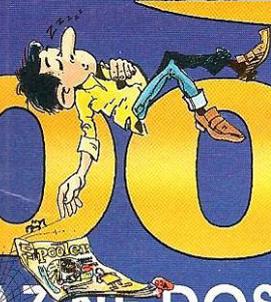


N.º 8 MARÇO/MAIO 1990 PREÇO: 400\$00

Spooler



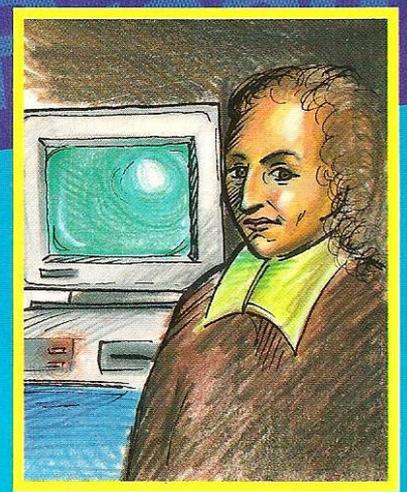
MAGAZINE DOS UTILIZADORES PC

PostScript A LINGUAGEM GRÁFICA

DISCOS: OS DUROS

VISUALIZAÇÃO CIENTÍFICA

- MODEMS
- PONHA O SEU TECLADO A FALAR PORTUGUÊS
- VITAMINAS PARA O LOTUS 123
- CONTAS EM DIA
- VAGUEANDO PELO INFORMIX SQL



PASCAL: PARA ENSINAR E APRENDER

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

PRECISION™

FLEXIBLE DISKS Manufactured by Xidex®

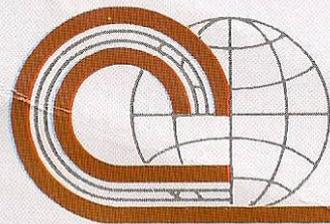
**PARA POUPAR
OS SEUS DADOS
E O SEU DINHEIRO!**



- GARANTIA VITALÍCIA
- 100% ISENTAS DE ERROS
- REVESTIMENTO ESPECIAL DE PROTECÇÃO E LIMPEZA

**A MELHOR PERFORMANCE
AO MAIS BAIXO CUSTO**

REPRESENTANTE
E DISTRIBUIDOR EXCLUSIVO:



Consumíveis para Informática
SERVIÇOS DE INFORMÁTICA, LDA.

ADMINISTRAÇÃO E VENDAS:

R. da Quinta do Almagem, 7-A, 7-B — 1300 LISBOA — PORTUGAL
Telefs: 64 48-54/64 51 38/64 41 50 — Telex: 63 755 ULDATA P — Telefax: (01) 64 91-08

Agentes no Norte:

Consumíveis para Informática
CONSUMÍVEIS PARA INFORMÁTICA, LDA.
Consumíveis para Informática

Avenida da República, 740, 2º, sala 26 — 4400 VILA NOVA DE GAIA
Telefs: 30 96 21/30 97 21

Nº 8 — Março a Maio de 1990

Spooler

MAGAZINE DOS UTILIZADORES PC

VALE MAIS TARDE QUE NUNCA...

Este número da SPOOLER esteve *embruxado*. «Isso tem muita graça!...» — dirá o leitor e mordazmente acrescentará: «Agora desculpem-se com provérbios populares e bruxas!». Apesar de tudo, nós, aqui na redacção, também não acreditamos em bruxas (mas que as há, há!...). Várias crises de todas as qualidades e feitios (que não vamos aqui enunciar para não fastidiar o nosso já zangado leitor), provocaram este atraso monumental na saída da revista de Março. Uma coisa é certa: cá estamos, em Maio, a publicar o n.º 8 da SPOOLER que deveria ter saído em Março.

Houve grandes mudanças na nossa redacção e serviços de apoio, em termos humanos e estruturais, com o objectivo de tornar a nossa revista mensal digna dessa periodicidade.

Fazer uma revista de informática para o utilizador português, escrita por portugueses com todo o seu material inédito e original, juntar a isso uma disquete que acompanha fielmente cada exemplar e tentar que esse empreendimento não seja um desastre financeiro, é uma tarefa para Golias que tem vindo a ser desempenhada por minúsculos Davides.

Vamos continuar, pois possuímos um trunfo muito precioso de que poucas publicações de natureza técnica se podem vangloriar: **LEITORES!** A SPOOLER, para além dos seus já numerosos (e um pouco maltratados) assinantes, esgota tecnicamente todas as suas edições, o que nos vai obrigar, brevemente, a reeditar alguns dos números atrasados, devido aos numerosos pedidos por parte dos novos assinantes e leitores em geral, que vão estabelecendo contacto com a revista.

Não vamos prometer que a SPOOLER vá sair, a partir de agora, sempre a tempo e horas. Prometemos, isso sim, que vamos fazer todos os possíveis nesse sentido.

Louvando a paciência e impaciência que tem demonstrado, agradecemos-lhe, Amigo Leitor as incontáveis vezes que já perguntou na livraria pela SPOOLER ou telefonou para os nossos serviços a saber se este número já tinha sido publicado.

A resposta dos leitores ao nosso passatempo **O melhor jogo** foi espectacular. Vários jogos de todos os tipos e em todas as linguagens chegaram até nós. No próximo número daremos a conhecer os resultados desse passatempo, assim como um novo tema para outra iniciativa semelhante.

Franco Gomes

2 Informática em Notícia

4 Livros

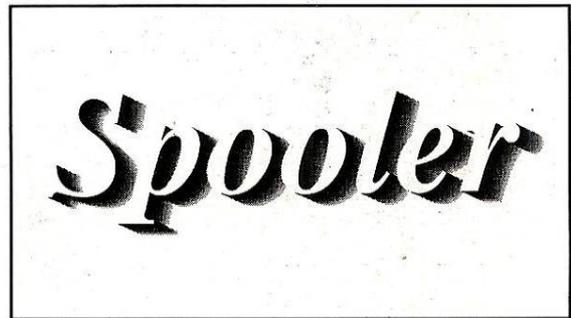
O conteúdo da disquete
Os passatempos Spooler

8 Um teclado patriota

10 Anatomia do disco rígido
(II parte)

16 Pascal: para ensinar e aprender

21 «ADD-INS» — vitamina
para o Lotus 1.2.3.!



24 PostScript — a linguagem que produz
a Spooler

30 Contas em dia — um programa
para gerir a sua conta bancária

32 Visualização Científica
— Curvas de nível

36 Modems

38 Vagueando pelo Informix SQL

40 O disco por dentro e por fora

43 Ganhe um PSION

44 Montra Spooler

Spooler

- **Propriedade:** Graficria, Publicidade e Artes Gráficas, Lda.
- **Director de Edição:** Franco Gomes ■ **Director:** Eng.º Miguel Vitorino
- **Redacção:** F. Serãos, João Cardoso, Luís Alho, Paulo Pinheiro, Vicente Gracias e Eng.º Vítor Guerreiro
- **Colaboradores:** Armando Santos, Júlio Anjos, Paulo Bernardo, Pedro Ferreira, Renato Casquilho e Vítor Hugo Marques
- **Ilustrações:** Franco e Leonor Gomes
- **Relações Públicas e Publicidade:** Jorge Parreira
- **Serviços Administrativos e Assinaturas:** Rua Alfredo Roque Gameiro, 21-1.º Dt.º, 1600 LISBOA — Tel. 76 27 32
- **Produção Gráfica:** Graficria, Publicidade e Artes Gráficas, Lda.
- **Impressão:** Empresa Litográfica do Sul, S.A. ■ **Distribuição:** Electroliber
- **Revista mensal** ■ **Preço:** 400\$00 ■ **SRIP N.º** 213900
- **Tiragem:** 12000 exemplares ■ **Dep. Legal N.º:** 28587/89

MEGASIS OFERECE SERVIÇOS EM SISTEMAS DE INFORMAÇÃO

A Megasis – Sociedade de Serviços e Engenharia Informática, SA anunciou recentemente o seu primeiro leque de serviços no domínio dos sistemas de informação, que constitui uma proposta global e completa, podendo ser implementada de uma forma modular e aberta, recorrendo às mais recentes técnicas e metodologias, e ao apoio dos mais conceituados técnicos nacionais e estrangeiros.

O Departamento de Estudos e Soluções Informáticas (DESI) da Megasis põe, assim, à disposição dos seus clientes uma proposta que engloba três grandes tipologias de serviços: Elaboração de Diagnósticos/Estudos de Oportunidade (DSI), Criação do Plano Director de Sistema de Informação (PDSI) e Engenharia de Sistemas de Informação (ESI).

NOVO COMPUTADOR DA APPLE

O Macintosh IIfx, recentemente anunciado, é o mais potente Macintosh até hoje desenvolvido pela Apple. Funcionando a 40MHz, dispõe da mais avançada tecnologia e apresenta ganhos de performance quatro vezes superiores quando comparado com os modelos anteriores da linha modular Macintosh II.

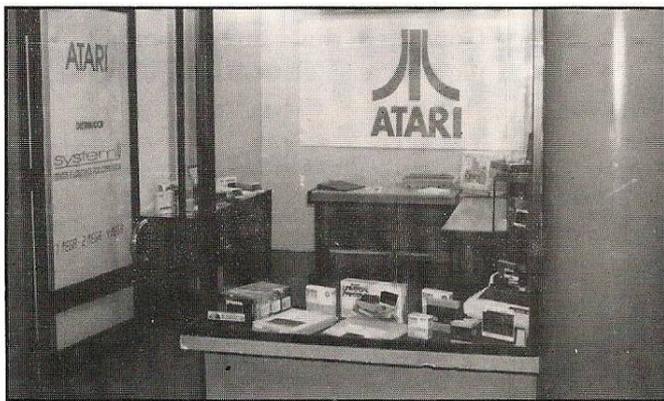
A combinação do microprocessador Motorola 68030 e do coprocessador para cálculo em vírgula flutuante Motorola 68882, a 40MHz, é a mais significativa das suas características. O sistema utiliza memória SRAM para corresponder à frequência aumentada do processador. Para além disso, o código na memória ROM explora integralmente os 256 bytes

de memória cache para instruções do 68030.

O Macintosh IIfx dispõe de 512K de memória ROM que oferecem suporte para as suas características físicas e para endereçamento em 32 bits. A memória ROM inclui ainda suporte QuickDraw de 32 bits, permitindo aos sistemas policromáticos a visualização simultânea de mais de 16 milhões de cores, mediante a utilização de placa com suporte gráfico adequado.

O Apple Macintosh IIfx abre as portas não só a aplicações tecnicamente sofisticadas, como também a aplicações interactivas em 3D e à manipulação de imagens com qualidade gráfica superior.

I.O.M. APOSTA NA INFORMÁTICA PARA TODOS



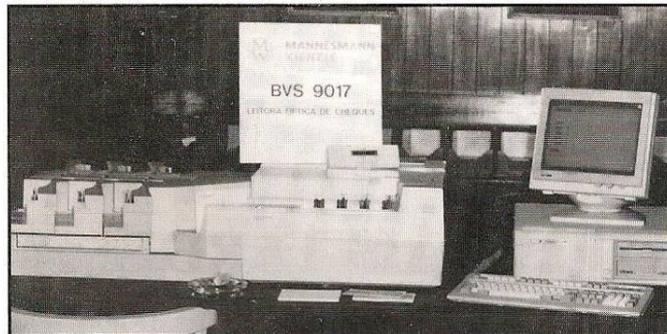
Loja I.O.M. No Centro Comercial da Mouraria

A I.O.M. – Informática, Organização e Métodos, Lda. inaugurou uma nova loja no Centro Comercial da Mouraria, decisão que se insere na nova estratégia da I.O.M. – tradicionalmente dirigida para o mercado dos profissionais – e que visa a diversificação da actividade da empresa, incluindo também o mercado de venda directa ao grande público. Para além dos equipamentos Atari esta loja destina-se também à comercialização

de computadores pessoais compatíveis, como os ACE/Diginova, bem como de uma vasta gama de acessórios destinados a todos os “compatíveis” do mercado.

O Centro Comercial da Mouraria é um espaço recentemente criado, bem localizado e numa zona da cidade que, certamente, se tornará num importante pólo comercial: o Martim Moniz.

EXPOSIÇÕES DE EQUIPAMENTOS MANNESMANN KIENZLE



Leitor óptico/separador de cheques BVS 9017 – Um dos muitos equipamentos apresentados nas exposições da CIL, realizadas no Porto e em Sintra

A CIL – Centro de Informática, Lda. realizou duas exposições dedicadas à divulgação dos produtos da Mannesmann Kienzle para o mercado financeiro, que tiveram lugar no Novotel, no Porto, e no Hotel Tivoli Sintra. Estas exposições tiveram como principal objectivo mostrar a integração desses equipamentos na solução que a CIL tem a correr nas Caixas de Crédito Agrícola Mútuo (CCAM). Tal facto justifica que grande parte dos visitantes fosse proveniente das CCAM.

Os produtos da Kienzle ultrapassam, no entanto, o âmbito das CCAM, estando a despertar o interesse dos bancos e outras entidades que operam com produtos financeiros, dada a grande variedade de opções disponíveis especialmente vocacionadas para *front office*.

A CIL reforça, assim, a sua presença no mercado das CCAM (mais de 160 informatizadas), mas não perde de vista um mercado mais vasto, representado pelas empresas e organismos que trabalham com produtos financeiros.

CYBERS DESTINADOS À SEGURANÇA NUCLEAR

O Instituto Soviético para a Pesquisa e Desenvolvimento de Energia (RDIPE) encomendou recentemente 6 mainframes Cyber 962, num valor global de 32 milhões de dólares, que irão ser utilizados na análise dos factores de segurança na exploração de centrais nucleares civis. A Escolha recaiu neste equipamento Control Data devido à sua aptidão para o tratamento, em larga escala, de dados científicos e de engenharia, sempre envolvidos na análise da segurança nuclear.

Os Cyber 962 irão permitir não só melhorar os níveis de segurança das centrais actualmente existentes, como desempenhar papel importante

no estudo e desenho de novas centrais que vão ao encontro dos standards futuros no que respeita a normas de segurança. Irão, também, ser utilizados na previsão, prevenção e controlo de potenciais acidentes, assim como no controlo da já sinistrada Central de Chernobyl. Caso haja qualquer tipo de acidente, estes equipamentos permitirão a sua correcta avaliação e cálculo de impacte ecológico, assim como de níveis previsíveis de radiação.

Desde 1960 que organizações pioneiras em programas de análise de segurança nuclear utilizam equipamentos Control Data.

EXPOSIÇÃO DE SOFTWARE DE GESTÃO

A TI – Tecnologia Informática, Lda., empresa sediada em Lisboa e produtora de software de gestão, levou a efeito no Porto uma exposição dos seus produtos. Esta exposição foi bastante participada pelas cerca de 30 empresas de informática que tiveram oportunidade de ver o software produzido pela TI que, imediatamente, assinou acordos de comercialização com 11 revendedores.

Os responsáveis pela empresa orientam a estratégia para 1990 segundo três vectores base: consolidar a sua posição de mercado na zona sul; penetrar na zona norte, onde pretende apresentar uma versão alternativa ao software normalmente comercializado nessa região e lançamento de todos os seus actuais packages em versões para Unix.

HICOM EM ESPANHA

O sistema de comunicação Hicom, de tecnologia digital, concebido para a rede ISDN e até há pouco tempo exclusivamente fabricado na Alemanha pela Siemens AG, passou a sê-lo também em Espanha, nas fábricas da empresa Tecosa – Telecommunicación Electrónica Conmutación, S.A. A Tecosa já se dedica há algum tempo ao fabrico de centrais telefónicas de

tecnologia Siemens, nomeadamente de sistemas da linha EMS, para médias e grandes empresas.

De entre os projectos que contribuíram para a obtenção deste resultado salientam-se o Governo Regional do País Basco, a RENFE – Ferroviária Estatal Espanhola e, ainda, a Feira Mundial "Expo'92", em Sevilha.

OPEN-SERVER UNIX

A Wang Laboratories Inc. anunciou, no início de 1990, o lançamento de um conjunto de produtos hardware e software para Unix, disponíveis a partir do 2.º trimestre, onde se incluem os dois primeiros membros de uma nova linha de minicomputadores Wang para Unix: a família Open/Server. Baseados em microprocessador com tecnologia Intel e sistemas operativos da Santa Cruz Operation, estes novos sistemas suportam até 128 utilizadores.

Foi igualmente anunciada a disponibilidade de 2 packages de software: o Unix Clearview, que proporciona ferramentas para gestão de ficheiros e aplicações para o ambiente de *windows* em Unix; o WP/X, que transporta as potencialidades do tratamento de texto Wang para os sistemas Unix. À disposição dos utilizadores estarão também as tecnologias-chave de tratamento de informação em sistemas Unix, que poderão funcionar como *hosts* multiutilizador e como *servers* de redes.

As plataformas estratégicas da Wang incluirão: a nova família Open/Server, de médio porte; sistemas Wang VS; PCs compatíveis com DOS, OS/2 e Unix.

Ao funcionar no ambiente *client/server*, qualquer destas plataformas suportará *clients* na generalidade das LAN disponíveis actualmente. Estas plataformas irão funcionar como *servers* de imagem, fornecendo imagens a PCs e diversos fabricantes.

Nesta arquitectura, a Wang irá apresentar 3 gamas de famílias: na gama baixa serão PCs compatíveis configurados com terminais assíncronos de baixo custo; a gama média será constituída pela família Open/Server, suportando até 128 utilizadores; na gama alta os sistemas Open/Server irão incorporar a tecnologia de multiprocessadores simétricos que aumentarão a capacidade de multiprocessadores simétricos e, em consequência, a capacidade de processamento disponível.

NOVOS MODELOS E OPÇÕES IBM PARA A LINHA PS/2



A Companhia IBM acaba de anunciar novos modelos e opções para os Personal System/2, cujas elevadas capacidades e performances proporcionam uma plataforma para os anos 90.

O SCSI – Small Computer System Interface, anunciado nesta ocasião, permite a ligação de sete dispositivos periféricos a um único slot de expansão, garantindo deste modo a flexibilidade de crescimento dos seus clientes. A versatilidade da arquitectura IBM Micro Channel permite aos seus utilizadores adicionar os dispositivos SCSI aos sistemas já instalados, usufruindo assim das capacidades e performances dos modelos agora anunciados.

Foram ainda anunciados pela IBM os seguintes produtos:

- O IBM PS/2 modelo P70-031, uma versão de 30 megabytes do modelo 70.
- A IBM LaserPrinter 4019E.
- Um novo ecrã IBM PS/2 Color Display 8515 de alta resolução.
- Quatro novos modelos do IBM PS/2 8580.
- Dois novos modelos do IBM PS/2 8565.
- Vários dispositivos IBM PS/2 SCSI.

A combinação deste produtos oferece uma solução global para qualquer utilizador que necessite de grandes capacidades de armazenamento e cálculo em sistemas centralizados ou distribuídos, sendo simultaneamente uma oferta para os utilizadores como primeira máquina, ou melhoria de sistemas já instalados, proporcionando sempre um alto nível de qualidade e performance.

CHIP DE MEMÓRIA DE 16 MEGABITS

A Companhia IBM anunciou o fabrico de um chip de memória de 16 milhões de bits (megabits), utilizando uma linha de produção de semicondutores já existente.

O DRAM (Dynamic Random Access Memory) de 16 megabits tem uma capacidade de armazenamento quatro vezes superior à dos mais avançados chips de memória já existentes. Na mesma ocasião foi também apresentada uma nova família de circuitos integrados de aplicação específica ASIC (Application-Specific Integrated Circuit), chips lógicos capazes de sustentar até 300000 circuitos, os mais densos anunciados até hoje.

O chip de 16 megabits é o mais recente avanço no contínuo desenvolvimento, por parte da IBM, de chips de memória com maior densidade e rapidez, para computadores. Esta companhia abriu o caminho com a recente introdução de DRAMs de 1 milhão e de 4 milhões de bits.

CIL APRESENTA SÉRIE M DA VICTOR

A CIL – Centro de Informática, Lda. acaba de lançar no mercado português uma nova gama de microcomputadores da Victor: a série M, destinada a utilização profissional.

A série M é constituída por modelos "Mini-Tower", estando desde já disponíveis quatro (V86M, V286M, V386MX e V386M), que apresentam diversas configurações possíveis para cada um dos modelos.

Uma das inovações de a série M apresenta é um "verdadeiro" Add Pak que, pela sua capacidade de armazenamento de dados e pelas reduzidas dimensões que apresenta, é a solução ideal para guardar cópias de segurança em cofres ou para transportar informação de uma máquina para outra. Como tal, o Add Pak3 é compatível com todos os modelos da série M.

Outra vantagem anunciada pela Victor é a de um "silêncio absoluto" do equipamento bem como a possibilidade de aumentar as suas capacidades recorrendo apenas à substituição da placa principal.

Todos os "Mini-Tower" da série M



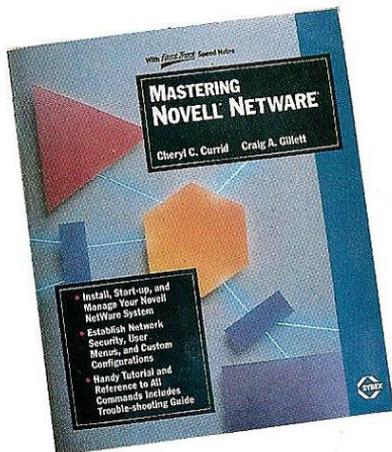
Um dos equipamentos da Série M da Victor

estão equipados com: RAM variável entre 640K e 8Mb, disco duro de 20 a 120Mb, leitor de 3,5" de 720K a 1,44Mb, velocidade de 10 a 25MHz. Acompanhando a evolução do mercado informático e a exigência por parte dos utilizadores, a Victor alia na série M as tecnologias de ponta ao conforto do ambiente profissional, pelo que recorreu às mais

avancadas técnicas de integração, com vista a produzir um equipamento potente, mas que ocupa uma área não superior a um arquivo de pequeno porte.

É de realçar que, tendo em vista a realidade europeia de 1992, a Victor, como fabricante europeia, produz esta nova gama de equipamentos no interior da CEE.

MASTERING NOVELL NETWARE



Mastering Novell NetWare é um guia conciso, fácil de utilizar para configurar, instalar e operar um computador que utilize uma rede Novell. Quase todas as funções importantes, comandos e utilidades disponíveis em rede são demonstrados passo a passo. Este livro está dividido em seis partes. Na primeira são revistas as noções fundamentais do funcionamento em rede. Esta parte explica a terminologia LAN, como a rede comunica os dados, e os vários componentes usados numa rede comum.

A Parte II, capítulos 5 a 8, debruça-se sobre a preparação para instalação e, ainda, configuração e inicialização do sistema operativo em rede. Uma lista de verificação da instalação realça os

passos necessários para, com sucesso, activar um sistema em rede.

Na Parte III, capítulos 9 a 14, é desenvolvido o ambiente da utilização em rede. Também aqui são destacadas as várias técnicas, básicas e avançadas, que poderão ajudar a melhorar a funcionalidade da rede.

Em seguida, na Parte V, capítulos 15 e 16, o tema é o utilizador da rede. O capítulo 15 ilustra o modo de entrar em rede pela primeira vez e descreve o processo de imprimir através da rede. O capítulo 16 explora as utilidades desenvolvidas para o utilizador.

Nos quatro capítulos seguintes, Parte V, discute-se a manutenção e as possibilidades de expansão da rede. Esta parte indica onde procurar os problemas e como corrigi-los, realçando os utilitários existentes para este efeito.

Os dois capítulos finais, Parte VI, sintetizam os comandos utilitários.

Este manual é constituído, ainda, por dois apêndices: A, que descreve a NetWare 386, a mais recente e poderosa versão do sistema operativo Novell; B, glossário da terminologia de rede.

Este livro destina-se a todos aqueles que necessitem instalar, manter operativa ou, simplesmente, utilizar uma rede local, constituindo um apoio fundamental para a rápida e eficiente resolução de problemas e esclarecimento de dúvidas.

Título: Mastering Novell NetWare
Autores: Cheryl C. Currid e Craig A. Gillett
Edição: Sybex
Distribuição: Lidel, Edições Técnicas, Lda. Telefons: 53 44 37-57 59 95-55 48 98

O CONTEÚDO DA DISQUETE

A disquete da Spooler n.º 8 não necessita de grandes explicações. Estas encontram-se na própria disquete e nos artigos respectivos. São três as directorias, a saber:

- BANCO
- PORTUGAL
- CNIVEL

- O programa BANCO.EXE, que deve copiar para outra disquete ou disco rígido, vai permitir-lhe gerir eficazmente a sua conta bancária. Encontra todas as explicações necessárias no artigo sobre o assunto.

- PORTUGAL, é um programa que põe o seu teclado a falar português. Se quiser saber mais, leia o artigo.

- Sabe o que é Visualização Científica? O artigo com o mesmo nome talvez o esclareça. Copie o ficheiro CVNIVEL.EXE do directorio CVNIVEL para outra disquete, ou disco rígido, e a partir daí execute-o. Depois da descompactação de vários ficheiros, execute o programa curvas. Note-se que este programa só corre em placas EGA ou VGA. Um rato compatível Microsoft também é necessário.

O MELHOR JOGO

PASSATEMPO SPOOLER

A resposta dos leitores a este passatempo foi imediata e espectacular. Recebemos várias dezenas de jogos que estão a ser criteriosamente analisados. Já no próximo número da Spooler serão divulgados os resultados deste passatempo, assim como um novo tema para motivar a criatividade e engenho dos nossos leitores.

GANHE UM PSION

E pronto, o primeiro PSION vai a caminho de Lisboa, Rua 6 - n.º 42 - Bairro da Encarnação, que é a morada do nosso assinante n.º 547 - LUÍS MANUEL VIEIRA GASPARGAR, que foi o feliz contemplado com este extraordinário computador de bolso.

DÊ AO SEU COMPUTADOR UM PLOTTER

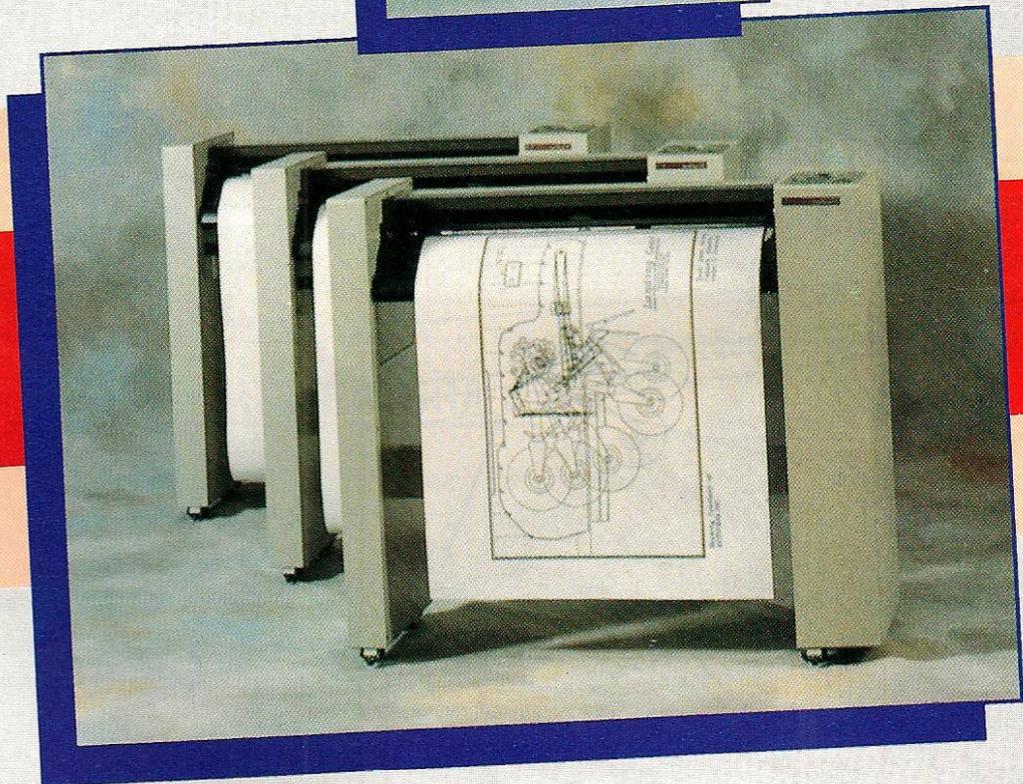


HEWLETT
PACKARD

NOVOS MODELOS AØ

CARACTERÍSTICAS:

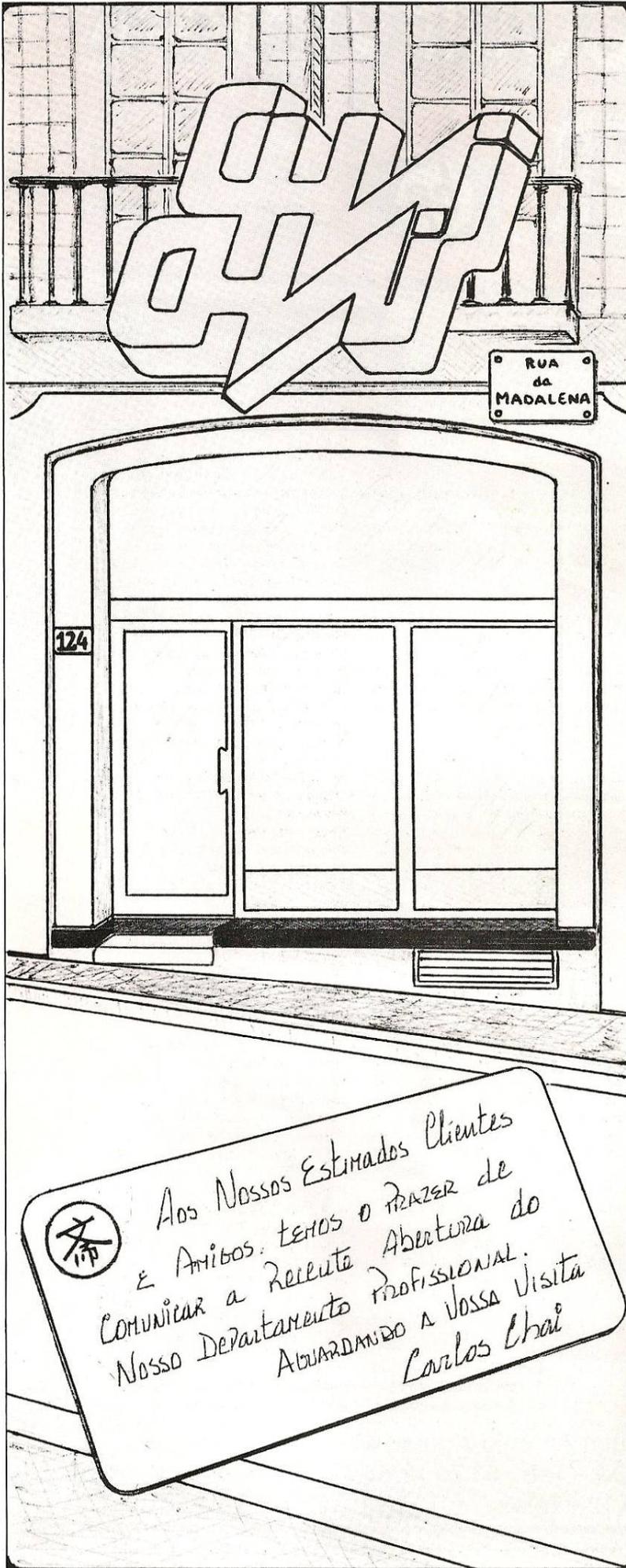
- 8 canetas
- Alimentação c/ ou s/ rolo
- Formatos A4 a AØ
- 1 Mb RAM
- Aceleração 5,7 g
- Resolução 0,006250,025 mm
- Disco rígido 20 Mb (Mx)



CONCESSIONÁRIO
AUTORIZADO

DIMECRO
Representações Comerciais e Técnicas, Lda

Rua Joaquim António Aguiar, 41 - 6º Esqº
1000 LISBOA - Tels.: 53 76 12/53 79 49
Fax: 53 76 12 - Telex: 14093



BBS

EM NOTÍCIAS

A CATS-BBS tem-se afirmado cada vez mais como a maior e a melhor BBS de Portugal (e não só). Para além de ter mais utilizadores do que o total de utilizadores das outras BBSs juntas possui ainda maior número de ficheiros, apoio ao utilizador (infelizmente, a CATS-BBS é a única que dá este privilégio aos utilizadores), etc.

