

**CALCULATOARE
PERSONALE**

7 / 91

(9)

69 Lei

Revistă lunară editată de Micro ATCI S.R.L. Tîrgu Mureş

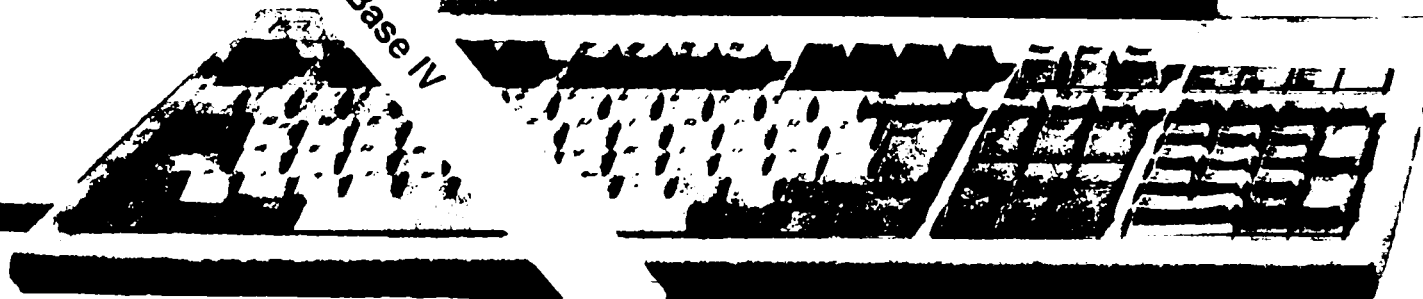
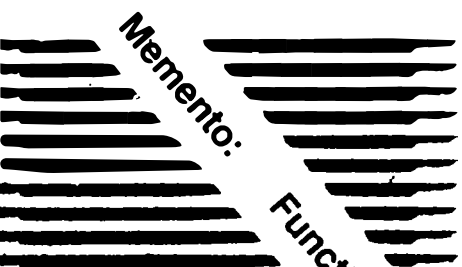
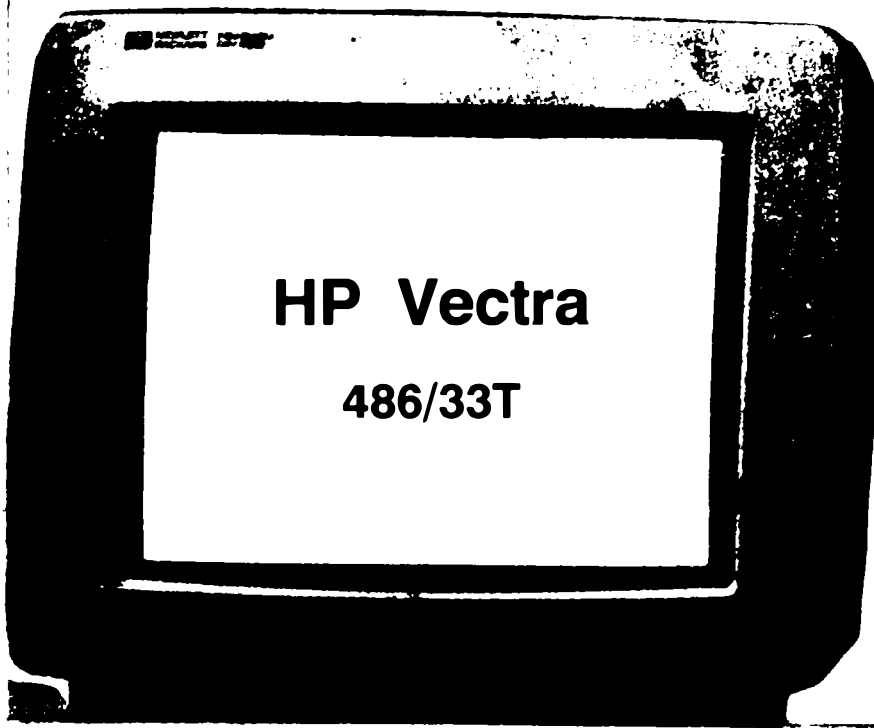
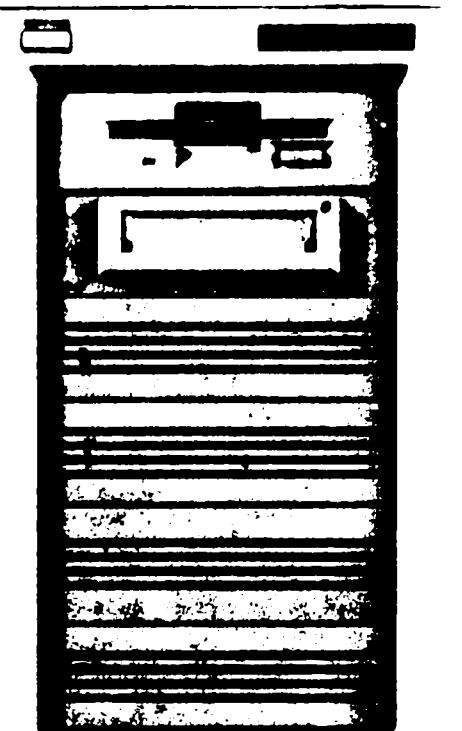
ISSN 1220-1529

Fileservere

10 programe antivirus

Multiuser DOS 5.0

Cele mai bune jocuri pe calculator



if

revistă de informatică
editată de firma Micro ATCI

Director: ing. Dumitru Dunca

Redacția:

ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:

ing. Vasile Brînduș
ing. Attila Darvas
ing. Alin Flaidăr

Tiparul: tipografia Tîrgu Mureș

Revista apare lunar.

Preț: 69 lei

Manuscrise originale sau listin-
guri de programe sînt primite cu
plăcere de redacție, cu condiția
să nu fi fost publicate și în altă
parte. Prin expedierea unui ma-
nuscris pe adresa redacției, au-
torul consimte implicit la
publicarea materialului său în ca-
drul revistei. Onorariul se nego-
ciază cu redacția. Materialele
nepublicate nu se înapoiază și
nu se rețin.

Revista noastră vă oferă spațiu
pentru reclamă și publicitate.
Pentru amănunte vă rugăm să
luați legătura cu redacția.

Cei care doresc să anexeze re-
vistei pliante publicitare tipărite
în regie proprie, sînt rugați și ei
să se adreseze redacției.

Adresa și telefonul redacției:

Micro ATCI, redacția "if",
RO-4300 Tîrgu Mureș,
C.P. 172, O.P. 1,
tel. 954/31660 (direct),
33612, 24158, 20057, 33511,
int. 134 sau 189
41417 sau 46885 (după ora 19)
fax 954/36489 (Intertours)
telex 65354 (Intertours)

Panta rhei

Totul curge, dar apa trece, structurile rămîn și cum nu știm, (și nu vrem a-nvăța), să umblăm cu palma întinsă, și neavînd abilitatea de-a ne pune sub oblăduirea puterilor momentului, nu vă putem întîmpina nici de această dată cu un zîmbet liniștitor.

"Trăim în miezul unui ev aprins" în care în continuă schimbare nu este doar geografia politică ci și geografia informaticii. Firme care pînă mai deunăzi se băteau pe viață și pe moarte, pentru acapararea pieții, devin aliate. I.B.M. și Apple au semnat un contract de colaborare, (detalii în pag. 4), care va modifica sensibil relațiile în lumea "celor mari". În afara proiectelor comune și a faptului că vor păși împreună pe piață, în viitor se va miza pe microprocesoare Motorola, și pe platforme soft comune, ceea ce fără îndoială este o lovitură puternică pentru Intel și Microsoft.

La închiderea ediției am aflat și că firma Ashton-Tate a fost preluată de Borland, iar firma Digital Research a fost preluată de Novell. Ce ne va mai aduce viitorul, și ce implicații vor avea aceste schimbări?

Altfel ce mai faceți? Bine, sănătoși? Calculatoarele pe care lucrați nu se comportă ciudat? Nu aveți musafiri nepoftiți? Dacă da, citind articolul de prezentare a 10 programe antivirus, puteți alege antidotul cel mai potrivit. Dacă există!

Pentru ne-"liniștea" Dvs., vă informăm că zilele trecute am văzut la lucru primul virus autohton. Dacă virusul este activ și se tastează întîmplător combinația de litere "Jos", calculatorul răspunde afișînd în continuare combinația de litere "Iliescu!". Virusul se răspîndește cu ajutorul fișierelor cu extensia COM și corpul lui conține mesajul "The first Romanian Political Virussian". Puteți scăpa de el, (de virus), ștergînd fișierele infectate.

Dar să vă dăm și o veste bună. După, de acum, cunoscutele serii de cărți de informatică editate de MicroInformatica din Cluj, noi edituri lansează serii de titluri dedicate informaticienilor. Editura Cristian din București, a lansat colecția intitulată "Tehnocrat", prima carte apărută numindu-se "Prezentarea sistemului de operare MS-DOS", DATACOM din Cluj a lansat colecția "Informatica de buzunar", colecție din care au văzut lumina tiparului volumele "IBM - PC în lucrări", "AutoCAD" și "Lotus 1-2-3", iar editura Facla din Timișoara este prezentă în librării cu volumul "Forth - Concept informatic și limbaj de programare", avîndu-i ca autori pe Radu Berindeanu și Agota Matekovits.

Începînd din numărul 9/91 al revistei inițiem un concurs: "1024 lei pentru cel mai bun program sursă de cel mult 1024 Byte". Sînteți programator? Ați scris unul sau mai multe programe care rezolvă o anumită problemă în mod genial? Atunci nu ezitați, și trimiteți-ne programul sursă, împreună cu datele despre compilatorul utilizat și adresa Dumneavoastră. În fiecare lună vom premia cel mai bun listing cu 1024 lei. Singura condiție: textul sursă nu trebuie să fie mai lung de 1024 Byte (1 KByte). Listingurile publicate și nepremiate vor fi onorate cu tariful obișnuit. Vă așteptăm!

Ing. Romulus Maler



Cuprins

Știri

- IBM și Apple împreună pag.4
- dBase cu suport SQL pag.5
- Compilator C OOP pentru Windows pag.5
- Coprocesoarele vor accelera Windows pag.6
- Noi harddisk-uri de 2,5' pag.6
- Dumping pag.6
- Hello CAD-FANS ! pag.6

Magazin

- Plăcerea jocului pag.7

Cele mai interesante jocuri pe calculator și cele mai bune programe utilizator ale anului 1990 au fost premiate la al treilea festival "European Computer Trade Show" din Londra.

Vă prezentăm

- 10 programe antivirus pag. 10

- Antivir Plus
- Virusfinder PC
- Dr. Solomon's Antivirenwerkzeuge
- Iris Antivirus Plus
- Norton Antivirus
- PCcillin
- PC Plus Virus Doctor
- McAfee Scan
- Turbo Antivirus
- Virus Blocker

Fundamente

- Multitasking pag.20

În legătură cu apariția Windows 3.0, tot mai des apare termenul de multitasking. Cîteva din elementele de bază ale acestui concept sînt analizate în acest articol.

- Partiționarea harddisk-ului pag.38

- DOS - extindere pag.42

Programarea sub DOS în mod protejat

Sisteme de operare

- Digital Research Multiuser-DOS 5.0 . . . pag.24

Rețelele sînt prea scumpe pentru multe întreprinderi. Se caută soluții mai avantajoase - una din ele ar putea fi Multiuser DOS 5.0 .

Rețele locale

- Fileserver: alegere corectă pag.26

Dintre toate PC-urile din rețea, fileserver-ele sînt cele mai importante.

Cursuri

- OOP - partea a 5-a pag.29

În partea finală a acestui curs sînt examinate problemele care apar la planificarea ierarhiilor de obiecte.

- Turbo-Prolog - partea întâi pag.34

Limbajul de programare Prolog este un instrument excelent pentru rezolvarea de probleme logice. În prima parte, a unei serii de 3 episoade, cîteva din principiile de bază.

Hardware

- Interfața SCSI - Prezentare și utilizare . . pag.46

Practică

- Lupa de timp pag.50
- Tabela ASCII confortabilă pag.50
- Tabela ASCII rezidentă pag.51
- Economie de memorie RAM pag.51
- Igiena zilnică pag.52

Rubrici

- Caseta redacției pag. 1
- Editorial pag. 1
- Poșta redacției pag.54
- Mica publicitate pag.54
- Ne-am interesat pentru D-voastră . . . pag.53

Lista modemurilor avizate de Ministerul Comunicațiilor

IBM și Apple împreună

În lumea calculatoarelor se schimbă întreaga geografie. Cei mai mari rivali IBM și Apple au încheiat un contract preliminar care prevede cooperarea lor în toată lumea. Împreună cei doi parteneri dețin aproximativ 40% din piața mondială de PC-uri.

Șeful firmei Microsoft, Bill Gates a avut o presimțire. "Concurentul se cheamă acum IBM", a anunțat el cu câteva săptămîni mai înainte, printr-o notă internă, pe toți membrii conducerii Microsoft. El a presimțit deja de câteva săptămîni că harta valabilă pînă în acest moment a alianțelor și a rivalităților lumii calculatoarelor va ajunge de nerecunoscut.

Presimțirile nu l-au înșelat pe Gates: IBM și Apple au anunțat că vor apărea de-acum înainte împreună în fața întregii lumi. A fost semnat deja un contract preliminar de cooperare în toate domeniile. Contractul dintre IBM și Apple schimbă cu 180 de grade geografia "cine cu cine împotriva cui".

Cei doi inventatori ai PC-ului, Apple și IBM, au speriat restul lumii

din branșă anunțînd că de acum încolo ei vor coopera, în toate domeniile: dezvoltarea de software, integrarea calculatoarelor Apple-Macintosh în arhitectura calculatoarelor IBM AIX, dezvoltarea procesoarelor RISC cît și o platformă Multimedia comună pentru industrie.

Contractul de cooperare al foștilor rivali este o știre extrem de proastă pentru unii dintre colegii lor de branșă: producătorul de procesoare INTEL, care pînă acum aproape că a deținut monopolul asupra procesoarelor PC va afla din contract că IBM și Apple își vor procura de acum înainte procesoarele de la Motorola. Și nici Bill Gates nu se va bucura atunci cînd va citi detaliile cooperării IBM-Apple: cei doi parteneri vor dezvolta printre altele și o nouă "platformă deschisă de software". "Noua platformă" va fi integrată în sistemele de operare existente. Nu se spune nici un cuvînt despre faptul că Microsoft va avea ceva de spus în această dezvoltare.

Unitatea celor care pierd

Odată cu acest pact cele două puteri ale lumii PC-urilor s-au unit. Atît Apple cît și IBM au creat cîndva PC-ul. Apple a reușit deja în 1976, IBM, în schimb, a ignorat noutatea timp de 5 ani și în 1981 s-a aflat în fața faptului împlinit că a întîrziat în dezvoltarea platformei anilor 80. Rîvna IBM-ului de a anula distanța atinsă de Apple a dus la poziția sa în cadrul pieții de PC-uri, care a rămas valabilă un deceniu.

Atunci IBM a căutat un furnizor de procesoare PC și a dat de Intel. Pentru a anula avansul de 5 ani față de Apple, IBM a oferit sistemul de operare DOS, creat de Microsoft, spre difuzare și altor producători. A apărut efectul scontat: cu toate că lumea IBM-DOS a PC-urilor a schiopătat în urma standardului creat de Apple, industria de calculatoare s-a bazat în întreaga lume pe standardul "deschis" IBM și a inundat piața mondială cu PC-uri ieftine după standardul IBM. PC-ul a fost descoperit a doua oară.

Cu această politică IBM a reușit să cîștige doar 25% din piața mondială la PC-uri, dar scopul principal a fost atins: partea acaparată de Apple a scăzut la 15%. Uriașul rest al pieții de calculatoare de 60% și l-au împărțit între ele clonele IBM-ului. Direct de acest lucru s-a legat și triumful lui Intel: în afară de Apple nu mai folosea aproape nimeni procesoare Motorola, toți lucrau cu furnizorul de procesoare al IBM-ului Intel, care și-a asigurat astfel un cvasimonopol.

Au existat probleme la ambele întreprinderi, care au dus la uitarea vechii rivalități. La IBM afacerile se tîrîie, în primul trimestru al anului 1991 a fost înregistrată o pierdere de 1,73 miliarde de dolari. Șeful de la IBM Jone Akers a reacționat destul de violent vis-a-vis de manageri, punîndu-le în vedere că dacă nu-și schimbă stilul lent de acțiune îi va da afară. Și Apple a avut greutăți neașteptate. Tipurile noi "Clasic" și

Înțelegerea

Contractul preliminar dintre IBM și Apple în vederea colaborării în viitor, pe plan mondial, conține patru puncte principale. Iată cele mai importante pasaje din declarația care a fost dată publicității:

1. Joint-Venture pentru software.

Apple și IBM au de gînd să dezvolte o nouă platformă deschisă de soft de sistem, care să se bazeze pe o tehnologie orientată obiect. Această nouă platformă va cuprinde toate sistemele de calculatoare începînd de la Laptop-uri și pînă la servere performante. În acest scop se va înființa o firmă de software de sistem, care va rămîne în proprietatea lui Apple și IBM.

2. Integrarea de Apple Macintosh în sistemele IBM

Ambele întreprinderi vor dezvolta și vor comercializa produse de rețea și de comunicații. Aceste produse vor mări posibilitatea de introducere a unui

Apple Macintosh într-un mediu IBM. În plus ambele întreprinderi vor continua dezvoltarea AIX, care va dispune de suprafețele utilizator Macintosh și OSF/Motif.

3. Piețe semnificativ noi pentru arhitectura IBM-Power-RISC.

Apple are în plan implementarea Single-Chip-ului IBM-RS/6000-Power în calculatoarele Macintosh. Ambele întreprinderi vor să integreze micro-procesoarele Power-PC în stații de lucru și în Fileservere.

Motorola și IBM vor să-și aducă contribuția la dezvoltarea și la producția pe viitor a unei noi generații de Power-PC-Chips cu performanțe superioare.

4. Platformă Multimedia comună pentru industrie

IBM și Apple planifică crearea și licențierea în comun a unui mediu software independent de platformă.

"LC" s-au vândut bine, dacă se însu-mează numărul de calculatoare vândute, dar la prețurile joase practice nici Apple nu s-a putut descurca. Și conducătorul întreprinderii John Sculley a fost obligat să anunțe concedieri. Astfel, la nevoie, Akero și Sculley, s-au aliat. Rezultatul tratativelor a fost pactul Akers-Sculley.

dBase cu suport SQL

Se ia dBase IV, se completează cu suport SQL și se obține dBase IV-Server-Edition. Împreună cu Serverul Microsoft-SQL el va deveni platforma pentru aplicații Client-Server.

Mult așteptatul "Server Edition" este gata. La 26 iunie Ashton Tate a prezentat cel mai tânăr copil al familiei dBase, în timpul expoziției de PC-uri din New York. Server Edition nu este o concurență la Oracle sau la SQL-Server, ci un Front-End capabil de SQL. În cadrul aplicațiilor Client-Server el joacă rolul clientului care filtrează date din serverul bazei de date via SQL. În momentul de față Server Edition lucrează doar împreună cu serverul Microsoft-SQL, sprijinirea altor sisteme, ca Oracle sau Sybase, va urma într-o versiune ulterioară.

"Strategia Ashton Tate este de a da utilizatorului acces asupra datelor indiferent de locul în care au fost memorate, sau indiferent cu ce calculator se lucrează, a afirmat David Procter, președintele firmei Ashton Tate, în cadrul prezentării lui Server Edition. Primul pas a fost implementarea lui dBase pe cât mai multe platforme începând cu DOS, UNIX și până la VMS. Urmează deschiderea către serverele de baze de date. Pentru a simplifica trecerea programatorilor dBase de la o platformă la alta, comenzile SQL au fost preluate, în mod transparent, în liniile de comandă dBase. Astfel același program poate accede atât date locale în format dBase, cât și interoga baze de date prin serverul SQL. Toate interogările se traduc de Server Edition în SQL, astfel încât este ne-

Compiler OOP pentru Windows

Cea mai nouă versiune a compilatorului "Zortech C++ 3.0" și a bibliotecii "Zink Interface Library 2.0" strălucesc printr-o gamă variată a noilor funcțiuni speciale pentru programarea orientată obiect sub Windows.

Produsele Zortech se numără deja de ani de zile printre primele de pe piața compilatoarelor. Cu versiunea 3.0 a lui C++ Zortech vrea să-și consolideze această poziție. Ea conține șase îmbunătățiri, dintre care una în programul propriu-zis: optimizarea codului de program și astfel o creștere considerabilă de viteză.

Celelalte cinci sînt îmbunătățiri ale pachetului prin adăugiri: astfel o documentație voluminoasă de Windows-API prototipul "Windows Maker", la care Zortech a cumpărat licența de la Blue Sky, "WINC Libraries", cu care programele DOS pot fi recompilate pentru Windows și un 16-Bit-DOS-Extender, pentru care firma a cumpărat licența de la Rational System. Astfel cu un comutator al compilatorului dintr-un program normal DOS se obține unul pentru Protected Mode.

Întregul pachet a lui Zortech C++ 3.0 va fi oferit pe piață în trei versiuni diferite.

Ediția de bază se pretează pentru DOS și pentru Windows 3.0. Versiunea dezvoltată utilizează în locul unui 16-Bit-DOS-Extender un 32-Bit-Extender, care consumă doar 15 KByte RAM.

Pachetul tehnico-științific sprijină în plus coprocesoarele de la Weitek și specificațiile "IEEE-748" - de calcul în virgulă flotantă, extensiile "NCEG C" și mai multe module M++, care pot fi obținute separat pentru statistică și alte aplicații tehnico-științifice speciale. Versiunea de bază costă 250 DM versiunea dezvoltată 400 DM și cea tehnico-științifică 600 DM.

Foarte intensată a fost și prezentarea bibliotecii Zinc Interface Library 2.0. Ea lucrează împreună cu compilatoarele firmelor Borland și Zortech. Astfel programarea aplicațiilor se poate face în același timp în 3 moduri: DOS-text, grafic și Windows 3.0. Codul sursă trebuie mărit cu doar câteva rînduri de cod suplimentar, pentru a obține alternativ o fereastră pentru DOS și alta pentru Windows.

cesară o adaptare la diferitele servere de baze de date. Doar astfel pot fi exploatate funcțiunile specifice ale fiecărui server în parte. Din punct de vedere al programatorului, aceasta înseamnă că aplicațiile de astăzi vor putea fi adaptate cu minimum de efort la versiunile ulterioare, deoarece Server Edition preia el comunicația cu sistemul SQL și nu trebuie învățate diferitele dialecte SQL. Dacă se dorește, un program poate referi direct baza de date prin comanda SENDSQL. Interogările interactive se fac cu ajutorul comenzii SQL SELECT. Utilizatorul poate introduce comenzile direct în timp ce programul se

afală în execuție, iar dBase le interpretează în mod dinamic.

dBase-IV-Server-Edition funcționează pe toate calculatoarele MS-DOS începînd cu 286 cu o memorie de cel puțin 2 MByte. Ashton Tate recomandă însă cel puțin 3 MByte Extended Memory (XMS) pentru a se putea atinge o viteză optimă. Programul sprijină pînă la 15 MByte XMS și funcționează și atunci cînd este pornit din modul 386 al lui Windows 3.0. Poate lucra împreună cu servere sub Novell Netware 286/386, 3COM3 Plus și Microsoft-OS/2-LAN-Manager.

În SUA dBase-IV-Server-Edition costă în jur de 8000 \$ pentru 20 de utilizatori și poate fi cumpărat deja.

Coprocresoarele vor accelera Windows

Nici Bill Gates n-ar fi visat acest lucru. Branșa calculatoarelor a fost cuprinsă de o adevărată manie Windows.

Aplicațiile Windows răsar peste tot ca ciupercile după ploaie, foști opozanți recunosc deschis aprecierea lor față de suprafețele grafice iar tot mai mulți producători oferă hardware specific Windows.

În cazul imprimantei laser Windows sau a diferitelor PC-uri Windows (care ar fi create mai ales pentru Windows) este vorba de niște produse ale unor manageri plini de fantezie, care vor să vîndă tehnică depășită sub un nou nume.

Lucruri interesante se pot observa în schimb pe ecranele locurilor de muncă CAD din Silicon Valley. Ingineri rafinați lucrează aici la coprocresoare care sînt făcute special pentru Windows. Coprocresor nu înseamnă doar "componentă pentru operații matematice" cum s-ar putea crede la vederea soclurilor (folosite ce-i drept foarte rar) 287 și 387 din PC-uri.

Un exemplu bun este cartela XGA de la IBM. Acest cip video po-

sedă exact funcțiunile care iau driver-ului grafic Windows cel mai mult timp: trasarea liniilor, deplasarea blocurilor grafice etc. Mulți producători vor să urmeze acest exemplu. Weitek, Cyprix și Intel deja bricolează la cîte un Windows-Turbo-Chip. Ideile lor ajung de la simple cipuri grafice pînă la combinații de controlere cache, grafice și de sunet. În momentul de față se pare că la Weitek lucrările sînt în stadiul cel mai avansat.

Inginerii de acolo au prezentat doar de cîteva săptămîni o soluție cu două cipuri SPARC, care nu conține doar procesorul, ci și unitatea de virgulă flotantă, unitatea de management a memoriei și video-cipul. La o integrare corespunzătoare, un cip asemănător ar putea scoate PC-urile din multe situații dificile care apar la lucrul cu Windows; mai ales în ceea ce privește memoria de lucru și performanțele grafice.

Și de la acest procesor ne putem aștepta la accelerarea funcțiunilor matematice, aceasta fiind o specialitate a celor de la Weitek.

Rămîn încă de clarificat cum se va integra cipul Weitek în PC-uri. Unele surse vorbesc despre o cartelă de accelerare Windows, altele de un cip OEM, care va fi integrat pe placa de bază. În acest caz producătorii de PC-uri ar trebui să treacă în fapt la PC-uri speciale pentru Windows.

La Intel toate reflecțiile se învîrt în jurul soclului 487SX. Deoarece după ultimele știri 487SX este doar un cip complet 486 cu o modificare foarte mică a configurației de pini, soclul poate fi ocupat și de un procesor independent.

Acesta oferă pentru viitor posibilități interesante de dezvoltare: cine cumpără astăzi un calculator 486SX cu tactul de 20 MHz mîine îl va putea completa cu un procesor de 40 MHz. Căci Intel lucrează, după propriile sale afirmații, la cipuri, care funcționează intern cu o viteză de tact dublă decît BUS-ul. Dau de gîndit și noile cipuri speciale, cum ar fi 486SX cu cip video integrat sau cu memorie cache mult mărită.

Noi harddisk-uri de 2,5"

Seagate Technologies iese pe piață cu două harddisk-uri noi de 2,5" pentru Laptop-uri. Acestea vor fi mult mai rapide - timp mediu de acces sub 20 ms - și vor avea un buffer de 32 KByte.

Cele două harddisk-uri "ST9077A" și "ST9038A" oferă o capacitate de stocare netă de 64 MByte și respectiv 32 MByte și ambele sînt dotate cu o interfață AT. Viața medie a celor două harddisk-uri este estimată la 150.000 MTBF.

Dumping

Ministerul comerțului din Statele Unite reproșează producătorilor japonezi Matsushita și Brother că ar fi vîndut coprocresoare de texte la prețuri de dumping situate cu 58% sub prețurile "oneste" practice.

Cei învinovați neagă aceste reproșuri. Dacă în constătuirea ce va dura 45 de zile aceste învinuiri nu vor fi ridicate, atunci producătorii japonezi vor trebui să achite niște amenzi vama care vor trebui să acopere diferențele dintre prețul de dumping și prețul stabilit de minister ca fiind onest.

Hello CAD-FANS !

Numărul revistelor de informatică este în continuă creștere. După PC Magazin, Infoclub, hobBIT, 01 Informatica, salutăm apariția revistei Hello CAD-FANS, revistă dedicată proiectării asistate de calculator. Spicuim din cuprins: "Ghidul începătorului", "AutoCAD pe felii", "Atelier CAD", "Introducere în AutoLISP. Revista are 32 de pagini și costă 30 de lei. Pentru detalii suplimentare vă puteți adresa la: Revista "Hello CAD_FANS", C.P. 39-119 București, tel. 53.43.98. Urăm colegilor noștri: "La cît mai multe numere!"

Plăcerea jocului

Cele mai interesante jocuri pe calculator și cele mai bune programe utilizator ale anului 1990 au fost premiate la al treilea festival "European Computer Trade Show" din Londra.

Dacă doriți jocuri foarte noi nu trebuie să faceți altceva decât să mergeți primăvara la Londra la târgul european de calculatoare - un târg pentru specialiști, care a avut loc doar pentru a treia oară, dar unde se întâlnesc toți producătorii de hardware și software, care au ceva de spus în acest domeniu.

În Business Design Center aflat în cartierul Islington au fost prezentate noutăți extrem de fierbinți, de exemplu : simulatoare de zbor foarte realiste, care fac să-ți crească tensiunea (cum ar fi "Birds of Prey" - Electronic Arts sau "The

Secret Weapons of the Luftwaffe" - Lucasfilm Games).

Pentru a încerca modul de viață în cadrul unor lumi sintetice pe calculator există un demo de Virtual Reality cu cască pentru vizualizare 3D și cu mânășă de date.

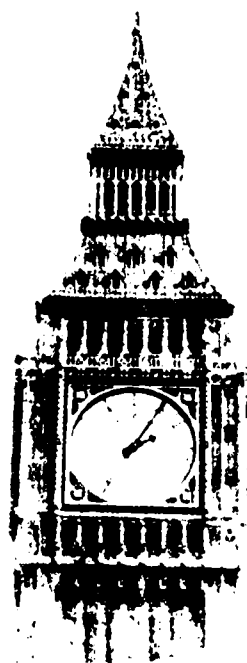
În 1991 a fost decernat pentru a doua oară la această întâlnire europeană a producătorilor de jocuri premiul "European Computer Leisure Award". Au fost premiate cele mai bune jocuri ale anului cât și cele mai bune programe utilizator pentru folosință privată. Juriul a fost format din redactori de la 32 de reviste de calculatoare din Europa, SUA și Japonia, având în total în jur de 8 milioane de cititori. Produsele au fost premiate datorită supe-

EUROPEAN COMPUTER TRADE SHOW
91



riorității lor la categoriile : grafică, sunet, acțiune, originalitate, animație, jocuri de aventuri, respectiv cu roluri cât și jocuri logice, împachetare și accesorii ("packaging") și jocuri de simulare foarte apropiate de realitate, jocuri pentru console și console.

Dintre programele utilizator au fost premiate cele mai bune soft-uri de pictură "Art Package", de realizat muzică "Musik Package", de învățare "Home Education", în plus



Cîștigătorii	
Jocuri	
Jocul anului:	Lemmings
Cea mai bună grafică:	Shadow of the Beast II
Cel mai bun sunet:	Shadow of the Beast II
Cea mai palpitantă acțiune:	Killing Game Show
Cel mai original joc:	Lemmings
Cea mai bună animație:	Dragons Lair II - Time Wrap
Cele mai palpitante aventuri:	Secret of Monkey Island
Cel mai bun joc de gândire:	Klax
Cel mai bun suport:	Ultima VI
Cea mai bună simulare:	F19 Stealth-Fighter
Aplicații	
Cel mai bun program de pictură:	Deluxe Paint III
Cel mai bun program de muzică:	Bars & Pipes
Cel mai bun program pentru învățămînt:	Fun School 3
Cel mai bun "Home Productivity Package":	Microsoft Works
Cel mai inovativ periferic:	Video Toaster
Cel mai bun pachet multimedia:	Amiga Vision
Home Computer-ul anului:	Amiga 500

perifericul, pachetul multimedia ca și Home Computer-ul anului (din nou: Amiga 500).

Cele mai multe premii le-a primit casa de soft britanică Psygnosis în calitatea de cel mai bun producător de jocuri al anului cât și în calitatea de producător al celui mai bun joc pe calculator din 1990 : Lemmings - care este jucat pretutindeni unde se află ecrane Atari ST și Amiga. În continuare vor fi prezentate produsele premiate.

Psygnosis a obținut, în clubul de noapte Mega "Hippodrome" de pe Leicester Square, în fața a peste 500 de oaspeți internaționali, alte premii întii, și anume pentru jocul cu cea mai bună grafică și cel mai bun sunet (Shadow of the Beast II), pentru cel mai bun joc de acțiune (Killing Game Show) și pentru jocul cel mai original (Lemmings).

Cele mai bune programe de jocuri

Cea mai bună grafică: Shadow of the Beast II.

Joc de acțiune, lupte mitice în mediu fantastic (Fantasy-Ambiente), cu un spectacol de grafică color impresionant încă de la început (așa-numitul "Edel-Intro"), împodobit cu secvențe muzicale patetice și cu un generic final lung. În schimb este foarte greu de jucat și în plus și incomod (cu toate că se vinde împreună cu un T-Shirt este extrem de scump: 110 DM)

Sistem: Amiga - ST
Producător: Psygnosis
Ofertant: United Software
Preț: 110 DM

Cel mai bun joc de inteligență: Klax

Este un joc de îndemânare plin de idei, care folosește o grafică medie, dar ideea de joc este foarte incitantă, asemănătoare cu ideea jocului "Tetris".

