Nº 19 AGOSTO DE 1992 PREÇO: 550$00

Spooler

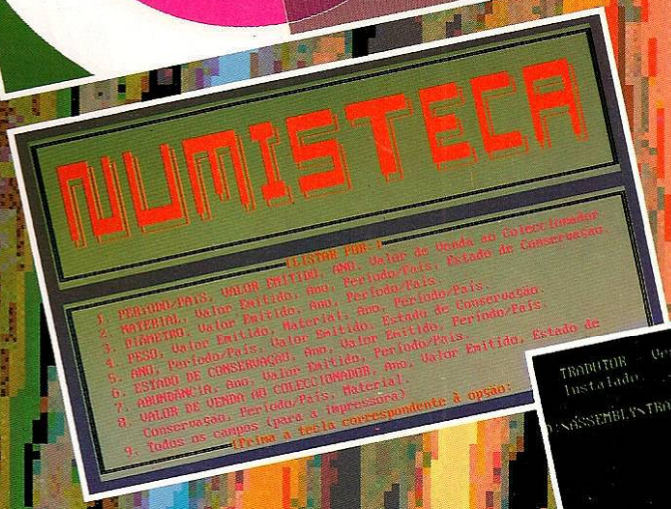


MAGAZINE DOS UTILIZADORES PC

ANIMAÇÃO GRÁFICA



- TECLADO
- HISTORY
- CURSOR/EDITOR
- MEMORANDO
- FERY





TRADUTOR

PROGRAMAR EM:

- ASSEMBLY
- C
- CLIPPER

A disqueta gratuita que acompanha cada exemplar da revista, é parte integrante da mesma e não pode ser comercializada separadamente.

Consuldata
SERVIÇOS DE INFORMÁTICA, LDA.

Graficria PUB

UM MUNDO EM CONSUMÍVEIS PARA INFORMÁTICA

AS MELHORES MARCAS
O MELHOR ATENDIMENTO
A ENTREGA MAIS RÁPIDA
OS MELHORES PREÇOS!!!

DISQUETES DE 3", 3,5", 5,25" E 8"

DATA CARTRIDGES (STREAMTAPES)
DE 10, 20, 45, 60, 150, 250, 320,
525 MEGABYTES E 1,3 GIGABYTES

DISK CARTRIDGES E DISK PACKS
PARA GRANDES SISTEMAS

PASTAS PARA ARQUIVO DE FORMULÁRIOS
DE 9,5x12, 12x9,5, 12x15,5 E 15,5x12
E RESPECTIVOS SEPARADORES

FILTROS ANTI-MAGNÉTICOS E ANTI-ESTÁTICOS
EM REDE E VIDRO PARA TODOS OS TIPOS
DE ECRÃS DE COMPUTADOR

MOBILIÁRIO PRÓPRIO
PARA CENTROS DE INFORMÁTICA

TAPE CARTRIDGES PARA IBM 34800
INCLUINDO CARROS DE TRANSPORTE
E MÓVEIS ESPECIAIS DE ARQUIVO

D. D. S. - 4 MM (1,3 E 2,0 GIGABYTES)
- 8 MM (2,3 GIGABYTES)

E MUITO MAIS!
PARA MAIS INFORMAÇÕES CONTACTE-NOS
OU SOLICITE O CATÁLOGO CONSULDATA

Consuldata - Serviços de Informática, Lda.
Rua Quinta do Almargem, 7 A/B - 1300 LISBOA
Telef.: 364 98 00 Fax: 364 91 08

Nº 19 – Agosto de 1992

Spooler



2 Informática em Notícia

3 Livros

4 O Conteúdo da Disqueta

6 Correio dos Leitores

9 Numisteca

13 Fery Spooler

16 Spoolpack

18 History

20 Cursor Editor

22 Memorando

24 Animação Gráfica

27 Teclado

28 Programar em Assembly (III)

32 Programar em Clipper (III)

35 Programar em C (III)

41 Tradutor (I)

46 Conectividade

50 Montra Spooler

A SPOOLER E A INFORPOR-92

De 22 a 25 de Outubro vai realizar-se, na FIL, a «IX Exposição Internacional de Informática e Tecnologias de Informação». Como não podia deixar de ser, a **Spooler** estará mais uma vez presente no certame para, com alegria, receber os seus muitos amigos, assinantes e leitores. E também para se revelar a quantos ainda não a conhecem «pessoalmente». Não deixem de nos visitar.

CINQUENTA E SEIS PÁGINAS?!?

É verdade. Mas não se assustem ou entusiasmem os nossos amigos leitores, consoante a forma como encararem a questão. Tratando-se do número que antecede a INFORPOR-92, tivemos de satisfazer compromissos com empresas que pretenderam inserir publicidade nas nossas páginas. E como a publicidade ocuparia espaço normalmente reservado aos artigos, só nos restou uma solução – a mais cara, mas a única que salvaguardaria o «nome» da **Spooler** – aumentar o número de páginas em conformidade. Se, como esperamos, a tendência se mantiver, então o caso deixará de ter um carácter excepcional para se tornar numa situação normal. A ver vamos.

A DISQUETA SPOOLER

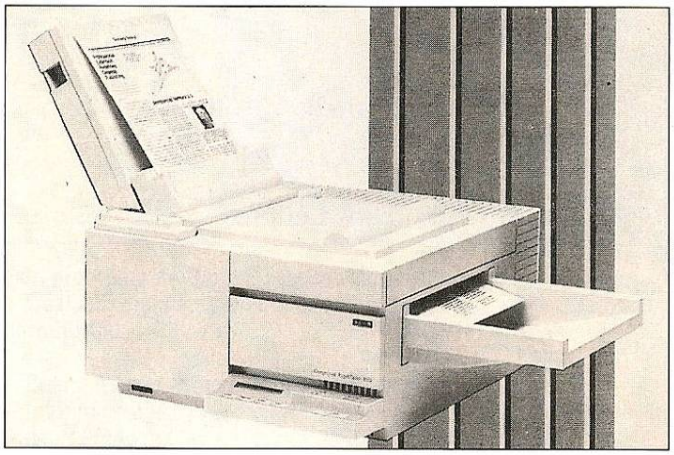
Os leitores já se habituaram às nossas constantes lamentações a respeito da «ginástica» que somos obrigados a fazer para meter na disqueta o material desejado. Nem sempre conseguimos e o resultado por vezes não é o melhor . Temos recorrido a vários «expedientes», o último dos quais foi a eliminação dos directórios. Outras vezes optamos por não incluir as «fontes» com a promessa da sua publicação no número seguinte. E temos cumprido. Mas, enquanto as circunstâncias nos impedirem de adoptarmos a disqueta de 3,5" como formato *standard*, (um sonho antigo que um dia será realidade), teremos de continuar a «inventar» formas de diminuir a dimensão dos programas. A disqueta do próximo número da **Spooler** beneficiará de um novo «expediente»: os ficheiros deixarão de ser autodescompactáveis, ficando só no formato comprimido. A antecedência com que fazemos este anuncio destina-se a preparar o «espírito» do leitor – em última análise o grande beneficiário desta decisão. Voltaremos a esta questão, com mais pormenores, na **Spooler Nº 20**.

Renato Casquilho

Spooler

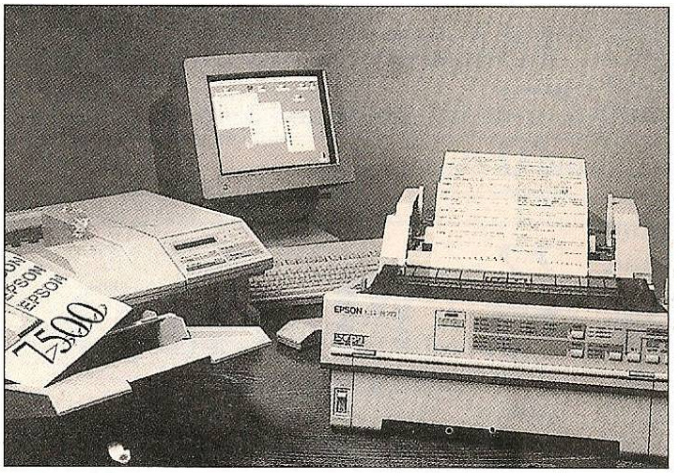
- **Propriedade:** Graficria, Publicidade e Artes Gráficas, Lda.
- **Director de Edição:** Franco Gomes ■ **Director**: Renato Casquilho
- **Redacção:** Franco Gomes, Paulo Bernardoe Renato Casquilho.
- **Colaboradores:** Carlos Ladeira, Cunha Brandão, José Orlando Pereira, José Páscoa, Paulo Bernardo, Vítor Monteiro.
- **Ilustrações:** Leonor Gomes
- **Relações Públicas e Publicidade:** Fátima Ribeiro
- **Assinaturas e encomendas:** Paula Martinez
- **Redacção, Publicidade e Serviços Administrativos:** Rua Duarte Pacheco Pereira, 69-A – Damaia – 2700 AMADORA – Telfs.: 475 22 89/474 60 11/41
- **Produção Gráfica:** Graficria, Publicidade e Artes Gráficas, Lda.
- **Impressão:** Empresa Litográfica do Sul, S.A. ■ **Distribuição:** Electroliber
- **Revista bimestral** ■ **Preço:** 550$00 ■ **SRIP N.º** 213900
- **Tiragem:** 12500 exemplares ■ **Dep. Legal N.º:** 28587/89

INFORMÁTICA EM NOTÍCIA



IMPRESSORAS BULL-COMPUPRINT LASER

A CIL Centro de Informática, Lda. anunciou novas impressoras Bull-Compuprint, com tecnologia laser, velocidades de 7 a 10 páginas por minuto que apresentam a particulariedade de imprimirem na frente e verso da folha.



MACINTOSH E IMPRESSORAS EPSON

Os utilizadores de produtos Macintosh ou Quadra podem agora aceder às impressoras Epson de todos os tipos, graças aos *packages de conexão*, recentemente apresentados pela Epson Portugal.

NOTEBOOK COMPAQ 386SX/20

Com processador 30386 SX funcionando a 20 Mhz, 4 Kb de memópria *cache*, 2 Mb de memória RAM, 60 Mb em disco rígido, monitor VGA (640x480) e DOS 5.0, este portátil da Compaq é comercializado pela Computerland ao preço notável de 301 600$00 (IVA incluído).

BBSpooler

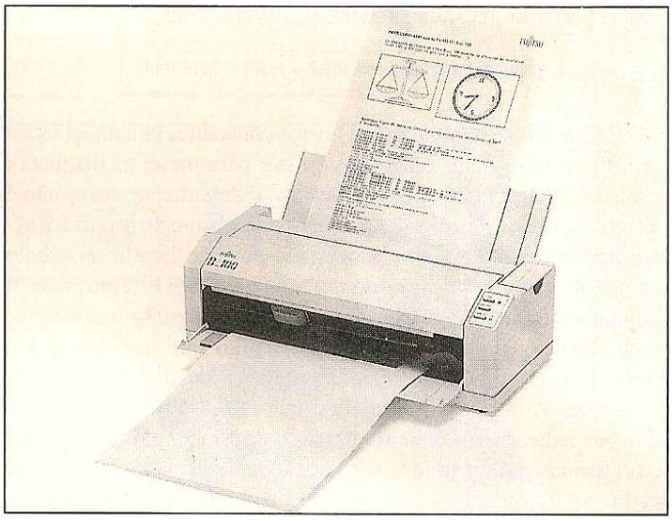
☎ (01) 475 22 89

Dias úteis: das 18.30 às 9.00 horas
Feriados e fins-de-semana: 24 horas

DIODE APRESENTA CONTROLADOR SCSI DA ADAPTEC E IMPRESSORA FUJITSU B-100



Com o objectivo de ultrapassar a limitação de capacidade nos ambientes DOS, a Adaptec apresentou o controlador AHA-1542B/GIG KIT.
Este controlador, para disco ou disqueta, possui uma velocidade de 10 Mbytes/seg., permitindo ultrapassar a limitação de capacidade de 1 Gb do DOS que, como se sabe, estava limitado a este valor por condicionamento aos 1024 cilindros por disco.
Este dispositivo da Adaptec utiliza 255 cabeças e 63 sectores por pista, parâmetros que aplicados aos tais 1024 cilindros proporcionam capacidades até 8 Gb.
Outra novidade anunciada foi o AHA-1742KIT, controlador BUS EISA (32 bits) com uma velocidade de tranferência superior a 33 Mb/seg., que incorpora também controlador de *floppy* e apresenta-se num conjunto completo, com cabos, manuais de instalação e software, para trabalhar em ambientes DOS, OS/2, e Netware 386 versões 3.0 e 3.11.
Este produtos e outros da Adaptec são representados no nosso país pela DIODE.

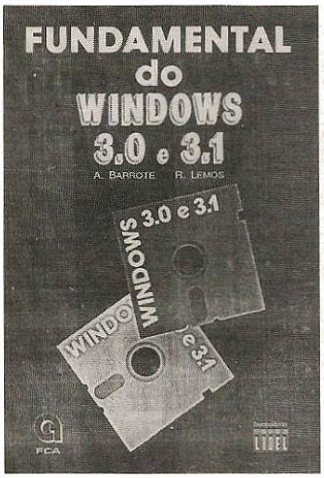


Um outro produto distribuido por esta empresa e recentemente anunciado é a impressora de injecção de tinta Fujitsu B-100. O único consumível necessário para esta impressora é a recarga de tinta que incorpora também a cabeça de impressão, garantindo, deste modo, a qualidade constante de impressão.
Entre outras destacamos as seguintes características: baixo nível de ruído (menos de 45 dBA), baixo custo, resolução de 300 pontos por polegada, velocidade de impressão de 1,44 ppm e 160 cps em modo *draft*, pequena dimensão (342x147 mm – 2,5 Kg de peso), 14 tipos de letra residentes e vários «IC Card Fonts e Softfonts» opcionais.

COMPUTADORES ABC

A ABC – Informática é uma empresa do grupo Casa Viola, que se dedica à montagem de computadores de marca própria.
Com componentes importados de vários países (E. U. A., Japão, etc.) a ABC procede à montagem de configurações com processadores 80386 e 80486, memórias RAM de 1 a 64 Mb, e discos rígidos de 20 MB a 4 Gb. Oferecendo um prazo de garantia ao equipamento de 2 anos e de assistência técnica por 10 anos, a ABC aposta na qualidade dos componentes e rigor de montagem.
Quanto ao *software*, a ABC só dispõe de programas originais.

Livros



FUNDAMENTAL DO WINDOWS 3.0 E 3.1

Os autores basearam-se na experiência adquirida com a utilização diária do Windows para dedicarem esta obra aos que, pela primeira vez, se encontram envolvidos neste poderoso «ambiente». Entre os principais temas abordados, salientam-se:

- Como instalar o Windows
- Como personalizar o Windows
- O coração do Windows: *Program Manager* e *File Manager*
- O painel de controlo
- Os utilitários do Windows.

Título: Fundamental do Windows 3.0 e 3.1
Idioma: Português
Autores: A. Barrote e R. Lemos
Edição: FCA – Editora de Informática, Lda
Distribuição: LIDEL, Edições Técnicas, Lda
I.S.B.N: 972-722-002-9
Preço: 1.420$00



LINGUAGEM ASSEMBLER ORIENTADA PARA OBJETO

O autor, Len Dorfman, é escritor, programador e accionista de uma empresa para desenvolvimento de *software*. Em 1988, o seu programa PC TYPESETER recebeu o prémio «Melhor de 1988» atribuído pela revista *PC RESOURCE*. Neste livro, mostra-nos como simular a estrutura de linguagem de alto nível na programação orientada para objectos. Começa por uma discussão dos conceitos da programação orientada para objectos e descreve a sua aplicação prática na programação *Assembly*. Eis algumas das técnicas apresentadas:

- Como criar classes, métodos e objectos
- Como escrever código reutilizável
- Como simplificar rotinas para manipulação de cadeias
- Como escrever caracteres e cadeias de texto no ecrã
- Como ligar e desligar o modo de impressão «condensado»
- Como construir janelas
- Como escrever programas para carta EGA
- Como criar ícones

Título: Linguagem Assembler Orientada para Objeto
Idioma: Português
Autor: Len Dorfman
Tradução: Mário Moro Fecchio
Edição e Distribuição: Editora McGRAW-HILL de Portugal, Lda
I.S.B.N: 0-07-460614-X

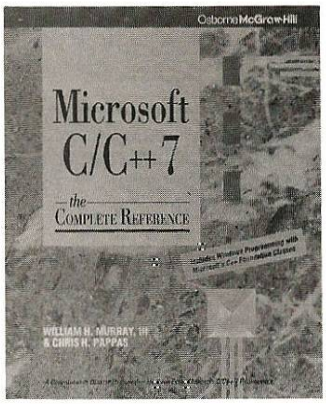


MICROSOFT C – DICAS, SEGREDOS E TRUQUES

O livro destina-se, essencialmente, aos principiantes na programação da linguagem C. Mas também revela, aos mais experimentados, inúmeras «dicas» fruto da experiência do autor. Eis os principais tópicos do livro:

- Construindo os fundamentos
- Além do básico
- Libertando o potencial do C
- Tornando-se profissional
- Glossário compilado
- Índice analítico

Título: MICROSOFT C – Dicas, Segredos e Truques
Idioma: Português
Autor: Kris Jamsa
Edição e Distribuição: Editora McGRAW-HILL de Portugal, Lda
I.S.B.N: 0-07-460801-0



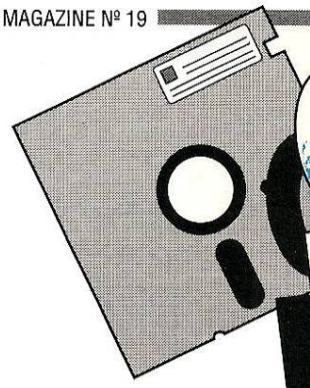
MICROSOFT C/C ++ 7 – THE COMPLETE REFERENCE

Os autores especificam dois grandes objectivos: familiarizar o leitor com o *package* Microsoft C/C++ e auxiliar os programadores com diferentes graus de conhecimento a tornarem-se mais eficientes e a programar para o Windows. O livro cobre todas as potencialidades do Microsoft C/C++ 7, incluindo programação para Windows com as *Microsoft's C++ Foundation Classes*. Também inclui inúmeros exemplos, através dos quais o principiante pode construir aplicações para imediata utilização. Para além dos fundamentos, o livro desenvolve as seguintes técnicas:

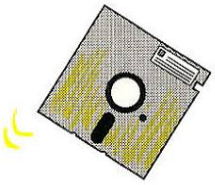
- O compilador C/C++
- A mesa de trabalho do programador
- O editor/depurador *CodeView*
- O examinador de código-fonte *Source Profiler*
- As ferramentas especiais para criação de aplicações Windows.

Título: Microsoft C/C++ 7 The Complete Reference
Idioma: Inglês
Autores: William H. Murray, III & Chris H. Pappas
Edição e Distribuição: Editora McGRAW-HILL de Portugal, Lda
I.S.B.N.: 0-07-881664-5

Estes e outros livros sobre informática editados pela **Sybex, QUE, Ziff, Editorial O Livro, FCA, Edições Cetop** 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.



O CONTEÚDO DA DISQUETA



ANIMAEXE.EXE

O Rui Anastácio jé é nosso conhecido pelo interesse e qualidade dos seus programas. Portanto, não precisa de apresentação. Oferece-nos, desta vez, uma curiosa demonstração das capacidades do Pascal na computação e animação gráfica.



BMOUSE.EXE

O nosso leitor Victor Manuel Peres Pereira adquiriu-nos uma disqueta de *shareware* com utilitários para a integração do «rato» em programas Clipper. Teve dificuldades na utilização do utilitário «BMOUSE» e pediu-nos ajuda. Segundo a opinião do nosso colaborador Eng. José Páscoa, o dito utilitário não estaria completo. Na disqueta, segue o utilitário devidamente corrigido, destinando-se, igualmente, a todos os leitores que compraram a disqueta Ref.ª SH18/19.



CLIP3.EXE

No número anterior da Spooler dissemos que publicaríamos as «fontes» do programa «AGENDA» no presente número. Assim é, com uma pequena diferença que pensamos não prejudicar o leitor: as «fontes» presentes na disqueta referem-se à nova versão aqui descrita pelo nosso colaborador Victor Monteiro e não à versão anterior.



CURSOEXE.EXE

O nosso leitor Armando Frazão preocupou-se com a «sanidade mental» dos utilizadores do rato. E muito a propósito pois, há dias, soubémos de um caso, aliás triste, de um senhor que, de tanto ver aquela «setinha» na sua frente, cada vez que via uma seta corria a fugir pensando que era um rato. Enfim. Tivemos de o recomendar a uma instituição adequada...



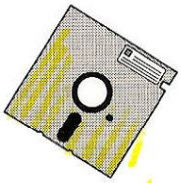
C_3.EXE

Esta é a nova versão da calculadora, agora bem mais complicada que a anterior. Foi introduzido um interpretador de expressões que utiliza um algoritmo recursivo para processar a expressão. Mas leia a lição para mais pormenores.



FERYPRG.EXE

O nosso amigo Herberto Brazão é, decididamente, um «spoolarista» de gema. Claro que ficámos sensibilizados (somos de carne e osso, como toda a gente) com o programa que nos enviou para apreciação. Mas o que verdadeiramente pesou na decisão de o publicar foi o facto de constituir mais uma demonstração da arte de programar em dBASE e da «transição» para o Clipper. O autor enviou-nos o executável, mas a sua dimensão era de tal ordem que pouco espaço sobrava para outro material. E então pensámos: «Porque não aproveitar a deixa e «obrigar» os leitores a aprender a compilar o programa? – Pois o Herberto Brazão até manda um «batch» com todas as instruções para a compilação?» Dito e feito. Na disqueta está tudo o que é preciso.





HELP.EXE

O prometido é devido e aqui vão as «fontes» do executável publicado no anterior número da Spooler.

HISTEXE.EXE

Miguel Gonçalves é um nosso fiel leitor e activo utilizador da **BBSPooler**, a nossa/vossa BBS. O programa que nos enviou foi considerado merecedor de publicação e a oportunidade verificou-se neste número da **Spooler**. Leia atentamente o artigo para ficar a saber como instalar e utilizar o seu programa.

MEMO.EXE

Eis um programa escrito em Pascal, especialmente orientado para os utilizadores que apreciam o «método e a organização» nas suas tarefas diárias. O título diz quase tudo. O restante encontra-se no artigo do Rui Pestana.

MESTRE2.EXE

Vários leitores mostraram-se interessados na possibilidade de imprimir o conteúdo das bases de dados do programa MESTRE. O nosso colaborador Victor Monteiro, autor do programa, elaborou uma pequena rotina em dBASE III Plus que permite essa impressão em diferentes formatos. Veja as explicações na rubrica «Correio dos Leitores».

NUMISTEC.EXE

Decididamente, o Afonso Infante especializou-se em programas para coleccionadores. Depois dos filatelistas e dos amantes de livros, é a vez dos numismatas serem presenteados com uma explêndida aplicação destinada a gerir as suas moedas.

SPOOLPAC.EXE

Mais uma demonstração das capacidades do BASIC, para quem se dá ao trabalho de estudar a fundo esta linguagem aparentemente «simplória». O Ricardo e o Luciano mostram-nos como se pode realizar um interessante grafismo sem recurso às linguagens de «elite»...

TECQCD.EXE

Tal como prometido, aqui estão os outros dois utilitários referidos no anterior número da Spooler. São eles «TECLADO» e »QCD». É obrigatório ler o artigo do Nelson Pires.

TRADUTOR.EXE

O Franco Gomes começa assim o seu artigo: *«Este programa, concebido em linguagem Assembly, é um TSR (Terminate and Stay Resident). Para quem não esteja familiarizado com estes anglicismos, digamos que se trata de um programa que, depois de executado, fica em ''hibernação'' na memória do processador, num sono de ''bela adormecida'' aguardando a beijoca despertadora do ''príncipe encantado''»*. Quem, depois disto, não ficou «com água na boca...» Vá, refreie o seu natural impulso para descompactar o programa e corra a ler o artigo.

FUJITSU

Periféricos

Impressionado com as Matriciais Fujitsu



Pergunte pelas Impressoras Fujitsu

FUJITSU

Liberdade de Impressão

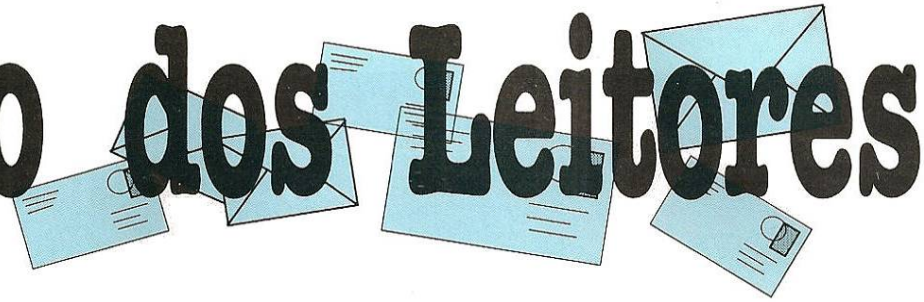
- Ampla gama em 24 agulhas cobrindo as necessidades mínimas e máximas.
- Velocidade de 180 a 505 caracteres/segundo. Alta fiabilidade com MTBF* até 8.000 horas.
- Impressão de formulários e impressos até 7 vias.
- Programação desde PC por menús em português.

(*) Tempo médio entre avarias.

Porque na FUJITSU há uma impressora por cada nível de necessidade.
Porque a sua tecnologia lhe permite conseguir um produto silencioso, rápido e versátil. Tudo o que possa imaginar encontrará na gama FUJITSU.

Comprove-o no seu distribuidor e ficará interessado.

Correio dos Leitores



Por Renato Casquilho

Vários leitores mostraram-se interessados na possibilidade de imprimirem o conteúdo das bases de dados do programa MESTRE. O nosso colaborador Victor Monteiro, autor do programa, elaborou uma pequena rotina em dBASE III Plus que permite essa impressão em diferentes formatos. Eis a sua explicação:

«Sendo o programa utilitário MESTRE uma base de dados onde é possivel guardar qualquer tipo de informação, resolvi (seguindo a sugestão de alguns leitores) escrever um pequeno programa em dBase III Plus que permita passar a informação contida nas bases de dados, para a impressora.

O programa é auto-explicativo, no entanto, gostaria de fazer uma pequena introdução. São quatro os ficheiros que compõem o programa:

MESTRE2.PRG – Ficheiro do programa
MESTRE2.LBL – Ficheiro das etiquetas, por defeito
MESTRE21.LBL – Ficheiro das etiquetas simples
MESTRE22.LBL – Ficheiro das etiquetas duplas

Para executar o programa, copie estes ficheiros para a directoria do dBase e execute-o. Na linha de comando digite:

DO mestre2

O programa perguntará se deseja etiquetas simples (1) ou duplas (2) e a directoria do MESTRE. Surgirá então uma listagem dos temas precedidos por um número. É o número do tema que deseja imprimir que deverá ser indicado à pergunta: «Qual o número do tema que deseja imprimir?».
Caso tenha mais do que 50 temas, digite 00 (ou ENTER) para aceder à segunda página. Para abandonar o programa introduza um número negativo.»

O nosso leitor Victor Manuel Peres Pereira adquiriu-nos uma disqueta de *shareware* com utilitários para a integração do «rato» em programas Clipper. Teve dificuldades na utilização do utilitário «BMOUSE» e pediu-nos ajuda. Segundo a opinião do nosso colaborador Eng. José Páscoa, o dito utilitário não estaria completo. Aqui vai a resposta e, na disqueta, segue o utilitário devidamente corrigido, destinando-se, igualmente, a todos os leitores que compraram a disqueta Ref.ª SH18/19.
«O leitor tem toda a razão pois o módulo BMOUSE não implementou o movimento horizontal. Aparentemente o respectivo autor pensou fazê-lo mas por algum motivo acabou por desistir. Introduzimos algumas pequenas correcções na fonte em *Assembly* e assemblámos com o MASM 6.0 e TASM 3.0 sem problemas. As dificuldades do leitor na assemblagem com o MASM podem ser devidas à utilização de uma versão antiga do dito.
Naturalmente que o leitor se apercebeu que o BMOUSE actua «introduzindo teclas» no *buffer* do teclado. Por esse motivo, e porque o Clipper também responde no menu horizontal do programa MENU.PRG às teclas Cursor Up e Cursor Down com movimento horizontal, então o mesmo efeito também é produzido com o movimento vertical do rato.
Seguidamente recompilámos o MENU.PRG com o Clipper 5 e ligámo-lo ao BMOUSE.OBJ e às as livrarias do Clipper com o TLINK da Borland e obtendo o MENU_2.EXE que enviamos na disquete.»

1) Possuo um PC/XT com um processador 8088 e 640 KB de RAM e que segundo o fornecedor seriam extensíveis a 1 MB. Ora pelo que sei, o 8088 só pode endereçar até 1 MB já com os 384 KB reservados a adapter cards *ROM BIOS, etc.*
2) E na sequência da questão anterior como se poderá aceder então aos UMBs (Upper Memory Blocks) existentes no meu XT, já que actualmente o DOS 5.0 fornece um gestor de memória alta (o HIMEM.SYS).

Luis Manuel de Sousa F. Diniz
Sacavém

Mais uma vez recorremos à «sapiência» do nosso colaborador Eng. José Páscoa. Eis a sua resposta: «O leitor tem razão e o vendedor do equipamento aparentemente só lhe terá dito uma parte da verdade. Alguns computadores que utilizam os processadores 8088 e 8086 possuem um **chipset** que permite mapear RAM na zona normalmente reservado ao ROM, isto é, os tais 384 KB que estão reservados a *adapter cards* e BIOS como muito bem referiu. Naturalmente que o leitor só poderá colocar os seus programas naquelas zonas de alta memória que ficarem livres. Existem também umas placas de memória expandida de acordo com a especificação EEMS que proporcionam um resultado semelhante. Quer num caso quer noutro, fique ciente o leitor que com o processador 8088 nunca, mas nunca mesmo, poderá endereçar acima de 1 MB.

O HIMEM.SYS na sua versão actual não cria UMBs, o programa que o faz é o EMM386.SYS, e este programa exige, como se sabe, possuir-se no mínimo um sistema 386 SX. Se efectivamente o XT do leitor possui um **chipset** que permite mapear RAM na zona do ROM tente a sua sorte com um programa de nome LastByte (a versão 2.0 existe nas BBS com o nome TLBV200.ZIP).»

... escrevo-lhes para lhes perguntar o que é uma rede NOVELL e quais as suas características... Outra pergunta será qual a diferença entre terminais estúpidos e terminais "«nteligentes», sobre os quais existe tanta celeuma nas discussões de restaurante.

João Paulo Gomes Martins

Pedimos ao nosso colaborador Eng.Paulo Bernardo que preparasse uma resposta a este nosso leitor. Pelo seu conteúdo didático, considerámos ser de interesse geral, motivo da sua inclusão nesta rubrica:
«O NOVELL NETWARE é um sistema operativo de rede, e não uma rede propriamente dita. As suas principais características são a elevada rapidez, segurança e ... preço! O NETWARE exige também um *server* dedicado, totalmente sob o seu controlo, onde o disco é formatado com um *file-system* próprio, que não o DOS. Os postos de trabalho (as estações da rede) são PCs vulgares correndo um sistema operativo que tanto pode ser DOS como OS/2 e onde o acesso à rede é garantido por um conjunto de programas residentes, que tratam das comunicações. Todos os dispositivos que estiverem ligados ao *file-server* podem ser partilhados: discos, impressoras, *scanners*, etc... As estações podem possuir dispositivos próprios (impressoras ou discos) que, no entanto, serão sempre locais, não sendo possível a sua partilha. Apesar do NETWARE permitir que o *file-server* seja usado simultâneamente como estação de trabalho, esta dualidade de funções é desaconselhada por questões de segurança e quebra de *performance* global da rede.
Aos olhos do utilizador de um dado posto de trabalho, a rede aparece como um conjunto de discos extra (F:, G:, H:, etc...) – estes discos não existem na realidade, pois são parcelas do disco do *file-server*, com o conteúdo e dimensão determinados pelas permissões concedidas a esse utilizador.
De todo o *software* que está instalado na rede, as estações apenas podem executar aquele que «veêm» nos discos, ou seja, aquele para o qual têm permissões. Quando as aplicações estão instaladas nas suas versões de rede, os diferentes utilizadores podem partilhar os dados (por exemplo, uma mesma base de dados ou folha de cálculo). Existem muitos outros sistemas operativos de rede, a maioria a preços inferiores ao NETWARE: LAN SERVER (IBM), LAN MANAGER (MICROSOFT), D-LINK LANSMART (LANCOM INTERNATIONAL), LANTASTIC (SOFTWARE PARADISE), etc.. Como é evidente, não apresentarão as *performances* do NETWARE, mas resolvem o problema da maioria das empresas onde as questões de segurança não sejam demasiado prementes – o NETWARE é incrível no que concerne às permissões, pois premite definir níveis de acesso, por utilizador, ao nível dos próprios ficheiros!
Para além do NETWARE, que é o líder mundial do mercado dos S.Os de rede, começam a popularizar-se em Portugal as redes ponto-a-ponto (LANtastic e LANsmart) que têm a característica de não necessitarem de um *file-server* dedicado, pois partilham os discos rígidos de todos os computadores da rede entre todos os utilizadores. Estes sistemas operativos de rede apresentam as grandes vantagens de serem, à partida, bastante mais baratos que o NETWARE, além de não exigirem a aquisição de um *file-server* dispendioso e aproveitarem com sucesso o «parque instalado» de computadores... Quanto à questão dos terminais «estúpidos» e «inteligentes», ela não se põe no caso das redes locais. Um terminal estúpido é essencialmente constituído por um teclado, um ecrã e um módulo *hardware* de comunicações, que envia através de um cabo as teclas que foram premidas e recebe pelo mesmo cabo os caracteres a escrever no ecrã.
Um terminal deste género pode ser mais ou menos evoluído, mas não possui capacidade de processamento (não tem CPU) razão pela qual é chamado «estúpido». A arquitectura de um sistema informático baseado no sistema operativo UNIX é normalmente constituída por um computador «central» com um CPU i386 ou i486 e um conjunto de terminais «estúpidos», razão pela qual o seu preço é bastante competitivo para quem pretende um ambente multi-utilizador. Uma LAN (Local Area Network) é constituída por um conjunto de computadores (portanto «inteligentes», por contraste com os outros) interligados entre si e, eventualmente, ligados a um *file-server*. Para aproveitar a proliferação das redes locais, alguns fabricantes vendem agora computadores concebidos especialmente para serem estações de rede – não têm disco nem drive e já apresentam instalada a placa de comunicações de rede – mas são verdadeiros PCs, funcionando em DOS ou OS/2.»