Há cerca de dois meses foi ultrapassada a barreira dos 1500 utilizadores, o que é notável tendo em conta que Portugal foi sensibilizado para o mundo das telecomunicações tarde de mais.

A confirmar a nossa expansão, a CATS-BBS está a ser amplamente falada nos Estados Unidos, conforme nos foi dito por vários novos utilizadores oriundos desse país. Inclusivamente, será tema de um artigo para a maior revista inglesa de computadores tal como, possivelmente, para uma ou duas revistas americanas.

Falando especificamente de novidades, temos que está pronto um manual de utilizador com 24 páginas que é fornecido gratuitamente a todos os utilizadores registados explicando passo a passo como operar com a BBS. Foi também lançado o "Jornal do Sistema" sendo este um pequeno jornal com 12 páginas onde se encontram notícias de interesse para quem estiver inserido no mundo das telecomunicações. Só a título de exemplo aqui vão alguns dos temas tratados neste número: um artigo sobre a base de dados da CEE, a Rede Digital de Integração de Serviços, estatísticas de uso da CATS-BBS, entrevista com um dos nossos utilizadores, análise de novo software chegado, etc.

Foi recentemente adquirido um disco de 350Mb para comportar os ficheiros que temos, sendo de esperar que, quando este artigo sair, um segundo disco de 350Mb esteja instalado. Este facto vai-nos proporcionar incluir os cerca de 200Mb em ficheiros que temos em stock, isto além dos cerca de 5000 ficheiros de que actualmente dispomos! Foi firmado um acordo com um SysOp dos Estados Unidos que nos permitirá SEMANALMENTE termos as últimas novidades em ficheiros "Public Domain" e "Shareware" na CATS-BBS. Isto fará com que recebamos todas as semanas cerca de 10Mb de ficheiros (o que será alguma coisa entre 200 e 400 ficheiros novos) e isto todas as semanas!!!!

Para evitar que a CATS-BBS seja monopolizada por utilizadores a capturarem as últimas novidades para os respectivos discos foi implementado um sistema de venda de software em disquetes com preços bastante baixos de maneira a pagar apenas os portes de correio e o trabalho da cópia.

Pensando nos novos utilizadores, está disponível um conjunto de duas disquetes com as respectivas instruções que contém todo o tipo de software que será necessário para o correcto uso da CATS-BBS.

E quanto é que todo este potencial custa? Concorde conosco certamente que os preços por nós praticados são absolutamente razoáveis tendo em conta os elevados custos de manutenção de toda esta máquina que é a CATS-BBS. Assim, os preços já com IVA são os seguintes :

- Registo Trimestral - 2 340\$00
- Registo Semestral - 4 095\$00
- Registo Anual - 7 020\$00

A estes preços há que acrescentar a jóia no valor de 2 340\$00 estando todos os estudantes isentos deste pagamento.

Note-se que para entrar na CATS-BBS não é necessário registar-se, cabendo essa decisão apenas ao utilizador que, após algum tempo de uso, decidirá se vale ou não a pena registar-se. As vantagens para os utilizadores registados são as seguintes:

- Passa a dispor de uma hora de utilização diária (contra 25 minutos de não registados)
- Pode puxar cerca de 250Kb de ficheiros diariamente
- Acesso a todos os programas externos (portas)
- Acesso a todas as conferências
- Receber o manual de utilização
- Receber mensalmente o Jornal do Sistema
- Desconto na compra de material informático em diversas lojas
- Desconto de 20% na compra de software por correio
- Desfrutar de todas as novas iniciativas que aparecerão

A decisão é sua: se quiser estar por dentro da revolução das telecomunicações que se está a processar em Portugal não poderá passar sem uma visita à nossa BBS.

OS SYSOPS
(operadores de sistema)

CATS-BBS

352 64 22/3/4/5/6, notícias diárias,
 jogos multi-utilizador em linha

Telepac: 005 229 048

cerca de 5000 ficheiros - notícias diárias

Correio ecoado mundialmente
 programas multiutilizador em linha

Correio dos Leitores

por Miguel Vitorino

O volume de cartas que tem chegado à nossa redacção tem sido de tal modo elevado que nos é totalmente impossível responder a todas as cartas através desta nossa coluna. Tentaremos, no entanto, colocar aqui o maior número de cartas cujo conteúdo possa abranger perguntas colocadas por vários leitores. Peça que continuem a enviar-nos as vossas críticas – boas ou más – para que, com a vossa colaboração, possamos melhorar esta vossa revista. Mais uma vez lembro que as rotinas e/ou artigos que nós achamos terem qualidade suficiente serão impressos na Spooler (recebendo os seus autores a respectiva recompensa monetária).

Respondendo ao convite feito na revista n.º 6, aqui envio algumas sugestões e também algumas dúvidas que gostaria de ver respondidas. Aproveito para vos felicitar pelo excelente trabalho realizado e espero que mantenham a qualidade a que nos habituaram.

– É possível ter num único disco rígido, várias partições, cada uma com um sistema operativo diferente? Como fazer para mudar de sistema operativo em que o computador começa a trabalhar? Que é exactamente uma partição? É uma partição activa? Ao executar o comando Format, todo o disco é formatado ou apenas a partição especificada? O que é a RAM CMOS? (Que tal fazer um artigo que respondesse a todas estas perguntas?)

– O filtro para monitor vendido na Montra Spooler necessita de estar ligado à terra? Quais as vantagens? Pode ser utilizado sem estar ligado à terra? Quais as desvantagens?

– Qual a função dos comandos Share, Exe2bin, Recover e Link?

– Seria possível a inclusão, nas disquetes, de utilitários public domain tais como emuladores de CGA para Hercules, compressores de ficheiros (como os utilizados para comprimir os programas das Spooler, números 2, 3 e 5), vacinas e programas antivírus?

– Onde se meteram as imagens da revista n.º 6?

– Uma vez que os GIFs para carta gráfica VGA, devido a serem excessivamente grandes, não podem ser incluídos nas disquetes Spooler, há alguma possibilidade de eu as conseguir dirigindo-me à redacção da vossa revista?

– Qual o preço aproximado de um co-processor aritmético 80287 (para um Amstrad PC 2286)? Onde o posso adquirir? Quais as vantagens dessa aquisição?

– O disco rígido do Amstrad P2286 (40Mb) tem auto-park?

– Seria possível a publicação de um teste ao Amstrad PC2286 com disco de 40Mb e monitor policromático de alta resolução (12")?

– Desde já agradeço a atenção que me dispensaram, apresentando as minhas desculpas pelo número de perguntas, bem como pela extensão de algumas.

JOÃO COSTA
(Castelo Branco)

O nosso leitor João Costa coloca-nos muitas perguntas, algumas de bastante interesse para muitos dos leitores. Embora em certas questões esteja longe

de ser um perito, tentarei responder a tudo.

– Primeiro que tudo: uma partição é uma porção do disco que é identificada por uma dada drive, podendo o disco ter, teoricamente, um número elevado de partições.

Quanto a ter dois sistemas operativos no disco: Sim, é possível. O truque é o seguinte: faz-se a formatação com um utilitário como o "Disk Manager" (que costuma vir a acompanhar os discos) ou, então, com uma das várias versões do "Fdisk" feitas por fabricantes (como seja a Olivetti) que permitem várias partições num mesmo disco. A partir daí põe-se na primeira partição um primeiro sistema operativo e pronto! Quando arrancar com o computador ele vai arrancar com essa versão a partir da primeira partição.

Agora suponha que queria numa segunda partição uma outra versão do MS-DOS, por exemplo. Nesse caso, basta executar o comando "Fdisk", alterar o número da partição activa (que nos indica qual a partição por onde o computador vai arrancar) colocando como activa a segunda. De seguida, arranque com uma disquete com a nova versão do MS-DOS, faça um "SYS C:" (garantindo que essa partição estava limpa) e da próxima vez que arrancar o sistema será por essa nova versão. A partir daí para mudar de Sistema Operativo, terá que seleccionar a partição activa no "Fdisk" e re-arrancar com o sistema.

Tudo o que foi dito refere-se apenas a MS-DOS pois se quiser por exemplo, instalar MS-DOS e Xenix numa mesma máquina, o procedimento será totalmente diferente!

Quanto à RAM CMOS para não me alongar mais direi que é um banco de memória que costuma ter uma pilha para nunca se perder o seu conteúdo onde figura toda a configuração da máquina. Obrigado pela sua sugestão sobre um artigo dedicado a estas questões; pode bem ser que em breve este seu desejo se tome realidade.

– O filtro anunciado na Montra Spooler NÃO necessita, de forma alguma, estar ligado à terra. Embora não seja francamente um especialista nesta matéria, posso adiantar que dado o material de que o filtro é feito é desnecessária a ligação à terra.

Resumidamente:

– SHARE - Permite partilhar drives entre vários computadores ligados em rede.

– EXE2BIN - Converte um ficheiro com

extensão .EXE para um com extensão .COM poupando, assim, em espaço no disco.

– RECOVER - Permite a recuperação de ficheiros que tenham sectores danificados.

– LINK - Este é algo mais complexo de explicar. Basicamente permite "juntar" vários ficheiros com a extensão .OBJ (gerados por assembladores e compiladores) e .LIB num único executável com extensão .EXE

– Caro João: tem-nos sido difícil pôr todo o material dos últimos números da revista na disquete, pelo que o que nos pede é totalmente impossível. Caso esteja interessado nesse tipo de software contacte, por exemplo, a CATS-BBS onde encontrará virtualmente todo o software de que necessitará. Poderá, inclusive, pedi-lo por correio como é anunciado noutra parte desta revista.

– Quanto ao disco rígido do Amstrad P2286 penso que tenha auto-park, mas peço que confirme junto do representante em Portugal que lhe poderá dar esse tipo de informação com total fiabilidade.

– Infelizmente não é possível esse tipo de teste. Os testes que fazemos a máquinas ocorrem apenas quando pensamos que apresentam alguma característica que as distingue da competição e terá que ter alguma actualidade. A máquina que menciona não preenche nenhum dos anteriores parâmetros.

Possuo um Commodore Amiga 500.

Gostava de me fazer sócio da Spooler mas, antes, gostava que me dissesse se o conteúdo da disquete e se os jogos que a revista vende dão para o mesmo aqui referido. Um amigo meu gostava de saber o que é a assinatura com CATS-BBS.

Agradeço uma resposta na revista.

JOÃO LUÍS CARTAS ROSADO
(Ponte de Sor)

Quanto à questão do Amiga: se se estiver a referir aos programas fonte então não terá problemas em lê-los. Os executáveis é que apresentaram problemas pois foram feitos a pensar exclusivamente no mundo de IBM PCs. No entanto, poderá sempre tentar compilar as fontes dos programas que apresentámos que apenas precisarão de alguns retoques para funcionar nesse tipo de computador.

Quanto à questão do amigo: CATS-BBS é um serviço de telecomunicações fornecidas por modem. Inclusive tem uma área dedicada aos computadores da família Amiga com várias centenas de ficheiros. Através da assinatura da Spooler conjuntamente com a CATS-BBS além de receber a revista Spooler durante 12 números poderá ainda aceder à CATS-BBS durante um ano.

O meu primeiro contacto com a vossa revista (n.º 6, Janeiro 1990) teve dois aspectos perfeitamente opostos. Em primeiro lugar, gostaria de vos dar os parabéns por uma revista cujo conteúdo e apresentação são de louvar. De facto, gostei muito dos artigos apresentados e dos miniprogramas que a disquete incluiu.

Em segundo lugar, e esta é a principal razão pela qual escrevo esta carta, diz

respeito exactamente à disquete que acompanhava a revista. Por muito estranho que pareça a qualquer profissional de informática, como é o meu caso, essa mesma disquete vinha acompanhada de um vírus. Exactamente, o vírus Ping-Pong tipo B, mais vulgarmente conhecido como "italiano saltitante". É muito estranho que a revista, que já apresentou um artigo sobre este assunto, venha agora distribuí-lo.

Enquanto for um vírus inofensivo como este, o problema de desinfectar disquetes e discos duros não é muito grande. Quando for outro qualquer, de consequências mais nefastas, não sei onde iremos parar (porventura, deixaremos de comprar revistas que tenham disquetes).

Gostaria que entendessem esta minha nota apenas como uma chamada de atenção e não como uma crítica destrutiva.

Queria referir mais dois erros que encontrei no programa Revistas, em dBASE. Quando pretendemos corrigir registos (Procedure Revent4), ao pedir para corrigir mais um registo (depois de termos corrigido o primeiro) o programa dá erro na linha 171. Tem a ver com o facto de a variável "num" ter sido definida como string e depois transformada em numérica através da função VAL. Ao repetir o DO...WHILE, a mesma transformação já não é possível, pois a variável já é numérica, dando erro. Pode resolver-se o problema passando a linha num=SPACE(4) para a linha abaixo do DO...WHILE.

Na mesma procedure, quando premimos a tecla Escape, como essa mesma variável está definida com valor nulo, o programa ao fazer o GOTO num, dá erro novamente.

Estes erros foram os que encontrei após dois minutos de utilização do programa. Se calhar existem mais.

Esperando que esta minha carta seja bem recebida e que os maus funcionamentos e vírus não se repitam, despeço-me.

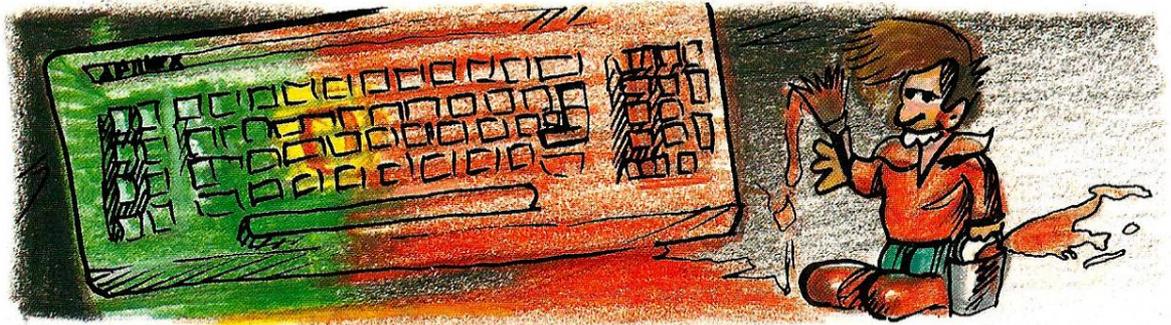
Atenciosamente

ROBERTO SILVA
(Lisboa)

Ó Sr. Roberto Silva, dizer que nós infectámos uma disquete da nossa revista?! Só para seu esclarecimento: NENHUM dos milhares de compradores dessa revista se tinha queixado disso e eu inclusivamente (sou o responsável pela feitura da disquete) tenho-a corrido múltiplas vezes no meu computador e nem sombra de vírus! E não pense que é por falta de atenção pois tenho ao meu dispor dezenas de detectores e antídotos de vírus. O seu problema é que ANTES de usar a nossa disquete já a disquete ou computador estariam infectados. Gostaria que me confirmasse depois disso, ok? Não deprenda pelas minhas palavras alguma má-fé ou raiva. Apenas não se precipite a dizer coisas para as quais não tem provas.

Quanto às correcções que menciona tem toda a razão em ambos os casos ficando já aqui a correcção para todos aqueles que tiverem usado este programa.

UM TECLADO PATRIOTA



“
**Quem apenas
 deseje utilizar
 esta pequena
 preciosidade
 escusa de ler
 este artigo e
 correr o
 respectivo
 programa que
 se encontra na
 Spooler deste
 mês.**
 ”

Por acaso o leitor não tem, tal como eu, um teclado do tipo americano ou inglês? Nesse caso, se quiser escrever um ‘c’ cedilhado como fará? Neste ponto alguns leitores franzem a testa como sinal de ignorância e outros, mais conhecedores, dirão logo: “Ora, basta consultar uma tabela ASCII!” Correcto, mas pouco prático. Para consultar uma tabela ASCII sempre que se quiser por uma letra acentuada perde-se muito tempo desnecessariamente.

O ideal talvez fosse mesmo ter um teclado português ou... “algo” que o emulasse. Acho que o leitor já descobriu que o motivo deste artigo não é mais nem menos que um emulador de teclado português.

Para quem não conhece este tipo de teclado acrescentarei que, por exemplo, para se ter a letra ‘a’ com acento basta carregar primeiro na tecla com o tipo de acento desejado e depois teclar a tecla ‘a’. Nada mais simples e prático!

Este emulador trabalha exactamente da mesma maneira, mas tem algumas alterações, nomeadamente: o ‘c’ cedilhado é obtido através da tecla de vírgula seguida do ‘c’, pode ainda obter-se os caracteres ‘ç’ e ‘ã’ bastando carregar na tecla de “underline” seguida de ‘a’ ou ‘o’ respectivamente e, ainda, quem escrever em espanhol dispõe do comum carácter ‘ñ’.

Como pode ver trata-se dum emulador que, além de copiar o original, implementou algumas características por ele esquecidas.

Adicionalmente, supondo que você carrega na tecla de plica e quer mesmo que saia apenas a plica não devendo ser interpretado como acento deverá carregar na barra de espaços depois da plica. O que se disse para a pelica é válido para os outros tipos de acentos.

Note-se que uma segunda vez que correr este programa o desinstalará de memória.

Quem apenas deseje utilizar esta pequena preciosidade escusa de ler o resto deste artigo e correr o respectivo programa que se encontra na **Spooler** deste mês. Para todos os outros aqui vão umas pequenas explicações de como este programa foi estruturado.

COMO???

A ROM BIOS dos PC’s dispõe dum interrupt por software (o número 16 em hexadecimal) que implementa as funções de leitura de teclado. Foi, assim, necessário

implementar um “front-end”, ou seja, uma rotina que correrá antes do código da BIOS para fazer uma pré-interpretação das teclas.

Para complicar um pouco as coisas, a função 0 desse interrupt deverá consumir a tecla que for digitada, enquanto que a função 1 apenas dirá se foi ou não permitida uma tecla e, em caso afirmativo, devolvê-la sem a limpar, contudo, do buffer do teclado. Esta foi a principal causa dos problemas que foram aparecendo aquando do aperfeiçoamento deste programa, dado que todo o código deveria ter em conta estas duas situações devendo, no caso da função 1, devolver sempre o estado do buffer do teclado (vazio ou não).

Depois de alguns *crashes*, lá consegui correr pela primeira vez este programa sem nenhum tipo de problemas.

Basicamente, o algoritmo usado foi o seguinte:

- Ver se existe algum carácter para consumir; se existir, devolvê-lo e sair;
- Caso não exista, chamar a ROM BIOS à espera de uma tecla;
- Comparar a tecla com a de um acento;
- Se não for acento, devolvê-la e sair;
- Se for acento, espera por uma segunda tecla
- Caso essa segunda tecla em conjunção significar um carácter acentuado, devolve-se o respectivo código ASCII;
- No caso contrário é devolvido o primeiro carácter e coloca-se o segundo carácter para consumo posterior (ver início desta explicação).

Simples? Não tanto como parece. Tem que se ter em conta que por exemplo a função 1 nunca poderá consumir o carácter fazendo com que apenas uma posterior chamada à função 1 o consuma. Isto tudo implica técnicas de *bufferização* para não existirem perdas de teclas.

Mas mais palavras para quê? Quem perceber de *Assembler* poderá ver por si próprio como este programa foi feito, analisando o ficheiro “Portugal.asm” que consta da disquete deste mês.

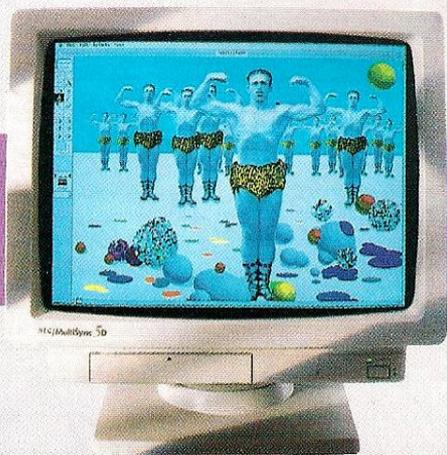
Boas tecladas!

 Miguel Vitorino

DeskTop Publishing

AS MELHORES SOLUÇÕES EM AUTO-EDIÇÃO

CONDIÇÕES
ESPECIAIS
PARA REVENDA

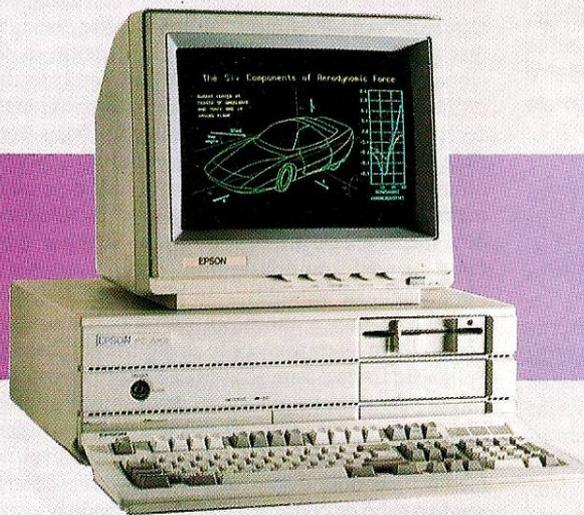


NEC MultiSync® 5D

- Policromático 20 "
- Resolução de 1280x1024

NEC Laser LC 890

- PostScript Adobe original
- 8 páginas por minuto
- 2 tabuleiros de alimentação (250 folhas cada)
- 3 Mb RAM



Epson AX3

- 16/20 MHz
- 2 Mb RAM
- Hard disk 40 / 100 / 150 Mb

NEC

DATACOMP
Sistemas de Informática, Lda.

Av. Prof. Reinaldo dos Santos, nº15 - r/c - D
Telefs: 418 76 80/418 79 19 Fax: 418 86 11
CARNAXIDE 2795 LINDA-A-VELHA

ANATOMIA DO DISCO RÍGIDO

(II Parte)



Nesta segunda parte analisaremos com algum detalhe os diversos tipos de interface de discos rígidos. O campo que se abre é suficientemente vasto para ocupar vários volumes, mas tentaremos sensibilizar o leitor para as diversas vantagens e desvantagens dos vários tipos de interface. De passagem, mencionaremos alguns aspectos que, embora não dizendo estritamente respeito aos interfaces são significativos no conjunto do sistema.

Como vimos na parte anterior, abusa-se muitas vezes da designação de interface, dizendo, por exemplo, que determinado disco rígido é MFM (quando isso não classifica o interface, mas apenas a técnica de codificação dos dados). De facto, a classificação do interface pode ser feita a vários níveis, partindo da cabeça de leitura/escrita até CPU do computador. Interessam-nos nesta fase os interfaces a nível físico daquilo que se costuma designar por disco rígido, ou seja, aquela caixa com uma forma vagamente rectangular dentro da qual se encontram os verdadeiros discos, o motor que os faz rodar, as cabeças de leitura, e os circuitos associados. Essa "caixa" tem, no mínimo, uma ficha de alimentação (tipicamente uma Molex com 4 pinos) e uma ou duas fichas (ou encaixes de circuito impresso) que dizem respeito aos dados e controlo do disco. A este nível, distinguem-se os seguintes principais tipos de interface: ST506, ST412, SCSI, IDE e ESDI. Não nos ocuparemos de variedades mais exóticas no campo dos microcomputadores, como o IPI.

OS INTERFACES ST506/ST412

Estes dois tipos de interface são muito semelhantes, devendo a sua designação aos modelos da Shugart Technologies que pela primeira vez os apresentaram. Assentam numa ligação de dados não codificada, em série (NRZ). Portanto, é função do controlador distinguir se o disco é usado como MFM ou RLL. O controlo do posicionamento da cabeça feito por um sinal de direcção (indicando se a cabeça se deve aproximar ou afastar do centro do disco) e por um impulso de "step" (em que cada impulso avança ou recua uma pista). Surge aqui a diferença prática entre ST506 e ST412, pois que o primeiro requer que os impulsos sejam temporizados segundo a mecânica do disco, enquanto o segundo tem "buffered stepping", ou seja, regista os impulsos, que podem vir a qualquer velocidade, e efectua-os fisicamente à velocidade que a mecânica permite. Estes interfaces possibilitam ainda a selecção de até 16 cabeças por disco. A ligação física é feita através de duas fichas ou encaixes (normalmente estes últimos), um dos quais referente à ligação de dados, com 20 pinos, e o outro referente à ligação de controlo, com 34 pinos. É possível normalmente ligar dois destes discos ao mesmo controlador, sendo o cabo de controlo único e partilhado por ambos os discos, e havendo um cabo de dados por disco.

O cabo de controlo dispõe de sinais identificando o disco a que se pretende aceder, e os discos ST506/412 têm, normalmente, postes ou "jumpers" que seleccionam o endereço, ou seja, a qual dos sinais devem responder. A IBM introduziu uma técnica pela qual o

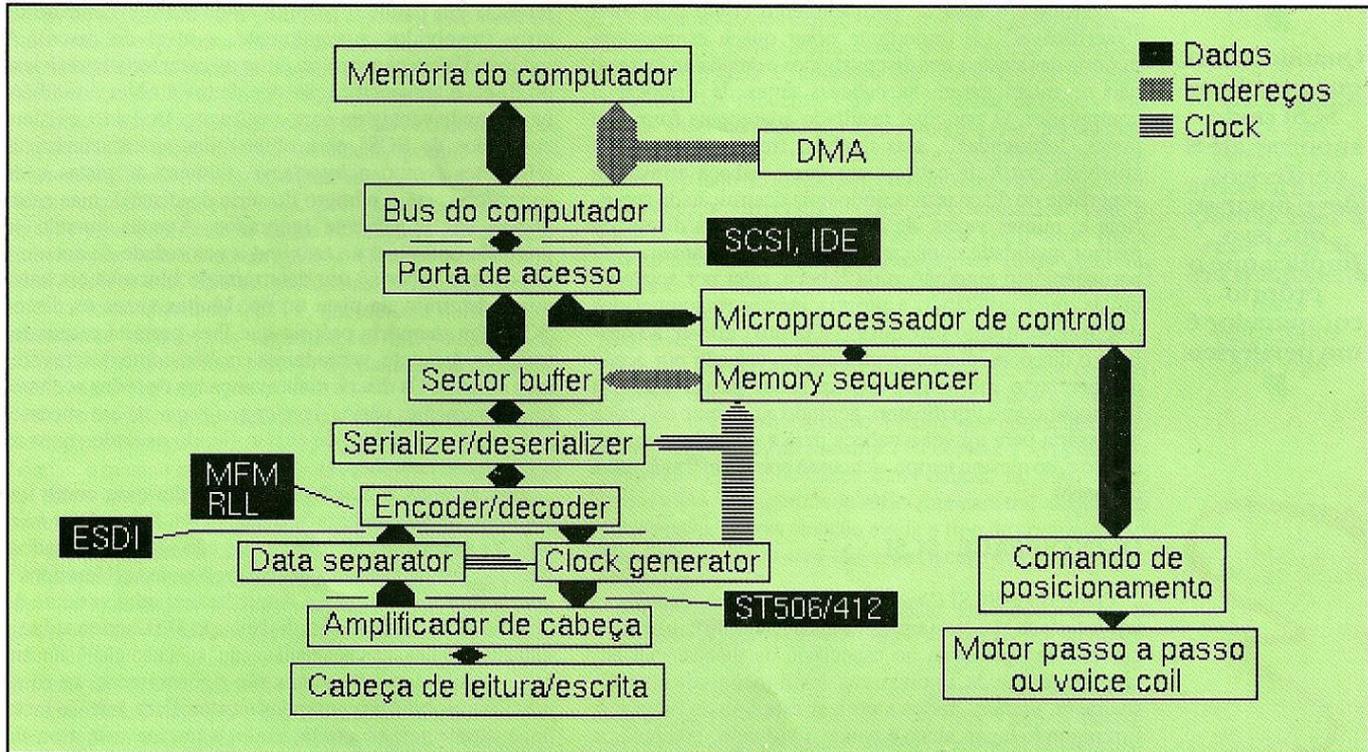


Figura 12 – (Diagrama de fluxo de dados no conjunto do sistema equipado com discos rígidos)

endereço dos discos podia ser sempre o mesmo, e que correspondia a “trocar” a posição dos sinais de ficha de ligação para ficha de ligação. Esses cabos de controlo distinguem-se por apresentarem uma “torcidela” nos fios e, neste caso, o endereço dos dois discos deve ser o mesmo (normalmente o 0). Se o cabo não apresentar essa dita torcidela, os discos têm de ter endereços diferentes. É também o cabo de controlo a conter os sinais de *Dir* (direcção em que se deve deslocar a cabeça), *Step* (ordena um movimento da cabeça) e *Head select* (dos quais vários, para seleccionar as diversas cabeças). O cabo de comando afectado pela chamada terminação, que será descrita adiante.

À ligação do cabo de dados do tipo diferencial existem dois sinais de dados opostos entre si, e é a polaridade relativa dos mesmos que define se o fluxo deve ocorrer numa direcção ou noutra.

Estes sinais são diferenciais para evitar a introdução de ruídos: se ocorrer um impulso “ruidoso”, esse impulso vai afectar em simultâneo ambos os sinais, fazendo-os aumentar ou diminuir de tensão em paralelo, mas não alterando a polaridade relativa. Este tipo de ruído, o chamado *common mode noise*, é o mais frequente. Existem neste cabo de dados dois pares de sinais: um par de escrita e um par de leitura. São os circuitos internos do disco, assistidos pelos sinais do cabo de controlo, a seleccionar qual deles está activo em determinado momento, e a que cabeça diz respeito.

Embora este tipo de interfaces seja o mais divulgado e económico, tem como principal inconveniente limitar a velocidade de transferência de dados: pelo tipo de transmissão de dados, em série; e pelas distâncias envolvidas no comprimento dos cabos, que actuam como verdadeiras antenas de captação de ruídos. Impossibilitam, ainda, a leitura, em simultâneo, de mais de um disco (dado o cabo de controlo ser partilhado). As variações de construção dos discos levam ainda a que seja por vezes complexa a configuração dos controladores, sendo necessário dispor de informações tais como o comprimento dos sectores, o número de faces, o número de cilindros, o número de

sectores por cilindro, etc. Em contrapartida, dada a sua grande divulgação e tecnologia bem conhecida, estes interfaces são fiáveis e económicos.