Problema : La capătul unei benzi rulante cu 5 coloane se află câte un câmp de stivuire. De pe banda rulantă se apropie de jucător blocuri de culori diferite. Cu un fel de lopată jucătorul trebuie să ridice blocurile și să le depoziteze pe una dintre stive. Stivuirea trebuie realizată astfel încât cel puțin 3 blocuri de aceeași culoare să se afe succesiv pe orizontală, pe verticală sau pe diagonală. Dacă acest lucru reușește, aceste blocuri dispar făcând loc pentru altă tură. În total există o sută de nivele de joc cu grad de dificultate crescător care dau tot mai mult de furcă jucătorului.

Sistem: PC, Amiga, ST, C64, PC-Engine (pentru Mega Drive este în pregătire)
Producător: Tengen, Japonia
Ofertant: Bomico
Preț: între 50 și 100 DM (în funcție de sistem)

Cel mai bun joc de acțiune: Killing Game Show

Este un joc de dificultate medie cu lovituri, pocnituri, cu grafică color deosebit de bună, cu muzică excelentă, cu efecte sonore foarte bune și cu vorbire.



Klax

Ca participant la un spectacol de televiziune din viitor, jucătorul este introdus într-un costum zornăitor de robot și expus într-o zonă plină de primejdii unde trebuie să lupte pentru supraviețuire - amenințat de marea mortale de acizi și de formațiunile maligne și răuvoitoare de extraterestri.

Sistem: Amiga (ST în pregătire)
Producător: Psygnosis
Ofertant: United Software
Preț: 80 DM

Cea mai bună animație: Dragon's Lair II Time Warp

Este un joc de acțiune cu grafică bună și saturat cu sunet, chef de joc este limitat. Prințesa Daphne a fost răpită de vrăjitorul cel rău Morlock și bunul Dirk trebuie s-o salveze. Eroul călătorește cu mașina timpului în trecut, în grădina Eden, și în viitor, mereu în luptă cu Morlock, cu soacra acestuia și cu câțiva alți monștri. Scenele animate pot fi influențate în momentul oportun de către jucător.

Sistem: Amiga, IC
Producător: Redysoft
Ofertant: United Software
Preț: 110 DM.

Jocul de consolă al anului: Tetris

Consola anului: Atari Lynx

Consolele de joc sînt automate de joc, video, de buzunar - computere complete în format Walkman - avînd un ecran de mărimea unei cutii de chibrituri, pe care se pot derula doar jocuri. Soft-ul este sau încorporat sau este livrat în module care se introduc în aparat.

Au ajuns foarte îndrăgite consolele independente de rețea ale specialistului în console Nintendo și anume Game-Boy și Atari "Lynx".

Odată cu Game-Boy se livrează jocul de îndemînare Tetris, unul din jocurile care prinde foarte mult, producînd chiar dependență.

Prețul - cu consolă cu tot: aproximativ 160DM. Pînă acuma au fost vîndute 400.000 de bucăți și Nintendo ar vrea să ajungă pînă la sfîrșitul lui '91 la 1 milion de bucăți.

Consola Lynx nu este nici ea mai prejos, și oferă o veritabilă porție de Computer High-Tech în format de buzunar, cu monitor color LCD și cu confort mai mare în deservire decît Game-Boy. În schimb bateria Lynx ține mai puțin. Ea costă cam 200 DM, și va fi înlocuită încă în acest an de o versiune mai compactă, mai ieftină și mai puțin consumatoare de baterii.

Cel mai bun joc de aventuri: Secret of Monkey Island

Este o comedie hazlie și plină de divertisment folosind o grafică de vîrf (256 de culori din cele 256.000 ale versiunii VGA pentru PC-urile MS-DOS), sunet puternic (sprijină plăcile Adlib și Soundblastes) și ritmuri sacadate Calypso și Reggae.

Acțiune în stil Indiana-Jones: Guybrush Threepwood, care doarește mult să fie pirat, trebuie să treacă multe probe și aventuri,

pentru a fi acceptat de breasla pi-raților Karibik. Simpatice sînt duerurile cu înjurături în loc de săbii.

Sistem: PC, Amiga și ST
Producător: Lucasfilm Games, USA
Ofertant: Rushware
Preț: 100 DM

Cel ami bun suport (Packaging): Ultima IV

Acțiunea în sine este relativ banală - lumea Sosaria trebuie salvată de răii Gargyles, ceva asemănător cu alte jocuri scenice, dar este un joc fantastic realizat într-un mod foarte teatral și cu o grafică excelentă. Totul se petrece într-un scenariu de ev mediu, un cosmos jucărie cu caractere diferențiate psihologic, nuanțări morale, toate fiind completate de un inventar admirabil de consecvent și plin de detalii. Ca supliment se livrează o hartă din pînză, un manual minuscul și o piatră "magică" - în orice caz marfă multă pentru bani puțini (100 DM).

Sistem: MS-DOS (Amiga și C64 în pregătire)
Producător: Origin, Texas, USA
Ofertant: Rushware
Preț: 100



Jocul anului: Lemmings

Este un joc Amiga animat într-un mod deosebit, cu grafică superioară și sunet deosebit.

Problema: De a conduce poporul Lemmings, cît se poate de nedecimat, peste distanțe cu obstacole cu grad de dificultate tot mai ridicat pînă la "Templul Salvării". Handicapul: Lemmingii sînt prostuți și îl urmează pe primul dintre ei în orice situație - chiar și la moarte. De aceea este nevoie de un simț foarte dezvoltat de reacție, planificare și gîndire strategică și tactică - altfel prostuții cad în prăpastii, vor fi lipiți de pămînt de o greutate, prăjiți în foc sau mîncăți de o plantă omnivoră.

Lemmings a transformat chiar și scepticii care afirmau: "jocurile pe calculator sînt sterpe", în fani.

Sistem: Amiga, Arari-ST (Versiunea PC în pregătire)
Producător: Psygnosis
Ofertant: United Software
Preț: în jur de 80 DM

Cel mai bun simulator: F19 Stealth-Fighter

Problema: De a zbura cu cel mai modern bombardier american de camuflaj în misiuni secrete deasupra zonelor de criză (pentru începători: Libia, pentru ași: Europa de Est sau zonele potențiale de război: regiunea golfului și anume Iran).

Jocul de simulare a zborului și a loviturilor este destul de macabru, sînt atacate chiar și uzine atomice. O simulare depășită din punct de vedere politic (rămasă la nivelul anilor războiului rece), dar foarte realistă din punct de vedere militar, tehnic și aviatic.

Asemănător cu filmul Top Gun, jocul este aproape perfect și aduce la suprafață instinctul de ucigaș fără muștrări de cunoștință pe care îl poate descoperi în sine orice pacifist convins.

Sistem: MS-DOS, Amiga, Atari ST și C64
Producător: Microprose USA
Ofertant: Rushware
Preț: 100 DM.

(I.M.)

Dăunătorii calculatoarelor

Virusii calculatoarelor sînt spaima oricărui posesor de PC. În acest articol vom porni împreună pe urmele celor mai întîlniți virusi.

Virusii calculatoarelor sînt una din cele mai "secrete" teme ale prelucrării electronice a datelor. Cum trebuie să te comporți cu ei, și chiar recunoașterea existenței lor, nu este recunoscută ca o temă în sine de mulți producători hard și soft. Abia după ce se întîmplă ne-cazul, se reacționează. Și totuși, tocmai această opinie este eronată. Căci altfel cum ar fi putut "virusii" să-și capete acest renume de temut.

Spaima corelată cu neștiința reprezintă de regulă un bun teren de acțiune pentru afaceriști. Din acest motiv, în ultima vreme, numărul virusilor calculatoarelor a fost umblat intenționat. Se poate pune întrebarea, dacă toate programele de sabotaj, căutate de programele antivirus, există și în realitate. De exemplu se vorbește în literatură de virusul "Agiplan" (prezent și în listele virusilor "recunoscuți"), o specie care a fost stîrpită cu siguranță. El a apărut la sfîrșitul anilor 80, la o singură firmă și nu a mai fost semnalat nicăieri în altă parte. Numai primii 17 octeți ai acestui virus au fost recunoscuți, motiv suficient ca de atunci încoa să se caute prezența acestor octeți. În Africa de Sud, un programator a croșetat în jurul acestor 17 octeți un nou virus. Cine caută, acela găsește.

Uimitor este faptul că numai foarte puțini dintre virusii efectiv răspîndiți, sînt programați să producă pagube. Majoritatea acestor programele se țin de farse, pe moșitor, la imprimantă sau în difuzor. Și introducerea datelor de la tastatură pot deveni opera activității unui virus. Toate acestea nu justifică însă panica.

Există foarte puțini virusi care să reprezinte un pericol real pentru utilizator. Cel mai renumit dintre aceștia este virusul "dBase", care manipulează baze de date și care a fost apreciat ca fiind "cea mai periculoasă lovitură dată prelucrării electronice a datelor". În fapt acest virus a fost semnalat numai de două sau de trei ori. Probabil că acest virus nu mai poate fi descoperit decît în cercurile "cercetătorilor virusilor".

Gurile rele spun că problema principală vizată de virusii calculatoarelor nu sînt virusii înșiși ci chiar acești cercetători. Cu acribia unui pasionat colecționar de timbre, virusii sînt mai întii "descoperiți", adică: ei sînt pasaiți cercetătorilor și cît mai urgent sînt analizați, clasificați și catalogați. Ca urmare a instinctului ancestral al colecționarului, fiecare modificare a unui virus, chiar și banală, este clasificată ca "variantă" sau chiar ca "mutație". Pentru publicul cu mai puține cunoștințe de specialitate, este indiferent dacă, intenționat sau nu, un virus a suportat o modificare necontrolată. În fapt există tot mai mulți programatori care, cu mai mult sau cu mai puțin succes, manipulează virusii astfel încît aceștia nu mai sînt recunoscuți de programele antivirus.

"Din punctul de vedere al majorității tehnicilor neinteligente de

căutare a virusilor, utilizate încă, aceasta este o cutezanță."

În sine vorbind, un virus care nu mai este semnalat, trebuie să devină neapărat un "nou" virus. Așa se face că numărul virusilor catalogați trece peste 500. Doar una, pînă la două, duzini din aceste "boli" au însă o răspîndire reală. Aproape ca și în viața reală, căci pojarul este mult mai răspîndit decît elefantiazisul.

Printre primii virusi s-a numărat și cel programat de germanul Ralf Burger. Pentru scopuri pur demonstrative, se înțelege: honni soit qui mal y pense. Astăzi autorul, avansat ca "expert în sisteme" și "popă al virusilor", în cartea sa "Grossen Computer-Viren Buches" (Marea carte a virusilor calculatoarelor), preferă să nu-și mai amintească de acele vremuri, fie din cauza înduioșătoarei simplități a programului, fie din cauză că în cercurile de specialitate este considerat ca "agent provocateur". Variațiunile pe tema lui Burger au devenit între timp legiuni și sînt desemnate ca "Pixel", "Vienna" sau "Lisabon".

O cunoștință veche este și "170X", unde "X" poate lua valorile "1" sau "4", numărul rezultat dînd lungimea virusului. Una din cele mai noi variante are lungimea de 1714 byte, dar poate fi considerată

Top 25

Cei mai răspîndiți virusi în Germania (în ordine alfabetică)

1260	Fish	Perfume
170x	Frodo	Stoned
8 Tunes	FuManchu	Vaccina
April 1st	Ghost	Whale
Burger și variante	Invader	XA1
Dark Avenger	Jerusalem	Yankee Doodle
DataCrime	Kennedy	Yoshi
Devil's Dance	Music Bug	
Disk Killer	Oropax	

identică cu celelalte. În primele versiuni acest virus lăsa literele de pe monitor "să cadă", fapt care i-a adus numele de "Herbstlaub" (vierme de toamnă), "Cascade" sau "Falling Letters". Unele versiuni distrug și date prin suprascrisere pe hard-disk. Dacă aceste versiuni sînt "experimente genetice" produse în laboratoarele "doctorilor" de viruși, a rămas o enigmă.

Un alt veteran este "Vineri 13", denumit și "Jerusalem Virus". În zilele de vineri care cad pe data de 13 a lunii, toate programele lansate în execuție sînt șterse în mod automat. În celelalte zile, pe monitor apare un dreptunghi negru și calculatorul este "frînat" prin bucle de întîrziere. Și în acest caz există variante care pot deveni active în alte zile ale săptămîinii.

În 1989 a făcut vîlvă virusul cu sonorul nume "Data Crime". Dacă un program infectat este apelat după data de 13 octombrie 1989, pe monitor apare mesajul "DATA-CRIME VIRUS, RELEASED 1ST MARCH 1989". Apoi harddisk-ul este făcut inutilizabil prin ștergerea unor informații importante de pe el. Conform unei declarații în revista americană "Mondo 2000", în acest caz ar fi vorba despre un virus al mediilor: o nadă aruncată de ziariști. După cît se spune, se pare că cel puțin în SUA, nu ar fi fost anunțat nici un caz credibil de pagube datorate acestui virus.

Unii viruși au fost gîndiți ca mici glume. Astfel așa-numitul "Stoned-Virus", care se ascunde în sectorul de boot al dischetelor sau hard-disk-urilor, afișează pe monitor mesajul: "Your Computer is stoned now. Legalize Marijuana!" Virusul "FuManchu" își mărturisește opiniile politice interactiv: dacă se tastează numele unui anumit om politic, face niște comentarii care nu sînt tocmai pentru urechile copiilor.

După ce calculatorul este pornit din nou cu o combinație de taste, este afișat mesajul: "The world will

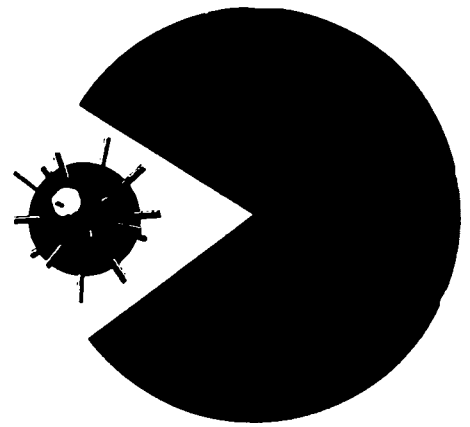
hear from me again!!!". "Oropax" și "8 Tunes" enervează utilizatorul cîntînd fals melodii cunoscute, cum ar fi valsul "Dunărea albastră". Și "Yankee Doodle", cu multiplele sale variante, face parte din această categorie, ca și "Music Bug", un virus al sectorului de boot din cea mai nouă generație.

Ziua de 1 aprilie, Crăciunul, și alte zile importante îi inspiră de asemena pe programatori. Virusul "1 April", de exemplu, după fiecare infestare reușită scrie pe ecran: "You have a virus". În data de 1 aprilie este afișat mesajul "Ha Ha Ha It's April 1st-you have a Virus". Virusul "XA1", o prescurtare de la "X-Mas" (Crăciun) și "A(pril) 1", nu se manifestă numai iarna: de Crăciun este afișat pe monitor un brad de Crăciun cu lumînări. La 1 aprilie, însă, virusul distruge domeniul sistem importante.

De o natură mai ciudată este virusul "Yoshi" care provine din India. Într-o anumită zi din ianuarie, la pornirea sistemului, el cere: "Type Happy Birthday, Yoshi." Dacă dorința îi este îndeplinită, atunci se poate lucra normal mai departe. Probleme pot să apară doar la utilizarea tastaturilor germane, deoarece virusul "nu știe" că tasta "Y" se află în locul tastei "Z" și invers.

Dezvoltarea calculatoarelor de la mașini de calcul pure la aparate multimedia este reflectată și în tematica virușilor. Virusul "Kennedy", face reclamă formației Punk-Rock "The Dead Kennedys", căreia îi urează "viață lungă". Cercetătorii de viruși, americani, au greșit considerînd că este vorba despre o trimitere la familia Kennedy.

O perplexitate asemănătoare a domnit și la apariția virusului "Devil's Dance", care la pornirea "la cald" (warmstart) a calculatorului pune întrebarea "Have you ever danced with the devil under the weak light of the moon?", semnată: "The Joker". Celor cărora li s-a întîmplat să observe virusul, și care



au pierdut filmul "Batman", le-a fost ușor să creadă că este vorba despre ceva mistic. Ca și în cazul celorlalți viruși în care au fost cifrate cuvintele "line after death" (Phoenix), naturile mai sperioase au putut să creadă că acesta este primul virus care "supraviețuiește" opririi calculatorului.

Se pare că cea mai plăcută activitate a programatorilor de viruși este să-i facă pe cercetătorii virușilor să-și piardă siguranța. Astfel există un virus care își modifică în permanență chipul, numit "Whale", și a cărei principală sarcină pare să fie de a fi cît mai greu de analizat. Ceva asemănător se întîmplă și în cazul lui "Fish". Și virusul cu numele "Casper", denumit și "1260", din cauza lungimii sale, a stîrnit vîlvă în cercurile cercetătorilor, deoarece el nu mai conține un șir de caractere constant și devine astfel imun în fața metodelor tradiționale de căutare. Tipătul de spaimă al cercetătorilor nu este justificat, autorului articolului ce a servit ca bază acestei prezentări, Thomas Dehn, nu i-a trebuit mai mult de o oră pentru "a sparge" cifrul lui "1260". Virușii de acest gen pot fi priviți ca farse studentești.

O situație asemănătoare a fost valabilă și în cazul virusului "Tequila", care a fost "descoperit" și catalogat în mai multe locuri din Germania. Și acest virus este cifrat și conține printre altele mesajele: "TEQUILA and BEER forever" și "Loving thoughts to L.I.N.D.A."

Cine a trebuit să tasteze în timpul lucrului la calculator, cele patru cifre care alcătuiesc numerele unei cunoscute ape de colonie germane (4711), a fost victima virusului "Perfume". Și această specie este relativ liniștită și poate crea probleme numai în legătură cu alte soft-uri.

Numele date noilor viruși vorbesc despre fantezia programatorilor, dar și a cercetătorilor, de viruși. Numele lor se aeamănă tot mai mult cu numele jocurilor pe calculator. Astfel există virușii "Invade", "Lamer Exterminator" (Amiga) și "Frodo". Nume ca "Dark Avenger", "Ghost" sau "Disk Killer" sugerează pe bună dreptate anumite pericole. Acești viruși fie șterg date, fie duc la blocarea calculatorului. De remarcat este faptul că "imensele pagube datorate virușilor calculatoarelor", atît de citate în literatură, cel mai adesea sînt cunoscute numai de la a treia mînă. Unii experți consideră că este vorba despre "o panică creată voit pentru a facilita afaceri". Probabil că și în acest caz funcționează veșnicele legi ale zvonurilor: fiecare mai adaugă puțin senzațional, și la a cincea mînă și cele mai nerușinate exagerări par să conțină un simbul de adevăr.

Tehnici de tratament

Virușii calculatoarelor sînt renumiți și temuți. Există aproape tot atîtea programe care oferă protecție, cît și viruși sînt. Vă vom prezenta în continuare zece dintre aceste leacuri bune la toate.

Cînd se vorbește în mijloacele mass-media de calculatoare, adesea dintr-o răsuflare este rostit și cuvîntul virus. Este vorba despre programe care se instalează în calculator și care execută anumite acțiuni pe care utilizatorul nu le poate controla. Cei mai temuți sînt virușii care infectează programele

de aplicație și care șterg sau formatează hardisk-ul.

Programele antivirus promit o protecție simplă și sigură împotriva unor astfel de dăunători. Zece reprezentanți ai acestei categorii de programe, au fost luați în vizor și puși să-și măsoare forțele cu peste 1000 de viruși. Este vorba despre: Antivirus plus, Dr. Solomon's Antivirenwerkzeuge, Iris Antivirus Plus, McAfee Scan & Cleanup, Norton Antivirus, PC Plus Virus Doctor, PCCillin, Turbo Antivirus, Virus Blocker și Virus Finder.

Antivir Plus

"Antivir Plus" (AV+), al firmei H&B EDV, din Tett nang, este programul care și-a protejat prin lege denumirea, și ar fi singurul program de acest fel din Germania care ar avea voie să poarte apelativul "Antivirus". Acest apelativ este utilizat pentru programele care servesc luptei împotriva virușilor pe calculatoare. Aceste programe nu lucrează cum s-ar putea crede, ca un controlavirus, ci localizează virușii mai cunoscuți și/sau îi distrug.

Toate programele antivirus au comun faptul că trebuie să fie în permanență dezvoltate, deoarece ele trebuie să fie la zi pentru a putea proteja și împotriva celor mai noi viruși. Din acest motiv se lucrează și la AV+.

Pachetul nu constă dintr-un singur program ci din trei programe de protecție distincte. Un program rezident în memorie "Avir", care controlează memoria de lucru și care anunță utilizatorul în cazul în care sînt efectuate accese nepermise la date.

Apoi există programul care "caută" și "îndepărtează" virușii numit "Antivir". Acest duo poate fi înlocuit de un program care se mulțumește doar să caute viruși,

numit "Avsearch" și care este distribuit gratuit de către H&B EDV.

Avir este asemănător veteranului shareware "Flushot+", chiar dacă puterea i-a fost ridicată. El verifică și sumele de control. Sarcina principală a acestui program este supravegherea sistemului pentru a observa activitatea unor posibili viruși. Programul TSR dă o multitudine de semnale de alarmă, care în cazul utilizării practice, în birou, pot deveni pătuboase. Căci o operațiune care nu face altceva decît să culeagă texte cu un editor, mai ales la prima avertizare, nu poate descifra semnificația acestor mesaje.

Motivul acestei conștiințiozități exagerate l-ar putea constitui faptul că AV+ a fost conceput nu ca o unealtă pentru lucrul de zi cu zi ci ca o ultimă salvare în caz de necesitate. Astfel acest pachet de programe elimină numai acei viruși despre care există rapoarte credibile privind apariția lor pe PC-uri. Înlăturarea lor este însă cu atît mai conștiințioasă. Un alt indiciu despre acribia celor de la HGB este faptul că, în timpul eliminării virușilor, Antivir indică dacă este posibil ca unul sau altul dintre fișiere să fie mai lung decît originalul, din cauză că anumite erori în programarea virusului (de exemplu la "Vacsina") au dus la suprascierea fără rost a unor octeți și cu aceasta au distrus probabil date.

În lupta pentru memoria de lucru" împotriva virusului "CSFR1000", AV+ s-a dovedit a fi un tovarăș de luptă de încredere. Nici un dăunător nu a avut vreo șansă să se copieze în alte programe.

Virusfinder PC

Virusfinder este, după cum afirmă editura DMV-Verlag din Eschwege, cel mai vîndut program în Germania de acest gen. Textul

imprimat pe partea din spate a ambalajului promite "anunțarea" unui posibil virus "în câteva secunde". Cum diagnostichează programul rămîne secretul producătorului.

Programul este livrat pe o dischetă neprotejată la scriere, împreună cu un manual. Acesta conține o introducere explicită pe tema viruși ca și indicațiile de operare.

Operarea este destul de greoaie, deoarece ecranul este umplut cu un potop de mesaje. Citirea directoarelor este făcută atît de încet încît ai putea crede că s-a blocat calculatorul. Nici viteza de lucru nu este mulțumitoare.

Cea mai mare lipsă a programului o constituie faptul că lipsește autoverificarea. Programul poate fi lansat în execuție și după ce a fost infectat și poate servi chiar el la răspîndirea virușilor.

Principiul de căutare este identificarea unor șiruri de caractere semnificative. Aceste așa-numite "scancodes" (semnături ale virușilor) pot fi specificate și de utilizator. Faptul că în cazul acestei operații codurile de scanare existente pot fi suprascrise, și deci modificate, este o concesie care poate deschide poarta actelor de sabotaj. Înafară de aceasta, cei mai noi viruși, care se autocifrează, scapă de sub control. Așa se întîmplă în cazul virușilor "Omicron", "Fish" și "Tequila".

Virusfinder are și slăbiciuni de concepție. Anumiți viruși sînt recunoscuți doar ocazional. Din 12 fișiere infectate cu "Solano", au fost anunțate ca "bolnave" doar 3. Programul poate fi utilizat pentru căutarea de viruși, dar ca program antivirus și-a ratat de puțin scopul.

Dr. Solomons's Antiviren- werkzeuge

Sonorul nume al acestui pachet de programe nu este găselnița unui expert în marketing ci face trimitere directă la producătorul lui: Dr. Alan Solomon este șeful casei de soft S&S Entreprises din Marea Britanie, specializată în recuperarea datelor. Distribuirea versiunii germane a fost preluată de editura Percomp Verlag din Hamburg.

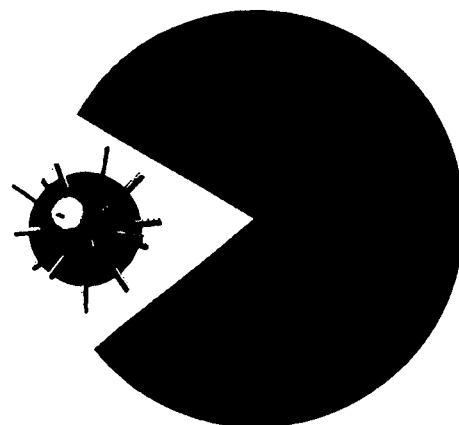
Pozitivă este în primul rînd prezentarea manualului. Utilizatorul este familiarizat cu problematica virușilor. Cu multe exemple și observații, de asemenea în manual fiind prezente și indicații de operare și fiind explicate problemele care pot să apară la utilizarea pachetului.

Instalarea lui se realizează prin simpla copiere de pe dischetă pe harddisk. Soft-ul constă dintr-un program de căutare a virușilor și dintr-o serie de programe de validare a sumelor de control. Operația este preluată de suprafața comună "Toolkit", putînd fi utilizat mouse-ul și avînd la dispoziție un sistem de help-uri bine pus la punct.

Deoarece protecția rezidentă memorie poate fi activată la nivel de program nu este oferită o astfel de protecție on-line. Ca și compromis este oferit programul "Guardian" care verifică programele înainte de lansarea lor în execuție.

La căutarea virușilor cu programul "Findviru" a fost impresionantă viteza de căutare. Chiar și mesajul afișat la descoperirea unui virus "infectat cu siguranță" vorbește despre încrederea în sine a programatorului.

Funcția "eliminarea virușilor", punctul slab al pachetului în urmă cu o



jumătate de an, a devenit între timp o funcție puternică. Chiar și virușii exotici, care nu au fost semnalăți încă în Europa, au fost recunoscuți și îndepărtați corect.

Posibilitatea oferită, de a căuta "mutații" ale unui virus cunoscut, s-a dovedit a fi bine intenționată, dar n-a avut succes. Nu numai că a fost încetinită căutarea, dar nu s-au găsit nici mutații.

Filozofia programului, vizavi de "CSFR 1000" s-a dovedit a nu fi infaibilă. Virusul s-a putut răspîndi copiindu-se în fișierele COM. Abia după o pornire "la rece" a calculatorului, de pe dischetă, programul și-a făcut corect treaba.

Ca program de căutare și de îndepărtare a virușilor Dr. Solomon's Antivirenwerkzeuge, poate fi considerat un program de referință.

Iris Antivirus Plus

Iris Antivirus Plus este un produs al casei soft israeliene Iris. În Germania el este distribuit de firma Hoffmann din Dusseldorf. Ea a preluat și suportul tehnic "hotline".

Pachetul Iris constă dintr-o dischetă de 5,25", neprotejată, și dintr-un manual de cca. 50 de pagini. Acesta descrie modul de funcționare al programului și filozofia sa.

Un program de supraveghere on-line oferă protecție atât împotriva virușilor cunoscuți cât și a celor necunoscuți.

Programul este copiat pe hard-disk cu un program propriu de instalare. Aceasta se întâmplă automat. Fără ca utilizatorul să mai fie interogat, este creat directorul "ANTIVIRUS" și fișierul "AUTO-EXEC. BAT" este modificat pentru protecție on-line. Directorul creat este protejat la scriere, astfel încât utilizatorii pot avea dureri de cap dacă vor să-l mute în altă parte, sau să-l ștergă (trebuie utilizată comanda "attrib"). Căutarea virușilor este realizată elegant și conștient. La eliminarea virușilor cunoscuți, Iris s-a descurcat bine.

În cursul instalării programului, fișierele sistem (IO.SYS, MSDOS.SYS), ca și sectorul de boot, sînt copiate pe o dischetă, și sînt făcute copii și pe harddisk. În caz de distrugere, ele pot fi reconstituite.

De neînțeles rămîne, de ce Iris preia programele executabile fără o supraveghere generală a surselor de control. Încrederea nelimitată acordată programului de supraveghere s-a dovedit a fi o eroare fatală. În duelul direct cu virusul de test, Iris a constatat infectarea memoriei și a crezut că a rezolvat eficient îndepărtarea virusului. Ștergerea programului infectat a fost făcută prea tîrziu. La o nouă pornire a sistemului s-a dovedit că interpretorul de comenzi COMMAND.COM era infectat. Ca urmare sistemul s-a blocat. Deoarece era vorba despre un virus necunoscut, nu se știa unde trebuia căutată cauza: Iris a deblocat cu succes, totuși, încercarea de infectare.

După o modificare a fișierului "AUTOEXEC.BAT", în care Iris a fost dezactivat, s-a dat ocazia virusului "CSFR 1000", ca în numai cîteva minute, să infecteze nu mai puțin de 112 fișiere. Din cauza absenței unei funcții de validare a su-

mei de control, nu s-a putut stabili care fișiere erau infectate. Utilizatorului nu i-a rămas altă soluție decît să restaureze în întregime colecția de programe de pe dischete.

Utilizat împotriva unor variații necunoscute ale unor viruși "vechi", Iris s-a descurcat, dar faptul că programul oferă siguranță și împotriva virușilor care vor veni, și că deci nu mai are nevoie de actualizări este doar un slogan publicitar.

În ciuda lipsurilor, Iris este un produs stabil. În orice caz, nu strică să fie completat cu un program de validare a surselor de control.

Norton Antivirus

Numele "Peter Norton" este valabil ca un certificat de calitate în rîndurile utilizatorilor PC. Cu "Norton Antivirus" (NAV), renumita casă de soft își face intrarea și pe piața programelor antivirus.

Manualul explică funcțiile și modul de operare al programului. Introducerea în problematica virușilor este limitată la remarcă: "un virus este un program care determină calculatorul D-vs să reacționeze altfel decît sînteți obișnuit". După această definiție aproape orice actualizare a unui program este un "virus".

Furnitura de livrare conține două dischete de 5,25" și respectiv 3,5" și un manual. Instalarea este realizată automat, la fel ca și căutarea unor eventuali viruși. Viteza de execuție este remarcabilă.

În cazul în care sînt depistați viruși ei pot fi parțial înlăturați. Spre deosebire de alte programe, cuprinse în acest test, un fișier infectat cu mai mulți viruși nu poate fi curățat la o singură trecere. Acest aspect, deși este bine documentat, poate duce, în cazul unor utilizatori cărora le-a fost lene să citească documentația, la erori de

operare. Norton Computing poate aduce îmbunătățiri în acest caz.

Protecția, în continuare, este realizată cu programul rezident în memorie "Interceptor". El necesită cca. 24 KByte RAM. Puterea lui se limitează însă la verificarea infectării programelor cu virușii cunoscuți.

În cazul unor noi viruși, programul s-a făcut remarcat doar prin încetinirea accesului la harddisk. Dubios pare faptul că driver-ul NAV.SYS al lui Interceptor, a fost modificat, fapt care ar fi trebuit să fie un semn clar al activității virușilor.

Pentru depistarea unor noi dăunători există o funcție de verificare a sursei de control. Pentru fiecare fișier COM, EXE și SYS, este depus, în același director, un fișier cu numele de control, cu atributele sistem "Read Only, Hidden, Sys". Aceasta duce însă la o enormă risipă de spațiu (cca. 200 KByte pentru 400 de fișiere program).

În plus această funcție poate produce neplăceri ce nu pot fi înlăturate. Astfel optimizarea harddisk-ului, cu programul aceeași case de soft, "Speed Disk" are sens numai cu ștergerea în prealabil a fișierelor cu sursele de control și refacerea lor după optimizare. Chiar și mutarea programelor în alte directoare implică măsuri complexe, deoarece în acest caz cele două surse de control nu vor mai corespunde.

În testul practic, NAV s-a dovedit neajutorat în fața unor viruși noi. Nu sînt remarcate nici modificările surselor de control, nici activitățile virușilor în memorie și nici măcar nu sînt semnalate încercările de modificare ale programelor.

Paleta de funcțiuni a programului se limitează la căutarea și îndepărtarea virușilor cunoscuți în SUA. Deoarece multe variante eu-

ropene nu sînt recunoscute, NAV nu asigură o protecție satisfăcătoare.

PCCillin

PCCillin (PCC) crează impresia unei protecții profesionale împotriva virușilor. Din furnitura de livrare, pe lângă programul însuși, mai face parte și un dongle. Acesta este o placă de interfață, ale cărui informații interne sînt puse la dispoziția programului, și fără de care el nu poate funcționa. El servește, după datele producătorului, în primul rînd pentru memorarea unor informații despre sectorul de boot.

La o primă instalare, automată, reușită, conținutul acestor sectoare este scris în EEPROM (Electrically Erasable Programmable Read Only Memory), care este inclusă în dongle.