CLIPPER 5.01
GUIA DO PROGRAMADOR

A GRAFICRIA tem o prazer de anunciar mais uma edição dedicada aos programadores. Desta vez os contemplados são os entusiastas do Clipper. O livro, da autoria de Renato Casquilho, 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 correcções introduzidas pela versão 5.01.
A obra é acompanhada por uma disqueta cujo conteúdo constitui uma demonstração das inúmeras capacidades desta linguagem. O Capítulo 6 é inteiramente dedicado à discussão dos diferentes módulos de um programa destinado à gestão de *software,* desde a concepção à utilização. Para além do executável pronto a correr em qualquer computador pessoal, a disqueta contém os módulos no formato .PRG, a base de dados, os ficheiros indexados e um ecrã de abertura no formato .COM. Não foram esquecidos os ficheiros .RMK e .LNK destinados a tirar todo o partido do utilitário RMAKE constante do pacote Clipper 5.01.
O código-fonte deste programa, também apresentado no formato .TXT com números de linha serviu, sempre que possível, como exemplo da sintaxe, dos argumentos e da utilização dos diferentes comandos, funções e declarações.
Eis um resumo dos capítulos do livro:

Capítulo 1 – Configuração do sistema
Capítulo 2 – Compilação e ligação
Capítulo 3 – Declarações
Capítulo 4 – Comandos
Capítulo 5 – Funções
Capítulo 6 – O programa SOFTECA
Apêndice A – Mensagens de erro
Apêndice B – Códigos ASCII e valores para INKEY( )
Apêndice C – Comandos e funções do dBASE incompatíveis com o Clipper.

O lançamento do «Clipper 5.01 – Guia do Programador» terá lugar durante a INFORPOR-92, no *stand* **Spooler**, onde o autor estará presente para satisfazer todas as solicitações de esclarecimentos e autógrafos.

Graficria
Publicidade e Artes Gráficas, Lda.

Rua Duarte Pacheco Pereira, 69-A
Damaia-de-Baixo – 2700 AMADORA
Telefs.: 474 60 11/474 60 41 – Fax: 475 22 89

INVISTA NA TECNOLOGIA QUE IMPRESSIONA

Sim, a tecnologia FUJITSU impressiona. As novas impressoras DL900 e DL1100 de 24 agulhas pensam em grande. Imprimem em A3 terminando de vez com as limitações que a maioria das impressoras A4 tradicionais. De facto são ligeiramente maiores do que o papel onde imprimem. Não podem ser culpadas de lhe roubar espaço de secretária. Apesar do tamanho possuem a maior memória existente em impressoras de secretária. 24 KBytes mais precisamente podendo armazenar 11 páginas típicas de dados. Graças ao programa DLMENU em português, programá-las do seu PC é extremamente simples e cómodo. Com tudo isto, é uma pena se gasta dinheiro em impressoras menos avançadas. Principalmente se pode ter a sua FUJITSU DL900 por 72 000$00 a DL1100 por 92 000$00 e por 115 000$00 pode ter a DL1100 cor (preços sem I.V.A.). Consulte o seu revendedor autorizado fujitsu e conheça a tecnologia que impressiona.



GARANTIA 2 ANOS

FUJITSU

À VENDA NA «MONTRA SPOOLER»

DIODE Grupo Getronics
Rua Rodrigo da Fonseca, n.º 72-4.º - Dto. – 1200 LISBOA – Tels. 386 41 57 - 386 41 65 - 386 31 58 - 386 31 64 • Fax 386 09 87

NOVA SÉRIE INVES PC



IVES PC-325 S — **Desde 187 000$00 + IVA**

- roc. 80386SX-33 MHz
- 1 a 8 Mb RAM

PROMOÇÃO DE LANÇAMENTO

320 000$00 + IVA





INVES NB-320S (Notebook)

- Processador 80386SX-20 MHz
- 2 a 4 Mb RAM
- 2 Portas série, 1 Paralela, 1 Game
- Porta 8 pinos para Handy Scanner
- Drive 3 ½" – 1,44 Mb
- Disco rígido 2½" – 80 Mb
- Ecrã cristais líquidos contra-iluminados VGA 640x480, 32 tonalidades de cinzento
- Socket para 80387SX
- Teclado 81 teclas, 12 funções
- Dimensões: 290x220x53 mm
- Peso: 2,6 Kg

INVES PC-340D — **Desde 257 000$00 + IVA**

- Proc. 80386DX-40 MHz
- 2 a 64 Mb RAM

INVES PC-433D — **Desde 347 000$00 + IVA**

- Proc. 80486DX-33 MHz
- 2 a 64 Mb RAM

INVES PC-450D — **Desde 432 000$00 + IVA**

- Proc. 80486DX-50 MHz
- 2 a 64 Mb RAM

OUTRAS CONFIGURAÇÕES POSSÍVEIS (todos os modelos):

- Drives de 3½" - 1,44 Mb e 5¼" - 1,2 Mb
- Discos de diversas capacidades
- Placas vídeo VGA de 256 Kb a 1 Mb RAM
- Monitores VGA mono ou SuperVGA (1024x768)

GARANTIA 2 ANOS

CONDIÇÕES ESPECIAIS DE PAGAMENTO

SAT
SERVIÇO ASSISTÊNCIA TÉCNICA
TODAS AS MARCAS
Tel. 778 73 24
Fax 778 53 01

inves

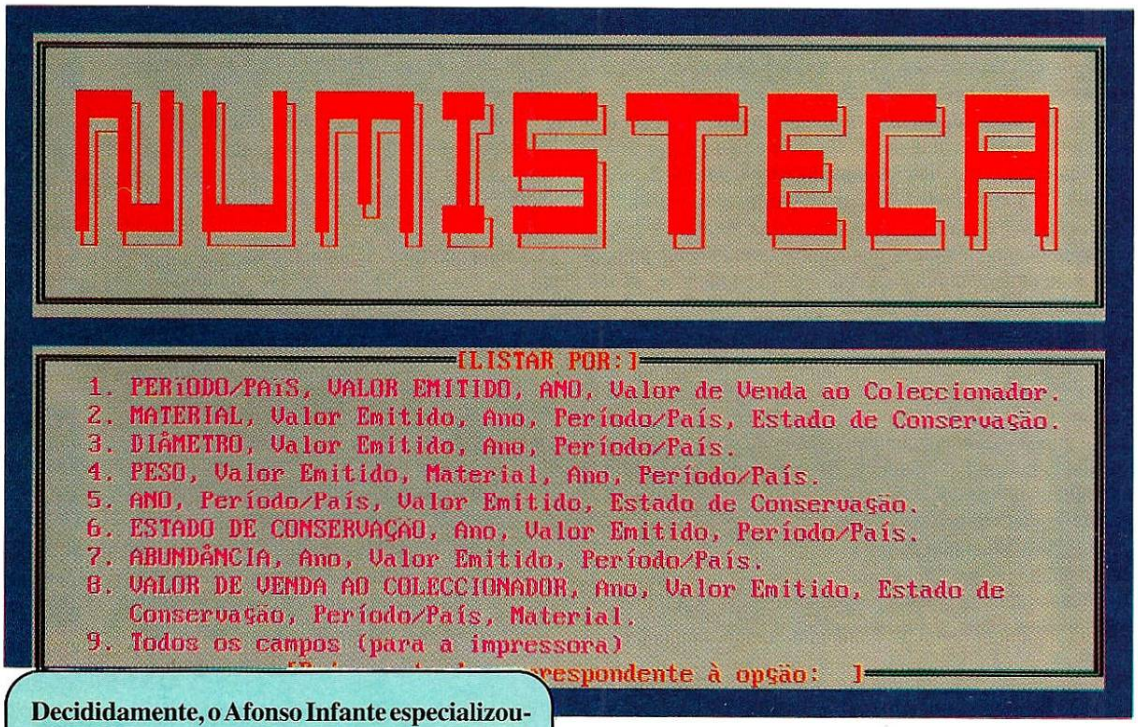
REPRESENTANTE OFICIAL:

INVESMÁTICA – Investigação e Informática, Lda.
Rua Abel Botelho, 8-A – 1500 LISBOA – PORTUGAL
Telefs.: 351-1-7740880/7787324/7787326 – Fax: 351-1-7785301

Graficria pub

NUMISTECA

COLABORAÇÃO DO LEITOR



Decididamente, o Afonso Infante especializou-se em programas para coleccionadores. Depois dos filatelistas e dos amantes de livros, é a vez dos numismatas serem presenteados com uma explêndida aplicação destinada a gerir as suas moedas. O artigo deste nosso amigo, como habitualmente, está tão completo que me dispenso de mais comentários.

Depois de ter feito o FILATECA, programa destinado aos coleccionadores de selos, e o BIBLIOTECA, destinado aos coleccionadores de livros, resolvi criar um programa destinado aos coleccionadores de moedas.

Resolvi dar-lhe o nome que, no fundo, serviu de base a outros programas já aparecidos nesta revista: NUMISTECA. Foi realizado em dBASE III Plus, e não está nem compilado, nem pseudo-compilado, nem linkado.

O PROGRAMA

É constituído pelas seguintes rotinas:

NU_S1_C.COM	Menu principal para monitores a cores
NU_S1_BW.COM	Menu principal para monitores a preto e branco
NU_S2_C.COM	Submenu da opção 2 para monitores a cores
NU_S2_BW.COM	Submenu da opção 2 para monitores a preto e branco
NUMISTEC.PRG	Rotina principal
NU_ENTRA.PRG	Opção 1, Entrada de dados
NU_LIST.PRG	Opção 2, Listar todo o ficheiro
NU_PVEM.PRG	Opção 3, Procurar pelo valor emitido
NU_PPRPS.PRG	Opção 4, Procurar pelo período/país
NU_PMAT.PRG	Opção 5, Procurar pelo material
NU_PANO.PRG	Opção 6, Procurar pela ano
NU_PECON.PRG	Opção 7, Procurar pelo estado de conservação
NU_EMEND.PRG	Opção 8, Emendar registo
NU_APTOD.PRG	Opção 9, Apagar todos os registos
NU_APUM.PRG	Opção 10, Apagar um registo
NU_SAIR.PRG	Opção 11, Sair do programa

A Base de Dados (MOEDAS.DBF) tem os seguintes campos:

Campo	Nome do Campo	Tipo	Comprimento
1	VALOR EMITIDO	Caracter	20
2	PERÍODO/PAÍS	Caracter	20
3	MATERIAL	Caracter	15
4	DIÂMETRO	Numérico	2
5	PESO	Numérico	6
6	ANO	Caracter	4
7	CONSERVAÇÃO	Caracter	4
8	ABUNDÂNCIA	Caracter	4
9	V DE VENDA	Numérico	6
10	COLOCAÇÃO	Caracter	40
		TOTAL	121

Função de cada campo:

Valor emitido: O valor que a moeda tinha quando em circulação, ou seja, o valor inscrito na moeda (Exemplos: 5 Escudos, 500 Reis, 1 Rupia, etc.)

Período/País: O período político em que a moeda foi emitida e/ou simplesmente o país em que foi emitida (Exemplos: D.Pedro V/Portugal, República/Portugal, Índia, Angola, etc.).

Material: O metal (ou liga metálica) de que é constituída a moeda (Exemplos: Ouro, Prata, Cobre, Bronze, Latão-Níquel, etc.).

Diâmetro: O diâmetro (em milímetros) da moeda (Exemplos: 17 mm, 18 mm, 23,5 mm, etc.).

Peso: O peso (em gramas) da moeda. Possui espaço para 2 casas decimais. (Exemplos: 7.00 g, 21.50 g, 17.00 g, etc.).

Ano: O ano de emissão da moeda (Exemplos: 1851, 1906, 1985)

Estado de Conservação: O estado de conservação da moeda. Seguindo o standard da numismática, deverão usar-se 7 graus de classificação:

1. FDC – Flor-de-cunho

Moeda absolutamente impecável, sem apresentar o mais pequeno vestígio de manchas, mossas ou qualquer desgaste.

2. SOB – Soberba

Moeda que nunca circulou, mantendo o brilho ou o baço provocados pela cunhagem. Pode apresentar ligeiras marcas de abrasão por ter estado em contacto com outras moedas durante o fabrico ou transporte.

3. BELA

Moeda que aparenta não ter circulado. Ligeiros sinais de desgaste nos pontos mais latos do relevo da gravura, alguma oxidação ou pátina uniforme.

4. MBC – Muito Bem Conservada

Moeda pouco circulada. Superfícies mais elevadas da gravura com sinais de desgaste. Pode apresentar pequenas mossas no rebordo.

5. BC – Bem Conservada

Moeda Circulada. Relevo da gravura com consideráveis sinais de desgaste, mas com todos os pormenores bem visíveis, incluindo a legenda e a data. Pode apresentar mossas ou riscos.

6. REG – Regular

Moeda com bastante desgaste no desenho, legendas e data, apresentando porém visíveis as características principais de classificação.

7. MC – Mal conservada

Moeda muitíssimo gasta, corroída, legendas e data obliteradas, o que as torna quase ilegíveis, sem que, contudo, haja dúvidas na sua identificação.

> **Para os que têm o programa CATÁLOGO da Spooler nº 9, terão que criar um subdirectório chamado NUMISTEC**

Neste campo deverá ser colocada a sigla correspondente ao estado de conservação da moeda.

Abundância: Este campo deverá ser preenchido só quando a raridade da moeda o justificar. Como sigla, recomendo a seguinte nomenclatura, tão conhecida na gíria do coleccionismo:

R	valor até 30.000$ – Escassa
RR	30.000$ a 80.000$ – Rara
RRR	80.000$00 a 150.000$ – Muito Rara
RRRR	mais de 150.000$ – Raríssima

Valor de Venda ao Coleccionador: o valor da moeda no circuito de coleccionismo actual, ou seja, o preço que a moeda teria se fosse vendida a um outro coleccionador. No fundo, o valor real da moeda actualmente.

Colocação: O local onde a moeda se encontra guardada.

INSTALAÇÃO

A sua instalação é semelhante à do FILATECA e do BIBLIOTECA, mas já que existem diferenças, eu vou repetir tudo:

1. Instalação em disco rígido, com o CATÁLOGO:

Para os que têm o programa CATÁLOGO da Spooler nº 9, terão que criar um subdirectório chamado NUMISTEC (tem mesmo que ter este nome) no vosso directório DBASE. Depois, copiem todos os 27 ficheiros para esse subdirectório. Como são utilizadores do CATALOGO, podem apagar os ficheiros "NU_SAIR2.PRG" e "DISQUETE.BAT", para pouparem espaço e evitarem erros. Finalmente, acrescentem o novo programa ao CATÁLOGO. Para o efeito, coloquem-se no directório DBASE, chamem o dBASE, e quando aparecer o CATÁLOGO, escolham a opção «Aumentar programas» do programa CATSET; depois é só preencher o registo com o número do novo programa (que é o número do registo), com o nome (que é NUMISTEC) e a data (que é a data em que for feita a instalação). Regressem ao CATÁLOGO premindo «0», o que lhes facultará a escolha do programa que vão ver – claro que será NUMISTEC!

2. Instalação em disco rígido, sem o CATÁLOGO.

Para os que têm disco rígido, mas não têm o CATÁLOGO, têm duas opções: ou pedem pelo correio a **Spooler Nº 9**, ou então fazem o seguinte:

Criem um subdirectório chamado NUMISTEC (tem mesmo que ter este nome) no vosso directório DBASE. Depois, copiem todos os 27 ficheiros para esse subdirectório. Seguidamente, depois de se terem colocado no novo subdirectório, terão que executar o batch DISQUETE.BAT, para que o subprograma de saída do NUMISTECA passe a ser outro, que estará preparado para voltar ao dBASE III Plus, em vez de ao CATÁLOGO. Para tal, basta digitar DISQUETE depois do «prompt». O nome do ficheiro NU_SAIR.PRG passará a ser NU_SAIR.BAK e o do ficheiro NU_SAIR2.PRG passará a ser NU_SAIR.PRG. Só assim é que o programa funcionará correctamente sem o CATÁLOGO.

3. Instalação em disquete

Seguindo o que foi dito na **Spooler Nº 10**, vou partir do princípio que são utilizadas disquetes de 3 1/2" DS DD (713Kb):

a) Copiar todos os ficheiros pertencentes ao dBASE

para uma disquete limpa.

b) Instalar na mesma (com a ajuda de qualquer editor ASCII) o ficheiro CONFIG.DB, cuja última linha deverá ser:

command = do numistec

c) Instalar nessa disquete os 27 ficheiros do programa NUMISTEC.

d) Depois, terão que executar o batch DISQUETE.BAT, para que o subprograma de saída do NUMISTEC passe a ser outro, que estará preparado para voltar ao dBASE III Plus, em vez de ao CATÁLOGO. Para tal, basta digitar DISQUETE depois do «prompt». O nome do ficheiro NU_SAIR.PRG passará a ser NU_SAIR.BAK e o do ficheiro NU_SAIR2.PRG passará a ser NU_SAIR.PRG. Só assim é que o programa funcionará correctamente sem o CATÁLOGO.

e) Finalmente, digitem DBASE. O programa NUMISTEC arrancará automaticamente.

É provável que os utilizadores de disquete vejam, na entrada e na saída do programa, de passagem, a mensagem *Invalid directory*. Não se importem, que essa mensagem não tem qualquer importância. É perfeitamente normal e não prejudica, em nada, o funcionamento do programa. Aliás, ela aparece apenas por umas fracções de segundo.

Se seguirem estas instruções à risca, tudo funcionará perfeitamente.

UTILIZAÇÃO

Voltei a utilizar o sistema já adoptado no BIBLIOTECA, em relação ao ecrã do menu principal e ao submenu da opção 2: este está contido num ficheiro .COM, que, ao ser chamado pelo dBASE, o «desenha». Assim ganha-se bastante em velocidade de execução, além de que este sistema é bastante mais prático.

Neste Menu principal, basta digitar o número da opção que se deseja utilizar, premindo <ENTER> a seguir, no caso deste ser menor que 10 (1 só dígito). Agora explicarei a utilização de cada uma das opções do Menu:

1 - Entrada de Dados

Nesta opção, basta ir inserindo os dados nos campos, um a um. Podem-se utilizar os cursores para se movimentar ao longo dos campos. Quando os dados tiverem sido terminados de inserir, basta premir <T> para terminar a inserção de registos (e voltar ao Menu), ou então qualquer outra tecla para inserir mais registos.

2 - Listar todo o ficheiro

Esta opção leva a um 1º submenu, no qual se escolhem os campos que irão ser listados. Neste, o(s) campo(s) que se encontra(m) em letras maiúsculas é(são) o(s) campo(s) pelo(s) qual(is) a listagem irá ser ordenada.

Se forem escolhidas as opções de 1 a 8 (inclusivé), a listagem será apresentada no ecrã, e, quando esta tiver acabado, será perguntado ao utlizador se deseja imprimir essa listagem na impressora. Se a resposta for <S> ou <s>, proceder-se-á à impressão. Caso contrário, voltar-se-á ao Menu. Antes da impressão, ter-se-ão que introduzir 2 dados: o tipo de papel usado na impressora (tipo formulário contínuo ou folhas normais, soltas) e o título que se deseja que seja impresso no topo de cada página. No caso de não se querer nenhum título, basta introduzir <ENTER>. Atenção, que só se pode introduzir um título até 70 caracteres. Os caracteres excedentes serão ignorados. Se for escolhida a opção 8, todos os dados de todos os campos serão enviados para a impressora. Mas, primeiro, também terão que ser introduzidos alguns dados. Inicialmente, aparece um 2º Menu (ou 3º, se contarmos com o principal) que nos perguntará qual o campo pelo qual deverão os registos ser ordenados ao serem impressos. Seguidamente, faz-se a mesma pergunta em relação ao tipo de papel utilizado e o título (vide parágrafo anterior).

3 a 7 - Procurar pelo...

Nestas opções, tal como no FILATECA e BIBLIOTECA, não é necessário introduzir todas as letras (ou números) do campo, porque o programa mostrará todas as fichas cujo campo respectivo comece por essas letras (ou números). Depois de introduzida a cadeia da busca, serão apresentadas todas as fichas coincidentes, uma a uma, bastando premir qualquer tecla para ver a seguinte.

8 - Emendar registos

Esta opção funciona basicamente como a opção nº 1, com a diferença de que tem que se inserir o nº do registo a emendar, o qual pode ser obtido com qualquer uma das opções 2 a 7. Há ainda outra diferença: quando se acabarem de introduzir os dados, a tecla a premir para emendar mais registos é a tecla <P>, e para voltar ao Menu basta premir qualquer outra tecla.

9 - Apagar todos os registos

Nesta opção, por motivos de segurança, a única letra que faz com que o ficheiro seja «limpo», quando aparece a pergunta «Tem a certeza absoluta que quer apagar todo o ficheiro?», é o <S> maiúsculo. Assim, no caso de o teclado não estar em CAPS LOCK, é necessário premir, junto com o <s>, a tecla <SHIFT> de maiúsculas. Deste modo, é mais difícil a activação acidental desta opção.

10 - Apagar um registo

Aqui, basta introduzir o número do registo que se quer apagar, o qual pode ser obtido com qualquer uma das opções 2 a 7. Seguidamente, aparecerá no écran o conteúdo desse mesmo registo para confirmação. A tecla <E> confirma, a <A> (ou qualquer outra), faz voltar ao Menu, sem que o registo seja apagado. Atenção: uma vez que o registo seja apagado com esta opção, nunca mais poderá ser recuperado.

11 - Sair deste programa

É apenas necessário responder se se deseja fazer um (novo) *backup* dos dados existentes no ficheiro. A tecla <S> confirma, procedendo-se então a nova pergunta: a *drive* para a qual se deverá fazer o *backup*. Depois disto, o programa fará a cópia automática de todos os dados.

Seguidamente, o programa voltará ao CATÁLOGO (se o tiverem) ou à linha de comandos do dBASE (na ausência do programa CATÁLOGO).

O programa vai já com um pequeno ficheiro de 4 moedas, para que possam experimentar todas as opções à vontade, de modo a aprenderem a usá-lo. Aliás, acho que só experimentando-o é que se poderá saber tudo sobre ele...

Espero que gostem do programa. De qualquer modo, se tiverem algum problema, contactem-me, ou através da BBSpooler, ou então pelo telefone 914 30 10 da Rede de Lisboa. Estou aberto a sugestões para mais programas, sejam eles de que estilo forem.

Afonso Infante

“O programa vai já com um pequeno ficheiro de 4 moedas, para que possam experimentar todas as opções à vontade, de modo a aprenderem a usá-lo”

Seagate



Nº 1 EM DISCOS DUROS

20 megabytes a 3,2 gigabytes

1 ano de garantia

DELEGAÇÃO NORTE
Edifício Fernão Magalhães
Av. Fernão Magalhães, n.°3548 - Loja 5
4300 PORTO
Telef.: 02-4105344

DELEGAÇÃO SUL
Rua António Feijó, 46-A
1500 LISBOA
Telef.: 01-7740301/392/395
Fax: 01:7782747

DISTRIBUIDOR AUTORIZADO

PARA PORTUGAL

cibernópolis
COMPUTADORES E PROGRAMAÇÃO, LDA.

* SEAGATE, SEAGATE TECHNOLOGY and the SEAGATE logo are registered trademarks of SEAGATE TECHNOLOGY INC.

FERY SPOOLER

(UMA BASE DE DADOS PARA A SPOOLER MAGAZINE)

COLABORAÇÃO DO LEITOR

O nosso amigo Herberto Brazão é, decididamente, um «spoolarista» de gema. Claro que ficámos sensibilizados (somos de carne e osso, como toda a gente) com o programa que nos enviou para apreciação. Mas o que verdadeiramente pesou na decisão de o publicar foi o facto de constituir mais uma demonstração da arte de programar em dBASE e da «transição» para o Clipper. O autor enviou-nos o executável, mas a sua dimensão era de tal ordem que pouco espaço sobrava para outro material. E então pensámos: «Porque não aproveitar a deixa e «obrigar» os leitores a aprender a compilar o programa? – Pois o Herberto Brazão até manda um «batch» com todas as instruções para a compilação?» Dito e feito. Na disqueta está tudo o que é preciso. Ouçamos o autor:



INTRODUÇÃO

Esta ideia surgiu quando comecei a minha leitura assídua à Spooler, e quando necessitei de utilizar os parâmetros para gravar os programas de GWBASIC em formato ASCII. Sabia que tinha já lido sobre esse assunto na *Spooler*. Mas onde? No Correio Dos Leitores? Na colaboração do leitor? Num artigo sobre esse assunto? E ainda pior qual foi a Spooler 7, 8, 9, 10, 11, 12.. ??? O certo é que levei cerca de 45 minutos para descobrir esse artigo novamente que fazia parte do Correio dos leitores. Então pensei: Daqui a um ano, e como vou ser assinante da Spooler e se me acontecer situação idêntica, ou seja, necessitar de consultar um artigo sobre determinado assunto publicado na Spooler? A situação de ler novamente todas as *Spoolers* para encontrar esse artigo seria dificultada pela quantidade de Spooler que tivesse. Em vez dos 45 minutos podia levar dias... Daí a ideia da FERY SPOOLER, baseado nas diferentes e mais importantes secções da *Spooler*.

Penso que a maioria dos leitores e assinantes da *Spooler* se pensarem da mesma maneira, não irão deixar de lado a FERY.

INSTALAÇÃO

Para instalar a FERY deve criar uma directoria chamada FERY e fazer a descompactação directamente para essa directoria. No caso de executar a descompactação para a própria disqueta execute o *batch* INSTALA.BAT, que criará em C o directório FERY e copiará para o mesmo o programa e os ficheiros FERY.EXE, *.DBF e *.FRM.

***Nota da Direcção**: Para quem não teve a paciência de ler a introdução a este artigo, lembramos que os ficheiros .PRG deverão ser previamente compilados com o Clipper Summer 87 ou Clipper 5.0/01. De notar, ainda, que os .PRG não correm sob o dBASE, pois contêm instruções próprias do Clipper.*

O QUE FAZ A FERY SPOOLER?

A **FERY SPOOLER** faz a gestão de 5 Ficheiros:

1 - NOTÍCIAS: neste ficheiro pode-se introduzir as notícias mais importantes publicadas na *Spooler* e que, mais tarde, possam a vir ser úteis em algum trabalho que venhamos fazer.

2 - DISQUETA: penso que é o ficheiro mais importante da FERY visto ser nele que vamos colocar os programas que fazem parte das disquetas que acompanham as *Spoolers*, a respectiva linguagem, para além de outros dados identificativos.

3 - ARTIGOS: serve para resumidamente, introduzir os artigos principais publicados nas *Spoolers*.

4 - CORREIO: onde podemos colocar as questões e as respostas de uma forma resumida feitas na secção do *Correio dos Leitores* de maior importância, sobre determinado assunto.

5 - MONTRA: através deste ficheiro podemos fazer um pedido sobre os produtos apresentados na *Montra Spooler*, pelo que após o preenchimento da encomenda basta carregar em *PrintScreen* para imprimir a nota de encomenda.

CONSIDERAÇÕES

– O ficheiro executável é o FERY.EXE. Para além deste são necessários todos os ficheiros com as extensões .DBF (bases de dados) e .FRM (ficheiros de listagem). Os ficheiros com extensão .PRG são as fontes do programa;

– Mesmo que não existam os ficheiros de indexação *.NTX, estes serão criados pelo próprio programa;

– Para escolha das opções utiliza-se as teclas de direcção premindo ENTER após a escolha pretendida;

– Para regressar ao MENU anterior, situação anterior ou anular operação utiliza-se a tecla ESC;

– No caso de efectuar alterações significativas (apagar,alterar) execute a Reindexação na opção UTILITARIOS;

– Nos ficheiros NOTÍCIAS, DISQUETA, ARTIGOS e

CORREIO, os campos Número, Spooler e Título são obrigatórios, caso contrário, as fichas não ficarão inseridas.

NOTÍCIAS

Este ficheiro serve para guardarmos as notícias mais importantes que vão sendo publicadas na Spooler. Tem as seguintes opções:

INSERIR: após a escolha desta opção surgem os campos referentes à ficha para serem preenchidos. Devem inserir obrigatoriamente os campos **Número Spooler** e **Título.**

Número Spooler – Número da *Spooler* a que se refere a notícia
Página – Número da página onde se encontra a notícia
Título – Título da notícia
Resumo – Escrever um resumo sobre a respectiva notícia

CONSULTAR: Permite fazer uma consulta através do título das notícias. Os títulos encontram-se ordenados alfabeticamente.
LISTAR: Lista para o ecrã as notícias ordenadas por Número de Spooler. Após premir ESC para sair desta opção, pode enviar esta listagem para a impressora premindo a tecla **I.**
ALTERAR: Alterar através do título uma ficha de notícia.
APAGAR: Eliminar ficha referente a uma notícia, digitando o título respectivo.

Nota: Para regressar à situação anterior, menu anterior ou anular determinada operação, carregue em ESC.

DISQUETA

Neste ficheiro introduz-se os dados identificativos dos programas que são publicados na secção «Colaboração do Leitor» e que fazem parte da disqueta que acompanha a *Spooler*.

INSERIR:

Número Spooler – Número da Spooler a que se refere o programa.
Página – Página onde se encontra o artigo alusivo ao programa.
Tipo Programa – Tipo do programa apresentado (Jogo, Didáctico, etc).
Título – Título ou nome do programa
Linguagem – Qual a linguagem que foi criado (Basic,Pascal,C,etc).
Resumo – Resumo ao programa baseado no texto alusivo ao mesmo.
Autor – Autor do programa.

CONSULTAR: Permite fazer uma consulta através do título dos programas. Os títulos encontram-se ordenados alfabéticamente.
LISTAR: Lista para o ecrã os Programas ordenadas por Número de Spooler. Após premir ESC para sair desta opção, pode enviar esta listagem para a impressora premindo a tecla **I.**
ALTERAR: Alterar através do título uma ficha de programa.
APAGAR: Eliminar ficha referente a um programa, digitando o título respectivo.

Nota: Para regressar à situação anterior, menu anterior ou anular determinada operação, carregue em ESC.

ARTIGOS

Neste ficheiro colocam-se as informações sobre os artigos principais e mais importantes publicados na *Spooler*, relacionados com as diversas áreas da informática: *Hardware*, *Software*, etc

INSERIR:

Número Spooler – Número da Spooler a que se refere o artigo.
Página – Página onde se encontra o artigo
Assunto – Que tipo de artigo.
Título – Título do respectivo artigo.
Resumo – Resumo sobre o artigo publicado.
Autor – Autor do artigo.

CONSULTAR: Permite fazer uma consulta através do título dos artigos . Os títulos encontram-se ordenados alfabéticamente.
LISTAR: Lista para o ecrã os artigos ordenados por Número de Spooler. Após premir ESC para sair desta opção, pode enviar esta listagem para a impressora premindo a tecla **I.**
ALTERAR: Alterar através do título uma ficha de um artigo.
APAGAR: Eliminar ficha referente a um artigo, digitando o título respectivo.

Nota: Para regressar à situação anterior, menu anterior ou anular determinada operação, carregue em ESC.

CORREIO

Este ficheiro é destinado aos assuntos tratados na secção do *Correio dos Leitores*.

INSERIR:

Número Spooler – Número da *Spooler*.
Página – Página da respectiva *Spooler*.
Título – Título sobre o assunto tratado.
Resumo – Resumo sobre esse mesmo assunto.

CONSULTAR: Permite fazer uma consulta através do título do assunto. Os títulos encontram-se ordenados alfabeticamente.
LISTAR: Lista para o ecrã o correio ordenado por Número de Spooler. Após premir ESC para sair desta opção, pode enviar esta listagem para a impressora primindo a tecla **I.**
ALTERAR: Alterar através do título uma ficha do correio.
APAGAR: Eliminar ficha referente a uma carta, digitando o título respectivo.

Nota: Para regressar à situação anterior, menu anterior ou anular determinada operação, carregue em ESC.

MONTRA

Neste ficheiro podem fazer-se os nossos pedidos dos produtos apresentados na *Montra Spooler*; o desenho do pedido é idêntico ao apresentado nas *Spoolers*. Após o

preenchimento de um pedido prime-se *Print Screen* para imprimir o respectivo pedido na impressora.

Nota importante: Para que a identificação de quem faz o pedido surja na respectiva nota de pedido, é necessário preencher a ficha de identificação que pode ser alterada na opção UTILITARIOS do programa (ver FICHA IDENTIFICAÇÃO).

INSERIR:

Número Spooler – Número da *Spooler* a que se refere o pedido.
Quant – Quantidades de cada produto a encomendar.
Referência – Referência do respectivo produto a encomendar.
Designação – Designação do respectivo produto.
Preço – Preço Total do(s) Produto(s).

O **TOTAL** final apresentar-se-á após o preenchimento do pedido, ficando em pausa para que possamos premir o *Print Screen* para enviar a nota de pedido para a impressora.

CONSULTA: Permite consultar as encomendas introduzindo o número de pedido. Por defeito apresenta sempre o último pedido no caso de premirmos ENTER quando solicita o número de Pedido. Nesta situação de consulta pode-se imprimir o pedido carregando em *Print Screen*.

ALTERAR: Introduzindo o número de pedido podem fazer-se as alterações necessárias ao respectivo pedido e imprimindo-o de igual forma.

APAGAR: permite apagar do ficheiro uma encomenda. Note-se que no caso de eliminar um pedido (encomenda) haverá uma renumeração dos mesmos.

FICHA DE IDENTIFICAÇÃO

Neste ficheiro encontram-se os elementos identificativos da pessoa que tem este programa, para além de serem necessários quando da emissão de um pedido. Para alterar a ficha de identificação escolha:

UTILITARIOS e Alterar Ficha de identificação

PREENCHER:

CLIENTE – Número de cliente caso já possua da *Montra Spooler*.
ASSINANTE – Número de assinante da *Spooler Magazine* .
NOME – Nome completo.
MORADA – Morada.
LOCALIDADE – Localidade.
C. POSTAL – Código Postal.
TELEFONES – Números de telefones.

Resta acrescentar que este programa foi desenvolvido em Clipper e que está a funcionar no meu computador já com os elementos mais importantes das últimas *Spoolers* publicadas.