Alguns “segredos” que normalmente se encontram associados a estes interfaces são o *reduced write current*, o *write precompensation* e o *parking cylinder*. As duas primeiras expressões referem-se ao facto de as pistas mais interiores dos discos serem mais densamente escritas (mais transições de fluxo por unidade de comprimento). Em certos tipos de superfície, essa densidade pode provocar a saturação da camada magnética a menos que se reduza a corrente de escrita ou, noutros termos, a amplitude do campo da cabeça. Certos discos mais antigos requerem mesmo três intensidades diferentes de corrente, correspondentes a outras tantas áreas de utilização da superfície.

Quando o disco requer esta redução de corrente, ela é expressa em termos da pista ou cilindro a partir do qual se deve entrar em regime reduzido. Outra forma de compensar a saturação consiste em variar ligeiramente os intervalos entre as transições de fluxo, aumentando os tempos em que o fluxo está em “cima” ou em “baixo”. Isto é a chamada *write precompensation*, e pode ser expressa tanto em termos do cilindro a partir do qual deve ocorrer, como na duração da “compensação”.

O *parking cylinder* refere-se ao cilindro em que a cabeça deve ser estacionada quando o disco não está em utilização. Como vimos na parte anterior, a distância entre a superfície do disco e a cabeça é muito pequena, pelo que, caso o disco esteja parado, não gerando, portanto, a corrente de ar que aerodinamicamente mantém a cabeça na posição, qualquer “abalo” pode provocar o choque da cabeça com o disco, com possíveis consequências desagradáveis. Por esse motivo, os fabricantes reservam uma área própria para estacionar a cabeça, que naturalmente se exprime em termos do cilindro ou pista em que isso deve ocorrer. Conforme o tipo de fabrico, assim a chamada zona de estacionamento se encontra na pista inicial ou na final. Em certos casos, intervém um mecanismo próprio de suporte da cabeça, enquanto noutros

“
A distância entre a superfície do disco e a cabeça é muito pequena, pelo que, caso o disco esteja parado, não gerando, portanto, a corrente de ar que aerodinamicamente mantém a cabeça na posição, qualquer “abalo” pode provocar o choque da cabeça com o disco, com possíveis consequências desagradáveis
”

“
**Quando se diz
 que um “bus”
 SCSI pode
 suportar até 8
 periféricos,
 deve notar-se
 que isso
 implica que o
 próprio
 computador é
 um periférico**
 ”

se considera a área de estacionamento como uma área “sacrificável”. É importante notar que a esmagadora maioria das avarias nos discos rígidos é devida ao facto de não se estacionarem as cabeças antes de deslocar o computador. O principal resultado aparece na forma de pistas “estragadas”, mas também frequente ocorrer o chamado *crash* da cabeça, em que a cabeça arrasta na superfície do disco provocando uma destruição da superfície e, muitas vezes, da própria cabeça. Os discos de melhor qualidade começam a dispor de mecanismos de estacionamento automático da cabeça, quer por acção de molas, quer utilizando a própria inércia rotacional dos discos. Este último mecanismo é muito curioso: ao desligar o disco, os discos continuam em rotação por acção da inércia; o motor que os faz rodar passa, então, a funcionar como um dínamo, gerando a corrente eléctrica necessária para mover a cabeça para a zona de estacionamento e, ao mesmo tempo, actuando como um travão para os discos.

O INTERFACE SCSI

O interface SCSI (*Small Computer Systems Interface*) é um interface muito simples de usar, mas difícil de descrever. Para começar, no específico de discos: existem unidades de banda, impressoras, instrumentos de medida, *scanners*, *plotters*, todos a utilizar este tipo de interface. Em segundo lugar, ainda é pouco divulgado, excepto nos microcomputadores mais avançados. E, finalmente, está em fase de redefinição, com o fim de se transformar no SCSI2.

A principal característica deste interface é que é um interface de alto nível. Significa isto que para o computador é irrelevante qual o tipo de construção do disco: se codificado em MFM ou RLL, e se tem muitos ou poucos cilindros ou cabeças. Para o computador, o disco apresenta-se como um periférico de armazenamento de dados com determinado número de blocos de comprimento definido, e cada bloco ter um número de ordem. Se esse bloco está no princípio ou no fim do disco, não interessa, é o disco que tem de o localizar.

Adicionalmente, os dados são transferidos em paralelo, ou seja, um byte de cada vez, e não em série como nos interfaces ST506 (o SCSI2 possibilita a transferência de mais de um byte de cada vez).

O interface SCSI constitui-se como um “bus” ou barramento, em que todos os sinais são partilhados entre todos os dispositivos ligados. Os dispositivos identificam-se por um número ou endereço entre 0 e 7, inclusivamente o controlador. Qualquer dos intervenientes pode iniciar as comunicações, que decorrem numa perspectiva “amo/escravo”: um dos dispositivos (designado por *master*) determina quais os movimentos de dados a efectuar (sendo obedecido pelo *slave*).

Tipicamente, os discos são “escravos”, sendo o interface do computador propriamente dito a actuar como “amo”. Os dados podem ser transferidos em modo assíncrono (em que o receptor tem de confirmar a recepção de cada byte) ou síncrono (em que a confirmação só é feita no fim do bloco, sendo naturalmente este modo o mais rápido). O interface SCSI suporta ainda o chamado “overlapped request”, em que o amo ordena que o escravo localize determinado bloco, enquanto o amo faz outra coisa. Quando o escravo localizar o bloco e estiver pronto a transferir, informa “respeitosamente” o amo. Desta forma é possível aceder a vários discos em simultâneo, sem ter de aguardar pelas pausas da mecânica mais do que o estritamente essencial.

A velocidade possível no interface SCSI, extremamente elevada, dado a codificação e descodificação serem locais ao disco, não envolvendo grandes distâncias de ligação entre a cabeça e o resto do sistema. A trans-

ferência em paralelo permite ainda melhor controlo de erros (resolvidos, normalmente, a nível do disco). A independência conseguida entre as características físicas do disco e a especificação dos dados a obter é também uma grande vantagem para o software. De facto, existem comandos de SCSI para obter todas as características relevantes (como o *interleave*, número de pistas e de superfícies, até o número de série do disco!), mas esses podem ser largamente ignorados. Apenas quando se pretenda otimizar ao máximo a velocidade de acesso é importante saber se um determinado bloco se encontra numa determinada pista ou no. Muitas vezes os discos SCSI têm memória própria que lhes permite responder mais rapidamente, verdadeiros *caches*, como descreveremos adiante. Os discos mais avançados de todos aceitam, inclusivamente, vários *overlapped requests* em sucesso, e optimizam a ordem de acesso (lendo os dados num só movimento de cabeças).

Fisicamente, o interface SCSI foi definido como um conjunto de 50 condutores ligados com fichas DIN *header*. Na prática, os 50 condutores não são todos usados, pois a maior parte deles são de ligação massa (“escudos” contra ruídos). Por isso, a Apple lançou uma variante da ligação SCSI, baseada em fichas tipo D (Cannon subminiatura) de 25 pinos, realizando o “escudo” doutra forma. No entanto, os sinais são rigorosamente os mesmos, assim como os comandos usados. É, de resto, o facto de a Apple ter adoptado exclusivamente este tipo de interface que tem vindo a reforçar a posição do SCSI no mercado. Apenas recentemente o SCSI foi ameaçado, como recordista de velocidade, pelo ESDI. Presumivelmente, o SCSI2 vai recuperar a dianteira.

O SCSI é, por enquanto, algo mais caro do que o ST506/412. Parte da razão está na sua menor divulgação, mas também no facto de implicar a duplicação de circuitos (cada disco tem de ter o seu separador de dados, por exemplo). Claro que apenas com a duplicação de circuitos se pode aumentar a velocidade de acesso e possibilitar um interface mais inteligente, mas de esperar que os custos venham a baixar com a progressiva integração de funções.

O principal “segredo” da utilização do SCSI consiste no correcto endereçamento dos discos. Haveria uma certa tendência, caso apenas existisse um disco, a endereçá-lo como disco 0. O mais das vezes, esta seria uma opção errada, pois que o próprio controlador a nível do computador poderia ser o endereço 0! Quando se diz que um “bus” SCSI pode suportar até 8 periféricos, deve notar-se que isso implica que o próprio computador é um periférico. Outra dificuldade típica nos discos SCSI tem a ver com o tempo de inicialização dos mesmos. Como os discos têm microprocessadores com tarefas complexas como a identificação dos blocos, requerem por vezes um certo tempo de espera antes de poderem responder a pedidos, isto após a ligação do sistema. Existe, de facto, uma linha de *Reset* no SCSI, e a especificação indica que após essa linha ser activada todos os periféricos se devem reinicializar. Certos discos demoram mais tempo a fazer isso, e podem por isso ser diagnosticados como inexistentes pelo software. Em máquinas que no foram concebidas desde o início para operar com SCSI (como os IBM PC compatíveis), existem também frequentes dificuldades na configuração do software, pelo que é importante, ao adquirir um disco SCSI, assegurar o apoio do fornecedor. Outro problema frequente tem a ver com a terminação do *bus* (descrita adiante) e com deficiências de contacto em fichas de tantos condutores.

O INTERFACE IDE

O interface IDE (*Intelligent Drive Electronics*) é uma tentativa de obter as vantagens do SCSI sem implicar os

custos do mesmo. Foi desenvolvido com base na observação de que a maioria dos sistemas apenas é equipada com um disco e, portanto, um só controlador. Como é lógico, pensou-se em reunir os dois num só dispositivo, de forma a reduzir o custo do conjunto do sistema. O interface IDE possui então a semelhança com o SCSI de transferir dados em paralelo, mas não separa de forma tão inteligente as características físicas do disco das características lógicas. Fala-se aqui de sectores e de pistas, não de blocos, mas os dados são transferidos em paralelo, e as pistas, identificadas por números. Pelo lado do computador, a ligação ao disco é realizada através de uma simples porta paralela bidireccional, semelhante em circuito a uma porta de impressora. A ligação ao disco é realizada por um cabo com um número de condutores variável de fabricante para fabricante (o IDE não está normalizado oficialmente).

Os discos IDE raramente são designados por discos IDE! Na prática, distinguem-se dois tipos de IDE, o chamado XT e o AT. No primeiro caso, os discos são de performance mais modesta (68 ms de tempo de acesso, frequentemente), e a transferência realizada byte a byte. No segundo caso, os discos são muitas vezes de elevada performance (14 ms de tempo de acesso!), e a transferência realizada aos dois bytes de cada vez. É evidente que a designação XT se refere aos XT da IBM (que são sistemas com acesso memória em 8 bits), e a AT aos AT do mesmo construtor (que são sistemas com acesso memória em 16 bits).

O IDE está muito mais divulgado do que possa pensar-se. Apesar de se ouvir falar muito pouco desta designação, muitos dos "hard-cards" mais recentes são de tecnologia IDE. Por outro lado, o IDE consegue frequentemente velocidades de transferência capazes de fazer empalidecer os seus concorrentes mais famosos. Os referidos AT, por exemplo, são dos discos mais rápidos do mercado. Porém, o quadro de funcionamento dos IDE está essencialmente limitado aos sistemas IBM PC compatíveis, dado estes não requererem comportamentos mais inteligentes como o *overlapped request*, que só tem verdadeiramente importância no caso dos sistemas multitarefa que suportem este conceito. É de esperar que os IDE se divulguem ainda mais, e provavelmente ganharão terreno aos ST506, ao ponto de os eclipsar. As variantes AT começam a aparecer em sistemas de muito alta performance não baseados na tecnologia IBM, e ser interessante observar-se a evolução desta área teoricamente cativa dos ESDI e SCSI.

O INTERFACE ESDI

O ESDI (*Enhanced Small Drive Interface*) uma tecnologia essencialmente intermédia entre o ST506 e o IDE. Optimiza, porém, a velocidade de transferência de dados recorrendo a codificação local ao disco, embora a ligação com este continue a ser realizada em série. Porém, os discos ESDI são de muito elevada especificação e, por isso, suportam muito elevadas frequências de leitura e escrita. A selecção de posicionamento de cabeças é feita em série, mas são dados os "números" directos das pistas. A intenção, portanto, aproveitar ao máximo a velocidade possível, reduzindo a um mínimo os componentes que se encontram entre a cabeça do disco e a memória do computador. E os ESDI conseguem este comportamento *a la RISC* mas com um preço muito elevado. No nos adiantaremos mais sobre este interface, a não ser para referir que é provável que em breve seja suplantado pelo SCSI2 (ou pelas variantes AT dos IDE), até por ser muito mais dificilmente "clonável".

O CACHE DE DISCO

Referimos atrás o *cache* de disco, e vamos ver de



seguida que este conceito pode vir a determinar a maior parte dos avanços em termos de velocidade de acesso.

Um cache de disco é essencialmente uma área de memória RAM que reproduz o conteúdo de uma parte do disco. Vejamos, por exemplo, o que acontece quando num programa se abre um ficheiro, se escrevem alguns dados, se lêem outros e, finalmente, se fecha o ficheiro:

- 1) O programa identifica o ficheiro pelo nome. O sistema operativo vai procurar no directório do disco esse nome, obtendo uma "entrada de directório" que identifica a posição do disco em que o ficheiro começa.
- 2) O programa pede a escrita, por exemplo, de um bloco de dados com 5 bytes de comprimento. O sistema operativo vai calcular o sector do disco em que o bloco deve ser escrito, lê esse sector para a memória, altera-o com os dados a escrever, e envia de novo o sector para o disco.
- 3) O programa pede a leitura de um bloco situado imediatamente a seguir aos anteriormente escritos. O sistema operativo vai ler o sector, e devolver ao programa os dados pedidos.
- 4) O programa pede a leitura de um bloco situado no fim do ficheiro. Se o ficheiro for muito grande, o sistema operativo vai ter de ler o directório várias vezes até localizar a área do disco em que se encontra o dito fim do ficheiro, e em seguida ler o sector referido.
- 5) O programa pede a escrita de dados mesmo a seguir ao fim do ficheiro. O sistema operativo determina que precisa de um novo sector para esse fim, tem de o determinar provavelmente a partir de várias leituras do directório, e após escrever o novo sector ainda tem de ir ao directório actualizar a informação do mesmo.
- 6) Finalmente, o programa pede o fecho do ficheiro, pelo que o sistema operativo tem de actualizar no directório a data da última escrita no ficheiro.

Depois de estudarmos estes detalhes todos é fácil de ver que a parte do disco relativa ao directório, e algumas outras, foram repetidamente acedidas, envolvendo de cada vez um ou mais movimentos da cabeça que, como

“
Um cache de disco é essencialmente uma área de memória RAM que reproduz o conteúdo de uma parte do disco
”

“
**Uma forma
 ainda mais
 inteligente de
 acelerar o
 acesso
 consistiria em
 fazer uma
 análise
 estatística das
 zonas de disco
 mais acedidas,
 e retê-las
 preferencial-
 mente em
 memória**
 ”

sabemos, é a parte mais lenta do disco. Claramente, se retivéssemos em memória os diversos sectores lidos e escritos, seria desnecessário tornar a lê-los. Em suma, isso mesmo é um cache de disco: a retenção, em memória, dos últimos sectores transferidos, na expectativa de tornar a ocorrer um acesso em breve.

Os sistemas operativos mais modernos utilizam esta técnica numa forma limitada, na medida em que retêm os sectores mais recentemente movimentados (comando *Buffers* do *Config.sys* do MS-DOS). Porém, há outro aspecto em que o cache pode acelerar o acesso: frequente que os sectores sejam lidos sequencialmente (por exemplo, num disco bem arrumadinho um programa está gravado em sectores sucessivos). Se o disco pode ler de uma só vez a pista e todos os seus sectores, não há necessidade de os tornar a ler! Esta forma específica de cache, em que se antecipam os pedidos de leitura, o chamado *track buffer*, e não é tipicamente implementado a nível do sistema operativo. Uma forma ainda mais inteligente de acelerar o acesso consistiria em fazer uma análise estatística das zonas de disco mais acedidas, e retê-las preferencialmente em memória. O conjunto de todas estas optimizações é hoje em dia realizável no próprio controlador, e de facto existe, para os IBM AT compatíveis, um controlador baseado nestes princípios que permite uma velocidade de acesso que os programas de teste “apanhados desprevenidos” diagnosticam como um tempo médio de acesso não de 28 ms (como o disco por si só teria, neste exemplo) mas de 0.1 ms! É de esperar que estes comportamentos se vulgarizem à medida que desçam os preços da memória RAM. Para quem não disponha deste controlador, há alguns programas que podem acelerar o acesso ao disco custa de alguma memória. É o caso, por exemplo, do PC-Cache, que possibilita a utilização de memória normalmente fora do raio de acção do MS-DOS (memória acima dos 640 K) para cache.

A TERMINAÇÃO

Outro assunto que referimos atrás é a terminação, e que local mais apropriado para falar disto que o fim do artigo?

Quando se ligam entre si dispositivos através de fios, e através desses fios se fazem passar dados digitais, é frequente que os mesmos se vejam reflectidos, como ruídos, pelas extremidades dos fios. É um fenómeno inteiramente análogo ao do eco num poço: o som chega ao fim do poço, é lá ouvido, mas é reflectido para trás, embora com menos intensidade. Se cobrirmos o fundo do poço com algodão, ou qualquer outro material que ofereça dificuldade à passagem do som, o eco é consideravelmente reduzido. É nisso que consiste a terminação: num conjunto de resistências colocadas no extremo mais afastado do cabo.

No caso dos discos que partilhem cabos, como é frequente no caso do SCSI e do ST506, torna-se importante que apenas o último disco da cadeia tenha esses terminadores instalados. Esses assumem a forma de um circuito integrado ou de um conjunto de “jumpers”, e deve estar claramente identificado.

PARA TERMINAR

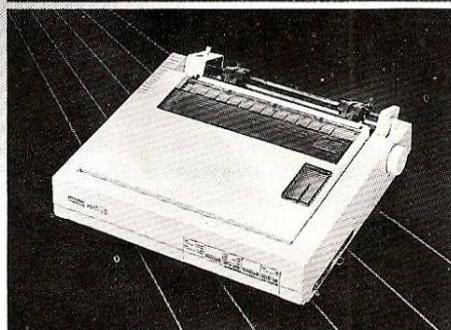
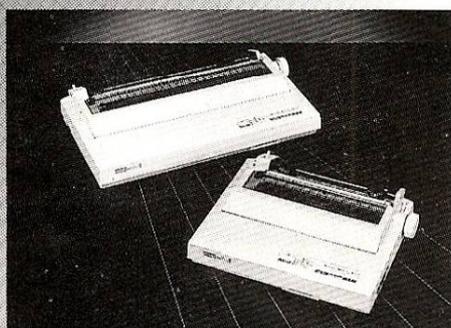
Ao longo deste artigo tentámos cobrir algumas das questões mais ignoradas que os discos levantam. Ser o leitor a decidir se o conseguimos, apenas poderemos sabê-lo se o leitor no-lo disser.

Muitas coisas ficaram por dizer, mas o essencial está aqui, quem sabe se numa forma excessivamente densa, mas, cremos, informativa. Quaisquer questões levantadas poderão ser endereçadas ao cuidado da redacção desta revista.

Armando Santos

CITIZEN

As impressoras profissionais!



PRODOT 9/9x

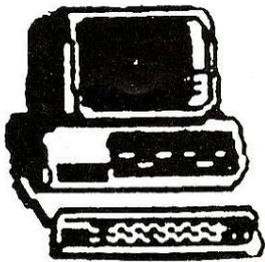
- 300 cps – 80 ou 136 colunas
- 4 fontes de caracteres
- parqueamento de papel
- kit de cores opcional
- caracteres Portugueses
- 2 anos de garantia (incluindo cabeça de impressão)

PRODOT 24

- 24 agulhas – 240 cps
- 7 fontes de caracteres
- parqueamento de papel
- kit de cores opcional
- caracteres Portugueses
- PRODISK 24: configuração remota
impressão de etiquetas
criação de caracteres
- 2 anos de garantia (incluindo cabeça de impressão)

A Informática tem a sua DÉCADA!

DÉCADA



abc - INFORMÁTICA, LDA.

Rua dos Sapateiros, 160 - 2.º - 1000 LISBOA

Telefs: 32 50 70-347 59 40 - Telex: ABCINF 13348 P - Fax: 346 22 40

CAMPANHA DE VERÃO

ABC 8088 - 10/12 MHz



ABC 8088

- Compatível PC XT
- CPU 8088-1 (CPU V20)
- 4,77/10 (12) MHz

➔ 10 Mhz - 1 Mb RAM, 12 Mhz - 640 Kb

- Drives de 5 1/4" ou 3 1/2"
- Portas série e paralelo
- Carta monocromática
- Teclado 102 teclas
- Monitor monocromático
- Mouse

PREÇO: ABC 8088 - 10 Mhz 120 000\$00*
 - 12 Mhz 125 000\$00*

ABC 80286

- Compatível PC AT
- CPU 80286 - 12/16 MHz

➔ 1 Mb RAM

- Drives de 5 1/4" ou 3 1/2" - Alta densidade
- Portas série e paralelo
- Relógio em tempo real / calendário / CMOS W / Bateria
- Carta monocromática (VGA opcional)
- Teclado 102 teclas
- Monitor monocromático
- Mouse

PREÇO: ABC 80286 - 12 Mhz 175 000\$00*
 - 16 Mhz 199 000\$00*

ABC 80286 - 12/16 MHz



**PARA REVENDA
CONSULTE-NOS**

ABC 80386

- Compatível PC AT
- CPU INTEL 80386 - 16/25/33 MHz

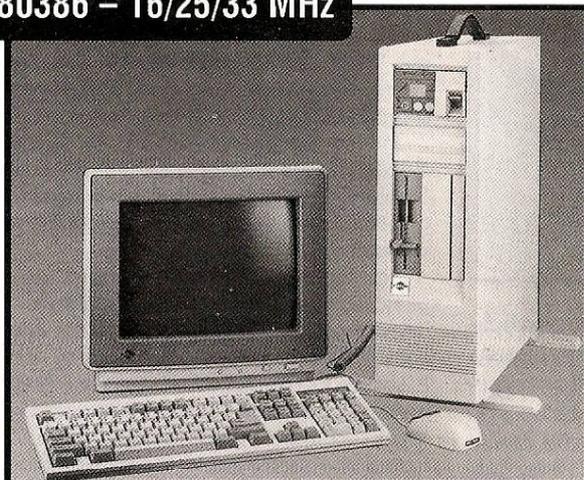
➔ 1 Mb RAM expansíveis até 8 Mb

- Drives de 5 1/4" ou 3 1/2" - Alta densidade
- Portas série e paralelo
- Relógio em tempo real / calendário / CMOS W / Bateria
- Carta monocromática (VGA opcional)
- Teclado 102 teclas
- Monitor monocromático
- Mouse

PREÇO: ABC 80386 - 16 Mhz 265 000\$00*
 - 25 Mhz 365 000\$00*
 - 33 Mhz 610 000\$00*

*Aos preços indicados serão acrescidos 17% IVA

ABC 80386 - 16/25/33 MHz



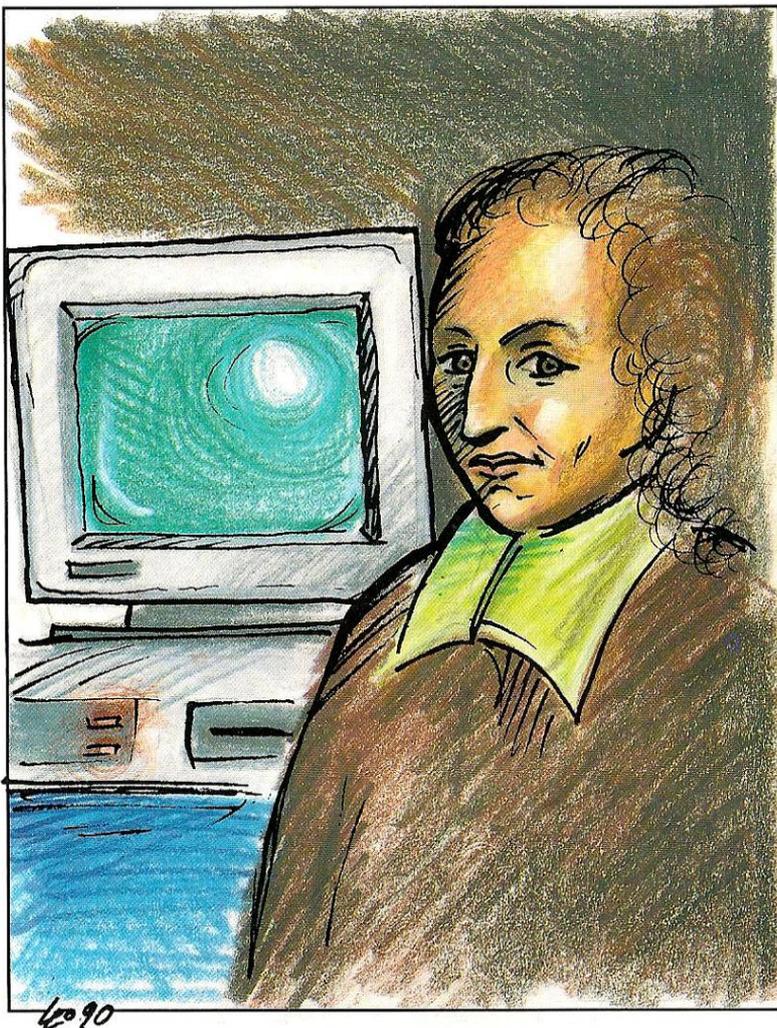
FACILIDADES PARA PAGAMENTO



abc AGENTES:

BRAGA: Avenida Central, 85 - 1.º - Telefs: 72 798-74 369
S. JOÃO ESTORIL: Av. Florinda Leal, Lote 1-A - Telef: 267 07 33
VISEU: Rua Direita, 79-1.º - Telef: 27 664
PORTIMÃO: Rua D. Carlos I - Telefs: 89 653-85670
SETÚBAL: Largo da Misericórdia, 28 - Telef: 31 432

PASCAL: PARA ENSINAR... E APRENDER



Este artigo pertence a uma série de textos dedicados à história e estudo, em geral, das linguagens de programação. Vamos, este mês, dedicar especial atenção, a uma linguagem com certeza já conhecida, se não mesmo completamente dominada pela maioria dos nossos leitores. Trata-se do Pascal.

Já aqui falámos do Fortran e do Algol 60. Vimos na altura que, enquanto que a primeira obteve um sucesso assinalável na época, já a segunda foi um fracasso comercial. E isto apesar da inovação que constituiu a nível de linguagens de programação.

O PL/1

Em 1964 surgiu, por parte da IBM, uma linguagem que se pretendia para todos os tipos de utilizadores, o PL/1. Incorporava conceitos e mecanismos de três linguagens simultaneamente: o Fortran, o Cobol, e o Algol 60. Era, por isso, enorme e extremamente difícil de dominar em toda a sua extensão. A isto contrapunha-se, na altura, que ninguém a precisava de conhecer totalmente, mas apenas a parte que realmente necessitasse. Infelizmente o argumento rapidamente se revelou falacioso: a interacção de conceitos era de tal forma intrincada que, para trabalhar em PL/1, se tornava realmente necessário conhecer a linguagem extensivamente.

Numerosas críticas foram, então, tecidas ao PL/1. Com elas, tornou-se clara a necessidade de criar linguagens mais simples de utilizar e aprender.

O NEWTON DA COMPUTAÇÃO

Paralelamente ao aparecimento do PL/1 desenrolava-se uma importante discussão que tinha por fim arranjar um sucessor para o Algol 60. É nesta altura, em 1965, que surge para a maioria das pessoas aquele que é talvez o Newton das linguagens de programação, se não mesmo da ciência da computação em geral: Niklaus Wirth. Apresentando um documento em parceria com um outro gigante, Hoare, Niklaus Wirth acabaria por ver as suas sugestões acerca do Algol 60 recusadas. Ao contrário do que Wirth pretendia, optou-se na altura por complicar e generalizar ainda mais o Algol 60, o que resultou no quase completamente ignorado Algol 68.

Mas o trabalho de Wirth acabaria por ser aproveitado. E é assim que aparece, pela mão da Universidade de Stanford, o Algol W. Embora sem nenhum impacto comercial, esta linguagem viria, contudo, a ser utilizada em numerosos círculos académicos.

Por seu lado, Wirth prosseguia a sua busca pela simplicidade. Produziu então as especificações de duas novas linguagens: o Euler e o PL 360. Seria apenas, contudo, com o Pascal que atingiria a notoriedade. Depois do Pascal, Wirth permaneceu imparável: Modula, Modula 2 e, mais recentemente, o Oberon. Nesta última linguagem, Wirth terá mesmo assumido o papel de censor, cortando tudo aquilo que lhe parecia menos necessário. O resultado é que nem o já velhinho ciclo for escapou ...

ENSINAR A PROGRAMAR

Mas regressemos então ao Pascal. Um objectivo máximo terá presidido à escrita da sua especificação: produzir uma ferramenta que ajudasse a ensinar as pessoas a programar. A programar bem. E, nesse aspecto, não há

dúvida que foi espectacularmente bem sucedido. Actualmente não haverá no mundo inteiro universidade que se preze, que não tenha os seus cursos de programação a funcionar em Pascal.

Para além deste nobre princípio, pretendia-se ainda do Pascal que fosse fiável e eficiente. Wirth havia já entendido que para conseguir vencer o Fortran, teria no mínimo que se manter a sua eficiência. Tanto a de execução, como a de compilação.

Tendo conseguido atingir estes objectivos, não é de admirar a actual popularidade do Pascal. Popularidade essa que, ao que parece, só tem tendência a aumentar. Tanto mais que o número de jovens programadores utilizando o Pascal não tem parado de crescer.

A EXTENSIBILIDADE

O Pascal aparece numa altura em que se discutia muito em torno da extensibilidade de uma linguagem. Esta extensibilidade tinha por objectivo reduzir a complexidade, tornando a linguagem tão pequena quanto possível. A linguagem acaba, afinal, por se tornar num pequeno núcleo, extremamente simples, que depois é ampliado de acordo com as necessidades de cada aplicação. Essencialmente, três mecanismos facultavam essa ampliação: definição de novos tipos, macros e definição de novos operadores.

Principalmente as macros acabariam por frustrar aqueles que haviam depositado na extensibilidade a sua aposta no futuro. Com efeito, tornava-se extremamente difícil ao programador detectar erros nos seus programas, já que estes vinham em função do pequeno núcleo, e não das extensões que utilizava. Por isso a extensibilidade, que tanta tinta fizera correr no fim da década de sessenta, acabou por cair no esquecimento. Pelo menos, nos moldes de então.

Mas algumas ideias ficaram. E esse é o caso das definições de tipos, que Wirth viria a incluir no Pascal. Uma outra ideia que sobreviveu foi a de definir novos operadores, surgindo hoje em linguagens tão distintas como o Ada e o Prolog. As próprias macros sobreviveram e são, hoje, intensamente utilizadas por qualquer programador de C.

DEFINIÇÕES DE NOVOS TIPOS

Wirth decidiu, pois, que no Pascal seria permitido definir novos tipos à custa de outros. Além disso acrescentou um novo tipo básico àqueles que eram já tradicionais: o tipo *char*. Assim, adicionou o tipo *char*, aos tipos *integer*, *real* e *boolean*, herdados do Algol 60 e do Fortran. Pretendia com isso, alargar o espectro de utilizadores destas duas linguagens, especialmente vocacionadas para o cálculo científico.