La pornirea sistemului, programul verifică dacă nu s-a modificat configurația memorată la prima instalare. Dacă s-a modificat ceva atunci este afișat un mesaj. PCC trebuie să fie primul program apelat în AUTOEXEC.BAT.

Căutarea virușilor cunoscuți, la instalare, are loc cu o viteză acceptabilă, numai că unii dintre cei mai răspîndiți viruși nu sînt depistați. În plus PCCillin nu îndepărtează dăunătorii din programe.

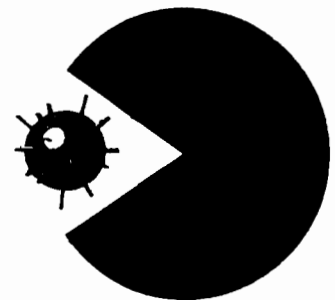
Manualul, pur informativ, face o introducere competentă și explicată în problematica virușilor și lămurește faptul că PCC este conceput mai puțin pentru căutarea de viruși cît mai ales ca program de prevenire a infectării cu viruși. Se pornește deci de la faptul că "starea normală" a unui PC este fără viruși și PCC trebuie să împiedice virușii cunoscuți și necunoscuți să pătrundă după mecanismele cunoscute și/sau imaginate în sistem.

Programul rezident în memorie supraveghează toate activitățile sistemului și în cazul unor activități dubioase oprește sistemul fără a-i mai lăsa utilizatorului nici o șansă. După aceasta sistemul nu mai poate fi pornit decît "la rece" (cu butonul de reset, sau cu comutatorul de alimentare). Metoda se dovedește a fi de încredere. Doar proaspătului venit pe piață Tequila i-a reușit să modifice conținutul harddisk-ului după propria plăcere. Abia la următorul boot, a fost observată modificarea și a fost refăcută. Dacă această acțiune a fost împiedicată, virusul s-a putut răspîndi în voie. Numai la părăsirea unui program infectat a observat PCC că ceva nu este în ordine. Totuși este discutabil dacă năzbîțiile celui care a făcut testul pot fi puse în circa programului. La instalarea pe un PC infectat apar aceleași probleme și producătorul va trebui să țină cont de acest lucru.

Supărătoare este înclinația lui PCC de a da mesaje de eroare. Pe un 386 programele TSR (Terminate-and-Stay-Resident) și drivele încărcate în zona de memorie cuprinsă între 640 KByte și 1 MByte, sînt văzute ca activități ale unor viruși și sistemul este blocat automat. Același lucru se întîmplă și cu compilatorul Turbo-Pascal, care utilizează metode proprii de tratare a memoriei.

Pentru eliminarea acestor efecte secundare, programul poate fi dezactivat cu o combinație de taste, dar atunci dispăre și protecția. Negativ este și faptul că PCC nu permite stabilirea gradului de infecție.

Pertotal PCC asigură o protecție de nădejde atît împotriva virușilor cunoscuți cît și a celor necunoscuți. Pentru "dezinfectarea" unor sisteme infectate, nu este nici gîndit, nici indicat. În ceea ce privește mesajele de eroare, producătorul mai poate aduce îmbunătățiri.



PC Plus Virus Doctor

Întreprinderea PC Plus GmbH, din Munchen, a lansat pe piață pachetul "Virus Doctor" care constă dintr-un program de căutare a virușilor, unul pentru îndepărtarea lor și dintr-un supraveghetor al memoriei numit "Guard". Instalarea se realizează prin simpla copiere a fișierelor pe harddisk. Guard poate fi instalat ulterior, dar se poate și autoinsera în CONFIG.SYS.

Manualul nu este voluminos, dar este suficient pentru operarea programului. Temele "Generalități despre viruși" și "Ce trebuie făcut în cazuri grave?" sînt totuși prea scurte.

După datele producătorului Virus Doctor este o versiune restrînsă a pachetului realizat în aceeași casă Virus Professor, gîndit pentru a oferi ajutor și protecție utilizatorilor fără cunoștințe PC avansate. Programul principal Virus Doctor oferă ca funcțiuni verificarea memoriei, căutarea și înlăturarea de viruși ca și cîteva posibilități de instalare pentru Guard. El nu poate însă să construiască sume de control și nici să creeze un fișier de pornire, care ar permite obsevarea apariției virușilor.

Virus Doctor caută toți virușii pe care-i cunoaște, sau numai pe aceia pe care îi poate și îndepărta. Căutarea după șiruri de caractere

(scancode) s-a dovedit sensibil mai lentă decât căutarea unor modele de biți cunoscute.

Programul de supraveghere Guard este inserat în CONFIG.SYS și devine activ încă înainte de încărcarea procesorului de comenzi. Din punctul de vedere al virușilor sectorului de boot există cel puțin teoretic posibilitatea controlării în întregime a sistemului.

În lupta împotriva virusului "CSFR 1000" (unul din cei mai noi), s-a dovedit că acesta a fost blocat cu succes. Nu a trebuit să ne temem de o răspîndire în continuare a acestuia.

Din punctul de vedere al mesajelor Guard prezintă câteva lipsuri. Modificarea sectorului de boot a rămas necomentată. Tocmai astfel de evenimente nedorite sînt, mai ales la partiționarea unui hardisk, un semn de alarmare. Produsul va trebui îmbunătățit la acest capitol.

Virus Doctor este puțin folositor la recunoașterea și îndepărtarea virușilor. Măsurile de protecție sînt realizate în primul rînd de programul de supraveghere Guard, care se descurcă mulțumitor atît timp cît nu apar viruși ai sectorului de boot.

PC Plus a anunțat modificări consistente în program, care să-i crească puterea. Pînă atunci însă programul trebuie să se mulțumească cu un loc în clasa de mijloc.

McAfee Scan

"Viruscan" al lui John McAfee este considerat un program de căutare a virușilor, pur și simplu pentru faptul că autorul programului este unul din cei mai cunoscuți experți în virusologie. Faptul că este atît de cunoscut nu prezintă însă o garanție în sine a calității programului. Din acest motiv a fost testat

și pachetul său antivirus, livrat ca produs shareware.

A fost testată versiunea 75C a lui Viruscan (Scan). Programul de îndepărtare a virușilor "Cleanup" purta numărul de versiune 75. În momentul în care apare acest articol va fi disponibilă deja versiunea V77, dacă nu chiar o versiune cu un număr mai mare. Suprinzătoare este frecvența cu care este actualizat programul. Producătorul explică aceasta prin faptul că în permanență apar noi viruși pe piață, care trebuie recunoscuți de program și împotriva cărora trebuie să poată lupta.

În documentația livrată pe dischetă se afirmă că sînt recunoscuți 480 de viruși. Cifra este exagerată puțin, deoarece parcurgîndu-se cu atenție lista se pot observa cîțiva viruși care au fost numărați de două ori. Mai rămîn totuși cca. 450 de viruși.

În afară de căutarea și îndepărtarea virușilor, de cîtva timp, McAfee oferă și o funcție de verificare a sumelor de control. Această funcție adaugă la fiecare fișier executabil 10 Byte în care se înregistrează informații pe baza cărora se poate depista ulterior o posibilă infecție. Dezavantajos este faptul că Scan nu face cunoscut faptul că a adăugat aceste informații, utilizatorul putînd interpreta greșit modificarea lungimilor fișierelor respective.

Acest fapt s-a dovedit la prima încercare de a "injecta" programe "curate". Copiile fișierului Command.com de pe lectorul de dischete verificat, după adăugirea sumelor de control au fost recunoscute ca "modificate". Acest fapt nu s-a modificat nici după cel de-al doilea control. Ca și în cazul anterior, Scan a crezut că a descoperit activitatea unor viruși. Deoarece și celelalte programe, din furnitura McAfee, au fost recunoscute ca "modificate" faptul stîrnește neîncredere. Neîncrederea a sporit atunci cînd Scan a raportat două

fișiere nevinovate ca fiind infectate cu "Icelandic" și respectiv cu "170X". Scuza că programul a depistat codurile de scanare tocmai în aceste fișiere nu funcționează, deoarece alte programe de căutare a virușilor nu dau alarme false.

Aceste motive de neîncredere au sporit în cazul programului "Cleanup". Remarca din documentație, conform căreia orice virus recunoscut, îl poate și îndepărta s-a dovedit a fi doar un slogan publicitar. Reparațiile programelor, în majoritatea cazurilor, constă nu în refacerea programelor infectate ci în ștergerea lor. Cînd programul se apucă de reparat, mai mult strică.

Împotriva virusului "CSFR 1000" programul s-a dovedit a fi neajutorat, desigur pentru faptul că acest virus este nou și nu este recunoscut de versiunea care a fost testată. Viruscan dă atît de des false mesaje de eroare încît nici cu ajutorul funcției de validare a sumelor de control nu se poate stabili răspîndirea infecției.

Este de presupus faptul că Scan și-a asigurat un nume atît de bun, tocmai din cauză că unii utilizatori au luat de bune aceste false mesaje de alarmă și le-au considerat o dovadă a "conștiinciozității" cu care lucrează programul, virușii anunțați nefiind oricum prezenți. În comparație cu partenerii de întrecere, pachetul de programe McAfee mai are mult de recuperat.

Turbo Antivirus

Programul "Turbo Antivirus" al producătorului israelian Carmel Software Engineering, este distribuit în Germania de către firma muncheneză EPG Software. Manualul de 75 de pagini conține pe lîngă o introducere în problematica virușilor și sfaturi pentru operarea în rețea, răspunsuri la întrebările cele mai frecvente, care apar în

asistența hotline, ca și descrierea celor mai cunoscuți viruși.

Instalarea se face fără probleme cu ajutorul unui program livrat în pachet. Programul principal "Tntvirus.exe" cuprinde funcții de căutare și de îndepărtare a virușilor, de calculare a sumelor de control ca și de "imuzizare" a fișierelor program.

Surprinzătoarea viteză la stabilirea și compararea sumelor de control se datorează faptului că în examinare sînt luate numai anumite secțiuni ale fișierelor. Cu toate acestea, după cum afirmă programatorul Eli Shapira, metoda poate determina cu certitudine un posibil atac. Informațiile cu sumele de control sînt depuse în cîte un fișier pentru fiecare director. Starea sectorului de boot poate fi păstrată intactă, și la nevoie poate fi refăcută cu ajutorul programului "Bootsafe.exe".

Imunizarea este realizată cu ajutorul unui program care se așează "ca o coajă" în jurul programului de protejat. Fiecare program astfel imunizat este verificat înainte de lansarea sa în execuție și în cazul în care acesta este atacat de un virus se dă un mesaj de eroare.

Ca programe de supraveghere sînt oferite programele "Defender" și "Tsafe". Defender verifică la încărcarea unui program dacă acesta conține vreun virus cunoscut. Dacă este depistat ceva, execuția se întrerupe. Tsafe supraveghează în plus ca și activitățile virulente ale unor viruși încă necunoscuți să fie recunoscute și întrerupte.

Pentru testul practic, mai întîi a fost lăsat virusul "CSFR 1000" să se răspîndească pe calculatorul de test. Apoi a fost pus la treabă Turbo Antivirus. Programul trebuia să depisteze, pe baza sumelor de control, care fișiere erau infectate. După căutarea în numai patru subdirectoare, programul a raportat că și-a terminat căutarea și n-a mai

putut fi urnit nicicum pentru a căuta pe tot harddisk-ul. Spre cinstea programului trebuie să mărturisim însă că la majoritatea fișierelor imunizate nu numai că a fost observată modificarea, dar a fost refăcută și starea lor inițială.

Programul caută virușii într-un ritm susținut. La repararea unor fișiere infectate pot să apară însă probleme: fie se blochează sistemul, fie fișierele nu zînt reparate corect. După afirmațiile distribuitorului, lipsurile observate vor fi ridicate în curînd.

Virus Blocker

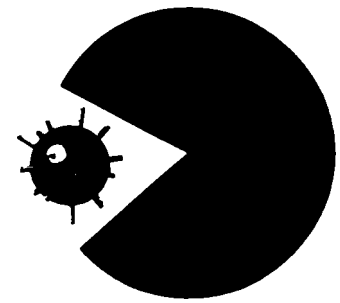
"Virus Blocker" al firmei Expert Informatik din Berlin, este gîndit mai mult ca protecție în cazul unor atacuri cu viruși și mai puțin pentru observarea virușilor. La dorință, programul permite și blocarea execuției unor programe nedorite.

Virus Blocker este livrat împreună cu un dongle, o interfață a cărei electronică conține informații fără de care programul nu poate funcționa. El servește în primul rînd ca metodă de protecție la copiere.

După instalare programul stabilește starea soft-urilor existente calculîndu-se sumele de control și memorîndu-le. De neînțeles este faptul că programul nu face mai întîi o verificare automată a sistemului pentru a vedea dacă nu sînt prezenți deja anumiți viruși.

Stabilirea sumelor de control este incomodă. Pentru fiecare tip de fișier (cum ar fi *.COM, *.EXE sau *.SYS) programul trebuie lansat din nou. Din păcate acest lucru se întîmplă și atunci cînd se dorește verificarea faptului că anumite fișiere au fost sau nu modificate.

Acest aspect este cu atît mai supărător cu cît producătorul recomandă o verificare zilnică. În plus



Virus Blocker lucrează atît de încet încît utilizatorii ar putea fi tentați să întrerupă modificarea. Programul nu permite însă să fie întrerupt. În plus Virus Blocker caută atît de încet virușii sectorului de boot încît mulți utilizatori vor fi tentați să renunțe la această funcție.

Ca alte măsuri de protecție sînt oferite programe care împiedică formatarea suporturilor de date, încărcarea de programe rezidente sau chiar scrierea pe harddisk. Unul din aceste programe permite crearea unui "fișier de logare", în care sînt înregistrate toate apelurile de programe, activitățile lor în memorie, vectorii de întrerupere utilizați ca și modul lor de terminare.

În testul împotriva virusului "CSFR 1000", Virus Blocker a determinat mai întîi căderea sistemului. La o nouă pornire a calculatorului s-a dovedit că fișierul COMMAND.COM era deja infectat. Toate programele .COM apelate din AUTOEXEC.BAT au fost infectate și ele. Nici după instalarea programului de supraveghere on-line, infecția nu a putut fi stopată. Fișierele infectate au putut fi localizate după sumele de control modificate.

Rezumînd, Virus Blocker poate fi recomandat, cu anumite rezerve, ca sistem de protecție. Programul nu excelează în căutarea virușilor.

(R.M.)

Tabel comparativ

Produs	Antivir Plus	Virusfinder PC	Dr. Solomon's Antivirenwerkzeuge	Iris Antivirus +	Norton Antivirus
Producător	H&BE dv	DMV Verlag	Percomp Verlag	- Hoffman	Symantec/ Access
Preț (DM)	198	149	360	298	295
Furnitura de livrare					
Manual		*			
Dischete					Dischete
Necesar RAM pt. partea rezidentă	14 KByte	9 KByte	4 KByte	22 KByte	24 KByte
Instalare program	Auto	Auto	Auto	Auto	Auto
Mod de lucru					
Elimină	da	nu	da	da	da
Sume de control					
Stabilește	(Opțiune TSR)	-	47 s	-	140 s
Verifică	(Opțiune TSR)	-	50 s	-	108 s
Caracteristici de operare					
Suprafață utilizator grafică	-	*	*	*	SAA
Operare cu mouse	-	-	*	-	-
Documentație					
Manual	*	*	*	*	*
Fișier	*	*	*	*	*
Help on-line	-	-	*	-	*
Rezultate de test (pe cei mai răspândiți 25 de virusi)					
Găsiți	146	123	151	165	148
Indepărtați	99	0	140	164	53
Distruse la reparare	0	-	0	0	0
Total					
Găsiți	926	726	877	912	920
Sume de control					
Stabilire	(Opțiune TSR)	-	47 s	-	140 s
Verificare	(Opțiune TSR)	-	50 s	-	108 s
Căutare pe un disc curat	78	452	42	100	63
Alarmer false	-	*	*	-	*
Fișier raport	*	*	*	*	*
Protecție on-line	*	-	-	*	
Interactiv	*	*	-	*	*
Tip (TSR/Device Driver)	TSR	-	DRV	TSR	DRV
Căutare virusi rezidentă	-	*	*		*
Poziție în clasament	4	8	1	2	5

Tabel comparativ

Produs	PCCillin	PC+ Virus Doktor	Scan & Cleanup	Turbo Antivirus	Virusblocker
Producător	Trend Micro	PC Plus GmbH	McAfee Ass.	EPG, Haar	Expert Informatik
Preț	cca. 380	430	18	298	480
Furnitura de livrare					
Manual			Manual ca fișier		
Dischete					Hardlock-Dongle
Necesar RAM pt. partea rezidentă	14 KByte	11 KByte	37 KByte	16/8 KByte	2 - 10 KByte
Instalare program	Auto	Copiere	Copiere	Auto	Auto
Mod de lucru					
Elimină	nu	da	da	da	nu
Sume de control					
Stabilește	-	-	405 s	45 s	706 s
Verifică	-	-	405 s	40 s	653 s
Caracteristici de operare					
Suprafață utilizator grafică	-	-	-	*	-
Operare cu mouse	-	-	-	-	-
Documentație					
Manual	*	*	*	*	
Fișier	*	*	*	*	*
Help on-line	-	-	-	-	-
Rezultate de test (pe cei mai răspândiți 25 de viruși)					
Găsiți	111	99	149	138	138
Indepărtați	0	51	88	125	0
Distruse la reparare	0	0	8	3	0
Total					
Găsiți	523	844	806	781	856
Sume de control					
Stabilire	-	-	405 s	45 s	706 s
Verificare	-	-	405 s	40 s	653 s
Căutare pe un disc curat	260	130	198	190	310
Alarmer false	-	*	*	-	*
Fișier raport	-	-	*	*	*
Protecție on-line	*	*	-	*	*
Interactiv	-	?	*	*	*
Tip (TSR/Device Driver)	TSR	DRV	TSR	TSR	TSR
Căutare viruși rezidentă		-	*	*	-
Poziție în clasament	9-10	6	7	3	9-10

Multitasking

Cel tîrziu de cînd a apărut Windows 3.0, noțiunea de "multitasking" este în vogă. Pe măsură ce discuțiile avansează, ceața se îndesește. Iată de ce am considerat necesară explicarea acestei tehnici, precum și a unor noțiuni ca "preemption" sau "deadlock".

În general, prin "multitasking" se înțelege executarea simultană a mai multor sarcini - "taskuri". Traducerea corectă spune doar că se execută mai multe taskuri - nici o vorbă despre simultaneitate.

Această ambiguitate își are soriginea în faptul că PC-urile uzuale sînt calculatoare cu un singur procesor, deci pot executa la un moment dat un singur program (task). O prelucrare paralelă a mai multor taskuri presupune însă existența mai multor procesoare.

Cu toate acestea, prin "multitasking" - la sistemele de operare de pe PC-uri - se înțelege tocmai acest paralelism tehnic imposibil. Și acest lucru, s-ar putea crede că se întîmplă cînd de sub "Windows 3.0" sau "Geoworks Ensemble" se pot derula mai multe programe aparent simultan. Numai aparent: pe calculatoare cu un singur procesor, multitasking-ul nu este decît o "iluzie optică".

Principiul ce stă la baza acestui efect se poate ilustra cu exemplul unui zidar care lucrează la un șir de case: de la ora 9 la 10 lucrează la casa A, de la ora 10 la 11 la casa B, de la 11 la 12 la casa C, apoi din nou la casa A și așa mai departe -

pentru spectatorul accidental, cele trei case par să se ridice simultan.

Tocmai această strategie o aplică și sistemele multitasking. Singura deosebire: schimbarea "locului de muncă" nu se face odată pe oră, ci, de regulă, cam la 0,05 secunde. Ca și în viața reală, și calculatorului îi trebuie un timp pentru a ajunge de la o casă la alta. Și chiar dacă sistemele de operare bune reușesc să facă extrem de repede trecerea, un lucru e sigur: totdeauna, în multitasking, un program se derulează mai încet decît dacă procesorul i-ar aparține în exclusivitate.

De ce este nevoie de această tehnică pretențioasă? Uneori, nu se poate altfel. Există, de exemplu, sistemele "multiuser" - constelații deci, la care mai mulți utilizatori, de la terminale diferite, lucrează pe același calculator. În astfel de cazuri, multitasking-ul este un imperativ - fiecare vrea să ajungă la rînd.

În sistemele cu un singur utilizator în schimb - și ele formează marea majoritate a PC-urilor instalate - necesitatea multitasking-ului nu mai e deloc atît de evidentă. Bietul utilizator nu poate introduce date cu mîna stîngă, să lucreze cu dreapta în contabilitate și să miște în același timp mouse-ul cu cotul pentru a produce eventual un nou logo al firmei dintr-un program de grafică. Și într-adevăr, nu este simplă găsirea unor aplicații. Cu toate acestea, cîteva există: așa-numitul "spooler" de exemplu, permite o ușurare semnificativă a

lucrului. El face posibilă tipărirea unui document în "background", în timp ce se lucrează deja la textul următor. O astfel de funcție - la fel ca un ceas de exemplu, afișat într-un colț al ecranului - reprezintă deja o formă simplă de multitasking: pornind de la un impuls periodic (întreruperea de timer) programul curent este mereu întrerupt. Controlorul procesorului trece atunci la un subprogram. Acesta trimite atunci cîteva caractere de text spre imprimantă sau actualizează afișajul de oră, redînd apoi controlul programului principal.

Astfel de subprograme se găsesc pe mai toate PC-urile, sub forma programelor rezidente (așa-numitele "TSR"-uri). Marele dezavantaj al acestora constă în faptul că un program întrerupe un altul totdeauna ritmic. Sistemele de operare moderne oferă de aceea un mecanism, în centrul căruia stă un "switcher" (comutator). Acesta permite păstrarea simultană în memorie a mai multor programe, trecerea de la unul la altul făcîndu-se numai la solicitări (prin "hotkey" = "tastă fierbinte", combinație de taste sau click pe mouse).

Astfel de exemple dintr-un program de grafică se poate lucra la o ilustrație, pentru a comuta apoi în prelucrarea de texte și a o insera acolo. O zonă de memorie comună - numită "clipboard" - face posibil acest transfer de date. Pentru ca acest lucru să funcționeze, ambele programe trebuie să fie scrise special pentru acest mediu. și acest lucru din două motive:

- - un transfer de date poate funcționa numai dacă ambele programe "văd" același format al datelor în clipboard
- - ambele programe trebuie să aibă loc simultan în memorie. Deoarece, din motive de spațiu, deseori acest lucru nu este posibil, este treaba programatorului să pregătească programele pen-

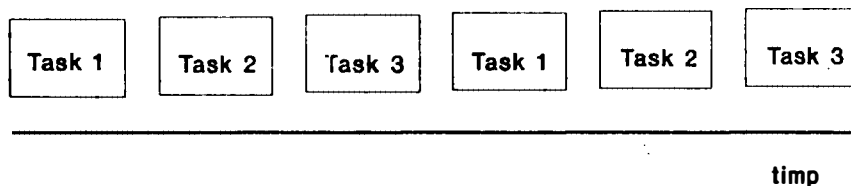


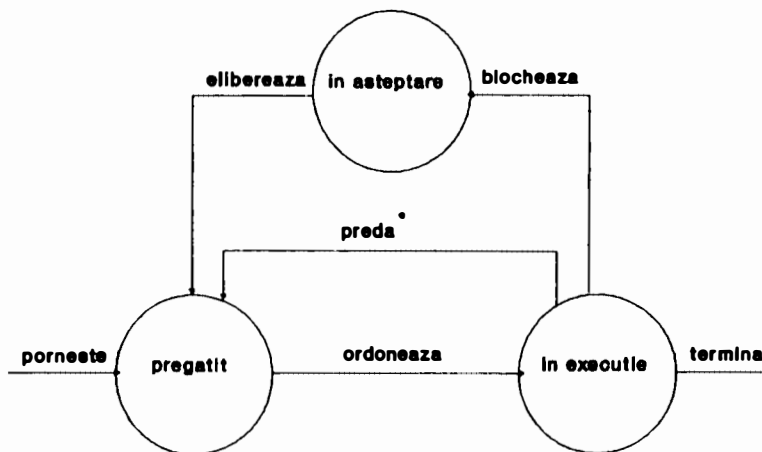
Fig. 1. Fiecărui task îi este alocată aceeași cuantă de timp

tru a putea fi rulate în mediul switcher-ului, deci să le "mărunțească" în multe fragmente mici, cărora să le atribuie clasificări de genul "poate fi temporar depozitat pe disc" sau "trebuie să rămână tot timpul în memorie". Dacă această premisă nu este îndeplinită, trebuie evacuat, respectiv restaurat, întotdeauna întreg programul pe harddisk - ceea ce durează foarte mult.

Cu o prelucrare paralelă - chiar aparentă - acest lucru încă nu are prea multe în comun. Deci între sistemele cu switcher și cele de multitasking mai există o diferență: comutarea de task-uri nu se face "de mână", ci automat, prin sistemul de operare. Se pune întrebarea: când?

Figura 1 prezintă un sistem în care la 3 taskuri li se alocă pe rând o cantitate de timp identică. Pe cât de onest poate să pară acest lucru la prima vedere, pe atât este de stupid. Un program de prelucrare texte este ocupat mai ales să aștepte tastări de la operator. Oricât de iute ar tasta un operator uman, la timp de comutare taskuri de ordinul milisecundelor, între două tastări trec mici veșnicii. Deci se risipește prețiosul timp al procesorului.

De aceea în general se are grijă ca unui task să i se aloce timp calculator numai atunci când nu așteaptă un "eveniment" - o apăsare de exemplu sau o mișcare a mouse-ului. Premisă necesară pentru acest lucru este desigur ca sistemul de operare să știe când un task așteaptă un eveniment. Sub Windows de exemplu, această problemă se rezolvă prin faptul că programele aplicative interoghează totdeauna sistemul de operare asupra producerii unui eveniment. Astfel, Windows cunoaște deja, în mod necesar, o stare a unui program: "în așteptare". În afară de aceasta, un task se mai poate găsi în stările "pregătit" sau "în execuție". O diagramă de stări corespunzătoare e prezentată în figura 2. "În execuție" este programul care tocmai are procesorul,



* - intervenția sistemului de operare la programe a
caror execuție durează mult

Fig. 2. Programele mari consumatoare de timp sînt readuse de către sistemul de operare în starea "pregătit".

"pregătit" este cel care ar vrea să-l obțină.

Comutarea efectivă a taskurilor o rezolvă o rutină sistem numită "dispecher" (distribuitor). Aceasta oprește un task și salvează toate

datele sale "personale", toți regiștri procesorului. Apoi încarcă datele salvate ale următorului task și îl lasă să continue de unde a fost întrerupt.

Cînd anume intervine dispecherul, este o decizie a altei funcții de nucleu, pe nume "scheduler". Acest "planificator" decide, care task, cînd ajunge la rînd și cîte cuante de timp să i se aloce. Decizia se ia cam în felul următor: generatorul de tact al scheduler-ului este deja amintită întrerupere de timer, declanșată de hard de 18,2 ori într-o secundă. Aceasta determină o unitate de timp de 0,055 secunde. O parte din acest timp, scheduler-ul îl consumă el însuși, deoarece trebuie să se uite dacă unitățile de timp alocate unui task s-au consumat deja. Ceea ce se rezolvă prin intermediul unui contor de ceas. Situația devine problematică abia cînd se execută mai multe taskuri. Pentru a păstra privirea de ansamblu, scheduler-ul lucrează de regulă cu liste, care în plus pot fi ordonate după priorități. În cel mai simplu caz există numai o singură astfel de listă. Schedulerul știe, care este taskul care tocmai se execută, și se uită la următorul. Dacă acesta este "în așteptare", atunci trece la următorul task și tot așa mai departe, pînă ce găsește unul care este "pregătit".

Chiar genială nu este această strategie, deoarece parcurgerea

Procese și taskuri

Două din noțiunile deseori confundate, relativ la sistemele de operare, sînt cele de "proces" și "task".

Un task este un subprogram care este "pornit" de un program principal, numit "inițiator". Prin urmare, el se execută eventual împreună cu alte taskuri în mediul inițiatorului. Că acest lucru poate duce ușor la conflicte, este evident: procesul, memoria și alte resurse trebuie folosite în comun.

Un proces în schimb este un program care se poate comporta de așa manieră de parcă ar dispune singur de procesor. Nu trebuie să țină seama în nici un fel de alte procese, și deci poate lansa și taskuri. Numai resurse precum ecranul sau imprimanta trebuie partajate.

Funcțiile de bază ale unui sistem multitasking nu sînt deci task-uri, ci, mai degrabă, procese.

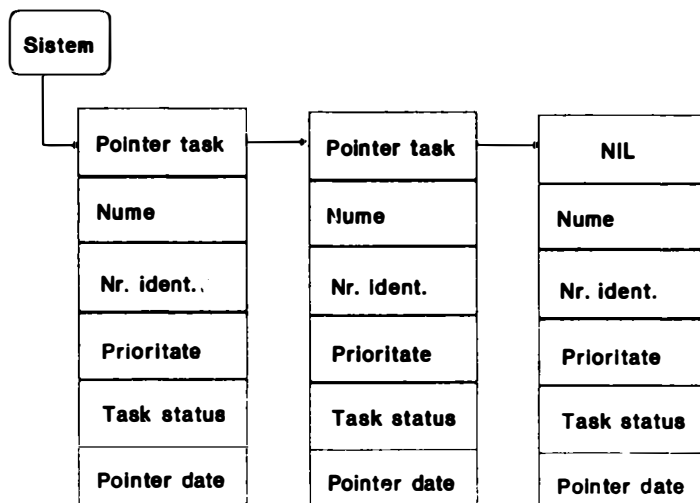


Fig. 3. Toate informațiile despre un task sînt conținute în "Task Control Block" (TCB)

listelor costă timp. Aceeași pierdere de timp se produce la apariția unui eveniment: în acest caz, lista trebuie parcursă pentru a afla care este taskul (sau care sînt taskurile) care îl aștepta. Acesta e trecut atunci din starea "în așteptare", în starea "pregătit".

Pentru a evita aceste întâzieri, unele sisteme gestionează liste separate pentru taskurile "în așteptare", "pregătite" și "active" (ultima fiind interesantă desigur numai în cazul sistemelor multiprocesor). Procedura descrisă se referă la o strategie secvențială: toate taskurile ajung pe rînd să lucreze, chiar dacă pentru intervale de timp diferite. Cînd s-a ajuns la ultimul, se continuă din nou de la primul - lista de taskuri este o listă circulară. Procedeu poartă numele de "round robin". Nucleul acestuia este așa-numita "preemption" - dislocare, înlocuire. Un task îl înlocuiește pe altul, de cum i s-a epuizat timpul afectat.

Contraponderea acestei strategii o reprezintă așa-numita "strategie de urgență". În acest caz, lista de taskuri este sortată după priorități. Adică: un task este lăsat să ruleze, următorul pornește abia cînd predecesorul lui a terminat. În acest caz există două feluri de dislocări (preemption):

- - dacă apare un task nou, de prioritate mai mare decît cel în curs, atunci acesta înlocuiește

task-ul care se execută. Există însă și modele, în care acest lucru nu este permis. În cazul acestora, task-ul este așezat la coada listei. Algoritmul se numește "FCFS" (first come, first served) sau "HOL" (head of line).

- - chiar și într-un sistem ghidat prin priorități, un task nu poate fi neîntreruptibil: sistemul de operare intervine regulat, pentru a afla dacă au apărut task-uri noi. Deseori, se ține și un fel de statistică, care poate duce la o redistribuire dinamică a priorităților. Motivul este simplu: dacă sistemul de operare constată că un task se găsește mereu în starea "în așteptare", înseamnă că acesta nu se grăbește prea mult. Deci i se reduce prioritatea, și el se va lăsa înlocuit de alt pretendent.

Nu așa se întîmplă în sistemele "non-preemptive": sub GEM, Windows 2.1 și în mare parte și sub Windows 3.0, în cazul normal se execută un singur task, cel care corespunde ferestrei active. Niemi și nimic nu îl poate înlocui - doar dacă task-ul întrebă sistemul de operare dacă nu cumva s-a produs vreun eveniment. De voie, de nevoie, trebuie să facă acest lucru destul de regulat, căci altfel nu află niciodată dacă utilizatorul a selectat un meniu sau a apăsător un tastă. Apelurile de funcție necesare pentru aceasta redau în mod automat controlul sistemului de operare, care categorisește taskul ca fiind

"în așteptare" și își îndreaptă atenția spre următorul task care poate fi de exemplu ceasul afișat. Acesta s-a interesat dacă s-a mai produs un eveniment tip timer - o întrebare de ceas. Dacă da, e servit cu promptitudine, deci el citește ceasul sistemului, desenează noua poziție a limbilor ceasului pe ecran și întreabă despre un nou eveniment relevant - moment în care, din nou, controlul trece la sistemul de operare.

Acest procedeu non-preemptiv este considerat deseori un minus (de altfel particula implică lipsa a ceva). Lucrul deseori trecut cu vederea, este simplu fapt că pentru sisteme cu un singur utilizator, această strategie este aproape ideală. Căci utilizatorul individual lucrează de regulă cu un singur program. Și de ce un program izolat să fie mereu întrerupt de sistemul de operare, numai și numai ca acesta să constate că nu mai există și altul care să vrea să ajungă la rînd?