Um *spool* amigo

Herberto Silva

• JOGOS • CIÊNCIA • CONTABILIDADE • IMAGENS • DESENHO •

EM BRAGA

UM «MUNDO» PARA TODOS QUE LIDAM COM COMPUTADORES

A MAIOR CASA DE SOFTWARE

FREEWARE - SHAREWARE - PUBLIC DOMAIN

EM PORTUGAL

COM UM SERVIÇO INÉDITO DE DISQUETE CATÁLOGO

ESTAMOS CONVICTOS QUE NA NOSSA VARIADÍSSIMA OFERTA ENCONTRARÁ CERTAMENTE OS PROGRAMAS QUE HÁ MUITO PRETENDIA

A nossa oferta engloba mais de 20.000 programas constantemente actualizados

NÃO PROCURE MAIS

PEÇA JÁ A DISQUETE CATÁLOGO! 250$00

Preços por Disquete 550$00 600$00

Fax: 053/76159
Telef.: 053/619244

UNIX • OS/2 • DIVULGASOFT, Lda. – Rua dos Barbosas, nº 151 – Loja 22

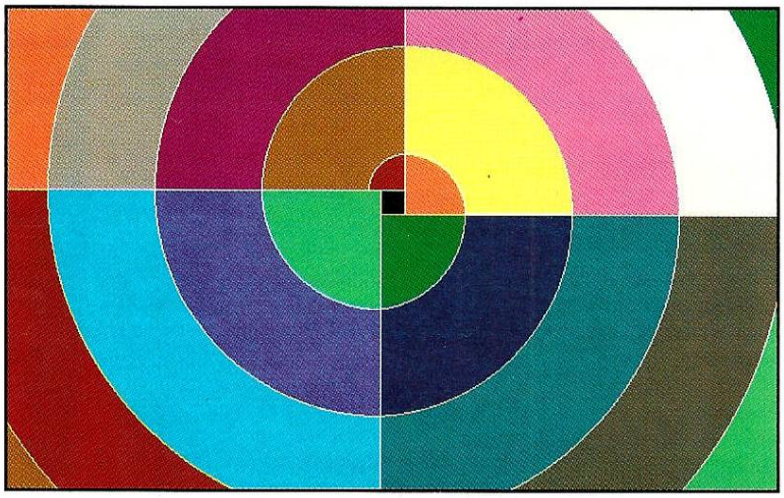
Utilitários de Redes • Processadores de Texto • Linguagens de Programação

• EDUCAÇÃO • JOGOS P/ ADULTOS • MÚSICA • WINDOWS • UTILITÁRIOS •

COLABORAÇÃO DO LEITOR

SPOOLPACK

Mais uma demonstração das capacidades do BASIC, para quem se dá ao trabalho de estudar a fundo esta linguagem aparentemente «simplória». O Ricardo e o Luciano mostram-nos como se pode realizar um interessante grafismo sem recurso às linguagens de «elite»...



Pretendemos assim demonstrar que mesmo um pequeno programa feito numa linguagem simples pode criar efeitos visuais muito interessantes

SPOOLPACK

Nós somos dois estudantes do 9º ano e temos 14 anos; por isso não nos é possível dedicarmo-nos por inteiro à programação. Mesmo assim, nos tempos livres fizemos o SPOOLPACK, que é um lote de programas óptimos para quem gosta de boas demonstrações gráficas (claro que esse aspecto fica ao critério do leitor). O SPOOLPACK é constituido por 4 programas:

– Duas demostrações gráficas que suportam placas VGA:

TUNEL VGA 1.0
ESPIRAL VGA 1.0

– Uma demostração gráfica que suporta placas VGA e EGA:

CIRCULOS VGA 1.0

– Uma demostração gráfica que suporta placas CGA, EGA e VGA:

SIMETRIA 1.0

É ainda de salientar que o SIMETRIA executa uma agradável música enquanto faz a demostração gráfica. Nós apresentamos uma curta explicação sobre cada programa, mas como pretendemos com este lote de programas «ensinar» algo sobre esta linguagem simples, mas com muitas capacidades, vamos fazer uma explicação mais promenorizada do programa CIRCULOS.

O programa CIRCULOS suporta placas gráficas EGA, VGA e compatíveis.

Este é um programa muito pequeno e simples e baseia-se principalmente em dois comandos:

CIRCLE – este comando permite desenhar círculos, elipses e ângulos dentro de um modo gráfico e tem a seguinte forma:

CIRCLE (x,y),raio,cor,partida,chegada,aspecto

Os parâmetros partida e chegada são ângulos radianos compreendidos entre -2*π e 2*π que especificam onde começa e acaba o desenho da elipse. Se os parâmetros partida ou chegada são negativos, a elipse é conectada para o ponto do centro com uma linha e os ângulos são tratados como se fossem positivos.

O parâmetro aspecto descreve a razão entre o raio X e o raio Y (X:Y). Se a razão do aspecto é menor que 1, então o raio da elipse é dado em pixels X, se a razão for maior que 1 o raio é dado em pixels Y.

PALETTE – este comando altera uma ou mais cores existentes no ecrã.

Este comando tem a seguinte forma:

PALETTE atributo,cor.

Em TURBO BASIC, tal como em GWBASIC, a paleta tem um valor de 16 cores, e cada cor é especificada por um atributo. O comando PALETTE vai mudar todas as cores existentes no ecrã com um determinado atributo, para outra cor. É de notar que esta mudança é instantânea. Daí que o efeito de movimento aparente, que ocorre quando as elipses são acabadas de desenhar, consiste em nada mais nada menos que numa mudança repentina de cores.

As elipses são desenhadas a partir de valores aleatórios, mas você poderá divertir-se mudando os valores aleatórios para valores fixos. No programa que enviamos o primeiro conjunto de círculos e ângulos é formado por valores fixos. Este programa cria um número infinito de elipses, círculos e ângulos, que são animados pela sua mudança de cores.

Pretendemos assim demonstrar que mesmo um pequeno programa feito numa linguagem simples pode criar efeitos visuais muito interessantes.

A listagem do programa contém a explicação pormenorizada de como ele funciona.

O SIMETRIA, tal como o nome diz, executa um desenho simétrico para bastantes modos gráficos, e com as opções de escolher 1, 2 ou 4 centros; se queremos escolher 2 cores ou deixar que o programa se execute com as cores máximas do modo e podemos ainda escolher se queremos que o programa toque uma música ou não enquanto desenha as linhas. O efeito de simetria é dado pela simples execução de linhas desde o centro até aos limites do monitor provocando assim a simetria desejada.

Quanto aos outros 2 programas o leitor encontrará nos ficheiros não compilados a sua constituição.

Para poder exprimentar cada um dos programas execute o seu ficheiro executável, e assim poderá apreciar os programas atrás referidos.

Esperamos que gostem tanto de usar o nosso programa, quanto nós gostámos de o escrever:

Ricardo Pereira e Luciano Oliveira

CURSOS DE INFORMÁTICA

Vários Cursos a iniciar em **Outubro**, nas áreas de **CAD**, **DOS** (Sistema Operativo, Programação e Aplicações), **WINDOWS** (Ambiente e Aplicações) e **OS/2** (Sistema Operativo).

As inscrições e as aulas decorrerão nas instalações da **GALILEU** em Lisboa (Santo Amaro).

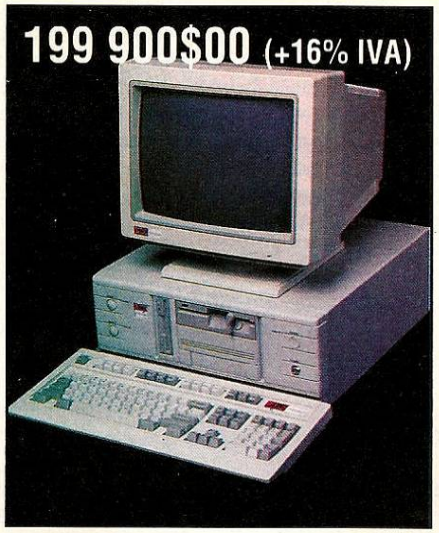
Condições especiais para Assinantes da Spooler.

Para mais informações sobre os programas dos cursos e dàtas de realização, contactar:

GALILEU

GALILEU – Serviços e Tecnologia, Lda.
(Divisão de Formação)
Rua Luís de Camões, 133 – 1300 LISBOA
Telef.: 362 34 64 Fax: 362 34 69

PROMOÇÃO • PROMOÇÃO • PROMOÇÃO • PROMOÇÃO



Computador ABC 386SX-25 Mhz

- Modelo Slim Line
- Processador 80386SX-25 Mhz
- 2 Mb memória RAM
- Disco Rígido de 44 Mb
- Drive de 3,5" – 1,44 Mb
- Placa VGA 512 Kb
- Teclado 102 teclas
- MS-DOS 5.0
- Monitor Super VGA 14" Policrimático – Res. 1024x768
- Mouse

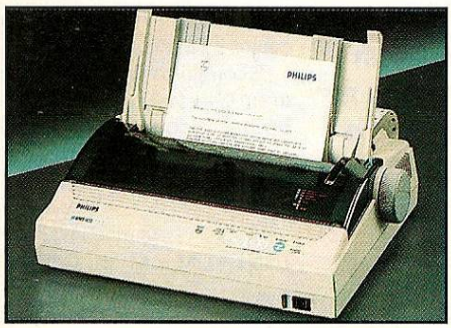
abc
A VERDADE EM SOLUÇÕES INFORMÁTICAS

Rua dos Sapateiros, 160 – 2º
1100 LISBOA
Telef.: (01) 342 44 48
347 80 94/347 78 96
Fax: 342 67 22

Impressora Philips NMS 1433 Plus

39 900$00 (+16% IVA)

- 80 colunas
- 9 agulhas
- 300 caracteres por segundo (modo draft)
- 48 caracteres por segundo (modo high quality)



AGENTES CASA VIOLA

LISBOA: Rua da Assunção, 67 – Telef.: 342 46 47
BRAGA: Av. Central, 85 – 1º – Telefs.: 72 798-74 369
S. JOÃO ESTORIL: Av. Florinda Leal, Lote 1-A – Telef.: 467 07 33
VISEU: Rua Direita, 79 – 1º – Telef.: 27 664
PORTIMÃO: Rua D. Carlos I Telefs.: 83 653-85 670
SETÚBAL: Largo da Misericórdia, 28 – Telef.: 31 432

COLABORAÇÃO DO LEITOR

HISTORY



> **O HISTORY é um programa de segurança que grava num ficheiro todos as instruções que forem executadas no DOS após a sua leitura**

Miguel Gonçalves é um nosso fiel leitor e activo utilizador da BBSPooler, **a nossa/vossa BBS. O programa que nos enviou foi considerado merecedor de publicação e a oportunidade verificou-se neste número da** Spooler. **Ninguém melhor que o próprio autor para nos dizer como instalar e utilizar o seu programa:**

O HISTORY é um programa de segurança que grava num ficheiro todos as instruções que forem executadas no DOS após a sua leitura, permitindo ainda impedir o acesso a *drives*, comandos ou programas. Tem busca de comandos já dados (*history*) que se podem usar através das teclas do cursor e *password* para sair.

É composto por 1 ficheiro (HISTORY.EXE) e foi escrito em C++ (v.1.0) com a preocupação de ser o mais rápido e pequeno possível, motivo pelo qual, no seu código fonte, alguns comandos podem parecer dispensáveis ou substituíveis por outros mais gerais.

INSTALAÇÃO

Se desejar instalar este programa no seu computador é aconselhável adaptá-lo às suas necessidades. Para tal, deve, a seguir ao seu nome, na linha de comandos, introduzir as opções que melhor lhe sirvam, na seguinte ordem:

– nome do ficheiro onde ficaram guardados os comandos dados;
– a *password* para sair do programa;
– a instrução, programa ou *drive* a que não se deverá ter acesso;
– a mensagem de erro a ser dada no caso desse acesso ser tentado, mensagem essa que deverá estar entre aspas (“”). Eis um exemplo:

HISTORY c:\<ficheiro> sair b: “Drive não acessível”

UTILIZAÇÃO

Para correr o HISTORY na linha de comando do DOS basta teclar o seu nome (em princípio HISTORY.EXE e, após os necessários parâmetros, premir ENTER). Para sair, a *password* (definida por si). No caso do programa ser corrido no interior de um ficheiro *batch* é obrigatório ele ser o último comando a ser dado.

Dentro dele será tudo igual ao DOS: existe uma *prompt* (inalterável), a seguir à qual se devem dar as instruções. Sempre que se carregue em ENTER, o programa verá se pode executar a instrução dada, abrirá o ficheiro, escreverá a instrução e fechará o ficheiro (que não deve ficar aberto). É conveniente ter um especial cuidado com o referido ficheiro já que, como se prevê, o espaço que ele ocupa aumenta com a utilização do programa, motivo pelo qual ele deve ser apagado sempre que a informação que contém já não interesse.

O HISTORY ocupa 25 Kbytes, pode ser posto na memória alta (com o LH do MS-DOS ou o *Hiload* do DR-DOS ou outro utilitário) e não cria conflitos com outras aplicações.

DICAS

Proponho que o HISTORY seja colocado na última linha do AUTOEXEC.BAT para «policiar» a utilização de um computador caso não seja o seu encarregado a utilizá-lo (que deverá escrever a *password*).

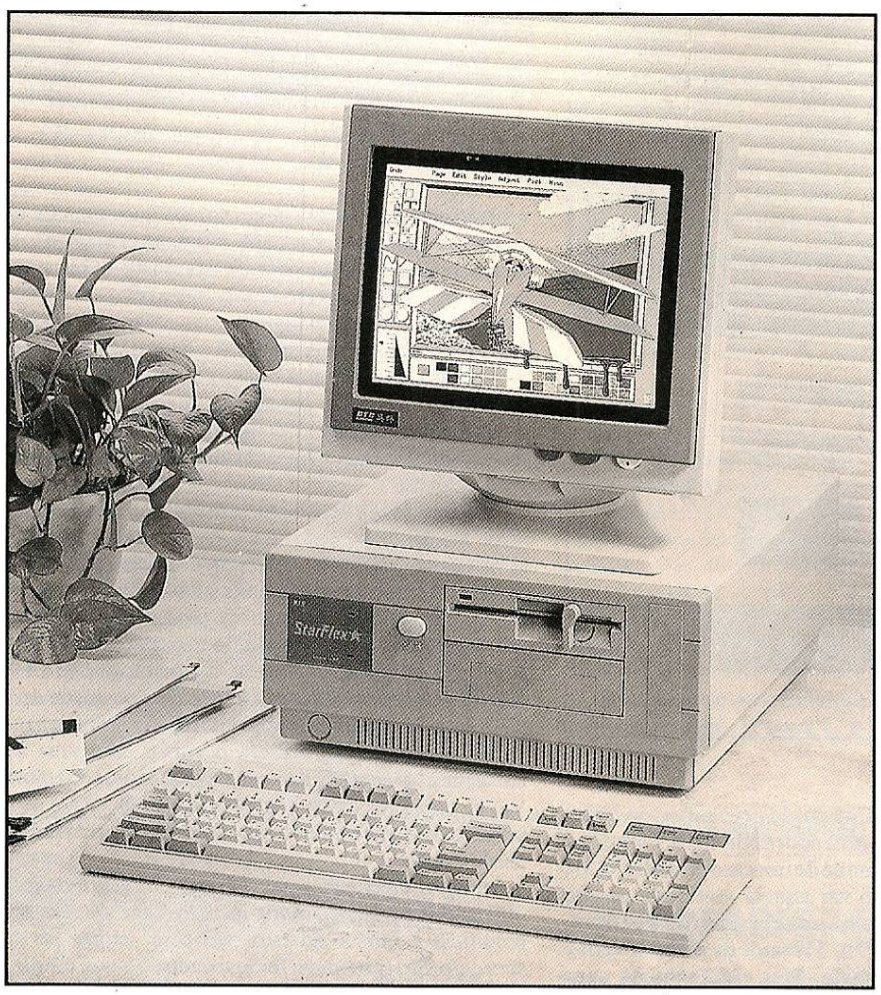
Aconselho o bloqueio do comando DEL ou do FORMAT, dando mensagens de erro iguais às que o do DOS daria, ou outras imaginadas pelo utilizador. Recomendo também a instalação de um *cache* que tornará o uso do HISTORY imperceptível.

Espero que este programa lhe seja útil (e não dê problemas!).

Miguel Gonçalves

StarFlex ★ 2000

ALR – Advanced Logic Research



intel inside™

BTC

BEHAVIOR TECH COMPUTER CORP.
USA/TAIWAN

AT286-16 DESKTOP

- 80286-16MHz
- 1 Mb RAM
- Drive 3,5" 1,44 Mb
- Disco rígido 40 Mb
- Carta VGA 256 Kb
- Teclado 102 teclas, português
- Monitor VGA mono 14"
- Rato BTC RS232

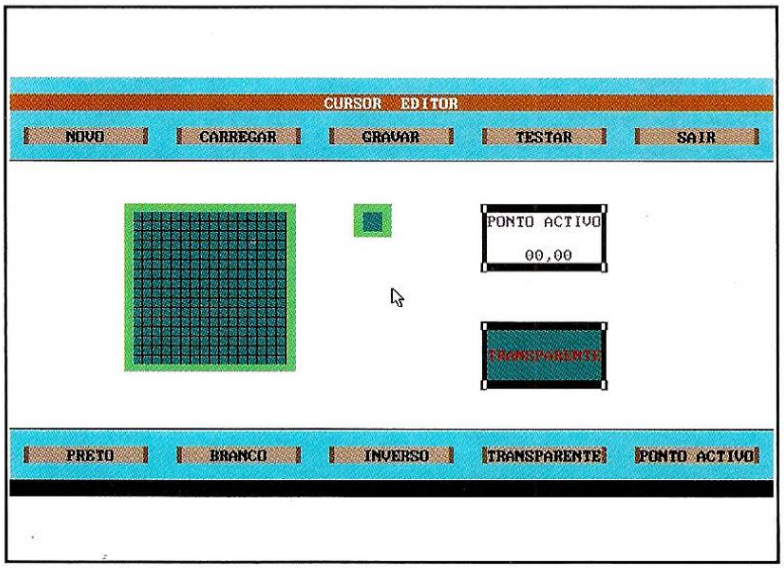
95 000$00 + IVA

POLIFIOS
INFORMÁTICA, LDA.

Rua Vitorino Nemésio, 2/3
Telefs.: 464 96 30 – 468 98 30
Telegramas: COBRE, Estoril
Telex: 15722 PAIXÃO P
Telefax: 467 37 15
2765 S. JOÃO DO ESTORIL

COLABORAÇÃO DO LEITOR

CURSOR EDITOR



Já alguma vez se fartaram de, ao usar o vosso rato, ver aquela seta – sempre a mesma – a apontar no vosso ecrã e tiveram vontade de pôr ali outra coisa qualquer que não fosse a dita seta?

O nosso leitor Armando Frazão preocupou-se com a «sanidade mental» dos utilizadores do rato. E muito a propósito pois, há dias, soube de um caso, aliás triste, de um senhor que, de tanto ver aquela «setinha» na sua frente, cada vez que via uma seta corria a fugir pensando que era um rato. Enfim. Tivemos de o recomendar a uma instituição adequada. Mas eis a solução para dramas semelhantes:

Já alguma vez se fartaram de, ao usar o vosso rato, ver aquela seta – sempre a mesma – a apontar no vosso ecrã e tiveram vontade de pôr ali outra coisa qualquer que não fosse a dita seta? Pois, quase como um seguimento às vossas lições de como domar um «roedor informático», este é um utilitário completamente dedicado à dita espécie e que permite substituir o habitual cursor gráfico do rato por outro qualquer por nós definido.

Divide-se em dois programas separados: um editor gráfico (CURSOR.EXE) e um utilitário residente (CURRES.COM).

Com o editor podemos criar um novo cursor gráfico, experimentá-lo, gravar em disco o trabalho feito, enfim todas as opções obrigatórias em qualquer editor. Esta parte do invento tem uns meses de idade a mais em relação à segunda parte e, como eu por defeito nunca ponho grandes comentários naquilo que faço, pode ser um passatempo moroso compreender como funcionam algumas partes do programa. Passado este tempo, eu próprio não consegui comentá-lo devidamente, pelo que peço as minhas desculpas. Surgiu a ideia de o desenvolver para testar (e aprender) aquilo que eu seria capaz de fazer com a linguagem *assembly*, com a manipulação do rato e com a manipulação directa da placa gráfica (sem utilizar funções BIOS). Este último aspecto serve para justificar o único modo gráfico em que o programa funciona (640x350x16). Como não sabia manipular placa nenhuma, tinha que aprender tudo de novo, e como as placas CGA, na minha opinião, não têm futuro, resolvi virar-me para a VGA. Visto VGA ou EGA ser quase a mesma coisa e uma placa VGA suportar os 640x350, em EGA sempre se garante que o programa funcione em mais meia duzia de máquinas.

A segunda parte do utilitário é um programa que se instala em memória e garante que apareça o nosso cursor no ecrã quando o rato é usado. Tem dois modos de funcionamento e uma linha de comando do tipo:

CURRES [/U] [/A] ficheiro de definição de cursor (.CED)

O cursor gráfico do rato pode se alterado de duas maneiras simples, ou com uma inicialização que coloca em uso o cursor por defeito (a seta), ou com a chamada à função 09h, ou seja, um pedido expresso para mudar o cursor. O programa intercepta a interrupção 33h e por defeito apenas actua quando existe uma chamada de inicialização (AX=0); na realidade ele faz a inicialização normalmente e depois altera o cursor com uma nova chamada à função 09h, mas se o utilizador usar o parâmetro '/A', o programa passa a actuar também quando é pedida uma modificação do cursor e quando é chamada a função 1 *Show Cursor'* .

Isto quer dizer que, se o programa que está a funcionar utilizar um qualquer outro método para apresentar ou modificar o cursor do rato, vai com certeza ser bem sucedido. Se o programa for invocado de novo, ele diz em que condições está instalado, faz uma actualização de acordo com o novo comando e substitui o cursor gráfico residente pelo fornecido na nova chamada. A opção '/U' permite libertar a memória usada pelo residente e voltar a uma utilização normal do *mouse*.

O programa residente funciona em qualquer modo gráfico; por isso, mesmo que não tenham EGA, podem desenhar em casa do vizinho e usar o trabalho em vossa casa, e deve ainda funcionar com qualquer *mouse* que seja compatível *Microsoft* ou *Mouse Systems*.

Agora, e como o que mais aprecio na **Spooler** é o caracter didático de que esta se reveste, aqui vão umas dicas para quem gosta de programar e que em parte me foram esclarecidas pela **Spooler**. Todo o código foi gerado para o Turbo Assembler usando directivas simplificadas.

Em relação à programação da placa gráfica consegui, por tentativas, descobrir uma maneira de escrever um *pixel* em qualquer parte do ecrã usando apenas dois registos da placa. Coloca-se no registo *Máscara de bits* (a máscara que indica os bits a escrever no ecrã), depois coloca-se no registo *Permissão de Activa/Desactiva* o inverso da cor que pretendemos escrever. Por exemplo, se queremos escrever em preto (00000000 em binário) colocamos em *Permissão de Activa/Desactiva* 00001111. Não se esqueçam de carregar os *Latches* se não escreverem um byte inteiro e os procedimentos de integridade referidos na **Spooler nº 12**, onde aliás podem encontrar muito mais sobre este assunto.

Para criar o CURRES.COM usei a opção '/t' do TLINK,

depois de na fonte indicar que o programa começa a execução no OFFSET 100h, com a directiva ORG 100h seguida de um *label* e terminado o código com a directiva END *label* .

O programa residente não intercepta só a interrupção 33h, mas também o *interrupt multiplex* 2fh. Pretende-se com isto poder detectar se existe uma cópia do nosso programa em memória mesmo que depois da primeira instalação os vectores tenham sido interceptados por outro qualquer programa. Evitam-se assim instalações múltiplas do mesmo residente e verifica-se a possibilidade de poder retirar ou não o programa de memória. Além disso, este facto permite em algumas situações que se possa alterar a configuração do residente sem contudo poder retirá-lo de memória, sendo falsa a recíproca. Se os vectores de interrupção ainda apontarem para a nossa rotina, então podem-os repor a normalidade alterando novamente os vectores de interrupção para os seus valores originais e depois libertar o segmento de memória onde se encontra o residente. Isto pode ser feito usando a função 49h do DOS:

```
mov ax,segmento de memória a libertar
mov es,ax
mov ah,49h
int 21h
```

Para alterar os vectores de interrupção pode-se fazer um acesso directo à tabela dos vectores que se encontra no princípio da memória (0000:0000). Para cada vector são usados quatro bytes: os dois primeiros para o*offset* e os dois últimos para o segmento de memória. Podemos assim ler/escrever o vector de uma determinada rotina de interrupção na posição de memória calculada através da expressão ((nº da interrupção) * 4) para o*offset* e ((nº da interrupção) * 4 +2) para o respectivo segmento. Mais fácil pode ser usar as funções 25h e 35h do DOS:

Para ler:

```
mov ah,35h
mov al,nº da interrupção
int 21h
```

é-nos devolvido em ES o segmento e em BX o *offset*.

Para alterar:

```
mov ax,segmento da nova rotina
mov ds,ax
mov dx,OFFSET da posição onde começa
                a nossa rotina de
                interrupção
mov ah,25h
mov al,nº da interrupção
int 21h
```

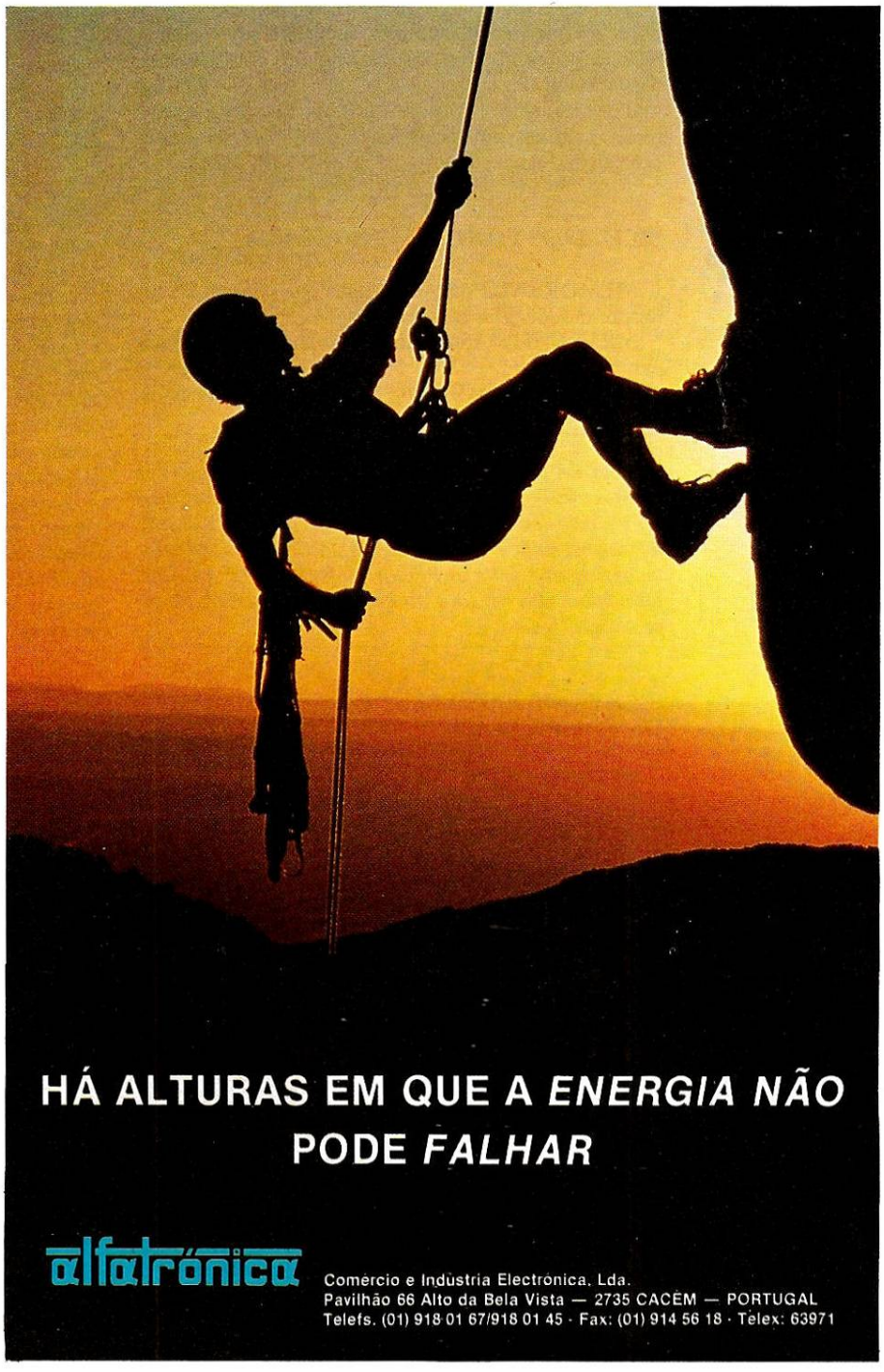
Só falta referir como deixar um programa residente. Pois bem, toda a gente aconselha a usar a função 31h do DOS por facilitar um valor de retorno para o DOS e por permitir reservar mais de 64k. Colocamos em DX o número de parágrafos (conjuntos de 16 bytes) de memória a reservar tendo para isso que efectuar uma divisão, e em AL o valor de retorno.

Por exemplo:

```
mov dx,OFFSET  primeiro byte a não precisar de
               ficar residente
mov cl,4
shr dx,cl      ;divisão por 16
inc dx         ;porque na divisão se perdeu o resto
mov ah,31h
mov al,valor de retorno
int 21h
```

E por agora é tudo, espero que gostem.

Armando Frazão



COLABORAÇÃO DO LEITOR

MEMORANDO

Eis um programa escrito em Pascal, especialmente orientado para os utilizadores que apreciam o «método e a organização» nas suas tarefas diárias. O título diz quase tudo. O restante fica a cargo do Rui Pestana.

Foi pensando nos mais distraídos que resolvi criar um pequeno utilitário que resolve todos esses problemas

INTRODUÇÃO

Nos tempos que correm, todos aqueles que passam horas a «fio» trabalhando com computadores correm o risco de se esquecerem de um ou outro compromisso, dando assim origem a situações que às vezes podem ser muito embaraçosas. Foi pensando nos mais distraídos que resolvi criar um pequeno utilitário que resolve todos esses problemas, ao qual dei o nome de MEMORANDO.

FICHEIROS FONTE DO PROGRAMA

MEMORAND.PAS – Programa que permite ao utilizador introduzir os seus memorandos;
MEM_UNIT.PAS – Unit de apoio ao programa principal;
MEM_READ.PAS – Programa que lê os memorandos.

---

*Nota da Direcção: Tal como vem acontecendo nos últimos números da **Spooler**, o espaço na disqueta está a ser «gerido» em termos de «economia caseira», isto é, de forma pragmática. Daí resultam situações como a presente: os ficheiros-fonte em Pascal não foram incluídos. Mas, tal como temos vindo a proceder, serão publicados na próxima disqueta.*

FICHEIROS RESULTANTES DA COMPILAÇÃO

MEMORAND.EXE
MEM_READ.EXE

A INSTALAÇÃO DO UTILITÁRIO

A instalação do utilitário MEMORANDO não apresenta pontos críticos, podendo o ficheiro MEMORAND.EXE, a sua respectiva *unit* MEM_UNIT.TPU e o MEM_READ.EXE serem colocados em qualquer directório da disqueta ou do disco rígido.

Em relação ao leitor das mensagens – o MEM_READ.EXE, há que ter em atenção que o seu nome deve ser referido no AUTOEXEC.BAT do sistema, isto para permitir que as mensagens possam ser lidas aquando do arranque.

Exemplo: No caso de ter os ficheiros acima referidos no directório C:\UTILS\MEM, então poderá colocar no seu AUTOEXEC.BAT o seguinte comando:

```
PATH=C:\UTILS\MEM; .... {Seguido de outros «caminhos» }
C:\UTILS\MEM\MEM_READ.EXE {Linha de chamada}
```

No entanto existe outra linha de chamada que também pode ser considerada em substituição da acima referida:

```
PATH=C:\UTILS\MEM; .... {Seguido de outros «caminhos»}
CD UTILS\MEM
MEM_READ.EXE
```

Nota: Aconselho que a linha de chamada acima referida seja colocada em último lugar no AUTOEXEC.BAT, depois de carregados os ficheiros de instalação do teclado, rato, etc ...

CARACTERÍSTICAS E FUNCIONAMENTO DO PROGRAMA

Este programa foi escrito em Turbo Pascal V6. Inicialmente, quando me propuz realizar este utilitário,era minha ideia fazer conter na ficha de memorandos as diversas horas que fazem parte de um dia normal de trabalho, acompanhadas pelo respectivo campo de escrita. No entanto e pela variedade de horas que poderão ser necessárias a cada utilizador, entendi que o mesmo se tornaria mais funcional (e até mesmo poder comportar uma mensagem mais alongada) se a referida ficha de memorando não referisse qualquer hora, deixando assim ao critério e necessidade dos utilizadores a maneira de preenchimento que julgarem adaptar-se melhor às suas necessidades.

A referida ficha de memorandos é constituída por doze campos, podendo cada um deles comportar até trinta e oito caracteres. Isto é possível graças a um ciclo que é responsável pela introdução dos vários dados do utilizador nos campos de uma variável pré-definida, constituída por um *array*, tal como a seguir refiro:

```
TYPE MEMO=ARRAY[1..12] OF STRING[38];

{Criação da variável MEMO constituída por um array com 12
células, sendo cada célula constituída por uma string de 38
caracteres}

VAR
   STR1:STRING; {Define STR1 como sendo uma variável tipo STRING}
   D:^WORD;     {Define 1 variável pointer do tipo WORD }
   FIC:MEMO;    {Define a variável FIC, ficando esta com as características
                 da variável MEMO}
...

   NEW(D);      {Inicializa a variável pointer}
   D^:=1;       {Atribuição do valor 1 à variável}
   REPEAT;      {Início do ciclo}

... {Introdução dos dados recorrendo à variável STR1}

   FIC[D^]:=STR1; {Iguala a célula cujo nº corresponde ao valor
                   D^ à variável STR1}
   D^:=D^+1;      {Junta 1 unidade ao valor de D^}
   UNTIL D^=13;   {Repete o ciclo até que D^ tenha o valor 13}
```

Introduzidos todos os dados nas respectivas células, o programa vai proceder à transferência dos dados para a disqueta ou disco, certificando-se primeiro se já existe algum ficheiro criado com a mesma data. Assim:

```
VAR  NOME,NOME1:STRING;
     DIA,MES:INTEGER;
     FICH:FILE OF MENO          {MEMO já foi referido no esquema anterior}
...

STR(DIA,NOME) ;                 {Transfere o conteúdo da variável integer DIA para uma variável string NOME}
STR(MES,NOME1);                 {Transfere o conteúdo da variável integer MES para uma variável string NOME1}
NOME:=CONCAT(NOME,'-',NOME1);   {Junta as duas variáveis numa só, ficando estas separadas por um hífen}
ASSIGN(FICH,(NOME+'.MEM');      {Associa o nome do arquivo referido em (NOME+ '.MEM') com a variável FICH}
{$I-};          {Ordem para identificar erros, caso eles existam}
RESET(FICH);    {Ordem para abrir ficheiro já existente}
{$I+};          {No caso do IORESULT ser igual a zero então isto significa que já existe um ficheiro com a mesma data, sendo o
                 utilizador alertado para o facto. Se IORESULT=2 então o ficheiro não existe, ficando assim o programa a saber que
                 não há nenhum ficheiro com a mesma data}

{Se o ficheiro não existe, ou o utilizador dá ordem para para gravar o ficheiro, então:}

ASSIGN(FICH,(NOME+'.MEM');
REWRITE(FICH);          {Cria e abre um novo ficheiro}
WRITE(FICH,FIC);        {Saída de dados para o disco}
CLOSE(FICH)             {Fecha o ficheiro}
```

Depois deste pequeno aparte, será altura de começar a referir a função de cada uma das opções deste programa a que chamei MEMORAND.EXE. Assim:

Criar um Memorando – Tal como o nome indica, esta opção permite ao utilizador criar um memorando, tendo para isso de proceder à indicação da data em que a mensagem deve ser apresentada no computador. Depois de introduzir a data, o utilizador pode registar as suas notas no conjunto de 12 campos que já foram acima referidos. Para passar de um campo para o outro, utilize a tecla ENTER. No caso da data indicada ser anterior à data do sistema, o programa alerta o utilizador, perguntando se quer ou não criar a ficha com a data com que está a pretender introduzí-la.