Como forma de cativar utilizadores mais interessados em processamento simbólico, foram criados os tipos enumerados. O programador via agora a sua expressividade substancialmente aumentada, em grande número de situações. Isso acontecia, afinal, sempre que necessitava referir-se a algo que, embora não fosse um número inteiro, arrastava consigo e de forma intrínseca, uma relação de ordem. É o que acontece quando nos referimos aos dias da semana, ou aos meses do ano. Em vez de utilizar inteiros de um a sete, o programador pode agora optar por definir um tipo enumerado:

```
type diaSemana = (Domingo, Segunda,
Terca, Quarta, Quinta, Sexta, Sabado);
```

Uma vez definido o tipo, pode evidentemente declarar variáveis desse mesmo tipo:

```
var dia : diaSemana;
```

Variável essa que pode, por exemplo, ser usada num ciclo *for*:

```
for dia := Segunda to Sexta do
... .. .
```

Com os tipos enumerados vem um conjunto de operadores que nos permitem compará-los. São o *<*, *<=*, *>*, *>=*, *=* e o *<>*. Estes operadores têm o mesmo aspecto gráfico, que os que nos permitem comparar reais, por exemplo. Não se trata, contudo, do mesmo operador, já que é diferente comparar dois enumerados de comparar dois reais. Diz-se, pois, que existe *overloading* de operadores.

Mas as virtudes dos tipos enumerados, não se esgotam no seu poder de expressividade. Permitem ainda que o armazenamento das variáveis seja feito de uma forma mais económica. A verdade é que o compilador vai proceder com os tipos enumerados como se de inteiros se tratasse. Ou seja, vai fazer aquilo que o programador faria, no caso de não possuir este mecanismo. Simplesmente, vai poder economizar em espaço. No caso dos dias da semana, por exemplo, precisará apenas de representar inteiros entre 1 e 7. Poderá, pois, fazê-lo, recorrendo a apenas 3 bits e não ao espaço normalmente ocupado por um inteiro.

INTERVALOS FINITOS

Um outro aspecto do Pascal é o de permitir definir os chamados *subranges*.

Imaginemos que queremos ter, num programa, uma variável que nos indique uma determinada hora. Não nos interessa aqui com que fim será afinal utilizada essa variável, mas seja lá ele qual for, uma coisa devemos sempre poder dizer acerca dela: o seu valor nunca sairá do intervalo [0,24]. Ora seria conveniente que o programador pudesse deixar isso bem claro no texto do programa. E melhor seria, ainda, que o compilador pudesse gerar um código por forma a que fosse possível validar o valor da variável, em tempo de execução.

Para fazê-lo, o programador não tem mais que declarar a variável, como pertencendo a um *subrange*:

```
var hora : 0 .. 24;
```

Mais uma vez, o compilador poderá ainda aproveitar o facto de a variável só poder assumir valores de 0 a 24.

Note-se, contudo, que nem todos os compiladores se preocuparão com este tipo de coisas. Nem com as validações, nem com este tipo de optimizações.

CONSTRUTORES DE TIPOS

Como já se disse, o Pascal permite definir tipos à custa de outros. Para o conseguirmos fazer, o Pascal põe à nossa disposição construtores de tipos. Nestas condições, temos assim o *array*, o *record*, o *set* e o *^*.

O *array* é já sobejamente conhecido. Trata-se, afinal, de uma variável indexada, em quase tudo idêntica à do Fortran ou do Algol 60. No entanto, assume no Pascal um papel ligeiramente diferente, já que o tipo base do *array* pode ser qualquer outro tipo anteriormente definido. Por outro lado, o tipo dos índices do *array* poderá ser qualquer tipo enumerado, e não só inteiro. Se nos quiséssemos referir a cada uma das horas, para todos os dias da semana, faríamos, por exemplo:

```
type horario = array [ diaSemana ] of
array [ 0..24 ] of
boolean; { ou qualquer outro tipo }
```

“

Um objectivo máximo terá presidido à escrita da sua especificação: produzir uma ferramenta que ajudasse a ensinar as pessoas a programar. A programar bem

”

“
A necessidade de utilizarmos registos surge sempre que queremos associar um determinado número de informações diversas sobre um determinado objecto
 ”

Criámos assim um novo tipo, horário. Trata-se, afinal, de um vector indexado pelos dias da semana, em que cada posição é novamente um vector, agora de valores booleanos.

PROBLEMAS...

Os arrays de dimensão variável, que apareciam no Algol 60, foram banidos do Pascal. Isto resultou de duas decisões que Wirth tomou em relação ao Pascal. A primeira foi de que todos os tipos teriam que ser conhecidos em tempo de compilação. A segunda diz que as dimensões do array fazem parte do seu tipo. Da interacção destas duas características resulta imediatamente que as dimensões dos arrays têm de ser conhecidas em tempo de compilação.

Note-se que os arrays de dimensão variável, se bem que interessantes, não se revestem, contudo, de grande importância. Daí que não tenha sido grave o facto de terem sido excluídos. Infelizmente, o mesmo não se passa com uma outra característica do Pascal, que resulta do facto de esta linguagem ser fortemente tipada. Nestas linguagens, exige-se naturalmente que os tipos dos argumentos que passamos a funções e procedimentos coincidam com os dos seus parâmetros formais. Ora, uma vez que do tipo dos arrays faz parte a sua dimensão, torna-se imediatamente impossível escrever procedimentos genéricos para processar arrays.

Foi para resolver este problema que se viria mais tarde a estender a especificação inicial do Pascal com um novo mecanismo: os quadros conformes.

REGISTOS

Um outro construtor de tipos é, como já referimos, o record. Um record, ou registo em português, é um aglomerado heterogéneo de dados. Surge, pois, em oposição ao

array, já que para este existe um tipo base, e todas as posições têm esse mesmo tipo. Note-se que outras linguagens, como o Cobol, dispunham já de aglomerados deste tipo.

A necessidade de utilizarmos registos surge sempre que queremos associar um determinado número de informações diversas sobre um determinado objecto. Imaginemos que estamos, por exemplo, a falar de livros. Cada livro terá um título, um autor, um determinado número de exemplares, um código e uma data de aquisição. Desta forma, declararíamos os seguintes tipos:

```
type data = record dia, mes, ano : integer
end;
livro = record
  titulo : packed array [ 1..50 ] of char;
  autor : packed array [ 1 .. 30 ] of char;
  nExempls : integer;
  aquisicao : data
end;
```

Mais uma vez, poderíamos declarar variáveis deste tipo:

```
var l : livro;
```

Poderíamos, agora, aceder a cada um dos campos desta variável, através do selector '.'. Para sabermos o título do livro l, bastar-nos-ia pois fazer l.titulo.

O Pascal dispõe ainda de uma forma ligeiramente diferente de registo. Trata-se do registo com variante. Utilizando este mecanismo, todos os campos de um determinado registo passam a ocupar o mesmo espaço físico de memória. É utilizado em situações em que queremos guardar dados diversos, mas em que não faz sentido os campos coexistirem.

Fazendo mais uma vez uma comparação entre arrays e



AUTOEDIÇÃO

- ★ Grafismo superpotente
- ★ 4 Megas Ram
- ★ 20 Mb Disco
- ★ Impressora Laser
- ★ Monitor alta resolução

O mais alto nível de qualidade
 a um preço que nenhuma outra
 marca poderá oferecer.



a b c **INFORMÁTICA, Lda.**

R. Sapateiros, 160 - 2ª 1100 LISBOA. Tel. 32 50 70

Agentes:

LISBOA: Rua da Assunção, 67 Telf: 32 72 96

BRAGA: Av. Central, 85-1º Telf: 74 369

S. JOÃO DO ESTORIL: Telf: 267 07 33

VISEU: Rua Direita, 77 Telf: 27 664

PORTIMÃO: Rua D. Carlos I Telf: 83 653

SETÚBAL: Largo da Misericórdia, 28 Telf: 31 432

records, vemos que enquanto os primeiros são aglomerados homogêneos de dados, os segundos são aglomerados heterogêneos. Por outro lado, enquanto os primeiros permitem um acesso dinâmico, já que o índice pode ser uma variável, nos segundos o acesso é sempre estático.

CONJUNTOS

Vimos já os construtores array e record. Ora o Pascal dispõe de um outro construtor, o set, extremamente inovador. Permite-nos trabalhar com conjuntos, facultando-nos para isso as operações mais frequentes entre estes, como sejam a reunião, a intersecção e a diferença. Está ainda definido o operador *in*, que inspeciona a existência, ou não, de um elemento dentro de um conjunto. Em Pascal podemos ter conjuntos de qualquer tipo enumerado ou, ainda, de subranges.

Imaginemos, por exemplo, que estávamos a escrever um formatador de texto. Necessitávamos classificar os caracteres, como sendo separadores, dígitos e letras. Para tal, e tirando partido de o tipo char ser um tipo enumerado, definíamos e inicializávamos as seguintes variáveis:

```
var digitos, letras, separadores : set of char;
c : char;
```

```
begin
{ inicializacao }
digitos := [ '0'..'9' ];
letras := [ 'á'..'z', 'Á'..'Z' ];
separadores := [ ' ' ];
... ..
end.
```

Sempre que quiséssemos verificar se a variável *c* contém ou não uma letra, bastar-nos-ia então fazer:

```
if c in letras then
... ..
```

Se, por outro lado, precisássemos testar se *c* é alfanumérico, não teríamos mais que fazer:

```
if c in letras + digitos then
... ..
```

A pergunta, surge, agora naturalmente: como implementar conjuntos de uma forma eficiente?

A ideia é reservar um bit por cada elemento que possa, eventualmente, fazer parte do conjunto. Desta forma, as variáveis digitos, letras e separadores ocupariam 256 bits, no caso de haver 256 caracteres. O facto de o bit 102 da variável letras estar posicionado a 1 significa que o carácter com o código ASCII 102, pertence ao conjunto das letras. Isto admitindo que a convenção preferida é de que o bit a 1 significa presença e a 0, ausência.

A intersecção entre dois conjuntos é agora trivial: basta fazer o 'and' bit a bit, entre as duas variáveis. Para a reunião, basta-nos fazer o 'or' bit a bit. Como facilmente se compreende, todas as operações entre conjuntos são, pois, extremamente eficientes.

APONTADORES

Vamos agora, finalmente, debruçar-nos sobre o construtor de aspecto mais sinistro. Trata-se do ^, que nos permite definir um novo tipo, como sendo um apontador para um tipo previamente definido.

De facto, o Pascal permite-nos manipular variáveis dinâmicas. Ou seja, variáveis que só são alocadas em tempo de execução. Estas variáveis não ficam, de resto,

situadas no mesmo local das outras. Os leitores mais atentos lembrar-se-ão que, no artigo sobre o Algol 60, se referiu que as variáveis, nas linguagens organizadas em bloco como o Pascal, ficavam localizadas num stack. Pois as variáveis dinâmicas ficam, não no stack, mas numa zona normalmente designada por heap.

Para declararmos um apontador para o tipo livro, por exemplo, faríamos:

```
var apLivro : ^livro;
```

O que está declarado neste momento é uma variável do tipo apontador para livro. Essa variável em nada difere daquelas que já vimos e, como tal, fica no stack de execução. No entanto, tal como está de pouco nos serve, uma vez que não aponta para nada de interesse.

Para que passe a apontar para uma variável do tipo livro, teremos que inicializá-la. Para tal, recorre-se a um procedimento predefinido do Pascal. Trata-se do new:

```
begin
new( apLivro );
apLivro^.nExempls := 5;
... ..
end.
```

Sempre que se faz new, o programa encarrega-se de arranjar espaço na heap e cria uma nova variável. Para se aceder a essa variável, utiliza-se também o símbolo ^.

A utilização de variáveis dinâmicas, permite-nos definir estruturas de dados recursivas, como listas e árvores.

ESTRUTURAS DE CONTROLO

O Pascal apresenta três ciclos distintos. O primeiro, o ciclo for, representa uma versão ultra-simplificada do for do Algol 60. Quanto aos outros dois, o while e o repeat until, implementam ciclos com teste, respectivamente, à entrada e à saída.

Além disso, o Pascal apresenta ainda um goto, também ele simplificado.

Como estruturas de controlo para selecção, o Pascal apresenta-nos o if e o case. Este último, utiliza-se como um if de múltiplas entradas.

CONCLUSÃO

O Pascal foi especialmente concebido para ensinar a programar. Neste campo cumpriu excepcionalmente. Basta, para tal, atendermos ao ainda crescente número de estabelecimentos de ensino que o adoptam como forma de ensinar a programar. Poucas pessoas discordam já, que esta seja de facto a ligação pedagógica ideal. Além disso, a esmagadora maioria dos livros de algoritmia e estruturas de dados no mercado, usam o Pascal.

O Pascal é, porém, muitas vezes criticado, por ser usado em outros domínios. Isso resulta do facto de o Pascal se ter tornado a primeira linguagem para muita gente, que assim se recusa a abandonar a sua «língua mãe». Refira-se, por exemplo, o facto de as chamadas à toolbox do Macintosh serem feitas em Pascal.

Por forma a poder ser usado em situações, para o qual não foi desenhado, o Pascal tem sido estendido e expandido em todos os sentidos e direcções. Desde a programação concorrente, até à programação por objectos.

É preciso ainda não esquecer, a extraordinária influência que o Pascal teve nas linguagens que se lhe sucederam. Com efeito, praticamente todas as linguagens de terceira geração, desde então, são *Pascal like*.

“
É preciso
ainda não
esquecer, a
extraordinária
influência que
o Pascal teve
nas
linguagens
que se lhe
sucederam.
Com efeito,
praticamente
todas as
linguagens de
terceira
geração, desde
então, são
Pascal like
”

Ganhe Tempo e Organização

TRIUDUS

a sua melhor opção informática

Somos a empresa que mais computadores profissionais vendeu em Portugal, directamente ao público.

Seja mais um cliente a comprovar, e a descobrir, as razões da preferência TRIUDUS.



“ADD-INS”

VITAMINAS PARA O LOTUS 1.2.3!

Os “add-ins” para o Lotus 1.2.3 são programas desenvolvidos com o auxílio de uma aplicação denominada “Add-in Tool Kit”, fornecida pela própria Lotus Development Corporation às diversas casas de software que pretendem escrever extensões para o Lotus 1.2.3. Geralmente, estas extensões implementam funções não suportadas na versão original da folha de cálculo, ou seja, são “acrescentos” a todas as capacidades já possuídas pelo Lotus. No entanto, graças à forma como são integradas, estas novas funções comportam-se como se sempre tivessem estado lá – o utilizador não nota qualquer diferença no desempenho do Lotus quando usa uma função standard ou outra proveniente de um add-in.

Ao permitir que esta sua folha de cálculo pudesse sofrer expansões com facilidade, os programadores da Lotus Development Corporation prepararam o terreno para o desenvolvimento de novas versões com um mínimo de esforço; ao ceder o “Add-in Tool Kit” às software houses que manifestaram interesse em trabalhar com ele, o departamento comercial da LDC revelou uma inteligência acima do comum, contribuindo sobremaneira para o sucesso do seu produto.

Tal como o grande sucesso dos computadores IBM PC se deve, em larga escala, à existência de compatíveis, também em grande parte o Lotus 1.2.3 se tornou um *best-seller* devido às potencialidades que lhe foram acrescentadas pelos add-ins desenvolvidos por empresas independentes do fabricante original.

A maioria dos add-ins é distribuída com um programa intitulado Add-in Manager, igualmente cedido pela Lotus Development Corporation. A função deste programa é instalar (ou desinstalar) os add-ins na folha de cálculo do utilizador. Durante o processo de instalação, define-se o modo de chamada (por hot-key ou por menu) e ainda se os add-ins são automaticamente colocados em memória no início da sessão de trabalho. O Add-in Manager pode ser posteriormente chamado em qualquer altura através da combinação [Alt]+[F10]; os add-ins usam as combinações [Alt]+[F7], [Alt]+[F8] e [Alt]+[F9] para hot key, podendo ser instalados até ao máximo de oito simultaneamente.

ALLWAYS – THE SPREADSHEET PUBLISHER

O objectivo do Allways é levar até aos utilizadores da folha de cálculo mais famosa do mundo (Lotus 1.2.3 – era preciso dizer?) as capacidades da edição assistida por computador, vulgo Desktop Publishing.

O Allways instala-se como um add-in, funcionando dentro da própria folha de cálculo: durante uma sessão de trabalho com o Lotus, invoca-se premindo uma determi-

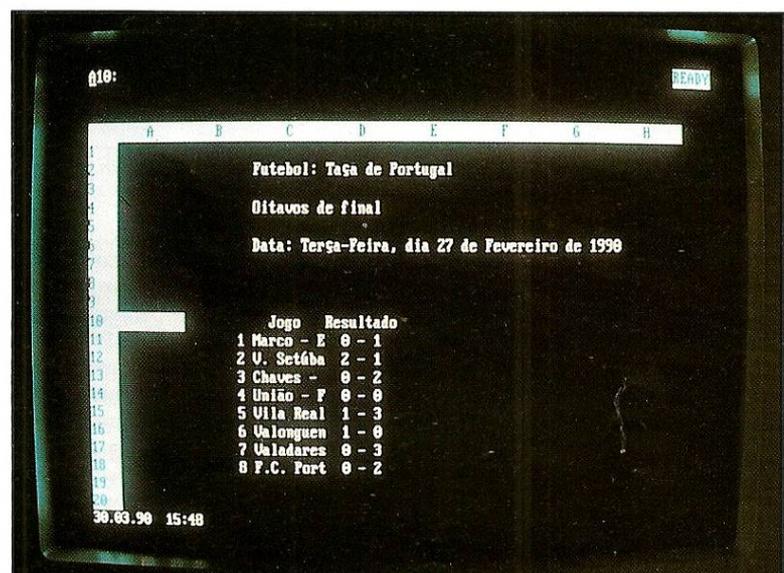


Figura 1 – aspecto do ecrã com o Lotus



Figura 2 – aspecto do ecrã com o Allways

nada combinação de teclas, usa-se e regressa-se de seguida ao Lotus. Os comandos são escolhidos por menu (tecla “/”) estando assim dispostos segundo o formato familiar aos utilizadores do Lotus.

Verdadeiramente curioso é o modo de funcionamento deste add-in, porque todo o conteúdo da folha tem de ser

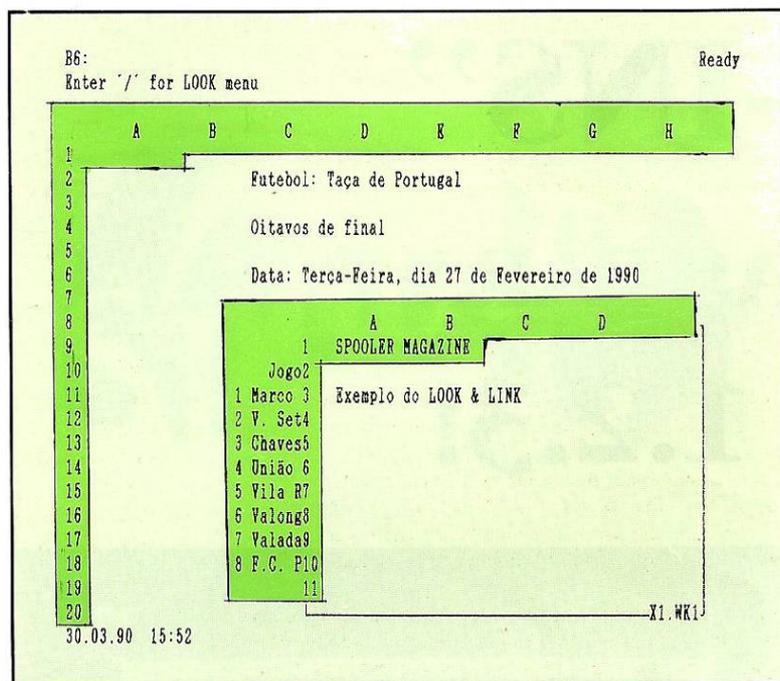


Figura 3 – aspecto do ecrã quando são visualizadas duas folhas em simultâneo

“
**Grças ao
 Look, é-nos
 possível
 visualizar uma
 segunda folha
 a um canto do
 ecrã enquanto
 se trabalha**
 ”

inserido dentro do Lotus, exactamente como se não existisse o Allways. Como é do conhecimento geral, o conteúdo de uma folha de cálculo tem geralmente um aspecto bastante grosseiro, tanto no ecrã do PC como, depois, na listagem final; este facto leva a que muitas vezes os utilizadores do Lotus percarn bastante tempo, depois de terminadas as tarefas úteis, a melhorar a apresentação do seu trabalho.

Dispondo do Allways, os utilizadores devem apenas preocupar-se com os dados e não com a sua apresentação! O Allways vai permitir controlar, *a posteriori*, todo o aspecto visual da folha; se o utilizador possuir um monitor gráfico (Hercules, CGA ou EGA) vai ter inclusivamente disponível a previsão do aspecto da folha impressa (WYSIWYG).

O utilizador dispõe de um extenso lote de potencialidades: possibilidade de combinar até 8 fontes diferentes na mesma folha, sublinhar ou destacar texto (“underline” e “boldface”), usar sublinhado duplo, ajustar individualmente a altura das linhas ou a largura das colunas com precisão ao nível do pixel, sombrear (com 3 tonalidades diferentes) quaisquer conjuntos de células da folha, desenhar linhas horizontais e verticais, envolver as células em caixas, incluir **simultaneamente** gráficos na folha e imprimir a cores (caso a impressora o permita).

Melhor que as palavras, as imagens falam por si: as figuras 1 e 2 são uma pequena, mas elucidativa, demonstração daquilo de que o Allways é capaz.

O que aqui é mostrado não demora mais de 10 minutos a ser feito por um utilizador com alguma experiência; para um utilizador médio de Lotus 1.2.3, a aprendizagem do Allways não apresenta problemas de maior, podendo considerar-se uma semana como período mais do que suficiente para uma “licenciatura” em Allways.

O programa chegou-nos em 5 disquetes de 51/4” acompanhado por um óptimo manual que, sem ser demasiado extenso – 96 páginas não assustam ninguém! – aborda com clareza todos os aspectos relacionados com a utilização do Allways. Uma grande diversidade de impressoras laser ou de agulhas (9 ou 24) são suportadas por este add-in, em cujas disquetes são também enviadas 3 fontes de caracteres (Courier, Times e Triumvirate) em tamanhos de 5 a 24 pontos.

LOOK & LINK

As metas deste add-in, produzido pela Personics Corporation, são essencialmente duas, já que na realidade se trata de dois programas independentes: a possibilidade de trabalhar com duas folhas visíveis simultaneamente no ecrã (aplicação Look) e a utilização de múltiplas folhas ligadas entre si, podendo ser encaradas na sua globalidade como uma só (aplicação Link).

Gracas ao Look, é-nos possível visualizar uma segunda folha a um canto do ecrã enquanto se trabalha; a dimensão e a localização da janela desta nova folha podem ser modificadas de acordo com as necessidades; é ainda possível efectuar “scroll” da área visível, trocar as folhas entre si e copiar gamas de células de uma para outra. Tal como no caso do Allways, só se podem efectuar alterações na folha principal – é por esta razão que existe o comando de troca (swap) das folhas (ver figura 3).

Com o Link pode ligar-se várias folhas entre si, graças a um novo conjunto de funções @ que passam a estar disponíveis ao utilizador. Desta forma, uma folha pode usar valores de células ou gamas de células de outros ficheiros, localizados no disco ou em memória – são permitidas, inclusivamente, acções sobre tabelas remotas!

A lista das novas funções é a seguinte:

@setgroup	@wksblank
@addgroup	@wksvlookup
@cleargroup	@wkslookup
@wksload	@wkssum
@wksunload	@wksavg
@wkscell	@wksmin
@wksmax	

A forma de utilizar estas novas funções não tem qualquer diferença em relação às funções standard equivalentes. Por exemplo:

@sum(a1..a10) ... Com esta função (standard) calcula-se a soma dos números existentes na gama A1..A10

@wkssum(“C:\LOTUS\TESTE”, “a1..a10”)... Com esta função é calculada a soma da gama de células A1..A10 da folha TESTE.WK1 situada na directoria C:\LOTUS, sendo o resultado colocado na folha corrente. Como se pode verificar, a alteração na sintaxe da função tem como objectivo identificar o ficheiro sobre o qual a função irá ser efectuada.

As funções rigorosamente novas são as que permitem juntar uma série de ficheiros num grupo (@setgroup, @addgroup, @cleargroup) atribuindo um nome ao grupo no sentido de referenciar todo o conjunto apenas por um identificador – desta forma, todas as funções que referirem o nome do grupo serão na realidade aplicadas ao conjunto de ficheiros que formam o grupo. As funções @wksload e @wksunload são também novidade – o seu objectivo é permitir carregar em memória principal o conjunto de ficheiros a ser usado pelo Link, melhorando o desempenho do programa no acesso a ficheiros remotos. Todas as outras funções têm o seu equivalente no Lotus standard.

E OUTROS ...

É vastíssima a quantidade de add-ins disponíveis para o Lotus 1.2.3. Porém, não nos é possível abordar todos os restantes de forma tão detalhada como fizemos em relação ao Allways e ao Look & Link, especialmente porque

o director desta honesta publicação decidiu que este não era um número especial da **Spooler** com as 48 páginas dedicadas ao fascinante tema dos add-ins – é estranho – no entanto, o boss (tal como o cliente) tem sempre razão...

A Personics Corporation possui também no mercado um outro produto, a que deu o nome de SeeMORE – com o SeeMORE, é possível efectuar um zoom out de determinada zona da folha, comprimindo os caracteres e permitindo assim visualizar quase o dobro da área de trabalho.

Por outro lado, a Funk Software (que desconfiamos, vive apenas disto!) apresenta mais quatro outros add-ins para o Lotus 1.2.3: Sideways, InWord, NoteWorty e The WorkSheet Utilities.

O SideWays trabalha apenas com a impressora; o que faz é uma rotação de 90 graus a todos os caracteres da folha, permitindo assim imprimir o seu conteúdo horizontalmente – esta forma de listagem é designada correntemente por landscape. A utilidade deste “add-in” é evidente, pois um qualquer trabalho efectuado com o Lotus 1.2.3 ocupa quase sempre a largura de várias folhas de papel. Sem recorrer ao SideWays, a solução passa pela tesoura e cola para juntar as folhas, como se fosse um *puzzle*; com o SideWays, para além de se obter uma listagem sem interrupções no papel, é permitido escolher a fonte entre 9 (!) disponíveis, alterar o tamanho das letras e usar “boldface” ou “underline”, entre outras facilidades.

O InWord é um editor de texto, implementando todas as funções que se esperam de um editor minimamente simpático: word wrap, formatação de parágrafos, paginação, pesquisa e alteração de texto, cabeçalhos, rodapés, numeração das páginas, ...

Devido ao facto de estar embebido no próprio Lotus 1.2.3 (apesar de o texto ser guardado num ficheiro separado da folha de cálculo propriamente dita) este editor permite importar para o texto qualquer conjunto de células. Uma outra habilidade é aquilo que os autores designam por hot links – se as células da folha de cálculo a que se referem estes “links” forem alteradas, as modificações serão automaticamente sentidas no texto!

NoteWorty é o nome de um “add-in” que permite associar comentários às células, até ao limite de 8kBytes por célula. É útil essencialmente para os programadores que desenvolvem aplicações em Lotus 1.2.3, que desta

forma passam a dispor da possibilidade de comentar esta ou aquela fórmula mais complexa. O facto de uma célula possuir um comentário não altera o seu aspecto porque o comentário não fica visível – cada vez que o cursor está numa célula comentada, aparece no fundo do ecrã um indicador a assinalar o facto. Os comentários podem ser consultados e alterados em qualquer altura, bastando para tal premir a combinação de teclas da hot key.

As WorkSheet Utilities são um conjunto de utilitários dos quais vamos referir apenas os três mais importantes.

O utilitário Search and Replace permite pesquisar todas as ocorrências de um valor, fórmula, label ou referência de célula na folha de cálculo e alterá-las (em toda a área da folha) por um qualquer conteúdo à nossa escolha; com o Formula Editor pode, entre outras coisas, executar-se uma fórmula passo a passo para detectar eventuais erros; o Autosave serve para temporizar a acção de salvar em disco o trabalho que se está a desenvolver na folha de cálculo – uma vez determinado o intervalo de tempo, o save será feito periodicamente (e automaticamente) diminuindo o risco de perda accidental do trabalho.

Não temos conhecimento de que qualquer dos programas referidos neste artigo esteja comercializado em Portugal – para satisfazer os potenciais interessados vamos indicar as moradas a contactar:

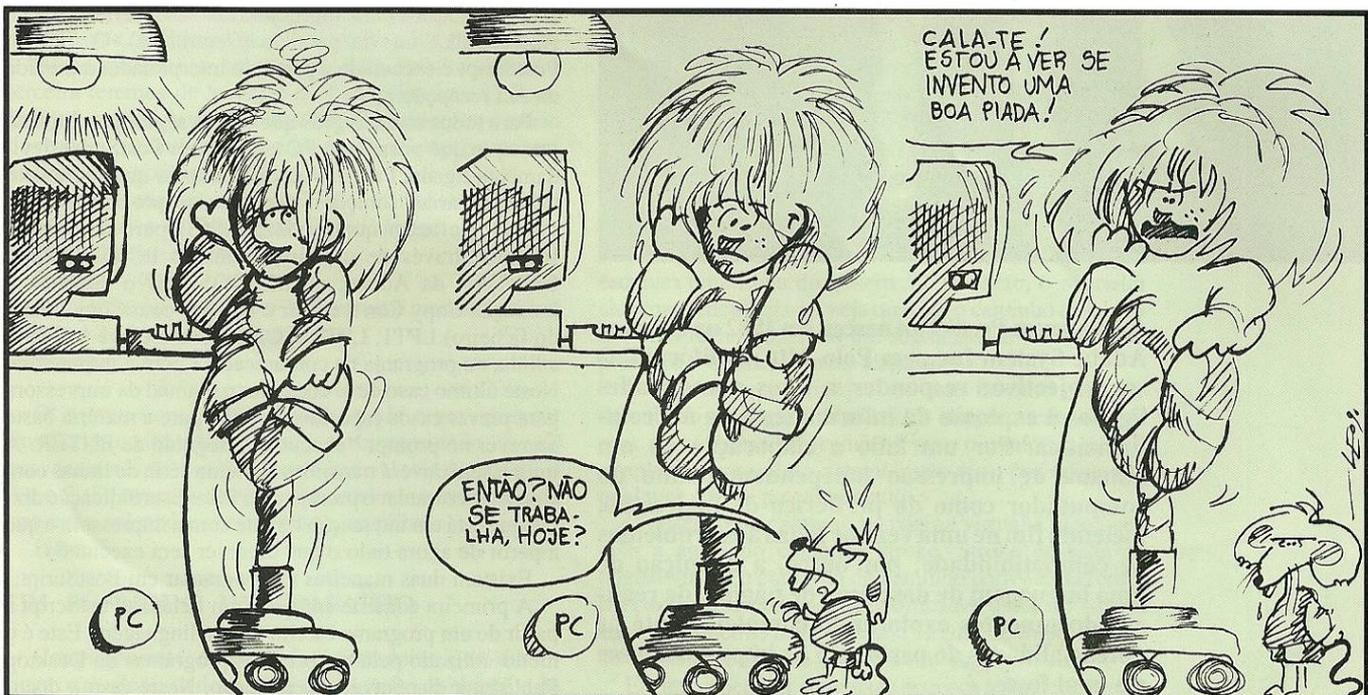
Funk Software, Inc
222 Third Street
Cambridge, MA 02142

Personics Corporation
63 Great Road,
Maynard MA 01754
USA

Supomos que o que foi dito sobre os add-ins para o Lotus 1.2.3 terá sido o bastante para aliciar alguns utilizadores desta folha de cálculo a adquirir um ou outro utilitário dos referidos. Aliás, os preços não são exorbitantes, rondando cerca de 100 dólares. Se tiverem de optar por um, a opção deverá recair sobre o Allways.