Renunțarea la astfel de întreruperi inutile nu aduce numai avantaje în ce privește viteza de execuție, dar simplifică și conlucrarea diverselor componente ale unui sistem. Căci chiar neproblematică nu sînt nici întreruperile; orice task se poate afla într-o fază în care să nu poată fi întrerupt. O astfel de fază se numește "secțiune critică". Ce anume este critic, este stabilit fie de sistemul de operare, fie chiar de task-ul în cauză. Acesta poate opri, multitasking-ul la începutul unei secțiuni critice printr-o funcție de genul "forbid ()", pentru a-l reporni mai tîrziu cu "permit ()".

Sistemul de operare la rîndul lui poate și trebuie să împiedice accesul simultan a două taskuri la o resursă unică, cum ar fi harddisk-ul sau imprimanta. Imaginați-vă că două taskuri tipăresc simultan două texte diferite, fiecare punînd cite o literă cînd e la rînd.

Pentru soluționarea acestei probleme există două strategii, și anume excluderea reciprocă ("mutual exclusion") și accesul exclusiv ("exclusive access"). În primul caz, sis-

temul de operare decide asupra distribuirii drepturilor de acces, în cel de-al doilea, task-ul trebuie să le solicite și să le predea.

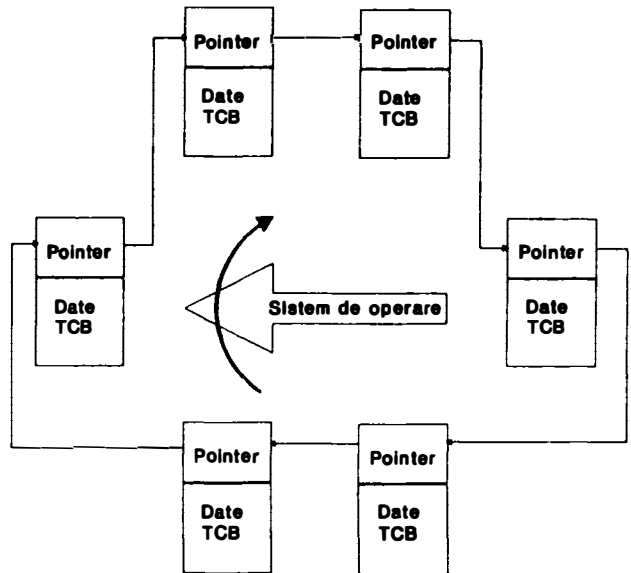
Astfel de solicitări constituie una dintre componentele de bază ale comunicației inter-task-uri bazată pe așa-numitele "mesaje". În acest context, și unele rutine trebuie văzute ca task-uri.

De desfășurarea impecabilă a acestei comunicații răspunde sistemul de operare. Că rolul de "regizor" nu este dintre cele mai facile, îl exemplifică așa-numitul "deadlock" (scurtcircuit): task-ul A așteaptă un mesaj de la B, iar taskul B așteaptă un mesaj de la A. Astfel de situații duc negreșit la blocarea sistemului și de aceea ar trebui pe cât posibil evitate sau măcar corectate de sistemul de operare.

Nu așa se întâmplă sub Windows: aici, întreaga responsabilitate rămâne pe umerii programatorului. El trebuie să aibă grijă ca să nu apară deadlock-uri (blocaje). Ceea ce uneori este dificil, mai ales când intră în joc funcția "Yield ()" ("cedează trecerea"). Aceasta oprește taskul curent și pornește un altul, care deja

așteaptă. Dacă însă un task "cedează trecerea" în timp ce prelucreză un mesaj, pe care un alt task i l-a trimis, se ajunge la blocajul amintit. Căci taskul expeditor așteaptă pînă cînd taskul destinat ar achită mesajul prin intermediul funcției "Reply Message ()". Ori, destinatarul deja nu mai este activ.

Și acest lucru ar fi ușor de evitat, dacă n-ar exista "amănunte" importante: "Yield ()" nu se cheamă totdeauna explicit, ci este deseori apelată automat de funcții deseori utilizate precum "Dialog Box ()" sau "Message Box". Deoarece nu poate să fie ceea ce nu are voie să fie, Windows oferă o funcție numită "InSendMessage ()", cu care se poate constata dacă mai există mesaje neachitate. Deci



"Round Robin" este o ordonare circulară a listei "TCB"

dacă programatorul e cuminte (cu minte) și testează totdeauna cu "InSendMessage ()" dacă apelul uneia din cele 8 funcții critice (care apelează "Yield ()") este permis, atunci nu există blocaje. După cum se vede însă din practică, nu toți programatorii respectă aceste indicații.

(I.F.)

Glosar

deadlock - două taskuri se așteaptă reciproc, ducînd la blocajul întregului sistem

distributor - rutină sistem care asigură oprirea unui task și lansarea altuia

eveniment - o acțiune externă, de care trebuie să se țină seama (o tastare, o mișcare a mouse-ului, etc.)

FCFS - First Come, First Served (primul venit, primul servit)

HOL - Head of Line - toate taskurile sînt într-o coadă de așteptare. Cel din capul listei e primul servit

non-preemptiv - sistem multitasking fără control de timp sau priorități. Fiecare task trebuie să redea de bună voie controlul sistemului de operare, care poate lansa atunci un alt task

PCB - Process Control Block - formă extinsă a TCB

preemptiv - sistem multitasking ghidat de timp sau de priorități. Un task

dislocă un altul (preia procesul) cînd a trecut un anumit interval de timp sau pentru că are prioritate mai mare

prelucrare cvasi-paralelă - pe calculatoare cu un singur procesor, o prelucrare paralelă nu este posibilă. În schimb fiecare task este prelucrat pe o "feliuță de timp", după care se trece la următorul

prelucrare paralelă - mai multe taskuri sau sarcini parțiale se desfășoară simultan pe mai multe procesoare, sub un sistem de operare unic.

round-robin - aranjament circular al listei de taskuri (după ultimul, urmează din nou primul)

scheduler - rutină sistem care decide, din mai multe taskuri posibile, care este cel selectat pentru continuare și cît timp i se alocă.

sistem multiprocesor - calculator realizat cu mai mult de un procesor, în

ultimul timp folosind mai ales transputere.

sistem multiluser - calculator ce poate fi folosit simultan de mai mulți utilizatori, de la mai multe terminale (de exemplu Unix)

spooler - task ce se derulează în background (plan secund) care preia foarte repede în zona de memorie un fișier ce urmează a fi tipărit, pentru a-l tipări apoi în întregime, consumînd pentru aceasta foarte puțin din timpul procesorului, disponibilizat astfel pentru alte activități în foreground (prim plan)

TCB - Task Control Block - zonă de date a unui task, în care sînt memorate toate datele interesante pentru sistemul de operare relative la task (nume, prioritate, stare, etc.)

Digital Research Multiuser-DOS 5.0

Rețelele sînt prea scumpe pentru multe întreprinderi. Se caută soluții mai avantajoase - una din ele ar putea fi Multiuser - DOS 5.0 de la Digital Research, cu care pot lucra pînă la 8 terminale simultan.

Din principiu, ceea ce un întreprinzător își dorește cel mai mult la colaboratorii săi este o productivitate fără limite. De aceea, o dotare modernă a biroului, cu echipamente permanente disponibile, a devenit un imperativ. Cînd să se lege însă în rețea calculatoarele, mulți întreprinzători nu îndrăznesc să facă investițiile necesare. De aceea, deseori, în firmele mici există un singur calculator care - pînă acum - se pornește doar de cîteva ori pe zi. Și atunci, imediat, 3 colaboratori stau la rînd, pentru a-și putea introduce datele. Dar timpul de așteptare este timp pierdut.

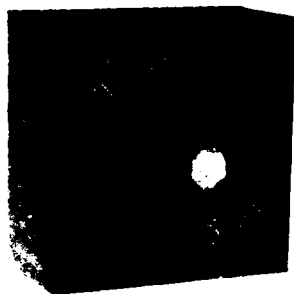
Astfel că un sistem pe măsură ar putea arăta în felul următor: un calculator central - bine dotat, cu memorie multă (minim 2 Mbyte RAM, preferabil 4), disc de masă de capacitate mare (min 80 Mbyte) precum și echipamente periferice - și mai multe terminale "proaste", conectate ca stații de lucru. Această "rețea" trebuie să fie ușor de instalat. Siguranța în funcționare este oricum o premisă indispensabilă.

Digital Research oferă o soluție care, la prima privire, satisface toate aceste cerințe: Multiuser - DOS 5.0 pentru circa 1500 DM. Rezistă la o verificare mai atentă?

Multiuser - DOS 5.0 este întîi de toate un sistem de operare MS - DOS compatibil. Folosește capacitățile procesorului 80386 - deci calculatorul central trebuie să fie dotat cu un cip 80386SX, 80386 sau 80486. Acest tip de procesor este capabil de multitasking, deci poate rezolva mai multe sarcini simultan. În mod normal, funcționarea multitasking, el emulează mai multe procesoare 8086, care rulează

"simultan". Este exact ce folosește Multiuser - DOS. Atunci, pînă la 8 terminale "proaste" (stații de lucru fără disc de masă și UC) pot lucra deodată. Ele se leagă prin interfețe seriale - deci calculatorul central trebuie să aibă suficiente racorduri.

Acest mod de transfer al datelor evidențiază, aparent, un prim dezavantaj - viteza de lucru este mult redusă față de rețelele "adevărate". Și totuși, nu e chiar așa. MU-DOS tratează fiecare PC conectat ca pe un terminal pur și simplu, ignorînd complet "inteligenta" disponibilă în acesta. Spre deosebire de multe



rețele locale, centralul nu oferă stației conectate date și programe, spre a fi prelucrate pe procesorul stației, ci trimite pe linie numai "ecrane" gata pregătite, preluînd apoi de la terminal eventuală tastări. Centralul "cadorisește" stațiile conectate cu o parte din capacitatea sa de calcul și preia integral execuția tuturor programelor și toate prelucrările de efectuat. Astfel, pe linie circulă un volum de date pentru care vitezele de transmisie realizabile cu o simplă legătură serială sînt absolut satisfăcătoare.

Funcție de cablul serial folosit, se poate ajunge la ecrane "cu pu-reci", dacă există erori în transmisie (nu se face nici un fel de verificare, terminalele sînt considerate total neajutorate...) În mod normal, reducerea vitezei de transmisie rezolvă problema. Cabluri de calitate

bună, scurte, pot lucra chiar pînă la 38.400 Baud, deși 19.200 pare a fi o recomandare mai bună, chiar pentru cabluri de pînă la 10 m lungime. Sistemul "se vede" suficient de rapid chiar și la 9600 Baud.

Este evident faptul că orice surplus de capacitate la terminale este risipă, dacă acestea urmează să fie folosite numai cu MU-DOS. XT - uri venerabile sau AT - uri descompletate sînt candidați ideali pentru acest scop.

Nu este posibilă nici integrarea suporturilor de date locali în rețea. Odată lansat "PCTERM", programul de emulare terminal, toate referințele se adresează la perifericele centralului. "A:" nu mai este unitatea de floppy cu care tocmai ați lucrat, ci este cea de pe central - dacă vreți ca datele de pe discheta Dvs. să ajungă la central, fie duceți pur și simplu discheta acolo, fie renunțați temporar la MU-DOS și lansați un program de comunicație adecvat (FILELINK de ex., cunoscut deja din DR-DOS).

Sub controlul PCTERM, ecranul și tastatura terminalului pot fi privite ca parte integrantă a centralului. Toate celelalte resurse locale (harddisk-uri, unități floppy, imprimantă, mouse etc.) sînt în hibernare. Chiar și adaptorul grafic "doarme". Indiferent de rezoluția de care este capabil adaptorul din terminal, centralul nu "prepară" în modul grafic, pentru terminalele conectate, ecrane care să depășească rezoluția unui CGA. Păcat, căci la ora actuală chiar și cele mai ieftine PC-uri sînt dotate cu adaptoare mai bune.

Deși în mod standard DOS lucrează numai cu maxim 4 interfețe seriale, MU-DOS n-are probleme cu de 4 ori pe-atîtea. Drive-uri potrivite pentru 64 de cartele multiport (cartele ce diferă de interfața serială standard, avînd de exemplu posibilități de conectare pentru 8 sau 16 echipamente seriale) oferă

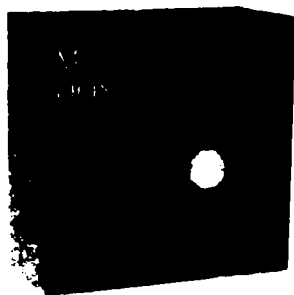
variate posibilități de realizare a legăturilor.

Pentru ca în sistem să nu poată pătrundă spioni, MU-DOS oferă un sistem de securizare administrat de utilizator. Se pot proteja, cu parole individuale, programe, fișiere de date, directoare întregi, hard-disk-uri sau dischete.

Există o comandă MAIL, prin care se pot transmite mesaje de la un terminal la altul. Dar rămâne în grija fiecărui utilizator să se "uite" în cutia poștală, să vadă dacă are ceva - sistemul nu-l anunță în nici un fel.

CSPPOOL poate administra mai multe imprimante deodată, pentru toate terminalele conectate - un serviciu indispensabil oricărei rețele.

Probleme cu utilizatori răuvoitori ar putea fi generate de comanda XSTOP. Aceasta permite oprirea



oricărei aplicații de pe orice terminal. Și mai periculos este REBOOT - cu această comandă, se execută o restartare a centralului. Chiar și atunci când comanda a fost dată de la un terminal. Conform manualului de utilizare, comanda ar trebui să fie la dispoziția exclusivă a "super-utilizatorului" - cel de la central. Acest lucru se pare însă că nu este numai adevărat. Dacă REBOOT nu se protejează printr-o parolă, atunci orice utilizator supărat poate termina toate activitățile în curs - brusc și irevocabil. Rezultatul poate fi catastrofal pentru datele de pe sistem.

Spre deosebire de REBOOT, comanda SECURITY este într-adevăr la dispoziția exclusivă a super-utilizatorului. Apelarea neautorizată a acesteia aduce pe ecran mesajul

"only be used by the system administrator" - convingător (când nu ești tu acela...)

Confortabil pe de altă parte este că fiecare din procesoarele 8086 virtuale este complet independent. Dacă - dintr-o tastare eronată de exemplu - un terminal "cade", restartarea se face, ca de obicei, tastând

<Ctrl> + <Alt> + . Ceilalți utilizatori nu observă nimic din întreaga dandana (cu condiția, totuși, ca la instalare să se fi dezactivat opțiunea "Global Reset"-altfel.).

Ca și pe central, orice terminal - chiar și un XT prăpădit - poate comuta între mai multe programe prin simplă apăsare de taste, rezolvând concomitent mai multe joburi deodată. Multitasking "real", de cea mai bună calitate. Funcție de memorie disponibilă pe central, fiecare terminal poate rula deodată pînă la maxim 8 aplicații.

Sistemul de operare diferențiază noțiunile de "station", "console" și respectiv "session". O stație este un terminal conectat la o interfață serială. Pentru fiecare stație, pot fi deschise pînă la 8 sesiuni, fiecărei sesiuni corespunzându-i câte o consolă. Fiecare sesiune constituie un domeniu DOS distinct, ce poate fi configurat individual, inclusiv memoria extinsă ce i se alocă. În afară de un AUTOEXEC.BAT local, aflat în directorul personal al utilizatorului, se pot face și alte diferențieri prin intermediul variabilelor de sistem, astfel încît la pornirea sistemului fiecare sesiune să fie pregătită pentru anumite sarcini.

Sesiunea care lucrează în "prim-plan" folosește în principiu întregul ecran - nu se lucrează, ca la Windows, în ferestre. Trecerea la alt ecran se face folosind o combinație de taste. Se poate tasta fie <Ctrl> + <Esc>, care aduce un meniu pe ecran, din care apoi se selectează aplicația dorită, fie <Alt> + <Esc>, care trece în sesiunea următoare, fie <Ctrl> + o cifră de pe blocul numeric al tastaturii, avînd ca efect trecerea directă în sesiunea cu numărul indicat.

O sesiune deschisă poate rula și în "background" (în plan secund, în spate, fără să se vadă). Dacă nu este vorba de calcule multe, atunci existența ei nu împieteză prea mult asupra performanței generale. Există și o "idle-detection" (dețecția unei stări în care programul nu face nimic util, așteptînd de exemplu o tastă) care permite "liniștirea" unei sesiuni care risipește timpul procesorului. Opțiunea SUSPEND permite "înghețarea" temporară a tuturor programelor ce nu lucrează în prim-plan. În acest caz, sigur că nu mai este vorba de multitasking.

Centralul acceptă moduri video diferite pentru sesiuni diferite, cît timp este vorba de moduri standard. Se poate lucra deci într-o sesiune cu un program grafic și o rezoluție VGA de 640 x 480 pixeli, în timp ce un altul lucrează în mod text iar un al treilea folosește modul CGA. Probleme apar la adaptoare VGA moderne, pe care MU-DOS nu le poate trata corect la trecerea dintr-o sesiune într-alta și înapoi. Ecranul "se murdărește" și nu mai poate fi "curățat" decît prin intermediul unor macroui speciale.

Probleme reale și deranjante există la lucrul cu mouse. Cît timp se folosește un mouse compatibil Microsoft și se dorește folosirea lui exclusiv pe central, totul e clar. Un singur mouse fizic, poate fi folosit, complet independent, în toate sesiunile, afectînd totdeauna numai sesiunea aflată în prim-plan.

Acest mouse poate fi disponibilizat și altor utilizatori, care lucrează la terminale - fără însă a-l deconecta de la central. Ceea ce în practică înseamnă că toate terminalele care vor dori să lucreze cu aplicații care necesită mouse trebuie să se afle în imediata vecinătate a centralului, pentru a permite mutarea mouse-ului la terminalul ce are nevoie de el, fără a-l "scoate" din central. Sigur că se pot conecta mai multe mouse-uri la central - afectarea lor unui anumit terminal nu este o problemă și se face prin procedura de instalare (setup).

(continuare în pag. 28)

Fileserver: alegere corectă

Dintre toate PC-urile dintr-o rețea, fileserver-ale sînt cele mai importante. Responsabile de păstrarea tuturor datelor, alegerii lor ar trebui să i se acorde o importanță deosebită. Nu numai viteza este importantă - un rol determinant îl mai au și lărgimea bus-ului, partea de rețea, carcasa și multe altele.

Server, în traducere directă, înseamnă servitor. Și fără server, nu se descurcă nici o rețea locală. Astfel, un fileserver este un pilon de bază în arhitectura unui LAN. Inițial, rolul său era de a păstra datele pentru toate stațiile de lucru reunite în rețea, servindu-le astfel pe acestea.

Și astăzi, acesta este unul din serviciile cele mai pretențioase pe care le oferă, chiar dacă acesta a câștigat mult în complexitate (de exemplu datorită conceptului de client - server - computing). Importanța fileserver-ului nu a fost diminuată la căderea server-ului, toată rețeaua stă, de aceea, la rețele mari se recomandă oglindirea discului de masă sau chiar dublarea fileserver-ului. Alegerea fileserver-ului ar trebui să i se acorde o importanță deosebită.

Fileserverul este "inima" întregii rețele, și pentru că pe el este instalat sistemul de operare al rețelei și toate programele și datele ce trebuie să fie disponibile la nivelul întregii rețele. În plus, foarte multe din caracteristicile importante pentru o rețea locală, cum ar fi securitatea și protecția datelor, depind fizic de tipul și dotarea fileserver-ului.

În fond, desigur, un fileserver nu este altceva decît un PC normal, deci cu UC, memorie operativă, disc de masă, etc. Și el poate fi apreciat după cîțiva factori de performanță, care, în legătură cu sarcinile sale funcționale, ar trebui să

servească drept criterii de selecție. Acest lucru este la fel de important în rețele mici ca și în cele mari. Desigur cu diferența că în rețelele mari importanța fileserver-ului crește. Nu numai pentru că sînt conectate mai multe stații de lucru și se desfășoară mai multe aplicații, ci pentru că în astfel de cazuri se folosesc întotdeauna servere dedicate.

Adică: un PC este instalat ca server și aceasta este funcțiunea sa exclusivă, spre deosebire de server-ul nededicat, care simultan - ca activitate "secundară" - servește unui utilizator ca punct de lucru. Dacă în LAN-uri cu servere nededicate se definesc PC-uri servere, ele pot funcționa atît ca servere cît și ca puncte de lucru; de aceea, "căderea" unui server nu este așa de tragică ca în rețele mari, cu server dedicat.

Criterii de selecție

După cum am mai pomenit, fileserverele sînt în fond PC-uri normale, deși parțial special configurate pentru rețea, lucru asupra căruia vom reveni. Ceea ce face posibilă aprecierea lor după criterii aplicabile oricărui astfel de echipament.

Procesorul și frecvența de ceas: constituie desigur unul din cele mai importante criterii de selecție; lucru pe care îl realizează desigur oricine a făcut vreodată trecerea de la 8088 la 80286. Și cît de mult s-ar repercuta acest lucru asupra vitezei întregii rețele, ar trebui să fie evident. În timp ce 80286 lucrează atît intern cît și extern pe 16 biți, procesoarele 80386 respectiv 68030 de la Motorola sînt orientate pe 32 biți. Și astfel de calculatoare "state of the art" (cele mai bune la momentul discuției) ar trebui să fie avute în vedere pentru cine ține la rețeaua lui. Dacă nu se face direct trecerea la

procesoare de tip 486. Relativ la procesoare Intel sau Motorola: în domeniul PC-urilor, procesoarele Intel sînt în continuare regula. Cu ele, există garanția că nu numai cele mai importante sisteme de operare de pe PC-uri (DOS, OS/2) sînt funcționale, ci și sistemele de operare în rețea le sînt adaptate. Astfel, decizia pentru procesoarele Intel oferă certitudinea că întregul soft deja existent "merge" desigur, în măsura în care e apt de funcționare în rețea.

Frecvența ceasului contribuie și ea la performanța unui PC. Oricum însă, performanța nu crește linear cu frecvența de ceas. Deci, în mod normal, un PC la 80 MHz nu e de două ori pe atît de iute ca unul la 15 MHz. Performanța depinde puternic de celelalte componente.

Cicli de wait (asteptare): un procesor care lucrează la o frecvență de ceas mare este relativ puțin, în comparație cu celelalte componente electronice, în mod special memorie. De aceea, deseori este nevoie de o adaptare două viteze diferite: la operații pur interne, procesorul lucrează la toată viteza, deci execută o operație la fiecare ciclu de ceas, în timp ce la o operație externă (acces la memoria principală sau cartele suplimentare) trebuie să prelungească durata operației pe mai mulți cicli. De aici numele de ciclu wait. La alegerea unui server, nu ar trebui deci să se pună bază doar pe frecvența de ceas a procesorului. Un PC normal la 10 MHz fără ciclu de așteptare este de exemplu mai rapid decît unul la 12 MHz, dar care lucrează cu un ciclu de așteptare atunci cînd accedează memoria. "State of the art" sînt fileservere ce lucrează la peste 30 MHz, unde trebuie să se țină seama însă și de cicli de așteptare.

Memoria principală: cu cît sînt mai mari rețelele, cu atît cantități mai mari de date trebuie să prelu-

crate în memoria principală - și memoria cache trebuie luată în considerare. Două puncte sînt importante. Mai întîi, care este memoria maximă adresabilă. Astfel de exemplu, la rețele mai mari pot fi necesare pînă la 16 Mbyte de memorie, pentru a garanta o prelucrare optimă a datelor.

În al doilea rînd, dacă implementarea acesteia se poate face direct pe cartela principală sau printr-un racord special, sau se face prin cartele suplimentare. Căci acestea din urmă implică, în mod normal, iarăși, cicli de wait.

Bus-ul sistemului și lărgimea bus-ului: la alegerea bus-ului sistemului, există între timp posibilitatea opțiunii între arhitectura (tradițională) ISA, arhitectura MCA (microchannel) sau EISA (extended ISA). Din punct de vedere al ratei de transfer, nu se poate face o recomandare globală. Deoarece decizia odată luată determină posibilitățile ulterioare de extensie (cel puțin între ISA/EISA pe de o parte și MCA pe de altă parte), alegerea ar trebui luată funcție de strategia generală a întreprinderii relativă la achiziționarea de PC-uri.

Lărgimea bus-ului a fost pomenită deja în legătură cu tipul procesorului. Ea are importanță și pentru că ea determină cantitatea de date care se vehiculează într-un ciclu de ceas între UC și o cartelă de extensie. La un bus de 32 biți, această cantitate e desigur semnificativ mai mare decît la unul de 16 sau 8 biți. Pentru fileserver, ar trebui luate în considerare PC-uri care permit utilizarea unei cartele adaptor de 32 biți. La punctele de lucru, desigur că acest lucru nu trebuie neapărat să se îndeplinească.

Harddisk-uri și subsisteme de memorie de masă de capacitate mare: Pe lîngă harddisk-uri, tot mai mult se folosesc și discuri optice. Deoarece datele efective cresc totdeauna rapid, un PC care permite folosirea numai a unui sau două harddisk-uri nu este indicat ca fileserver. Pentru a fi pregătit

pentru viitor, ar trebui să existe loc pentru cel puțin opt.

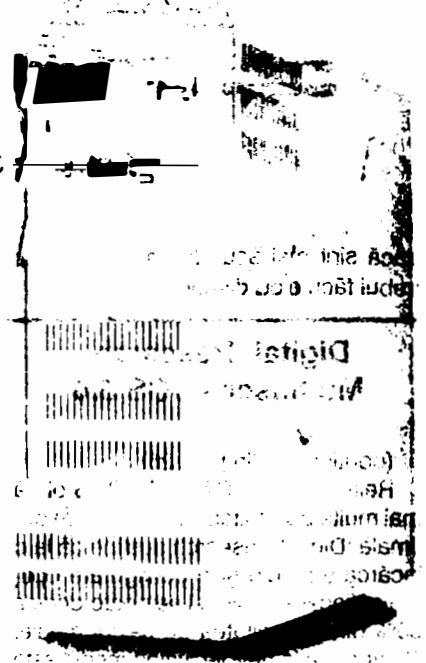
Subsistemele disc oferă desigur și opțiunea capacității de memorare liber extensibile - din punct de vedere al fileserver-ului însă, acest lucru prezintă inconveniente. E nevoie astfel de întreținerea mai multor sisteme, e necesară o carcasă și o alimentare proprie. Dacă fileserver-ul dispune de o sursă neîntreruptibilă - ceea ce este strict necesar de altfel - aceasta nu prea e de mare folos la harddisk-urile externe.

Posibilitățile de extensie într-un server trebuie să fie disponibile multe slot-uri libere. Această nu numai din motive de extensie a sistemului (ca de exemplu eventuale cartele RAM, controlere suplimentare sau adaptoare serial-paralele etc.), ci și pentru eventuala folosire a unor cartele de adaptare LAN. Astfel, se întîmplă deseori să apară necesitatea "legării" mai multor rețele - prin "bridge" - uri (poduri) de exemplu. Atunci, se poate întîmpla să trebuiască implantate pînă la 4 cartele adaptor într-un server.

Carcasa/design-ul: la fileserver, cu certitudine apare necesitatea schimbării, adăugirii sau eliminării unor părți. De aceea, este important ca PC-ul utilizat ca server să fie ușor accesibil, preferabil din mai multe părți. Deasemenea, toate componentele interne ar trebui să fie ușor și simplu accesibile. Totodată ar trebui să fie oferite mecanisme de protecție care să facă imposibil - sau măcar foarte dificil - accesul nevizat la alt de importantul server.

Se poate desigur - și în anumite circumstanțe chiar ar trebui - instala server-ul într-o încăpere separată. Dacă acest lucru nu este posibil, carcasa ar trebui să poată fi măcar încuiată.

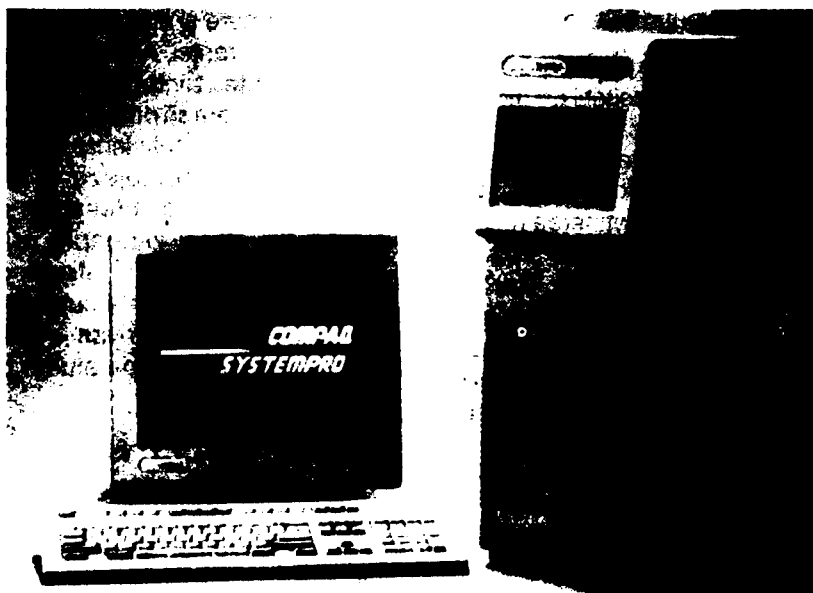
Tocmai la fileserver ar trebui avute în vedere și alte măsuri care servesc siguranței. Astfel, unitatea floppy ar putea fi integrată sub panoul frontal, care să poată fi încuiat. Trebuie să existe suficiente ventilaatoare care să protejeze la supraîncălzire, căci fileserver-ul merge de obicei zi-noapte. Sînt avantajoase desigur și afișaje cu LED-uri care să permită cuprinderea dintr-o singură privire a stării celor mai importanți parametri de funcționare.



IBM PS/2-Modell 95

Fiabilitatea: este desigur factorul cel mai important. Acest lucru începe cu sursa de alimentare și se continuă cu robustețea electronică, proiectată pentru o utilizare non-stop. Cel tîrziu aici, se poate separa făina de tîrțe. În două privințe.

În primul rînd, PC-urile "de marcă", cu care deja s-au făcut experiențe pozitive ca fileservere, se deosebesc de cele "no name", care sînt ieftine, dar deseori nu conțin o electronică bine acordată. Numele prezintă garanții în acest domeniu sînt Compaq, NCR și IBM. Chiar



Compaq Systempro

dacă sînt mai scumpe: economiile ar trebui făcute cu discernămint.

În al doilea rînd, se deosebesc PC-urile de marcă, de PC-urile special con-

cepute ca file-servere. Căci există producători care s-au specializat în producerea de fileservere. Corespunzător, multe din caracteristicile discutate mai sus se regăsesc în acestea. Ca nume, ar merita pomenite 3COM, NCI, Monolithics sau AST. Multe echipamente oferă pe lîngă procesoare foarte rapide și posibilitatea de a utiliza mai multe UC-uri de mare performanță (386 și/sau 486) în mod mixt, într-un singur calculator, de exemplu Compaq cu Systempro sau Parallon cu Hyperserver. În măsura în care sistemele de operare "colaborează", se pot obține astfel caracteristici care le ating pe cele ale calculatoarelor mari.

Dacă acestea nu sînt însă necesare - ceea ce, de regulă, este adevărat în marea majoritate a rețelelor locale, care numără la pînă în 10 puncte de lucru reunite - se recomandă folosirea unui PC cu 386 sau 486 ca fileserver, cu considerarea cerințelor punctate mai sus. "Creșterea" unui LAN de regulă este preprogramată - astfel că un file-server performant merită avut în vedere de la bun început.

(I.F.)

Digital Research Multiuser - DOS 5.0

(continuare din pag. 25)

Relativ la tema RAM, MU-DOS oferă mai multe posibilități de configurare optimă. Digital Research susține că la o încărcare medie și o configurare optimă, fiecărei sesiuni i se poate dedica 80% din capacitatea de calcul medie. Chiar dacă la nevoie "valorii medii" este o noțiune la fel de extensibilă ca guma, diverse teste de performanță lasă o impresie bună.

Sigur că sînt probleme principale cu aplicații ca 'Windows 3.0' în modul extins sau 'UltraScript', care comută 80386 în Protected Mode și depind foarte mult de memoria extinsă. Deoarece MU-DOS lucrează în mod virtual, conceput special pentru utilizarea multitasking a UC-urilor cu 80386, întreaga memorie este deja protejată și nu mai poate fi trecută încă odată în această stare de alte programe.

Pentru instalarea pachetelor de programe cele mai des folosite, există indicații în manualul de utilizare (pentru Paradox 3.5, Word 5.0, Lotus 1-2-3 2.2 (Server Edition), Dataease 4.2 (LAN), Quattro-Pro 2.0, Lotus Manuscript 2.1 (LAN), WordPerfect 5.1, Letter-Perfect 1.0, Q&A 3.0 (LAN), SuperCalc 5, Framework III 1.1 și dBase IV 1.1).

Digital Research își protejează sistemul - comercializat la 1500 DM - cu un "dongle", care trebuie introdus în prima interfață paralelă. Ca la multe alte astfel de scheme de protecție, imprimanta legată la dongle trebuie să fie neapărat pe automat (online) pentru ca sistemul, la pornire, să depășească faza de verificare a protecției. E discutabilă această soluție - imaginați-vă ce-ar fi dacă oricare aplicație mai bună ar fi protejată cu un astfel de dispozitiv. S-ar deschide o nouă piață pentru produse de stabilizare a conectorilor disponibili pe panoul din spate al PC - urilor ...