Ler um Memorando – Para ler uma ficha basta indicar a data para a qual a ficha de memorando foi criada.

Apagar Memorando – É possível apagar uma ficha, sendo para isso necessária a introdução da data com a qual a ficha foi criada.

Acertar Data do Sistema – Caso a data do seu computador não esteja correcta, esta opção dá-lhe a possibilidade de modificar a data do Sistema.

Basta para isso que introduza o dia, mês e ano corrente, podendo, de imediato, visualizar a respectiva alteração através da janela do menu principal.



DEPOIS DE GRAVADO UM MEMORANDO

Quando instalado no AUTOEXEC.BAT, o ficheiro MEM_READ.EXE vai ler a data do sistema e vai procurar um possível ficheiro que tenha como nome a respectiva data, sendo isto possivel com os comandos FSEARCH e FEXPAND que têm como objectivo «correr» todo o PATH do sistema à procura de um eventual ficheiro com a data do dia.

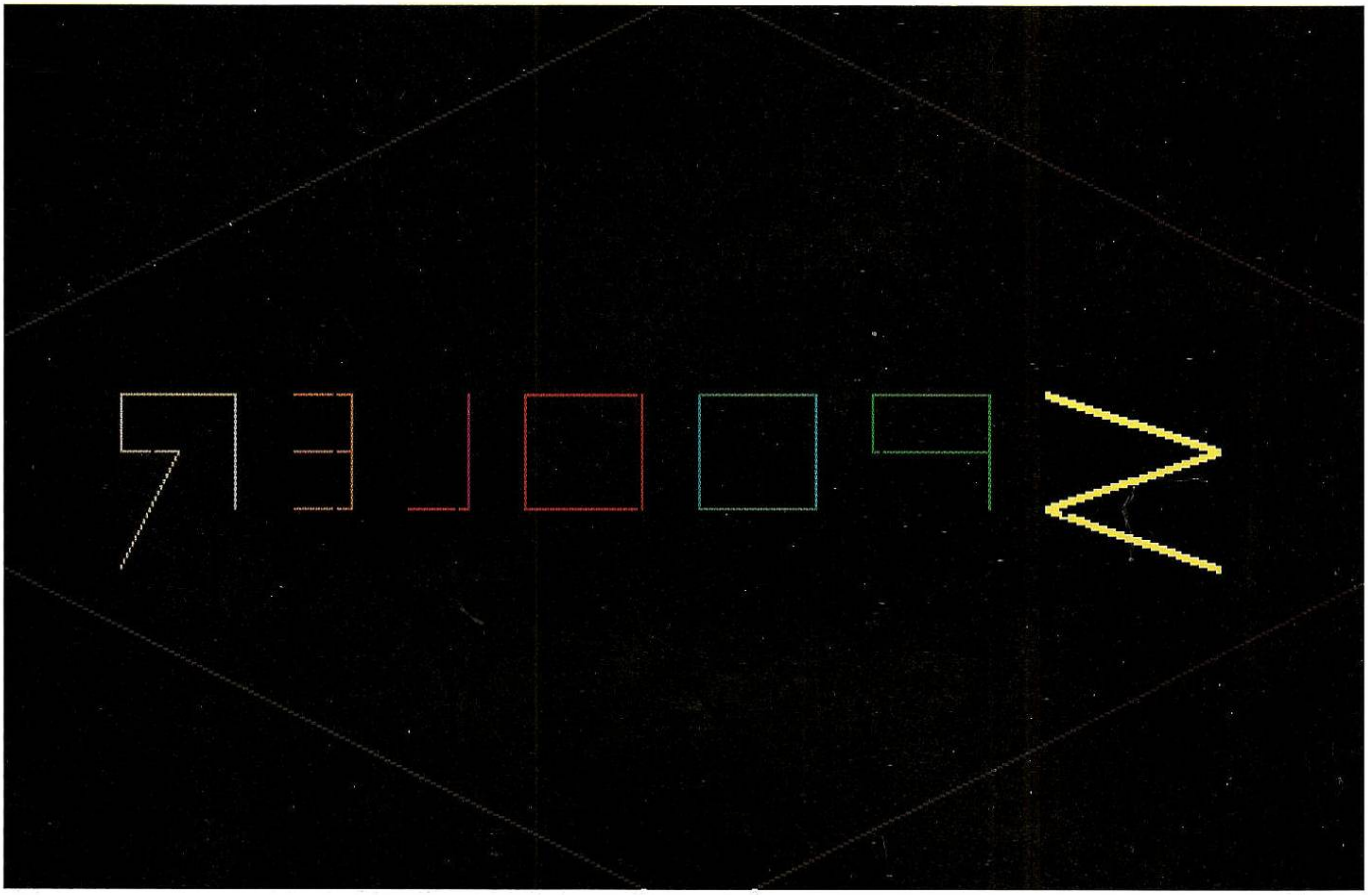
No caso de não ser encontrado nenhum ficheiro com data coincidente com a do sistema, então o programa de leitura de memorandos não apresentará qualquer mensagem, continuando assim o arranque do computador.

No caso do ficheiro ser encontrado, este é apresentado ao utilizador, juntamente com a mensagem «APAGAR MEMORANDO... (S/N)?». Se o utilizador premir **S** então a mensagem será apagada do disco. No caso de não se autorizar o apagamento da ficha, então, durante essa data, e sempre que for carregado o AUTOEXEC.BAT, a mensagem voltará a surgir no ecrã.

Rui Pestana

COLABORAÇÃO DO LEITOR

ANIMAÇÃO GRÁFICA



> Dentro da computação gráfica a animação despertou-me um interesse especial pelo que comecei a fazer alguma investigação

O Rui Anastácio jé é nosso conhecido pelo interesse e qualidade dos seus programas. Portanto, não precisa de apresentação. Oferece-nos, desta vez, uma curiosa demonstração das capacidades do Pascal na computação e animação gráfica.

Uma das áreas de maior desenvolvimento na informática nos últimos anos tem sido a Computação Gráfica. Os computadores têm sido utilizados para criar imagens de todos os tipos. Muitos filmes comerciais utilizam estes grafismos e a animação por computador, como por exemplo o filme «TRON» (filme que foi criado quase exclusivamente com o auxílio dos gráficos compturizados). Os gráficos de computador são utilizados em muitas áreas da nossa vida como, por exemplo, na contrução de esquemas e relatórios comerciais, na televisão, na arquitectura e engenharia etc.

Dentro da computação gráfica a animação despertou-me um interesse especial pelo que comecei a fazer alguma investigação. Para interessar os leitores nesta área, que pode ser muito criativa, resolvi apresentar dois pequenos programas que demonstram algumas transformações possíveis numa imagem tridimensional. Serão incluidas a **rotação**, **zoom**, **perspectiva** e **movimentação**.

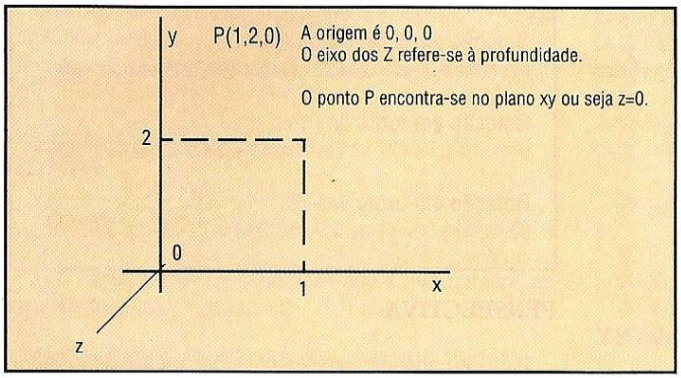
Em primeiro lugar vou fazer uma referência ao funcionamento dos programas, para que o leitor possa criar ou simplesmente animar alguns exemplos, passando depois à parte da programação. Os dois programas são:

CRIAFIG (Cria figura) – Este programa permite a criação de uma figura a 3 dimensões que é definida através de um ficheiro de texto.

ANIMAFIG (Anima figura) – Depois de criada, a figura poderá ser animada pelo utilizador utilizando o teclado.

FIGURAS A 3D

Para criar uma figura a 3D é necessário em primeiro lugar definir os pontos do espaço. Os pontos são definidos por as coordenadas x,y e z.



Depois vêm as linhas que vão ligar os pontos atrás definidos.

CRIAFIG.EXE

A partir de um ficheiro de texto, que contém a definição de pontos e linhas, o programa CRIAFIG cria dois ficheiros, um com extensão COO (ficheiro com as coordenadas dos pontos) e outro com extensão LIN (definição das linhas). O formato do ficheiro de texto deve ser o seguinte:

```
<inicio do ficheiro>
Comentários - esta linha não interessa
0       0       0       Origem
4       4       4
9       9       8
*
1       2       15      0
2       3       1       1
1       3       7       10
ultima linha também não interessa
<fim do ficheiro>
```

A primeira linha pode ser utilizada para indicar o nome da figura e quaisquer outros comentários porque será ignorada. A partir da segunda linha e até ser encontrada uma linha igual a '*' (linha nº 5 no exemplo) vem a definição dos pontos. O primeiro valor será o **x**, depois o **y** e o **z**. Os valores têm que começar nas posições **1,9** e **17** da linha. O resto da linha não interessa. No exemplo estão definidos o pontos (**0,0,0**), (**4,4,4**) e (**9,9,8**).

A linha que contém apenas um '*' tem que existir e separa a definição de pontos da definição de linhas. Depois da linha separadora vem a definição das linhas. Os primeiros dois valores indicam quais os pontos a serem ligados. O terceiro indica a cor (0 a 15) e a quarta linha indica o tipo de linha.

Tipo de linha: 0 - contínua
1,2,3 - tracejados
Para obter uma linha mais grossa soma-se 10.

No exemplo a figura é definida pelas linhas:

– **1(0,0,0) a 2(4,4,4)** cor branco intenso (15) linha contínua.

– **2(4,4,4) a 3(9,9,8)** cor azul (1) linha a tracejado.

– **1(0,0,0) a 3(9,9,8)** cor branco (7) a grosso

A última linha pode ser utilizada para comentários pois será ignorada.

Depois de criado este ficheiro de definições, deverá gravá-lo com a extensão .DEF e passá-lo como parâmetro ao programa CRIAFIG da seguinte forma:

CRIAFIG nome_do_ficheiro

Exemplo:
A partir da disquete execute o comando:

A:>CRIAFIG solido

Deverá então ver toda a definição da figura.

ANIMAFIG

Após a criação dos ficheiros .COO e .LIN será possivel então animar a figura. Para isso basta correr o programa ANIMAFIG seguido do nome da figura. Exemplo:

A:>ANIMAFIG solido

Os ficheiros .BGI deverão encontrar-se na directoria corrente para que



Para desenhar em perspectiva colocamos o observador no eixo dos ZZ negativo e consideramos o plano de imagem Z=0

o programa os encontre. Este programa corre na placas gráficas mais utilizadas como sejam a VGA e a EGA. Agora que tem a sua figura no ecrã utilize as seguintes teclas para a animar:

Teclado numérico:
8 – para cima
2 – para baixo
6 – para a direita
4 – para a esquerda
5 – para parar
7 – liga e desliga a rotação em torno do eixo dos YY
1 – igual para o eixo dos ZZ
2 – Igual para o eixo dos XX
/ – inverte o sentido de rotação
- – diminue a figura
+ – aumenta
*** – mantém o tamanho
0 – para de rodar

Teclado normal:
Z – dininue a velocidade de rotação
X – aumenta a velocidade de rotação
C – mantém num nível constante
A – liga/desliga o desenho
^A – apaga o ecrã
S – pausa
P – desenha em perspectiva
ESC – sai do programa

ACERCA DA PROGRAMAÇÃO

Os dois programas foram realizados em Turbo Pascal 6.0 mas deverão funcionar perfeitamente em versões anteriores.

CRIAFIG – Este programa não utiliza quaisquer rotinas gráficas; a sua única função é ler um ficheiro de texto com o formato atrás descrito e criar dois ficheiros que definem a figura. Para mais informação analize o CRIAFIG.PAS.

ANIMAFIG – Todos os procedimentos gráficos utilizados podem encontram-se em livros de grafismos em computador. O livro que eu utilizei foi o seguinte:

Grafismos em Computador
Autor: John Lansdown
Editorial Presença - Colecção Sistemas

Este é um bom livro para aqueles que estão interessados em aprender a criar gráficos no computador.

De seguida serão descritos os procedimentos necessários para efectuar a **rotação**, **perspectiva** e **zoom** de um ponto com coordenadas (Xant,Yant, Zant).

ROTAÇÃO

Para rodar um ponto é necessário o **seno** e **coseno** do ângulo. Vamos considerar o ângulo **n** e o ponto **(Xant,Yant, Zant)**. Para calcular o **seno** e o **coseno** utilizando as funções **sin** e **cos** do Pascal é necessário que o ângulo seja dado em **radianos** e não em **graus**. Assim, se **n** for dado em **graus**, para acharmos o **seno** e o **coseno** faz-se:

```
SenA=sin(n*0.017453292)
CosA=cos(n*0.017453292)
```

Multiplica-se por 0.017453292 para converter **graus** em **radianos**.

Rotação em torno dos XX:
(Xnovo,Ynovo,Znovo)=(Xant,Yant*cosA+Zant*senA,Zant*cosA-Yant*senA)

Rotação em torno dos YY:
(Xnovo,Ynovo,Znovo)=(Xant*CosA-Zant*SenA,Yant,Xant*SenA+Zant*CosA)

Rotação em torno dos ZZ:
(Xnovo,Ynovo,Znovo)=(Xant*CosA-Yant*SenA,Xant*SenA+Yant*CosA,Zant)

PERSPECTIVA

Para desenhar em perspectiva colocamos o observador no eixo dos **ZZ** negativo e consideramos o plano de imagem **Z=0**. Os cálculos necessários devem ser efectuados depois das transformações (rotação etc.) e antes do desenho. Considerando o observador em **(0,0,-D)** e o plano de imagem **Z=0** para calcular a perspectiva temos:

```
F=1/((Zant/D)+1);
(Xnovo,Ynovo,Znovo)=(Xant*F,Yant*F,0);
```

Quando desenhamos uma linha em perspectiva podem acontecer 3 situações:

- A linha encontra-se à nossa frente
  (ambas as extremidades têm Z>-D). Linha visível.
- A linha encontra-se atrás de nós
  (ambas as extremidades têm Z<-D). Linha invisível.
- Uma parte da linha é visível
  (uma extremidade tem Z<-D e outra Z>-D).
  Neste caso é necessário um corte na linha de modo a desenhar penas a parte visível.

Neste exemplo não considerei o corte da figura para não complicar demasiado. Por isso, se a imagem for muito aproximada, irão surgir problemas de desenho.

ZOOM

Para aumentar ou diminuir a figura basta multiplicar as coordenadas dos pontos por um valor qualquer:

```
(Xnovo,Ynovo,Znovo)=(Xant*val,Yant*val,Zant*val)
```

se **val>1** a figura aumenta. Se **val** está entre **]0,1[** a figura diminue.

O PROCEDIMENTO DESENHAR

Depois das transformações feitas, com excepção da perspectiva, este procedimentos irá desenhar a figura. Em primeiro lugar a linha é apagada, ou seja, desenhada com a cor do fundo. Em segundo lugar são feitos os cálculos da perspectiva caso esteja activa e os cálculos de posisionamento da imagem no ecrã. Para transformar a imagem de 3D para 2D basta ignorar a componete **Z**. Os resultados são armazenados na tabela coo2. Em terceiro lugar a linha transformada é desenhada. O resto do programa pode ser compreendido com um pouco de paciência e dedicação.

Espero que toda esta matemática não vos desanime e que tenham ficado motivados para a animação gráfica que é muito interessante.

Rui Miguel Dias Anastácio

TECLADO

COLABORAÇÃO DO LEITOR

Tal como prometido, aqui estão os outros dois utilitários referidos no anterior número da Spooler. São eles o TECLADO e o QCD. O Nelson Pires é a pessoa indicada para nos falar deles:



Um programa escrito para os ases do teclado, e que gostam de escrever a «abrir». Este programa permite configurar o controlador do teclado e, assim, regular a entrada de caracteres (o nº de caracteres por segundo) e o tempo antes da auto-repetição de caracteres. O programa abre uma janela no ecrã, (gravando o que está de baixo, para depois o repor no final do programa).

As teclas disponíveis são os cursores, e a tecla **ESC** que termina o programa, ajustando os valores para os escolhidos. As teclas **seta_cima** e **seta_baixo** deslocam o rectângulo que indica a opção presentemente activa (nº caracteres ou auto-repetição). As teclas **seta_esquerda** e **seta_direita** deslocam a barra luminosa que indica os valores actuais da opção activa, para a esquerda e a direita, respectivamente.

Caso disponha de um **rato**, pode utilizá-lo no TECLADO. Para ser franco, o programa original apenas era controlado desta forma, sendo esta a sua segunda versão. Utilizando o **rato**, para se escolher os valores basta premir botão esquerdo, quando o cursor se encontrar em cima dos sinais de mais e de menos. Para teminar e utilizando o **rato**, basta premir o botão esquerdo no pequeno quadrado que se encontra no canto superior esquerdo. A utilização simultanêa do **rato** e teclado para controlar o programa é possível (pelo menos, no meu computador, não houve problemas).

O *output* do programa é feito quase todo directamente para a memória de vídeo por questões de rapidez, pelo que não posso garantir qual será o resultado em todo o tipo de placas (nas que testei (Ega,Vga,Hercules) não encontrei problemas).

O TECLADO inclui, também, uma linha de teste, para ser possível observar a velocidade dos valores actuais (qualquer tecla de A a Z, espaços, backspace, etc). Para um teclado a «abrir» as barras luminosas devem estar ambas nos extremos esquerdos.

Junto segue também um pequeno ficheiro, o FAST.COM, que altera para os valores máximos permitidos. Este ficheiro é para colocar no AUTOEXEC.BAT, e é por isso que é tão pequeno.

QCD – QUICK CHANGE DIRECTORY

Uma versão do estafado CD (Change Directory), que permite através da utilização das teclas **S** e **N** deslocar-se para qualquer directoria pretendida.

Utiliza-se:

QCD <Directoria_procurar> [drive]

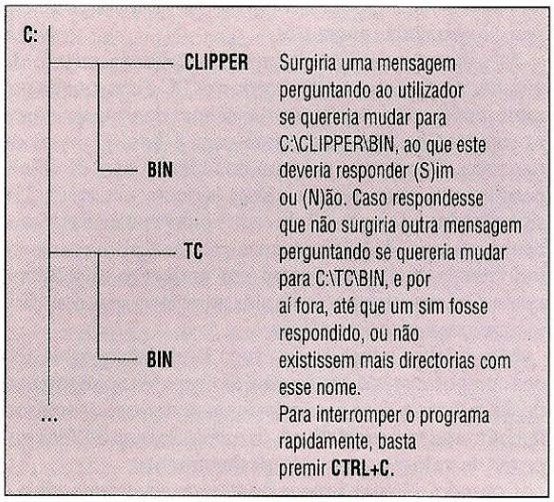
Se o *drive* não for especificado, então a procura é feita no *drive* actual.

<Directoria_procurar> indica o nome da directoria para que se quer mudar.

Os caracteres * e ? são permitidos, caso não sejam conhecidos todos os caracteres da directoria a procurar. Embora o programa seja um pouco lento a correr (tem que percorrer todas as directorias e subdirectorias do disco, até encontrar as desejadas), é bem mais rápido que estar com uma grande sequencia de CD's. Num exemplo:

A:\> QCD bin c:

vai ao *drive* C e procura por uma directoria de nome BIN. Supondo que neste *drive* existe uma estrutura semelhante à seguinte:



Se nenhum (**S**)im for respondido, ou nehuma directoria for encontrada com o nome especificado, então não haverá alteração da directoria corrente.

Todos os programas foram escritos em Turbo Pascal e compilados com a versão 6.0. Bom, por hoje é tudo, despeço-me com um abraço

Nelson Pires

> **“Este programa permite configurar o controlador do teclado”**

PROGRAMAR EM ASSEMBLY (PARTE III)

> **A função da memória nos PC é aproximadamente a mesma da memória nos humanos: guardar fielmente a informação**

No artigo anterior tivemos oportunidade de afirmar que o estudo da linguagem *Assembly* implica por si só a necessidade de ter um conhecimento algo (mais) pormenorizado de dois dos componentes deste sofisticado **sistema** que é o nosso computador pessoal compatível IBM PC.

Já abordámos um desses componentes, o **CPU** ou **microprocessador** e hoje iremos tratar de um outro, a **memória**.

Vamos introduzir ainda um outro tema de grande importância para a programação *Assembly* (e não só), os **interrupts** (ou **interrupções**) mas isso só após um ligeiro capítulo sobre *tabuada* **hexadecimal**.

1 – A MEMÓRIA

A função da memória nos PC é aproximadamente a mesma da memória nos humanos: guardar fielmente a informação. A memória dos PC porém, diga-se de passagem, é muito mais digna de confiança que a memória dos homens, pois é praticamente infalível. Ao contrário dos humanos que têm toda a sua memória no cérebro, os PC não localizam a totalidade (ou melhor, a quase totalidade) da sua memória no CPU mas sim em *chips* colocados, quer na placa principal (*motherboard*), quer em placas de expansão instaladas numa *slot*.

Já sabemos que para cumprir a delicada missão de guardar a informação, a memória dos PC é constituída por muitos milhares (ou milhões) de células de armazenamento (ou localizações) cada uma com 8 bits (1 byte) de tamanho. O valor de cada um dos bits do byte da célula permite que essa mesma célula assuma um entre 256 significados diferentes (2 elevado à oitava potência). Também cada uma dessas pequenas células é absolutamente individualizável, pois possui um **endereço absoluto** o qual se materializa por um valor numérico que identifica inequivocamente essa célula.

Mas o leitor talvez ainda esteja lembrado que os CPU dos computadores desta família têm uma *visão* segmentada da memória, isto é, não formam endereços absolutos, mas sim endereços relativos, os quais são constituídos por **pares de valores segmento\deslocamento.**

Contudo, a lógica interna de suporte ao sistema salva a situação e transforma directamente esses pares de valores em endereços absolutos quando for caso de aceder à memória. E, assim, a quantidade de memória *directamente endereçável* que pode ser instalada num sistema fica condicionada deste modo e, em última análise, ao número de Linhas de Sinal do Bus de Endereçamento.

Os «velhos» processadores 8088 e 8086 só têm 20 Linhas de Sinal e, por conseguinte, só podem endereçar 1 MB (2 elevado à vigésima potência), mas os processadores 80286 ou superiores podem endereçar pelo menos 16 MB e, por isso, os sistemas que deles dispõem já vêm hoje em dia quase sempre equipados à partida e no acto da compra com mais de 1 MB de memória, normalmente 2 MB ou 4 MB.

Mas o primeiro megabyte de memória apresenta a característica muito especial de poder ser endereçado pelo CPU quando este está a trabalhar em **modo real**, enquanto a memória acima de 1 MB só é endereçável e, por conseguinte, acedível pelo CPU a trabalhar em **modo protegido**.

Por esse motivo, a esse primeiro megabyte de memória dá-se muitas vezes o nome de **memória real** e à restante memória existentente acima desse primeiro megabyte dá-se o nome de **memória extendida**.

Punhamos agora aqui um importante parêntesis, para referir que pode existir ainda um terceiro tipo de memória (que gera muita confusão) e que dá correntemente pelos nomes de **memória expandida,** memória EMS (Expanded Memory Specification) ou memória LIM (Lotus/Intel/Microsoft). Este tipo de memória não é directamente endereçável pelo CPU; ao invés disso é constituída por blocos designados **páginas**, cada uma das quais normalmente com 16 KB de tamanho.

À informação guardada pelos programas em uma ou mais páginas da memória expandida é associada pelo *device driver gestor da memória expandida* um número designado por **handle**. As páginas de memória expandida associadas a um determinado *handle* são, por sua vez, numeradas a partir de zero recebendo o nome de **páginas lógicas.**

Os programas acedem à memória expandida informando o *device driver* sobre qual o *handle* e páginas lógicas que pretendem e este procede ao seu «mapeamento» numa zona normalmente de 64 KB da memória real que se designa por **page frame**.

A memória expandida pode ser obtida basicamente por três métodos:

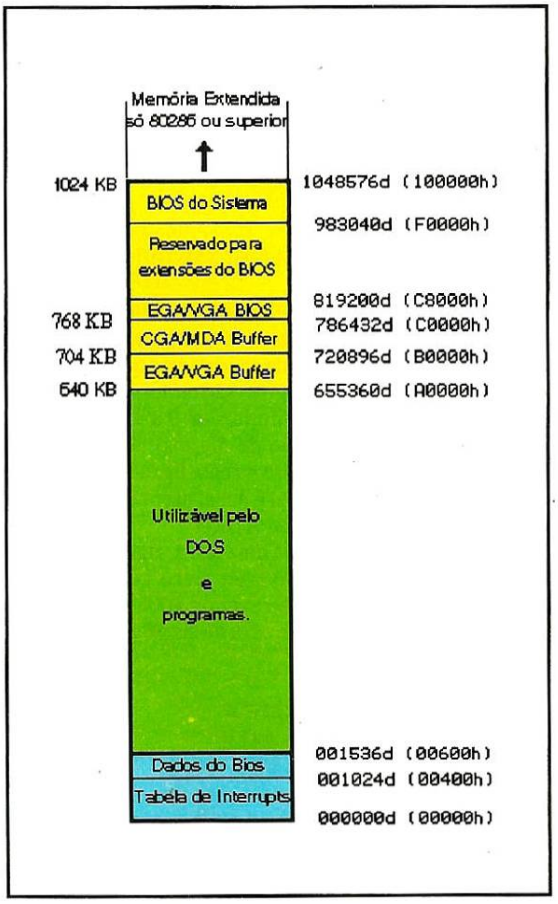
a) Através de placas adequadas instaladas numa *slot* de expansão ou algumas vezes *chips* integrados na placa principal. Este é o método clássico e é largamente utilizado nos computadores de processador 80286 (e alguns 8088/8086...). Para fazer o *interface* da placa com os programas é normalmente fornecido pelo fabricante um *device driver* adequado.

b) Por emulação e utilizando a memória extendida como base. É um procedimento também por *hardware* (ao nível do CPU) e só possível em computadores com processadores 80386 ou superiores e com o processador a trabalhar em modo virtual 86. O *device driver* EMM386.SYS (existem outros tais como o QEMM386.SYS e o 386MAX.SYS) fornecido com as versões recentes do MS-DOS faz aqui o *interface* entre os programas e o CPU.

c) Por simulação pura e simples (totalmente por *software*) utilizando memória extendida ou convencional como base. O conhecido *device driver* EMS40.SYS da PcMag é um bom exemplo do que se pode fazer mesmo quando não se tem o *hardware* adequado (e se está disposto a sacrifícios em termos de velocidade).

Mas esqueçamos este tipo especial de memória e debruce-mo-nos agora sobre a memória directamente en-

dereçável e com particular detalhe sobre a memória real. Repare na figura seguinte e nas três regiões coloridas respectivamente a azul,verde e amarelo:



As regiões azul e verde constituem aquilo que se designa por **memória convencional.** Nos computadores com processadores 8088 e 8086 a RAM é localizada apenas nesta zona, o que significa que esta poderá ter um máximo de 640 KB. Os computadores com processadores 80286 ou superiores podem, contudo, ter mais de 640 KB de RAM mas, nesse caso, o excesso estará localizado «fisicamente» a partir do endereço 1048576 (vd. a zona branca na figura) naquilo que se referiu atrás como sendo a **memória extendida.**

Os primeiros 64 KB menos 16 bytes de memória extendida recebem o nome de **HMA** (High Memory Area) e apresentam a propriedade interessante de poderem ser endereçados em modo real quando a 21ª linha de endereçamento (linha A20) é disponibilizada para uso nesse mesmo modo modo real. A faculdade de se poder endereçar a HMA em modo real só é possível devido à interpretação segmentada da memória (ora cá está uma vantagem, pensarão os leitores que estavam a ficar com a opinião que a segmentação da memória só parecia ter aspectos negativos...) pelos processadores desta família.

Imagine que um registo de segmento, por exemplo o registo ES, contém o valor hexadecimal FFFFh (ou a 15x4096+15x256+15x16+15x1 = 65535 em decimal). Imagine também que o registo SI contém o valor hexadecimal 10h (que corresponde em decimal a: 1x16+0x1 = 16).

Se se recorda ainda do que dissemos no artigo anterior, sabe que os registos de segmento percorrem a memória a «passadas» de 1 parágrafo de cada vez (cada parágrafo são 16 bytes) e os registos que nos dão o deslocamento percorrem essa mesma memória a «passadas» de apenas 1 byte de cada vez. Deste modo, uma instrução *Assembly* como por exemplo:

MOV AX, ES:[SI]

que informa o processador para **MOV**er para o registo AX o valor existente na célula de memória cujo endereço absoluto é dado pelo valor do registo de segmento ES vezes 16 (decimal) e somado do valor do registo SI, tem como resultado, no nosso caso, carregar para o registo AX o valor contido na célula de memória cujo endereço absoluto é 1,048,576 (65535x16 + 16).

Se reparar na figura anterior, verá que o endereço absoluto que foi obtido corresponde exactamente ao ínicio da memória extendida!

Pela mesma linha de raciocínio veríamos que o endereço maior que é possível de produzir pelo processador a trabalhar em modo real é FFFF:FFFF (esta é a convenção para se representar um endereço segmentado: segmento:deslocamento), isto é, os registos de segmento e deslocamento estarem carregados ambos com o valor FFFFh. Isto corresponde ao endereço absoluto:

65535x16+65535 = 1,114,095

A área de memória compreendida entre os endereços absolutos 1,048,576 e 1,114,095 (ou 100000h e 10FFEFh em hexadecimal) é precisamente a HMA. E, como se vê, o seu tamanho é precisamente 64 KB menos 16 bytes. Note que é condição para a HMA poder ser acedida em modo real, a linha A20 estar aberta (e esta é uma das missões do conhecido *device driver* HIMEM.SYS: abrir e fechar a HMA consoante as necessidades e as instruções que recebe do MS-DOS e outros programas), pois em caso contrário dá-se um *wrap around* (isto é, começamos de novo pelo endereço absoluto 00000h). A importância da HMA tem sido extraordinária nos tempos mais recentes, em particular pelo uso que o MS-DOS, Windows e alguns outros programas lhe têm dado. Mas regressemos à descrição das áreas de memória tomando como «pano de fundo» o desenho anterior.

A memória convencional é sempre limitada superiormente pelo endereço absoluto 655360 em decimal ou A0000h em hexadecimal, excepto quando o computador tem menos de 640 KB de RAM; nesse caso a memória convencional ficará reduzida na respectiva proporção. A região azul ocupa o início da memória convencional e é reservada por um programa residente no topo do ROM (chamado *Basic Input Output System* ou abreviadamente BIOS) para seu uso próprio e para uso pelo sistema operativo MS-DOS.

A região azul compreende a **Tabela de Interrupts** de que falaremos mais abaixo e a **Bios Data Area.**

Como os Bios (do Sistema e do Video) estão em ROM que é memória que só permite a escrita, surge a necessidade de guardar a informação temporária em algum sítio; esse sítio é precisamente a *Bios Data Area.*

Para além dos BIOS, ainda o DOS e a linguagem BASIC (porque é boa companheira dos PC *made by* IBM) têm direito a alguns endereços nessa região. Com tantos candidatos houve necessidade de reservar 512 bytes para guardar a informação temporária de todos eles.

Bem, para ser mais preciso, os PS/2 da IBM e *clones* muito semelhantes reservam ainda uma outra área em local não predeterminado (algumas vezes no último kilobyte da memória convencional) designada *Extended Bios Data Area.* Guarda-se na EBDA por exemplo informação temporária do *mouse.*

Muitos endereços da Bios Data Area têm excepcional interesse para o programador em linguagem *Assembly.* Contudo abstemo-nos de os transcrever, pois o espaço de revista que isso exigiria seria excessivo, e eles constam de qualquer bom livro sobre programação de PC.

Esperamos contudo dar nos próximos números um ou outro exemplo de programação que inclua informação

“A memória convencional é sempre limitada superiormente pelo endereço absoluto 655360 em decimal ou A0000h em hexadecimal”

recolhida nesses famosos endereços.

A região verde é ocupada parcialmente pelo sistema operativo MS-DOS (entre um mínimo actual de cerca de 12 KB até mais de 100 KB consoante a versão e a configuração, e também reconhecendo que uma parte do sistema operativo pode ser colocada na HMA nas versões mais recentes) e é também onde os nossos programas são normalmente carregados para serem executados. O sistema operativo MS-DOS é constituído, como se sabe, por três programas: IO.SYS, MSDOS.SYS e COMMAND.COM.

Acima da memória convencional e com início **sempre** em A0000H inicía-se o ROM (a amarelo na figura). Esta região, que totaliza 384 KB, é designada também por **Upper Memory** (de preferência não traduza pois existe alguma falta de uniformidade no significado das várias traduções existentes). Os primeiros 128 KB do ROM são reservados às placas gráficas de todos os tipos existentes para poderem apresentar ao sistema uma área que permita a escrita (a qual deixará nesta altura de ser ROM na verdadeira acepção da palavra). Essa área é muitas vezes designada por **buffer da placa gráfica** (veja-a como se fosse uma janela aberta pelo sistema para a **memória video** ou **display memory** existente na placa gráfica) e pode ocupar a região B0000H-B8000H (placas MDA e Hercules), a região B8000H-C0000H (placas CGA) ou eventualmente qualquer parte ou toda a região A0000H-C0000H (placas EGA, VGA e SuperVGA), tudo dependendo do modo gráfico emulado.