 Paulo Bernardo

“
Uma outra
habilidade é
aquilo que os
autores desi-
gnam por hot
links – se as
células da
folha de
cálculo a que
se referem
estes “links”
forem
alteradas, as
modificações
serão
automaticamente
sentidas no
texto
”



PostScript

A LINGUAGEM QUE PRODUZ A SPOOLER



A linguagem PostScript nasceu em 1982 na empresa Adobe System Inc., em Palo Alto, Califórnia. O seu objectivo: responder a duas necessidades ligadas à explosão da informática e da microinformática. Por um lado a elaboração de um sistema de impressão independente tanto do computador como do periférico de impressão, metendo fim de uma vez por todas aos problemas de compatibilidade; por outro, a definição de uma linguagem de descrição de página, de realizar documentos explorando completamente as potencialidades do periférico de impressão fosse ele qual fosse.

O PostScript é uma linguagem de programação de corpo inteiro. Exclusivamente destinada à composição e impressão de documentos, o interpretador do PostScript está implantado directamente no periférico de impressão, transformando-o num microcomputador. Assim, a impressora LaserWriter da Apple vem equipada com um microprocessador Motorola 68000 e 1,5Mega de RAM e 1,0 de ROM. No entanto, é corrente hoje em dia ver impressoras PostScript com dois e três Mega de RAM. Poucos computadores podem gabar-se de ter tal capacidade de memória.

Em teoria basta possuir um terminal (teclado e monitor) dotado de um interface compatível com a impressora para imprimir tudo o que se quiser no PostScript. Normalmente, esta tarefa está confiada a potentes PCs com elevadas capacidades gráficas e associados a programas de paginação compatíveis PostScript capazes de representar no ecrã a imagem do documento antes de a enviar para a impressora.

Tal como o Basic (com o qual esta é a única semelhança), o PostScript é uma linguagem interpretada. Um programa PostScript é um ficheiro ASCII que pode ser lido e modificado com a ajuda de qualquer editor de texto. A comunicação com a impressora faz-se via série (RS 232, Apple Talk), ou via paralela (Centronics). O ficheiro PostScript é executado a partir do interpretador à medida da sua recepção.

Para todos os exemplos que se vão seguir partimos do princípio que possui um PC e uma laser PostScript devidamente ligada. Para enviar os exemplos que se seguem existem várias soluções: escrever os programas num tratador de texto que produza ASCII puro e enviar o ficheiro através de utilitários como o PCSEND ou o PSDOWN da Adobe; no MS-DOS usar o edlin ou a instrução Copy Con e enviar o ficheiro com Copy (nome do ficheiro) LPT1, LPT2, COM1, etc.; escrevê-lo linha a linha no programa de comunicações com a impressora. Neste último caso deve consultar o manual da impressora para prever casos especiais, embora para a maioria baste escrever no prompt "executive" seguido de ENTER. A impressora deverá transmitir-lhe uma série de linhas com Copyright e mudar o prompt para PS>. Esta indicação diz-lhe que está em interacção directa com a impressora e que a partir de agora tudo o que escrever será executado.

Existem duas maneiras de programar em PostScript:

A primeira consiste em gerar um ficheiro PostScript a partir de um programa escrito noutra linguagem. Este é o modo utilizado pela maioria dos programas de Desktop Publishing disponíveis no mercado. Neste caso o docu-

mento realizado no ecrã é guardado na memória do PC no seu formato próprio. A tradução para PostScript é efectuada no momento da impressão.

O segundo método é escrever directamente um programa de PostScript. Isso é relativamente raro, dada a complexidade da linguagem. No entanto, ter alguns conhecimentos de base permite modificar os ficheiros PostScript criados por um outro programa e ter acesso a todas as suas potencialidades, o que é de uma grande utilidade para resolver alguns problemas que, às vezes, surgem na impressão.

PROGRAMAR EM POSTSCRIPT

O PostScript é uma linguagem dedicada à composição de documentos. Ela oferece ao programador uma grande quantidade de operadores que permitem a manipulação de objectos gráficos ou de cadeias de caracteres, em diferentes fontes e tipos. O PostScript é completamente independente da resolução do periférico utilizado para a saída final do documento. Isso significa que o mesmo ficheiro pode ser enviado, sem qualquer modificação, tanto para uma laser com uma definição de 300 pontos por polegada (dpi) como para uma Linotronic com 2540 pontos.

Para assegurar esta independência, um documento PostScript é descrito numa página virtual numa resolução arbitrária de 72,27 dpi. A transferência desta página virtual para o documento final na resolução do periférico é feita automaticamente pelo interpretador no momento da impressão. Para o utilizador, uma página de formato A4 (21x29,7 cm) terá, assim, 597,5x845 pontos. Note-se de passagem que ninguém está limitado aos valores destas coordenadas. Por exemplo, ninguém nos impede de traçar uma linha a partir do ponto 54,523 66,782. Por outro lado, o operador *scale* permite a qualquer um definir aquela que mais lhe convier. Mas já lá vamos.

Na origem das coordenadas, o ponto 0 0 encontra-se no canto inferior esquerdo. A lógica dizia-nos que devíamos começar pelo canto superior, mas isso iria fazer com que tivéssemos de endereçar todas as coordenadas em valores negativos, o que dificultaria a nossa programação.

PILHAS E DICIONÁRIOS

O PostScript é uma linguagem baseada na utilização de pilhas. Estas são do tipo LIFO (Last In, First Out: a última a entrar é a primeira a sair). Imagine uma pilha de revistas numa banca. A última a ser colocada na pilha é a primeira a ser comprada e a ser levada. Para termos acesso à terceira teremos de levantar as duas primeiras. O PostScript funciona da mesma maneira. Por exemplo, o envio do comando *2 3 add* ao interpretador PostScript provocará as seguintes acções:

o interpretador receberá um 2. Ele verifica se possui um operador com o mesmo nome. Como não tem vai colocar o 2 na pilha de instruções:

Pilha: 2

O Interpretador recebe o 3. Efectua a mesma busca e coloca o 3 no princípio da pilha:

**Pilha: 3
2**

O interpretador recebe o *add*. O *add* é um operador PostScript conhecido e começa a ser executado. O *add* provoca a adição dos dois primeiros valores da pilha — o 3 e o 2 são sucessivamente adicionados e o resultado da operação colocado, por seu turno, na pilha:

Pilha: 5

UM SÓ MODO DE TRABALHO

Contrariamente a certos interpretadores que distinguem o modo texto do modo gráfico, todos os objectos do

PostScript são considerados em modo gráfico. Um carácter passado como tal num programa PostScript será substituído na memória do interpretador pela sucessão de curvas e de linhas que o desenhavam.

Um documento PostScript é composto de uma ou várias páginas, cada uma dividida por um conjunto de caminhos (*path*) eles também compostos por objectos, texto ou gráficos (os quais, como já vimos, para o PostScript são a mesma coisa).

Aquando da composição do documento, cada caminho será definido em memória, depois desenhado na página virtual na forma de contornos (um traço é desenhado através do contorno do caminho) ou sob a forma de preenchimento (o caminho é preenchido com uma trama de cinzentos definida previamente). Depois de o conjunto de caminhos que preenchem a página terem sido desenhados na página virtual, ela será impressa no papel.

Um caminho (*path*) é formado de contornos, eles mesmo uma sucessão de linhas e curvas definidas com a ajuda de operadores gráficos ou sob a forma de traçado de caracteres.

Um caminho pode ser descontínuo, pois ele não tem qualquer existência real até ser desenhado na página virtual, seja em contorno (*stroke*) seja a cheio (*fill*).

Peguemos num exemplo. Para desenhar um quadrado escreve-se o seguinte programa (convém diferenciar as minúsculas das maiúsculas, pois o PostScript encara-as de maneira diferente):

```
newpath
50 50 moveto
50 60 lineto
60 60 lineto
60 50 lineto
closepath
stroke
showpage
```

O operador *newpath* define o começo de um novo caminho. A este nível de programação é necessário definir o ponto corrente (*currentpoint*) ponto de partida do desenho.

É o objecto da instrução *moveto*, na segunda linha que coloca o ponto corrente nas coordenadas especificadas na ocorrência (50 50).

Recorde-se que a origem das coordenadas se encontra no canto inferior esquerdo da página e que os parâmetros do operador se devem enviar antes do próprio operador.

A terceira linha traça o lado esquerdo do quadrado. O operador *lineto* junta ao caminho corrente um segmento de recta do ponto corrente até às novas coordenadas.

As linhas quatro e cinco traçam respectivamente os lados superior e direito do quadrado. Depois de cada operador gráfico o ponto corrente fica nas coordenadas do último ponto traçado.

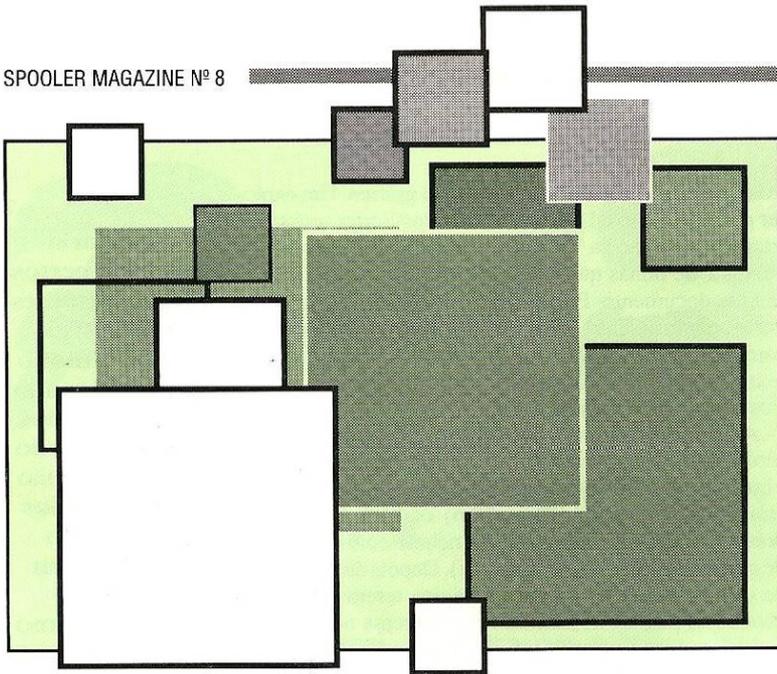
A linha 6 é de uma grande utilidade. Em vez de se escrever uma linha do género 50 50 lineto, o operador *closepath* encarrega-se, seja qual for o caminho desenhado, de o fechar juntando um segmento de recta entre o primeiro e o último ponto traçados.

É importante referir que até este instante nada foi desenhado na página virtual. Nós simplesmente definimos na memória um caminho para o contorno de um simples objecto. Esse caminho não tem, por enquanto, qualquer cor ou preenchimento.

O desenho do caminho na página virtual é efectuado com a ajuda do comando *stroke*. *Stroke* desenha na página virtual o contorno do caminho corrente. Diferentes parâmetros podem ser modificados antes do *stroke*. Esses parâmetros influenciam a largura do traço, o nível de cinzento, o tipo de junção entre os traços, etc.

Enfim o operador *showpage* provoca a transferência da

“
Contrariamente a certos interpretadores que distinguem o modo texto do modo gráfico, todos os objectos do PostScript são considerados em modo gráfico. Um carácter passado como tal num programa PostScript será substituído na memória do interpretador pela sucessão de curvas e de linhas que o desenhavam
”



página virtual para a memória da impressora e a saída da página impressa.

É possível de mudar o desenho do quadrado trocando o contorno por um fill:

```
newpath
50 50 moveto
50 60 lineto
60 60 lineto
60 50 lineto
closepath
0.5 setgray
fill
showpage
```

O comando stroke foi substituído por duas linhas:

setgray definindo a percentagem de cinzento utilizado pelo desenhado (atenção o setgray diz respeito tanto ao traçado das linhas como ao preenchimento do quadrado). O 0.5 representa um valor médio, 0 o negro e 1 o branco.

O fill provoca o preenchimento do caminho corrente por um cinzento de uma densidade definida pelo setgray (o 0 é o defeito).

É de notar que fill e stroke geram um newpath implícito e perdem todos os dados relativos ao caminho corrente. Depois de um dos dois comandos, o caminho corrente não é conhecido e o ponto corrente, indefinido.

Para desenhar no papel, por sua vez, um quadrado com um contorno a preto e o interior a cinzento podíamos escrever os dois programas um a seguir ao outro. Mas então imaginem o tamanho dos programas em PostScript se adoptássemos o método, de definir sempre o caminho de cada vez que quiséssemos escrever uma recta ou preencher um espaço.

Para evitar isso existem dois operadores que asseguram a salvaguarda e a restituição do contexto gráfico corrente. O termo contexto gráfico corrente reporta-se ao caminho corrente e aos parâmetros gráficos em vigor no momento da salvaguarda em memória (espessura da linha, cores, etc.). Assim, o nosso programa passaria:

```
newpath
50 50 moveto
50 60 lineto
60 60 lineto
60 50 lineto
closepath
gsave
0.5 setgray
fill
grestore
stroke
showpage
```

O gsave e o grestore encarregam-se, respectivamente, de gravar e reconstituir o contexto gráfico. O valor de defeito da cor de preenchimento (setgray) é 0, valor implícito no momento do gsave. Não é, portanto, necessário redefini-lo depois do comando stroke.

OS CARACTERES

O interpretador de PostScript possui, num dicionário particular, a descrição das diferentes fontes de letras (35 como standard da maioria das impressoras compatíveis PostScript).

```
/Times-Roman findfont 12
scalefont setfont
newpath
250 400 moveto
(Spooler) show
showpage
```

O exemplo acima precede a impressão de uma palavra (Spooler) em Times corpo 12.

/Times-Roman é o nome da fonte. A barra oblíqua e as maiúsculas indicam um nome literal. O operador findfont procura a fonte no dicionário de fontes. O operador scalefont dá-lhe o tamanho (corpo) desejado e o operador setfont define-a como fonte corrente. Todos os caracteres impressos nas páginas serão nesta fonte e neste corpo até o PostScript encontrar outro operador setfont. Como os exemplos anteriores, as linhas 2 e 3 definem o começo de um novo caminho e o ponto corrente. No caso da impressão, o ponto corrente é fixado a partir do canto inferior esquerdo de um quadrado imaginário que rodeia a primeira letra da frase.

Depois da impressão, o ponto corrente está situado no mesmo local, mas no quadrado imaginário que rodeia o último carácter da frase.

Uma cadeia de caracteres deve ser fechada entre parêntesis (linha 4) para ser reconhecida como tal pelo interpretador). O carácter \ (barra oblíqua) serve para introduzir os caracteres ilegais como o próprio parêntesis. Por exemplo a cadeia:

(O carácter \ é normalmente interdito)

é válida porque o segundo parentêsis está precedido da barra invertida \ que indica que ele pertence à cadeia de texto. O operador show projecta o desenho da cadeia para a página virtual e o showpage imprime no papel.

TRANSLACÇÕES, ROTAÇÕES E ESCALAS

Três operadores particulares modificam em parte ou na totalidade a apresentação do documento. O Translate efectua uma mudança de coordenadas no plano da folha. A linha:

```
100 100 translate
```

vai colocar o ponto de origem das coordenadas (o 0 0) no ponto 100 100. Embora seja possível, é de todo inútil definir o ponto de origem fora da área da folha.

De uma maneira geral, um traço cujas coordenadas estão fora da folha não provoca qualquer erro no PostScript. É simplesmente ignorado. O operador rotate efectua uma rotação no sentido trigonométrico (contrário aos ponteiros dos relógios).

Por exemplo:

```
30 rotate
```

vai enviar tudo o que se seguir com trinta graus em relação ao original.

O operador scale efectua uma mudança no tamanho no objecto.

A instrução:

```
0.5 2 scale
```

vai reduzir a metade o objecto relativamente ao seu

eixo horizontal e aumentá-lo no dobro relativamente ao eixo vertical. É de reparar a não correspondência entre estes vários operadores. Ou seja, uma translação seguida de uma rotação não dará o mesmo resultado que uma rotação seguida de uma translação. Estes três operadores modificam definitivamente o contexto gráfico corrente. A única maneira de voltar ao estado precedente é de aplicar o operador grestore (na condição, claro, de se ter feito um gsave antes de modificar o contexto). O método normal para efectuar uma mudança temporária do contexto é a seguinte:

```
... (programa PostScript)
gsave (gravar o contexto gráfico)
(scale, rotate, translate ou qualquer
outra
      modificação do contexto)
grestore (restituição do contexto
precedente)
... (continuação do programa)
```

ROTINAS

É sempre necessário qualquer que seja a linguagem de programação utilizada executar muitas vezes a mesma parte de um programa. Para isso existem três possibilidades:

- 1 – Escrever tantas vezes quantas as necessárias a mesma parte do programa, o que às vezes resulta se quisermos ganhar em rapidez, mas que se torna completamente ineficaz ao fim de meia dúzia de ocorrências.
- 2 – Definir um subprograma, uma rotina, que será chamada tantas vezes quantas as necessárias. Este método é utilizado desde que apareçam as mesmas linhas de programação em diversas partes do programa, permitindo diminuir o número total de instruções.
- 3 – Utilizar uma estrutura particular para se executar determinado número de vezes as mesmas linhas. Este método juntamente com o de cima é o melhor se quisermos executar muitas vezes a mesma operação apenas com a mudança regular de um parâmetro.

Voltemos ao primeiro exemplo. Para desenhar múltiplas vezes o quadrado na página podíamos contruir uma rotina chamada "quadrado" que será chamada quantas vezes forem necessárias:

```
/quadrado {
  10 0 rlineto
  0 10 rlineto
  - 10 0 rlineto
  closepath
  stroke
} def

newpath
10 10 moveto
quadrado
100 100 moveto
quadrado
-
-
-
showpage
```

Uma rotina pode-se definir assim:

```
/nome da rotina
{ instruções } def
```

O arranjo, em várias linhas, da rotina, não é obrigatório, mas apenas aconselhável para que seja mais compreensível e de fácil modificação. O operador rlineto é idêntico ao lineto com a única diferença de que a colocação das coordenadas é relativa ao último ponto traçado. A rotina quadrado vai desenhar um quadrado com dez unidades de lado onde o canto inferior esquerdo será o ponto corrente no momento da chamada pelo programa. Mas se nós quiséssemos reproduzir o quadrado um grande número de vezes a repetição das linhas x x moveto quadrado tornava-se fastidiosa. Podemos no entanto utilizar uma estrutura em bloco:

```
/quadrado {
  10 0 rlineto
  0 10 rlineto
  - 10 0 rlineto
  closepath
  stroke
} def

newpath
20 10 100 {
  dup moveto
  quadrado
} for
showpage
```

Esta estrutura assemelha-se ao ciclo:

FOR F = a TO b STEP c ... NEXT F

do Basic, onde a variável F toma todos os valores de a e b aumentando c em cada interacção. A representação em PostScript é a seguinte:

a c b (parte a executar) for

a, b e c têm o mesmo uso que no Basic. No nosso exemplo a variável de controlo (que não tem nome, ao contrário do Basic) toma todos os valores de 20 a 100 de dez em dez (20, 30, 40, 50... 100)

Num ciclo FOR/NEXT em PostScript a variável de controlo é duplicada na pilha na eventualidade de o programa ainda poder vir a precisar dela. Se o programa não a usar, retira-a da pilha. O comando pop vai buscar o último valor da pilha de instruções. No nosso exemplo a variável de controlo serve para posicionar o canto inferior esquerdo de cada quadrado (via operador moveto). Infelizmente o operador moveto precisa de duas instruções: as coordenadas X e Y do ponto de colocação. Para resolver este problema corneliano utilizamos o operador dup que duplica o a primeira instrução da pilha. Para melhor compreender o que se passa, vamos analisar a pilha durante a execução do ciclo:

- entrada no ciclo; valor inicial da variável de controlo: 20; a variável é empilhada.

Pilha: 20

- execução do operador dup; o valor que está em cima da pilha é duplicado:

Pilha: 20

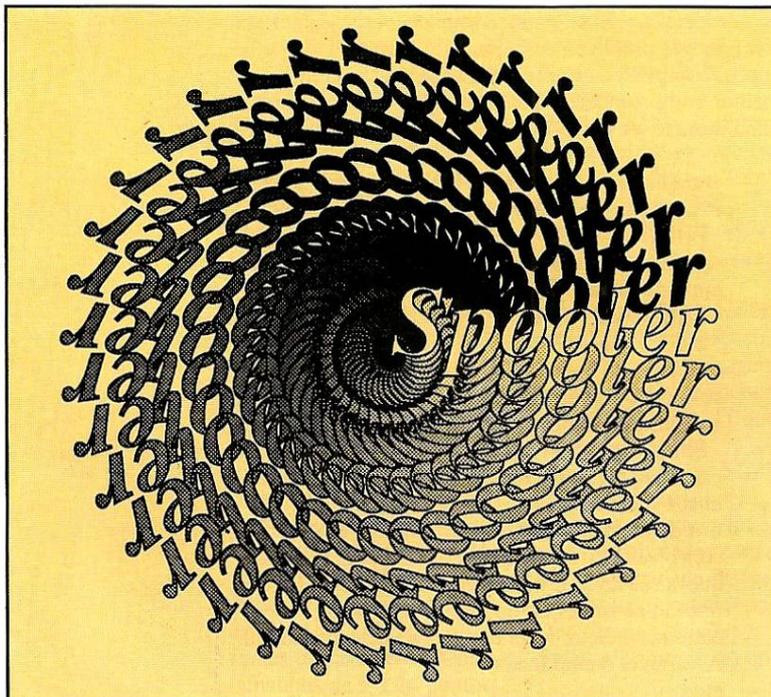
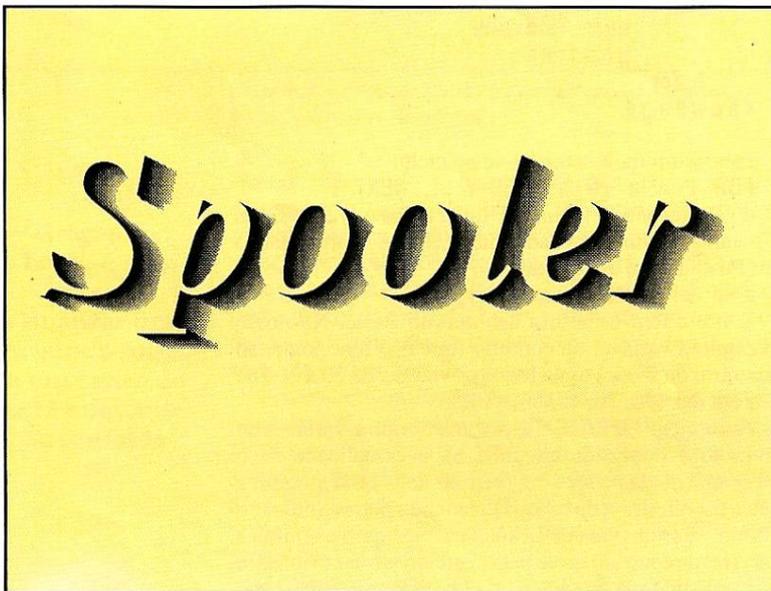
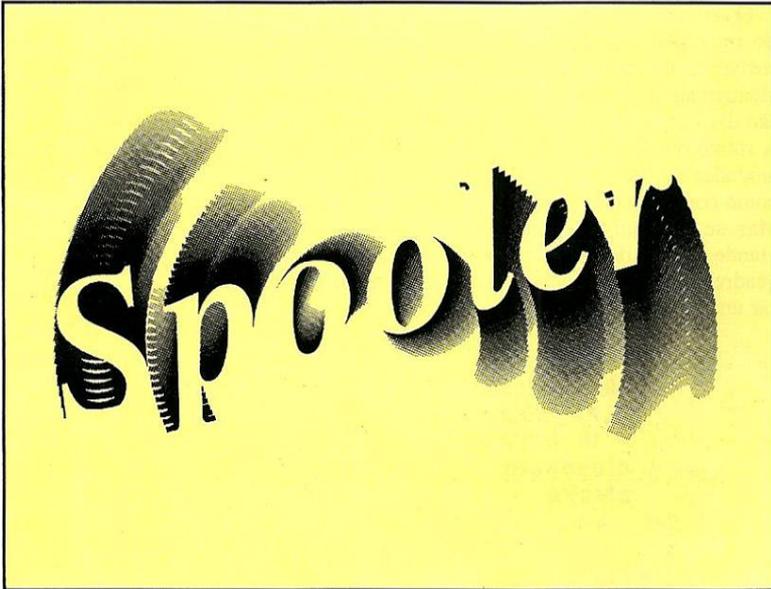
20

- execução do operador moveto; os dois valores que se encontram na pilha são retirados e servem para encontrar o ponto corrente:

Pilha: vazia com o ponto corrente em 20 20

- execução da rotina quadrado; o quadrado é desenhado a partir do ponto corrente 20 20
- fim do ciclo; comparação da variável de controlo com o valor do fim do ciclo; 20 é inferior a 200 pelo que a variável é incrementada em mais 10 e o retorno ao princípio do ciclo é efectuado.
- entrada no ciclo; valor da variável de controlo: 30. A variável é empilhada
- etc.

“
Num ciclo FOR/NEXT em PostScript a variável de controlo é duplicada na pilha na eventualidade de o programa ainda poder vir a precisar dela
 ”



Para ilustrar o funcionamento dos diferentes operadores que acabámos de descobrir aqui vai ao jeito de prato de resistência alguns programas muitos curtos e que ilustram bem o partido que se pode tirar de uma linguagem como o PostScript

% Texto em Rotação

```
/Times-BoldItalic findfont 60 scalefont setfont
/print { (Spooler) dup stringwidth pop 2 div neg 0 moveto
show } def
300 400 translate
315 rotate
.95 -.05 0 {
setgray print 2 rotate
} for
1 setgray print
showpage
```

% Texto em Translação

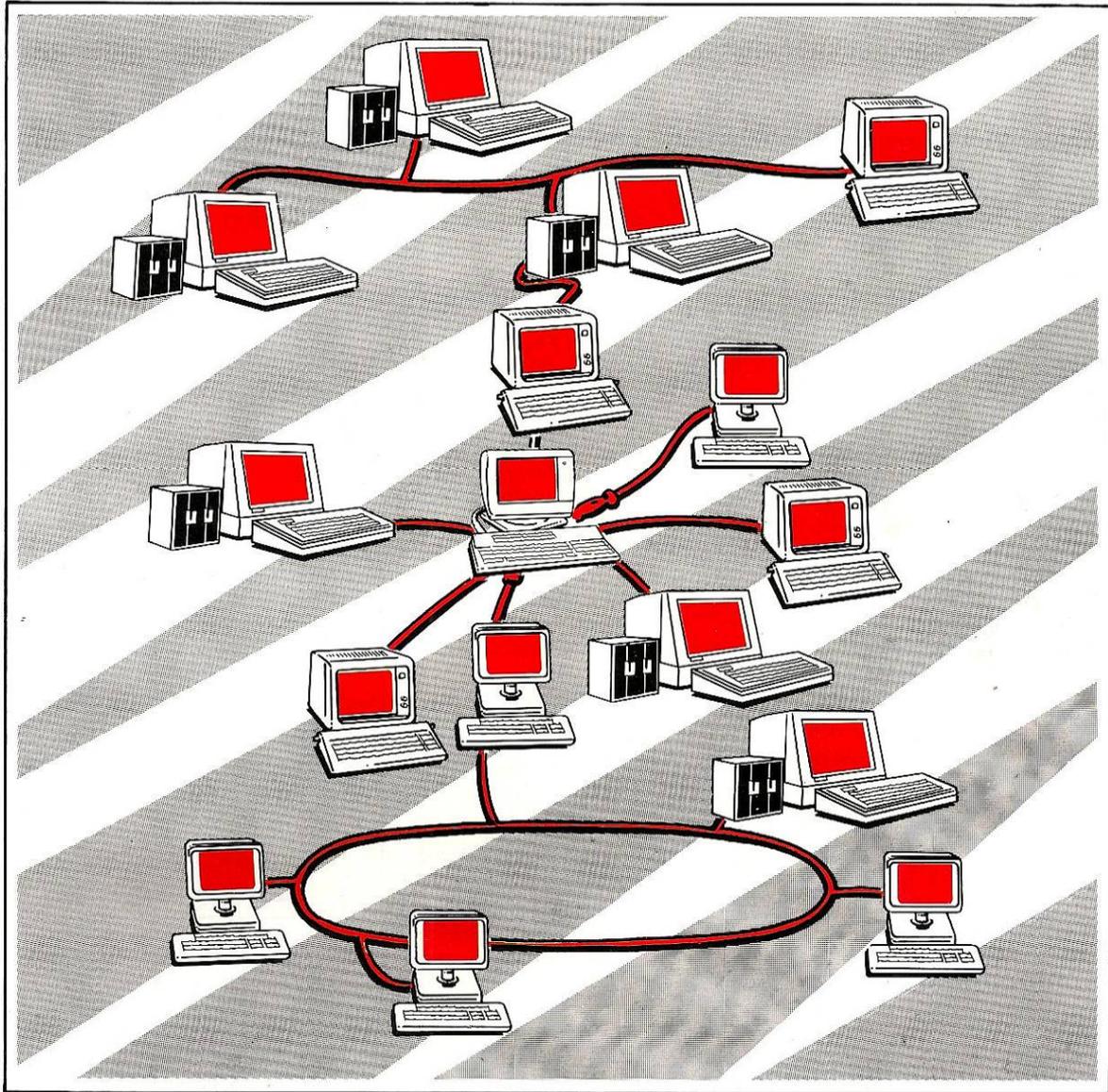
```
/Times-BoldItalic findfont 60 scalefont setfont
300 400 translate
.95 -.05 0 {
setgray 0 0 moveto (Spooler) show
-1 0.5 translate
} for
1 setgray 0 0 moveto
(Spooler) show
showpage
```

% Texto em rotação 2

```
/Times-BoldItalic findfont 30 scalefont setfont
/chaine (Spooler) def
0.5 setlinewidth
300 400 translate
10 10 360 { % princípio do ciclo
gsave
newpath
0 0 moveto
dup rotate
dup 360 div setgray
chaine false charpath
gsave fill grestore
0 setgray stroke
grestore
} for
showpage
```

A rotina print do primeiro destes programas merece algumas linhas de explicações suplementares. O objectivo é imprimir uma cadeia de caracteres em rotação em torno do seu centro seja qual for a sua dimensão, a sua fonte, ou o seu corpo. A rotina print está lá para calcular esse centro e o imprimir. A cadeia de caracteres Spooler é empilhada. Ela é, de seguida, duplicada (operador dup) porque o operador stringwidth que a segue precisa de uma cadeia de caracteres como instrução e não queremos perder o original. O operador stringwidth reenvia para a pilha a colocação horizontal e vertical do ponto corrente que provocará a impressão da cadeia. Neste caso só a primeira informação nos interessa, ela representa o comprimento da cadeia. Assim, o operador pop que se segue vai tirar o valor da colocação vertical. Resta na pilha o comprimento da cadeia. Este é dividido por dois (2 div) e invertido (neg). Temos agora na pilha um valor negativo representado metade do comprimento da cadeia de caracteres Spooler. Na linha seguinte vamos empilhar a colocação vertical que servirá para o moveto (0, pois não queremos mexer nesse eixo), em seguida o moveto coloca o ponto corrente nas coordenadas especificadas, ou seja ao meio da cadeia e atrás do ponto 0 0.