MU-DOS poate lucra cu harddisk-uri de pînă la 2 Gbyte (!). Ceea ce înseamnă peste 200 de discuri de 100 Mbyte... Pentru unități CD-ROM uzuale sînt disponibile drivere adecvate, astfel încît merită să se investească în central, făcînd din el o veritabilă centrală de date.

Per ansamblu, Multiuser - DOS 5.0 de la Digital Research se dovedește a fi o soluție performantă, cu un multitasking aproape perfect. Sistem multiutilizator, permițînd instalarea unor locuri de muncă ieftine pentru pînă la 8 colaboratori, el se pretează îndeosebi la eficientizarea unor laptop-uri mai slabe sau a unor XT-uri vechi, care deseori dispun numai de adaptoare video CGA sau monocrome și au doar o unitate floppy în configurație.

Programatorii de la Digital Research au realizat un sistem "curat", care în mod cert va avea un impact serios asupra pieții. Totuși, ar trebui să rezolve neapărat, și nu prea tîrziu, și lucrul cu cartele grafice de rezoluție înaltă, care sînt, în momentul de față, standarde acceptate. Altfel, la prima ocazie, din nou Microsoft va culege toți laurii. Cum s-a mai întîmplat, de altfel.

(I.F.)

OLIVETTI * SZEGED

CAUȚĂ
PARTENER DE AFACERI ÎN
VEDEREA UNEI COLABORĂRI
COMUNE ÎN CONDIȚII
DEOSEBIT DE AVANTAJOASE!

OFERĂ:

ÎNTREAGA GAMĂ
DE ECHIPAMENT DE CALCUL,
COPIERE ȘI BIROTICĂ
(COMPUTERE, COPIATOARE,
MAȘINI DE CALCULAT,
DESK-TOP PUBLISHING,
TELEFAX-URI,
MAȘINI DE SCRIS, ETC.)

INNOV TECHNIK Ltd.
6722 SZEGED, Gutenberg u. 3
HUNGARY
Tel/Fax: (36)62-19026

Lista comenzilor dBase IV

!	- identic cu RUN	CREATE REPORT	- apelează generatorul de rapoarte
;	- următoarea linie sursă este privită ca o linie de continuare a liniei curente	CREATE SCREEN	- apelează generatorul de machete
?	- afișează conținutul unor cîmpuri sau expresii și face un avans de linie	CREATE STRUCTURE	- apelează programul de definire a unui fișier DBF
??	- afișează conținutul unor cîmpuri sau expresii dar nu face și un avans de linie	CREATE VIEW	- apelează programul pentru definirea unui fișier VIEW
???	- trimite comenzi de imprimare direct la imprimantă	CREATE VIEW FROM ENVIRONMENT	- memorează modul curent de lucru ca fișier VIEW
@..CLEAR	- șterge zone ale ecranului	DBLINK	- programul extern cuprinde mai multe fișiere obiect
@..FILL	- modifică atributele de culoare în domeniul specificat de pe monitor	DEACTIVATE MENU	- dezactivează un meniu linie
@..SAY..GET	- afișează sau preia conținutul unei variabile sau al unul cîmp	DEACTIVATE POPUP	- dezactivează un meniu pop-up
@..TO	- desenează o linie sau un cadru între coordonatele specificate	DEACTIVATE WINDOW	- dezactivează o fereastră
*	- ca prim caracter într-o linie marchează linia respectivă ca pe o linie de comentariu	DEBUG	- apelează depanatorul (debugger)
&	- caracter macrou, determină înlocuirea numelui unei variabile cu conținutul ei	DECLARE	- definește un tablou
&&	- marchează restul liniei ca și comentariu	DEFINE BAR	- definește o opțiune pentru un meniu pop-up
ACCEPT	- cere o valoare pentru o variabilă caracter afișată pe monitor	DEFINE BOX	- definește un chenar pentru o expresie
ACTIVATE MENU	- activează un meniu linie	DEFINE MENU	- definește un meniu linie
ACTIVATE POPUP	- activează un meniu pop-up	DEFINE PAD	- definește o opțiune pentru un meniu linie
ACTIVATE SCREEN	- activează întregul domeniu monitor	DEFINE POPUP	- definește un meniu pop-up
ACTIVATE WINDOW	- activează o fereastră	DEFINE WINDOW	- definește o fereastră
APPEND	- inserează noi articole în fișierul curent	DELETE	- poziționează Indicatorul de ștergere pentru articole
APPEND FROM	- preia articole dintr-un alt fișier în fișierul curent	DELETE FILE	- șterge un fișier
APPEND FROM ARRAY	- adaugă articolele dintr-un tablou (array) la fișierul curent	DELETE TAG	- șterge o intrare într-un fișier MDX
APPEND MEMO	- preia un fișier de text într-un cîmp memo	DIR	- afișează conținutul unui director
ASSIST	- apelează centrul de regie	DIRECTORY	- identic cu DIR
AVERAGE	- calculează media aritmetică	DISPLAY	- corespunde cu LIST dar se oprește după fiecare pagină afișată
BEGIN TRANSACTION	- pornește o tranzacție	DISPLAY FILES	- corespunde cu LIST FILES dar se oprește după fiecare pagină afișată
BROWSE	- afișează datele sub formă de tabelă	DISPLAY HISTORY	- corespunde cu LIST HISTORY dar se oprește după fiecare pagină afișată
CALCULATE	- efectuează anumite calcule într-un fișier	DISPLAY MEMORY	- corespunde cu DISPLAY MEMORY dar se oprește după fiecare pagină afișată
CALL	- apelează un modul extern	DISPLAY STATUS	- corespunde cu LIST STATUS dar se oprește după fiecare pagină afișată
CANCEL	- întrerupe un fișier de comenzi	DISPLAY STRUCTURE	- corespunde cu LIST STRUCTURE dar se oprește după fiecare pagină afișată
CASE	- parte a construcției DO CASE	DISPLAY USERS	- corespunde cu LIST USERS dar se oprește după fiecare pagină afișată
CHANGE	- identic cu EDIT	DO	- apelează un program sau o procedură
CLEAR	- șterge monitorul sau parametrul specificat	DO CASE	- deschide un bloc 'case'
CLOSE	- închide fișiere	DO WHILE	- pornește un ciclu 'atît timp cît'
COMPILE	- compilează un fișier	EDIT	- comută în modul editare
CONTINUE	- caută următoarea noțiune căutată cu LOCATE	EJECT	- determină un salt de pagină la imprimantă
CONVERT	- convertește un fișier DBF pentru a fi utilizat în rețea	EJECT PAGE	- determină un salt de pagină controlat la imprimantă
COPY	- copiază un fișier, sau părți dintr-un fișier, într-alt fișier	ELSE	- parte a construcției IF
COPY FILE	- copiază un fișier	END TRANSACTION	- sfîrșitu! buclei BEGIN TRANSACTION
COPY INDEXES	- copiază fișiere NDX într-un fișier MDX	ENDCASE	- parte a construcției DO CASE
COPY MEMO	- copiază un cîmp memo într-un fișier text	ENDDO	- parte a construcției DO WHILE
COPY STRUCTURE	- copiază structura unui fișier DBF	ENDIF	- parte a construcției IF
COPY STRUCTURE EXTENDED	- copiază o formă specială a structurii unui fișier DBF	ENDPRINTJOB	- sfîrșitul buclei PRINTJOB
COPY TAG	- convertește intrările dintr-un fișier MDX în fișiere NDX	ENDSCAN	- sfîrșitul buclei SCAN
COPY TO ARRAY	- copiază date dintr-un fișier DBF într-un tablou	ENDTEXT	- sfîrșitul buclei TEXT
COUNT	- contorizează articolele unui fișier DBF care satisfac o condiție dată	ERASE	- șterge un fișier
CREATE APPLICATION	- apelează generatorul de programe pentru crearea unei aplicații	EXIT	- determină ieșirea dintr-o buclă DO WHILE sau SCAN
CREATE FROM	- creează un nou fișier DBF dintr-o structură de date creată cu COPY STRUCTURE EXTENDED	EXPORT	- convertește un fișier DBF într-un format străin
CREATE LABEL	- apelează generatorul de etichete	FIND	- caută o noțiune într-un fișier index
CREATE QUERY	- identic cu CREATE VIEW	FUNCTION	- denumește o funcție autodefinită
		GET	- vezi: @..SAY..GET
		GO	- salt la un anumit articol
		GOTO	- vezi: GO
		HELP	- apelează sistemul de mesaje de ajutor
		IF	- deschide un bloc 'dacă'
		IMPORT	- convertește un format străin în format DBF

INDEX	- creează un fișier index	PUBLIC	- definește variabilele valabile în întregul program
INPUT	- cere introducerea de la tastatură a valorii unei variabile numerice pe monitor	QUIT	- termină sesiunea dBase
INSERT	- înserează un articol vid în fișierul curent	READ	- citește toate variabilele enumerate după GET
JOIN	- creează un nou fișier DBF din două fișiere DBF	RECALL	- îndepărtează marcajele de articole șterse
KEYBOARD	- pune caractere în buffer-ul de tastatură	REINDEX	- creează din nou indexul activ
LABEL FORM	- pornește o imprimare de etichetă	RELEASE	- șterge variabile din memorie
LIST	- afișează articole, fără întrerupere	RELEASE MENUS	- șterge meniurile linie
LIST FILES	- afișează fișiere, fără întrerupere	RELEASE POPUPS	- șterge meniurile pop-up
LIST HISTORY	- afișează ultimele comenzi tastate, fără întrerupere	RELEASE SCREENS	- șterge ecranele memorate
LIST MEMORY	- listează variabilele de memorie, fără întrerupere	RELEASE WINDOWS	- șterge ferestre
LIST STATUS	- listează starea curentă, fără întrerupere	RENAME	- modifică numele unui fișier
LIST STRUCTURE	- afișează structura fișierului curent, fără întrerupere	REPLACE	- înlocuiește valoarea unor cîmpuri cu noi valori
LIST USERS	- afișează utilizatorii din rețea, fără întrerupere	REPLACE FROM ARRAY	- preia înregistrările dintr-un tablou în fișierul curent
LOAD	- încarcă un modul extern	REPORT FORM	- pornește listarea unui raport
LOCATE	- caută o noțiune într-un fișier DBF	RESET	- elimină marcajele care au fost poziționate cu o comandă de tranzacție și care nu au fost îndepărtate încă
LOGOUT	- anunță ieșirea unui utilizator din rețea	RESTORE	- încarcă variabilele memorat
LOOP	- parte a construcției DO WHILE	RESTORE MACROS	- încarcă macrourile memorate
MODIFY APPLICATION	- corespunde cu CREATE APPLICATION dar servește doar la modificări	RESTORE SCREEN	- încarcă un ecran memorat
MODIFY COMMAND	- apelează editorul de programe	RESTORE WINDOW	- încarcă ferestrele memorate
MODIFY FILE	- corespunde cu MODIFY COMMAND, dar servește numai la modificări	RESUME	- relansează un program întrerupt
MODIFY LABEL	- corespunde cu CREATE LABEL, dar servește numai la modificări	RETRY	- relansează o comandă care nu a fost executată
MODIFY QUERY	- corespunde cu CREATE QUERY, dar servește numai la modificări	RETURN	- revenire în programul apelant
MODIFY REPORT	- corespunde cu CREATE REPORT, dar servește numai la modificări	ROLLBACK	- derulează înapoi acțiunile unei tranzacții
MODIFY SCREEN	- corespunde cu CREATE SCREEN, dar servește numai la modificări	RUN	- apelează o comandă a sistemului de operare
MODIFY STRUCTURE	- corespunde cu CREATE STRUCTURE, dar servește numai la modificări	SAVE	- memorează variabilele din memorie ca fișier
MODIFY VIEW	- corespunde cu CREATE VIEW, dar servește numai la modificări	SAVE MACROS	- memorează macrourile ca fișier
MOVE WINDOW	- mută o fereastră pe ecran	SAVE SCREEN	- memorează conținutul ecranului curent
NOTE	- marchează o linie ca linie de comentariu	SAVE WINDOW	- memorează o fereastră ca fișier
ON ERROR	- ramificație spre o rutină de tratare a erorilor în cazul apariției unei erori dBase IV	SAY	- vezi: @_SAY
ON ESCAPE	- după apăsarea tastei <Esc> programul execută secvența de comenzi care urmează după ON ESCAPE	SCAN	- prelucrează articole care satisfac o anumită condiție
ON KEY	- după apăsarea tastei desemnate se execută secvența de comenzi care urmează	SEEK	- caută o noțiune într-un fișier index
ON PAD	- apelează un meniu pop-up dintr-un meniu linie	SELECT	- selectează un domeniu de lucru
ON PAGE	- execută anumite comenzi la imprimarea anumitor pagini	SET	- apelează un meniu pentru specificarea parametrilor de setare a mediului de lucru
ON READERROR	- în cazul unor erori de intrare execută comanda dată	SHOW MENU	- afișează un meniu linie
ON SELECTION PAD	- tratează o intrare dintr-un meniu linie	SHOW POPUP	- afișează un meniu pop-up
ON SELECTION POPUP	- tratează o intrare dintr-un meniu pop-up	SKIP	- mută pointerul articol în fișierul curent
OTHERWISE	- parte a construcției DO CASE	SORT	- sortează articolele într-un nou fișier
PACK	- șterge dintr-un fișier toate articolele marcate ca șterse	STORE	- stochează o valoare într-o variabilă
PARAMETERS	- atribuie unor variabile valorile parametrilor de apel din programul apelant	SUM	- adună numere într-un nou fișier
PLAY MACRO	- execută un macro specificat	SUSPEND	- întrerupe un program dBase
PRINTJOB	- pornește o listare la imprimantă	TEXT	- marchează începutul unui bloc de text
PRIVATE	- limitează valabilitatea variabilelor la procedura sau funcția curentă	TOTAL	- adună valorile numerice ale unui fișier într-un al doilea fișier
PROCEDURE	- definește o procedură autodefinită	TYPE	- afișează conținutul unui fișier text
PROTECT	- apelează sistemul de protecție dBase	UNLOCK	- deblochează articole și fișiere în rețea
		UPDATE	- înlocuiește conținutul unor cîmpuri cu valori din alte fișiere DBF
		USE	- deschide un fișier DBF
		WAIT	- așteaptă apăsarea unei taste
		WHILE	- parte a construcției DO WHILE
		ZAP	- șterge toate articolele dintr-o bază de date

Lista operatorilor dBase IV		
+	- adunare	>= sau => - mai mare sau egal
-	- scădere	<= sau =< - mai mic sau egal
*	- înmulțire	\$ - conținut în
/	- împărțire	.and. - și
** sau ^	- ridicare la putere	.or. - sau
<	- mai mic	.not. - nu
>	- mai mare	
=	- egal	
<>	- diferit	

Lista funcțiilor dBase IV

ABS()	- returnează valoarea absolută a unui număr	OS()	- returnează versiunea sistemului de operare
ACCESS()	- determină drepturile de acces ale utilizatorului	PAD()	- dă numele opțiunii selectate dintr-un meniu pop-up
ACOS()	- calculează arccosinusul unui unghi	PAYMENT()	- calculează valoarea de plătit pentru un împrumut cu o dobândă fixă pentru un număr dat de rate
ALIAS()	- returnează numele alias al unui domeniu de lucru	PCOL()	- returnează coloana poziției curente a imprimantei
ASC()	- returnează codul ASCII al unui caracter	PI()	- dă valoarea pentru π
ASIN()	- calculează arcsinusul unui unghi	POPUP()	- dă numele meniului pop-up activ
AT()	- dă poziția de început a unui șir de caractere în cadrul unui alt șir de caractere	PRINTSTATUS()	- specifică dacă imprimanta este pregătită pentru listare
ATAN()	- calculează arctangentă unui unghi	PROGRAM()	- specifică numele programului care este în curs de execuție
ATN2()	- calculează arctangentă unui unghi	PROMPT()	- returnează textul punctului de meniu pop-up selectat
BAR()	- dă numărul opțiunii pop-up selectate	PROW()	- returnează linia curentă a imprimantei
BOF()	- indică începutul unui fișier	PV()	- calculează valoarea actuală a unor plăți cuprinzând dobânzile pentru un interval de timp specificat
CALL()	- apelează un modul extern	RAND()	- generează un număr aleator
CADOW()	- returnează ziua din săptămână a unei date	READKEY()	- specifică tasta cu care a fost încheiată o comandă orientată ecran
CEILING()	- rotunjește un număr la valoarea întregă imediat superioară	RECCOUNT()	- returnează numărul de articole dintr-un fișier DBF
ERROR()	- dă codul de eroare al ultimei erori de compliere	RECHO()	- returnează numărul de articole al articolului curent
CHANGE()	- determină orice modificare a articolului curent de către un alt utilizator în rețea	RECSIZE()	- returnează dimensiunea unui articol dintr-un fișier DBF
CHR()	- returnează caracterul ASCII indicat	REPLICATE()	- repetă o succesiune de caractere de mai multe ori
CMONTH()	- returnează numele lunii unei date	RIGHT()	- returnează partea dreaptă a unui șir de caractere
COL()	- returnează numărul coloanei poziției curente a cursorului	RLOCK()	- blochează un articol în rețea
COMPLETED()	- specifică dacă o tranzacție a fost încheiată cu succes	ROLLBACK()	- specifică dacă un rollback a fost încheiat cu succes
COS()	- calculează cosinusul unui unghi	ROUND()	- rotunjește un număr în sus sau în jos
CTOD()	- convertește o succesiune de caractere într-o dată	ROW()	- returnează numărul de linii al poziției curente
DATE()	- returnează data sistem	RTOD()	- convertește valoarea unui unghi dat în radiani în grade
DAY()	- returnează ziua unei date	RTREIB()	- elimină spațiile de la sfârșitul unui șir
DBF()	- returnează numele DBF al unui fișier deschis	SEEK()	- caută o noțiune după un index
DELETED()	- determină dacă un articol a fost șters	SELECT()	- specifică numărul celui mai mare domeniu de lucru liber
DIFFERENCE()	- returnează gradul de corespondență a două șiruri	SET()	- returnează poziționarea unei comenzi SET
DISKSPACE()	- returnează spațiul liber de pe unitatea de disc curentă	SIGN()	- returnează semnul unui număr
DMY()	- convertește o dată într-o succesiune de caractere	SIN()	- calculează sinusul unui unghi
DOW()	- returnează ziua din săptămână a unei date	SINDEX()	- returnează codul fonetic al unui șir de caractere
DTOC()	- convertește o dată într-o succesiune de caractere	SPACE()	- produce un șir de spații
DTOR()	- convertește mări mea în grade a unui unghi în radiani	SORT()	- calculează rădăcina pătrată dintr-un număr
DTOB()	- convertește o dată într-o variabilă caracter pentru indexare	STR()	- convertește valoarea unui număr într-un șir de caractere
EOF()	- indică sfârșitul unui fișier	STUFF()	- înlocuiește o parte a unui șir de caractere printr-o alta
ERROR()	- returnează codul de eroare al ultimei erori semnificate	SUBSTR()	- returnează o parte a unui șir de caractere
EXP()	- calculează e la puterea unui număr	TAG()	- returnează numele indexului dorit (fișier TAG sau NDX)
FIELD()	- dă numele câmpului dorit dintr-un fișier	TAN()	- calculează tangenta unui unghi
FILE()	- determină existența fișierelor indicate	TIME()	- returnează ora sistem
FIXED()	- convertește numerele de tip G în numere de tip N	TRANSFORM()	- reprezintă variabilele prin intermediul unei măști
FKLABEL()	- dă numele tastei funcționale dorite	TRIM()	- elimină spațiile de la sfârșitul unui șir
FKMAX()	- dă numărul tastei funcționale programabile	TYPE()	- specifică tipul unei variabile
FLOAT()	- convertește numere de tip N în numere de tip G	UPPER()	- convertește toate literele unui șir în litere mari
FLOCK()	- blochează un fișier într-o rețea	USER()	- returnează numele unui utilizator din rețea
FLOOR()	- rotunjește un număr la întregul imediat inferior	VAL()	- convertește un șir de caractere într-un număr
FLOOR()	- specifică dacă o căutare s-a încheiat cu succes	VARREAD()	- returnează numele variabilei sau câmpului în curs de prelucrare
FOUND()	- calculează valoarea vizitare a unor plăți, inclusiv dobânzile, într-un anumit interval de timp	VERSION()	- returnează versiunea dBase curentă
FV()	- returnează caracteristicile sistem dorite	YEAR()	- returnează anul unei date
GETENV()	- reacționează la o condiție		
HF()	- returnează numărul tastei apășate		
INKEY()	- returnează partea întregă a unui număr		
INT()	- determină faptul dacă primul caracter dintr-un șir este o literă		
ISALPHA()	- determină faptul dacă în calculator este instalat un adaptor grafic color		
BCOLOR()	- determină faptul dacă primul caracter a unui șir este o literă mică		
BLOWER()	- indică dacă un fișier este implicat într-o tranzacție		
ISMARKED()	- determină faptul dacă primul caracter al unui șir este o literă mare		
ISUPPER()	- dă expresia de cheie a unui fișier index		
KEY()	- returnează codul ultimei taste apășate		
LASTKEY()	- returnează partea sfârșit a unui șir de caractere		
LEFT()	- dă lungimea unui câmp sau a unei variabile		
LEN()	- indică dacă un șir de caractere este conținut într-un alt șir de caractere		
LIKE()	- dă numărul de linii dintr-un program în care a apărut o eroare		
LINENO()	- dă numele utilizatorului unui fișier într-o rețea		
LKSYS()	- identic cu RLOCK()		
LOCK()	- calculează logaritmul în baza e al unui număr		
LOG()	- calculează logaritmul în baza 10 al unui număr		
LOG10()	- caută o noțiune într-un alt fișier		
LOOKUP()	- convertește toate literele dintr-un șir în litere mici		
LOWER()	- elimină spațiile de început dintr-un șir		
LTRIM()	- returnează data ultimei modificări a unui fișier DBF		
LUPDATE()	- determină maximul dintre două expresii numerice		
MAX()	- returnează numele fișierelor MDX deschise		
MDX()	- convertește o dată într-un șir de caractere		
MDY()	- dă numărul de linii pentru un câmp memo		
MEMLINES()	- returnează numărul memoriei rămase liberă		
MEMORY()	- returnează numele meniului linie activ		
MENU()	- returnează mesajul de eroare pentru o eroare apărută		
MESSAGE()	- determină minimul dintre două expresii numerice		
MIN()	- returnează o linie - : - unui câmp memo		
MLINE()	- returnează restul unei împărțiri întregi		
MOD()	- returnează luna unei date		
MONTH()	- returnează numele fișierelor NDX deschise		
NDX()	- determină faptul dacă dBase IV lucrează în rețea		
NETWORK()	- dă indexul activ		
ORDER()			

Lista variabilelor sistem în dBase IV

_align	- comutator pentru alinierea textelor
_box	- stabilește dacă vor fi sau nu imprimare chenarele definite cu DEFINE BOX
_indent	- stabilește indentarea primei linii a unui paragraf
_margin	- stabilește marginea din stânga pentru afișarea cu ???
_page	- stabilește modul de avans al hirtiei
_page	- stabilește numărul paginii curente
_page	- stabilește prima pagină de imprimat
_page	- stabilește coloana listării
_printer	- stabilește în care exemplare se va face imprimarea
_driver	- stabilește driver-ul de imprimantă care va fi folosit
_pscode	- trimite coduri la imprimantă la sfârșitul imprimării
_pscode	- controlează modul de salt de pagină în timpul unei imprimări cu PRINTJOB
_pspage	- controlează numărul paginilor de imprimat
_psform	- așază un fișier cu un format de imprimare
_pslength	- stabilește numărul de linii pe pagină
_pslineno	- stabilește linia curentă la listare
_psfloat	- stabilește marginea din stânga la imprimare
_psfloat	- controlează distribuția dintre caractere
_psfloat	- comutator pentru calitatea imprimării
_psfloat	- trimite coduri la imprimantă înainte de începerea listării
_pspage	- stabilește distanța dintre linii
_pswait	- stabilește dacă imprimanta va aștepta după fiecare pagină listată
_rmargin	- stabilește marginea din dreapta la afișarea cu ???
_tabe	- poziționează tabulele
_wrap	- stabilește dacă se face "ruperea" liniilor lungi

Lista comenzilor SET în dBase IV

SET ALTERNATE	- copiază ieșirea pe ecran și într-un fișier
SET AUTOSAVE	- stabilește dacă modificările de articole să fie efectuate imediat și pe disc sau nu
SET BELL	- stabilește durata și frecvența alarmei sonore, puțin și să o anuleze
SET BLOCKSIZE	- stabilește dimensiunea blocurilor cîmpurilor memo și fișierelor MDX
SET BORDER	- stabilește forma chenarului ferestrelor și meniurilor
SET CARRY	- stabilește dacă, și care, conținutul de cîmpuri vor fi preluate din articolul precedent într-un nou articol
SET CATALOG	- deschide un fișier catalog
SET CENTURY	- stabilește forma de afișare a anilor (cu sau fără sute)
SET CLOCK	- controlează afișarea orei sistem în programe
SET COLOR	- stabilește culorile de afișare pe monitor
SET CONFIRM	- stabilește dacă cursorul va sări sau nu în cîmpul următor la terminarea unui cîmp
SET CONSOLE	- stabilește dacă se face sau nu afișarea pe monitor
SET CURRENCY	- stabilește simbolul monedei utilizate
SET CURSOR	- stabilește dacă cursorul va fi afișat sau nu
SET DATE	- stabilește formatul datei
SET DEBUG	- redirecționează afișările comenzii SET ECHO la imprimantă
SET DECIMALS	- stabilește numărul pozițiilor zecimale afișate
SET DEFAULT	- stabilește unitatea de disc curentă
SET DELETED	- stabilește dacă să fie ignorate sau nu și articolele șterse
SET DELIMITERS	- stabilește delimitatorii pentru afișarea cîmpurilor
SET DESIGN	- stabilește dacă generatorul de programe poate fi lansat din centrul de regie
SET DEVELOPMENT	- activează validarea datelor de creare a unor fișiere program originale și compilate
SET DEVICE	- stabilește perifericul de ieșire pentru comanda @..SAY
SET DIRECTORY TO	- stabilește unitatea și directorul curent
SET DISPLAY	- stabilește modul de afișare pe ecran
SET DOHISTORY	- fără funcție
SET ECHO	- afișează pe ecran liniile de program executate
SET ENCRYPTION	- activează cifrarea fișierelor
SET ESCAPE	- stabilește dacă tasta <Esc> este activă
SET EXACT	- cere o corespondență exactă a două șiruri de caractere
SET EXCLUSIVE	- deschide fișiere în rețea exclusiv
SET FIELDS	- limitează cîmpurile de prelucrat
SET FILTER	- limitează fișierul la articolele care să respecte o anumită condiție
SET FIXED	- fără funcție
SET FORMAT	- atribuie o mască comenzilor orientate ecran
SET FULLPATH	- activează afișarea integrală a căii (path)
SET FUNCTION	- încarcă o succesiune de caractere pe o tastă funcțională
SET HEADING	- activează afișarea antetelor de coloană
SET HELP	- activează afișarea automată a mesajelor de help
SET HISTORY	- activează și stabilește dimensiunea memoriei de comenzi
SET HOURS	- stabilește modul de afișare al orei sistem
SET INDEX	- stabilește indexul activ
SET INSTRUCT	- activează afișarea de indicații
SET INTENSITY	- activează reprezentarea evidențiată a cîmpurilor în timpul introducerii
SET LOCK	- stabilește dacă articolele vor fi blocate automat în timpul prelucrării în rețea
SET MARGIN	- stabilește marginea din stînga la imprimare
SET MARK	- stabilește separatorul între ziua, luna și anul unei date
SET MEMOWIDTH	- stabilește lățimea de ieșire pentru un cîmp memo
SET MENU	- fără funcție
SET MESSAGE	- afișează un mesaj al programului
SET NEAR	- stabilește dacă la o căutare a unei valori este necesară o potrivire exactă sau este suficientă asemănarea
SET ODOMETER	- stabilește intervalul contorului de articole
SET ORDER	- stabilește indexul curent
SET PATH	- stabilește calea (path-ul)
SET PAUSE	- stabilește afișarea datelor în modul SQL

SET POINT	- stabilește forma semnului zecimal
SET PRECISION	- stabilește precizia de calcul în dBase
SET PRINTER	- stabilește ieșirea pentru imprimare
SET PROCEDURE	- activează un fișier de proceduri
SET REFRESH	- stabilește intervalul de timp după care va fi reimprospătată imaginea afișată pe ecran cu datele actualizate la lucrul în rețea
SET RELATION	- stabilește o legătură între două fișiere
SET REPROCESS	- stabilește de câte ori va fi repetată încercarea de executare a unei comenzi la lucrul în rețea în cazul în care prima încercare eșuează
SET SAFETY	- împiedică suprascriserea din greșeală a fișierelor
SET SCOREBOARD	- controlează afișarea setărilor pentru tastatură
SET SEPARATOR	- stabilește forma de reprezentare a miilor (separatorului)
SET SKIP	- stabilește în care fișier va fi mișcat pointerul articol în cazul fișierelor legate prin SET RELATION
SET SPACE	- înserază un spațiu între cîmpurile afișate
SET SQL	- activează modul SQL
SET STATUS	- activează afișarea liniei de stare în partea de jos a ecranului
SET STEP	- execută un program pas cu pas
SET TALK	- controlează afișarea mesajelor
SET TITLE	- editează liniile de comentariu ale unui catalog
SET TRAP	- apelează debugger-ul în cazul apariției unei erori în program
SET TYPEAHEAD	- stabilește mărimea buffer-ului de intrare
SET UNIQUE	- stabilește dacă un index va conține toate cheile sau numai cheile unice
SET VIEW	- deschide un fișier VIEW
SET WINDOW	- atașează o fereastră unui cîmp memo

Lista comenzilor SQL disponibile în dBase IV

ALTER TABLE	- înserază coloane într-o tabelă
CLOSE	- încheie accesul dBase la o tabelă
CREATE DATABASE	- creează o nouă bază de date
CREATE INDEX	- creează un index pentru una sau mai multe coloane sau pentru un view
CREATE SYNONYM	- creează un sinonim pentru o tabelă sau pentru un view
CREATE TABLE	- creează o nouă tabelă
CREATE VIEW	- creează o tabelă virtuală (view)
DCHECK	- verifică corespondența dintre o tabelă SQL și un fișier dBase IV
DEFINE	- convertește o bază de date dBase într-o tabelă a bazei de date SQL active
DECLARE CURSOR	- definește un acces dBase la o tabelă
DELETE	- șterge linii dintr-o tabelă sau dintr-un view
DROP DATABASE	- șterge o bază de date
DROP INDEX	- șterge un index
DROP SYNONYM	- șterge un sinonim
DROP TABLE	- șterge o tabelă
DROP VIEW	- șterge un view
FETCH	- sare la un acces dBase la o tabelă la intrarea următoare
GRANT	- stabilește drepturile de acces pentru o tabelă
INSERT	- înserază linii într-o tabelă sau într-un view
LOAD DATA	- importă date într-o tabelă SQL
OPEN	- începe accesul dBase la o tabelă
REVOKE	- șterge (revocă) drepturile de acces pentru o tabelă
ROLLBACK	- anulează înapoi modificările unei tranzacții neîncheiate
RUNSTATS	- actualizează intrările statistice ale tabeli actuale
SELECT	- lansează o interogare SQL
SHOW DATABASE	- afișează bazele de date existente
START DATABASE	- deschide o bază de date
STOP DATABASE	- închide baza de date activă
UNLOAD DATA	- exportă date dintr-o tabelă SQL
UPDATE	- actualizează datele unei selecții

Planificarea este alfa și omega

OOP - Partea a cincea

În partea finală a acestui curs sînt examinate problemele care apar la planificarea ierarhiilor de obiecte.

Primele trei părți ale acestui curs s-au ocupat cu bazele programării orientate obiect. În partea a patra au fost prezentate tehnicile de construire a unei ierarhii. În partea finală a cursului va fi demonstrat modul de cuplare al unor noi obiecte într-o ierarhie.

În partea a patra a cursului ne-am concentrat atenția asupra modului de realizare a Unit-ului **BS_Tools**. Ultimul tip de obiecte adăugat a fost obiectul **BS_Intr**, destinat introducerii confortabile de șiruri de caractere. Problema era să "adaptăm" Unit-ul astfel încît caracteristicile moștenite să funcționeze fără probleme. În anumite condiții, însă, nu aceasta este cea mai potrivită cale de a-ți atinge scopul. Din acest motiv nu trebuie să se programeze orbește. Dacă, de exemplu, în ierarhia de obiecte **BS_Tools** este necesar un tip de obiecte pentru introducerea de șiruri de caractere, atunci este recomandabilă analizarea în prealabil a ierarhiei existente.

Cunoscînd-o se poate găsi ușor cel mai potrivit punct de cuplare. Pe de altă parte se determină cu precizie caracteristicile noului tip de obiecte.

