope RSF anexo.

CUPÃO DE ENCOMENDA
- MONTRA SPOOLER

Spooler 37

Cliente nº: _____

A PREENCHER PELOS NOSSOS SERVIÇOS

Desejo receber os seguintes produtos:

Quant.	Ref.	Designação	Preço

ENVIO O CHEQUE Nº. DATA

--	--	--	--	--

S/ BANCO

SUB-TOTAL	
PORTES	+ 450,00
TOTAL	

NOME
MORADA
C. POSTAL LOCALIDADE
TELEFONE NÃO SOU ASSINANTE SOU ASSINANTE Nº

ESTIMADO LEITOR:

Leia com atenção o quadro de abertura da «Montra Spooler».

A partir deste número, esta sofre algumas alterações.

Graficaria

PUBLICIDADE E ARTES GRÁFICAS, LDA.

- PROJECTOS GRÁFICOS
- COMPOSIÇÃO
- ARTE-FINALIZAÇÃO
- SELECÇÃO DE CORES
- TRATAMENTO DIGITAL DE IMAGEM
- EDIÇÃO ELECTRÓNICA
- SERVICE BUREAU
- MONTAGEM
- IMPRESSÃO

Qualidade
Eficiência
Rapidez

REEDIÇÃO DOS NÚMEROS ESGOTADOS (1 a 12)

Use o envelope RSF anexo para nos remeter este cupão (ou fotocópia) e respectivo pagamento.

CUPÃO DE RESERVA DAS EDIÇÕES ESPECIAIS SPOOLER 1-2-3, 4-5-6, 7-8-9 e 10-11-12

NOME

ENDEREÇO

C. POSTAL LOCALIDADE TELEF.

DESEJO RECEBER A EDIÇÃO ESPECIAL

«SPOOLER 1 - 2 - 3»

«SPOOLER 4 - 5 - 6»

«SPOOLER 7 - 8 - 9»

«SPOOLER 10 - 11 - 12»

Cada EDIÇÃO ESPECIAL: 1 500\$00 + 450\$00 (Portes)

ENVIO O CHEQUE Nº. S/O BANCO

DATA

--	--	--	--	--	--

Spooler 37 Assinatura

ASSINATURAS



Use o envelope RSF anexo para nos remeter este cupão (ou fotocópia) e respectivo pagamento.

CUPÃO DE ASSINATURA

Spooler 37 **Assinante nº:** * *
A PREENCHER PELOS NOSSOS SERVIÇOS

NOME

DATA DE NASCIMENTO* ____ / ____ / ____

PROFISSÃO*

TRABALHA COM COMPUTADORES?* SIM NÃO

SE SIM ESPECIFIQUE A MARCA E MODELO*

ENDEREÇO

C. POSTAL LOCALIDADE TELEF(S)

**DESEJO ASSINAR A SPOOLER A PARTIR DO N.º 38 (inclusive)
E DURANTE 12 NÚMEROS – 8.000\$00.**

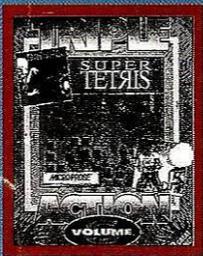
PARA PAGAMENTO DA ASSINATURA ENVIO A IMPORTÂNCIA DE 8.000\$00:

EM CHEQUE Nº. S/O BANCO

POR VALE DE CORREIO Nº.

* A resposta aos itens assinalados com um (*) é opcional.

Jogos



CIGEST

Edições em CD-ROM

Colecção U.S. & Russian Military / American Space Exploration / Natural Parks

U.S. AIR POWER	2.765\$00	U.S. AIRCRAFT WEAPONS	2.765\$00
OPERATION DESERT STORM	2.765\$00	U.S. NAVAL WEAPONS	2.765\$00
U.S. ATTACK PLANES & BOMBERS	2.765\$00	U.S. AIR DEFENSE WEAPONS	2.765\$00
U.S. BATTLE TANKS	2.765\$00	U.S. INFANTRY & ARTILLERY WEAPONS	2.765\$00
U.S. AIRCRAFT CARRIERS	2.765\$00	U.S. NAVY SHIPS	2.765\$00
U.S. MILITARY HELICOPTERS	2.765\$00	U.S. NAVY & MARINE AIRCRAFT	2.765\$00
U.S. SUBMARINE	2.765\$00	U.S. MARINE CORPS	2.765\$00
U.S. BATTLESHIPS AND CRUISERS	2.765\$00	AMERICAN SPACE EXPLORATION	2.910\$00
U.S. DESTROYERS AND FRIGATES	2.765\$00	EXPLORING THE SOLAR SYSTEM	2.910\$00
U.S. AMPHIBIOUS ASSAULT	2.765\$00	RACE TO THE MOON	2.910\$00
U.S. FIGHTER JETS	2.765\$00	THE EDGE OF FOREVER	2.910\$00
U.S. MISSILES	2.765\$00	THE VOYAGER LEGACY	2.910\$00
RUSSIAN POWER	2.765\$00	U.S. SHUTTLE FLEET	2.910\$00
RUSSIAN WARSHIPS	2.765\$00	YOSEMITE NATIONAL PARK	2.910\$00
RUSSIAN GROUND FORCES	2.765\$00	YELLOWSTONE NATIONAL PARK	2.910\$00
U.S. SPY PLANES	2.765\$00	GRAND CANYON NATIONAL PARK	2.910\$00
U.S. SEAPOWER	2.765\$00		