REDES NOVELL



JSL Designers

QUALQUER QUE SEJA A CONFIGURAÇÃO **NOVELL** É A SOLUÇÃO

A NOVELL é o maior fornecedor de produtos de comunicações de elevado desempenho. Os sistemas de rede NOVELL permitem obter em cada posto de trabalho capacidades até agora só disponíveis em sistemas médios.

O sistema operativo NETWARE é hoje uma norma aceite por todos os fabricantes, com mais de 300 000 instalações ligando 3 milhões de postos de trabalho no mundo inteiro.

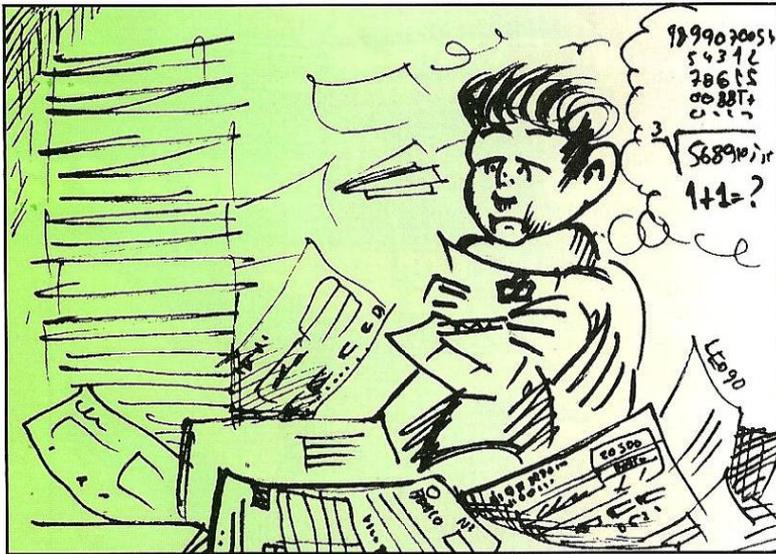
Os produtos NETWARE SQL, NETWARE VMS, NETWARE UNIX, NETWARE OS/2 e NETWARE Mackintosh, estabelecem a ponte entre os vários sistemas informáticos actuais.

A INFORGAL orgulha-se de oferecer tão importante factor de desenvolvimento às organizações portuguesas contando com um corpo técnico, comercial e de formação de elevada qualidade para suportar estas soluções.

INFORGAL
.....
A informática total

Um programa para gerir a sua conta bancária

CONTAS EM DIA



Julgo que o meu caso não será uma excepção. Aliás, estou convicto de que o mesmo se deve passar com a maioria das pessoas. Por muito cuidado que tenhamos, as coisas, por vezes, não batem certas, colocando-se-nos uma terrível dúvida: errámos as contas, ou esquecemos algum dado?

Estou a referir-me à gestão da nossa conta de depósitos à ordem e àquela tão trivial situação de precisarmos de saber, em qualquer momento, qual é, exactamente, o nosso saldo. E mais: sem termos de gastar horas e "fósforo" a fazer contas e a folhear papelada para sabermos quanto gastámos e em quê, e quanto ganhámos e como.

Na verdade, quando temos por norma fazer os nossos pagamentos por meio de cheque ou cartão e encarregamos o banco de liquidar as nossas facturas de electricidade, telefone, água, etc., das duas uma: ou desfrutamos de um confortável saldo com muitos "zeros" (situação difícil de conseguir para quem só sabe trabalhar), ou é preciso uma grande cautela com esse saldo... Os imprevistos acarretam, quase sempre, situações embaraçosas...

Este tipo de preocupações e a "mania" de fazer com que o computador cumpra as suas obrigações – trabalhar para o dono – levou-me a construir um programa capaz de gerir a minha conta bancária e, já agora, as dos estimados leitores da Spooler.

CONSTITUIÇÃO DO PROGRAMA

O programa foi construído em BASIC e compilado com o Turbo-Basic. No directório "BANCO", da disquete Spooler, o leitor tem o ficheiro BANCO.EXE, que como a sua extensão indica, corre directamente a partir do DOS. O ficheiro da listagem *fonte* em BASIC não foi incluído na disquete por absoluta falta de espaço.

Existem dois ficheiros de dados automaticamente criados, aquando do início dos lançamentos.

INSTALAÇÃO DO PROGRAMA

O programa BANCO.EXE pode ser instalado em disquete ou num directório de utilitários do disco rígido e "chamado" a partir dum "menu" em "batch". Se o leitor estiver interessado nesta "facilidade", diga-nos.

FUNCIONAMENTO DO PROGRAMA

Quando se chama o programa aparece um ecrã com o seguinte aspecto:



Começemos pela opção "6 – Dados Iniciais". Esta opção destina-se à introdução dos seguintes elementos:

- Nome do banco (6 caracteres)
- Nome da dependência (10 caracteres)
- Nº da conta (10 caracteres)
- Nome do titular (12 caracteres)
- Saldo inicial (10 dígitos – 99999999.9)
- Data inicial (8 dígitos – ddmmaaa)

O "Saldo Inicial" (ou transporte) será, naturalmente, o valor em saldo na altura em que se começar a utilizar o programa. Como sei que o leitor é uma pessoa metódica, terá em seu poder, e devidamente actualizado, um daqueles impressos que os bancos fornecem com os livros de cheques: retire daí o valor do seu saldo. A data inicial será, obviamente, a desse saldo.

Feito isto, passemos à opção "2 – Lançamentos D/C", isto é, a opção que nos permite dar entrada aos valores dos nossos pagamentos ou recebimentos. O programa colocará as seguintes questões:

- Data do lançamento (ddmmaa)
- Referência (Nº do cheque, etc – 6 caracteres)
- Descrição (da operação – 20 caracteres)

- Valor do lançamento (10 dígitos)
- Débito ou Crédito (D/C) – se for um pagamento será débito (D); se for um recebimento será crédito (C)

Terminadas as entradas, o programa dá-lhe oportunidade de conferir os elementos introduzidos, antes de os passar ao ficheiro: se houver erros pode repetir o lançamento; se tudo estiver correcto terá de responder à pergunta “Mais lançamentos?” – premindo “s” o ecrã fica preparado para o próximo lançamento; premindo “n” o programa regressa ao MENU PRINCIPAL.

Nesta altura estamos em condições de ver a opção: “4 – Saldo à data de hoje”, ou a opção: “1 – Extracto da Conta”.

A primeira apresenta o saldo inicial e calcula e apresenta o saldo actual, tendo em conta os lançamentos acabados de fazer. O extracto da Conta tem duas alternativas:

- Extracto Total (são listados todos os lançamentos)
- Extracto por Mês (só são listados os lançamentos do mês que for indicado)

Os extractos obedecem ao mesmo critério de apresentação dos seus homólogos bancários, salvo num aspecto que satisfaz um dos requisitos que considero importantes: a última linha dá-nos o somatório dos débitos e dos créditos, isto é, quanto já pagámos e quanto já recebemos.

NOTA: Qualquer destas alternativas permite a listagem para o ecrã, ou para a impressora.

A opção “3 – Corrigir lançamentos” obriga a que se faça uma listagem para a impressora, pois só esta modalidade apresenta o número do item, ou seja, o número do registo de cada lançamento. Detectado um erro, toma-se nota do item respectivo e será esse número a ser intro-

duzido nesta opção. As operações de correcção são intuitivas, bastando seguir as instruções do programa.

O leitor recebe, periodicamente, o extracto do seu banco. Se as suas contas de electricidade, telefone, água, etc., forem pagas pelo banco, será somente nessa altura que conhece a data desses pagamentos, o que lhe permitirá completar os lançamentos dum determinado mês. O mesmo acontece se, por exemplo, tiver depósitos a prazo e os juros respectivos forem creditados na conta de depósitos à ordem: através do extracto do banco terá acesso à data desses créditos e ao valor exacto dos juros.

Se tirar um extracto após ter feito esses últimos lançamentos verificará que os mesmos não se encontram por ordem cronológica. A opção “3 – Ordenar por data”, como o seu nome indica, coloca os lançamentos na sua devida ordem.

A finalidade da opção “7 – Cópias de segurança” é óbvia. Mas não será de mais repetir que estas deverão ser feitas sempre que se tenham feito ou corrigido lançamentos: a EDP não o compensa pelo trabalho perdido no caso de falha de corrente...

Chegámos, finalmente, ao fim do ano e o banco enviou-nos o extracto de Dezembro: podemos, então, terminar os lançamentos e fechar o ano em curso através da opção “8 – Fecho do ano”.

Como esta operação acarreta a destruição dos dados do ficheiro, está sujeita a confirmação: o saldo é passado para o ficheiro dos dados iniciais e o ficheiro dos lançamentos é apagado. Podemos começar o novo ano sem ser necessário repetir a entrada dos dados iniciais.

Como habitualmente, estamos à sua disposição para o auxiliar a resolver qualquer eventual dificuldade.

Desejo-lhe uma “saudável” conta bancária.

 Renato Casquilho

“
Terminadas as
entradas, o
programa dá-
lhe oportuni-
dade de
conferir os
elementos
introduzidos,
antes de os
passar ao
ficheiro
”

Graficria

- ⇒ Executamos todo o tipo de trabalhos gráficos com criatividade, profissionalismo e competência
- ⇒ Utilizamos a eficácia e rapidez da edição computadorizada aliada à experiência das artes gráficas convencionais
- ⇒ Oferecemos alta qualidade por baixo preço
- ⇒ Cumprimos prazos
- ⇒ Contacte-nos

VISUALIZAÇÃO CIENTÍFICA



“
O conceito de
visualização
científica já
vem da anti-
guidade,
quando o
homem das
cavernas
desenhou pela
primeira vez
nas paredes
da sua
caverna o
ambiente do
local onde
viveu
”

O conceito de visualização científica já vem da antiguidade, quando o homem das cavernas desenhou pela primeira vez nas paredes da sua caverna o ambiente do local onde viveu. Numa lenda grega, Arquimedes foi abatido por um soldado romano enquanto estudava algumas figuras desenhadas na areia. No século presente, químicos começam a perceber a estrutura das moléculas visualizando-a numa tela do computador. Assim fazem também os engenheiros aeronáuticos, os físicos, os geólogos e demais, que recorrem ao computador para obter uma simulação gráfica do problema que estão a estudar.

A visualização computacional não é mais um suplemento útil para a investigação científica, e passou a ser uma ferramenta indispensável para qualquer investigador. Desde o mapeamento do solo oceânico utilizando sonares até ao estudo dos céus recorrendo a radiotelescópios, a quantidade de dados gerados é tão grande que impossibilita o seu estudo só através dos números. Sendo assim, estes são convertidos em imagens as quais fornecem ao investigador uma visão simulada e de fácil percepção do problema.

CURVAS DE NÍVEL

Neste artigo pretendo apresentar um programa para a visualização científica de curvas de nível.

A versão apresentada é uma versão simplificada (mesmo assim, são mais de 4000 linhas de código) do programa. Não pretendo que este seja um programa comercial, mas sim um programa de estudo, no qual posso descrever as várias técnicas e algoritmos utilizados na computação gráfica. O programa foi desenvolvido sobre a placa gráfica EGA, visto ser esta a placa gráfica mais popular entre os utilizadores de PC e que fornece uma resolução

razoável. É de salientar que, nos meios gráfico-computacionais são considerados *workstations* minimamente satisfatórias para a computação gráfica as que possuem 8 planos de cor e 24 bits por pixel numa resolução de 1024x1024, o que corresponde à possibilidade da existência de $2^8 = 256$ cores simultâneas numa paleta de $2^{24} = 16777216$ cores. Na tabela seguinte, apresento uma comparação entre as várias placas gráficas mais comuns suportadas pelos PC's e de uma *workstation* comum:

Placa	Resolução máxima	Nº de cores simultâneas	Nº de cores possíveis
CGA	320x200 640x200	4 1	7 dividida em 2 paletes
EGA	640x350	16	$2^6 = 64$
VGA	350x200 640x480	256 16	$2^{18} = 262144$
"work-station"	1024x1024	256	$2^{24} = 16777216$

Comparando os valores da tabela, podemos concluir que é ridículo desenvolver aplicações gráficas nas quais o realismo seja fundamental numa placa CGA, o que sugere que possuidores desta placa e que estão interessados na computação gráfica devem começar a pensar em adquirir uma EGA ou VGA.

Voltando ao programa, as características principais deste é de permitir ao utilizador (um geólogo ou geógrafo), após ter adquirido dados numéricos numa determinada zona do solo, visualizar uma simulação em três dimensões do solo em questão. Uma primeira tarefa que é normalmente feita por este especialista, após a obtenção de dados numéricos através de várias medidas cuidadosas no local, é a elaboração de um mapa de curvas de nível. Estas são curvas fechadas e sem qualquer intersecção entre elas, e que representam uma altura, normalmente em relação ao nível médio das águas do mar, dum determinado terreno. Claro que só pelo mapa não se consegue ter uma noção clara do relevo e dos acidentes do terreno, mas se podemos obter um modelo visual deste, consegue-se perceber melhor o terreno a ser estudado. Portanto, o programa permite que o utilizador indique pontos que correspondem a uma determinada altura, numa grelha de resolução variável, entre 1 e 100. A altura dum ponto é seleccionada variando um potenciômetro em forma dum *scroll-bar*. Para indicar um ponto na grelha, primeiro deve seleccionar uma altura colocando o apontador do rato sobre o *scroll-bar* e premir o botão esquerdo do rato. Para incrementos ou decrementos de uma unidade, deve colocar o apontador do rato sobre um dos extremos do *scroll-bar*. Depois deve *clickar* no ícone do canto superior esquerdo para poder começar a indicar pontos na grelha. Esta serve simplesmente para assegurar

que o utilizador não vai indicar pontos que não quer, acidentalmente. Assim, quando o utilizador terminar a tarefa de indicar pontos na grelha, deve pôr aquela *flag* novamente a *off* antes de proceder qualquer outra acção tornando, assim, a grelha inacessível caso o utilizador posicione o apontador do rato sobre a grelha e prima o botão esquerdo. Para seleccionar um ponto na grelha, basta posicionar o apontador do rato sobre o ponto pretendido e clicar no botão esquerdo do rato. Ao seleccionar um segundo ponto, será traçada uma recta que une estes dois pontos, e assim por diante, até o utilizador premir o botão direito do rato para que o primeiro ponto una com o último, formando assim uma linha fechada. Sendo assim, cada ponto numa curva de nível deve ser indicado numa só passagem, o que se resume na seguinte sequência de tarefas:

- 1 – Pôr a *flag* de desenho na grelha a *on*.
- 2 – Seleccionar altura, variando o *scroll-bar*.
- 3 – Seleccionar pontos numa mesma curva de nível na grelha premindo o botão esquerdo do rato.
- 4 – Fechar a curva de nível premindo o botão direito do rato.
- 5 – Proceder à seguinte curva de nível a partir da tarefa 2.
- 6 – Se não houver mais curvas, pôr a *flag* de desenho a *off*.

Para visualizar a imagem em três dimensões, basta *clickar* no ícone correspondente sem antes de seleccionar o método de interpolação pretendido e o modo de visualização, de que falarei mais adiante. Existe um ícone de ajuda (*Help*) o qual indica ao utilizador a função de cada um dos doze ícones. Sugiro ao utilizador que leia atentamente as instruções fornecidas antes de começar a trabalhar. Para além dos três ícones já mencionados, existem ainda outros que resumirei a seguir:

– Apagar a grelha

Permite ao utilizador apagar a grelha corrente, criando uma nova grelha com uma nova resolução e altura inicial fornecida pelo utilizador.

– Refrescar a grelha

Quando a grelha se torna demasiada confusa devido às linhas que são desenhadas para ligar os vários pontos indicados, ao seleccionar este ícone, a grelha é redeseenhada. Nota que os pontos indicados permanecem na grelha, portanto não confundir esta acção com a de apagar.

– Alteração da fonte de iluminação

Permite ao utilizador alterar a direcção da fonte de iluminação quando o modo de visualização com as faces dos polígonos preenchidas for seleccionado.

– Load

Carregar, a partir dum ficheiro guardado anteriormente, uma grelha.

– Save

Guardar a grelha actual num ficheiro.

– Corte a duas dimensões (2 D)

Permite o corte do terreno segundo uma recta. Quando esta acção é seleccionada, o programa mostra-nos uma recta horizontal a meio da grelha. Para alterar a direcção da recta, basta aproximar o apontador do rato a um dos extremos da recta e premir o botão esquerdo do rato. Mantendo o botão esquerdo premido e movendo o rato, podemos alterar a direcção da recta até ao ponto desejado, onde devemos largar o botão do rato. Quando a recta

estiver na posição desejada para fazer o corte, devemos premir o botão direito do rato, para que a imagem do corte nos seja mostrada.

– Corte a três dimensões (3 D)

Semelhante ao anterior, só que em vez duma recta, são duas rectas perpendiculares entre si.

– Sair

Quando desejamos terminar o programa, devemos seleccionar este ícone.

– Autoria

Mostra o nome do autor (eu!) que fez o programa. Note que esta parte não consta da listagem fornecida, devido a motivos óbvios. Esta falta não prejudicará a compreensão do programa para quem vai ler a listagem, nem provocará erros para quem vai recompilar as fontes. A única diferença é que, quando alguém recompilar as fontes do programa, a autoria desaparecerá. Para os que dominam a linguagem Pascal, não será difícil descobrir onde isto é feito.

Para finalizar a descrição do modo de utilização do programa devo referir que, dum modo geral, o botão esquerdo do rato serve para indicar ou evocar uma determinada acção do programa e o botão direito ou, em alguns casos a tecla de Escape, serve para abortar a acção. O botão direito do rato também é utilizado para indicar o fim duma acção em alguns casos, como as que mencionei anteriormente.

MÉTODOS DE DESENHAR LINHAS

Os dois métodos que a seguir vou descrever são geralmente aplicadas em *displays* de *raster* (duma forma geral, as que possuem uma memória video de acesso aleatório), em oposição aos *displays* por refrescamento de vector ou outras tecnologias. Para um conhecimento mais profundo das tecnologias dos *displays* sugiro a consulta de um dos livros sobre computação gráfica que apresentarei no fim do artigo.

Uma forma bastante simples de desenhar linhas é recorrer aos nossos conhecimentos de matemática, mais propriamente da geometria analítica. Uma equação geral de uma recta é a seguinte :

$$aX + bY + c = 0$$

Esta equação é bastante geral, só que requer muitos cálculos tomando-a imprópria para ser programada. Note-se que, para $b = 0$ é $a \neq 0$, e então $aX + bY + c = 0$ é equivalente a $X = -(c/a)$, que define uma recta vertical. E para $b \neq 0$, $aX + bY + c = 0$ é equivalente a $Y = -(a/b)X - (c/b)$ o que traduz-se na equação reduzida de uma recta, $Y = mX + b$, em que m é o declive da recta, que pode ser calculada como $m = dY/dX$, (d representa "delta") em que dY e dX são as variações dos valores de Y e X na recta, respectivamente, e a constante b é a origem da ordenada. Temos assim uma forma bastante simples de calcular rectas não verticais. Então, para determinar qualquer ponto sobre a recta r definida pelos pontos (X_1, Y_1) e (X_2, Y_2) , basta recorrer à equação $Y = mX + b$, com $m = (Y_2 - Y_1)/(X_2 - X_1)$. Se pretendemos desenhar a recta incrementalmente, basta neste caso, determinar o valor de m . Se este for menor ou igual a 1 então a cada incremento de X em uma unidade, $X_{i+1} = X_i + 1$, corresponde a um incremento de Y em m unidades, $Y_{i+1} = Y_i + m$. Caso m for maior que 1, o papel de X e de Y devem ser invertidos, incrementando Y em uma unidade e X em $1/m$ unidades. Obtemos assim um procedimento bastante fácil de desenhar linhas :

“
Para visualizar a imagem em três dimensões, basta *clickar* no ícone correspondente sem antes de seleccionar o método de interpolação pretendido e o modo de visualização, de que falarei mais adiante
”

“
Um outro método bastante mais simples e “inventado” por mim, a que batizei de erosão, utiliza um método de interpolação mais simples
 ”

```

procedure DDALinha(x1,y1,x2,y2,nivel,color:integer);
var x,y,xinc,yinc:single;
    i,length,deltax,deltay:integer;
begin
    deltax := x2 - x1;
    deltay := y2 - y1;

    length := abs(deltax);
    if abs(deltay) > length then length := abs(deltay);
    if length <= 0 then
    begin
        xinc := (deltax)/length;
        yinc := (deltay)/length;

        x := x1 + 0.5 ; y := y1 + 0.5;

        for i := 1 to succ(length) do
        begin
            Dot(trunc(x),trunc(y),nivel,color);
            x := x + xinc;
            y := y + yinc;
        end
    end
end;
    
```

O algoritmo acima tem a desvantagem de trabalhar sobre números reais: por exemplo, Y tem que ser arredondado para um número inteiro antes de ser aplicado no ecrã, e as operações com números reais são lentas, a não ser que o utilizador possua um co-processor matemático. Um outro algoritmo, desenvolvido por Bresenham, tem a vantagem de só trabalhar com números inteiros, e portanto é frequentemente utilizado para a implementação em hardware. Devido à complexidade deste algoritmo, não o apresentarei em pormenor neste artigo, aconselhando o leitor, mais uma vez, a consultar a bibliografia sugerida. Mas como uma abordagem superficial, o algoritmo recorre a uma variável de erro **err**, a qual, a cada passo indica a proximidade da recta num ou noutro ponto vizinho por onde passa a recta, decidindo por este método qual dos pontos deve ser activado como sendo um ponto sobre a recta. Este algoritmo traduz-se no seguinte procedimento:

```

procedure BresLinha(x1,y1,x2,y2,nivel,color:integer);
var i,err,deltax,deltay,incx,incy,tmpx,tmpy:integer;
begin
    deltax := x2 - x1;
    deltay := y2 - y1;

    if deltax > 0 then incx := 1
    else if deltax = 0 then incx := 0
    else incx := -1;

    if deltay > 0 then incy := 1
    else if deltay = 0 then incy := 0
    else incy := -1;

    deltax := abs(deltax); deltay := abs(deltay);
    tmpx := 2*deltax; tmpy := 2*deltay;

    if deltax > deltay then
    begin
        err := tmpy - deltax;
        for i := 0 to deltax do
        begin
            Dot(x1,y1,nivel,color);
            if err > 0 then
            begin
                y1 := y1 + incy;
                err := err + (tmpy - tmpx)
            end
            else err := err + tmpy;
            x1 := x1 + incx;
        end
    end

    else
    begin
        err := tmpx - deltay;
        for i := 0 to deltay do
        begin
            Dot(x1,y1,nivel,color);
            if err > 0 then
            begin
                x1 := x1 + incx;
                err := err + (tmpx - tmpy)
            end
            else err := err + tmpx;
            y1 := y1 + incy;
        end
    end
end;
    
```

Este último algoritmo foi utilizado para ligar dois pontos que o utilizador seleccionou na grelha.

MÉTODOS DE INTERPOLAÇÃO

O que é uma interpolação? Ao definir dois pontos quaisquer e tentarmos unir estes dois pontos por uma linha, recta ou curva, temos que determinar os pontos intermédios entre estes dois pontos e os que estão sobre a linha. No caso de uma recta, já vimos como isso é feito no parágrafo anterior, e o método utilizado chama-se **interpolação linear**. Existem outros métodos, cada qual com as suas vantagens e desvantagens. No caso de a linha ser curvilínea, o método de interpolação linear dá resultados pouco realistas. Os **métodos de Bézier** ou **B-Spline** são os mais utilizados para obter linhas curvilíneas ou superfícies curvas. Nestes dois métodos são aplicadas as equações paramétricas cúbicas. Uma equação paramétrica cúbica é um polinómio do terceiro grau com um parâmetro t, que para o nosso caso $0 \leq t \leq 1$. Podemos, então, representar as coordenadas de X, Y e Z em função de t, como sendo :

$$\begin{aligned}
 X(t) &= a_x * t^3 + b_x * t^2 + c_x * t + d_x \\
 Y(t) &= a_y * t^3 + b_y * t^2 + c_y * t + d_y \\
 Z(t) &= a_z * t^3 + b_z * t^2 + c_z * t + d_z \\
 \text{com } 0 &\leq t \leq 1
 \end{aligned}$$

Podemos então determinar a equação paramétrica a utilizar no método de Bézier, como:

$$P(t) = \sum_{i=0}^n p_i B_{i,n}(t)$$

com $B_{i,n}(t) = C(n,i) * t^i * (1-t)^{n-i}$

em que $C(n,i) = n! / (i!(n-i)!)$ é a combinação de n elementos i a i.

Para uma superfície, podemos determinar a seguinte equação:

$$P(u,v) = \sum_{i=0}^n \sum_{j=0}^n p_{i,j} B_{i,n}(u) * B_{j,m}(v)$$

Para o método de B-Spline temos:

$$P(t) = \sum_{i=0}^n p_i N_{i,k}(t)$$

com a função N (função básica de B-Spline) determinada recursivamente como :

$$N_{i,1}(u) = 1 \text{ se } t_i \leq u \leq t_{i+1}$$

O noutros casos;

$$N_{i,k}(u) = \frac{(u-t_{i-1})N_{i,k-1}(u)}{t_{i+k-1} - t_{i-1}} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}}$$

e generalizando-a para uma superfície, temos :

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^m p_{i,j} N(u)_{i,k} * N(v)_{j,l}$$

Um outro método bastante mais simples e "inventado" por mim, a que batizei de **erosão**, utiliza um método de interpolação mais simples. Simplesmente calculo a média da soma do ponto em questão com os seus vizinhos, fazendo isso em várias passagens. Consigo desta maneira nivelar a superfície. Note-se que, quando o número de passagens for demasiado grande, pode acontecer que a superfície se torne num plano cuja altura seja a média de todos os seus pontos que a compõe. Por isso, existe um outro método, a **erosão fixa**, que mantém os pontos da linha indicada pelo utilizador. Para que servem os algoritmos de interpolação neste contexto? Após a indicação das várias curvas de nível pelo utilizador, temos de calcular a altura dos pontos da grelha que não foram indicadas pelo mesmo, interpolando-as à altura correspondente, antes de serem visualizadas em três dimensões.

ILUMINAÇÃO

A iluminação é aplicada para dar um maior grau de realismo na visualização em três dimensões. Devido a limitações de hardware das placas gráficas (com excepção da VGA), muitas vezes obtemos resultados que ficam muito por desejar. Sendo assim, a técnica de iluminação que foi adoptada foi só a da utilização da reflexão difusa e da luz ambiente, ficando de fora a reflexão especular, assim como as sombras e as texturas. A reflexão difusa é a reflexão da luz incidente sobre uma superfície homogénea a qual é reflectida em todas as direcções com a mesma intensidade. Portanto podemos calculá-la utilizando a lei do co-seno do ângulo entre a normal à superfície e a direcção da fonte de iluminação, a que se traduz na **lei de Lambert**. Temos assim a equação :

$$I_d = I_r * K_d * \cos(t)$$

em que I_d é a intensidade da reflexão difusa, I_r a intensidade da fonte de iluminação, K_d o coeficiente de reflexão difusa, que varia entre 0 e 1 dependendo do material que a superfície é composta, e o t ângulo já referido. Se nós calcularmos o vector da normal à super-

fície e do vector da direcção da fonte de iluminação, $\cos(t) = N \cdot L$, em que N é o vector da normal e L o da direcção da fonte (nota-se que o ponto representa o produto interno destes dois vectores). Se utilizarmos simplesmente esta equação, as superfícies não iluminadas directamente pela fonte ficam completamente às escuras o que corresponde pouco à realidade, visto que existe sempre uma luz bastante ténue devido à reflexão da luz da fonte por outros objectos na cena, à qual chamaremos de iluminação ou luz ambiente. Temos assim a equação:

$$I = I_a * K_a + I_r * K_d * (N \cdot L)$$

em que I_a e K_a são a intensidade da luz ambiente e o coeficiente da luz ambiente, respectivamente, e I a intensidade a aplicar numa superfície. É esta a equação que utilizei no programa.

CONCLUSÃO

Devido ao espaço disponível e à complexidade de alguns algoritmos que requerem um conhecimento mais profundo da geometria analítica, fiz somente uma abordagem bastante superficial dos algoritmos utilizados no programa. Omiti ainda outros, como a projecção oblíqua que foi aplicada para a visualização em três dimensões e a utilização de linhas de crista para a eliminação de linhas invisíveis (não confundir com a eliminação de faces invisíveis), as quais requerem um conhecimento de algumas noções básicas de computação gráfica. Num dos próximos artigos tentarei descrever com mais pormenor estes algoritmos. Como em qualquer software com algumas milhares de linhas de código, o programa que desenvolvi não está livre de "bugs", se bem que ainda não tenha encontrado nenhum. Caso o leitor encontre algum, agradeço que me envie uma carta reportando o sucedido.

Vicente Luís Gracias

Bibliografia :

- 1 - Fundamentals of Interactive Computer Graphics
James D. Foley & Andries Van Dam
- 2 - Computer Graphics - Systems & Concepts
R. Salman & M. Slater
- 3 - An Improved Illumination Model for Shaded Display
Turner Whitted
Artigo inserido na Communications of A.C.M. de Junho de 1980

“
A iluminação
é aplicada
para dar um
maior grau de
realismo na
visualização
em três
dimensões
”



MODEMS



“
Os modems (MODulator-DEModulator) surgiram com a necessidade de dois computadores poderem comunicar à distância entre si
 ”

Hoje em dia, quase toda a gente ligada ao meio informático já precisou ou ouviu falar em Modems.

Os modems (MODulator-DEModulator) surgiram com a necessidade de dois computadores poderem comunicar à distância entre si, ou seja: podemos definir um modem como um dispositivo que converte um conjunto de bits em série, transmitindo um sinal modulado por uma linha telefónica, podendo também reverter o sinal recebido num conjunto de bits em série. O modem é, então, inserido entre uma porta de comunicação série e uma linha telefónica.

É agora a melhor altura para falarmos das linhas telefónicas onde vão ligar os modems. Os CTT põem-nos à disposição dois tipos de linhas: banda de base e banda de canal. Para estes dois tipos de linha vão existir dois tipos de modems.

Torna-se, assim, importante definir estes dois tipos de linhas: podemos dizer que uma linha banda de base não é limitada em frequência, mas a sua atenuação, mediante a distância, limita a sua utilização máxima entre 15 e 20km. Numa linha banda de canal a atenuação já não é crítica, pois os CTT garantem-nos valores de atenuação aceitáveis, independentemente da distância, mas a linha é limitada em frequência entre 600Hz e 3000Hz.

Podemos então pedir aos CTT uma linha privada, banda de canal ou banda de base, consoante a distância necessária ou, ainda, utilizar a linha que o nosso telefone utiliza, para ligar um modem banda de canal, pois elas são banda de canal, como é óbvio. Quando pedimos uma linha banda de canal aos CTT, se não referirmos a qualidade, eles dão-nos uma qualidade do tipo 1040. No entanto, podemos pedir uma qualidade superior, 1025 ou 1020. Genericamente, uma linha 1040 servirá para comunicações até 2400 bps, 1025 até 4800 bps e 1020 até 9600 bps. Na prática consegue-se muitas vezes numa linha 1040 comunicar a 9600 bps.