> **“ Acima da memória convencional e com início sempre em A0000H inicía-se o ROM ”**

A região seguinte do ROM e com início no endereço C0000H é normalmente ocupada pelo programa gestor do funcionamento das placas gráficas EGA, VGA e SuperVGA, o qual é designado por *Video Bios* dessas placas. O *Video Bios* ocupa uma área variável entre 24 KB e 32 KB. Ao *Video Bios* segue-se uma região muitas vezes desocupada excepto quando o sistema possui equipamentos periféricos que requeiram o **mapeamento** de programas nessa zona, os quais são designados **extensões do BIOS**.

Finalmente, a região de ROM com início em F0000H, é ocupada pelos vários programas que permitem o arranque do sistema e se ocupam, a partir daí, em faciltar um diálogo entre o *hardware* e o MS-DOS ou as nossas aplicações. Esse conjunto de programas é designado por **BIOS do Sistema**.

Todos os programas residentes em ROM ou mapeados nessa zona por equipamentos ou dispositivos são também genéricamente designados por **firmware,** para pôr em evidência o seu carácter de permanência. As áreas da Upper Memory não utilizadas pelo *firmware* podem, muitas vezes, ser aproveitadas para «mapeamento» de RAM existente na memória extendida ou expandida. Isso é possível com todos os computadores de processador 80386 ou superior e também com alguns 80286.

Após o mapeamento de RAM ficam constituídos **Upper Memory Blocks (UMB)** que podem ser aproveitados para carregamento de programas ou dados. Os UMB são criados e geridos com o auxílio de programas designado **gestores de UMB** (UMB *providers*), os mais conhecidos dos quais são o EMM386.SYS, QEMM, 386MAX e QRAM.

E é apenas isto o que de mais essencial há a saber sobre a memória.

Mas não queremos passar ao capítulo seguinte sem lhe chamar a atenção para um facto muito interessante.

O leitor deve já ter reparado que a base númerica hexadecimal se «encaixa às mil maravilhas» na estrutura segmentada de memória dos PC, e que é muito mais fácil descrever um endereço de memória em termos hexadecimais do que em termos decimais. Basicamente, isto tem a ver com o facto do tamanho em bytes de 1segmento de memória ser representado pelo valor 65536 em numeração decimal, mas por 10000 em numeração hexadecimal. E se fizer o favor de olhar mais uma vez para a figura, vai notar que os endereços absolutos dos ínicios das varias regiões e blocos de memória são mais facilmente memorizáveis se descritos pela numeração hexadecimal. Por exemplo, é mais fácil dizer que o BIOS do sistema se inicia no segmento F000h (ou no endereço absoluto F0000h), do que dizer que se inicia no segmento 61440 (ou no endereço absoluto 983040). Por ser tão importante a base de numeração hexadecimal vamos ver como se faz contas com ela.

2- A «TABUADA» HEXADECIMAL

Diz-se (muito intuitivamente) que uma **base** (ou **radix**) é hexadecimal porque existem 16 símbolos (designados **dígitos**) com os quais se pode representar qualquer quantidade mensurável. Esses dígitos são os nossos já conhecidos (pelo sistema decimal) dígitos de 0 a 9 e as letras do alfabeto (aqui com a função de dígitos) de A a F. Para diferenciar um número escrito na base hexadecimal de um outro escrito na base decimal é usual escrever-se um *h* no final. Por exemplo: 4321h ou A100h

Quando o primeiro dígito é uma **letra**, alguns assembladores exigem que se escreva um **zero** como primeiro dígito. Assim, dever-se-á escrever 0A100h e não A100h.

O leitor já sabe que um número no sistema decimal, por exemplo 1234 equivale a:

$$1x10^3 + 2x10^2 + 3x10^1 + 4x10^0 =$$
$$1x1000 + 2x100 + 3x10 + 4x1 =$$
$$1000 + 200 + 30 + 4 = 1234$$

Pois na base hexadecimal é a mesma coisa, só que 10h no sistema hexadecimal equivale a 16 no sistema decimal!

Se o número 1234h estivesse na base hexadecimal, qual seria o seu equivalente na base decimal?

Muito simplesmente:

$$1x16^3 + 2x16^2 + 3x16^1 + 4x16^0 =$$
$$1x4096 + 2x256 + 3x16 + 4 =$$
$$4096 + 512 + 48 + 4 = 4660$$

E no caso contrário, se tivéssemos o número 4660 em decimal e quiséssemos saber o seu equivalente em hexadecimal? Procedíamos a divisões sucessivas por 16:

4660/16 = 291 e resto 4.
291/16 = 18 e resto 3
18/16 = 1 e resto 2

E o número procurado é constituído pelo último quociente e pelos vários restos, isto é o número 1234h. E agora esta pequena questão:

Qual é o método prático de converter um endereço segmentado num endereço absoluto? Como se sabe os endereços segmentados vêm em princípio em formato hexadecimal e separados por 2 pontos, por exemplo:

1AB4:21A5

O método consiste em efectuar uma soma em que o valor do segmento é multiplicado implicitamente por 10h (um parágrafo) antes de ser adicionado ao deslocamento. Para o endereço segmentado acima indicado ficaria:

1AB40h
+21A5h

1CCE5h

Claro que neste exemplo não houve situações de «e vai um». Mas se houvesse, também não seria «nada do outro mundo». Por exemplo, para o endereço segmentado FFFF:FFFF teremos:

FFFF0h
+FFFFh
10FFEFh

E o número 10FFEFh corresponde a que número no sistema decimal? Vamos ver:

$1x16^5 + 0x16^4 + 15x16^3 + 15x16^2 + 14x16^1 + 15x16^0 =$
1x1048576 + 0x65536 + 15x4096 + 15x256 + 14x16 +
+15x1= 1 114 095

Que foi o mesmo número que obtivemos mais atrás quando falámos do limite da HMA. E de facto a matemática não mente!

3 – A TABELA DE INTERRUPTS

Interrupts (designados em*assembly* pela mnemónica **INT**) são acções que determinam interrupções no fluxo normal de um programa e desviam a atenção do CPU para outros programas designados *interrupt handlers* (**rotinas de serviço de interrupts**).

Existem 256 *interrupts* diferentes e numerados de 00h a FFh. Os *interrupts* podem ser gerados quer por *hardware* quer por *software*. Quando são gerados por *hardware* (por exemplo por um periférico como o teclado, pelo relógio do sistema ou pelo controlador do disco rígido) é necessário um *chip* que coordene os *interrupts* já que não é possível ao CPU fazer executar dois *interrupt handlers* simultaneamente. Esse*chip* designa-se **PIC** *(Programmable Interrupt Controller)*. Quando os*interrupts* são gerados por *software*, é a própria instrução **INT** seguida de um número que provoca a iniciação de um *interrupt*.. Por exemplo: INT 20h. Os *interrupts* são interessantes porque representam um modo bastante económico de fazer executar um trabalho.

Por exemplo, introduz-se INT 20h no seio de um programa e o CPU sabe exactamente o que tem de fazer.

Quer o *interrupt* seja emitido por *hardware*, quer por-*software*, o microprocessador apenas necessita conhecer o seu número para proceder sempre do seguinte modo:

1 – O conteúdo do registo *flags* é armazenado no *stack*. Logo, o valor do *stack pointer* (registo SP) é diminuido em 2.

2 – Os flags *Interrupt Flag* e *Trap Flag* são levados a 0. Recorde-se que no artigo anterior escrevemos que o *Interrupt Flag* em 0 impede que outra interrupção ocorra antes desta que se inicia ter sido tratada.

O *Trap Flag* a 0 impede a execução passo a passo do programa. Programas como o Debug do MS-DOS usam este *flag* para poderem efectuar a execução de programas ao «ralenti».

3 – O conteúdo do registo CS *(Code Segment)* é também armazenado no *stack*.

4 – O conteúdo do registo IP *(Instruction Pointer)* é tal como os anteriores copiado para o *stack*. Neste momento o valor de SP já diminuiu de 6 unidades em relação ao valor inicial.

5 – Multiplica o número do *interrupt* por 4 e o resultado obtido serve de índice na Tabela de Interrupts. Por exemplo, se o *interrupt* for o INT 10h, o microprocessador multiplica 10h por 4 e obtém 40h. Os quatro bytes que se iniciam no endereço 0000:0040 formam por sua vez um endereço onde se encontra o respectivo *interrupt handler*. A situação é um pouco semelhante a ir perguntar à Maria onde está o Manel.

Imagine agora que os bytes eram os seguintes:

0000:0040 12
0000:0041 .... A0
0000:0042 30
0000:0043 14

Então o endereço do *interrupt handler* é: 1430:A012
Exactamente este e não: 12A0:3014

Isto porque os PC armazenam em memória os números de trás para diante: um byte **mais significativo** encontra-se posicionado num endereço de memória mais elevado que um byte **menos significativo**. Fique alertado, contudo, para o facto de que nem todos os computadores assim procedem, mas também não é um «mal exclusivo» dos PC.

6 – Então (e aproveitando o endereço que obtivemos no ponto 4), o CPU coloca os 2 primeiros bytes no registo IP, logo IP ficará com o valor 0A012h.
Coloca os dois seguinte, 1430h no registo CS.

7 – Após essa operação a execução inicia-se precisamente no endereço 1430:A012, que é o ínicio do *interrupt handler*.

8 – No final do *interrupt handler* existe uma instrução representada em *Assembly* por **IRET** (Interrupt Return). A execução dessa instrução determina a inversão do processo, designadamente,

9 – O conteúdo do «velho» IP é retirado do *stack* e restaurado nesse registo.

10 – O conteúdo do «velho» CS idem aspas.

11 – Os *flags* também são restaurados.

12 – E muito naturalmente a execução prossegue como se nada se tivesse passado.

Mas, na realidade, muita coisa mesmo se pode ter passado naqueles milisegundos que durou a execução desse *interrupt handler*. Muito, muito mais há a dizer sobre os *interrupts*, por isso reservamos o nosso próximo artigo exactamente para desenvolver o tema e fazer tentativamente um pequeno programa que lide com os ditos cujos *interrupts*. E vai ver que muito em breve está a programar em *Assembly* sem dar por isso, porque a linguagem *Assembly* é efectivamente muito simples, o que é dificil é... a *entourage*.

“ Interrupts (designados em*assembly* pela mnemónica INT) são acções que determinam interrupções no fluxo normal de um programa e desviam a atenção do CPU para outros programas ”



José Páscoa

Programar em Clipper (PARTE III)

IMPRESSÃO DE UMA BASE DE DADOS

Para a saída da informação em papel, o dBASE dispõe de vários comandos que lhe permitem imprimir o conteúdo da base de dados. Esses comandos são: **LIST**, **DISPLAY**, **@...SAY**.

COMANDO LIST

A sintaxe deste comando é:

```
LIST [<registos>] [<campos>]
[FOR/WHILE <condição>] [OFF] [TO PRINT]
```

A opção <registos> foi descrita no número anterior. <Campos> indica quais os campos a listar e que obedeçam à <condição> definida. O dBASE imprime automaticamente o número do registo antes dos campos. Para suprimir este procedimento, acrescente a opção OFF ao comando. TO PRINT indica que a saída será dirigida para a impressora.

Por exemplo, para listar os campos NOME,MORADA e TELEFONE da nossa Agenda, executamos o comando:

```
.LIST nome,morada,telefone
```

No caso de se pretender apenas os registos que obedeçam a determinada condição e sem o número do registo:

```
.LIST OFF nome,morada,telefone FOR telefone="215"
```

Para enviar a mesma listagem para a impressora:

```
.LIST OFF nome,morada,telefone FOR telefone="215" TO PRINT
```

COMANDO DISPLAY

Este comando é similar ao LIST, e tem a seguinte sintaxe:

```
DISPLAY [<registos>] [<campos>]
[FOR/WHILE <condição>] [OFF] [TO PRINT]
```

Exibe uma listagem idêntica ao LIST, provocando no entanto uma pausa, página a página. Digitando o comando sem qualquer opção, é mostrada somente a informação relativa aos campos do registo em que estiver posicionado o ponteiro do dBASE.

Por exemplo, se o ponteiro do dBASE se encontrar posicionado no registo número 3 e pretendermos ver os campos NOME e TELEFONE dos restantes registos, o comando será o seguinte:

```
.DISPLAY REST nome,telefone OFF
```

Para obter uma listagem contendo o NOME e o CÓDIGO POSTAL dos próximos 10 registos, o comando seria:

```
.DISPLAY NEXT 10 nome,cod_postal
```

COMANDO @...SAY

Para escrever determinada informação (texto ou variável) numa posição específica do ecrã ou da impressora, utiliza-se este comando com a seginte sintaxe:

```
@ <linha>,<coluna> [SAY <expressão>
[PICTURE <imagem>]]
```

Linha e coluna são as coordenadas onde deverá surgir a <expressão> indicada. O ecrã está dividido em 25 linhas e 80 colunas, como a figura 1 indica:

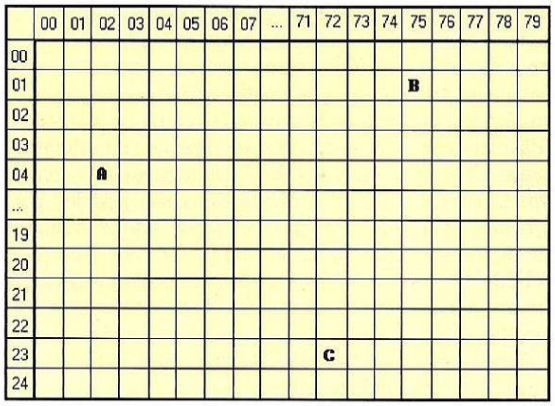


Figura 1 – Coordenadas do ecrã

As coordenadas dos pontos assinalados são: A = 4,2; B = 1,75; C = 23,72

As linhas são numeradas de 0 a 24 e as colunas de 0 a 79. A coordenada 0,0 é o canto superior esquerdo e a coordenada 24,79 é o canto inferior direito. Depois de indicar um posicionamento correcto, com o comando @...SAY, poderá ser exibida uma determinada mensagem ou o conteúdo de uma variável.

Se digitar a seguinte instrução na linha de comando do dBASE, a mensagem SPOOLER aparecerá na linha 12,coluna 35 (os dados do tipo carácter deverão ser escritos entre aspas):

```
.@ 12,35 SAY "SPOOLER"
```

A apresentação de variáveis no ecrã é feita da seguinte forma:

```
.soma=30+12
.@ 10,40 say soma
```

Neste caso a variável não pode estar entre aspas.

Opção PICTURE

A opção PICTURE é usada com o comando @...SAY para formatar dados. Esta opção estabelece o formato de visualização dos dados.

Símbolo	Significado
A	Mostra apenas os caracteres alfanuméricos
B	Os dados numéricos alinham-se à esquerda
C	Mostra "CR" (crédito) após um número positivo
D	Mostra as datas no formato americano
E	Mostra as datas no formato Britânico
X	Mostra "DB" (débito) após um número negativo
Z	Não mostra o zero num valor numérico
(	Envolve com parênteses os números negativos

Figura 2 – Funções disponíveis para a opção PICTURE

Símbolo	Significado
9	Permite apenas dígitos e sinais usados nas operações matemáticas
#	Idêntico ao anterior, mas permite espaços em branco
A	Permite apenas caracteres alfabéticos
L	Permite apenas dados lógicos (.T.,.F.)
N	Permite apenas letras e dígitos
X	Permite qualquer carácter
!	Converte letras minúsculas em maiúsculas
$	Mostra o cifrão à esquerda de um número
*	Mostra um asterisco à esquerda do número
.	Permite especificar a posição do ponto décimal
,	Mostra uma vírgula se for necessário separar os milhares

Figura 3 – Caracteres de selecção

As funções e os caracteres de selecção deverão estar entre aspas. O simbolo @ deve preceder sempre qualquer função. Exemplos:

```
.soma=20+55
.@ 10,10 SAY soma PICTURE "@C"
```

Estas linhas enviam para a linha 10,coluna 10 a mensagem: 75 CR

```
.texto="revista spooler magazine"
.@ 15,10 SAY texto PICTURE "!XXXXXXX!!!!!!!X!XXXXXXX"
```

O resultado deste exemplo seria: Revista SPOOLER Magazine.

```
.var=128787845.798
.@ 20,10 SAY var PICTURE "###,###,###.##"
Na coordenada 20,10 apareceria: 128,787,845.79
```

Comando SET DEVICE TO...

Sintaxe: SET DEVICE TO SCREENIPRINTER

A saída é dirigida para a impressora ou para o ecrã através do comando SET DEVICE TO... Se adicionarmos a palavra SCREEN ao comando, envia a saída para o ecrã, enquanto que PRINTER envia-a para a impressora.

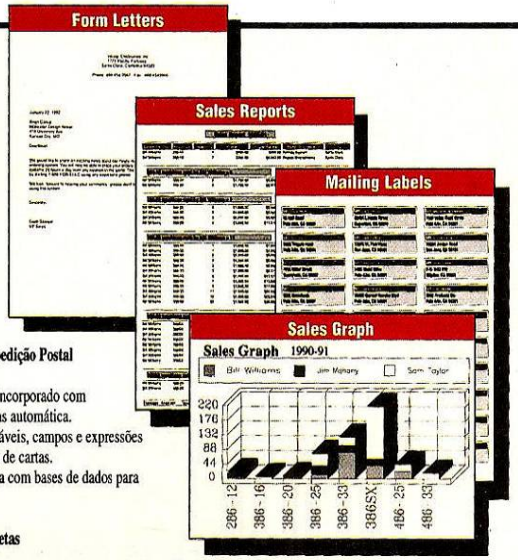
As seguintes linhas de programa exemplificam o que acabámos de dizer:

```
SET DEVICE TO SCREEN   && por defeito o resultado é para o ecrã
@ 10,05 SAY "Esta mensagem será escrita na linha 10,coluna 5 do monitor"
SET DEVICE TO PRINTER
@ 15,20 SAY "Esta será escrita na linha 15,coluna 20 da impressora"
EJECT   && faz um salto de página
```


ACEFILE™

dBASE em Windows
Potência e Simplicidade

1992 W 100 WINDOWS MAGAZINE ANNUAL WIN AWARD



AceFile proporciona-lhe a gestão de bases de dados de modo simples, rápido e intuitivo. AceFile reúne a potência do mais popular gestor de bases de dados – dBASE – com a simplicidade do ambiente Windows. No AceFile encontrará, também, procedimentos que não existem em qualquer outro gestor de bases de dados para o Windows!

Especificações da base de dados

- Compatibilidade com ficheiros dBASE e de índice.
- Abertura simultanea até 10 ficheiros de bases de dados.
- Abertura até 10 ficheiros de índices por base de dados.
- Expressões e filtros compatíveis com dBASE.

Relatórios

- Número ilimitado de relatórios
- Opções rápidas de construção: através de um clique no «rato» gera o relatório mais apropriado.
- Contruções WYSIWYG com linhas, rectângulos, círculos e campos.
- Suporta fontes, cabeçalhos, rodapés e números de página.
- Suporte para agrupamentos e sub-totais.
- Envio de relatórios para impressora, ecrã ou pré-visualização em modo WYSIWYG.
- Campos calculados e ordenação automática de grupos calculados.
- Suporte para fontes e estilos Windows.

Utilização para Expedição Postal

- Editor de texto incorporado com partição de linhas automática.
- Inserção de variáveis, campos e expressões para formatação de cartas.
- Mistura selectiva com bases de dados para impressão.

Impressão de Etiquetas

- Inclusão de formatos de etiquetas normalizadas Avery Laser Printer Labels, com um simples clique com o «rato».
- Fácil personalização do desenho das etiquetas.

Visualização de Formulários

- Apresentação de dados em formato corrido ou registo a registo.
- Desenho de formatos personalizados, utilizando campos, fontes, cores e ferramentas de desenho.
- Utilização de campos calculados nos formulários.
- Visualização da informação a partir de tabelas ou bases de dados.
- Controlo através de botões para fácil movimentação entre janelas.

Pesquisas

- Utilização de expressões-filtro , para selecção de registos.
- Pesquisa em qualquer campo de base de dados incluindo campos memo.
- Apagamento, cópia, actualização de registos seleccionados ou tabelas de registos.
- Ordenação, ascendente ou descendente, por qualquer campo.

Importação/Exportação

- Importação/exportação de dados para ficheiros Lotus 123, Symphony e ASCII.

AceFile, dBASE, Windows, Lotus 123 e Symphony são marcas registadas

Distribuidor:

CIPEVE Lda

Rua Vasques de Mesquita, 259 - 4º C – 4300 PORTO
Telef.: (02) 49-6551 – Fax: (02) 410-6327

ACE SOFTWARE™

SOLICITE-NOS VERSÃO DE DEMONSTRAÇÃO GRÁTIS

INTRODUÇÃO AO CLIPPER

O Clipper Sumer 87 é o compilador mais conhecido do dBASE (entende-se por compilador o programa que transforma a linguagem fonte num programa executável, de extensão COM ou EXE). Apesar de ser compatível com quase todas as instruções do dBASE, o Clipper dispõe de muitas outras próprias que lhe permitem controlar quase todo o ambiente em que opera.

Podemos dizer que enquanto o dBASE se destina ao utilizador em geral, o Clipper está mais dirigido para os programadores.

Como todas as linguagens, o Clipper também tem vantagens e desvantagens. Eis algumas:

Vantagens:
1) Rapidez;
2) Impossibilidade de alteração da linguagem fonte por terceiros;
3) Possibilidade de utilizar funções de outras linguagens.

Desvantagens:
1) Tamanho de cada programa depois de compilado;
2) Impossibilidade de trabalhar em modo gráfico (sem funções externas).

“Podemos dizer que enquanto o dBASE se destina ao utilizador em geral, o Clipper está mais dirigido para os programadores”

FICHEIROS DO CLIPPER

Para compilar e executar um programa, apenas são necessários os seguintes ficheiros:

CLIPPER.EXE
TLINK.EXE ou PLINK86.EXE ou LINK.COM
CLIPPER.LIB
EXTEND.LIB

O ficheiro CLIPPER.EXE transforma as linhas de comandos (ficheiros PRG's) em «programas-objecto» (OBJ). A passagem de um programa-objecto para um executável (extensão EXE), é feita através de um «ligador». Este tem como missão reunir todos os programas objectos num só programa executável.

Existem vários ligadores no mercado: a Microsoft criou o LINK, a Nantucket (a criadora do Clipper) o PLINK86 e a Borland o TLINK (Turbo Link). É este último que vamos explorar devido à sua maior rapidez e simplicidade.

Os ficheiros CLIPPER.LIB e EXTEND.LIB são as bibliotecas do Clipper. É nestas bibliotecas que estão guardados os comandos e as funções do Clipper.

COMO UTILIZAR O COMPILADOR

Existem várias opções que determinam qual o modo em que o compilador deve operar:

-l (sem o número das linhas)
-m (compilar apenas um módulo)
-o (path do programa objecto)
-p (pausa para mudança de disco)
-s (verificação da sintaxe)
-v (as variáveis passam a M->)

A seguinte instrução indica ao compilador que deverá transformar o programa PROG.PRG em PROG.OBJ, sem incluir o número de cada linha (-l) e os sub-programas existentes (-m).

```
C>clipper prog.prg -l -m
```

COMO UTILIZAR O LIGADOR

O TLink tem a seguinte sintaxe:

```
TLINK prog_obj, prog_exe, fich_map, fich_libs
```

FICH_MAP é um ficheiro onde o TLINK guarda todas as mensagens resultantes do processo de ligação. FICH_LIBS são os nomes das bibliotecas que são necessárias.

Para transformar o nosso programa PROG.OBJ em executável, o comando seria o seguinte:

```
C>TLINK prog,progexe,mapa,extend clipper
```

PROGRAMA AGENDA II

*Nota da Direcção: No número anterior da **Spooler** dissemos que publicaríamos as «fontes» do programa AGENDA no presente número. Assim é, com uma pequena diferença que pensamos não prejudicar o leitor: as «fontes» presentes na disqueta referem-se à nova versão aqui descrita pelo nosso colaborador Victor Monteiro e não à versão anterior.*

Na disqueta que segue com a revista, existe um ficheiro denominado CLIP3.EXE, que depois de descompactado dará origem aos seguintes ficheiros:

AGENDA.PRG – listagem do programa agenda;
AGENDA.DBF – base de dados da agenda;
AGENDA.LBL – ficheiro das etiquetas.

Para executar o programa, copie estes três ficheiros para a directoria do dBASE e chame o dBASE. Na linha de comando digite:

```
.DO agenda
```

Este programa tem as opções que foram descritas neste e no artigo anterior. Para ver como estas foram desenvolvidas, digite:

```
.RUN TYPE agenda.prg >PRN
```

A opção >PRN envia a listagem para a impressora.

NO PRÓXIMO NÚMERO...

Introdução à programação em Clipper.
Passagem do programa escrito em dBASE para Clipper (utilizando os novos comandos).

Vítor Manuel Monteiro

PROGRAMAR EM C

(PARTE III)

INTRODUÇÃO

Nesta terceira lição do nosso curso vão ser analisados mais alguns aspectos fundamentais da linguagem C. Entre eles destaca-se um dos mais importantes de toda a linguagem: as **funções**. Segue-se um breve sumário dos assuntos focados ao longo desta lição:

- Estruturas
- Uniões
- Funções
- Recursividade
- Sintaxe das funções mais importantes da *Standard Library*

ESTRUTURAS

Em C uma estrutura (*struct*) é a forma utilizada para agrupar variáveis, relacionando-as todas com um nome. Este tipo de dados é muito útil quando pretendemos agrupar dados estreitamente relacionados.

Sintaxe básica

A declaração básica de uma estrutura é da seguinte forma:

```
struct nome {
        tipo var1;
        tipo var2;
        ...
        tipo varn;
        } lista_de_vars;
```

nome indica ao compilador o nome pelo qual a estrutura será conhecida. **lista_de_vars** é uma lista (separada por vírgulas) de identificadores que serão associados à estrutura. Às variáveis da estrutura é normalmente dado o nome de «elementos da estrutura». Por exemplo:

```
struct morada {
        char nome[30];
        char morada[30];
        char localidade[25];
        int codigo_postal;
        } carlos, manuel, rui;
```

Para aceder a um elemento de uma estrutura, faz-se referência ao nome da variável, seguindo-se um ponto (.) e por último o nome do elemento da estrutura que se pretende aceder. Considerando o último exemplo, os seguintes casos são válidos:

```
carlos.codigo_postal=2665;
strcpy(rui.nome, "Rui ...");
printf("A morada do rui é: %s\n", rui.morada);
```

Inicialização de estruturas

A inicialização de uma **estrutura** é idêntica à dos array, estudados na lição anterior. Os valores inicializadores deverão ser separados por vírgulas e deverão ser todos envolvidos por chavetas ({...}). Segue-se um exemplo:

```
struct abc {
        int num;      /* inteiro */
        char str[80]; /* string de 79 caracteres */
        char ch;      /* char */
        } a, b={55, "SPOOLER", 'K'}, c={0, "LINGUAGEM C", 90};
```

Arrays de estruturas

Tal como com variáveis normais, também é possível declarar **arrays de estruturas**. Basta colocar, seguido do nome da variável associada à estrutura,o número de elementos do array entre parêntesis rectos. Por exemplo:

```
struct {
      int x, y, z;
      } _3D[20];

int i;

for  (i=0; i<20; i) {
      _3D[i].x=i;
      _3D[i].y=-i;
      _3D[i].z=0;
}
```

Determinando o espaço ocupado por uma estrutura

Para evitar que o programador tenha de fazer contas para determinar o tamanho de uma estrutura, existe um operador no C que nos dá o tamanho em bytes do seu argumento. Este argumento poderá ser um **tipo** ou então uma **variável**. O nome deste operador é **sizeof(...)**, seguem-se alguns exemplos:

```
struct morada rui;

sizeof(int);              /* tamanho de um int */
sizeof(struct morada);    /* tamanho da estrutura morada */
sizeof(rui);              /* tamanho da estrutura morada */
```

Ponteiros para estruturas

Tal como para os tipos em geral, também podemos declarar **ponteiros para uma estrutura**. A declaração é idêntica à dos outros tipos (precedendo o nome com um *), como se pode ver no exemplo seguinte:

```
struct morada * pessoa;
```

> **“Em C uma estrutura *(struct)* é a forma utilizada para agrupar variáveis, relacionando-as todas com um nome”**

Neste caso, a variável **pessoa** contém simplesmente um endereço de memória. Para aceder aos elementos da estrutura colocada no local onde **pessoa** aponta, poderemos proceder de dois modos:

```
(*pessoa).morada
  ou
pessoa->morada
```

Como facilmente se percebe, o segundo é bem mais simples, daí ser o mais utilizado. Segue-se um exemplo de utilização de um ponteiro para uma estrutura:

```
#include <stdio.h>

struct ponto {
    int x, y;
    };

void main(void)
{
    struct ponto a, * p_ptr;

    p_ptr=&a;   /* p_ptr aponta para a */
    p_ptr->x=20; /* igual a a.x=20 */
    p_ptr->y=55; /* igual a a.y=55 */

    /* escreve o conteúdo de a via p_ptr */
    printf("\n a.x=%i, a.y=%i \n", a.x, a.y);
}
```

> **Ao contrário de muitas outras linguagens de programação, o C permite aceder individualmente a um ou mais bits dentro de um byte**

Bit-fields

Ao contrário de muitas outras linguagens de programação, o C permite aceder individualmente a um ou mais **bits dentro de um byte**. O método utilizado para fazer este tipo de operação é através de uma estrutura. A sintaxe básica é a seguinte:

```
struct nome {
    tipo nome1:tamanho;
    tipo nome2:tamanho;
    ...
    tipo nomen:tamanho;
    } lista_de_vars;
```

tipo poderá ser: **int**, **unsigned** ou **signed**. Se o tamanho for 1, o tipo terá de ser obrigatoriamente unsigned, isto porque só com um bit é impossível especificar um valor e um sinal. Segue-se um exemplo:

```
struct abc {
    unsigned val1:2;
    int val2:5;
    unsigned val3:1;
    int val4:6;
    }
```

As variáveis em **lista_de_vars** terão dois bytes de tamanho máximo, isto é, 16 bits. Resta ainda focar que neste tipo de estrutura tudo se passa como se de uma simples estrutura se tratasse. Por exemplo:

```
struct abc a;
a.val3=1;
a.val6=25;
a.val2=-6;
```

UNIÕES

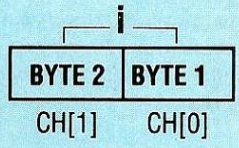
Em C, as **uniões são um tipo de estrutura de dados que permite que mais que uma variável partilhe o mesmo endereço de memoria**. A declaração, tal como quase todas as operações de uma união, são idênticas às de uma estrutura. Vejamos a declaração:

```
union nome {
    tipo nome1;
    tipo nome2;
    ...
    tipo nomen;
    } lista_de_vars;
```

Geralmente, as **uniões** são utilizadas quando é necessário «mexer» com partes separadas de um tipo. Segue-se um exemplo de um inteiro que pode ser acedido através dos seus dois bytes separadamente:

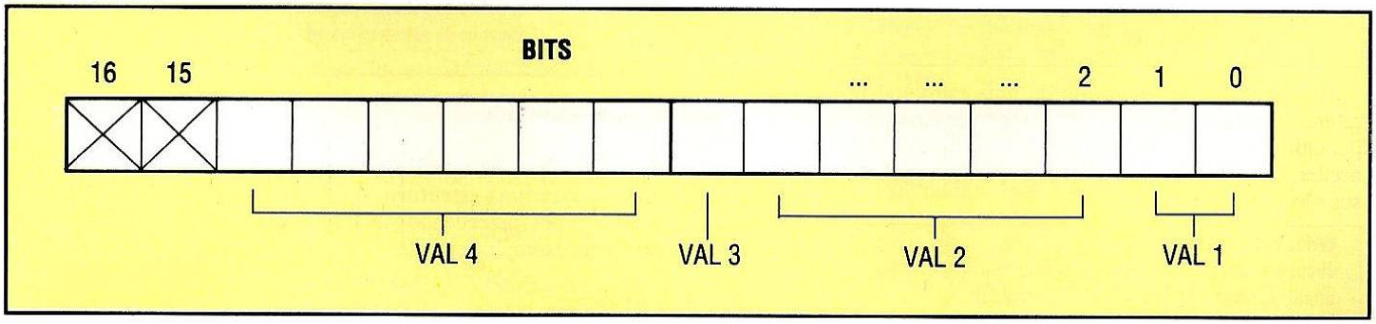
```
union {
    int i;
    char ch[2];
    } valor;

valor.i=100;
printf("byte mais significativo de valor.i=%i\n", valor.ch[2]);
printf("byte menos significativo de valor.i=%i\n", valor.ch[1]);
```



FUNÇÕES

As funções são um dos aspectos mais importantes de qualquer linguagem. Em C, tem um papel fundamental no desenvolvimento de programas estruturados e modulares. Básicamente, **uma função é um bloco de código que pode ser executado de vários locais de um mesmo programa.** Opcionalmente poderemos passar parâmetros para a função (valores com os quais a função vai operar). Depois de terminada a sua execução, poderemos opcionalmente retomar (ou devolver) um valor, que poderá ou não ser utilizado no local de onde a função foi «chamada».



Forma geral de uma função

Tal como as variáveis, as funções também têm de ser declaradas. A forma geral da declaração de uma função é a seguinte:

```
tipo_de_retorno nome_da_função(lista de parâmetros)
{
    corpo da função
}
```

tipo de retorno especifica o tipo de valor que a função irá retornar. Se não for especificado, será assumido por defeito o tipo int.

nome da função deverá ser um identificador válido.

lista de parâmetros é composta pelo tipo do parâmetro seguido do nome que lhe será associado; se existirem vários parâmetros, estes deverão ser separados por vírgulas. Caso não existam, deverá ser colocada a palavra-chave **void** entre os parêntesis. Falta analisar o **corpo da função**, o qual é sempre envolvido por chavetas "{...}". No seu início, poderão ser declaradas variáveis, que são «criadas» no início da execução da função e «destruídas» quando esta termina. Estas variáveis são **locais**, o que implica que só podem ser acedidas dentro da função em que foram declaradas. Como foi dito na primeira lição, se estas variáveis não forem inicializadas, o seu conteúdo será um valor qualquer. Depois de declaradas as variáveis, vem o código da função. Se o leitor seguiu as lições anteriores, já teve oportunidade de ver exemplos de funções, mais concretamente a função **main()**. Como o leitor também já deve saber, é nesta função que se inicia a execução do programa. Segue-se um programa exemplo que utiliza uma função para somar dois inteiros:

```
#include <stdio.h>

soma(int a, int b) /* retoma um int */
{
    return a+b;    /* retoma a+b */
}

void main(void)
{
    int n1, n2;

    printf("\nintroduza dois números: ");
    scanf("%i %i", &n1, &n2);
    printf("\noresultado de %i+%i e %i", n1, n2, soma(n1, n2));
}
```

O valor de retorno de uma função

A palavra-chave **return** é utilizada para terminar a execução de uma função e pode ser colocada em qulquer local do código de uma função. Se a função for do tipo **void** (sem valor de retorno) o **return** poderá ser desprezado, a não ser que exista a necessidade de colocá-lo antes do fim do código da função (return;). No caso da função não ser do tipo void, deverá sempre existir um return **xxx**; onde **xxx** é do tipo da função.