Colecção 5 PLUS 1

BOMBUZAL + 5 JOGOS	1.645\$00	MEAN 18 + 5 JOGOS	1.645\$00
FOOTBALL MANAGER I + 5 JOGOS	1.645\$00	ROCKET RANGER + 5 JOGOS	1.645\$00
THE THREE STOOGES + 5 JOGOS	1.645\$00	3D WORLD BOXING + 5 JOGOS	1.645\$00
IT GAME FROM THE DESERT + 5 JOGOS	1.645\$00	TV SPORTS FOOTBALL + 5 JOGOS	1.645\$00
HARD BALL + 5 JOGOS	1.645\$00	3D WORLD TENNIS + 5 JOGOS	1.645\$00
SINBAD + 5 JOGOS	1.645\$00	BLADE WARRIOR + 5 JOGOS	1.645\$00
TV SPORTS BASKETBALL + 5 JOGOS	1.645\$00	BUNNY BRICKS + 5 JOGOS	1.645\$00
MILLENGUA: THE 1000-MILE CLASSIC + 5 JOGOS	1.645\$00	DEFENDER OF THE CROWN + 5 JOGOS	1.645\$00
SUPREMACY + 5 JOGOS	1.645\$00	FOOTBALL MANAGER III + 5 JOGOS	1.645\$00
THE BLUES BROTHERS + JOGOS	1.645\$00	GRAND PRIX MASTER + 5 JOGOS	1.645\$00
CRAZY CARS III + 5 JOGOS	1.645\$00	THE KING OF THE CHICAGO + 5 JOGOS	1.645\$00
GHOSTBUSTERS II + 5 JOGOS	1.645\$00	LAST NINJA II + 5 JOGOS	1.645\$00
TITUS THE FOX + 5 JOGOS	1.645\$00	PREDATOR II + 5 JOGOS	1.645\$00
INTERNATIONAL ATHLETICS + 5 JOGOS	1.645\$00	SAVAGE + 5 JOGOS	1.645\$00
FT 4 TOM CAT + 5 JOGOS	1.645\$00	DELUXE STRIP POKER II + 5 JOGOS	1.645\$00

Colecção Triple Action

TRIPLE ACTION 1 - FANTASY WARRIOR BASHERS	2.201\$00
TRIPLE ACTION 2 - BATTLE OF MOUNTAIN GIGANT	2.201\$00
TRIPLE ACTION 3 - TITUS THE FOX VS GHOSTBUSTERS II	2.201\$00
TRIPLE ACTION 4 - THE BLUES BROTHERS SATAN MAMA	2.201\$00
TRIPLE ACTION 5 - CRAZY CARS GRAND PRIX MASTER	2.201\$00
BATTLE TECH	2.201\$00
TRIPLE ACTION 6 - SUPER TETRIS AIRBALL TIME BANDIT	2.201\$00

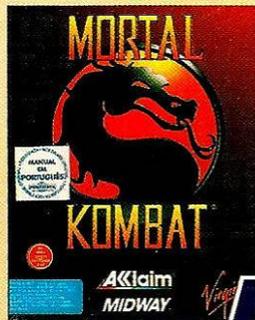


Colecção 50 Games Plus One

GUNSHIP + 50 JOGOS	2.765\$00
RAILROAD TYCOON + 50 JOGOS	2.765\$00
SUPER TETRIS + 50 JOGOS	2.765\$00
KNIGHTS OF THE SKY + 50 JOGOS	2.765\$00
SILENT SERVICE II + 50 JOGOS	2.765\$00