Dacă se observă cu atenție **BS_Tools** se pot observa următoarele caracteristici:

În primul rînd există doi parametri **X** și **Y** cu ajutorul cărora se stabilește poziția de ieșire. În al doilea rînd există diferite obiecte care declară metode virtuale. În al treilea rînd metoda specifică obiect **Inlt** pune la dispoziție un constructor pentru inițializarea metodelor vir-

tuale. În al patrulea rînd au fost declarați pointeri care permit tratarea dinamică a obiectelor. Și în al cincilea rînd pentru o tratare impecabilă au fost declarați și destructorii necesari. Analiza poate continua. Pentru moment ne ajung însă aceste cinci puncte.

Ar fi de bun simț ca parametrii pentru poziția de ieșire să fie utilizați și pentru intrări. Întrucît nu se cunoaște însă dinainte, dacă și în ce condiții sînt activate metodele virtuale, acest lucru trebuie planificat dinainte. Metoda **Inlt** trebuie așadar scrisă ca și constructor. Pointerul **ReadStrPtr** trebuie să permită tratarea dinamică a obiectului obținut. Mai trebuie verificat dacă este necesar un destructor propriu, sau dacă este suficient cel din "BS_Semn".

După aceste considerații preliminare putem trece la programare. La început nu ne vom atinge de Unit-ul **BS_Tools**. Vom declara un nou Unit cu numele "Inputs", pentru a semnaliza faptul că în el vor fi definite obiecte de intrare. Obiectul **BS_Intr** disponibil din **BS_Tools** nu are nici o semnificație pentru **Inputs**.

Obiectul mamă pentru toate obiectele de intrare este **ReadStr**. El este definit ca urmaș al obiectului **BS_String**. Pentru a nu se obține mesaje de eroare de la compilator, nu trebuie uitată instrucțiunea **Uses BS_Tools** (linia 5).

Ne-am putea declara mulțumiți cu acest rezultat, dar se observă imediat că nu s-a lucrat cu suficientă atenție pentru ca și metodele moștenite **Draw**, **Delete**, **Move**, **GetStr** și **Done** să funcționeze fără greșală. Dacă se compară **ListIngul 1** cu **ListIngul 2** se observă imediat unde s-au făcut corecții (liniile 32, 35 și 96). Și din nou nu

ne-am atins de **BS_Tools**. Atît doar că parametrii operativi au fost transmiși obiectelor. Acum funcționează și metodele moștenite.

Dar acum devine vizibilă și dilema. Atunci cînd a fost conceput **BS_Tools** nu s-a dat o considerație deosebită modului de tratare al octeților atributelor în memoria de împrăștiere a imaginii. Astfel un șir (string) este mutat corect și exact, dar nu aceleași lucru se întîmplă și cu atributele sale. De data aceasta eroarea nu se află în modul de dezvoltare al ierarhiei de obiecte. Ba mai mult la planificarea lui **BS_Tools** s-au omis condițiile de specificare ale limitelor (marginilor). În acest caz concret obiectele dezvoltate pentru ieșirea pe ecran nu funcționează corect.

Este evidentă necesitatea de a concepe din nou Unit-ul **BS_Tools**. Erorile se regăseau mai ales în insuficiența previziunilor privind ansamblul de caracteristici ce trebuie realizate de obiectele lui **BS_Tools**. Dacă se dorește elaborarea unei ierarhii de obiecte, atunci trebuie dinainte știut ce anume trebuie să facă și cum trebuie să facă acele obiecte. Astfel pot fi prevăzute problemele posibile și pot fi stabilite detaliile de funcționare.

Prima soluție deplin valabilă este prezentată în **Listing 3**. Tipul de obiecte **BS_Intr** a fost îndepărtat. În locul acestuia este folosită o variantă îmbunătățită a tipului **BS_Read**. Pentru o tratare corectă a atributelor în obiectul **BS_Semn** au fost introduse variabilele **ZAttr** și **LAttr** ca parametri pentru afișarea și ștergerea de caractere individuale. În afară de aceasta au fost adaptate corespunzător și **Draw** și **Delete**.

--->

Unit Inputs;

Interface

Uses Crt, BS_Tools;

Type

```

ReadStrPtr = ^ReadStr;

ReadStr = Object (BS_String)
  Prompt : String;
  MaxLen : Integer;
  PAttr, SAttr : Byte;
  Constructor Init (xPos,yPos : Integer;
    PrStr : String);
  Procedure SetAttr (PromptAttr, StrAttr : Byte);
  Procedure SetLen (L : Integer);
  Function Intrare (St : String) : String;
End;
    
```

Implementation

```

Constructor ReadStr.Init (xPos,yPos : Integer;
  PrStr : String);
Begin
  X := xPos;
  Y := yPos;
  Prompt := PrStr;
  PAttr := Yellow + 16 * Blue;
  SAttr := PAttr;
  MaxLen := 255;
End;

Procedure ReadStr.SetAttr (PromptAttr, StrAttr : Byte);
Begin
  PAttr := PromptAttr;
  SAttr := StrAttr;
End;

Procedure ReadStr.SetLen (L : Integer);
Begin
  If L < 256 Then
    MaxLen := L Else MaxLen := 255;
End;

Function ReadStr.Intrare (St : String) : String;
Var
  OfcBws, SegBws : Word;
    
```

Unit Inputs;

Interface

Uses Crt, BS_Tools;

Type

```

ReadStrPtr = ^ReadStr;

ReadStr = Object (BS_String)
  Prompt : String;
  MaxLen : Integer;
  PAttr, SAttr, OldAttr : Byte;
  Constructor Init (xPos,yPos : Integer;
    PrStr : String);
  Procedure SetAttr (PromptAttr, StrAttr : Byte);
  Procedure SetLen (L : Integer);
  Function Intrare (St : String) : String;
End;
    
```

Implementation

```

Constructor ReadStr.Init (xPos,yPos : Integer;
  PrStr : String);
Begin
  X := xPos;
  Y := yPos;
  Prompt := PrStr;
  Str := Prompt;
  PAttr := Yellow + 16 * Blue;
  SAttr := PAttr;
  OldAttr := TextAttr;
  MaxLen := 255;
End;

Procedure ReadStr.SetAttr (PromptAttr, StrAttr : Byte);
Begin
  PAttr := PromptAttr;
  SAttr := StrAttr;
End;

Procedure ReadStr.SetLen (L : Integer);
Begin
  If L < 256 Then
    MaxLen := L Else MaxLen := 255;
End;
    
```

OldAttr : Byte;
 PozIni, PozAct : Integer;
 Ch : Char;

Const

Sfirsit : Set Of Char = [Chr(13)];

Begin

```

Intrate := St;
GotoXY (X,Y);
OldAttr := TextAttr;
TextAttr := PAttr;
Write (Prompt);
PozIni := X + Length(Prompt);
PozAct := PozIni;
GotoXY (PozIni,Y);
TextAttr := SAttr;
Write (St);
GotoXY (PozIni,Y);
Repeat
  Ch := ReadKey;
  If Ch = 50 Then Begin
    Ch := ReadKey;
    Case Ch Of
      'K': {stinga}
        If PozAct > PozIni Then Dec (PozAct);
      'M': {dreapta}
        If PozAct < PozIni + MaxLen Then
          Inc (PozAct);
    End;
    GotoXY (PozAct,Y);
  End
Else Begin
  If (PozAct - PozIni) < MaxLen Then Begin
    Write (Ch);
    Inc (PozAct);
    St := Copy (St,1,PozAct-PozIni+1) + Ch +
      Copy (St,PozAct-PozIni+2,Length (St));
  End;
End;
Intrate := St;
Until Ch In Sfirsit;
TextAttr := OldAttr;
End;
End;
    
```

{Listing 1}

{Prima prezentare incorecta a Unit-ului Inputs}

PrStr : String);

Begin

```

  X := xPos;
  Y := yPos;
  Prompt := PrStr;
  Str := Prompt;
  PAttr := Yellow + 16 * Blue;
  SAttr := PAttr;
  OldAttr := TextAttr;
  MaxLen := 255;
End;

Procedure ReadStr.SetAttr (PromptAttr, StrAttr : Byte);
Begin
  PAttr := PromptAttr;
  SAttr := StrAttr;
End;

Procedure ReadStr.SetLen (L : Integer);
Begin
  If L < 256 Then
    MaxLen := L Else MaxLen := 255;
End;
    
```

```

Function ReadStr.Intrare (St : String) : String;
Var
  OfsBws, SegBWS : Word;
  PozIni, PozAct, i : Integer;
  Ch : Char;
Const
  Sfsrit : Set Of Char = [ Chr(13) ];
  Set1 : Set Of Char = ['.',','];
Begin
  Intrare := St;
  GotoXY (X,Y);
  TextAttr := PAttr;
  Write (Prompt);
  PozIni := X + Length(Prompt);
  PozAct := PozIni;
  GotoXY (PozIni,Y);
  TextAttr := SAttr;
  Write (St);
  GotoXY (PozIni,Y);
  Repeat
    Ch := ReadKey;
    If Ch = #0 Then Begin
      Ch := ReadKey;
      Case Ch Of
        'K': {stinga}
      End;
    End;
  End;

```

```

If PozAct > PozIni Then Dec (PozAct);
'M': {dreapta}
If PozAct < PozIni + MaxLen Then
  Inc (PozAct);
End;
GotoXY (PozAct,Y);
End
Else Begin
  If (PozAct - PozIni) < MaxLen Then Begin
    If Ch In Set1 Then Begin
      Write (Ch);
      Inc (PozAct);
      St := Copy(St,1,PozAct-PozIni + 1) + Ch +
        Copy(St,PozAct-PozIni + 2,Length(St));
    End;
  End;
  End;
  Intrare := St;
  Until Ch In Sfsrit;
  Str := Prompt + St;
  TextAttr := OldAttr;
End;
{Listing 2}
{Unit-ul Inputs adaptat pentru BS_Tools}

```

```

Unit Bs_Tools;
Interface
Uses Crt;
Const
  SegBWS = $B800; { Color }
  {SegBWS = $B000; Mono }
  OfsBWS : Word = $0000;
Terminator : Set Of Char = {Chr(0),Chr(9),
  Chr(10),Chr(12),Chr(13)};
Set1 : Set Of Char = ['.',','];
Type
  BS_LocPtr = ^BS_Loc;
  BS_SemnPtr = ^BS_Semn;
  BS_StringPtr = ^BS_String;
  BS_ReadPtr = ^BS_Read;
  BS_Loc = Object
    X,Y : Integer;
    Procedure Init (xPos, yPos : Integer);
    Function GetX : Integer;
    Function GetY : Integer;
  End;
  BS_Semn = Object (BS_Loc)
    Semn : Char;
    ZAttr, LAttr : Byte;
    Constructor Init (xPos, yPos : Integer,
      Ch : Char; Attr : Byte);
    Procedure Draw; Virtual;
    Procedure Delete; Virtual;
    Procedure Move (xNou, yNou : Integer);
    Procedure SetLAttr (LA : Byte);
    Destructor Done; Virtual;
  End;
  BS_String = Object (BS_Semn)
    Str : String;
    Constructor Init (xPos, yPos : Integer,
      S : String; StrAttr : Byte);
    Procedure Draw; Virtual;

```

```

Function GetStr : String;
Procedure Delete; Virtual;
End;
BS_Read = Object (BS_String)
  Prompt : String;
  MaxLen : Integer;
  OldAttr : Byte;
  Constructor Init (xPos, yPos : Integer,
    PrStr : String);
  Procedure SetLen (L : Integer);
  Function Intrare (St : String) : String;
  End;
Implementation
Procedure BS_Loc.Init (xPos, yPos : Integer);
Begin
  X := xPos;
  Y := yPos;
End;
Function BS_Loc.GetX : Integer;
Begin
  GetX := X;
End;
Function BS_Loc.GetY : Integer;
Begin
  GetY := Y;
End;
Constructor BS_Semn.Init (xPos, yPos : Integer,
  Ch : Char; Attr : Byte);
Begin
  BS_Loc.Init (xPos,yPos);
  Semn := Ch;
  ZAttr := Attr;
  LAttr := TextAttr;
End;
Procedure BS_Semn.Draw;
Begin
  OfsBWS := 2*(X-1) + 160*(Y-1);
  Mem[SegBWS : OfsBWS + 1] := ZAttr;
  Mem[SegBWS : OfsBWS] := Ord(Semn);

```

```

End;

Procedure BS_Semn.Delete;
Begin
  ofsBWS := 2*(X-1) + 160*(Y-1);
  Mem[SegBWS : ofsBWS] := 32;
  Mem[SegBWS : ofsBWS + 1] := LAttr;
End;

Procedure BS_Semn.Move (xNou, yNou : Integer);
Begin
  Delete;
  BS_Loc.Init (xNou, yNou);
  Draw;
End;

Procedure BS_Semn.SetLAttr (LA : Byte);
Begin
  LAttr := LA;
End;

Destructor BS_Semn.Done;
Begin
End;

Constructor BS_String.Init (xPos, yPos : Integer;
  S : String; StrAttr : Byte);
Begin
  BS_Loc.Init (xPos, yPos);
  Str := S;
  ZAttr := StrAttr;
End;

Procedure BS_String.Draw;
Var i, s : Integer;
  Ch : Char;
Begin
  s := X;
  For i := 1 To Length (Str) Do Begin
    BS_Semn.Init ((i-1) + s, Y, Str[i], ZAttr);
    BS_Semn.Draw;
  End;
  X := s;
  Semn := Str[1];
End;

Function BS_String.GetSt. : String;
Begin
  GetStr := Str;
End;

Procedure BS_String.Delete;
Var i : Integer;
Begin
  i := X;
  Repeat
    ofsBWS := 2*(X-1) + 160*(Y-1);
    If Chr (Mem[SegBWS : ofsBWS]) in Terminator
      Then Begin
        X := i;
        Exit;
      End;
    Bs_Loc.Init(X, Y);
    BS_Semn.Delete;
    Inc(X);
  Until X = 80;
  X := i;
End;

Constructor BS_Read.Init (xPos, yPos : Integer;
  PrStr : String);
Begin
  X := xPos;

```

```

  Y := yPos;
  Prompt := PrStr;
  Str := Prompt;
  ZAttr := Yellow + 16 * Blue;
  OldAttr := TextAttr;
  MaxLen := 255;
End;

Procedure BS_Read.SetLen (L : Integer);
Begin
  If L < 256 Then
    MaxLen := L Else MaxLen := 255;
End;

Function BS_Read.Intrare (St : String) : String;
Var
  PozIni, PozAct : Integer;
  Ch : Char;
  XOld, YOld : Integer;
Begin
  XOld := X;
  YOld := Y;
  Intrare := St;
  BS_String.Init (X, Y, Prompt, ZAttr);
  BS_String.Draw;
  PozIni := X + Length(Prompt);
  PozAct := PozIni;
  BS_String.Init (PozIni, Y, St, ZAttr);
  BS_String.Draw;
  GotoXY (PozIni, Y);
  Repeat
    Ch := ReadKey;
    If Ch = #0 Then Begin
      Ch := ReadKey;
      Case Ch Of
        'K': {stinga}
          If PozAct > PozIni Then Dec (PozAct);
        'M': {dreapta}
          If PozAct < PozIni + MaxLen Then
            Inc (PozAct);
      End;
      GotoXY (PozAct, Y);
    End
  Else Begin
    If (PozAct - PozIni) < MaxLen Then Begin
      If Ch In Set1 Then Begin
        BS_Semn.Init (PozAct, Y, Ch, ZAttr);
        BS_Semn.Draw;
        Inc (PozAct);
        GotoXY (PozAct, Y);
        St := Copy(St, 1, PozAct - PozIni - 1) + Ch +
          Copy(St, PozAct - PozIni + 1, Length(St));
      End;
    End;
  End;
  Intrare := St;
  Until Ch In Terminator;
  Str := Prompt + St;
  TextAttr := OldAttr;
  BS_Loc.Init (XOld, YOld);
End;

```

{Listing 3}
 {Prezentarea completa a Unit-ului BS_Tools}

<pre> Program DoneTest; Uses BS_Tools; Var A1 : BS_ReadPtr; S : String; Begin Str (MemAvail:8,S); WriteLn ('Heap liber : ' + S); </pre>	<pre> New (A1,Init(5,5,'Intrare:')); Dispose (A1,Done); Str (MemAvail:8,S); WriteLn ('Heap liber : ' + S); End. {Listing 4} {Programul permite sa se controleze corecta tratare a memoriei} </pre>
---	---

→

Și rutinele de intrare din "Inputs" (Listing 2) și **BS_Tools** (Listing 3) trebuie comparate. Modificările de dataliu lămuresc mai buna utilizare a variabilelor anterior amintite. Au fost gândite și posibilele extensii. Astfel sînt prezentate tipurile de seturi de caractere Toate_Semnele, NumReal și NumOnly.

Înainte de a continua extensia consecventă a lui **BS_Tools**, trebuie verificat dacă metoda Done a obiectului **BS_Semn** eliberează într-adevăr întreaga memorie pentru următorul obiect. Un mic program (Listing 4) ne va lămuri imediat. Cu toate că un obiect de tipul **BS_Semn** ocupă un alt număr de octeți de memorie în heap, decât de exemplu un obiect de tipul **BS_Read**, metoda Done eliberează o cantitate corectă de memorie.

Acum se poate trece la realizarea de obiecte pentru introducerea de numere. Soluțiile corespunzătoare devin cît se poate de simple datorită puterii metodei **BS.Read_Intr**. Poate fi extins afit **BS_Tools**, dar poate fi realizat și un alt Unit. Din motive de spațiu vom prefera, în continuare, ultima variantă. Cu ajutorul editorului însă noul cod poate fi inserat oricînd în codul lui **BS_Tools**.

Scrierea unui nou obiect, să-i spunem **Read_Real**, nu mai pune probleme. În acesta pot fi utilizați doi noi parametri **Min** și **Max** care să permită specificarea intervalului de valori în care este așteptată

specificarea valorii de intrare. Față de obiectul mamă, constructorul va trebui astfel modificat încît să permită introducerea de valori numerice. Prin intermediul unei proceduri, să-i spunem **SetMin-Max**, aceste limite vor trebui să poată fi modificate.

Astfel se poate continua și se pot elabora obiecte și pentru introducerea celorlalte tipuri de valori numerice. Obiectele **ReadInt**, **ReadLongInt** sau **ReadByte** pot fi programate toate ca urmași ai obiectului **BS_Read**.

Dacă v-ați chnuit pînă în acest moment și ați introdus în calculator Unit-ul **BS_Tools** și probabil l-ați mai și extins, atunci probabil că nu veți fi mulțumiți cu posibilitățile de editare oferite de **BS_Read**. Nici nu ne-am putut permite în condițiile date să încercăm să ridicăm cît se poate de mult acest confort.

Este mai bine cu OOP sau nu? Așa, sau asemănător, își pun întrebarea utilizatorii care au auzit de programarea orientată obiect, dar care nu știu ce se ascunde în spațiile ei.

Versiunile 5.5 și 6.0 ale lui Turbo Pascal nu obligă pe nimeni să programeze orientat obiect. Și pe mai departe se poate programa în maniera tradițională. Este și bine așa.

Limbajele orientate obiect (Pascal sau C) au fost create cu scopul sporirii eficienței și transparenței programării pentru a putea stăpîni mai bine complexitatea mereu sporită a programelor.

Cu ajutorul tehnicii de programare cu subprograme și Unit-uri se puteau concepe programe frumos structurate care să rezolve sarcini complexe. La conlucrarea lor trebuia să se țină seama de multe condiții de limite astfel încît apăreau probleme de potriviri și interschimbarea unor module era limitată sau chiar imposibilă.

De exemplu este foarte greu să se elaboreze Unit-uri generale și universale pentru o suprafață utilizator, care să poată fi utilizată fără probleme în cele mai diferite aplicații. Biblioteca de obiecte Borland "Turbo-Vision" este un prim pas de a sprijini programarea unor aplicații complexe cu metode orientate obiect.

Utilizarea tehnicilor orientate obiect are sens mai ales la scrierea unor aplicații complexe. Cu ajutorul structurilor poliforme se obține o transparență sporită a limbajului. În exemplul nostru **BS_Tools** dă metodele **Draw**, **Delete** și **Move** pentru toți urmașii lui **BS_Semn**. Toate obiectele utilizează aceeași însărcinare pentru a realiza acțiuni diferite.

Decizia de a utiliza sau nu programarea orientată obiect depinde de aplicație. Dacă s-a decis utilizarea acestei tehnici moderne de programare atunci ea trebuie folosită consecvent și cuprinzător.

(R.M.)

Programarea logică

Turbo-Prolog - Partea întâi

Limbajul de programare Prolog este un instrument excelent pentru rezolvarea de probleme logice. În prima parte, a unei serii de trei episoade, ne vom ocupa de câteva din principiile de bază.

Numele Prolog este o prescurtare a sintagmei PROgramming în LOGic. Acest limbaj de programare a fost elaborat în 1972 de A. Comlerauer și P. Rousel pentru scrierea de aplicații în domeniul inteligenței artificiale. Cu toate că atunci limbajul Lisp era greu de învățat, la vremea respectivă a fost preferat limbajului Prolog.

În jurul anului 1980 dezvoltarea intensivă a sistemelor expert a necesitat sisteme Prolog puternice. Astfel au apărut diferite dialecte ale acestui limbaj. Până în ziua de astăzi nu există încă un standard unitar. Ca fundamente orientative pentru majoritatea implementărilor Prolog este folosit cvasistandardul formulat de Clocksin și Mellish.

Programarea într-un limbaj de programare uzual, cum sînt Pascal, C, Fortran, etc., constă în principal în formularea de proceduri, funcții și instrucțiuni. Trebuie deci să i se spună exact calculatorului ce și cum trebuie făcut ceva. În cazul limbajului Prolog, programatorul are doar sarcina de a descrie problema de rezolvat într-o formă apropiată limbajului natural. Calea de rezolvare este lăsată în seama sistemului Prolog, care va încerca să rezolve problema prin algoritmi mai mult sau mai puțin eficienți. Pentru a accelera obținerea unei soluții, programatorul poate da unele indicații ajutoare.

Limbajul Prolog poate fi utilizat pentru toate aplicațiile care necesită deducții formale. Problemele numerice pot fi programate mai

puțin eficient, deoarece structura Prolog nu este recomandată pentru acest scop.

Sistemele expert care simulează reacțiile unui specialist prin intermediul unui mecanism deductiv, reprezintă un domeniu în care Prolog poate fi utilizat cu rezultate excelente. La acestea se adaugă sistemele informaționale în limbaj natural, care permit și utilizatorilor cu mai puține cunoștințe tehnice să descrie o problemă pentru a fi înțeleasă de calculator. Un domeniu larg de utilizare este și cel al jocurilor și simulărilor care necesită deducții formale.

Turbo-Prolog conține majoritatea caracteristicilor Prolog stabilite de Clocksin și Mellish. Spre deosebire de sistemele Prolog anterioare care erau interpretoare, Turbo Prolog este compilator. Programele obținute sînt astfel mai rapide decît cele obținute pe PC-uri cu alte sisteme Prolog.

Acest compilator conține o serie de predicate standard pentru funcții (prelucrare șiruri, acces date, grafică, tehnica ferestrelor și ieșiri sonore). Este posibilă cuplarea unor secvențe de program elaborate cu un limbaj procedural. Spre deosebire de versiunile Prolog mai vechi, pentru un control mai eficient pot fi declarate variabile.

Cu toate că printre domeniile speciale ale limbajului Prolog nu se numără și calculul numeric există și predicate pentru lucrul cu aritmetica cu virgulă fixă și cu virgulă flotantă. În plus spre deosebire de celelalte sisteme Prolog este disponibilă o suprafață utilizator tipică Turbo.

Un program Turbo Prolog poate conține următoarele secțiuni:

- - Comenzi compilator
- - Domains
- - Database
- - Predicates
- - Goal și
- - Clauses

Într-un program nu trebuie să fie conținute toate aceste secțiuni dar ordinea în care au fost enumerate este obligatorie. Secțiunile individuale sînt introduse printr-un cuvînt cheie corespunzător.

Pentru a se putea verifica faptul că toate obiectele corespund tipului declarat, în partea de declarații "domains" trebuie declarate toate tipurile de date.

Turbo - Prolog cunoaște următoarele tipuri standard de domenii predefinite:

- char** - caracter încadrat de apostroafe
- integer** - numere întregi între -32768 și 32767
- real** - numere în virgulă fixă între 1E-307 și 1E+308
- string** - șir de caractere încadrat de ghilimele
- symbol** - succesiune de caractere formată din litere, cifre și liniuțe de subliniere, primul caracter trebuind să fie o literă mică.
- file** - nume simbolic de fișier

Faptele pot fi descrise prin relații simbolice. Următoarele declarații reprezintă fapte care pot fi înțelese de Turbo-Prolog:

```
frati (ion, petre).
barbat (alexandru).
tata (ion, alexandru).
```

În Prolog faptele mai sînt denumite și clauze. În limba română faptele anterioare ar putea fi exprimate astfel:

```
Ion și Petre sînt frați.
```

Alexandru este bărbat.
Ion este tatăl lui Alexandru.

Relațiile clauzelor anterior enunțate sînt denumite predicat, iar parametrii clauzelor sînt denumiți argumente. Un predicat poate conține oricîte argumente, dar poate și să nu conțină niciunul. În Turbo-Prolog un predicat trebuie utilizat întotdeauna cu același număr de argumente.

Fiecare clauză se sfîrșește cu un punct, iar argumentele ei sînt separate cu virgule. Numele de predicat încep întotdeauna cu literă mică. Celelalte caractere pot fi atît litere mici cît și litere mari, cifre sau linii de subliniere.

În secțiunea predicates trebuie definite toate predicatetele. Pentru a declara un predicat trebuie specificate numele său și tipul argumentelor sale:

```
predicates
  data_nasterii (symbol,
                integer, integer, integer)
```

Predicatul anterior ar fi fost mai inteligibil dacă pentru fiecare argument s-ar fi definit un tip de domeniu explicit:

```
domains
  nume = symbol
  zi, luna, an = integer
predicates
  data_nasterii (nume, zi, luna, an)
```

Într-un program Prolog pot apărea mai multe secțiuni predicates. În secțiunea clauses sînt specificate clauzele și regulile pentru predicatetele anterior definite.

```
domains
  nume = symbol
  zi, luna, an = integer
predicates
  data_nasterii (nume, zi, luna, an)
clauses
  data_nasterii (ion, 15, 5, 1969).
  data_nasterii (gelu, 26, 10, 1940).
  data_nasterii (vasile, 11, 9, 1969).
```

Clauzele și regulile constituie așa-numita bază de date (database) a unui program Prolog, căreia i se pot adresa întrebări în timpul execuției.

Dacă programul de mai sus este lansat în execuție cu comanda RUN, sistemul cere introducerea unui goal extern, aceasta însemnînd că programatorul trebuie să pună o întrebare sistemului Prolog.

De exemplu:

Goal: data_nasterii (ion, 15, 5, 1969)

Turbo-Prolog răspunde cu True.

De modul în care ajunge sistemul la o soluție ne vom ocupa mai tîrziu. Un goal poate fi inserat și direct în interiorul unui program, caz în care vorbim despre un goal intern.

Într-o clauză sau într-un goal pot fi utilizate și variabile pentru a specifica relațiile unui obiect. Variabilele necesită rezervare de spațiu iar numele lor trebuie să înceapă cu o literă mare putînd continua cu litere, cifre sau linii de subliniere.

Un nume de variabilă nu poate conține mai mult de 250 de caractere și pentru o mai bună lizibilitate trebuie să facă o referire la obiectul reprezentat. Prin utilizarea variabilelor într-un goal devine posibilă interogarea de informații din baza de date și nu doar obținerea unor răspunsuri da-nu.

Cu ajutorul goal-ului următor, din baza de date anterior creată pot fi interogate anumite date de naștere.

Goal: data_nasterii (Nume,Zi,Luna,An)

Sistemul va răspunde astfel:

```
Nume=ion,Zi=15,Luna=5,An=1969
Nume=gelu,Zi=26,Luna=10,An=1940
Nume=vasile,Zi=11,Luna=9,An=1969
3 Solutions
```

De îndată ce sistemul Prolog atribuie o valoare unei variabile (de ex. Nume = ion), vorbim despre ea ca fiind o variabilă legată. Dacă nu i-a fost atribuită încă o valoare, atunci vorbim despre o variabilă liberă. La lansarea în execuție a

programului toate variabilele sînt libere.

Uneori la o interogare destinată unui anumit scop unele argumente sînt lipsite de semnificație. Acestea pot fi înlocuite, în acest caz, prin variabile anonime. Acestea sînt reprezentate prin intermediul unei linii de subliniere și nu li se atribuie o valoare în procesul de soluționare.

Dacă programatorul dorește să afle în ce an sînt născute persoanele din baza de date, atunci va trebui să formuleze următorul goal:

Goal: data_nasterii (Nume,_,_,An)

Sistemul va răspunde, în acest caz, cu:

```
Nume = ion, An = 1969
Nume = gelu, An = 1940
Nume = vasile, An = 1969
3 Solutions
```

Pînă acum relațiile dintre obiecte au fost date prin clauze. În continuare vom extinde baza de cunoștințe. Prin intermediul regulilor, Turbo Prolog este în stare să deducă noi clauze din cele deja existente în baza de cunoștințe. O regulă este o expresie prin care se stabilește valoarea de adevăr a unei clauze speciale în funcție de una sau mai multe alte clauze. Ea constă dintr-un așa-numit arțet de regulă, urmat de cuvîntul "if", sau de ":-". La urmă sînt exprimate toate condițiile de care depinde luarea deciziei. Condițiile sînt cuplate între ele prin operatori "și" respectiv "sau".

O cuplare și este reprezentată prin "and" sau ";", iar una sau prin "or" sau ":",. Punctul semnifică sfîrșitul. Cu următoarea regulă se poate stabili, de exemplu, dacă două persoane s-au născut în același an:

```
acelasi_an_nastere (Nume1,Nume2):-
  data_nasterii (Nume1,_,_,An),
  data_nasterii (Nume2,_,_,An).
```

Întregul program va arăta astfel:

```
domains
  nume = symbol
  zi, luna, an = integer
predicates
  data_nasterii (nume, zi, luna, an)
  acelasi_an_nastere (nume, nume)
clauses
  data_nasterii (ion, 15, 5, 1969).
  data_nasterii (gelu, 26, 10, 1940).
  data_nasterii (vasile, 11, 9, 1969).
  acelasi_an_nastere (Nume1, Nume2):-
    data_nasterii (Nume1,_,_,An),
    data_nasterii (Nume2,_,_,An).
```

Acum se poate pune, de exemplu, întrebarea:

Goal: `acelasi_an_nastere (ion, X)`

Sistemul va da următorul răspuns:

```
X = ion
X = vasile
2 Solutions
```

Mecanismul de transmitere a valorilor se realizează în Prolog prin unificare. În acest punct acest limbaj de programare se diferențiază de așa-numitele limbaje procedurale. Pentru a înțelege limbajul Prolog, este de o importanță hotărâtoare înțelegerea mecanismului de unificare. El descrie o procedură prin care limbajul Prolog încearcă realizarea unei corespondențe între un termen și o clauză sau un antet de regulă, pentru atingerea unui scop.

Un termen poate fi unificat cu alt termen (poate fi echivalat) dacă:

- 1) ambii termeni apar în predicate cu același număr de argumente, în aceeași poziție;
- 2) ambii termeni sînt de tipuri echivalente (char poate fi echivalent numai cu char, ș.a.m.d.)
- 3) subtermenii corespunzători pot fi de asemenea echivalați.

În cadrul unificării sînt valabile următoarele:

- 1) variabilele libere pot fi unificate cu orice termen care satisface condițiile mai sus menționate. Prin unificare variabila este legată de valoarea termenului.
- 2) constantele pot fi unificate cu ele însele, sau cu variabile libere.

Dacă o variabilă liberă este echivalată cu o constantă, atunci va lua valoarea acelei constante.

- 3) variabilele libere pot fi unificate cu oricare alte variabile. După echivalare ambele vor fi tratate ca o singură variabilă. Dacă în cele din urmă una din cele două este legată de o valoare, atunci și cealaltă va fi instanțiată cu această valoare.
- 4) predicatele pot fi echivalate dacă posedă același nume de relații, numărul argumentelor este același și perechile de argumente pot fi unificate între ele.

În continuare unificarea va fi explicată prin intermediul unui exemplu:

```
domains
  loc, persoana = symbol
  an = integer
  locul_nasterii = nascut (loc,an)
predicates
  date_de_nastere (persoana,locul_nasterii)
clauses
  date_de_nastere (ion,
    nascut(calarasi,1964)).
  date_de_nastere (ana,
    nascut(zimnicea,1968)).
```

Dacă se specifică acum goal-ul extern:

Goal: `date_de_nastere (X,Y)`

atunci sistemul Prolog încearcă să unifice clauzele din goal cu prima clauză din program. Acest lucru este posibil deoarece atât X cât și Y sînt variabile libere și pot fi deci legate de orice alt termen al aceleiași predicat.

În cazul celei de-a doua clauze, din aceleași motive, echivalarea este de asemenea posibilă. Sistemul va furniza deci următorul răspuns:

```
X = ion, Y = nascut(calarasi,1964)
X = ana, Y = nascut(zimnicea,1968)
2 Solutions
```

În cele din urmă să punem sistemului următoarea întrebare:

```
date_de_nastere (X,nascut(zimnicea,Y))
```

Din nou sistemul încearcă o unificare cu prima clauză. Astfel variabila X este legată mai întîi de valoarea "ion" și apoi se încearcă unificarea lui "(calarasi, 1964)" cu "(zimnicea, 1968)".