Ao falarmos de modems banda de canal e banda de base para serem utilizados nas respectivas linhas, é importante referir a diferença mais significativa entre eles – o sinal analógico, ou seja, o tipo de modulação utilizada.

Os modems banda de base não têm grandes problemas na frequência que utilizam na linha, pois estas não a limitam dentro dos valores estabelecidos anteriormente. Por isso, nos modems mais sofisticados são por vezes utilizadas modulações bifásica e FSK (neste caso com uma frequência de portadora na ordem dos 60 a 100KHz, valor este que faz com que o modem seja mais imune ao

ruído da linha) e modulações em amplitude modulada, com uma frequência de portadora na ordem dos 4800Hz a 9600Hz. Nos modems menos sofisticados, não se utiliza qualquer espécie de modulação, ou seja, os modems funcionam como amplificadores de linha, neste caso para curtas distâncias – 3 ou 4km.

Os modems de banda de canal são os mais complicados, em termos de hardware, devido ao tipo de modulação que utilizam.

As modulações mais utilizadas pelos diversos modems são basicamente de 4 tipos: FSK (Frequency Shift Keying), QAM (Quadrature Amplitude Modulation), DPSK (Differential Phase Shift Keying) e TCM (Trellis Code Modulation).

A modulação FSK é a mais simples, consistindo apenas em termos uma frequência central fixa, e quando queremos transmitir, por exemplo, 1 bit, basta deslocar a frequência central para um valor inferior ou superior, consoante o bit tenha valor 0 ou 1. Exemplo disso é o que os modems V21 utilizam, em que um deles tem uma frequência central de 1080Hz, e no outro, 1750Hz, existindo uma variação de 100Hz, quando queremos transmitir 1 bit. Outro exemplo desta modulação é o que utiliza os modems V23, com outros valores de frequência central e com uma variação de 400Hz.

A modulação DPSK é mais complexa. Vamos ter uma frequência central na mesma, mas em vez de haver uma variação na frequência central, vai haver uma variação na fase. Temos como exemplo a recomendação V22, que agrupa bits 2 a 2, indo estes então provocar uma variação na fase: 00 vai corresponder a uma mudança de fase relativa à inicial, de 90 graus e assim por diante.

A modulação QAM consiste agruparmos vários bits sucessivos, em que o primeiro bit ou os primeiros vão definir a amplitude, e os outros bits, a mudança de fase relativa. Temos também como exemplo a recomendação V29 que agrupa bits 4 a 4, em que o primeiro, serve para determinar a amplitude relativa à fase do sinal a transmitir, e os outros 3 bits vão então definir a diferença de fase relativa à inicial: 101 corresponde a uma variação de fase 315 graus.

Por último, a modulação TCM é a mais complexa, agrupando um conjunto de bits para definir a mudança de fase e amplitude, mas os primeiros bits são descodificados e, ao mesmo tempo, é gerado um bit redundante. Exemplo disso é o que utilizam as normas V32 e V33, utilizando esta última 7 bits para modular o sinal, em vez dos 5 da norma V32. É fácil chegar à conclusão que conforme queremos transmitir com mais velocidade, assim será necessário termos mais bits para modular o sinal. É importante distinguir a diferença entre o sinal que o modem mete na linha e o baud rate. Nos casos do V32 e V33 ambos metem na linha uma frequência de 1800Hz e têm um baud rate igual. Poderemos definir baud rate como a variação que o sinal (1800Hz) sofre na linha, e com o mesmo baud rate podemos transmitir a velocidades diferentes, pois cada variação no sinal corresponde à transmissão do número de bits que serviram para modular o sinal. Neste caso para podermos transmitir a 9600 em modulação TCM; como temos 4 bits (mais um redundante) para modular o sinal precisamos de um baud rate de

2400, ou seja, 2400 variações por segundo, este valor é o valor máximo que uma linha banda de canal suporta. Por isso, se queremos também transmitir a 14400 precisamos de 6 bits (mais um redundante) para modular o sinal.

Como o caro leitor se poderá aperceber, existem normas a que os modems têm de obedecer, porque se cada fabricante fizesse o seu modem sem qualquer tipo de normas, quando quiséssemos, por exemplo, fazer uma ligação para uma base de dados estrangeira por modem, tínhamos de ter um modem igual ao do outro lado. Assim, para resolver este tipo de problemas, existe um comité internacional chamado CCITT, que definiu normas conjuntamente com outras organizações internacionais de comunicação de dados, para os fabricantes de modems. (ver quadro anexo).

Desta forma os modems banda de canal podem conectar-se com diferentes marcas de modems graças às normas criadas, o mesmo não acontecendo com os modems bandas de base, sendo necessário que estes sejam da mesma marca e iguais.

As normas para os modems banda de canal são bastante completas, por isso só nos limitaremos a falar nos seus aspectos mais importantes, ou seja, modulação, velocidade, número de fios utilizados, tipo de comunicação (assíncrono ou síncrono) e se a comunicação é full-duplex ou half-duplex (ver quadro anexo).

A recomendação V21 em Portugal é ainda utilizada para aceder à rede Telepac via telefone a uma velocidade de 300 bps, assim como a recomendação V22, mas esta com uma velocidade de 1200 bps. De salientar ainda que um modem V22bis em 1200 bps é compatível com a recomendação V22, ou seja, um modem V22bis emula a norma V22. A recomendação V23 apesar de ser half-duplex em 600 e 1200 bps, é full-duplex em 1200/75 bps, sendo usado por isso para aceder ao videotexto. Temos, agora, as recomendações V26 e V26bis que, do mesmo modo, são compatíveis em 2400 bps, mas não são compatíveis com a V22bis. Foi uma norma bastante utilizada há uns anos atrás, estando ainda bastantes modems V26bis em

funcionamento nos centros regionais de segurança social de Évora e Beja, assim como em algumas entidades militares. Todos a funcionar em linhas banda de canal alugadas. Os modems V29 são bastante utilizados em linhas alugadas, tanto a 4 fios como a 2 fios, permitindo fazer multiponto, ou seja, temos por exemplo um modem master no lado do mainframe, e vamos ter vários modems colocados em diversos lugares em comunicação com o master. É necessário para este caso pedir uma linha multiponto aos CTT especificando os sítios onde os modems vão ficar: o master e os diversos nós.

Gostaria ainda de encorajar todos aqueles que gostariam e sentem a necessidade de comunicar a 9600bps (2 ou 4 fios) e mesmo a 14400bps (4 fios), pois com os modems que usem as normas V32 e V33 isso é possível – peçam uma demonstração à firma que contactarem. É de salientar que tanto a norma V32 como a V33 permitem trabalhar com velocidades mais baixas, no caso de as linhas não terem qualidade suficiente para trabalhar a alta velocidade. Ter em atenção ainda o tipo de comunicação em causa, síncrona ou assíncrona (o caso dos nossos PCs, XTs e ATs).

Ao concluir, queria ainda falar um pouco sobre o protocolo corrector de erros, MNP (“Microcom Network Protocol”). Este protocolo é aplicado sobretudo nos modems V22bis, V32 e V33, garantindo as nossas comunicações sem erros. O MNP tem vindo a desenvolver-se e neste momento existem 9 níveis que correspondem cada um a um protocolo diferente embora retrocompatível. O MNP 5 é o mais utilizado, fazendo também compressão de dados (assim como todos os níveis MNP superiores), no entanto, alguns modems V32 e V33 já possuem o MNP 9. Convém aqui salientar que, sendo utilizada a compressão de dados, o modem só vai funcionar com ficheiros de texto puro. Em contrapartida, devido à não existência de erros, e à compressão de dados, a comunicação processa-se de um modo mais lento.

 Pedro Ferreira

“
A
recomendação
V21 em
Portugal, é
ainda utilizada
para aceder à
rede telepac
via telefone, a
uma
velocidade
300 bps
”

Série	Modulação	Assin.	Sinc.	Half-d	Full-d	Nº fios	Veloc.
V21	FSK	Sim	Não	Não	Sim	2	300
V22	DPSK	Sim	Sim	Não	Sim	2	1200/600
V22 bis	QAM	Sim	Sim	Não	Sim	2	2400/1200
V23 1200-75	FSK	Sim	Sim	Sim	Sim	2 e 4	1200/600
V26	DPSK	Não	Sim	Não	Sim	4	2400
V26 bis	DPSK	Não	Sim	Sim	Sim	2 e 4	2400/1200
V26 ter	DPSK	Sim	Sim	Não	Sim	2	2400/1200
V27	DPSK	Não	Sim	Sim	Sim	2 e 4	4800
V27 bis	DPSK	Não	Sim	Sim	Sim	2 e 4	4800/2400
V27 ter	DPSK	Não	Sim	Sim	Sim	4	4800/2400
V29 4800	QAM	Não	Sim	Sim	Sim	2 e 4	9600/7200
V32 7200/2400	TCM	Sim	Sim	Não	Sim	2	9600/4800
V33 9600	TCM	Sim	Sim	Não	Sim	4	14400/12000

VAGUEANDO PELO INFORMIX SQL

O Informix tem vindo, desde há algum tempo, a assumir uma parcela importante no mercado das bases de dados. Arrisco-me até a afirmar que é das bases de dados mais utilizadas em Portugal desde que estas se afirmaram no nosso mercado informático.

O INFORMIX "ANTIGO"

O Informix tomou a sua posição no mercado quando surgiram as suas primeiras versões. Estas permitiam a criação das bases de dados e a sua posterior compilação, o que as tornava de bastante rápido acesso. Assim, podíamos definir os campos de uma base de dados, desenhar um ecrã e obter rapidamente um Form* que nos permitia obter Queries* sobre a base de dados, simples de utilizar e podendo definir-se qualquer condição de busca sobre o Form.

A utilização dos Reports* obtinha-se pela especificação, num ficheiro, do formato do relatório que se pretendia obter, que incluía definição do tamanho das páginas, formatação dos campos, limites de impressão, etc.

Todas estas facilidades podiam também ser obtidas através da chamada a um menu integrado denominado Informix.

Também o seu interface com a linguagem "C" era de bastante fácil utilização (o que pode também ter sido uma causa do rápido crescimento do Informix), pela utilização de algumas funções incluídas numa livraria que acompanhava o Informix. Os campos podiam ser lidos através de qualquer função standard (ou não) do "C", introduzidos dentro de uma estrutura e, pela utilização das funções do Informix, introduzidos, alterados ou apagados (dbbad, dbupdate, dbdelete, respectivamente), ou buscados (dbfind). Os programas desenvolvidos obtinham, assim, rápidas performances.

O INFORMIX "MODERNO"

O Informix SQL é, no fundo, um interface SQL sobre o Informix "antigo". Este é de funcionamento bastante agradável, uma vez que a sua utilização é guiada por menus. No seu menu principal podemos encontrar as opções Form, Report, Query-Language, User-Menu, Database e Table.

Vejamos, então, pormenorizadamente o que se pode obter com cada uma delas:

I) FORM

Através de um processador de texto comum podemos criar um ficheiro de extensão .PER no qual podemos introduzir as instruções que definirão o Form. Estas instruções estão agrupadas por conjuntos, definindo, assim, diversas áreas de controlo do Form.

O primeiro conjunto de instruções, apenas uma, permite identificar a base de dados com que vamos trabalhar (ver fig. 1 – Parte I).

O segundo conjunto define o formato do ecrã, onde poderemos definir as prompts dos campos e o seu tamanho (ver fig. 1 – Parte II).

O conjunto seguinte identifica as tabelas da base de dados com que vamos trabalhar (ver fig. 1 – Parte III)

O 4.º e último conjunto define os atributos dos campos referenciados na fig. 1 – Parte II (ver fig. 1 – Parte IV).

```

database      spooler      Parte I
screen                               Parte II

Numero do Assinante [f000]
Nome               [f001]
Data de Nascimento [f002]
Morada             [f003]
Codigo Postal     [f004] [f005]
Profissao         [f006]

end
tables
assinantes      Parte III
attributes      Parte IV
f000 = assinantes.numero;
f001 = assinantes.nome;
f002 = assinantes.dt_nasc;
f003 = assinantes.morada;
f004 = assinantes.cod_postal;
f005 = assinantes.local;
end

```

Fig. 1 — O formato de um Form

Um Form pode ser obtido ou desenhado através da selecção da opção do menu referido, ou compilado e/ou executado directamente da prompt do sistema.

No primeiro caso podemos utilizar uma ferramenta para gerar um formato do Form por defeito, o qual, através de um editor de texto, pode ser facilmente alterado para as especificidades do nosso ecrã (o ecrã gerado por defeito é do tipo do ecrã mostrado na fig.1).

No último caso, deveremos criar um ficheiro de formato ASCII (de extensão .PER), no qual introduziremos as instruções manualmente, utilizando posteriormente os comandos Sformbl e Sperform para compilar o Form, e para executá-lo, respectivamente.

II) REPORT

No que se refere aos Reports, estes são, também, de relativa facilidade de execução. Assim como nos Forms, podemos obter também um formato do Relatório por defeito. No entanto, o utilizador já algo experimentado não se vai limitar a ele.

A estrutura de um Relatório pode ser definida como um conjunto de secções (seis):

Assim como nos Forms, a primeira identifica a base de dados com que vamos trabalhar (fig. 2 – Parte I, Database Section);

A segunda utiliza o comando Select da SQL para definir as colunas e as linhas a imprimir (fig. 2 – Parte II, Select Section),

A terceira indica o formato a utilizar para o desenho do Relatório (na fig. 2, Format Section, a instrução "format every row" significa utilizar o formato por defeito);

```

database spooler end      Parte I
select                    Parte II
    numero,
    nome,
    dt_nasc,
    profissao,
    morada,
    cod_postal,
    local
from assinantes end
format every row end     Parte III
  
```

Fig. 2 — O formato de um Report

As três secções que não foram mencionadas incluem as definições de variáveis (Define Section, para definir variáveis acumuladoras, por exemplo), a secção de entrada de dados (Input Section, que permite ao utilizador definir intervalos de impressão, por exemplo) e a secção de formato da saída de dados (Output Section).

Assim como nos Forms, os Reports podem ser obtidos através da opção Report, ou directamente compilados e executados através da prompt do sistema, utilizando os comandos Saceprep e Sacego.

III) QUERY LANGUAGE

Com esta opção do menu poderemos interrogar a base de dados através de uma linguagem denominada RDSQL (um dialecto de SQL), introduzindo as suas instruções num ficheiro ASCII qualquer, através de um editor de texto.

Este dialecto de SQL possui todos os conjuntos de instruções que definem a DDL (Data Definition Language), a DML (Data Management Language) e a DCL (Data Control Language).

IV)

As últimas opções do menu (User-Menu, Database e tabe) permitem integrar os nossos Forms e Reports numa só aplicação, através da definição de um menu, obter informações e seleccionar as diversas bases de dados existentes, bem como criar novas, alterá-las e/ou apagá-las; e obter informações sobre as diversas tabelas existentes e criar, alterar ou anular outras já existentes.

O 4GL

O 4GL, Rapid Development System, como é conhecido, permite desenvolver um programa através de uma interface orientado por menus, numa linguagem própria do sistema, que não passa da linguagem SQL extendida a outras funções inexistentes no standard, o que a torna ainda mais potente.

```

SELECT [ALL|DISTINCT|UNIQUE] lista_selectiva
[INTO lista_variaveis]
FROM nome_tabela [alias_tabela]
[OUTER nome_tabela [alias_tabela]
[WHERE condicao]
[GROUP BY lista_colunas]
[HAVING condicao]
[ORDER BY nome_coluna [ASC | DESC][...]]
[INTO TEMP nome_tabela]
  
```

Fig. 3 – A instrução Select do 4GL

Como se pode ver pela fig. 3, a instrução Select permite obter diversos outputs agrupados. estes poderão também ser direccionados para uma tabela temporária, a criar no momento. É também possível aplicar funções de agregamento sobre as linhas, extraindo, assim outros dados não explícitos na base de dados.

Esta linguagem possui também processos de indexação e de *clustering* para que seja possível obter acessos ainda mais rápidos às tabelas.

As funções de segurança também não foram esquecidas, sendo aqui usadas as seguintes instruções SQL: Rollback, Commit Work, Grant e Revoke.

Por outro lado, a passagem da aplicação protótipo, desenvolvida em 4GL, para uma aplicação utilizando SQL embebido em linguagem "C" torna-se extremamente fácil, obtendo-se assim um programa verdadeiramente executável.

CONCLUSÕES

Excluindo o 4GL, as ferramentas mais poderosas do Informix são, sem dúvida alguma, os seus Forms. Estes permitem definir uma série de atributos sobre os campos, tais como definição de campos Display-Only (campos cujo conteúdo não pode ser alterado), a definição dos formatos dos mesmos, os atributos de cor, as relações existentes entre as diferentes tabelas (Joins), o despoletar de acções antes ou posteriormente ao preenchimento dos campos (Triggers) através da indicação de cláusulas, tais como Before e/ou After, etc.

Escusado será dizer que o 4GL constitui, sem dúvida alguma, uma ferramenta potentíssima no desenvolvimento de software. É necessária, no entanto, uma longa fase de adaptação ao sistema, para que seja possível extrair-lhe o máximo das suas capacidades.

Uma outra vantagem da utilização deste SGBC é o facto de funcionar numa enormíssima gama de equipamentos, que vão desde o mais simples equipamento utilizando o sistema operativo DOS, até a sistemas muito mais complexos, tais como OS/2, Unix, VAX/VMS, MVS e outros. Como é evidente, isto permite uma grande portabilidade das aplicações desenvolvidas.

Um dos únicos senãos desta base de dados (não só desta, como de muitas outras) é o facto de que uma aplicação desenvolvida sobre ela necessita de um Runtime que é cobrado a preço de ouro, o que de uma maneira ou de outra, se acaba por reflectir no preço final do produto desenvolvido.

Resta ainda referir que o Informix é representado em Portugal por APIE – Porto.

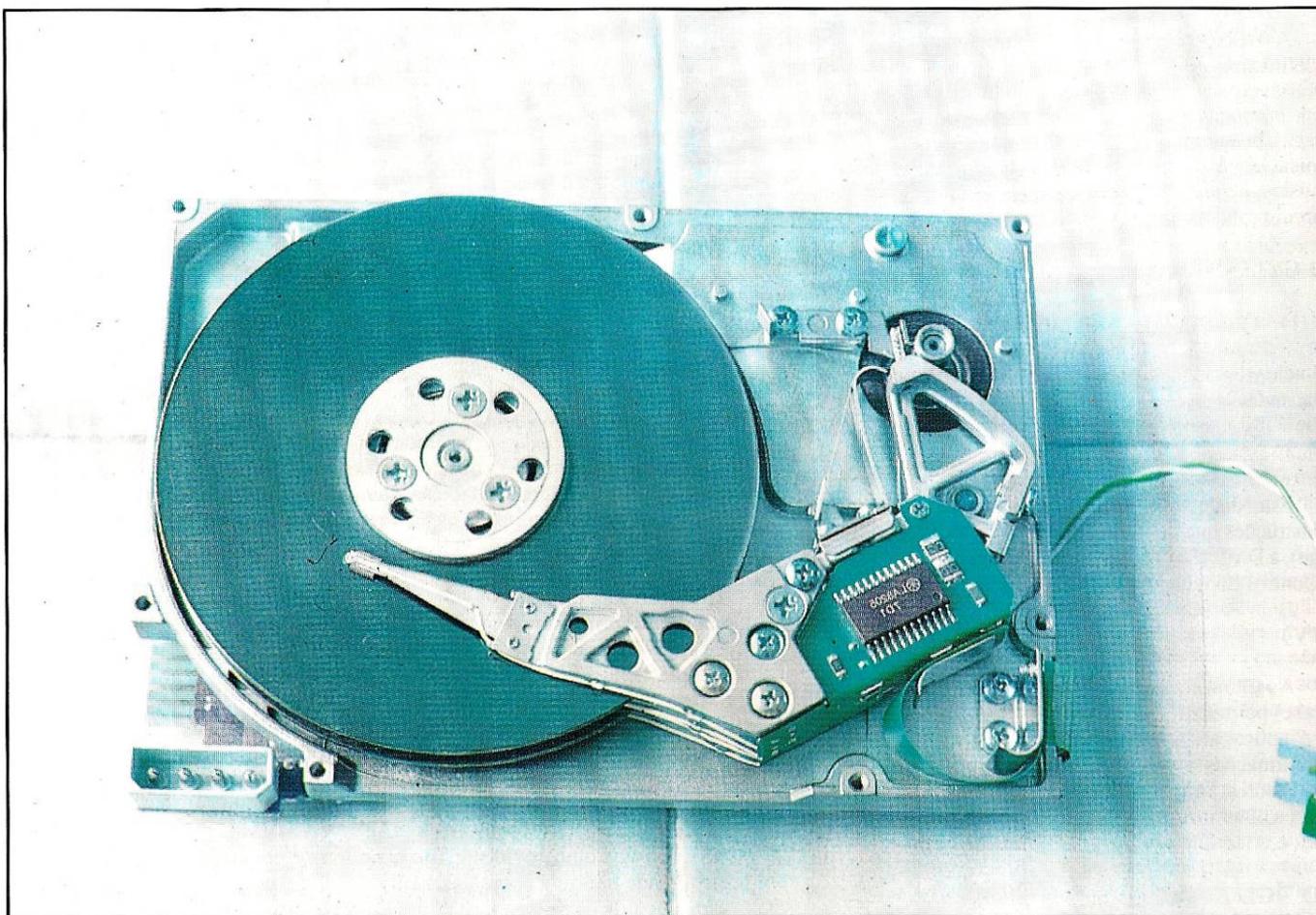
Paulo Pinheiro

CONJUNTOS COMERCIALIZADOS

(Os preços variam consoante o sistema operativo onde são aplicados os *packages*).

Informix – Informix SQL Relational Database Management System – Informix ESQ-C, Embedded SQL and Tools for "C" – Informix ESQ-COBOL, Embedded SQL for COBOL – Informix ESQ-Ada, Embedded SQL for ADA – Informix TURBO Enhanced Database Server – C-ISAM File Access Method – Informix Datasheet Add-In – Informix 4GL – Informix 4GL Fourth-Generation Application Development Language – Informix 4GL Interactive Debugger – Informix 4GL Rapid Development System – Informix 4GL, Rapid Development System and Interactive Debugger

O DISCO POR DENTRO E POR FORA



Quem é que não ouviu já falar de disquetes ou discos rígidos? Provavelmente toda a gente que já ouviu qualquer coisa sobre informática ou está, de qualquer forma, ligado a ela. A utilidade dos mesmos por certo que também não deixa dúvidas a ninguém. No entanto, uma pode ainda existir (e por certo existe) em algumas (ou muitas) pessoas sobre a constituição de uma disquete (ou disco rígido), como é feita, como são guardados e lidos os dados, como saber onde os guardar ou onde os ir ler. Pois é para isso que a **Spooler** existe: para informar e tirar todas (quaisquer que sejam) as dúvidas dos nossos leitores.

Existem dois aspectos numa disquete: o aspecto físico, relacionado com o hardware e com a forma como a disquete é organizada para os dados poderem ser guar-

dados e, mais tarde, lidos, e o aspecto de software, ou seja, qual a estrutura necessária para que se possam criar, escrever, ler, alterar e apagar ficheiros de uma disquete ou disco rígido.

Estes dois aspectos estão, como é óbvio, relacionados; no entanto, serão tratados separadamente, isto é, neste número da **Spooler** falaremos sobre o primeiro aspecto (hardware) e no próximo número falaremos sobre a perspectiva de software.

O HARDWARE

Podem parecer uma coisa muito complicada mas, na realidade, até é bastante simples a constituição de um

disco e o seu funcionamento. Funciona da mesma maneira que os vulgares gravadores de cassetes, ou seja, armazena a informação magneticamente. A gravação é feita passando uma cabeça de gravação sobre o material magnético do disco que, assim, fica magnetizado e com a informação guardada. Mas como consegue a cabeça magnetizar a fita (ou o disco)? A resposta é simples. Através de uma corrente eléctrica que passará através da cabeça e que irá polarizar as partículas magnéticas existentes na fita dependendo do sentido da corrente. A leitura passa-se ao contrário – as partículas magnéticas irão provocar uma corrente eléctrica na cabeça com sentido dependente da polarização de cada partícula eléctrica.

Nas disquetes, no entanto, não é necessário um disco inteiro para guardar uma pista de informação, pelo que num disco irão existir várias pistas de gravação a que foi dado o nome técnico de *tracks*.

A TRACK

Fisicamente uma track é uma zona circular do disco e o seu número irá depender da quantidade de informação que o disco pode armazenar por unidade de área e por a precisão do dispositivo físico que faz movimentar a cabeça de gravação (e leitura) entre as várias tracks existentes. Para melhor visualizar o que fisicamente uma track é pode imaginar qual seria o resultado que obteria ao poisar um dedo cheio de tinta sobre um disco vulgar (de música) a rodar em cima do prato. É óbvio que obteria um círculo; pois isso é uma track.

O CYLINDER

Tudo o que temos vindo a falar passa-se somente sobre um lado do disco, mas vamos lá pensar um bocadinho: quantas faces tem um disco? Obviamente que duas. E por que não aproveitá-las a ambas? Pois é isso exactamente que é feito: são feitas gravações em ambos os lados dos discos. Isto é feito aproveitando somente um dispositivo (geralmente um step-motor) para fazer deslocar as duas cabeças entre as várias tracks existentes em ambas as faces (uma cabeça de gravação por face), para isso é utilizada geralmente uma técnica em que as cabeças estão seguras por uma espécie de tesoura rodando o disco no meio. Portanto, sempre que uma cabeça é deslocada, a outra também o é, ficando sobre a mesma track que a outra, mas na face oposta; a estas duas tracks é dado o nome de *cylinder*. Um cylinder é, pois, constituído pelas duas tracks que estão uma sobre a outra em ambas as faces.

O SECTOR

A quantidade de informação que uma track pode suportar depende da qualidade do disco, assim como da codificação com que os dados são gravados; no caso da codificação conhecida por MFM poder-se-ão guardar entre 8 e 12 kilobytes (nos discos rígidos) com segurança. Ora os computadores não estão preparados (ou não estavam quando tudo isto foi criado) para ler uma track de uma só vez pois a quantidade de informação seria demasiada para os controladores de disco. Foi, pois, necessário criar uma unidade ainda menor que a track (um submúltiplo); a esta unidade foi dado o nome de *sector*.

Apesar de os controladores de disco suportarem várias dimensões do sector (128, 256, 512, 1024), a dimensão mais utilizada desde a criação do PC-DOS 1.0 é a de 512bytes. Devido ao facto de a dimensão do sector ter sido standardizada, o número de sectores contido em cada track irá variar dependendo da capacidade de armazenamento de cada track. Essa quantidade irá depender da densidade da disquete ou do disco. na disquete de 360 kilobytes (as mais comuns) cada track contém 9 sectores:

no entanto, poderia possuir até 10 sectores, conservando a mesma fiabilidade na leitura dos dados.

O HARD DISK

Não deve ser segredo para ninguém que é bem melhor possuir um disco rígido a 100 disquetes. Mas porquê? Além de ler os dados muito mais rapidamente, possui muito maior capacidade de armazenamento de dados, o que é bastante importante para a comodidade do utilizador e funcionamento de alguns programas. Mas qual a razão da maior velocidade de armazenamento e leitura dos dados? Isso deve-se à velocidade de rotação do disco (3600 rotações por minuto), que é muito maior que a de uma disquete. Mas como é isso possível? É que nos discos rígidos a cabeça de gravação e leitura não toca no disco, ao contrário do que acontece nas disquetes, em que a cabeça de gravação assenta sobre a própria disquete, gerando um grande atrito abrasivo que, se fosse maior, estragaria depressa as cabeças.

Quanto mais perto a cabeça estiver do disco, mais forte é a corrente eléctrica provocada pelas partículas magnéticas (no caso da leitura), assim a cabeça tem de ficar o mais próximo possível para uma melhor e mais rápida leitura dos dados. A solução que arranjaram para a cabeça ficar em equilíbrio perto da superfície foi muito engenhosa: entre a cabeça e o disco irá existir uma pequena camada de gás que a fará flutuar, pois a cabeça além de estar suspensa tem também umas pequenas "asas" que farão com que fique num equilíbrio aerodinâmico. A cabeça irá, assim, comportar-se como um pequeno planador, que quando tiver tendência para baixar será gerada uma força aerodinâmica para cima que a fará ficar no mesmo lugar; se tiver tendência para subir, o mesmo se passará mas no sentido oposto. Ficará tão próxima do disco que um simples cabelo não conseguiria passar por baixo.

Outra diferença que pode existir entre os discos rígidos e as disquetes é o número de faces. Numa disquete só podem existir duas faces; no entanto, num disco isso não se passa e podem existir mais faces por disco. É claro que por cada face irá existir uma cabeça; poderão, portanto, existir mais que duas cabeças.

UMA NOVA LINGUAGEM

Falemos agora de alguma da linguagem nova que apareceu com os discos rígidos mas que pode ser aplicada a ambos (discos rígidos e disquetes).

Seek time – tempo que cada cabeça leva de qualquer track onde esteja para uma outra track qualquer.

Track-track seek time – tempo que a cabeça demora de uma track para a track adjacente. Geralmente 8-10 mili-segundos.

Tempo de acesso – Tempo que a cabeça demora do sítio onde estiver até ao sítio onde estiverem os dados que se pretendem ler. Obviamente que o tempo de acesso irá depender da organização dos dados no disco, se estiverem muito espalhados este tempo irá ser muito grande, pois a cabeça terá de percorrer "grandes distâncias"; caso contrário, poderá ser pequeno.

Tempo de acesso médio – É óbvio que este tempo é a média do tempo anterior, ou seja, em média o tempo que a cabeça leva do sítio actual para o sítio onde estão os dados. Como o anterior, irá depender fortemente da organização dos dados.

Average Latency – Metade do tempo necessário para o disco dar uma volta. Como os discos rodam a uma velocidade de 3600 rpm., uma volta demorará, aproximadamente, 16,67 mili-segundos, pelo que o "average latency" será de, sensivelmente, 8,3 mili-segundos.

Transfer Rate – É o tempo que o disco leva a ler os

“
Quanto mais
perto a cabeça
estiver do
disco, mais
forte é a
corrente
eléctrica
provocada
pelas
partículas
magnéticas
”

“
**Em cada
 sector
 escreve-se o
 seu endereço,
 ou seja, o seu
 número**
 ”

dados do disco e a transferi-los para o computador. Irá depender da velocidade de rotação do disco (não se pode ler um byte enquanto ele não estiver debaixo da cabeça) assim como da codificação com que os dados estão gravados (quanto maior a densidade, maior a velocidade). A maioria dos hard disks possui um rate de 5 Megabits por segundo, e está em via de dar um salto com o aparecimento da interface de ESDI (Enhanced Small Device Interface) que permite transferências da ordem dos 10 e 15 Megabits por segundo.

O ENDEREÇO

Vamos agora falar da localização dos sectores dentro da track. Devido ao facto de fisicamente a track ser um círculo, à primeira vista não existe maneira de se diferenciar um sector de outro, pois a disquete é redonda e não tem pontos de referência a que nos possamos “agarrar”. Como foi este problema resolvido? No início, a solução que foi encontrada foi fazer tantos buracos na disquete como sectores por track, isto é, além do buraco central iriam existir, à volta, pequenos buracos que seriam detectados por hardware e que indicavam ser aí o início de certo sector. Este método era chamado de “hard sectoring”. No entanto esta técnica, além de ser muito “deselegante”, conduzia a baixas performances por parte da disquete, pelo que teve de se optar por outro método.