NOVOS

ANIMALS IN MOTION	1.645\$00	GROGERS ENCYCLOPEDIA 95	6.380\$00
WOMEN IN MOTION	1.645\$00	MICROSOFT ENCARTA	9.925\$00
HOUND OF THE BASKERVILLES	1.645\$00	MICROSOFT MONEY	5.316\$00
ISHAR	1.645\$00	MICROSOFT MUSIC INSTRUMENTS	6.380\$00
TRANSARCTICA	1.645\$00	MICROSOFT WORKS 3.0	8.965\$00
CLOCKWISER	2.201\$00	WORLD ATLAS 5.0	4.253\$00
KINGS QUEST COLLECTION (I-III-IV-V)	6.103\$00	MICROSOFT FAMILY PACK	24.100\$00
AWARD WINNERS GOLD EDITION		EMMANUELLE OF GREECE	2.700\$00
(SENSIBLE SOCCER ELITE PLUS/2000/JIMMY WHITESS)	4.990\$00	EMMANUELLE OF HOLLAND	2.700\$00
FLIGHT SIM TOOLKIT	2.765\$00	EMMANUELLE OF USA	2.700\$00
COLOUSSUS COLLECTION		EMMANUELLE OF PARIS	2.700\$00
(DANAS WIZARD/GAMMA BRIDGE)	2.765\$00	HOT PIE	2.657\$00
MARIO'S TIME MACHINE	4.431\$00	EROTIC MOVIES & ANIMATIONS	2.126\$00
MARIO TEACHES SUMS	4.431\$00	CHEEKY CHICS	2.126\$00
MARIO TEACHES WORDS	4.431\$00	CLIMAX	2.126\$00
MARIO'S MISSING	4.431\$00	ERUPTION	2.126\$00
TOP 50 GAMES FOR WINDOWS	2.062\$00	EXOTIC EXTASY	2.126\$00
TOP 100 GAMES VOL. 1 (MS-DOS)	2.062\$00	LIQUOR LIPS	2.126\$00
PACK + TEMPTATION (4 JOGOS)	13.408\$00	PUSSY GALORE	2.126\$00
MEGA PACK II (11 JOGOS)	11.001\$00	RED HOT	2.126\$00
WORLD OF FLIGHT SIMULATORS	2.062\$00	SMASHING	2.126\$00
WORLD OF COMMUNICATION	2.062\$00	TIGHT SQUEEZE	2.126\$00
WORLD OF FONTS & ICONS	2.062\$00	WIDE OPEN	2.126\$00
WORLD OF INTERNET	2.062\$00	ALL THE WAY UP	2.126\$00
WORLD OF OS/2	2.062\$00	NICE BUT NAUGHTY	2.126\$00
WORLD OF MPEG	2.062\$00	TONGUES TIED	2.126\$00
WORLD OF CD-ROM	2.062\$00	SWEET JUICES	2.126\$00



PORTIDATA

Edições em CD-ROM

Em Português

BENEATH A STEEL SKY	9.995\$00	WARCRAFT	11.995\$00
SANTA XMAS CAPER	1.995\$00	CREATURE SHOCK	13.995\$00
WORLD CUP USA 94	5.995\$00	CYBERIA	11.995\$00
DESERT STRIKE	5.995\$00	VOYEUR	9.995\$00
ZOO 2	5.995\$00	KINGS QUEST VII	11.995\$00
DARKSEED	8.995\$00	PREMIER MANAGER 3	7.995\$00
CYCLEMANIA	10.995\$00	JUNGLE STRIKE	5.995\$00
UNIVERSE	5.995\$00	DESCENT	11.995\$00
NOVASTORM	7.995\$00	MORTAL KOMBAT 2	11.995\$00
ECSTASIA	7.995\$00	DARK FORCES	12.995\$00
ALL NEW WORLD OF LEMMINGS 3	6.995\$00	DISCWORLD	8.995\$00
UNDER A KILLING MOON	14.995\$00	SUPERKARTS	11.995\$00
RISE OF THE ROBOTS	13.995\$00	LOST EDEN	11.995\$00
INFERNO	9.995\$00	IRON ASSAULT	11.995\$00
NASCAR	10.995\$00	DAEDALUS ENCOUNTER	11.995\$00
POWER DRIVE	6.995\$00	SUPSTREAM 5000	6.995\$00
RETRIBUTION	7.995\$00		

Em Inglês

AEGIS	14.995\$00	SANTA XMAS CAPERS	1.995\$00
ARNIE 2	1.495\$00	MICROGSM	15.995\$00
DARK LEGIONS	9.995\$00	PROTECHNICA	9.995\$00
POLICE QUEST IV	11.995\$00	TOTAL CARNAGE	8.995\$00
SIMON THE SORCERER	11.995\$00	FIGHTER WING	6.995\$00
FREDDY PHARGAS	9.995\$00	KINGDOM THE FAR REACHES	10.995\$00
TORNADO/FALCON 3.0	9.995\$00	LOUVRE ENCYCLOPEDIA	11.995\$00
ULTIMATE BODY BLOWS	7.995\$00	GINGERS READ	11.995\$00
LEMMINGS FOR WINDOWS	8.995\$00		