Cum am remarcat deja, unificarea termenilor este posibilă numai atunci cînd ambii au același nume de relație, același număr de argumente și în plus toți subtermenii pot fi echivalați.

În exemplul anterior deși ambii termeni au același nume de relație, "născut", și același număr de argumente, constanta "calarasi" poate fi unificată numai cu ea însăși sau cu o variabilă liberă; deci în acest caz unificarea nu este posibilă. Procedura de unificare eșuează deci la prima clauză, urmează cea de-a doua. În cazul ei vor fi comparați termenii "date_de_nastere (X,nascut(zimnicea,Y))" și "date_de_nastere (ana, nascut(zimnicea,1968))". Variabila X va fi instanțiată cu valoarea ana. Echivalarea celor doi termeni "nascut(zimnicea,Y)" și "nascut(zimnicea,1968)" are loc cu succes, deoarece este utilizat același nume de relație "nascut" și numărul argumentelor este același.

Constanta "zimnicea" este unificată cu ea însăși, iar variabila Y este instanțiată cu valoarea 1968. Astfel se va obține soluția:

```
X = ana, Y = 1968
1 Solution
```

Noțiunea backtracking desemnează o strategie de căutare specifică. Dacă pe calea pe care s-a pornit nu se ajunge la o soluție, atunci programul se întoarce și caută în altă direcție.

Prin backtracking sînt găsite toate soluțiile unei probleme. Cea mai bună comparație se poate face cu căutarea într-un labirint. Pentru a se face o deplasare dintr-un loc într-alt loc se urmărește un drum exact, atît timp cît nu există bifurcații. Dacă se ajunge la un loc în care există mai multe alternative,

atunci se selectează una dintre posibilități. Dacă se ajunge într-o fundătură, atunci se întoarce la punctul de bifurcare și se alege un drum care n-a fost încercat încă (ș.a.m.d.). Cu această metodă se poate atinge în orice caz scopul, în cazul în care un astfel de drum există (se poate întâmpla și să nu existe un astfel de drum).

Noțiunea backtracking desemnează revenirea la acea poziție în care existau mai multe variante, pentru a porni de acolo pe un alt drum.

Cum este utilizată această metodă în Turbo Prolog va fi înțeles cu ajutorul exemplului următor:

```
domains
  persoana, sport = symbol
predicates
  practica (persoana, sport)
clauses
  practica (gicu, tenis).
  practica (ion, volei).
  practica (radu, tenis).
  practica (nicolae, X) if
  practica (ion, X).
```

Pentru a determina toate persoanele care practică același sport va trebui specificat următorul goal:

```
Goal: practica (Persoana1,Sport),
      practica (Persoana2,Sport),
      Persoana 1 <> Persoana 2.
```

Turbo - Prolog ajunge la soluție cu ajutorul backtracking-ului și unificării pe următoarea cale:

- - sistemul pomește de la primul subgoal:
"practica (Persoana1,Sport)"
- - este cercetată prima clauză, este posibilă unificarea "Persoana1 = gicu" și "Sport = tenis".
- - se trece apoi la al doilea subgoal "practica (Persoana2,Sport)". Variabila "Sport" este deja legată de valoarea "tenis" ceea ce duce la transformarea celui de-al doilea subgoal în "practica (Persoana2, tenis)". Variabila "Persoana2" mai este încă liberă.
- - la depistarea celui de-al doilea scop Prolog pomește din nou de

la prima clauză. Se reușește unificarea "Persoana2 = gicu".

- - acum se încearcă verificarea și a celei de-a treia condiții "Persoana1 <> Persoana2". Deoarece ambele variabile sînt legate de valoarea "gicu", condiția nu se verifică și scopul nu poate fi atins cu această legare a variabilelor.
- - cu aceasta ultima legare de variabilă ("Persoana2 = gicu") este din nou dezlegată și sistemul încearcă unificarea celui de-al doilea subgoal cu cea de-a doua clauză. Acest lucru nu reușește deoarece variabila "Sport" este legată încă de valoarea "tenis".
- - din această cauză se trece la a treia clauză la care se reușește unificarea "Persoana2 = radu". Acum se verifică din nou al treilea subgoal. Deoarece în acest caz "Persoana1 = gicu" <> "Persoana2 = radu" și deci condiția este satisfăcută, s-a găsit o soluție la interogare ("gicu, tenis, radu"). Aceasta va fi și afișată.
- - Deoarece în cazul goal-urilor externe, Turbo-Prolog caută toate soluțiile posibile, prelucrarea programului nu ia încă sfîrșit. Ultima atribuire de variabilă ("Persoana2 = radu") este desfăcută.
- - sînt examinate regulile pentru a patra clauză. Sistemul încearcă s-o acopere cu al doilea subgoal. Pentru a verifica "practica (gicu, tenis)" trebuie să corespundă cu "practica (ion, tenis)". Acest lucru nu este posibil și deci nu există o a doua soluție a acestui subgoal.
- - deoarece au fost încercate toate posibilitățile pentru al doilea subgoal, sînt dezlegate (resetate) și variabilele primului subgoal și urmează unificarea primului scop cu cea de-a doua clauză.
- - acest lucru reușește cu "Persoana1 = ion" și "Sport = volei". Al doilea subgoal se transformă deci în "practica (Persoana2, volei)". Unificarea cu prima clauză nu este posibilă, echivalarea cu cea de-a doua clauză este posibilă, dar nu este satisfăcută cea de a treia condiție ("Persoana1 <> Persoana2"). Unificarea cu a treia clauză nu este nici ea posibilă.

- - mai rămîne de examinat a patra clauză. Pentru a verifica "practica (nicolae,volei)" trebuie să se verifice "practica (ion,volei)". Acest lucru este valabil prin cea de a doua clauză și cum și cea de-a treia condiție este satisfăcută ("Persoana1 = ion" <> "Persoana2 = nicolae") se obține o nouă soluție ("ion, volei, nicolae"), care va fi de asemenea afișată.
- - în mod analog primul goal va fi unificat cu cea de a treia și apoi cu cea de a patra clauză. Apoi sistemul va căuta soluții prin observarea celorlalte subgoal-uri.

Soluția finală va fi:

```
Persoana1=gicu,Sport=tenis,
  Persoana2=radu
Persoana1 = ion, Sport = volei,
  Persoana2 = nicolae
Persoana1 = radu, Sport = tenis,
  Persoana2 = gicu
Persoana1 = nicolae, Sport = volei,
  Persoana2 = ion
```

Comanda "trace" permite execuția pas cu pas. Mesajele ferestrei trace au următoarea semnificație:

- Call:** apelul predicatelor. Sistemul încearcă să îndeplinească scopul clauzei.
- Return:** clauza a putut fi îndeplinită, sistemul trece la următorul subgoal.
- Redo:** backtracking pentru a permite și alte soluții sau dacă nu s-a obținut o soluție cu valorile momentane ale variabilelor.
- Fall:** scopul nu poate fi realizat.

În partea următoare a cursului de programare logică ne vom ocupa de alte principii de bază. Pe lângă fundamentele limbajului de programare Prolog va fi dat ca exemplu și modul de rezolvare al unei probleme logice.

(R.M.)

Arta de a împărți

Partiționarea harddisk-ului

Aproape fiecare posesor de PC cu harddisk a făcut cel puțin odată o partiționare, dar foarte puțini știu exact, ce este și la ce folosește.

Cine nu dorește să-și stocheze datele pe harddisk efectuând de unul singur gestiunea sectoarelor (care date pe care sector se află - la un disc de 20 MByte, aproximativ 40.000 de sectoare de 512 Byte...), poate lăsa aceasta sub controlul și administrarea unui sistem de operare. Dar înainte ca acesta să-și intre în rol, harddiskul trebuie pregătit corespunzător: el trebuie partiționat și formatat corect.

Trebuie definite toate zonele de pe disc care sînt la dispoziția sistemului. Dacă, în plus, la pornirea calculatorului am vrea să pornesc direct de pe harddisk, este neapărată nevoie de "încărcătorul de partiții". Acesta este un program micuț fără de care pornirea "automată" a calculatorului nici nu ar fi posibilă.

Partiția este un domeniu continuu de pe harddisk, care fizic reprezintă toate sectoarele dintre două adrese de harddisk. O adresă de harddisk este formată dintr-un triplet de valori și anume numărul de cilindru, numărul de cap și numărul de sector. Sectoarele de pe o pistă se numerotează totdeauna începînd cu "1", iar capetele de citire-scriere precum și cilindri se numerotează totdeauna începînd cu 0 și terminînd cu (numărul_maxim -1).

Hotărîtor în partiționarea harddisk-ului este așa-numitul "articol de partiționare". El se găsește pe aproape toate harddisk-urile și ocupă întot-

deauna primul sector (absolut) de date (sectorul 1, cilindrul 0, capul 0). Trebuie spus că acesta este singurul articol de pe fiecare harddisk, a cărui poziție este cunoscută. Poziția tuturor celorlalte date, (de exemplu cea a sectorului de boot, a tabelii de alocare și a fișierelor) este variabilă. Poziția lor depinde direct, mai mult sau mai puțin, de datele din "articolul de partiționare".

Mărimea standard a sectorului la harddisk-urile PC-urilor compatibile IBM este de 512 Byte. În ciuda acestei dimensiuni, mai degrabă mici, toate datele necesare articolului de partiționare pot fi depuse ușor pe un singur sector.

După cum se poate observa din figura 1, articolul de partiționare se compune din 3 părți. La început se află încărcătorul de partiții, la deplasamentul constant 446 (1BC hex) se află tabela de partiții (de 64

Byte) iar la sfîrșit se află marcajul de date de sistem (2 Byte - anume "55AA" hex).

Acest marcaj nu se găsește doar în articolul de partiționare ci și în articolul de boot, și ocupă întotdeauna ultimii doi octeți.

Grație acestui marcaj, programele sînt în stare să determine repede și simplu, dacă datele din cadrul acestui sector sînt - sau nu - interpretabile conform așteptărilor.

Dacă de pildă la un articol de partiționare, altfel bun, se șterge acest marcaj, atunci, după pornirea sistemului de pe dischetă, sistemul de operare DOS nu poate accede harddisk-ul. Nici pornirea sistemului de pe harddisk nu se poate face în aceste condiții.

Încărcătorul de partiții este un program în limbaj mașină, lung de aproximativ 220 octeți, care joacă

Adresa harddisk: Cilindru 0, Pista 0, Sector 1						Format hexa
Offset 0						Încărcătorul de partiții
FA33C08E	D0BC007C	8BF45007	501FFBFC	BF0006B9	0001F2A5	Codul încărcătorului
EA1D0600	00BEBE07	B304803C	80740E80	3C00751C	83C610FE	de partiții cu mesaje
CB75EFCF	188B148B	4C028BEE	83C610FE	CB741A80	3C0074F4	de eroare în format
BE8B06AC	3C00740B	56BB0700	B40ECD10	5EEBF0EB	FEBF0500	ASCII.
BB007CB8	010257CD	135F730C	33C0CD13	4F75EDBE	A706EBD3	(lung. cca. 230 byte)
BECD06BF	FE7D813D	55AA75C7	8BF5EA00	7C0000U	ngul	
tige	Par	titi	onst	abel	le0F	
ehle	r be	im L	aden	des	Bet	
rieh	ssys	tems	0Kei	n Be	trie	
bssy	stem	vor	hand	en00	0000	Domeniul neutilizat
0000	0000	0000	0000	0000	0000	este completat cu 0
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	00			
Offset 446						Tabela de partiții
80010100	063F2027	20000000	E03F0100			4 intrări în tabelă
00000128	053F2072	00400100	00580200			a câte 16 byte.
0000	0000	0000	0000			
0000	0000	0000	0000			
Offset 510						Identificator sistem
55AA						

Fig. 1. Alcătuirea unui articol de partiționare de 512 Byte, în format hexa

un rol scurt, dar important, la fiecare pornire a calculatorului de pe harddisk (pornire "la rece" sau "la cald"). O descriere generală a pornirii sistemului va ajuta la o înțelegere mai bună a fenomenului.

De fiecare dată când un PC este pornit sau când s-a apăsata tasta RESET, calculatorul se inițializează. El se pune deci într-o stare inițială definită. Pentru aceasta procesorul execută rutinele POST (Power On Self Test) memorate în ROM. POST este o componentă fixă, care se găsește, pe toate PC-urile, în BIOS (Basic Input Output System) și răspunde de faptul ca toate unitățile externe să fie inițializate și aduse în stare operațională (din punct de vedere al procesorului). Rutinele acestea cunosc toate cartelele de comandă (de exemplu "Timer", "Controller de harddisk", "Controller de întreruperi" etc.). Ele verifică funcționalitatea fiecărei componente pentru ca apoi să le aducă într-o stare inițială definită.

În momentul în care calculatorul este pe deplin operațional, ceea ce înseamnă că toate resursele (cum ar fi RAM, harddisk-ul sau interfețele) sînt operaționale, intră în acțiune așa-numitul "încărcător BOOTSTRAP". Dacă în acest moment în unitatea A se află o dischetă, acesta citește primul sector de pe ea (0,0,1) în memoria de bază (la adresa "7C00 hex") și predă controlul asupra procesorului, fără nici o altă verificare, codului citit de pe disc. Dacă este vorba de articolul de boot al unei dischete sistem, atunci acest sector pornește procedura de încărcare, iar încărcătorul de partiții nu mai intră în acțiune. Deoarece dischetele nu pot fi partiționate, primul lor sector va conține întotdeauna articolul de boot. Deci, din punct de vedere logic, o dischetă este de fapt o partiție de sine stătătoare, conținând toate componentele specifice necesare sistemului de operare, dintre care primul este totdeauna articolul de boot.

Dacă în unitate nu se află nici o dischetă, încărcătorul de BOOTSTRAP citește articolul de partițio-

nare în memoria de bază (la adresa sa fixă "7C00 hex") și predă încărcătorului de partiții controlul asupra procesorului. Acesta, mai întâi, se copiază pe el însuși, cu tot cu tabela de partiții și marcajul de date sistem de la adresa "7C00 hex" la adresa "0600 hex" (în memoria de bază), astfel încît zona tampon de la adresa "7C00 hex" să se elibereze pentru alte date. În continuare, el caută în tabela de partiții (care se află la sfîrșitul articolului de partiționare), o așa-nu-

Articol de BOOT
F.A.T. 1
.
.
F.A.T. n
Director rădăcină
Domeniu de date

Fig. 2. Construcția schematică a unei partiții DOS

mită "partiție activă". În momentul în care în tabelă găsește o înregistrare cu caracteristicile căutate, încărcătorul de partiții extrage de aici adresa de început (pe harddisk) a partiției respective, preluînd exact un sector de la acea adresă. Acest sector va fi depus în memoria de bază, la adresa "7c00 hexa". Dacă acest sector conține marcajul de date sistem, atunci încărcătorul de partiții presupune că este vorba despre articolul de boot al partiției și îi predă acestuia controlul asupra procesorului.

Dacă marcajul de date sistem lipsește sau nu există partiție activă în tabelă, atunci încărcătorul de partiții dă un mesaj corespunzător la consolă (vezi și fig. 1) și apoi "oprește" calculatorul (lăsînd procesorul să execute o buclă infinită).

Predînd controlul articolului de boot al partiției active din tabela de partiții, încărcătorul de partiții inițiază încărcarea sistemului de operare care se află în cadrul acelei partiții. În cadrul partiției se folosește totdeauna un tip specific de gestiune a datelor, caracteristic unui anumit sistem de operare.

Din momentul în care articolul de boot obține controlul, între porni-

rea sistemului de operare de pe dischetă și pornirea de pe harddisk nu există diferențe principiale. Funcție de datele înglobate în el, stabilește poziția directorului rădăcină și verifică dacă există acolo cele două fișiere constitutive ale sistemului de operare, IBM-BIO.COM și IBMDOS.COM (respectiv, la MS-DOS, fișierele corespunzătoare: IO.SYS și MSDOS.SYS). Dacă le găsește pe amîndouă, atunci articolul de boot calculează poziția de început a zonei de date a partiției pe harddisk și încarcă de acolo fișierul IBM-BIO.COM în memoria de bază. Acesta conține administrarea fișierelor, încarcă IBMDOS.COM, prelucrează comenzile din CONFIG.SYS și încarcă la sfîrșit procesorul de comenzi COMMAND.COM. Acesta la rîndul său execută comenzile din fișierul AUTOEXEC.BAT, după care, în fine, pe ecran se afișează prompter-ul obișnuit - sistemul a pornit.

Și fișierul IBMBIO.COM citește articolul de partiționare în memoria de bază, parcurgînd tabela de partiții cu scopul de a disponibiliza, prin intermediul sistemului de operare, toate unitățile logice DOS (începînd cu C:).

Nu trebuie să mire faptul că primele două înregistrări din directorul rădăcină al unei partiții DOS încărcabile conțin întotdeauna date referitoare la cele două fișiere amintite ale sistemului de operare și că ele se află întotdeauna la începutul domeniului de date. Motivul este că articolul de boot, avînd de regulă doar 512 octeți, poate calcula, cu ajutorul datelor sale despre partiție, poziția exactă de început a directorului rădăcină și a zonei de date, dar, datorită mărimii sale reduse, el nu poate să găsească direct înregistrări de fișiere în directoare sau chiar fișiere din cadrul partiției.

Pentru completare vom detalia pe scurt structura unei partiții DOS. Excepțînd raportul de mărimi al celor patru componente importante, fiecare partiție DOS are aceeași structură (vezi fig. 2):

- - articolul de boot
- - tabela de alocare a fișierelor
- - directorul rădăcină
- - zona de date

Primul este întotdeauna articolul de boot, care ocupă, de regulă, un sector. El este urmat de tabela de alocare a fișierelor (File Allocation Table - FAT) și copiile de siguranță ale acesteia. FAT reprezintă un fel de plan de ocupare pentru administrarea fișierelor, iar mărimea ei depinde direct de mărimea totală a partiției. Directorul rădăcină ocupă de regulă 32 de sectoare la hard-disk-uri, oferind astfel loc pentru 512 înregistrări de fișiere (sau sub-directoare). În cazul dischetelor, de regulă, numărul înregistrărilor din fișierul rădăcină este mai mic.

În sfârșit, zona de date cuprinde atât toate fișierele cât și toate subdirectoarele (care, sub DOS, sînt administrate ca niște fișiere).

Să revenim acum la cea mai importantă parte a articolului de partiționare: tabela de partiții. În general ea are o mărime de 64 Byte și se află întotdeauna imediat înaintea marcajului de date sistem, la sfârșitul înregistrării. Tabelul oferă loc pentru exact patru înregistrări a 16 octeți. Fiecare înregistrare reprezintă datele cadru despre o partiție închisă în sine. Acest lucru înseamnă că un singur harddisk poate găzdui pînă la patru sisteme de operare diferite, fiecare cu propria lui administrare de date. Prin simpla mutare a indicatorului de "partiție activă" de la o înregistrare la alta în tabelă, încărcătorul de partiții va fi determinat să încarce, la o nouă pornire a sistemului, sistemul de operare din partiția nou aleasă. Se presupune că partiția este boot-abilă, deci că se poate încărca un sistem de operare de acolo.

Fig. 3 arată structura unei înregistrări din tabelă, identică pentru toate înregistrările de acest fel. Fiecare înregistrare este compusă din următoarele componente (în prima coloană - spațiul ocupat):

- - Indicatorul de BOOT (1 Byte)

- - Adresa de început a partiției (3 Byte)
- - Indicatorul de sistem (1 Byte)
- - Adresa de sfârșit a partiției (3 Byte)
- - Sectoare rezervate (4 Byte)
- Lungimea în sectoare a partiției (4 Byte)

Cu ajutorul "indicatorului de B O O T", încărcătorul de partiții determină dacă o partiție este "activă" sau nu. Valoarea înmagazinată acolo nu conține însă doar o informație binară, ci determină în plus și unitatea de disc pe care se află partiția.

Pentru o mai bună înțelegere, să explicăm următoarea stare de lucruri: toate accesările asupra unităților de harddisk și de dischetă se fac întotdeauna prin BIOS, printr-o interfață definită. Aceasta este integrată în calculatorul deja de producătorul calculatorului, sub forma unor componente ROM.

Pentru a adresa un anumit sector sau o grupă de sectoare (pe harddisk sau pe dischetă) sînt necesari întotdeauna 5 octeți. Un octet este folosit pentru determinarea unității, unul determină mărimea blocului (în sectoare) și restul de 3 octeți sînt ocupați cu informațiile despre cilindru, cap și sector.

Unitățile de dischetă se adresează întotdeauna începînd cu "0". Unitatea A este adresată deci cu valoarea "0" și unitatea B cu "1". Unitățile de harddisk - din care pot exista mai multe pe un calculator - se adresează întotdeauna începînd cu "128" ("80 hex"). Un al doilea harddisk ar putea fi adresat deci prin codul de unitate "129" ("81 hex").

Offset	Lung.	Semnificație
0	BYTE	Flag-ul partiției active Bit 7 = 0 : Înregistrare de partiție inactivă Bit 7 = 1 : Înregistrare de partiție activă
		Adresa de început a partiției (primul sector)
1	BYTE	Bit 0 - 7 : Cap (0 - 255)
2	BYTE	Bit 0 - 5 : Sector (1 - 63)
		Bit 6 & 7 : Biții superiori ai nr. de cilindru
3	BYTE	Bit 0 - 7 : Cilindru (0 - 1023)
4	BYTE	ID-ul sistemului de operare
		Adresa de sfârșit a partiției (ultimul sector)
5	BYTE	Bit 0 - 7 : Cap (0 - 255)
6	BYTE	Bit 0 - 5 : Sector (1 - 63)
		Bit 6 & 7 : Biții superiori ai nr. de cilindru
7	BYTE	Bit 0 - 7 : Cilindru (0 - 1023)
8	DWORD	Nr. de sectoare dinaintea partiției
12	DWORD	Nr. de sectoare din partiție

Fig. 3. Structura unei intrări din tabela de partiții

Încărcătorul de partiții distinge, în forma sa uzuală, doar valorile "0" (nu este partiția activă) și "80 hex" (partiție activă de pe harddisk-ul "1"), dar folosește indicatorul de BOOT pentru încărcarea articolului de BOOT al respectivei partiții. Astfel, cu cîteva modificări în codul încărcătorului de partiții, fără modificări hard de configurație, este posibilă încărcarea unui sistem de operare de pe oricare din harddisk-urile instalate.

Adresa de început a partiției descrie, sub forma unei combinații număr cilindru + număr cap + număr sector, adresa disc la care începe partiția. Informațiile conținute în cei 3 octeți nu sînt separate în întregime și corespund formatului în care trebuie să fie predate interfeței BIOS pentru încărcarea articolului de BOOT. Primul octet conține numărul de cap, următoarele două numărul cilindrului și al sectorului. Cei doi biți superiori ai numărului de cilindru - format din 10 cifre binare - (este vorba aici de sistemul "dual") se regăsesc în biții 6 și 7 ai octetului numărului de sector, (numărul de sector fiind format din 6 cifre binare). Se vede de aici că se pot adresa prin intermediul BIOS-ului incorporate harddisk-uri cu o di-

menșiune maximă de 256 de cape-te (2⁸), 1024 cilindri (2¹⁰) și 64 de sectoare pe pistă (2⁶).

Indicatorul de sistem descrie pe un octet tipul partiției, determină deci cărui sistem de operare respectiv cărui tip de gestiune a datelor îi aparține respectiva partiție. Valorile uzuale pentru partițiile DOS sînt: 1 = partiție DOS < 16 Mbyte, cu administrare FAT de 12 biți

4 = partiție DOS între 16 și 32 Mbyte și cu administrarea FAT de 16 biți

5 = partiție DOS extinsă (vezi mai jos) și

6 = partiție DOS > 32 Mbyte cu administrare FAT de 16 biți

Fiecare sistem de operare (CP/M, OS/2, Xenix, etc.) are identificatorul său specific, cu ajutorul căruia poate determina foarte repede, în faza de BOOT, dacă partiția respectivă va fi sau nu recunoscută (și odată cu ea administrarea de fișiere aferentă) și dacă va fi preluată în rîndul unităților logice disponibile.

Dacă aruncăm o privire pe articolul de partiționare cu ajutorul lui "Partiton Table Editor" din "Norton Utilities" și apăsăm de mai multe ori pe tasta <SPAȚIU> în cîmpul BOOT, se pot vizualiza toate tipurile de sistem posibile, în clar, cu numerele aferente în modul HEX.

Adresa de sfîrșit a partiției indică ultimul sector ce face parte din partiția respectivă și corespunde întru totul, ca structură, adresei de început a partiției. Cîmpul de 4 octeți folosit pentru numărul de sectoare rezervate conține o valoare de 32 de biți (dublu cuvînt) care indică cîte sectoare se află între începutul harddisk-ului (adresa 0,0,1) și adresa de start a partiției. Și lungimea partiției este dată de un dublu cuvînt care indică cîte sectoare conține partiția respectivă. În fond această informație și adresa de sfîrșit a partiției ar fi redundante, dar ea folosește programele care au nevoie de această informație, scutindu-le de conversii de la un format de date la altul.

În mod normal, după pornirea sistemului de operare de pe harddisk, ne aflăm pe unitatea logică C și acolo în directorul rădăcină (vizualizat pe ecran prin prompterul "C: "). Dacă sistemul este boot-at de pe unitatea de dischetă, atunci unitatea A: devine unitatea logică curentă (așa-numitul "current drive" în terminologia DOS).

Mai mult: cele mai multe sisteme "cunosc" unități logice suplimentare. Unitatea B este de obicei cea de-a doua unitate de dischete, iar harddisk-ul poate fi partiționat, la rîndul lui, în mai multe unități (logice!). Acestea se selectează la pornirea sistemului de operare și se denumesc în ordine crescătoare, începînd cu simbolul "D:", "E:", "F:" și așa mai departe.

Modul în care aceste unități logice sînt reprezentate pe harddisk și legătura lor cu partiționarea va fi obiectul ultimei părți a articolului. Partiția de boot devine întotdeauna unitatea logică C și ea nu poate cuprinde și alte unități logice. Toate celelalte unități logice nu se află într-o partiție proprie, ci sînt adunate, din prisma articolului de partiționare, în așa-numita partiție extinsă.

Deci, u... sistem "curat" DOS nu va avea nevoie niciodată de mai mult de două înregistrări în tabela de partiții. Indiferent de numărul de unități logice care există în afară de C, ele se vor găsi toate într-o singură partiție, deci într-un domeniu de disc continuu. Din punct de vedere fizic, unitățile logice de disc sînt zone de disc succesive, care "intră" exact în limitele partiției extinse.

Unitățile logice începînd cu D: se pot imagina ca un șir de partiții (extinse) încapsulate, fiecare partiție extinsă fiind formată (cu excepția ultimei unități logice) din domeniul de harddisk aferent unității logice și o altă partiție extinsă. Fiecare partiție extinsă posedă un "articol de partiție extinsă" propriu, structura sa fiind identică cu cea a articolului de partiționare.

Fiecare dintre aceste articole de partiționare conține o înregistrare,

ce caracterizează domeniul de disc aferent unității logice și tipul de gestiune a fișierelor aferent ei.

Atîta timp cît nu este vorba despre ultima unitate logică, există o înregistrare suplimentară pentru o altă partiție extinsă. Dimensiunea acesteia corespunde exact mărimii partiției extinse "actuale", din care se scade domeniul de disc corespunzător unității logice specificate în cealaltă înregistrare a tabeli. Articolul de partiționare extinsă a ultimei unități logice posedă desigur doar o singură înregistrare pentru domeniul său de pe disc și încheie astfel încapsularea, deoarece a fost atins sfîrșitul partiției extinse inițiale.

La pornirea sistemului, DOS trece prin întregul lanț de înregistrări și indicatoare. După care este în măsură să atașeze fiecărei litere -unitate logică un domeniu de harddisk exact definit.

Pentru a înțelege mai bine explicațiile de mai sus vom da un mic exemplu: un harddisk conține 60 de cilindri și este împărțit astfel: unitatea C ocupă 20 de cilindri, unitatea D tot 20, iar unitățile E și F cîte 10 cilindri. În articolul de partiționare se află la început 2 înregistrări: O partiție DOS începînd de la cilindrul 20 pînă la 49 și o partiție extinsă de la cilindrul 40 pînă la 59. Articolul de partiție extinsă conține în primul sector al cilindrului 40 din nou două înregistrări. O partiție DOS de la cilindrul 40 pînă la 49 și o partiție extinsă de la cilindrul 50 pînă la 59. În ultimul articol de partiție extinsă în primul sector al cilindrului 50 nu se găsește decît o singură înregistrare: o partiție DOS de la cilindrul 50 pînă la 59. Mai rămîne doar să amintim că articolul de partiționare este o componentă pasivă a programului FDISK.EXE și el este copiat de acesta pe harddisk la inițializare.

(I.M.)

DOS-extendere

Programarea sub DOS în mod protejat

În urmă cu un deceniu, primul IBM PC apărut pe piață era echipat cu un procesor Intel 8086 pe 16 biți, capabil să adreseze un singur mega de memorie și fără posibilități de dispecerizare a mai multor programe rezidente în memorie. Sistemul de operare i-a fost pe măsură, neprevăzînd direcții de dezvoltare înspre exploatarea unor memorii multimegabyte partajate între mai multe aplicații. Zece ani mai tîrziu, MS-DOS supraviețuiește fără schimbări majore, spre surpriza chiar a creatorilor lui, pe procesoarele 80286/386, ignorînd complet modul lor protejat de adresare a memoriei și exploatăndu-le precum microscopul-spărgător-de-nuci.

Între timp, convulsii precum standardul de memorie expandată LIM EMS 4.0 și specificația de memorie extinsă XMS I-au împins pînă în faza conturării unui adevărat sistem de operare în mod protejat.

Acceptarea lentă a OS/2, inventarea extender-ului de DOS, lansarea lui Windows 3.0 și standardizarea DPML (interfața DOS - mod protejat de adresare), oferă MS-DOS-ului o nouă șansă. Nimeni nu se îndoiește că PC-urile se îndreaptă spre un sistem de operare în mod protejat și totuși nu este deloc clar care va fi acesta și cînd anume se va impune. În ciuda design-ului remarcabil și a implementării specifice arhitecturilor 286/386, OS/2 are puține șanse să înlocuiască DOS-ul pe scară largă. Pare cu mult mai probabil că succesul lui MS-DOS va fi el însuși drapat într-o formă high-tech, un nucleu de instrucțiuni pe 32 biți deservind Windows.

În orice caz tehnologia DOS-extender "vine". Este previzibilă accentuarea competiției pe piața de soft, programele devenind tot mai complexe, pretențiile utilizatorilor crescînd, iar programatorii vor fi

nevoiți să apeleze la extender DOS pentru a putea îmbunătăți performanțele și ridica restricțiile de accesare a memoriei. Este de asemenea previzibil declinul utilizării memoriei expandate; noile implementări ale DPML pe 386 sau procesoare mai evolute oferă facilitățile de memorie virtuală, transformînd programarea asociată EMS într-un efort nejustificat.

Pînă în prezent extenderul DOS

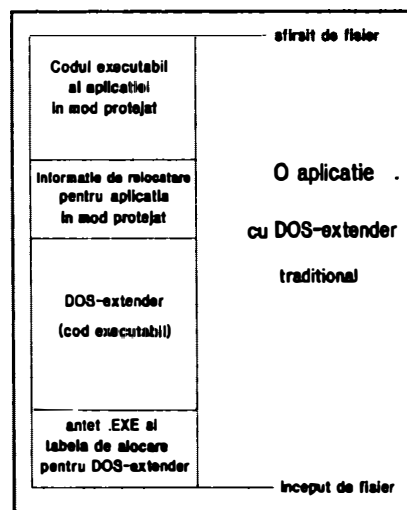


Figura 1. Fișier .EXE cu DOS-extender

au fost medii de programare relativ scumpe și greu accesibile unui programator obișnuit. Situația este în schimbare rapidă, întrucît Windows 3.0 încorporează un extender DOS care în mod protejat este utilizat automat. Chiar și aplicațiile non-Windows pot profita de extenderul DOS oferit de Windows 3.0, după apelul DPML pentru comutare în mod protejat.

Practic, avantajele programării în mod protejat sînt potențial disponibile oricărui programator care utilizează un compilator sub DOS.

Cum se utilizează un extender DOS tradițional ?

La procurarea unei aplicații anterioare lui Windows 3.0 care rulează în mod protejat, precum

Mathematica, Paradox 386 sau Interleaf Publisher, se achiziționează de fapt două programe: extenderul DOS (provenit de la un terț producător, precum Phar Lap sau Rational Systems) și aplicația propriu-zisă.

Programatorul aplicației respective trebuie să procure o licență de distribuție de la producătorul extender-ului DOS pentru fiecare fișier executabil distribuit, conținînd atît aplicația cît și extenderul propriu-zis.