Ignorando o valor de retorno de uma função

Esta é, sem dúvida, mais uma característica interessante desta linguagem. Pois em C, **o valor de retorno de uma**

“A palavra-chave return é utilizada para terminar a execução de uma função e pode ser colocada em qulquer local do código de uma função”

SOLICITE-NOS VERSÃO DE DEMONSTRAÇÃO GRÁTIS

VZ Programmer

Tão simples como A-B-C/C++

Programar para o Windows com Facilidade e Rapidez

A. Desenhar

B. Escrever

C. Utilizar



O VZ Programmer utiliza a linguagem C/C++ de modo a permitir elaborar aplicações profissionais para o Windows, visualizando-as. Utilizando o método VZ os seus programas podem ser elaborados em apenas algumas horas, em vez das várias semanas ou meses que eram necessários.

O VZ Programmer™ ajuda os programadores a desenvolver aplicações e interfaces para o Windows através do seu simples desenho; a aplicação final é também transportável através de plataformas Windows, DOS ou OS/2 PM. O VZ Programmer™ também faculta:

- Ferramentas pré-definidas para gráficos e controlo de objectos.
- Construção de GUI (Graphics User Interface) em modo *Drag/Drop*.
- *Clipboard* de capacidade total para objectos.
- Suporte de *Clipboard* para texto, *bitmap* e GUI.
- Agrupamentos hierárquicos.
- Ligação prática por objectos.
- Base de dados, por objectos, incluída.
- Suporte para menús.
- Compilador, depurador, leitor e ligador integrados.
- Linguagem ANSI C/C++.
- Reutilização de código de alto nível.
- Suporte para bibliotecas de C.
- Suporta várias versões de Windows
- Integração total de dados e funções.
- Manipulação de mensagens e eventos.
- Ambiente poderoso para manipulação de objectos.
- Plataforma de compatibilidade entre Windows e OS/2 PM.
- Gestão de relações entre objectos.
- Sistema de gestão de memória.
- Fácil aplicação, conservação e possibilidade de extensão.

Distribuidor:

CIPEVE Lda

Rua Vasques de Mesquita, 259 - 4º C – 4300 PORTO
Telef.: (02) 49-6551 – Fax: (02) 410-6327

VZ CORP

VZ Programmer, Windows, DOS e OS/2 são marcas registadas

função pode ser ignorado. Repare no seguinte exemplo:

```
#include <stdio.h>

void main(void)
{
    char ch;

    printf("\nPressione uma tecla ...");
    ch=getch(); /* coloca em ch o valor de retorno */
    printf("\nO valor ASCII da tecla pressionada e %i", ch);
    printf("\nAgora pressione uma tecla qualquer para terminar ...");
    getch();    /* o retorno e ignorado */
}
```

Os protótipos são a forma de definir uma função sem colocar o seu «corpo» no mesmo local

Protótipos de funções

Os protótipos são a forma de definir uma função sem colocar o seu «corpo» no mesmo local. Básicamente, são uma forma de «chamar» uma função sem o seu código estar definido: basta o **protótipo**. Como o leitor já deve ter reparado, todos os programas que usam a função **printf** incluem no seu início o ficheiro (*header*) **stdio.h**. Este contém protótipos, entre outras, da função **printf**. Um protótipo difere da declaração de uma função no bloco de código ({...}) o qual, no protótipo, é substituído por um ";". Vejamos um exemplo:

```
/* define protótipos */
void a(void);
void b(void);

void a(void)
{
 ...
}

void main(void)
{
 ...
    a(); /* não necessita de protótipo */
    b(); /* necessita de protótipo */
 ...
}

/* como a função b é chamada antes de ser */
/* definida, necessita de protótipo */
void b()
{
 ...
}
```

Recursividade

Uma forma simples de definir **recursividade**,será: **a forma que uma função tem de se chamar a ela própria**, isto é, algures no código da função existe uma chamada a ela própria. Claro que este «mecanismo» tem de ter um fim, normalmente imposto por uma condição que impede que a recursividade seja infinita. Este tipo de algoritmo é normalmente utilizado para simplificar os programas. De seguida vamos ver um exemplo muito comum de um algoritmo recursivo, o cálculo do factorial de um número. Para quem não sabe o factorial de um número, este define-se da seguinte forma:

```
factorial(n)=1*2*3*....*n

factorial(5)=1*2*3*4*5
factorial(3)=1*2*3
fat(int n)
{
   if (n==1) return 1;
  return fat(n-1)*n;
}
```

A versão não recursiva deste algoritmo poderia ser como se segue:

```
fat(int n)
{
  int i, resultado=1;

  for (i=2; i<=n; i++)
  resultado=resultado*i;

  return resultado;
}
```

Como facilmente se percebe, a versão recursiva é bem mais simples. No entanto é necessário tomar cuidado, pois as funções guardam o endereço de retorno e as variáveis locais no *stack* que tem um comprimento limitado. Assim, se uma função se chamar muitas vezes e/ou o espaço ocupado por variáveis locais for muito grande, os limites do *stack* podem ser excedidos, o que poderá provocar resultados imprevisíveis. Uma forma eficaz de impedir esta situação, é não utilizar muitas variáveis locais nem muito grandes, e limitar o número de vezes que a recursividade é feita.

Alterando o valor de um parâmetro de uma função

Em C não é possível passar parâmetros por referência, só por valor. No entanto existe uma forma de alterar o valor do parâmetro. O «truque» reside em passar um ponteiro para a variável a alterar, sendo assim possível alterar a variável de forma indirecta (através de um ponteiro). Segue-se um exemplo:

```
#include <stdio.h>

void soma(int a, int b, int * resultado)
{
    *resultado=a+b;
}

void main(void)
{
    int r;

    soma(5, 23, &r);
    printf("\nO resultado e %d", r);
}
```

Os parâmetros e valores de retorno da função *main*

A função **main**, embora seja uma função especial, também pode ter parâmetros e valores de retorno. Os parâmetros desta função estão relacionados com valores passados pelo sistema operativo para o programa; o valor de retorno é um valor que é devolvido para o sistema operativo (S.O.) no final da execução do programa. Básicamente, esta função pode tomar quatro formatos possíveis:

```
/* sem retorno para o S.O. */
void main(void);                    /* sem parâmetros */
void main(int argc, char * args[]); /* com parâmetros */

/* com retorno para o S.O. */
main(void);                         /* sem parâmetros */
main(int argc, char * args[]);      /* com parâmetros */
```

O valor de retorno:

A função main retorna um inteiro para o processo que «mandou» executar o programa (normalmente o sistema operativo). Se esta for definida sem retorno, o valor devolvido tanto poderá ser **zero** como um valor imprevisível, dependendo do compilador. No caso específico do sistema operativo MS-DOS, este valor de retorno poderá ser analisado através da variável de sistema **ERRORLEVEL.** No que diz respeito aos parâmetros da função main, são dois: **argc** que contém o número de parâmetros da linha de comandos e **args** que é um array de ponteiros para os parâmetros da linha de comandos. A primeira posição deste array (*args[0]) aponta para uma cadeira de caracteres que contém o nome do programa. A segunda (*args[1]) aponta para o primeiro parâmetro e assim por diante. Imagine que executava um programa feito em C no *prompt* do MS-DOS, da seguinte forma:

```
C:\>abc spooler magazine 123experiencia
```

os parâmetros da função main seriam preenchidos da seguinte forma:

```
argc = 3
*args[0] = "C:\ABC.EXE"
*args[1] = "spooler"
*args[2] = "magazine"
*args[3] = "123experiencia"
```

Número de parâmetros variável

Em C, ao contrário de grande parte das linguagens de programação, é possível passar um número variável de parâmetros para uma função. Um exemplo muito comum é a função **printf**. Para informar o compilador que a função vai receber um número variável de parâmetros, colocam-se três pontos seguidos do último parâmetro obrigatório. Por exemplo:

```
func_a(int x, int y, ...); /* 2 obrigatários */
```

Acedendo aos parâmetros «fantasma»

Para aceder aos parâmetros opcionais de uma função é necessário incluir no início do programa o ficheiro (*header*) **stdarg.h**. Este contém definidas três *macros* e um tipo:

va_list tipo que aponta para os argumentos;
va_start(va_list ptr, último_arg) inicializa **ptr**, fazendo-o apontar para o primeiro parâmetro opcional;
tipo va_arg(va_list ptr, tipo) retoma o valor do parâmetro apontado por **ptr** que é do tipo **tipo**. **ptr** é incrementado de modo a apontar para o próximo parâmetro;
va_end(va_list ptr) deverá ser sempre executada quando não existirem mais parâmetros para processar.

Segue-se um exemplo de uma função que soma inteiros, com a característica de aceitar um número de parcelas variável:

```
#include <stdarg.h>
#include <stdio.h>

/* a última parcela deve ser zero */
soma(int primeira, ...)
{
    va_list p;
    int valor, sub_total;
    sub_total=primeira;
    va_start(p, primeira);
    while ((valor=va_arg(p, int)) != 0)
        sub_total+=valor;
    return sub_total;
}

void main(void)
{
    printf("\nO total e %i \n", soma(44, 56, 28, 566, 344, -23, 0));
}
```

AS FUNÇÕES DA STANDARD LIBRARY

Para além das funções **printf**, **scanf** e **getch**, já vistas em exemplos anteriores, existem muitas outras definidas na Standard Library. Por economia de espaço não vamos descrever todas elas em pormenor; no entanto o leitor pode ficar descansado, pois as mais importantes serão descritas. Para podermos executar funções da Standard Library necessitamos dos seus protótipos, os quais estão distribuídos por vários ficheiros, normalmente chamados de **headers**. Estes podem ser incluídos o número de vezes que pretendermos e pela ordem que quisermos.

Segue-se uma descrição dos **headers** mais importantes, acompanhada das declarações neles efectuadas:

– **<stdio.h>** declarações básicas de I/O

Todas as funções referentes a operações com ficheiros pertencentes a este *header* serão estudadas na próxima lição quando os ficheiros forem abordados. **stdin** representa o *standard input* (teclado) e **stdout** representa o *standard output* (ecrã).

int printf(char *format, ...) estudada na primeira lição.
int sprintf(char *s, char *format, ...) igual a printf, com a diferença que que em vez de escrever no stdout, coloca no array **s** (string)
int scanf(char *format, ...) estudada na primeira lição
int sscanf(char *s, char *format, ...) igual a scanf com a diferença que em vez de ler do stdout, lê da string **s**
int getchar(void) lê um carácter do stdin
char *gets(char *s) lê caracteres do stdin até aparecer um ENTER ('\n'). Os caracteres são colocados na string **s**. Retoma **s** ou então NULL se ocorrer erro.
int putchar(int c) envia um carácter no stdout. Retoma **c** se não houver erro, caso contrario retoma EOF
int puts(char *s) envia a string **s** para o stdout. Retoma EOF se ocorrer erro.

– **<ctype.h>** funções de teste a caracteres

Todas as funções deste módulo retomam um int. No caso das primeiras sete retoma **um** se a condição imposta for verdadeira, caso contrario retoma **zero**.

isalnum(c)	se for uma letra ou um dígito
isalpha(c)	se for letra
iscntrl(c)	se for um carácter de controlo
isdigit(c)	se for um dígito decimal
islower(c)	se for letra minúscula
isupper(c)	se for letra maiúscula
isxdigit(c)	se for um dígito hexadecimal
tolower(c)	converte **c** para minúsculas
toupper(c)	converte **c** para maiúsculas

Em C, ao contrário de grande parte das linguagens de programação, é possível passar um número variável de parâmetros para uma função

– **<string.h>** funções para operar com cadeias

char *strcpy(char *d, char *s) copia a string **s** para **d**
char *strcat(char *d, char *s) adiciona **s** no fim de **d**
int strcmp(char *s1, char *s2) compara as duas strings. O valor de retorno será: **<0** se **s1<s2**; **0** se **s1==s2** e **>0** se **s1>s2**
char *strchr(char *s, char c) retoma um ponteiro para a primeira ocorrência de **c** em **s** ou NULL se **c** não existir em **s**
char *strstr(char *s1, char *s2) retoma um ponteiro para a primeira ocorrência da cadeia **s2** em **s1** ou retoma NULL se **s2** não existir em **s1**
strlen(char *s) retoma o número de caracteres da cadeia **s**

– **<math.h>** funções matemáticas

Nas funções seguintes **x** e **y** são do tipo **double**, **n** é do tipo **int** e todas as funções retomam um **double**. Os ângulos são expressos em radianos.

sin(x)	seno de x
cos(x)	coseno de x
tan(x)	tangente de x
exp(x)	exponencial de x
log(x)	logaritmo natural de x
log10(x)	logaritmo (base 10) de x
pow(x, y)	x elevado a y
sqrt(x)	raiz quadrada de x
ceil(x)	arredonda x para cima
floor(x)	arredonda para baixo
fabs(x)	valor absoluto de x
fmod(x, y)	resto da divisão de x por y

– **<stdlib.h>** funções utilitárias em geral

int atoi(char *s) retoma o inteiro contido na cadeira **s**
int rand(void) retoma um número (int) pseudo-aleatório
void srand(unsigned int n) reinicializa o gerador de números pseudo-aleatórios tendo como base **n**
void *malloc(tamanho) retoma um ponteiro para um bloco de memória de tamanho **tamanho,** ou NULL se não existir memória suficiente
void free(void *p) liberta um bloco de memória previamente alocado por malloc
void exit(int ret) termina a execução do programa, retomando **ret** para o sistema operativo
int abs(int n) retoma o valor absoluto de **n**

PROGRAMA EXEMPLO – VERSÃO 3

Esta versão da nossa calculadora já é bem mais complicada que a anterior. Foi introduzido um **interpretador de expressões** que utiliza um algoritmo recursivo para processar a expressão.

De momento, os operadores que este interpretador «conhece» são: **+, -, *, /.** É também permitido o uso de parêntesis. A fonte da Versão 3 da calculadora poderá ser encontrada no ficheiro **CALC2.C.**

NA PRÓXIMA LIÇÃO

Os tópicos da próxima lição são os seguintes:

- Tipos enumerados
- Declaração de tipos pelo programador
- Conversão de tipos
- Classes de variáveis
- Directivas do compilador
- Ficheiros
- Versão 4 da calculadora

Carlos Ladeira

Prolab®

Your Peripheral Partner

SCANNER 256 TONS DE CINZENTO PS-4000G

O PS-4000G DA PROLAB OFERECE UMA SOLUÇÃO PROFESSIONAL PARA DESKTOP PUBLISHING (DTP) E PARA OPTICAL CARACTER RECOGNITION (OCR)
- CLIPBOARD, Setembro 91 -



- Resolução 100 - 400 DPI.
- Para PC AT, 386 / 486.
- 256 tons de cinzento.
- 32 KB de FIFO inteligente na placa.
- Software c/ Auto Merge.
- OCR para 11 linguagens.
- Software - PROIMAGE para DOS e WINDOWS, - PROREADER para OCR.

SCANNER A CORES PS-4200C

MICROSOFT. WINDOWS.

SE PROCURA COR DE ALTA DEFINIÇÃO, RECOMENDAMOS O PS-4200C DA PROLAB PELO SOFTWARE DE QUALIDADE
- Personal Computer Magazine, Junho 91 -



- Resolução 100 - 400 DPI.
- Para PC AT, 386 / 486.
- 256 / 4096 cores.
- 32 KB de FIFO inteligente na placa.
- Software c/ Auto Merge.
- OCR para 11 linguagens.
- Software - PROIMAGE para DOS e WINDOWS, - PROREADER para OCR.

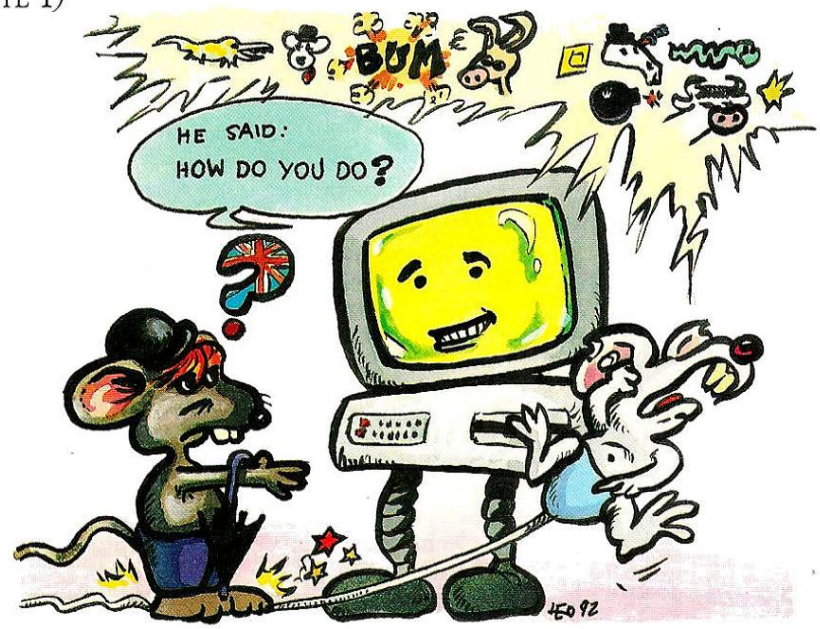
REPRESENTANTE EXCLUSIVO:
GENAFER EQUIPAMENTOS ELECTRÓNICOS, LDA.
Quinta D. João • Lote 1 - r/c Esq. • 3000 COIMBRA
Telefones: 401310 - 404555/61 • Fax: 404549
Travessa Senhor Matosinhos, 75 • 4400 GAIA
Telefone: 7124229 - 7124449 • Fax:7124528

☐ Pretendo que me enviem mais informações sobre PS-4000G
☐ Pretendo que me enviem mais informações sobre PS-4200C
☐ Pretendo que me indiquem o v. agente mais próximo.
☐ Pretendo ser contactado pelo v. delegado de vendas.

Empresa: _____
Nome: _____ Cargo: _____
Endereço: _____
Localidade: _____ Código Postal: _____
Telefone: _____ Fax: _____

TRADUTOR

(PARTE I)



Este programa, concebido em linguagem *Assembly*, é um TSR *(Terminate and Stay Resident)*. Para quem não esteja familiarizado com estes anglicismos, digamos que se trata de um programa que, depois de executado, fica em «hibernação» na memória do processador, num sono de «bela adormecida» aguardando a beijoca despertadora do «príncipe encantado».

Na origem deste utilitário esteve uma «encomenda». O nosso amigo, Renato Casquilho, vinha pacientemente, há algum tempo, organizando um dicionário inglês/português utilizando, para o efeito, os recursos do dBASE. Esse dicionário, já com alguns milhares de termos, tinha o inconveniente de só poder ser consultado em separado, ou seja, havia necessidade de abandonar o trabalho em curso, accionar o dBASE para consultar ou acrescentar determinado termo, fechar o dBASE e regressar ao trabalho anterior. Uma maçada!

Vai daí, nasce a ideia de fazer em *Assembly* um programa residente que permitisse a consulta rápida *on line* de qualquer palavra, sem os inconvenientes atrás descritos.

Já agora, esse programa deveria permitir a consulta de várias bases de dados (dicionários), proporcionar a introdução de novos termos, modificação de termos existentes, indexação alfabética, criação de novos dicionários, etc., etc. e etc.

Em abono da verdade tudo isso (e muito mais) está disponível nesta primeira versão do TRADUTOR. O que talvez lhes custe a acreditar é que desde a ideia inicial até aos «alto, basta, já chega!» finais foram gastas cerca de 700 horas de trabalho, 20 litros de café, dezenas de quilowats, 1 resma de papel contínuo e outros consumíveis diversos tais como lápis, esferográficas, bloco de apontamentos, disquetas...

ANATOMIA DO TRADUTOR

Se o leitor pertence ao grupo dos que não têm paciência, tempo ou não gostam destas «assemblices» então faça já o seguinte: copie para o seu disco rígido o ficheiro compactado TRADUTOR.EXE, descompacte-o, apague os ficheiros com a extensão .ASM que só lhe ocupam espaço e execute o ficheiro TRADUTOR.COM. Depois de instalar o programa (para isso basta escrever TRADUTOR e teclar ENTER), ser-lhe-á indicado que deve utilizar a tecla <F1> para activar o programa residente. Abra e leia os textos das «janelinhas». Verá como é fácil criar os seus próprios dicionários ou acrescentar/modificar aqueles que acompanham este programa(*). O próprio programa fornece todos os esclarecimentos que necessitar. Inté.

(...)

Agora que despachámos os não «assembliófilos», vamos calmamente falar daquilo que nos trouxe aqui: um utilitário integralmente realizado em linguagem *assembly* e que (presunção e água-benta...) contém (até prova em contrário) rotinas vanguardistas e inéditas!

Como é meu hábito (mania), utilizei o «assemblador» A86 versão 3.18 de Eric Isaacson. O que, obviamente, significa que os ficheiros-fonte foram escritos respeitando as particulariedades próprias daquele utilitário. As fontes estão divididas em duas partes: TRADUTOR.ASM e TRADUTIR.ASM. Para «linkar» tudo basta comandar - A86 TRADUTOR.ASM TRADUTIR.ASM (enter). Num abrir e fechar de olhos é gerado o ficheiro TRADUTOR.COM.

«OVO DE COLOMBO»?

Logo a seguir à ideia de produzir este programa comecei a confrontar-me com algumas dificuldades. Todas as rotinas TSR que estudei (e que pululam no *shareware* das BBSs) eram demasiado específicas, básicas e limitativas para aquilo que pretendia fazer.

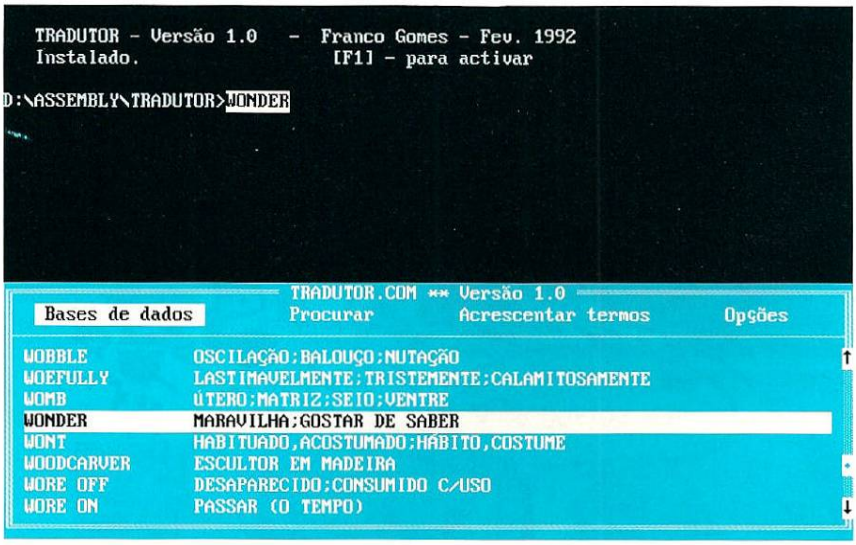
Um programa residente é, geralmente, activado por uma interrupção e, ainda mais vulgarmente, essa interrupção é activada pelo utilizador ao accionar o teclado do seu computador.

Quando é invocada uma interrupção, o processador desvia a sua atenção daquilo que estava a fazer para realizar a tarefa «urgente» que lhe foi solicitada por essa interrupção. Usando analogias domésticas, digamos que o processador que estava tranquilamente a jantar com os

---

*(*) **Nota da Direcção**: O Franco falou de um dicionário Inglês-Português que estava a ser «pacientemente» realizado por este vosso amigo. Em abono da verdade, logo que instalei o TRADUTOR na minha máquina «não quiz outra coisa», que é como quem diz, passou a tornar-se insuportável interromper o trabalho, chamar o dBASE e executar o meu DICING.PRG. A partir daí fui introduzindo novos termos e significados e corrigindo ou aumentando outros. Hoje, o ficheiro (ou base de dados) que é lido pelo TRADUTOR está bastante completo. Só que, mesmo comprimido, não cabia na disqueta da **Spooler** (como vem sendo hábito). Desta forma decidiu-se «criar» uma disqueta especial na série **shareware-freeware**, contendo o dicionário de Inglês-Português na sua fase actual, um pequeno dicionário de Francês-Português e um dicionário de siglas utilizadas na informática. A disqueta poderá ser adquirida nos nossos balcões ou encomendada pelo correio e o seu conteúdo é do domínio público. Os meus parabéns ao Franco Gomes pelo seu excelente trabalho.*

R.Casquilho



seus periféricos, foi interrompido pela campaínha da porta.

Tratava-se do cobrador dos «Vigilantes da Noite dos PCs», ou similar que, como todos sabemos, tem o «condão» de interromper as refeições às pessoas, digo, processadores .

Depois de paga a quota mensal, o processador retoma a sua refeição na exacta «garfada que levava à boca» no momento da interrupção.

O que se passa, e que constitui um verdadeiro problema, é que, quando se está a atender uma interrupção não nos é possível atender qualquer outra interrupção que ocorra dentro daquela, uma vez que o processador se encontra fora do seu circuito habitual de funcionamento.

Quase todas as rotinas que observei contornavam este problema simulando e reescrevendo rotinas substitutas para as interrupções que não poderiam ser executadas.

Um desses programas até afirmava pomposamente que funcionava até com as funções da Interrupção 21h do DOS! Pessoalmente não consegui utilizar nenhuma!

O que eu pretendia para o meu TSR era poder usar e abusar de todas as interrupções disponíveis (se necessário), sem ter que reescrever rotinas complicadíssimas tais como aquelas que tornam possíveis as operações de leitura/escrita em disco.

A Preguiça que é tida como a «mãe de todos os vícios» algumas vezes é, também, a matriarca de algumas invenções e inovações que resultam em maior comodidade para os humanos. Os telecomandos dos nossos aparelhos audiovisuais foram, decerto, obra desta senhora, evitando o «desgaste» que provoca a movimentação do sofá ao aparelho para mudar o canal ou executar qualquer outra função.

Disposto a investir o meu trabalho de maneira a evitar futuras canseiras, debrucei-me com energia sobre o assunto.

Pouco a pouco, a ideia foi-se materializando até assumir uma fulgurante e tão evidente configuração que ainda me custa acreditar ter sido o primeiro a pensar nisso (será que fui?).

Tente, estimado leitor, seguir o fio do raciocínio que a seguir desenrolo:

Uma interrupção (vulgo *interrupt)* quando solicitada ao processador provoca imediatamente as seguintes operações:

```
PUSHF
PUSH CS
PUSH IP
CALL FAR (INT*4)
```

Com **PUSHF** são guardadas na «pilha» os *flags*, **PUSH CS** guarda o *Code Segment* corrente, **PUSH IP** (esta instrução não existe) guarda o endereço da instrução que estava a ser executada e o **CALL FAR** chama a rotina de acordo com o número da interrupção invocada. Quando terminada a tarefa solicitada pela interrupção, o processador retoma as operações no exacto ponto onde as deixou com a lacónica instrução:

```
IRET
```

Que significa retorno da interrupção e que na prática corresponde às seguintes operações:

```
POP IP
POP CS
POPF
RET
```

que me abstenho de comentar dado o carácter óbvio das mesmas.

Resumindo: quando o processador recebe a solicitação **INT**, guarda na «pilha» o endereço e estado dos *flags* do programa que estava a executar no momento da interrupção, salta e executa a rotina que lhe foi «encomendada» e retorna à posição inicial depois de reaver o endereço e estado original dos *flags*. A instrução **IRET** comunica ao controlador de interrupções que o processador pode operar normalmente, uma vez que se encontra no seu posto habitual de trabalho.

Depois desta análise, conclui-se que o que faz falta (além de animar a malta) é qualquer coisa que, depois de passar ao processador uma instrução **IRET**, continue (como quem não quer a coisa) a funcionar no interior do nosso residente.

O grande truque (?...) é substituir o endereço «legal» de retorno da interrupção por um outro endereço – dentro do nosso programa residente – de modo que o processador retorne da interrupção mantendo-se, ainda e sempre, dentro da nossa rotina. Para o efeito basta fazer uns POPs seguidos duns PUSHs estratégicos.

Poderemos então usar todas e quaisquer interrupções possíveis, incluindo aquela que inicialmente desviámos para activar o nosso programa residente!

Eu disse todas e quaisquer interrupções?!... Peço desculpa. De facto, uma minoria de irredutíveis ainda resiste e resistirá ao invasor (onde é que li isto?...). Os energrumos, perdão, energumeros, ener... (hum!), os «coisos» são meia-dúzia de funções do INT21h (funções do DOS) como, por exemplo, a função 9 e congéneres que teimam em dar barraca e a escapar à minha compreensão.

Declarei guerra a esta situação e, cada vez que posso, retomo a luta, «debugando» o grande quebra-cabeças que representa o conjunto de rotinas do DOS.

De qualquer modo, *chez* TRADUTOR a manipulação da imagem está entregue a rotinas de fabrico próprio que vêm sendo retocadas ao longo dos tempos e que tornam obsoletas as irredutíveis funções mencionadas.

INGREDIENTES BÁSICOS

Embora correndo o risco de o «chatear» com coisas que já esqueceu, segue-se a listagem básica *(Figura 1)* que elaborei para instalar em memória o TRADUTOR e que pode encontrar no fim de TRADUTIR.ASM. Aqui resumida e comentada para melhor compreensão de todo o processo:

; INSTALAÇÃO DO PROGRAMA RESIDENTE

```
START: PUSH ES          ; Empilha-se o registo ES que vai ser afectado.
       MOV AX,03516h    ; Obtém-se o endereço da rotina da ROM que
                        ; é invocada pela interrupção 16h
       INT 33           ; (operações relativas ao teclado).
                        ; Usou-se a função 35h da Interrupção 21h
                        ; (33 em decimal) do DOS para obter esta
                        ; informação.
       MOV SI,BX        ; Recupera-se o registo ES depois
       MOV BX,ES        ; de copiar o valor deste para BX.
       POP ES
       PUSH DS          ; Aponta-se para os dois bytes
       SUB SI,2         ; imediatamente anteriores ao endereço em
                        ; BX:SI.
       MOV DS,BX
       LODSW
       POP DS
       CMP AX,'FG'      ; Se a word aí situada corresponder à nossa
                        ; «assinatura» é porque o programa já se
                        ; encontrava instalado.
       JNZ INSTL        ; Caso contrário, instala-se o residente
                        ; saltando para a etiqueta «INSTL».
       MOV AH,9         ; Escreve-se uma mensagem no ecrã
                        ; lembrando ao utilizador que o programa já
       MOV DX,TXT_2     ; estava instalado (por enquanto
       INT 33           ; ainda podemos usar esta função).

       MOV AX,04C00h    ; Retorno.
       INT 33

INSTL:                  ; Segue-se a rotina para instalar o residente
                        ; em memória:
L0:    PUSH ES          ; Guarda-se o registo ES que vai
                        ; ser afectado.
       MOV AX,03516h    ; Obtém-se o endereço da
                        ; Interrupção 16h da ROM.
       INT 33           ; Esta interrupção (como já disse)
                        ; dedica-se às operações relacionadas com o
                        ; teclado.
       MOV [OLD16],BX   ; O endereço do vector é fornecido
       MOV [OLD16+2],ES ; no par de registos ES:BX que
                        ; arquivamos de imediato
       POP ES           ; dentro do nosso programa num endereço de
                        ; 4 bytes a que chamámos «OLD16».
       MOV [ASSIN],'FG' ; Marca-se com uma «assinatura» a posição
            ; «ASSIN» o que nos permite verificar posteriormente se
            ; o programa já se encontra, ou não, instalado em
            ; memória. A etiqueta «ASSIN» deve situar-se
            ; imediatamente antes da etiqueta «NOV16».
       MOV AH,9
       MOV DX,TXT_1     ; Escreve-se no ecrã uma mensagem a
                        ; confirmar a instalação.
       INT 33
       MOV AX,02516h    ; Substituimos o endereço do INT16h
       MOV DX,NOV16     ; pelo nosso: «NOV16».
       INT 33
       MOV DX,START     ; DX especifica o endereço onde
            ; poderá carregar-se o próximo programa. No nosso caso
            ; poderá ser a partir de «START» uma vez que este
            ; bloco se encontra no fim do programa e a rotina aí
            ; iniciada já não vai ser usada.
       INT 39           ; Termina e permanece residente em memória.
```

Figura 1

Depois de executada a rotina «INSTL» da listagem anterior e, cada vez que seja solicitada uma operação relacionada com o teclado, o processador passará a executar a nossa rotina «NOV16» em vez da tradicional rotina da ROM BIOS cujo endereço preservámos na nossa etiqueta «OLD16».

No «dialecto» dos TSRs costuma designar-se por *HotKey* a tecla (ou conjunto de teclas) que activa o respectivo programa. É evidente que a primeira coisa a fazer na nossa rotina «NOV16» é verificar se a tecla premida (ou grupo de teclas) corresponde ao nosso *HotKey*, caso contrário o melhor é reendereçar o processador para a rotina original da ROM.

Também retirada de TRADUTOR.ASM, veja-se a listagem da figura 2.