Actualmente é usada uma técnica muito mais elegante: os sectores são simplesmente marcados por software, ou seja, em cada sector é escrita a sua identificação.

Com mais pormenor, a situação é a seguinte: em cada sector escreve-se o seu endereço, ou seja, o seu número – é o chamado preâmbulo, pois está imediatamente antes da zona de dados do sector. No preâmbulo antes do endereço do sector são escritos uns quantos bytes de sincronização para o controlador de disco saber que está prestes a ler o endereço de um sector. Por fim são ainda escritos uns quantos bytes entre cada sector para forçar a existência de um certo intervalo de tempo entre a cabeça de leitura a passar de um sector para outro, isto para que o controlador de disco não se “atrapalhar” – estes são os chamados “gap bytes”.

RESUMO

Vamos lá resumir tudo o que foi dito. Uma disquete é constituída por cylinders, cada cylinder tem duas tracks, cada track pode ser constituída por vários sectores, os sectores estão separados pelos “gap bytes”, cada sector é constituído pelo preâmbulo e pelos dados; o preâmbulo é constituído pelos bytes de sincronização e pelo endereço do sector; a zona de dados do sector contém 512 bytes (geralmente).

A FORMATAÇÃO

Tudo o que temos vindo a falar – cylinders, sectores “gap bytes”, etc. é definido aquando da formatação hard; outras acções serão tomadas aquando da formatação soft. Estes dois tipos de formatação só se tornam evidentes quando estamos a tratar com discos rígidos; quando se trata com disquetes não se notam. E porquê? É que cada vez que formatamos uma disquete, ambos os tipos de formatação são realizados, isto é, primeiro é feita uma formatação hard e, logo de seguida, uma formatação soft, não se notando, portanto, a existência dos dois tipos em separado. Com os discos rígidos isso já não acontece; quando se executa o comando “Format” sobre um disco rígido, somente a formatação soft é realizada. Para executar a formatação hard num disco rígido é necessário um programa especial ou, então, recorrer à ROM BIOS.

Na ROM BIOS existe um programa no endereço

C800:5 que executa a dita formatação. Como pode ser executado? Fazendo correr o programa “Debug” e, em seguida, dar o comando “G=C800:5”. Em seguida são feitas perguntas a que o utilizador experimentado tem obrigação de saber responder.

Apesar de explicarmos os procedimentos necessários para executar esta formatação, **não aconselhamos ninguém a fazê-la, pois perde-se todos os programas em disco.** Em seguida seria necessário criar partições para o disco, mas isso será explicado no próximo número da Spooler.

O INTERLEAVE

Quando um disco rígido é formatado a baixo nível (“hard format”) geralmente o programa de formatação faz uma pergunta, “qual o Interleave?”. O que é isso de interleave? A principal função de uma formatação de baixo nível é escrever em cada sector o seu preâmbulo, ou seja, o seu endereço. Quando se falou da existência de sectores numa track, nada se disse, mas quem nada sabia com certeza que ficou a pensar que os sectores seriam numerados sequencialmente. Realmente, assim pode acontecer (e muitas vezes acontece), mas nada diz que assim tem de ser e, na verdade, até existem boas razões para assim não ser. Porquê? Se nós possuíssemos o computador perfeito, a numeração poderia ser sequencial, pois a performance do computador seria a mesma do disco e teria velocidade suficiente para ler um sector, guardá-lo em memória, e voltar ao controlador a tempo de apanhar a cabeça de leitura do disco em cima do sector seguinte. No entanto, geralmente a situação atrás descrita é rara acontecer e, quando o computador devolve o controlo ao controlador para ler o sector seguinte, ele (o físico) já passou, tendo assim de esperar que o disco dê uma volta quase inteira para ter, de novo, acesso ao sector pretendido. A solução será, pois, numerar os sectores de maneira que a sua numeração não coincida com a sequência física para que o computador tenha tempo de tratar os dados de um sector e, quando for ler o seguinte, não tenha de esperar por uma volta quase inteira do disco, mas “esteja logo ali à mão”. Irá, assim, depender da velocidade do disco, do computador e do controlador qual o interleave que melhor performance dará a todo o processo de leitura ou escrita do disco.

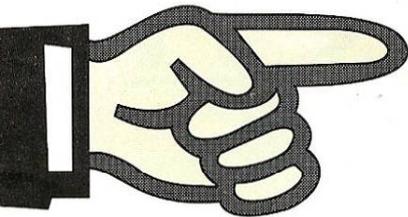
Vejamus um disco cuja track contenha 17 sectores. Se o interleave fosse de 1, então para ler uma track toda seriam necessárias 17 voltas do disco, pois por cada sector lido teria de esperar quase uma volta para ler o seguinte. No entanto, se fizermos a numeração dos sectores do seguinte modo, somente iremos precisar de duas voltas para ler uma track inteira (isto é somente um exemplo): andando à volta da track escreveríamos o sector número 1, depois deixava-se passar um e numerava-se o seguinte com o número 2, e assim sucessivamente até que o sector 10 ficaria entre os sectores números 1 e 2, e, assim sucessivamente – uma sucessão de sucess... Ficaríamos, pois, com um interleave de 2 em vez de 1, e a performance do disco melhoraria consideravelmente.

Por agora tudo foi dito sobre o hardware de uma disquete (e disco rígido). No próximo número da Spooler será apresentada a continuação deste artigo onde será falado do software envolvido numa disquete, organização dos dados, informação necessária e complementar existente..., enfim, tudo o que há para dizer (e ainda não foi) sobre disquetes e discos será dito no próximo número da Spooler. Não deixe de ler.

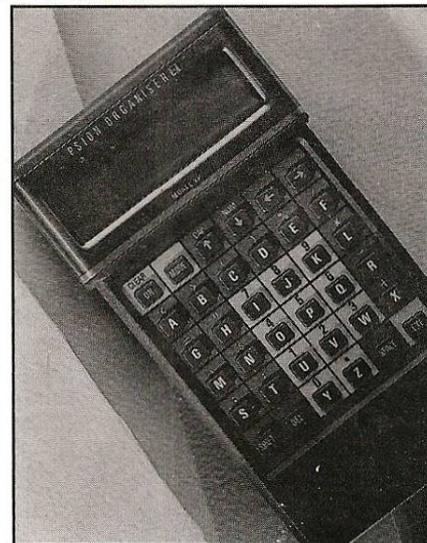
No próximo mês “veremos” de novo.

Vítor Guerreiro

GANHE UM



PSION



Ganhe um quê?!

Ganhe um Psion Organiser II – Modelo XP (P) no valor de 34.281\$00 (IVA incluído).

E isso, o que é?

Santa ignorância! O Psion Organiser é um espantoso e pequeno computador de bolso que pode reter toda a sua informação pessoal de carácter essencial – tal como nomes, endereços, números de telefone, despesas, etc. –, é também calculadora, programável com 10 memórias permanentes e mais de 50 funções matemáticas e científicas.

Serve de agenda e pode usar 8 alarmes horários separados, para soarem a intervalos de horas, dias ou semanas.

No Psion tem à sua disposição a linguagem de programação «OPL», que lhe permitirá criar os seus próprios programas.

Além do mais, você pode adicionar ao seu Organiser II «Packs» de memória e de software pré-programado.

Com o auxílio do «Comms Link» (elo de comunicações), o Psion pode ser ligado ao

seu PC e transferir dados nos dois sentidos com grande facilidade.

Isto para não falar na possibilidade de ligação via Modem, de adaptação de impressora, leitor de código de barras, leitor de cartões magnéticos, etc., etc..

Que devo fazer para possuir esse fabuloso aparelho?

Se já é assinante da **Spooler** basta aguardar que a sorte lhe sorria. Com efeito, todos os meses, e a começar já em Março, vamos sortear entre **todos** os assinantes um **PSION ORGANISER II – Modelo XP (P)**. Se não é assinante, tem duas alternativas: torna-se assinante ou consulta a **Montra Spooler**.

PSION ORGANISER II

O computador de bolso concebido para milhares de aplicações com uma memória de massa até 320 K, permite comunicar com outros sistemas de computadores, impressoras e modems. É, também, uma agenda telefónica, diário, bloco de notas e muito mais. Tudo isto na palma da sua mão.

ESTA INICIATIVA É PATROCINADA POR:



VER ANÚNCIO NAS PÁGINAS DESTA REVISTA

Assinatura conjunta com a CATS-BBS — 12 números da Spooler+12 meses de utilização total da CATS (artigo sobre a CATS/BBS na Spooler Nº 4).
Nesta modalidade poupa 2 520\$00 no total das duas assinaturas

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. Alfredo Roque Gameiro, 21 - 1º Dtº
1600 LISBOA

Spooler nº 8

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

LOCALIDADE CÓDIGO POSTAL

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

MODALIDADE COM DISQ. DE 5 1/4" 4 000\$00 ASSIN. ANUAL SPOOLER(5 1/4")+CATS-BBS 10 500\$00

MODALIDADE COM DISQ. DE 3 1/2" 7 400\$00 ASSIN. ANUAL SPOOLER(3 1/2")+CATS-BBS 13 900\$00

PARA PAGAMENTO DA ASSINATURA ENVIO :

O CHEQUE Nº..... S/O BANCO

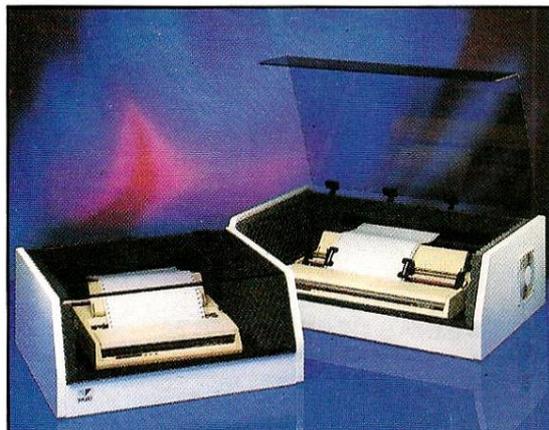
O VALE DE CORREIO Nº.....

* Os dados assinalados com um asterisco podem não ser preenchidos. Estas informações destinam-se a um maior conhecimento dos leitores por parte de todos os que fazem mensalmente esta publicação, a fim de tentar satisfazer melhor as suas necessidades como utilizadores de computadores.

montra Spooler

Para encomendar os produtos constantes deste catálogo, basta preencher o cupão publicado para o efeito, e remetê-lo para os nossos serviços. O leitor a partir deste número, NÃO deve enviar qualquer forma de pagamento. Todos os produtos serão enviados à cobrança e, aos preços indicados, serão acrescidos os portes e taxas de cobrança. Devido ao facto de alguns dos produtos aqui apresentados esgotarem rapidamente, pensamos ser este o melhor processo para evitar contratempos como seja encomendar produtos que já se encontram esgotados, etc. etc..

CAPOTAS INSONORIZANTES PARA IMPRESSORAS



Reduza até 95% o ruído irritante provocado pela sua impressora.

Ref.: H002 - 80 cols. Preço: 44 460\$00
Ref.: H003 - 132 cols. Preço: 52 650\$00

Representante: CONSULDATA

CAIXA ARQUIVADORA PARA DISQUETES VARIO 7



Ref.: H005
Preço: 2 457\$00

Representante:
CONSULDATA

Caixa arquivadora que permite várias combinações de disquetes de 5,25", 3,5" e data cartridges, com 2 chaves.

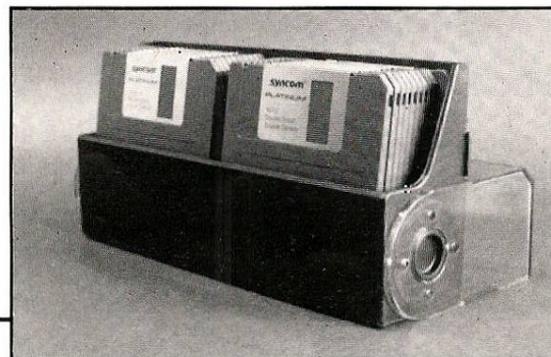
Combinações possíveis:

- 120 disquetes de 5,25"
- 60 disquetes de 5,25" + 4 Data cartridges
- 30 disquetes de 3,5" + 6 cartridges

ARQUIVADOR PARA 20 DISQUETES 3,5"

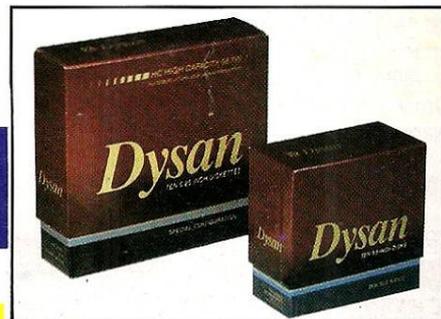
Ref.: H016
Preço: 1 654\$00

Representante:
C.A.T.S.



DISQUETES DYSAN

Representante: CONSULDATA



Caixas com 10 disquetes:

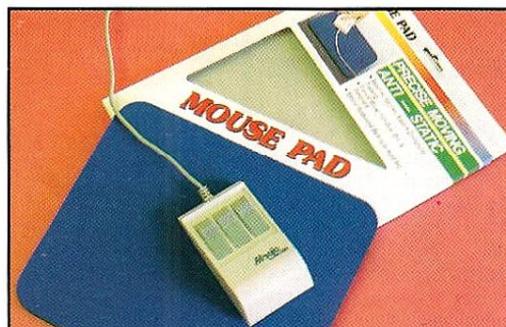
- 5,25" HD 1,6 Mb	Ref.: D001	Preço: 5 792\$00
- 5,25" DS/DD 48 TPI	Ref.: D002	Preço: 4 680\$00
- 3,5" 2D	Ref.: D003	Preço: 5 850\$00
- 3,5" HD 2Mb	Ref.: D004	Preço: 11 115\$00

TAPETE ANTI-ESTÁTICO PARA MOUSE

Trabalhe com mais precisão e conforto mantendo o seu mouse limpo.

Ref.: H011
Preço: 1 107\$00

Representante:
C.A.T.S.



DISQUETES PRECISION

Caixas com 10 disquetes:

- 5,25" HD 1,6 Mb	Ref.: D005	Preço: 3 276\$00
- 5,25" DS/DD 48 TPI	Ref.: D006	Preço: 1 521\$00
- 3,5" 2D	Ref.: D007	Preço: 3 510\$00

Representante: CONSULDATA



DISQUETES PLATINUM

Representante:
CONSULDATA



Caixas com 10 disquetes:

- 5,25" HD 1,6 Mb	Ref.: D008	Preço: 2 106\$00
- 5,25" DS/DD 48 TPI	Ref.: D009	Preço: 1 287\$00
- 3,5" 2D	Ref.: D010	Preço: 3 276\$00
- 3,5" HD 2Mb	Ref.: D011	Preço: 7 722\$00

CAIXAS ARQUIVADORAS PARA DISQUETES



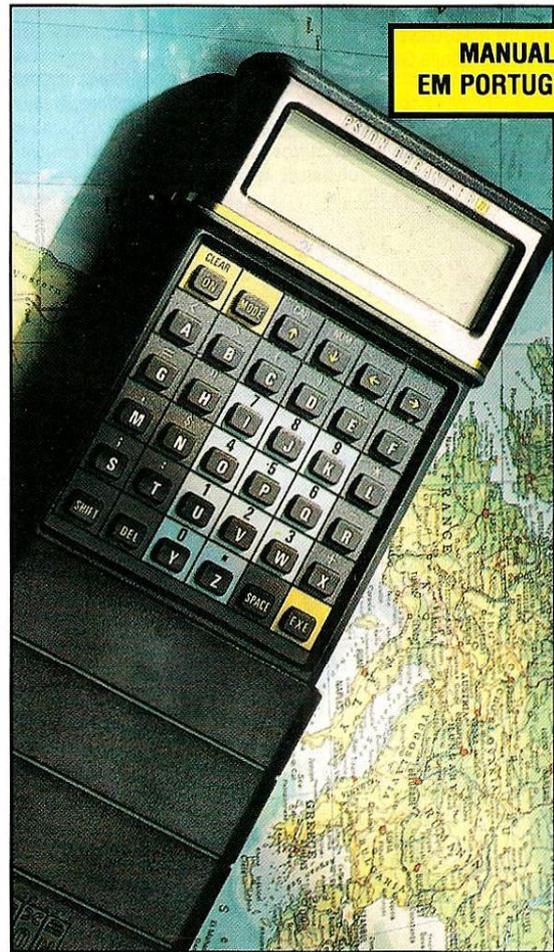
Todos os modelos com fechadura
e duas chaves

Representante: CONSULDATA

- Mod. DX 50A - 50 disq. 5,25"	Ref.: H006	Preço: 2 223\$00
- Mod. DX 100N - 100 disq. 5,25"	Ref.: H007	Preço: 2 340\$00
- Mod. 40 L - 40 disq. 3,5"	Ref.: H008	Preço: 1 463\$00
- Mod. SS50 - 50 disq. 3,5"	Ref.: H009	Preço: 2 223\$00
- Mod. 80 L - 80 disq. 3,5"	Ref.: H010	Preço: 2 340\$00

PSION ORGANISER II

MANUAL
EM PORTUGUÊS



MODELO XP (P)
Memória: 32 K ROM - 8 K RAM

UM PODEROSO COMPUTADOR NA PALMA DA SUA MÃO

Computador de bolso concebido para milhares de aplicações, com uma memória de massa até 320 K, permite comunicar com outros sistemas, impressoras e modems. É também Agenda Telefónica, Diário, Bloco de Notas e muito mais.

DATACOMP

Ref.: H050
Preço: 34 281\$00



Recorte ou fotocopie o cupão e remeta-o para:

SPOOLER MAGAZINE
Rua Alfredo Roque Gameiro, 21 - 1º Dtº
1600 LISBOA

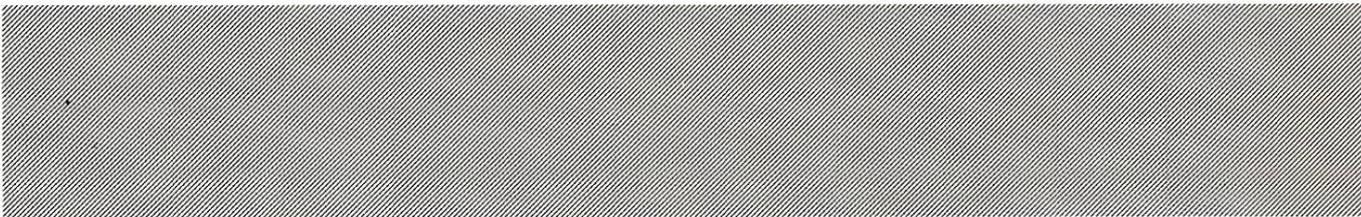
CUPÃO DE ENCOMENDA – MONTRA SPOOLER

Spooler nº 8

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

Quant.	Ref.	Designação	Preço	
TOTAL				

NOME
 MORADA
 C. POSTAL LOCALIDADE
 TELEFONE SOU ASSINANTE Nº



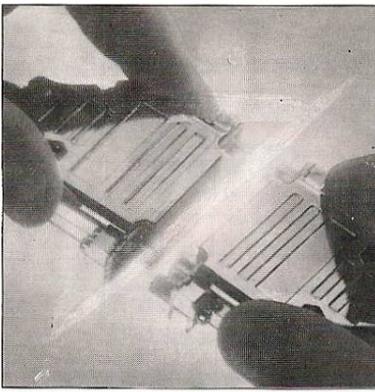
Computadores



Wuga

286 a 12 Mhz
 286 a 20 Mhz
 386 a 33 Mhz (32 Kb Cache)

Telf. 524027



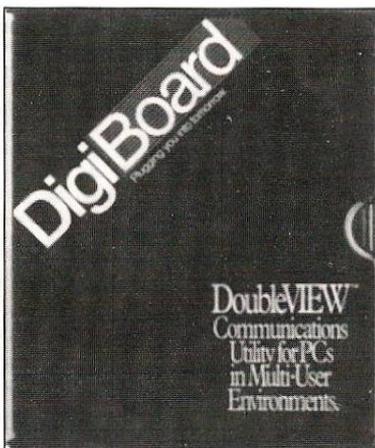
DigiBoard
Plugging you into tomorrow.

Placas Multivias para Sistemas operativos Multi User Xenix, Unix, C-DOS, PICK, PC-MOS 386, etc. Com 4, 8 ou 16 saídas. Inteligentes ou não. Para compatíveis incluindo os PS/2, EISA e para equipamentos com o processador 68000, como o Macintosh.

DigiChannel C/X™

Host Adapter Card and C/X Concentrator for the EISA Bus

- Sixteen, 32, 48 or 64 RS-232 asynchronous serial communication channels from one slot.
- Fastest multi-channel board available with data transmission rates of up to 38.4KB per channel depending upon configuration.
- High serial I/O performance at a low cost per user.



DoubleVIEW™

Communications Utility for PCs in Multi-User Environments

DoubleVIEW is a small but powerful memory resident communications utility which allows a PC user to access a multi-user system with his PC rather than a terminal. This state-of-the-art terminal emulation package allows instantaneous (hot key) toggling between a PC-based application and an application residing on a multi-user system, enabling users to switch between both operating systems without losing either screen or the data they contain. DoubleVIEW is a bridge between the operating systems, allowing the user to reap the benefits of both without ever leaving the comfort of a PC. The program is powerful, but due to its set-up screens, it is also easy to use and configure. In fact, DoubleVIEW is for anyone who depends on a PC for DOS applications (like spreadsheets, data-bases and word processors), but also needs to access the information on a multi-user host.

DigiCHANNEL PC/Xi for the PC-Standard

DigiCHANNEL PC/Xi

- Top-of-the-line solution
- 12.5 MHz 80186 processor (16 Mhz available)
- I/O mates with up to 16 channels
- RS-232
- 128K RAM standard (256 or 512K optional)
- Synchronous option

DigiCHANNEL COM/Xi

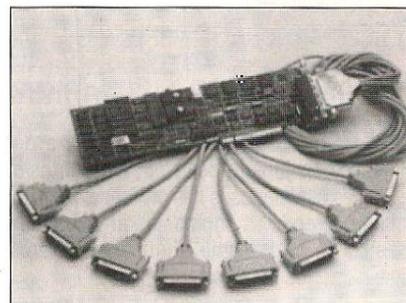
- Moderately priced, flexible solution
- 10 MHz 80188 processor
- Daughter cards supporting 4 or 8 channels
- RS-232 and RS-422
- 256K RAM

DigiCHANNEL PC/Xe

- Economical solution
- 8 MHz 80186 processor
- 4 or 8 channels
- RS-232
- 64K RAM

DigiCHANNEL PC/X

- Basic solution
- 4, 8 or 16 channels
- RS-232



DigiCHANNEL Series for the Micro Channel

DigiCHANNEL MC/Xi

- Full-featured solution
- 12.5 MHz 80186 processor
- I/O Mates with 4, 8 or 16 channels
- RS-232 and RS-422 options
- 256K RAM
- Synchronous option

DigiCHANNEL MC/X

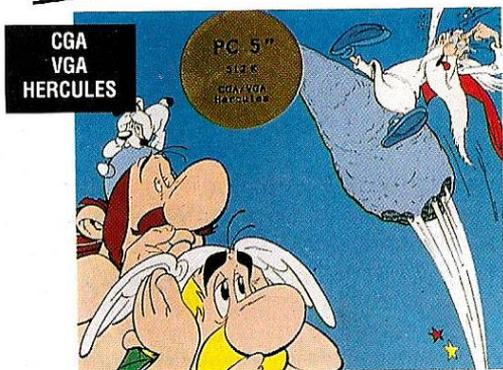
- Basic solution
- 4, 8 or 16 channels
- RS-232

RAMTOP

STOCK LIMITADO

JOGOS

ATENÇÃO: COM PLACA EGA, OS JOGOS SÓ REPRODUZEM 4 CORES. AS INSTRUÇÕES ORIGINAIS SÃO FORNECIDAS EM FRANCÊS. PARA QUEM O SOLICITAR, ENVIAREMOS, EM SEPARADO A RESPECTIVA TRADUÇÃO.



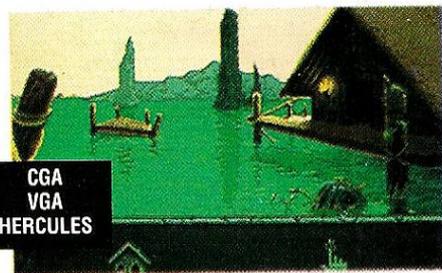
CGA
VGA
HERCULES

PC 5"
512 K
CGA VGA
HERCULES

ASTERIX E A PANCADA DO MENIR

O Panoramix levou com um menir na cabeça. É necessário ajudá-lo a lembrar-se da fórmula da poção mágica, procurando e misturando diversos ingredientes escondidos num bosque repleto de javalis e soldados romanos. Acção e astúcia constituem a mistura explosiva deste superjogo inspirado no filme homónimo.

REF. J007 PREÇO: 5 265\$00

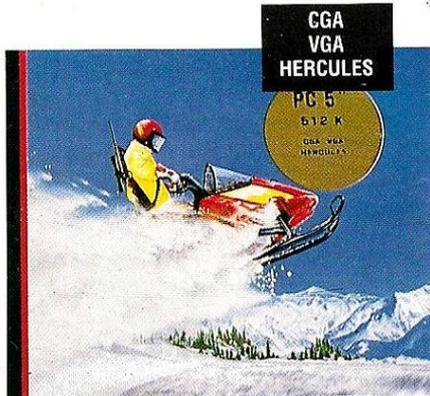


CGA
VGA
HERCULES

A LENDA DE DJEL

Procurar a todo o custo salvar o reino da fome e das epidemias, entre outros males não menos graves, é apenas uma das muitas tarefas de que está incumbido ao iniciar este jogo. Ao longo de cerca de três dezenas de ecrãs tudo se tornará possível. A vitória ou a derrota dependerão em grande parte de si.

REF. J002 PREÇO: 5 265\$00



CGA
VGA
HERCULES
PC 5"
512 K
CGA VGA
HERCULES

SKIDOO

Pilotando uma *scooter* para neve, você faz um *raid* de 6 meses para completar as etapas do campeonato. A 180 Km/h, o seu bólido dos gelos atravessa as florestas, lagos gelados e montanhas do grande norte-canadiano. Em cada uma das 25 povoações, você pode participar em corridas diabólicas sobre gelo, onde nove adversários tudo tentam para o fazer «comer» o gelo.

- Simulação excitante de condução dum *Skidoo*.
- Mapa realista do nordeste canadiano (4 000 000 km²)
- Paisagens inesquecíveis.
- 25 percursos de provas diferentes.

REF. J008 PREÇO: 5 265\$00



EGA
CGA
PC 1512

20000 LÉGUAS SUBMARINAS

Heróis desta fantástica aventura de Julio Verne, os jogadores vivem momentos inquietantes e misteriosos nas profundezas do oceano, percorrendo as páginas de um romance clássico de uma forma muito mais interessante e... menos clássica.

REF. J004 PREÇO: 5 265\$00

TODOS OS JOGOS SÃO ORIGINAIS, INCLUINDO A(S) DISQUETE(S), CONDICIONADA(S) EM EMBALAGEM PRÓPRIA ADEQUADA A CADA JOGO, E ACOMPANHADA(S) POR POSTERS E/OU MANUAIS DE INSTRUÇÕES. *SÃO PROIBIDAS A REPRODUÇÃO, POR FOTOCÓPIA, DA DOCUMENTAÇÃO IMPRESSA, E A CÓPIA DOS PROGRAMAS AQUI COMERCIALIZADOS.

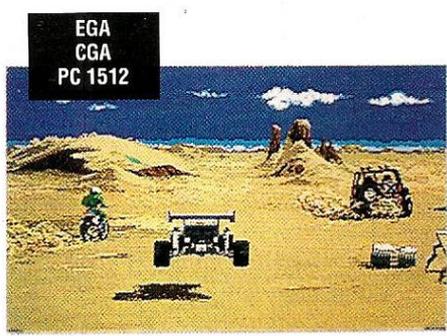


EGA
CGA
PC 1512

FREEDOM

Numa plantação do século XVIII a sua missão é tentar uma evasão na posição de líder de uma rebelião de escravos. Mesmo com a ajuda dos restantes escravos a tarefa não será nada fácil: enfrentar todos os inimigos, os cães, e o próprio dono da plantação não é um trabalho que qualquer um possa empreender. Um excelente jogo de estratégia e acção.

REF. J005 PREÇO: 5 265\$00



EGA
CGA
PC 1512

AFRICAN RAIDERS

Simulação de condução 4x4 no deserto. Simulação bastante realista, sem estradas, com algumas pistas, obstáculos naturais, e numerosos participantes para vos dificultarem a vida até Dakar. Dispondo de um satélite de bordo, uma bússola, e de um mapa como instrumentos principais de trabalho a tarefa de chegar "a bom porto" não se mostra nada fácil.

REF. J001 PREÇO: 5 265\$00

**Consultdata**

SERVIÇOS DE INFORMÁTICA, LDA.

— DISKETTES DE 3" — 3,5" — 5,25" E 8"
BAIXA E ALTA DENSIDADE

— DATA CARTRIDGES (STREAMTAPES)
DE 10 — 20 — 45 — 60 E 150 MEGABYTES

— DISK CARTRIDGES E DISK PACKS
PARA GRANDES SISTEMAS

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

— FILTROS ANTI-MAGNÉTICOS E ANTI-ESTÁTICOS
PARA TODOS OS ECRANS DE COMPUTADOR

— PASTAS PARA ARQUIVO DE FORMULÁRIOS
DE 9,5X12 — 12X9,5 — 12X15,5 E 15,5X12
E RESPECTIVOS SEPARADORES

— MOBILIÁRIO PRÓPRIO PARA CENTROS
DE INFORMÁTICA

Consultdata

R. da Quinta do Almargem, 7-A 7-B — 1300 LISBOA - PORTUGAL
Telefs.: 64 48 54 - 64 51 38 - 64 83 06/7/8
Telex: 63755 ULDATA P Telefax: 64 91 08

PHILIPS FAX SÉRIE 3000



Um destes FAX's resolve o seu problema de comunicação.

Desde o modelo 3100 ao mais sofisticado modelo 3500, a PHILIPS orgulha-se de apresentar uma completa linha de FAX's para satisfazer as suas necessidades de comunicação, de uma forma simples e inteligente.

Com os FAX's PHILIPS Série 3000, os seus documentos podem surgir automaticamente em mais de 15 mil locais diferentes, através do método normal de transmissão, ou em transmissão protegida por código de 4 dígitos.

A recepção de documentos, com impressão ajustada a 16 tons de cinzento e dimensões máximas de papel equivalentes ao formato A3, fazem dos FAX's PHILIPS da Série 3000 uma linha de equipamentos de telecópia perfeita para complementar o seu escritório electrónico com ligação a sistemas informáticos existentes.

A comunicação eficiente tem agora um nome próprio:

**TELECOMUNICAÇÕES
E INFORMÁTICA**

Av. Eng.º Duarte Pacheco, 6 — Apartado 1331 — 1009 LISBOA CODEX
Tel. 683121 — Telex 16494
Rua Sá da Bandeira, 752 - r/c — 4000 PORTO — Tel. 310555 — Fax. 310833



PHILIPS