Edições em Disq. 3,5"

Em Português

AMERICAN GLADIATORS	6.995\$00	NASCAR	7.995\$00
BENEATH A STEEL SKY	7.995\$00	POWER DRIVE	6.995\$00
CANNON FODDER	7.995\$00	RETRIBUTION	7.995\$00
GOAL!	5.995\$00	WARCRAFT	11.995\$00
HAND OF FATE	7.995\$00	SPACE FEDERATION	9.995\$00
MORTAL KOMBAT	7.995\$00	CANNON FODDER 2	6.995\$00
TERMINATOR 2	7.995\$00	PROJECT X	4.995\$00
WORLD CUP USA 94	5.995\$00	BLACK HAWK	9.995\$00
DESERT STRIKE	5.995\$00	PREMIER MANAGER 3	6.995\$00
ZOO 2	5.995\$00	JUNGLE STRIKE	6.995\$00
UNIVERSE	6.995\$00	HOKUM	7.995\$00
ECSTASIA	9.995\$00	DESCENT	11.995\$00
ALL NEW WORLD OF LEMMINGS 3	6.995\$00	MORTAL KOMBAT 2	9.995\$00
RISE OF ROBOTS	12.995\$00	DISCWORLD	10.995\$00
ALADIN	8.995\$00	SUPERKARTS	-
LION KING	8.995\$00	SUPSTREAM 5000	6.995\$00

Em Inglês

ALIEN LEGACY	9.995\$00	HEXSIDER	
DELTA V	10.995\$00		
HEIMDALL II	9.995\$00		
HEX WIZARD	7.995\$00		
HIBRID GUNS	8.995\$00		
OUTPOST	9.995\$00		
RIDER CUP	7.995\$00		
SUPER HERO LEAGUE	10.995\$00		
THE CLUE	9.995\$00		
THE HORDE	9.995\$00		
ARMORED FIST	9.995\$00		
LODE RUNNER	8.995\$00		
ULTIMATE BODY BLOWS	7.995\$00		
LEMMINGS FOR WINDOWS	8.995\$00		
PYROTECHNICA	9.995\$00		
SUPER FROG	5.995\$00		
OPERATION BODY COUNT	8.995\$00		
FIGHTER WING	6.995\$00		

TNTS

Disq. 3,5"

CARNAGE	1.995\$00
E-MOTION	1.995\$00
INTERNATIONAL TENNIS	1.995\$00
STACK UP	1.995\$00
SOCCER MANAGER	1.995\$00
F 29	1.995\$00
SIMPSONS	1.995\$00
INTERNATIONAL SOCCER	1.995\$00
SINK OR SWIM	1.995\$00
WZIKI	1.995\$00
GRAND PRIX UNLIMITED	1.995\$00
CLOCKWISER	1.995\$00
OPERATION COMBAT 2	1.995\$00
TEST DRIVE III	1.995\$00
BLADE WARRIOR	1.995\$00
EXTERMINATOR	1.995\$00
WRECKERS	1.995\$00
NICEL MANSSELL	1.995\$00
TEAM SUZUKI	1.995\$00
LOTUS III	1.995\$00
PREMIER MANAGER 2	1.995\$00

CD

CARNAGE	1.995\$00
INTERNATIONAL TENNIS	1.995\$00
STACK UP	1.995\$00
SOCCER MANAGER	1.995\$00



DUPLICAÇÃO DE SOFTWARE



TRABALHAMOS COM:

DISQUETES 3,5" DD + HD
5,25" DD + HD

CD-ROM
PROTECÇÃO DE SOFTWARE



DISQUETES PERSONALIZADAS

PROMOVA
A IMAGEM
DA SUA
EMPRESA



ETIQUETAS PERSONALIZADAS

- VÁRIOS TAMANHOS
- GRANDE QUALIDADE
- FAZEMOS A COLAGEM NA DISQUETE

EMBALAGENS PARA SOFTWARE

- GRANDE VARIEDADE DE SOLUÇÕES
- MANIPULAÇÃO DO PRODUTO FINAL
- PRODUÇÃO DE MANUAIS
- TEMOS A SOLUÇÃO QUE PRECISA



DupliSoft

APOIO À PRODUÇÃO DE SOFTWARE, LDA.

Rua António Enes, 18 - c/v Esq^a - 1050 LISBOA

Telefs. 356 08 98

356 09 01

Fax: 315 14 27