; SERÁ O HOTKEY?

```
NOV16: CMP AH,0         ; Vê-se qual a função (em AH) que foi usada
                        ; quando invocada a interrupção 16h
       JZ >L0
       CMP AH,16        ; Só nos interessam as funções AH=0 ou AH=16
       JZ >L0           ; que correspondem à leitura do teclado, caso
                        ; contrário salta-se, sem demora para a «velha»
                        ; rotina da ROM cujo endereço completo
                        ; havíamos guardado na etiqueta «OLD16».
       JMP FAR D[CS:OLD16]
L0:    STI              ; Simula-se uma interrupção:
       PUSHF            ; Empilham-se as «flags»,
       CLI              ; desligam-se as interrupções
       CALL FAR D[CS:OLD16]  ; e chama-se outra vez o «velho» Int16h
                        ; para fazer a parte «chata» do trabalho.
       CMP AX,[CS:HOTKY] ; AX retorna com o scan code
       JZ >L1           ; da tecla(s) premida(s).
                        ; Compara-se com a nossa HotKey.
       IRET             ; Se não for igual, a interrupção não passa daqui
                        ; e retorna sem fazer absolutamente nada.

L1:    MOV [CS:SAVBX],BX ; Foi accionada a nossa «Tecla Quente»!
       MOV [CS:SAVCX],CX ; Guardam-se todos os registos
                        ; cuidadosamente dentro do nosso
                        ; programa.
       MOV [CS:SAVDX],DX ; Só não se guarda o registo AX
       MOV [CS:SAVBP],BP
       MOV [CS:SAVDI],DI
       MOV [CS:SAVSI],SI
       MOV [CS:SAVDS],DS
       MOV [CS:SAVES],ES
       POP [CS:SAVIP]   ; «Rouba-se» à pilha o endereço
       POP [CS:SAVCS]   ; remetente da interrupção, que se
                        ; guarda em sítio seguro.
       POP [CS:FLAGS]   ; Guarda-se também o estado dos flags.
       MOV [CS:SAVSS],SS ; Não esquecer de guardar o endereço
       MOV [CS:SAVSP],SP ; da própria pilha.
                        ; (stack para os anglófonos).
       STI              ; Estas instruções proporcionam um
       CLD              ; certo à-vontade aos flags.
```

Figura 2 (continua na pág. seguinte)

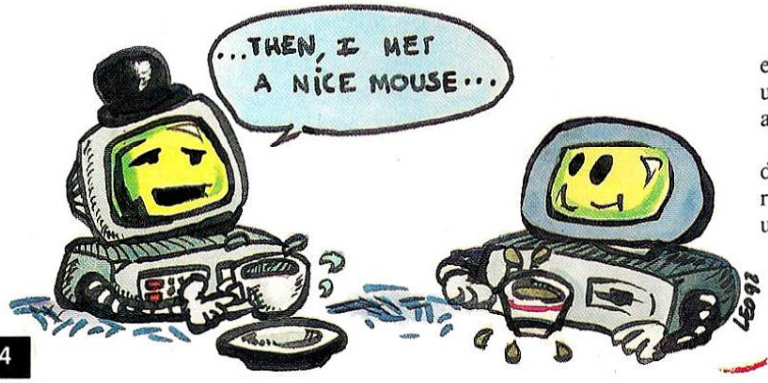
```
        PUSHF           ; Empilhamos os novos e descontraídos flags.
        MOV AX,CS       ; O AX vai ver qual o nosso Code Segment.
        PUSH AX         ; Empilhamos o nosso segmento de código e
                        ; offset para fazer uma partida ao IRET.
        MOV AX,VOLTA
        PUSH AX
        IRET                    ; E pronto! Em vez de voltar para onde
; devia, a interrupção termina, com as voltas trocadas, e a execução do
; processador vai retomar aqui, já a seguir na etiqueta «VOLTA».

VOLTA:  MOV AX,CS       ; Depois de lubridiado o novo Int16h, vamos
        MOV DS,AX       ; apontar todos os registos de segmento para
                        ; este nosso segmento.
        MOV ES,AX
        PUSH ES         ; verificamos se algum outro programa
                        ; residente foi posteriormente alojado em
                        ; memória desviando o nosso  INT16h.
        MOV AX,03516h   ; Obtém-se o end. do vector actual do Int16h
        INT 33          ; em ES:BX.
        MOV DX,ES
        POP ES
        CMP BX,NOV16    ; Se não é igual ao nosso INT16h é porque
        JNZ >L0         ; outro programa residente entretanto
                        ; o desviou.
        MOV AX,DS
        CMP DX,AX
        JNZ >L0
        MOV B[NAODA],0
        JMP >L1
L0:     MOV B[NAODA],1  ; Coloca-se a 1 o byte «NAODA» para assinalar
                        ; que, quando solicitada, não é possível a
                        ; desinstalação, caso contrário o outro
                        ; programa que foi instalado após este, ficará
                        ; «baralhado» de todo.
L1:     MOV W[ACT16+2],DX
        MOV W[ACT16],BX                 ; Guarda-se o corrente vector
do INT16h
                        ; (seja ele qual for...).
        PUSH DS
        MOV AX,02516h   ; Como já não precisamos de manter o
        LDS DX,[OLD16]  ; Int16h desviado (enganado já ele foi),
        INT 33          ; vamos repor em funcionamento a
        POP DS          ; velha interrupção de I/O do teclado.

        CALL ACCAO      ; Um estalo da claqueta e lá vamos...
                        ; ...
                        ; ...
                        ; ...
; O programa a iniciar em «ACCAO» poderá usar quase todas as
; interrupções como se se tratasse de um vulgar programa não residente.
; Pode, por exemplo, usar a própria interrupção 16h, que havia desviado,
; da forma usual.
                        ; ...
                        ; ...
                        ; ...
                        ; ... E já voltámos.
                        ;
        CMP AL,0        ; Se o valor 0 foi retornado em AL é porque
        JNZ >L2         ; o estimado utilizador resolveu desinstalar
                        ; o programa residente. Daqui se depreende
                        ; que o programa iniciado em «ACCAO»,
                        ; seja ele qual for, deve devolver no registo
                        ; AL o valor 0 para activar a desinstalação
                        ; do programa residente. Qualquer outro
                        ; valor em AL faz saltar para a etiqueta
«L2».
        MOV AH,73       ; Para tal, liberta-se a memória do segmento
        INT 33          ; ocupado por este programa (em ES).
        PUSH ES
        MOV ES,W[44]    ; Liberta-se também a memória ocupada
        MOV AH,73       ; pelo DOS Environment. Muitos residentes
        INT 33          ; não fazem isto, o que resulta ficarem
        POP ES          ; sempre uns bytes fantasmas a ocupar
                        ; espaço na dita cuja memória.
        JMP >L3         ; Siga-se o pequeno salto...
L2:     PUSH DS
        MOV AX,02516h   ; Aqui o respeitável utilizador desactivou
        LDS DX,[ACT16]  ; (sem desinstalar) o programa. Tornamos a
                        ; desviar o Int16h para a nossa rotina.
        INT 33
        POP DS

L3:     MOV BX,[SAVBX]  ; Vão buscar-se todos os registos
        MOV CX,[SAVCX]  ; anteriormente guardados com todos os
        MOV DX,[SAVDX]  ; cuidados.
        MOV BP,[SAVBP]
        MOV DI,[SAVDI]
        MOV SI,[SAVSI]
        MOV DS,[SAVDS]
        MOV ES,[CS:SAVES]
        MOV SS,[CS:SAVSS]
        MOV SP,[CS:SAVSP]
        MOV AX,[CS:FLAGS]
        PUSH AX
        MOV AX,[CS:SAVCS]
        PUSH AX                 ; Empilha-se o endereço para onde a
        MOV AX,[CS:SAVIP]       ; interrupção deveria ter voltado da
                                ; primeira vez...
        PUSH AX
        XOR AX,AX       ; Coloca-se o registo AX a zero.
        IRET            ; E záz! A interrupção volta sã e salva, como
                        ; se nada se tivesse passado. De facto,
; levou uma lavagem ao cérebro e jura que foi apanhada num disco
; voador...
```

Figura 2



Usando estas rotinas, poderá incluir qualquer programa a partir da etiqueta «ACCAO», que este funcionará sem complicações se evitar utilizar as funções 10 e menores do Int21h do DOS. Talvez o leitor me ajude a resolver esta questão.

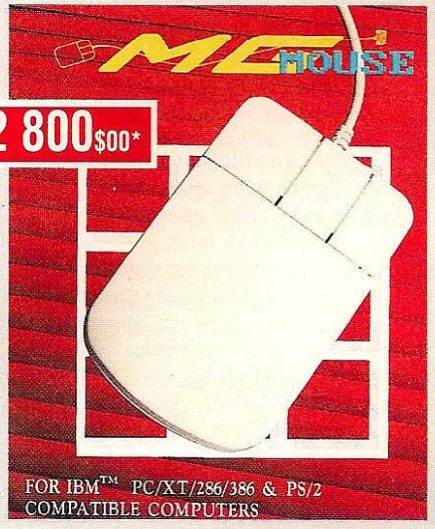
Epa!... Com a conversa perdi a noção das horas. Peço imensa desculpa, mas vou deixar para o próximo número os considerandos relativos ao programa TRADUTOR propriamente dito. Até lá façam um bom e proveitoso uso dele.

Franco Gomes

Porque o seu Computador Exige Acessórios da Mais Alta Qualidade

inves

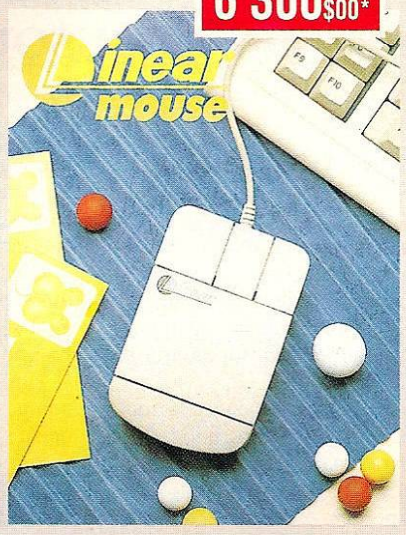
ACESSÓRIOS



2 800$00*

MG MOUSE

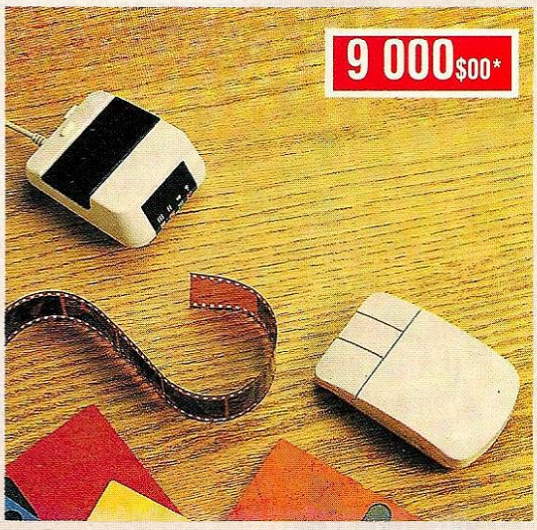
- 3 Botões
- Mouse Driver, Programa de Teste
- Modos Microsoft e Mouse System



6 300$00*

LINEAR MOUSE

- 4 Botões
- Adapt. 9-25 pinos
- Mouse Driver, Programa de Teste, Software Image 72
- Modos Microsoft e Mouse System
- Tapete



9 000$00*

INFRARED MOUSE

- 3 Botões
- Funcionamento por Infra-vermelhos (sem fio)
- Não usa Pilhas
- Bateria Interna com Carregador Próprio
- Adapt. 9-25 pinos
- Mouse Driver, Programa de Teste, Soft. Image 72
- Modos Microsoft e Mouse System
- Tapete



52 000$00*

ETHERNET LAN ADAPTER POCKET SIZE

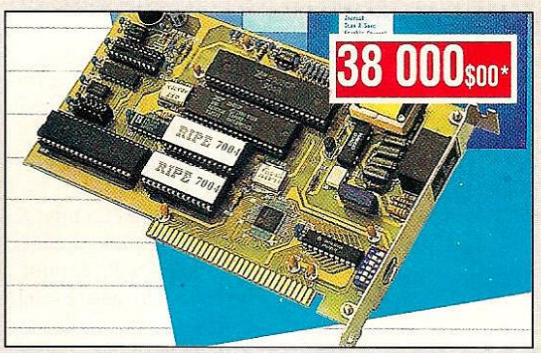
- Transferência de Dados: 10 Mbps
- 8 Kb RAM Buffer
- Conversor 10 BASE-2 BNC
- Instalação Externa – Porta Paralela



32 000$00*

DEMOKEY

- Conversor Sinal VGA para Vídeo
- Resolução 640x480, 256 cores
- Saída Modo PAL/NTSC
- Conectores de saída RCA e S-VHS
- Inclui Transformador de Alimentação, Cabos de Ligação, Manual e Software de Instalação.



38 000$00*

FAX/MODEM

- Fax: 9600/7200/4800/2400 bps
- Modem: 2400/1200/300 bps
- Software: Gestão Fax; Suporta Ficheiros PCX, PCC, TIF, MSP, GEM, CUT, FAX, TXT, IMG; Software Comunicações Modem Compatível Hayes
- Opções: Interno/Externo

CONDIÇÕES ESPECIAIS PARA REVENDA

REPRESENTANTE OFICIAL:

INVESMÁTICA – Investigação e Informática, Lda.
Rua Abel Botelho, 8-A – 1500 LISBOA – PORTUGAL
Telefs.: 351-1-7740880/7787324/7787326 – Fax: 351-1-7785301

TODOS ESTES PRODUTOS PODEM SER ENCOMENDADOS ATRAVÉS DA «MONTRA SPOOLER»

* TODOS OS PREÇOS APRESENTADOS ACRESCEM DE 16% DE IVA.

CONECTIVIDADE

TUDO O QUE VOCÊ SEMPRE QUIS SABER SOBRE AS «LÃS» (E NÃO OUSOU PERGUNTAR)

Uma rede local de computadores (LAN=*Local Area Network*) serve essencialmente para ligar PCs entre si

O QUE É E PARA QUE SERVE UMA REDE DE COMPUTADORES ?

Uma rede local de computadores (LAN=*Local Area Network*) serve essencialmente para ligar PCs entre si, independentemente destes se encontrarem na mesma sala ou em edifícios distantes. As necessidades de ligação surgem quando existe *software* ou *hardware* que tem de ser partilhado entre vários utilizadores.

A partilha de equipamento está normalmente relacionada com a impressora (possibilita que vários utilizadores enviem os seus trabalhos para uma mesma impressora) e com o acesso a discos de grande capacidade.

A partilha de programas permite reduzir a redundância de informação e garantir a sua consistência, porque possibilita que diferentes utilizadores acedam ao mesmo conjunto de informação em simultâneo.

Se considerarmos, por exemplo, uma empresa informatizada que efectue a sua gestão comercial através de um programa de computador, este programa deverá permitir que sejam efectuadas em paralelo diversas operações de encomenda, compra e venda de material, transferências de *stocks* entre armazéns, emissão de facturas, etc. Muito provavelmente, os utilizadores que originam estes movimentos estarão fisicamente distantes entre si (cada um no seu departamento) e, no final, tudo isto tem de bater certo (uf!).

Um programa desta envergadura só pode funcionar num sistema operativo multi-utilizador (UNIX, por exemplo) ou numa rede de computadores.

A ESCOLHA DE UMA REDE

Na escolha de uma rede é importante ponderar bem e acertar à primeira, porque o elevado investimento associado à instalação de uma LAN assim o exige. O aludido investimento não se refere apenas ao custo do equipamento e do *software* (onde, como em tudo na vida, existe o caro e o barato) mas também aos custos de formação dos utilizadores (a instalação de uma rede implica frequentemente uma restruturação total dos métodos de trabalho já implementados) e à necessidade de criação da figura do «administrador da rede», uma pessoa com sólida formação na àrea da informática que possa manter a rede em funcionamento sem grandes problemas, pois quando a dependência da informática é grande, se a informática não funciona, a empresa pára.

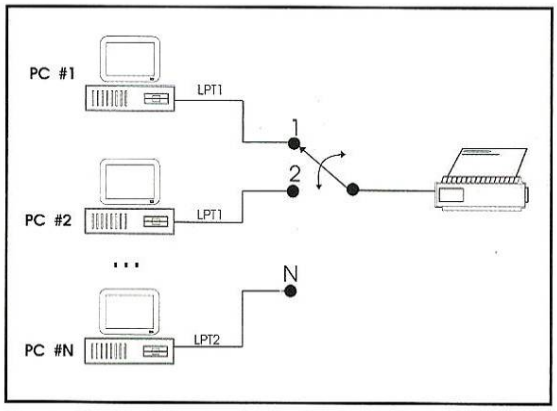
Antes de se tomar uma decisão desta envergadura, os prós (o aumento de produtividade) e os contras (investimento humano e material que pode ser elevado) devem ser cuidadosamente pesados para verificar se, em primeiro lugar, é indispensável a instalação de uma rede.

QUANDO É QUE UMA REDE NÃO É REALMENTE NECESSÁRIA?

Nem sempre as necessidades de interligação de computadores passam obrigatoriamente por uma rede. Quando o problema se resume a partilhar uma ou mais impressoras ou a transferir ficheiros entre utilizadores, TUDO se resolve sem recorrer a uma rede e gastando um DÉCIMO do que seria necessário para a instalar...

Quando a ideia é partilhar apenas uma dada impressora (ou impressoras) a solução apropriada é a aquisição de um *data-switch*, equipamento que custa apenas uma ínfima fracção do preço a pagar por uma rede.

Um *data-switch* não passa de um «PBX» com nome pomposo, onde as várias entradas podem ser ligadas a uma única saída; como é óbvio, apenas uma ligação pode ser feita de cada vez. Desta forma, vários computadores podem estar ligados a uma única impressora; no momento em que um utilizador deseja imprimir, estabelece a ligação entre a sua entrada e a saída.



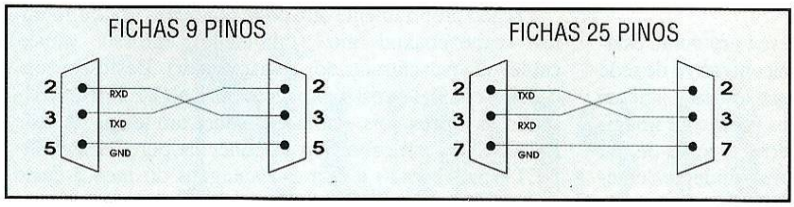
Esquema simplificado de um data-switch

Existem *data-switches* muito simples - um mero interruptor mecânico que liga uma das N entradas à saída - até aos *data-switches* inteligentes - estes últimos não necessitam de qualquer operação pois estão «atentos» ao estado das entradas e assim que detectam fluxo de informação, estabelecem automaticamente a ligação entre essa entrada e a saída.

Alguns *data-switches* especialmente concebidos para impressoras funcionam também como *printer-buffers*, ou seja, possuem memória RAM onde armazenam o trabalho a imprimir; assim que o PC termina o envio do trabalho para o *printer-buffer* fica imediatamente liberto da tarefa de impressão, encarregando-se então o *printer-buffer* de enviar para a impressora a totalidade do ficheiro, à velocidade que a impressora o permitir.

Outra facilidade que tipicamente se pretende implementar num escritório onde várias pessoas trabalham com computadores é a possibilidade de «file-transfer» ou, como diriamos em português, efectuar transferência de ficheiros entre computadores pessoais através de um cabo (algo que também pode ser conseguido com a cópia dos ficheiros para disquete e a circulação desta!).

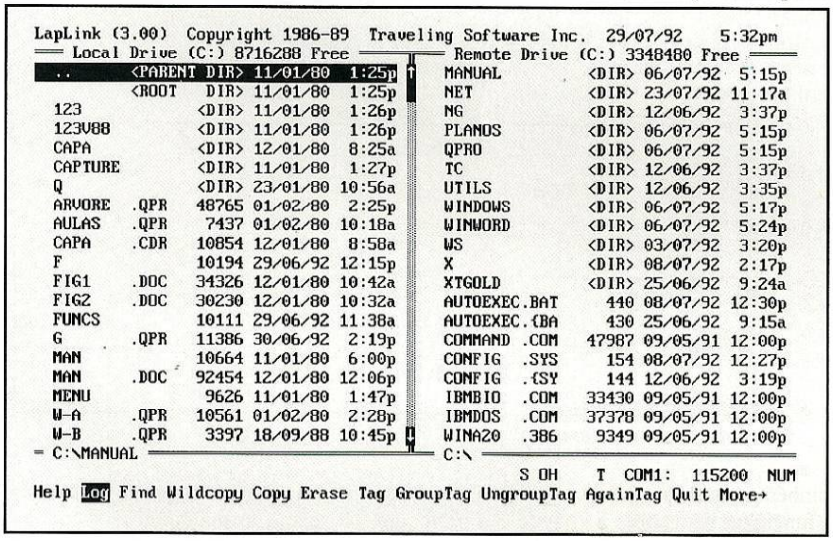
Se o problema a resolver for só este, existe uma solução simples.



O cabo de ligação para o LAP-LINK

A empresa Travelling *Software* concebeu um programa (LAPLINK) destinado a facilitar a transmissão de ficheiros entre um computador portátil e um computador fixo. Para ligar os dois computadores é necessário pô-los em contacto através do porto série (com um cabo) e correr o programa nos dois computadores.

Quando os computadores estão em comunicação, o ecrã de ambos fica dividido em duas janelas, representando a da esquerda o computador local e a da direita o computador remoto.



O ecrã do LAP-LINK

O conteúdo janelas é a lista de ficheiros da directoria corrente de cada computador. Os cursores permitem deslocação de uma janela para outra. Os comandos do LAPLINK permitem movimento na árvore de directorias de ambos os computadores, marcar ficheiros individuais (ou directorias inteiras) a enviar de um computador para outro, etc, etc.

O esquema das janelas está tão engenhosamente concebido que os ficheiros tanto podem ser enviados do computador local para o remoto como trazidos em sentido inverso. Quando se enviam directorias e subdirectorias, o LAPLINK reproduz no computador de destino a árvore de directorias, colocando os ficheiros nos sítios certos. Se conjugarmos um *data-switch* inteligente bi-direccional com o LAP-LINK, obtemos uma rede de transferência de ficheiros que resolve bastantes problemas. Os custos desta solução são ínfimos (cabo a 3 condutores, fichas comuns vulgarmente designadas por D25 ou D9, um porto série em cada computador e um *data-switch* bi-direccional com N<->N ligações) e os resultados excelentes.

QUANDO É QUE UMA REDE É REALMENTE NECESSÁRIA?

Se nem sempre é necessário adquirir uma rede para automatizar as tarefas de um escritório, em grande parte dos casos só uma rede resolverá o problema.

Quando falamos do acesso de vários utilizadores a uma mesma base de dados ou a um qualquer *software* de gestão comercial, em que vários utilizadores trabalham sobre o mesmo conjunto (consistente) de informação, então só uma rede resolverá o problema. A rede e o *software* de rede eliminam a duplicação indesejada de informação e garantem simultâneamente a consistência dos dados, pois todos os utilizadores trabalham sobre o mesmo ficheiro físico, ficando a arbitragem dos acessos concorrentes a cargo do sistema (e não dos utilizadores).

Ao optar por uma rede, muitos outros problemas se resolvem também:

A segurança da informação e o acesso aos programas instalados na rede são garantidos pelo próprio sistema operativo de rede e a preocupação dos *backups* (cópias de segurança) pode ser centralizada numa única pessoa (a vítima é o «administrador da rede») que garanta a periodicidade destas salvaguardas de informação.

Uma rede automatiza com facilidade o funcionamento de qualquer pequena/média empresa, e não tráz só problemas ao responsável pela manutenção da rede - permite-lhe também poupar imensas horas na instalação e configuração do *software* para todos os postos de trabalho, pois ao instalar e configurar um programa na rede, está a fazê-lo para toda a gente...

> **“Entre as facilidades oferecidas por qualquer rede estão o já referido «file-transfer» e o correio electrónico”**

Entre as facilidades oferecidas por qualquer rede estão o já referido «file-transfer» e o correio electrónico (E-MAIL), normalmente embebidos no próprio sistema operativo de rede. O E-MAIL permite que os utilizadores da rede comuniquem entre si, utilizando a própria rede de computadores como meio de transmissão das mensagens: a cada utilizador é associado um endereço (equivalente à morada) e uma caixa do correio; as mensagens são enviadas para a caixa do correio dos utilizadores sob a forma de ficheiros, que este consultará e dará seguimento, como se de mensagens em papel se tratassem.

Uma facilidade que se começa a implantar junto dos possuidores de redes é a gestão do envio e recepção de FAX pela rede: idealmente, um *server* de FAX instalado numa rede pode receber um FAX e redireccioná-lo em forma de ficheiro para o destinatário (que só o imprime se quiser); pode também enviar os FAX de qualquer utilizador sem que este tenha de produzir o documento em papel - basta o ficheiro; no caso de congestionamento do FAX por sobrecarga de pedidos de envio, o *server* gere automaticamente a lista de espera dos documentos a telecopiar, permitindo aos remetentes não perder tempo na «bicha do FAX».

O QUE HÁ A ESCOLHER NUMA REDE?

Normalmente verificam-se algumas confusões entre aquilo que é o sistema operativo da rede (de que são exemplo os nomes NetWare, Microsoft LAN Manager, IBM LAN Server) e aquilo que é o suporte físico da rede - as placas e os fios (onde emergem os termos TOKEN RING, ETHERNET, ARC-NET, etc).

O facto de se pretender pôr em comunicação um conjunto de computadores implica que se definam regras para o estabelecimento dessa mesma comunicação: estas regras constituem o «protocolo de comunicação». O protocolo define a velocidade da transmissão de informação, o modo como se identificam o emissor e o receptor, como é

feita a detecção de erros, etc.

Cada suporte físico existente tem o seu protocolo estabelecido, de modo que um dado sistema operativo de rede possa usar qualquer suporte físico sem ter de conhecer profundamente as suas particularidades (podemos imaginar esta situação atribuindo ao protocolo o papel de um intérprete que sabe «falar» com o *hardware* independentemente da comunicação ser a dois ou quatro fios, mais rápida ou mais lenta, etc...). A preocupação do protocolo é uniformizar tudo acerca da instalação da rede, indo ao pormenor de especificar quais as fichas a usar e com quantos pinos... Esta uniformização é desejada porque permite que diferentes fabricantes produzam equipamentos que, por seguirem as normas, são compatíveis entre si.

Com regras bem definidas, um mesmo sistema operativo de rede pode estar (idealmente) instalado sobre qualquer suporte físico, sem que essa diferença seja transparente ao utilizador, ou seja, não sendo diferente o que aparece no ecrã em qualquer dos casos...

Ao optar por um determinado suporte físico, por exemplo TOKEN-RING ou ETHERNET, estamos a optar por uma determinada topologia (modo como os computadores estão ligados), tipo de cabo, tipo de placas a instalar no computador, velocidades de transmissão, etc.

Para esta decisão contribuem factores importantes, como sejam a velocidade máxima de transmissão que se pretende, o número de utilizadores que se pensa instalar, o preço que se está disposto a pagar, etc.

E contribui um factor MUITO importante: saber se há compatibilidade entre o *hardware* da rede e o sistema operativo (de rede) que se pensa instalar.

Tal como o sistema operativo DOS funciona bem nos computadores ditos IBM compatíveis e mal nos que não são totalmente compatíveis, também um determinado sistema operativo de rede poderá funcionar bem sobre a rede X e menos bem sobre a rede Y. E o paralelismo entre o S.O. de rede e o DOS aplica-se aqui em toda a linha: tal como é o DOS que efectua a gestão do computador e disponibiliza os seus recursos, também é o S.O. de rede que gere a rede (os discos, utilizadores e *passwords*, E-MAIL, etc...), sendo uma rede tão inútil sem o S.O. como o é um computador sem o seu sistema operativo.

“O termo ETHERNET define a forma como a informação é transferida entre os computadores e o tipo de ligações físicas entre eles e NÃO É o nome de um sistema operativo de rede”

ETHERNET: ONDE É QUE EU JÁ OUVI ESTE NOME?

O termo ETHERNET define a forma como a informação é transferida entre os computadores e o tipo de ligações físicas entre eles e NÃO É, definitivamente, o nome de um sistema operativo de rede.

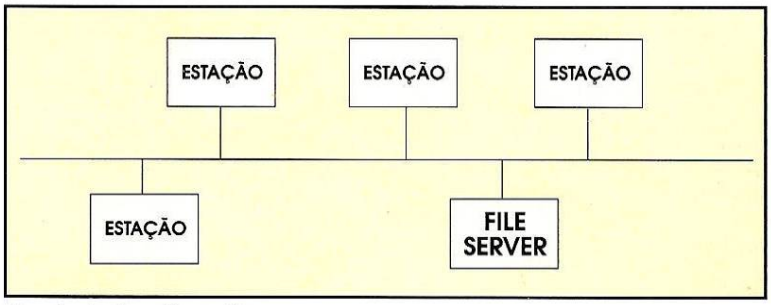
A este nível do processo determinam-se quais são as placas a montar nos computadores, o tipo de fios e de fichas e NÃO o que vai aparecer nos ecrãs; isto quer dizer que, para o utilizador comum, estar a trabalhar numa rede ETHERNET ou numa TOKEN-RING é-lhe rigorosamente indiferente...

Fisicamente falando, numa rede ETHERNET os computadores estão todos ligados ao mesmo fio, numa topologia designada por BUS. Com este esquema, um único cabo forma a rede, onde os computadores «escutam» o que se passa e «injectam» informações quando pretendem comunicar.

No ponto onde o computador se vai ligar à rede é colocado um pequeno aparelho designado por «transceiver», existindo tantos *transceivers* quantos os pontos de acesso.

O cabo propriamente dito pode ser de três tipos diferentes: «cabo coaxial fino» (thin-cable), «grosso» (thick-cable) ou «par-entrançado» (twisted-pair). Destes, apenas o cabo coaxial grosso é o genuíno cabo ETHERNET, sendo os outros dois «variações sobre um tema». A rede ETHERNET em cabo fino é conhecida por CHEAPER-NET (mais-barata) e tem as vantagens do menor custo deste tipo de cabo coaxial e da maior facilidade de instalação; por outro lado, sacrifica a robustez do cabo e alguma «performance».

A rede montada sobre o par-entrançado (o vulgar fio telefónico) permite, por seu lado, aproveitar as cablagens telefónicas dos edifícios para espinha-dorsal da rede, diminuindo significativamente os custos de instalação (a partir do momento em que os construtores passam a efectuar a cablagem do edifício para telefones - porque são obrigados a apresentar o projecto RITA - está montado o suporte físico de fio para a rede!); é evidente que, pela ausência de blindagem, este cabo é bastante sensível ao ruído eléctrico, sendo também bastante limitada a distância máxima admissível entre «nós» (pontos de acesso) sem perda significativa de *performance*.



Topologia da rede em Bus

Outro aspecto de grande importância (senão vital) numa rede é o protocolo, ou seja, a forma como as máquinas comunicam. O protocolo ETHERNET baseia-se no princípio de que, se vários computadores estão ligados ao mesmo cabo, todos podem «ouvir» em simultâneo o que se passa, mas apenas um pode «falar» de cada vez.

Quando esta regra é respeitada, tudo corre bem - cada «nó» (é assim que se designa o ponto de acesso à rede onde está ligado um computador) vai recolher apenas a informação que lhe diz respeito, sendo esta enviada em «blocos» pelo remetente com o endereço do destinatário - tal como as cartas chegam a nossa casa por terem a morada correcta, também o destinatário dos dados vai receber o que lhe foi enviado. Os problemas ocorrem quando dois «nós» tentam comunicar ao mesmo tempo, o que pode acontecer se ambos tiverem essa necessidade, ao verificarem que o BUS está disponível (ninguém tinha enviado dados) decidirem iniciar a comunicação AO MESMO TEMPO. É um azar que pode acontecer, aumentando a probabilidade a partir do momento em que a rede possua muitos utilizadores (mais de 50).

Quando ocorre a colisão (é o nome científico deste «azar»), todos os «nós» são alertados para ela e a comunicação sofre uma interrupção. Após um intervalo de tempo aleatório (diferente para cada nó), os «nós» que estavam a tentar comunicar efectuam nova tentativa - o que for mais rápido «ganhará» o BUS para sí e tudo regressa à normalidade.

Numa rede a informação circula em blocos - encomendas de «correio», com a «morada» do destinatário e com os dados lá dentro. Quando a informação a enviar é demasiada para caber num único bloco, é dividida em tantos blocos quantos os necessários para efectuar o envio.

Como se pode verificar, aqui funciona pura e simples-

mente o senso comum!

Os referidos blocos de informação designam-se por FRAMES e o seu conteúdo e dimensão estão perfeitamente definidos. Uma FRAME é constituída por um preâmbulo (8 bytes), o endereço do destinatário (6 bytes), o endereço do remetente (6 bytes), o tipo da frame (2 bytes), a área da informação (até 1500 bytes) e, por fim, o código de verificação da validade da FRAME (4 bytes de CRC).

PREÂMBULO 8 bytes	DESTINATÁRIO 6 bytes	REMETENTE 6 bytes	TIPO 2 bytes	DADOS 1500 bytes máximo	CRC 4 bytes

Layout da Frame Ethernet

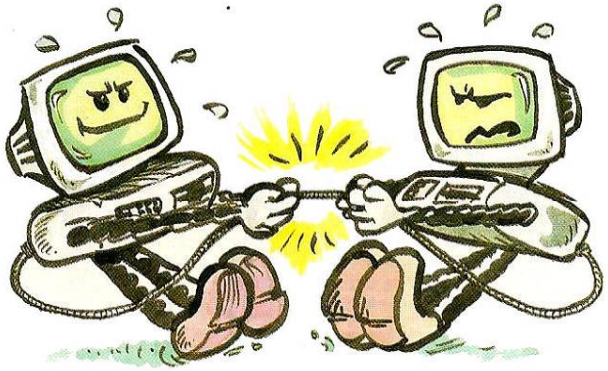
A primeira versão da rede ETHERNET foi tornada pública em 1980, resultando do esforço desenvolvido por um triunvirato constituído pela DEC, INTEL e XEROX no sentido de criar um standard para as redes de computadores. Até então, cada fabricante desenvolvia a sua própria arquitectura, sendo todas elas incompatíveis entre si - o comprador de um determinado sistema ficava eternamente «amarrado» àquele fabricante e todas as tentativas de interligar redes de diferentes fabricantes revelavam-se difíceis e onerosas.

Como base para o desenvolvimento da ETHERNET, esta equipa usou um sistema que a Xerox já possuia para uso interno e em funcionamento no Palo Alto Research Center. A arquitectura ETHERNET foi sempre uma arquitectura aberta, tendo as companhias que efectuavam o desenvolvimento compreendido (inteligentemente, dizemos nós) que o sistema só teria a ganhar com uma competição aberta, melhorando as performances, baixando inevitavelmente os preços mas contribuindo para a sua divulgação e, desta forma, aumentando substancialmente as vendas.

GOODBYE OLD FRIEND, THIS IS THE END...

«This is the end» mas só por agora, porque muito mais há a dizer sobre redes de computadores. Sabiam que já existem LANs que funcionam sem cabos? Por estas e por outras é que regressaremos ao tema no próximo número da SPOOLER, onde também ouvirão falar do NETWARE e do TOKEN-RING. Inté.

PAULO BERNARDO





AST Premium® Exec™ 386SX/25 Color

DESCONTO 10% A. P. P.

- Microprocessador AMD 386SXL, 25 Mhz
- 4 Mb RAM expansível a 8 Mb (SIMM de 4 Mb)
- Drive de 3,5" – 1,44 Mb
- Disco rígido de 2,5", 60, 80 ou 120 Mb
- Monitor LCD policrimático, compatível com VGA, EGA, CGA e MDA
- Teclado de 82/83 teclas
- Portas 9 pinos série, 25 pinos paralela, porta para monitor VGA externo, ligação para teclado 101/102 teclas externo ou mouse.
- MS-DOS 5.0 da AST, Gestor de vídeo LCDVGA, Super PC-Kwik Disk Caching, Windows Protected Mode Suspend Driver, Protected Mode Suspend Driver, Software Sleep Driver
- Manual do Utilizador, Guia de bolso
- Password de segurança da AST BIOS
- Mala de transporte com alça
- Dimensões: 28,9x5,8x22,8 cm
- Peso: 3,4 Kg

Acessórios Premium opcionais

Coprocessador – Modem – Fax/Modem – Teclado numérico – Mala de acessórios em pele ou nylon – Porta série adicional – Adaptador/carregador de corrente para automóvel – Conjunto de baterias secundário – Expansão de 4 Mb SIMM, etc.

670 000$00 + 16% IVA
60 MB HD

720 000$00 + 16% IVA
80 Mb HD

825 000$00 + 16% IVA
120 Mb HD

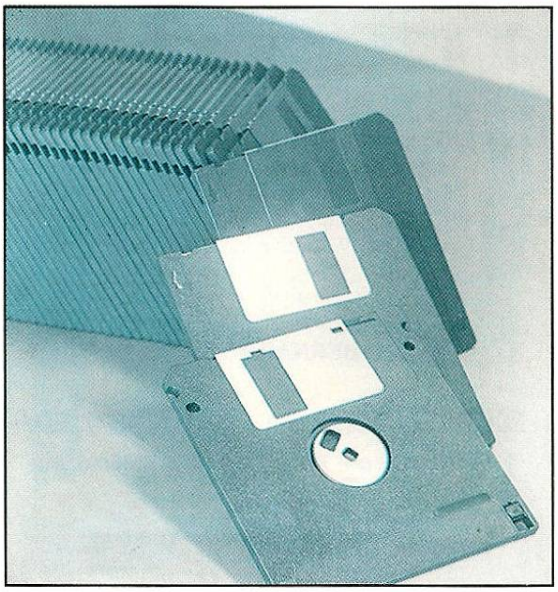
abc

A VERDADE EM SOLUÇÕES INFORMÁTICAS

Rua dos Sapateiros, 160 – 2º — 1100 LISBOA
Telef.: (01) 342 44 48/347 80 94/347 78 96 — Fax: 342 67 22

MONTRA Spooler

DISQUETAS BULK (sem marca)



50 DISQUETAS 3,5" DS/DD
Refª. D040 **Preço: 5 655$00**

50 DISQUETAS 3,5" DS/HD
Refª. D041 **Preço: 9 425$00**

RATOS PRIMAX



«RATO» COMPATÍVEL MICROSOFT E MOUSE SYSTEM 350-700 DPI – DYNAMIC RESOLUTION – 3 TECLAS

Embalagem com o «rato» e software de instalação
Refª PR01 **Preço: 5 052$00**

Embalagem com o «rato», software, tapete, caixa e ficha adaptadora 9/25 pinos
Refª PR02 **Preço: 6 485$00**

10% de desconto
para assinantes
em todos os produtos da Montra

TODOS OS PREÇOS APRESENTADOS JÁ INCLUEM O IVA.

SOUND BLASTER PRO



SISTEMA PROFISSIONAL PARA RECOLHA, EMISSÃO E TRATAMENTO DE SOM, COMPATÍVEL DOS 3.0 E WINDOWS 3.0 (E SUPERIORES).

Gravação estéreo; compatibilidade mono com AdLib e Sound Blaster; Twin FM chips proporcionando 22 vozes; misturador estéreo digital/analógico FM, DAC, line-in, microfone, CD e altifalante interno de PC; software para regulação de fade, pan, e volumes direito e esquerdo, mistura com microfone; cabo Midi *standard*; *Interface* CD-ROM com conector interno para CD-Audio; porta para *Joystick*; Amplificador de microfone com AGC (Automatic Gain Control); 4 W por canal; Controlo manual de volume na placa; etc...

A embalagem Sound Blaster Pro contém: Carta Sound Blaster Pro 16 bit; Cabo RCA Audio; Cabos Midi; Disquetas de 3,5" e 5,25"; Software SPJ Sequencer; Editor de voz II; SBTALKER (software texto/voz); Software FM (orgão inteligente); reprodutor CD; demonstração MMPLAY; *drivers* para DOS e Windows.

Requisitos em Hard e Software: PC AT compatível (286 ou superior), mínimo de 512 Kb RAM, DOS 3.0 ou superior, carta EGA ou VGA (VGA recomendada)

Refª SB01 **Preço: 51 272$00**

KITS PARA LIMPEZA DE DRIVES

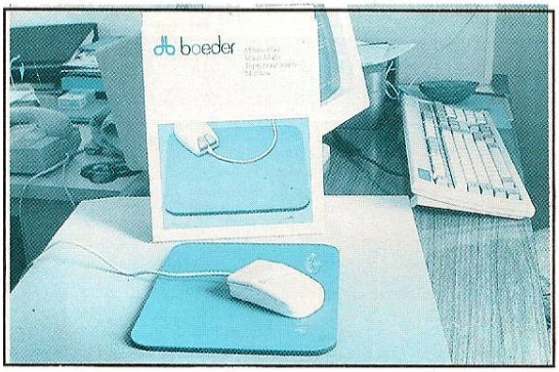


Ref.ª G001
Disqueta 3 ½
602$00

Ref.ª G002
Disqueta 5 ¼
602$00

Kits de limpeza de drives incluindo disqueta e frasco com o líquido de limpeza. Qualidade comprovada.

TAPETE PARA MOUSE



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

Ref.ª G005 PREÇO: 871$00

LIVROS EDITORIAL O LIVRO (Em português)

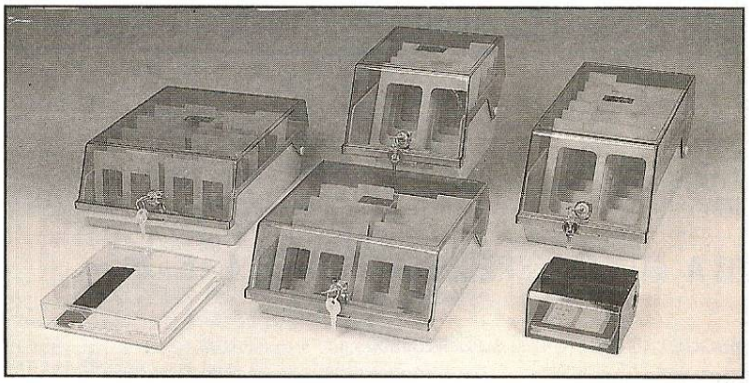


– INTRODUÇÃO À INFORMÁTICA E COMPUTADORES
REF.: L001 PREÇO: 1 806$00

– TÉCNICAS DE PROGRAMAÇÃO
REF.: L002 PREÇO: 1 575$00

– INTRODUÇÃO AO BASIC
REF.: L003 PREÇO: 2 352$00

CAIXAS ARQUIVADORAS PARA DISQUETAS



– 10 disquetas 3,5"	Ref.: H004	Preço:	272$00
– 10 disquetas 5,25"	Ref.: H005	Preço:	344$00
– 50 disquetas 5,25"	Ref.: H006	Preço:	1 131$00
– 100 disquetas 5,25"	Ref.: H007	Preço:	1 328$00
– 50 disquetas. 3,5"	Ref.: H008	Preço:	1 192$00
– 80 disquetas 3,5"	Ref.: H009	Preço:	1 356$00
– 100 disquetas. 3,5"	Ref.: H010	Preço:	1 500$00



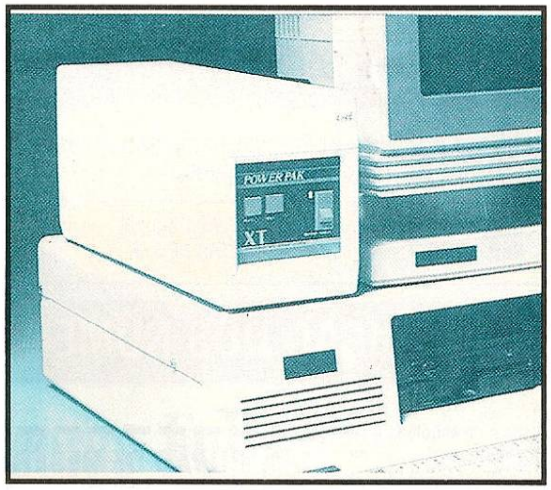
DBase III Plus
PROGRAMAÇÃO AVANÇADA

Refª. GR01 COM DISQUETA DE 5,25" 1 890$00

Refª. GR02 COM DISQUETA DE 3,5" 1 995$00

INVISUAIS: Quando solicitado forneceremos em disqueta suplementar o texto integral do livro.

POWER PAK – AT



Unidade de Estabilização e Manutenção de energia eléctrica, indicada para protecção da configuração completa de PC AT.
Bateria selada, sem manutenção.
Potência máxima: 550 VA – Autonomia: 25 minutos.

GARANTIA DE 1 ANO

Refª. PK01 PREÇO: 52 003$00

PAPEL CONTÍNUO

Embalagem de 500 folhas de papel contínuo (zebra americana) 12x9,5"

Refª. P020 Preço: 785$00

MODEMS BEST DATA



Modems de alta qualidade a preços notáveis.

Refª BD01	Modem interno – 2400 bps W/MNP 2-7 por software V.22 e V.22 bis)	**Preço: 19 000$00**
Refª BD02	Modem externo– 2400 bps W/MNP 2-7 por software V.22 e V.22 bis)	**Preço: 30 160$00**
Refª BD03	Fax/Modem interno – 9600 bps Fax – 2400 bps Modem W/MNP 7 por software V.22 e V.22 bis)	**Preço: 38 454$00**
Refª BD04	Fax/Modem externo – 9600 bps Fax – 2400 bps Modem W/MNP 7 por software V.22 e V.22 bis)	**Preço: 43 732$00**

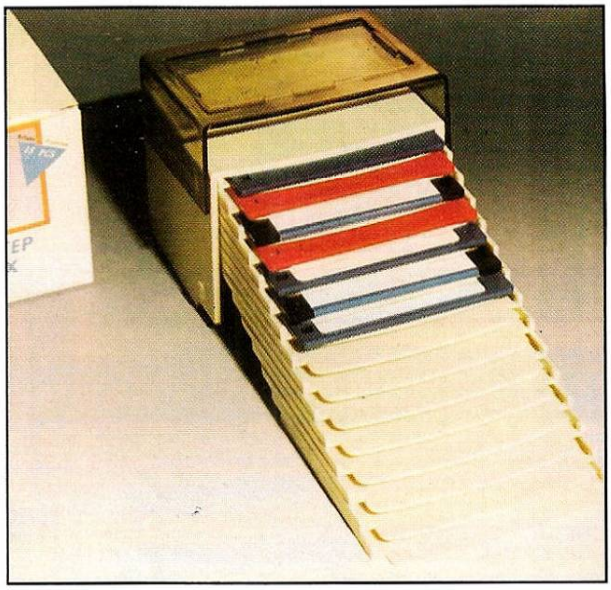
Nota: *estes Modems aguardam registo de homologação*

DOS 5.0 (EM PORTUGUÊS)

Conheça as novas e poderosas potencialidades do sistema operativo MS-DOS 5.0. Tudo em português.

Refª. MS01	**COM DISQUETAS DE 5,25"**	**20 300$00**
Refª. MS02	**COM DISQUETAS DE 3,5"**	**20 300$00**

CAIXA ARQUIVADORA - 15 DISQ. 3,5"



Caixa arquivadora «elástica» para o acondicionamento perfeito das suas disquetas de 3,5". Versátil e compacta.

Refª BX01	**Preço: 950$00**

BBSpooler

☎ (01) 475 22 89

Dias úteis: das 18.30 às 9.00 horas
Feriados e fins-de-semana: 24 horas

REEDIÇÃO DOS NÚMEROS ESGOTADOS DA SPOOLER (4, 5 e 6)

Depois de esgotada a reedição dos números 1, 2 e 3 da Spooler e para satisfazer a procura manifestada por inúmeros leitores vamos tornar a imprimir os números 4, 5 e 6, agrupados num só volume.

Esta edição tem uma tiragem reduzida e não será distribuida.

O único modo de possuir um exemplar é reservá-lo, desde já, preenchendo o cupão anexo (ou fotocópia) e remete-lo para os nossos serviços:

SPOOLER MAGAZINE
Rua Duarte Pacheco Pereira,
69-A – Damaia-de Baixo
2700 AMADORA

CUPÃO DE RESERVA DA EDIÇÃO ESPECIAL SPOOLER – NÚMEROS 4, 5 E 6

NOME .....

ENDEREÇO .....

.....

C. POSTAL ..... LOCALIDADE ..... TELEF. .....

DESEJO RECEBER A EDIÇÃO ESPECIAL «SPOOLER 4 - 5 - 6»

☐ **COM 3 DISQUETAS DE 5 1/4" ..... 1 200$00**
☐ **COM 2 DISQUETAS DE 3 1/2" ..... 1 200$00**

Não mande qualquer pagamento agora. A edição especial ser-lhe-á enviada à cobrança.
Aos preços indicados serão acrescidos os portes de envio e taxa de cobrança dos C.T.T.

Assinatura

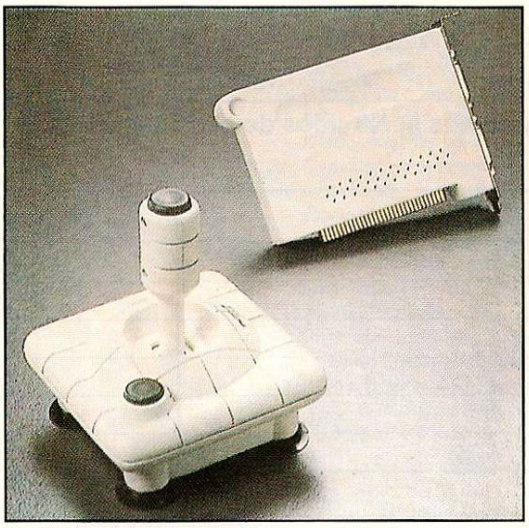
QUICKJOY M-5



Compatível para PC/XT/AT este joystick de *design* ergonómico é o ideal para os seus jogos de acção.
A placa pode ser adquirida em separado.

Refª G004 **Preço: 3 951$00**

JOYSTICK + PLACA SV 203



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

Refª G003 **Preço: 6 334$00**

DISQUETAS PLATINUM



Caixas com 10 disquetas:
– 5,25" DS/DD **Ref.: PL01** **Preço: 905$00**
– 3,5" DS/DD **Ref.: PL02** **Preço: 1 463$00**

PLACA QS-120 PARA JOYSTICK

Placa para joystick

Refª G04A **Preço: 3 861$00**

RATO MIC - MI-260



Regressado ao nosso convívio cá está o rato MIC, com nova embalagem e muito, muito mais barato!

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), ficha adaptadora 9/25 pinos.

Refª MIC1 **Preço: 2 500$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.

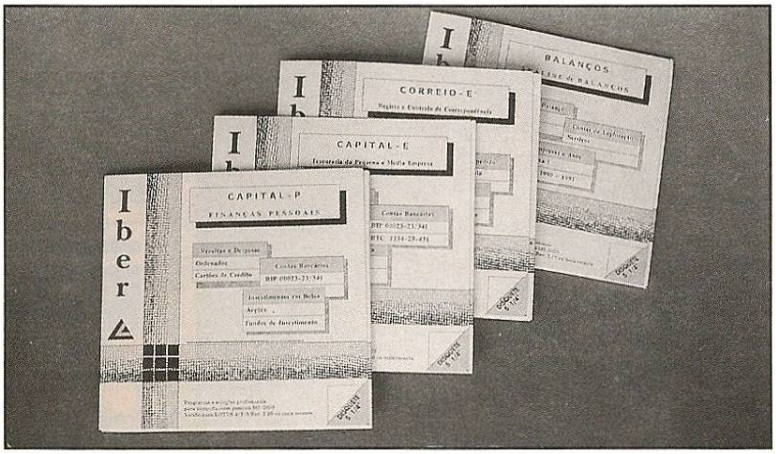
CAPA PROTECTORA PARA PCs E IMPRESSORAS



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.: G006 – Capa para PC	**1 674$00**
Ref.: G007 – Capa p/ imp. 80 c.	**913$00**
Ref.: G008 – Capa p/ imp. 132 c.	**913$00**

SOFTWARE IBER



Programas e soluções profissionais para compatíveis MS-DOS.
Não requerem conhecimentos de Lotus 1-2-3 ou de MS-DOS.

Refª IB01* **CAPITAL-P – FINANÇAS PESSOAIS.**

Refª IB02* **CAPITAL-E – TESOURARIA DE PEQUENA E MÉDIA EMPRESA.**

Refª IB03 **CORREIO-E – REGISTO E CONTROLO DE CORRESPONDÊNCIA..**

Refª IB04* **BALANÇOS – ANÁLISE DE BALANÇOS.**

Refª IB05 **BIBLOS – GESTÃO DE LIVROS, VIDEOS E DISCOS.**

Refª IB06 **CATÁLOGO – APOIO A COMPRAS – PRODUTOS FORNECEDORES – PREÇOS**

Em disquetas de 3,5" (720 Kb) PREÇO: 3 620$00

***Versões para Lotus 1-2-3 – Rev 2.10 ou mais recente**

CUPÃO MONTRA SPOOLER

Recorte ou fotocopie o cupão e remeta-o para:

SPOOLER MAGAZINE
Rua Duarte Pacheco Pereira, 69-A – Damaia
2700 AMADORA

ESTIMADO LEITOR:
Tentamos entregar as suas encomendas o mais rápido possível.
Quando um número da **Spooler** sai ao público, somos «bombardeados» com muitos milhares de pedidos.
Não desespere se não receber a sua encomenda no dia seguinte e **por favor não a devolva.** Os preços por nós praticados (quase isentos de lucro) não permitem o encargo adicional de ter que pagar os portes do retorno da encomenda. Comunique-nos se pretender, à *posteriori*, desistir da encomenda.

CUPÃO DE ENCOMENDA – MONTRA SPOOLER

Cliente nº.:
A PREENCHER PELOS NOSSOS SERVIÇOS

Desejo receber à cobrança (com os **custos acrescidos de portes e taxas**) os seguintes produtos:

Quant.	Refª.	Designação	Preço
		TOTAL	

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

Spooler 19

O valor indicado não paga os direitos destes programas, mas sim, apenas as próprias disquetas e o trabalho de cópia e distribuição. Será o utilizador dos programas tipo «shareware» a ter a responsabilidade de efectuar o respectivo pagamento, directamente aos autores, nos termos e condições por eles propostos. A Spooler limita-se a distribuir esses programas, devidamente testados e na sua versão integral.

Por motivos de espaço publicamos neste número apenas as novas referências. As referências anteriores (SH01 a SH 094) podem ser consultadas na Spooler Nº 18.

Utilitários para Windows

APOR13.EXE – Interface global para Windows 3.0
ALWIN15.EXE – Sistema para planeamento, gestão e controlo de tarefas
Refª. SH95 – 1 disqueta de 5,25" ..... 500$00
Refª. SH96 – 1 disqueta de 3,5" ..... 600$00

Utilitários e fontes para C

LCOMM30 – Rotinas para comunicações
PROWNDW1 – Janelas em C – Demonstrações
PROWNDW2 – Janelas em C – Fontes
PROWNDW3 – Janelas em C – Bibliotecas
MOUSE – Rotinas didáticas para o rato
Refª. SH97 – 1 disqueta de 5,25" ..... 500$00
Refª. SH98 – 1 disqueta de 3,5" ..... 600$00

Interrupções

INTER191 – Documentação completa sobre interrupções
INTERUPT – Os endereços das interrupções
Refª. SH99 – 1 disqueta de 5,25" ..... 500$00
Refª. SH100 – 1 disqueta de 3,5" ..... 600$00

ODYSSEY

ODY150SH – A versão 1.50 do famoso software para comunicações
Refª. SH101 – 1 disqueta de 5,25" ... 500$00
Refª. SH102 – 1 disqueta de 3,5" ..... 600$00

Utilitários diversos

DANCAD3D – Desenho e animação gráfica – Programa de CAD – **LQUALITY** – Melhore a qualidade da sua impressora – **STOP&GO** – Um cronómetro para o seu PC
Refª. SH103 – 1 disqueta de 5,25" .... 500$00
Refª. SH104 – 1 disqueta de 3,5" ..... 600$00

Utilitários diversos

ACE1, ACE2 – Programa de astronomia
ACTILOG – Um registo da utilização do computador – **GIP** – Um gerador de instalações para software
Refª. SH105 – 1 disqueta de 5,25" .... 500$00
Refª. SH106 – 1 disqueta de 3,5" ..... 600$00

GIF's Alta resolução

AUTOMATE – Um autómato 1024x768x256
PAINT – As ferramentas do pintor 640x480x256
Refª. SH107 – 1 disqueta de 5,25" .... 500$00
Refª. SH108 – 1 disqueta de 3,5" ..... 600$00

Análise antropométrica e Psiquiatria doméstica

CASP30 – Software para análise antropométrica, especialmente dedicado aos profissionais da saúde.
ELIZA2 – Psiquiatria doméstica. Corre em GW-Basic e a base de dados pode ser aumentada e alterada para português.
Refª. SH109 – 1 disqueta de 5,25" .... 500$00
Refª. SH110 – 1 disqueta de 3,5" ..... 600$00

GIF's Alta resolução

PLAT – Uma natureza...morta 640x512x16
SACKS – Uma natureza...viva 640x480x256
Refª. SH 111 – 1 disqueta de 5,25" .. 500$00
Refª. SH 112 – 1 disqueta de 3,5" .... 600$00

Jogos

ORION – Orion Odyssey
1MARS – Os monumentos de Marte
Refª. SH 113 – 1 disqueta de 5,25" .. 500$00
Refª. SH 114 – 1 disqueta de 3,5" .... 600$00

Jogos

WHEELS2 – Roda da Fortuna para DOS
CRYSTAL – Crystal Caves
TK – Solitário para DOS
Refª. SH 115 – 1 disqueta de 5,25" .. 500$00
Refª. SH 116 – 1 disqueta de 3,5" .... 600$00

FRAINT16

FRACTINT V 1.6 – Gerador e manipulador de fractais. Conjuntos de Julia, Mandel e muitos outros tipos. Animação em 3D. Grava em formato GIF.
Refª. SH 117 – 1 disqueta de 5,25" .. 500$00
Refª. SH 118 – 1 disqueta de 3,5" .... 600$00

Colecção Pintura – VELASQUEZ

JARDINS, COMTE, MENINAS
Refª. SH 119 – 1 disqueta de 5,25" .. 500$00

Colecção Pintura – VELASQUEZ

ESOPO, INFANTE, JARDINS, COMTE, MENINAS
Refª. SH 120 – 1 disqueta de 3,5" .... 600$00

Dicionários para TRADUTOR.COM

(Ver artigo neste número da Spooler)
ING_PORT.DIC – Dic. de Inglês-Português
FRN_PORT.DIC – Dicionário de Francês-Português
SIGLAS.DIC – Glossário de siglas, abreviaturas e acrónimos
Refª. SH 121 – 1 disqueta de 5,25" .. 500$00
Refª. SH 122 – 1 disqueta de 3,5" .... 600$00

ASSINATURAS Spooler

Recorte ou fotocopie o cupão, preencha-o com letra bem legível e remeta-o, acompanhado do respectivo pagamento em cheque ou vale de correio, para:

SPOOLER MAGAZINE
R. Duarte Pacheco Pereira, 69-A
Damaia – 2700 AMADORA

*Os números anteriores da **Spooler** podem ser encomendados à cobrança utilizando o cupão da **Montra Spooler**. Ao preço de capa serão acrescidos os custos dos portes e taxas de cobrança.*
Se preferir disquetas de 3,5", isto custar-lhe-á mais 140$00 por exemplar.

NOTA: Os números 1 a 6 e o 10 e 11 da **Spooler** encontram-se esgotados. A reedição especial dos números 1, 2 e 3, também!

Spooler 19

CUPÃO DE ASSINATURA

Assinante nº:
A PREENCHER PELOS NOSSOS SERVIÇOS

NOME .....
DATA DE NASCIMENTO* _____ / ____ / ____
PROFISSAO* .....
TRABALHA COM COMPUTADORES?* SIM☐ NÃO ☐
SE SIM ESPECIFIQUE A MARCA E MODELO* .....
ENDEREÇO .....
.....
C. POSTAL ..... LOCALIDADE ..... TELEF(S) .....

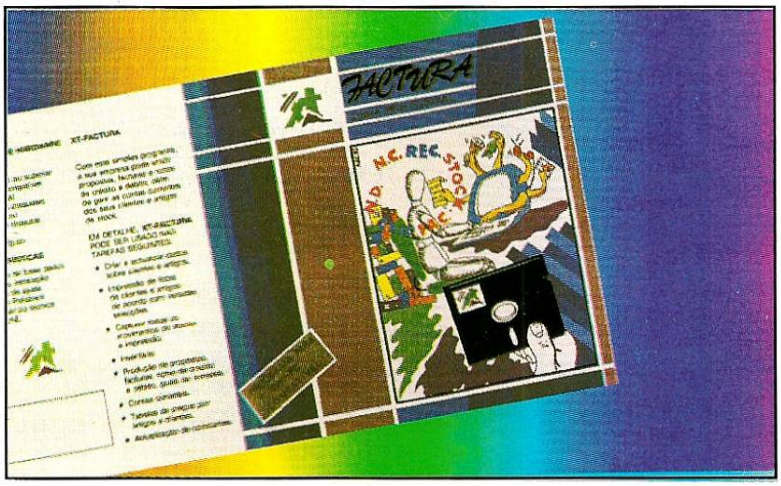
DESEJO ASSINAR A SPOOLER A PARTIR DO Nº. 20 (inclusive) E DURANTE 12 NÚMEROS.

☐ **MODALIDADE COM DISQ. DE 5 ¼" ..... 5 500$00**
☐ **MODALIDADE COM DISQ. DE 3 ½" ..... 6 000$00**

PARA PAGAMENTO DA ASSINATURA ENVIO :
☐ O CHEQUE Nº. ..... S/ O BANCO .....
☐ O VALE DE CORREIO Nº. .....

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

PROGRAMA PARA FACTURAÇÃO «FACTURA»



Este programa de fácil utilização, instalação simples, textos de ajuda claros e precisos, menus *pulldown*, obedece a todos os requisitos legais em vigor e permite-lhe:

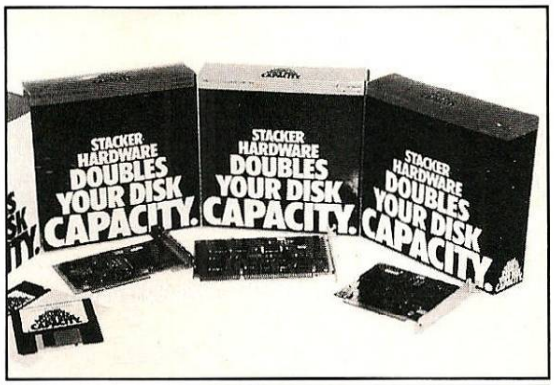
- Criar e manter actuais dados sobre clientes e artigos;
- Imprimir listagens de clientes e artigos de acordo com variadas opções de selecção;
- Manter actualizado o movimento e inventário de stocks;
- Produzir e imprimir propostas, facturas, notas de crédito, notas de débito e guias de remessa.
- Contas correntes, tabelas de preços por artigos e clientes.

REQUISITOS DE HARD E SOFTWARE:
- MS-DOS 3.1 ou superior; PC 100% compatível; 640 RAM ; 2 unidades de disquetas ou 1 unidade disquetas + disco rígido.

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

Refª FT01	Preço: 17 400$00

STACKER



STACKER – PARA DUPLICAR A CAPACIDADE DO SEU DISCO E INCREMENTAR O DESEMPENHO DO SISTEMA

O que obtém com o Stacker:
- O dobro da capacidade do seu disco (de qualquer tipo)
- Rápida compressão em tempo real
- «Cache» que aumenta consideravelmente o desempenho do sistema
- 100% de compatibilidade
- Instalação rápida e fácil (basta abrir a caixa do seu PC, colocar a placa **Stacker** numa *slot* livre e «correr» o software de instalação com instruções em português)
- 1 ano de garantia da placa

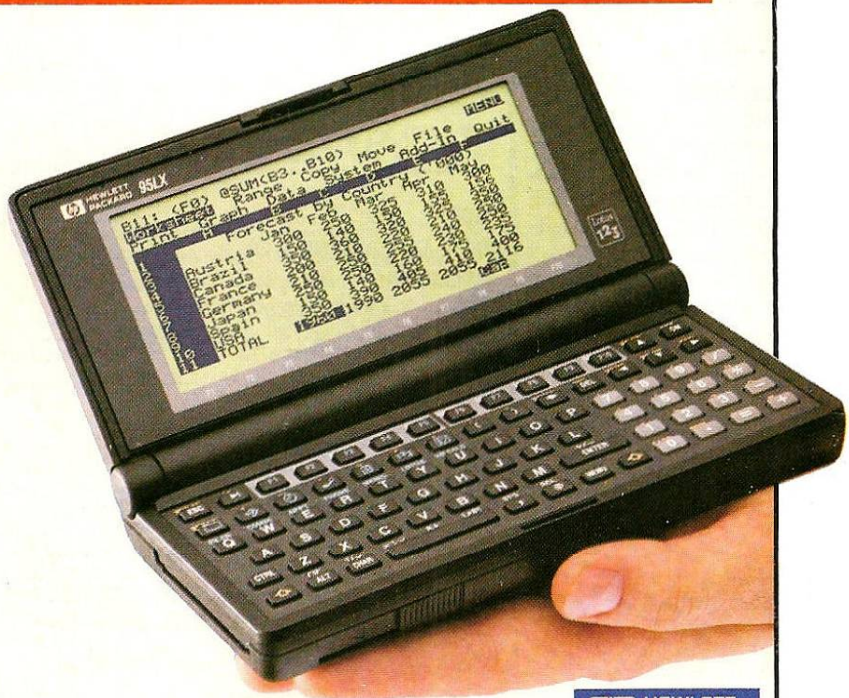
Conteúdo da embalagem:
- Placa **Stacker**
- Sofware **Stacker** (em disquetas de 3,5" e 5,25")
- Certificado de garantia
- Manual de instruções (em inglês)

Refª ST01	Só software	26 100$00
Refª ST02	Placa + Software p/ XT	43 848$00
Refª ST03	Placa + Software p/ AT (286/386/486)	49 300$00

O SEU PC DE BOLSO HP 95LX

Graficria pub

- 512 Kb a 1 Mb de ROM
- *Software*:
  - MS DOS®3.22
  - Lotus®1-2-3®, versão 2.2
  - Comunicação de dados
  - Editor de texto
  - Gestão de ficheiros
  - Calculadora H. P.
  - Agenda
  - Lista telefónica
- 512 Kb de RAM expansíveis até 1 Mb
- Ligação a PC, impressora, *modem*, etc.



Contacte:

DIMECRO
Representações Comerciais e Técnicas, Lda.
Rua Joaquim António de Aguiar, 41 - 6º Esqº – 1000 LISBOA
Telefones: 315 76 12/3/4/5 – Fax: 315 76 12

HEWLETT PACKARD
Distribuidor

Graficria pub adapt.

THE COMPLETE COLOUR SOLUTION

73 500$ + IVA

«The Complete Colour Solution» proporciona-lhe tirar todo o partido da sua placa VGA!
Você pode capturar imagens a cores de alta resolução a partir do seu vídeogravador caseiro, camâra ou qualquer outra fonte de sinal vídeo.
As imagens (não retocadas) que ilustram esta página foram digitalizadas a 256 cores. São um pequeno exemplo do que você pode obter em apenas alguns segundos.

DÊ LARGAS À SUA IMAGINAÇÃO E CRIATIVIDADE



CONTEÚDO DA EMBALAGEM
- Vidi-PC (Carta para a captura e digitalização de imagens)
- VidiChrome-PC (software para tratamento da cor em alta resolução)
- Vidi-RGB (separador electrónico de cores)
- PC Paintbrush IV Plus (provavelmente o melhor software do mundo para arte e grafismos informáticos)
- Picture Box (Conjunto de demonstração)

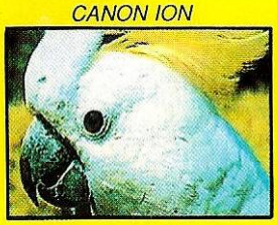
SOFTWARE:
Disponível em disquetes de 3,5" ou 5,25"

CARACTERÍSTICAS
Resoluções de imagem possíveis:
- 320x200/256 cores
- 640x400/256 cores
- 640x480/256 cores
- 800x600/256 cores
- Suporta todas as resoluções de 256 cores até 800x600 (EVGA)
- Captura até 4096 cores em menos de 1 segundo
- Avançado algoritmo para selecção de paletes
- Zoom interactivo
- Converte imagens negativas em positivas
- Imprime em todas as impressoras *standadrd* mono ou policromáticas
- Grava as imagens nos formatos Gem, PCX ou Tiff.

APLICAÇÕES:
- Apresentação de produtos ou serviços
- Demonstração e vendas
- DTP a cores (edição computorizada)
- Ensino e formação profissional
- Animação
- Concepção de catálogos e literatura diversa

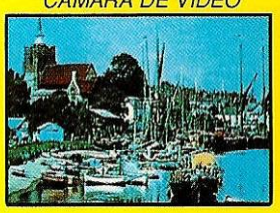


CANON ION





CAMÂRA DE VÍDEO



VIDEOGRAVADOR



Para obter estes resultados, apenas necessita possuir os seguintes equipamentos:
- Computador compatível AT ou XT
- 640 Kb de RAM
- Monitor VGA policromático
- Disco rígido
- Rato
- Uma fonte de vídeo (câmara, videogravador, etc.)

Game® informática

DISTRIBUÍDOR EM PORTUGAL DA ROMBA Limited

Loja 1 – R. Elias Garcia, 362-D – C. C. Babilónia, Loja 70 – 2700 AMADORA – Tel. 492 14 32
Loja 2 – **Departamento Profissional** – Rua D. Pedro IV, Lote 13, Loja C – 2745 QUELUZ
Telef.: 436 15 38 – Tel./Fax: 436 10 08

INFORPOR' 92 EXPOSIÇÃO INTERNACIONAL DE INFORMÁTICA E TECNOLOGIAS DA INFORMAÇÃO

INDEPENDENTE CRIATIVA OPERACIONAL PROFISSIONAL REVOLUCIONÁRIA INSATISFEITA DIDÁTICA SURPREENDENTE FANTÁSTICA INOVADORA BELA



22 a 25 de Outubro - F. I. L.

ORGANIZAÇÃO: CERTAME, Feiras Exposições e Congressos, R. Rodrigo da Fonseca 74 R/C Dt.º, 1200 LISBOA Tel. 3861041 FAX 3864245