Din punctul de vedere al interpretorului de comenzi MS-DOS, extenderul DOS este singurul prezent în fișierul executabil, fiind singura parte a acestui fișier descrisă de antetul tipului EXE. MS-DOS încarcă extenderul în memoria convențională utilizînd modul real de adresare al procesorului, realizează toate relocările necesare și transferă controlul extender-ului. Extenderul DOS alocă apoi memoria adițională (convențională, extinsă sau amîndouă), redeschide propriul său fișier executabil și încarcă apoi aplicația propriu-zisă în memorie. MS-DOS nu poate încărca direct aplicația în mod protejat, întrucît este absolut inocent în ceea ce privește selectoarele de mod protejat și tehnicile specifice codului obiect 80386.

Extenderul pregătește apoi un mediu de rulare în mod protejat a aplicației: construiește tabela de întreruperi, tabelele de descriptori locali și globali, tabele de paginare și segmentul de stare a programului care va furniza posibilitatea de adresare odată ce CPU este în mod protejat. Acest pas include de obicei crearea unor selectori speciali, care acoperă bufferul de împărsărire a memoriei video, prefixul segmentului de program, zona variabilelor BIOS precum și alte structuri pe care aplicația ar urma să le acceseze direct. În cele

din urmă extender-ul DOS comută CPU în mod protejat și transferă controlul adresei de început a programului.

În timp ce aplicația se derulează, extender-ul DOS îndeplinește rolul său vital de interfață între modul protejat al aplicației și modul real DOS via controlul sistemului de întreruperi. De fapt extender-ul poate fi imaginat ca un rezident care manipulează întreruperi, care - din punctul de vedere al MS-DOS-ului - se găsește în memoria alocată aplicației și urmează să dispară odată cu terminarea execuției aplicației. Extender-ul DOS trebuie să facă față la orice întrerupere care apare în mod protejat, indiferent că este cauzată de un eveniment extern (precum ceasul de timp real), o excepție internă (precum o împărțire la zero) sau o întrerupere software executată de aplicație.

În cele mai multe situații, cînd o întrerupere apare în mod protejat, extender-ul DOS comută pur și simplu CPU în mod real și transmite întreruperea rutinei de tratare inițiale, dar totuși unele întreruperi se bucură de un tratament special. De exemplu, Dos extender-ul de obicei reprogreamază controller-ul de întreruperi 8259, reolocînd noi indicative întreruperilor externe hardware de la 08h pînă la 0Fh, astfel încît aceste întreruperi pot fi diferențiate de întreruperile interne determinate de Protecția generală de erori, Erorile de stivă, etc. Aplicația poate instala propriile sale rutine de tratare în mod protejat a întreruperilor interne și externe, apelînd funcții speciale ale extender-ului DOS. Cele mai importante tratamente speciale apar la întreruperi utilizate pentru apeluri de funcții: MS-DOS INT 21h, ROM BIOS, video driver INT 10h ș.a.m.d.

Acțiunile întreprinse de extender-ul DOS, odată ce a interceptat un asemenea apel de funcții depind de tipul de funcție apelat. Funcțiile vizate de extender-ul DOS pot fi împărțite în patru clase de bază:

-funcții care impun ceva mai mult decît o simplă comutare mod protejat - mod real;

-funcții care adresează buffere ale aplicației și necesită transfer de date și translatore de adrese;

-funcții care trebuiesc complet înlocuite pentru a avea sens în mod protejat;

-apeluri de funcții care sînt specifice extender-ului DOS însuși și furnizează servicii speciale fără echivalent în MS-DOS sau ROM-BIOS.

În prima clasă de funcții toți parametri sînt transmiși în registre; parametrii nu trebuie să includă adrese. Funcțiile de intrare/ieșire pentru consolă, port serial și imprimantă sînt exemple ilustrative. Tratarea acestor apeluri de funcții este simplă. După interceptarea întreruperii software se salvează contextul de mod protejat, se comută CPU în mod real și se reinițiază întreruperea transmițînd controlul MS-DOS-ului sau ROM-BIOS-ului. Cînd apelul de funcție este finalizat, extender-ul DOS restaurează modul protejat (spre exemplu reface cei 16 biți superiori ai regiștrilor de 32 de biți ai procesorului 80386, care nu au fost neapărat menținuți de MS-DOS sau driverele lui) și returnează controlul aplicației.

Următoarea clasă, ceva mai complexă este compusă din funcții ale căror parametri includ adresa unui buffer sau a altor structuri de date. În forma lor originală aceste adrese sînt lipsite de sens pentru MS-DOS sau ROM-BIOS din două motive: adresa este sub formă de

selector:deplasare mai degrabă decît segment:deplasare iar buffer-ul și datele se găsesc mai întotdeauna deasupra limitei de 1 Megabyte.

Extender-ul DOS tratează această situație prezervînd propriile sale buffere sub 640 K pentru o stocare temporară. Spre exemplu pentru un apel de scriere într-un fișier extender-ul DOS interceptează cererea de scriere, copiază datele din buffer în memoria convențională, substituie adresa originală cu cea a buffer-ului de memorie convențională, comută CPU în mod real și în final inițiază apelul funcției de scriere MS-DOS.

Cea de-a treia categorie de funcții este cea în care extender-ul DOS trebuie să înlocuiască serviciile MS-DOS sau ROM-BIOS cu servicii noi adaptate modului protejat, înrudite cu gestiunea memoriei. Spre exemplu funcțiile MS-DOS INT 21h, 48H (alocă bloc de memorie), 49H (dealocă bloc de memorie) și 4AH (redimensionează bloc de memorie) sînt înlocuite de funcții DOS-extender care alocă, eliberează și redimensionează blocuri de memorie extinsă. Funcțiile extender-ului DOS utilizează selectorii de mod protejat, mai degrabă decît adrese de paragrafe sau segmente folosite de MS-DOS, dar și substituirea este practic invizibilă unei aplicații "cuminți", exceptînd tratarea volumelor mari de memorie disponibile în modul protejat.

Ultima clasă de servicii - funcții disponibile numai sub un extender

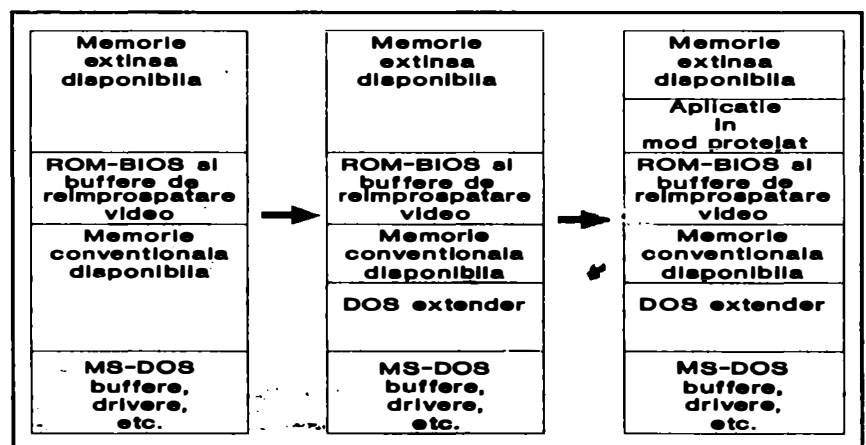


Figura 2. Instalarea unei aplicații în mod protejat

DOS - acoperă un spectru larg. Spre exemplu sînt apeluri de funcții care translatează adrese din modul real în modul protejat, modifică cîmpuri de descriptori, manipulează tabela de descriptori de întrepreri, permit un apel direct al unei subrutine în mod real dintr-o aplicație care rulează în mod protejat, alocă blocuri de memorie convențională, etc. Cele mai multe astfel de funcții sînt superspecializate și nu sînt utilizate în programe aplicative tipice.

Este de imaginat că autorii primelor DOS-extender trebuie să fi fost virtuoși ai programării eclecticice întrucît ei realmente scriau un mic sistem de operare în mod protejat care trebuia să supraviețuiască într-un mediu foarte ostil. Nu numai că acești creatori trebuiau să cunoască interfețele MS-DOS și ROM-BIOS (mai mult sau mai puțin documentate), pe dinăuntru și pe dinafară, dar trebuiau să fie familiarizați cu toate trucurile neortodoxe ale unor aplicații indispensabile precum Lotus 1-2-3 sau Sidekick, trebuiau să stăpînească complet toate diferitele arhitecturi PC, toate caracteristicile 80286 și 80386 și chiar să rezolve problemele ridicate de diferite cartele acceleratoare auxiliare bazate pe 80286 și 80386.

Pe măsură ce pachetele DPML, altele decît Windows, devin disponibile, DOS-extenderele vor fi relativ ușor de implementat, deoarece toate rutinele dependente de hard-

ware precum management-ul întreprerilor, comutarea modului de lucru al procesorului, întreținerea tabelor de pagini și descriptori rămîn în seama interfeței DPML. Este probabil că în următorii cîțiva ani vom găsi funcții de DOS-extender încorporate în bibliotecile runtime ale celor mai populare limbaje de nivel înalt, astfel încît orice program odată compilat va rula în mod protejat, utilizînd memoria extinsă dacă o interfață DPML este prezentă. Deocamdată sarcina unui autor de DOS-extender este mai dură ca oricînd. Nu numai că un DOS-extender comercial, viabil trebuie să ruleze pe un sistem simplu MS-DOS, mai trebuie să coexiste pașnic cu un driver XMS de memorie extinsă, cu o interfață VCPI sau chiar DPML.

Windows 3.0 DOS-extender

În Windows 3.0, un DOS-extender este încorporat în mediul de lucru spre deosebire de formula tradițională "fiecare aplicație cu DOS-extender-ul propriu".

Utilizînd această abordare - de prezentare a aceleiași interfețe pentru apeluri de funcții indiferent de modul real sau protejat în care rulează programul apelant - Microsoft a evitat numeroasele probleme care au apărut sub OS/2 de la introducerea acestuia. Programatorii nu sînt confrunțați cu o nouă API (interfață pentru programarea de aplicații), nu mai este necesară

conceperea de noi documentații, iar bibliotecile runtime nu mai trebuie rescrise. Tranziția nu este dureroasă pentru cei care au respectat regulile în elaborarea aplicațiilor Windows.

DOS-extender-ul lui Windows este adesea confundat cu interfața sa DPML - aceasta - fiind de fapt singura interfață DPML disponibilă. Confuzia are o explicație logică perfectă. Specificațiile interne ale lui Microsoft defineau interfața DPML pe două nivele: un set de funcții de nivel inferior pentru gestiunea întreprerilor, comutarea modurilor și gestiunea memoriei extinse și o interfață de nivel superior (DOS-extender-ul) care furnizează accesul la MS-DOS, ROM-BIOS și driver-ul de mouse, via execuția în mod protejat a INT 21h, INT 10h, INT 33h ș.a.m.d. Funcțiile de nivel superior erau implementate prin funcții de nivel inferior și serviciile modului real DOS și ROM-BIOS.

Dacă specificația DPML ar fi devenit publică într-o formă apropiată de cea originală, rezultatul imediat ar fi fost o versiune în mod protejat a MS-DOS-ului. Aceasta nu s-a întîmplat deoarece Microsoft a urmărit să elimine trăsăturile de DOS-extender ale produselor de pe piață pentru a impune interfața DPML. Cu alte cuvinte Windows conține totuși propriul său DOS-extender dar facilitățile lui nu mai fac parte din specificația DPML astfel încît programatorii nu mai au certitudinea prezenței lor în interfețe DPML altele decît Windows 3.0. DPML așa cum apare astăzi este un fel de versiune din a doua generație a standardului precedent, VCPI (interfața de programare pentru control virtual), iar funcțiile sale nu se intenționează să servească direct ca o platformă pentru o aplicație.

În prezent DOS-extender-ul lui Windows 3.0 este în țara-nimănui a semidocumentației și a semisupportului - precum funcțiile TSR ale MS-DOS-ului și interfețele de redirectare spre rețea. Este o pilulă destul de amară.

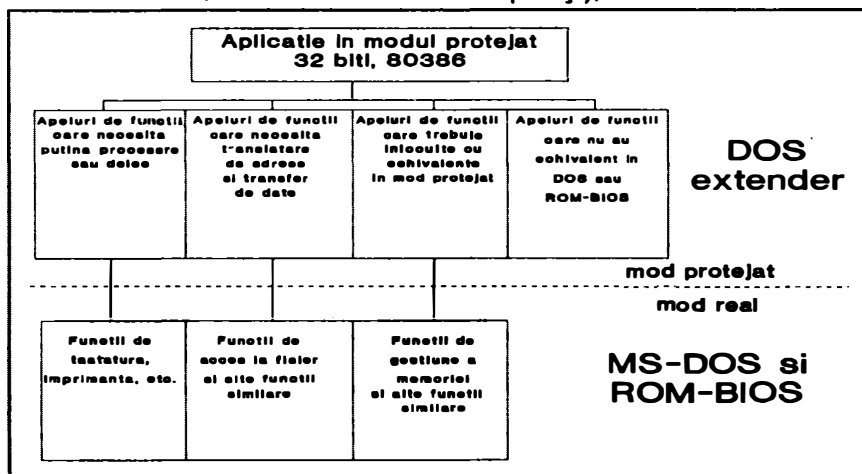


Figura 3.

Probabil că documentația de DOS-extender în forma originală a specificației Microsoft DPML este încă validă deoarece Windows 3.0 a fost foarte aproape de forma sa finală cînd documentația a fost tipărită, (din păcate sub interdicția de publicare).

Singura documentație publică este o recenzie de 5 pagini apărută după un seminar Windows ISV. Acest eseu intitulat "Suportul Windows DPML pentru INT 21h și NET-BIOS" lasă numeroase întrebări interesante fără răspuns, dar furnizează ceva informație critică.

Primul lucru care reiese din acest document este că deși Windows 3.0 în mod 386 conține întregul set de funcții DPML 0.9, în mod standard suportă doar un subset al funcțiilor DPML 0.9 care permit programelor în mod protejat să comunice cu programe rezidente (TSR-uri) și device-drivere din modul real.

Număr - Nume funcție DPML

- 0200h - Returnează vector de întrerupere în mod real
- 0201h - Setează vector de întrerupere în mod real
- 0300h - Simulează întrerupere în mod real
- 0302h - Apelează rutină în mod real cu IRET
- 0304h - Alocă adresă de întoarcere în mod real
- 0305h - Dealocă adresă de întoarcere în mod real.

Cea de-a doua aserțiune din acest important document este că extender-ul DOS din Windows 3.0 suportă aproape întreaga interfață "oficială" a MS-DOS-ului cu programele aplicative. Singurele apeluri nesuportate sînt:

- INT 20h - Returnează vector de întrerupere
- INT 25h - Citire absolută de pe disc
- INT 26h - Scriere absolută pe disc
- INT 27h - Instalare rezident

și următoarele funcții desuete ale INT 21h:

- Funcția INT 21h Nume MS-DOS
- 00h - Termină execuția
- 0Fh - Deschide fișier via FCB
- 10h - Închide fișier via FCB
- 14h - Citire secvențială cu FCB
- 15h - Scriere secvențială cu FCB
- 16h - Crează fișier cu FCB
- 21h - Citire în acces direct cu FCB
- 22h - Scriere în acces direct cu FCB
- 23h - Returnează lungime fișier cu FCB
- 24h - Setează înregistrare relativă cu FCB
- 27h - Citește bloc cu FCB
- 28h - Scrie bloc cu FCB.

Doar cîteva funcții sînt listate ca "parțial suportate" dar restricțiile sînt neesențiale:

INT 21h, funcțiile 25h și 35h (cere și alocă vector întrerupere) returnează și modifică adresele în mod protejat, neafectînd rutinele care tratează întreruperi în mod real;

Apeluri către INT 21h, funcțiile 38h și 65h (Country Information) trebuie să țină seama de faptul că adresa DWORD a rutinei de tratare returnată în structura de date este o adresă în mod real;

INT 21h, funcția 44h (IOCTL), subfuncțiile 2,3,4 și 5 restrîng dimensiunea datelor la mai puțin de 4K dacă buffer-ul este deasupra limitei de 1MB în timp ce facilitățile de coduri de pagină ale IOCTL, subfuncția 0Ch nu mai sînt suportate deloc.

Ce noutăți despre apelurile de ROMBIOS? Specificația originală

Microsoft DPML listează în mod explicit practic fiecare apel ROMBIOS și dacă este sau nu suportat. Documentul public de cinci pagini afirmă numai: "Dacă o întrerupere software API este în întregime bazată pe registre, fără pointeri, registre de segment și parametri de stivă, acea API ar trebui să ruleze sub Windows în mod protejat". Din fericire, aproape toate funcțiile de driver Microsoft Mouse (INT 33h) și ROMBIOS video (INT 10h), comunicații (INT 14h), tastatură (INT 16h), imprimantă (INT 17h), dată și timp (INT 1Ah) se plasează sub umbrela aplicațiilor bazate pe registre și sînt de așteptat să funcționeze corespunzător.

Programarea într-un limbaj de nivel înalt în mod protejat sub Windows este ușoară odată scrisă și depanată o unică rutină care comută în mod protejat printr-un apel la interfața DPML. Cu un pic de noroc biblioteca runtime a compilatorului nu va tolera următoarele practici care crează probleme în mod protejat:

- utilizarea registrelor de segment pentru stocare;
- operațiuni aritmetice cu conținutul registrelor de segment;
- comparare directă a doi pointeri din segmente diferite;
- citire/scriere în afara limitelor memoriei alocate;
- cod care se automodifică.

(A.F.)

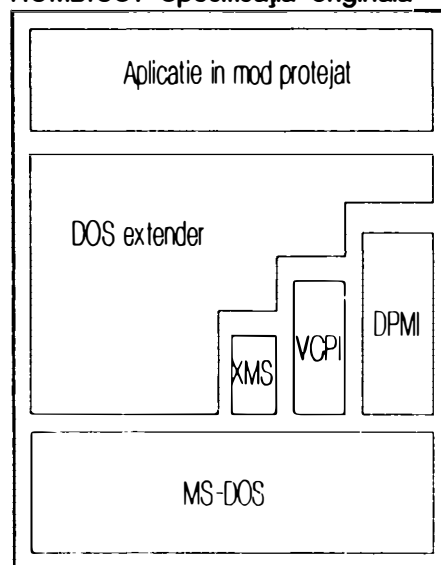


Figura 4

Un extender DOS modern trebuie să rezolve pe lîngă "bucătăria" modului protejat și coexistența pașnică cu un driver XMS (precum HIMEM.SYS), cu o interfață VCPI (precum QEMM-386 sau 386MAX) sau cu o interfață DPML (precum Windows 3.0). Întrucît toate aceste trei medii au interfețe de programare și caracteristici drastic diferențiate viața unui autor de DOS-extender este departe de a fi monotona!

Interfața SCSI - Prezentare și utilizare

Privire retrospectivă

Interfața SCSI (Small Computer System Interface, pronunțat în limba engleză și ca "scuzzy") lucrează la nivelul sistemului și a fost elaborată pentru a permite comunicațiile între sistemele de calcul mici și perifericele lor. Prima versiune a fost realizată la Shugart Associates, în versiunea SASI (Shugart Associates System Interface) pentru legătura între controllerul de disc și discurile hard produse de firmă. În 1979 ANSI (American National Standards Institute) a adoptat și standardizat acest sistem de interfață tip bus, l-a redenumit SCSI și i-a adăugat facilități. SCSI fiind o

specificație de interfață la nivel sistem, definește nu numai nivelele semnalelor pe bus și funcția lor logică, ci specifică și protocolul de comunicație și secvențe de comenzi. Aceasta duce la o mare autonomie a echipamentului conectat pe bus. Controller-ul de pe bus-ul SCSI poate comunica cu discuri floppy, discuri hard, streamere, etc. fără a ține seama de diferențele tehnice interne de realizare a acestor periferice. Se pot realiza modernizări în tehnologia de producție a echipamentelor compatibile SCSI fără schimbări majore în soft-ul și hard-ul de interfață, acestea păstrându-și modul de funcționare standard.

Structura generală

Comunicarea pe bus-ul SCSI are loc între un echipament care inițiază transferul (initiator) și un echipament destinație (target). Comunicarea pe bus-ul SCSI este permisă doar între două echipamente, la un moment dat. Unul dintre echipamente lucrează ca inițiator, selectând și comandând echipamentul destinație care efectuează operația dorită. Când comunicația este stabilită între inițiator și destinație, comenzile spre destinație sînt emise într-un format standard numit "Bloc Descriptor de Comenzi" (CDB - Command Descriptor Block). Standardul SCSI specifică formatul și semnificația CDB pentru fiecare tip de periferic SCSI.

Bus-ul SCSI permite conectarea a maximum 8 echipamente, fiecare avînd o adresă de bus unică, de la 0 la 7. Aceste echipamente pot fi orice combinație de inițiatori și destinații. Bus-ul fizic SCSI conectează serial echipamentele și are un grup de terminatori la capăt pentru a minimiza efectele datorate liniilor lungi.

Linile bus-ului

Există 11 linii de transmitere a semnalelor pentru bus-ul SCSI.

Nouă dintre acestea sînt folosite pentru date (8 biți date plus un bit de paritate impară) și nouă pentru control. Semnalele sînt:

BSY (Busy) o linie realizată SAU cablat, care indică faptul că bus-ul este ocupat.

SEL (Select) este folosit de inițiator pentru selectarea destinației sau de destinație pentru a reselecta un inițiator.

C/D (Command/Data) comandat de destinație pentru a indica dacă se vor transmite informații de date sau control pe bus.

I/O (Input/Output) comandat de destinație pentru a controla direcția de transfer a datelor pe bus.

MSG (Message) comandat de destinație în timpul fazei de mesaj.

REQ (Request) comandat de destinație pentru a indica o cerere de dialog pentru transfer de date.

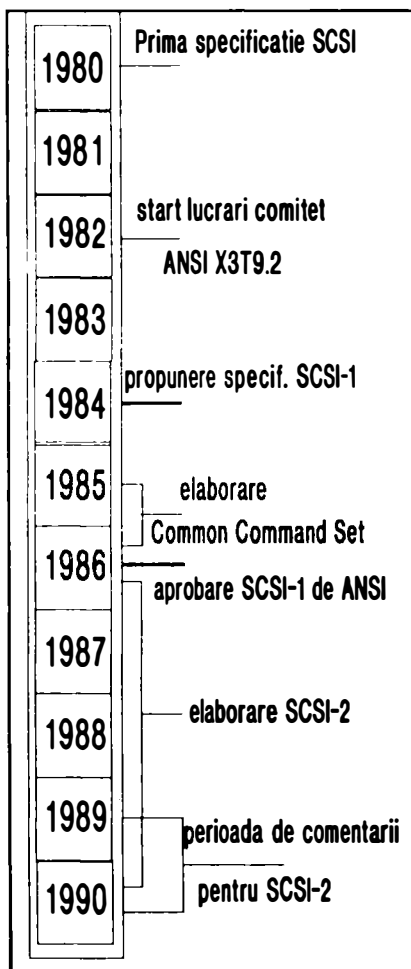
ACK (Acknowledge) comandat de inițiator pentru a indica recunoașterea într-un dialog REQ/ACK (hand shake)

ATN (Attention) comandat de inițiator pentru a indica o condiție de atenție pentru destinație.

RST (Reset) o linie SAU cablată care inițializează (reset) bus-ul SCSI și resetează toate echipamentele de pe bus.

DB 0-7, P (Data bus) opt linii de date și o linie de paritate asociată lor. Transferul tuturor datelor și comenzilor are loc pe liniile de date. Semnificația exactă a octeților de date și direcția transferului sînt determinate de destinație, prin liniile de comandă C/D, I/O și MSG.

(R)Evoluția SCSI



Linii de dialog (Handshake lines)

Nivelul cel mai de jos al protocolului de transfer SCSI este prin "handshake". Pentru fiecare octet transmis în oricare direcție, are loc un dialog pe liniile de cerere (REQ) și recunoaștere (ACK) cerere. Dialogul SCSI este total asincron.

Procesul de dialog începe când destinația stabilește cererea (REQ). Când inițiatorul sesizează REQ, răspunde citind sau scriind un octet de pe/pe bus-ul de date și poziționând linia ACK. Când destinația sesizează ACK răspunde prin aducerea în stare inițială a lui REQ. Acesta duce la aducerea în stare inițială a liniei ACK. Bus-ul este pregătit pentru o nouă secvență de dialog REQ;ACK. Acest proces de dialog este foarte fiabil, deoarece este nevoie de verificarea stării semnalului la fiecare capăt înaintea inițierii unui nou dialog.

Faze

Secvențele de comunicații pe bus-ul SCSI trec prin mai multe faze distincte (BUS FREE, ARBITRATION, SELECTION, COMMAND, DATA, STATUS, MESSAGE, RESELECTION). Bus-ul nu poate fi niciodată în mai mult de o singură fază la un moment dat. După selecție, schimbările de fază sînt controlate de destinație prin liniile de bus C/D, I/O și MSG.

Faza BUS FREE (Bus liber)

În această fază nu există echipamente SCSI care să folosească bus-ul și acesta este disponibil. Toate comunicațiile SCSI încep din faza BUS FREE.

Faza ARBITRATION (de arbitrare)

Această fază are loc doar la sisteme SCSI care au mai mult decît un singur inițiator sau o destinație cu reselectare. De vreme ce este permisă existența pe bus a mai multor inițiatori, pot exista cazuri cînd aceștia încearcă să acceseze bus-ul simultan. Pentru a putea arbitra conflictele se atașează priorității adreselor echipamentelor SCSI de pe bus. Prioritatea maximă o are adresa 7, cea minimă adresa 0.

Fiecărei adrese SCSI i se atribuie un bit de date, pentru a fi folosit în timpul fazei de arbitrare (de ex. DB0 este folosit de adresa 0 și DB7 de adresa 7). În timpul fazei de arbitrare, fiecare echipament SCSI care dorește să ocupe bus-ul poziționează BSY, un semnal SAU cablat, și de asemenea lotul de date corespunzător adresei de pe bus. Dacă după două microsecunde, există biți poziționați cu prioritate mai mare, echipamentul trebuie să elibereze bus-ul și să aștepte o altă fază BUS FREE.

Faza SELECTION (de selectare)

După ce a avut cîștig de cauză în faza ARBITRATION inițiatorul trece în faza de selectare SELECTION. În această fază poziționează semnalul SEL și biții de date corespunzători adresei proprii cît și cel al destinației care se va selecta. Astfel destinația poate determina ce adresă SCSI are inițiatorul.

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	ex.
	r	f	f	s	s	s	s	f	
ex.	r	f	f	0	0	0	0	f	ok
	r	f	f	0	0	0	1	f	check
	r	f	f	0	1	0	0	f	busy

Status byte

Semnificație: r=rezervat, s = predefinit SCSI,
f = specific furnizor

Faza COMMAND (de comenzi)

În această fază destinația cere informație de comenzi de la inițiator. Informația de comenzi este sub forma de Bloc descriptor de comenzi CDB (Command Descriptor Block). Acesta conține informația necesară pentru execuția oricărei operații definite pentru o clasă anume de echipament SCSI, în format standard.

Faza DATA (de date)

În această fază destinația transferă toate datele cerute de Blocul descriptor de comenzi CDB încărcat în faza anterioară. Unele comenzi SCSI nu necesită transfer de date sau vor sări peste această fază dacă se detectează eroare.

Faza STATUS (stări)

În această fază se trimit informații de stare despre ultimul transfer către inițiator. Faza STATUS urmează imediat după DATA, sau dacă această fază nu a fost necesară, după COMMAND. Starea returnată are lungime de 1 byte. Biții din acest octet sînt cu semnificație predefinită de specificațiile SCSI. Există trei biți liberi pentru implementări specifice furnizor.

Organizarea STATUS BYTE

Starea "ok" indică terminarea cu succes a operației precedente. Condiția de "check" ne semnalează că s-a întîmplat ceva neprevăzut în cursul operației SCSI. Inițiatorul trebuie să trimită o comandă de "REQUEST SENSE" (cerere sesizare) pentru a localiza evenimentul neprevăzut. Busy in-

dică faptul că destinația nu poate accepta comandă și inițiatorul trebuie să încerce din nou.

Faza MESSAGEIN (mesaj la inițiator)

Aceasta este ultima fază în cadrul unui transfer SCSI. Se transferă un octet de mesaj de la destinație la inițiator. După transferul octetului echipamentul care era destinație se deconectează de la bus-ul SCSI și revine în faza BUS FREE. În acest moment bus-ul SCSI devine pregătit pentru o altă operație de transfer SCSI.

Faza MESSAGEOUT (mesaj la destinație)

După terminarea fazei SELECTION de către inițiator, destinația are controlul complet al transferurilor de date și schimbării fazelor.

Dacă inițiatorul are un motiv de recuștigare a controlului pe bus (de exemplu detectarea unei erori de paritate) singura lui posibilitate este poziționarea liniei ATN. Dacă destinația detectează poziționarea liniei ATN în timpul unei operații SCSI, atunci va trece în faza de MESSAGE OUT. În acest caz inițiatorul poate trimite un octet de mesaj (dacă ATN este menținut se pot trimite mai mulți octeți) pentru a forța destinația să abandoneze și să reîncepe operația SCSI. Specificația SCSI definește multe configurații de octeți de mesaj, obligatorie fiind în orice implementare mesajul OO indicând terminarea comenzii (Command Complete).

Reconectarea

Pentru a preveni ocuparea pentru perioade lungi în cazul transe-

rurilor lente, anumite periferice de pe bus-ul SCSI au facilitatea numită "reconectare". La operații lente, ca de exemplu căutarea (seek) îndelungată pe disc, destinația va accepta un bloc descriptor de comenzi CDB descriind operația, apoi se va deconecta în mod deliberat de pe bus.

Cînd destinația devine gata (ready) pentru faza următoare a transferului, va acționa ca inițiator pentru a selecta inițiatorul operației și apoi își va relua rolul de destinație. Destinația își va relua acum rolul în faza următoare pentru a completa operația.

Blocul descriptor de comandă CDB (Command Descriptor Block)

Blocurile predefinite de comenzi sînt o parte importantă din ceea ce face ca bus-ul

SCSI să fie performant și popular. Folosind un set obligatoriu de comenzi, se pot scrie drivere soft pentru orice echipament SCSI. Acesta este o parte din ceea ce diferențiază SCSI ca interfață sistem, de interfețele la nivel echipament. SCSI specifică nu numai nivelele de semnal și temporizările pentru interfață ci și comenzi de nivel înalt, la care trebuie să se conformeze orice echipament pentru a fi un echipament SCSI.

Formatul exact și semnificația octeților din CDB depind de tipul de echipament. În mod obișnuit se descriu 6 tipuri de echipamente, împreună cu un set minim de comenzi obligatorii.

Echipament Exemplu

- cu acces direct - disc floppy, harddisc
- acces secvențial - unitate de bandă, streamer, imprimantă - matricială, laser
- procesor - orice echipament inteligent
- WORM - (write once, read multiple) medii optice
- acces direct, citire - CD ROM

Blocurile CDB pot avea trei dimensiuni: 6 octeți, 10 octeți și 12 octeți. Primul octet la orice CDB este un cod operație, și o parte din acesta specifică lungimea CDB. Octetul cod operație este divizat astfel: biții 7-5 conțin codul de grup iar biții 4-0 codul comenzii. Biții cei mai semnificativi sînt deci folosiți pentru codul de grup. În prezent avem 3 coduri de grup definite: grupurile 0,1 și 5.

CDB-urile din grupul 0 au totdeauna lungimea de 6 octeți, din grupul 1 de 10 octeți iar cele din grupul 5 au lungimea de 12 octeți. Grupurile 2 și 4 sînt rezervate pentru dezvoltări ulterioare iar 6 și 7 pot fi utilizate pentru implementări specifice furnizor.

Cîmpul de 5 biți de cod de comandă furnizează pînă la 32 comenzi pentru fiecare grup. Formatele pentru CDB-urile de 6 și 10 octeți sînt date în figură:

grupa 0:

- octet
- 0 cod operație
- 1 LUN(db7-5) / nr. bloc de start (dacă există) MSB
- 2 nr. bloc de start (dacă există)
- 3 nr. bloc de start (dacă există) LSB

Semnal	Pin	Pin	Semnal
masă ecran	1	2	masă
+db0	3	4	-db0
+db1	5	6	-db1
+db2	7	8	-db2
+db3	9	10	-db3
+db4	11	12	-db4
+db5	13	14	-db5
+db6	15	16	-db6
+db7	17	18	-db7
+dbp	19	20	-dbp
diffsens	21	22	masă
masă	23	24	masă
termpwr	25	26	termpwr
masă	27	28	masă
+atn	29	30	-atn
masă	31	32	masă
+bsy	33	34	-bsy
+ack	35	36	-ack
+rst	37	38	-rst
+msg	39	40	-msg
+sel	41	42	-sel
+c/d	43	44	-c/d
+req	45	46	-req
+i/o	47	48	-i/o
masă	49	50	masă

Pin	Semnal
2	-db0
4	-db1
6	-db2
8	-db3
10	-db4
12	-db5
14	-db6
16	-db7
18	-dbp
20	masă
22	masă
24	masă
26	termpwr
28	masă
30	masă
32	-atn
34	masă
36	-bsy
38	-ack
40	-rst
42	-msg
44	-sel
46	-c/d
48	-req
50	-i/o

Asignarea la pini pentru Differential-SCSI

Single-ended SCSI

Toți pinii impari sînt legați la masă

- 4 lungime transfer (dacă există)
 5 octet de control
 grupa 1
 0 cod operație
 1 LUN (db7-5)/rezervat(db4-1)
 / rel (db0)
 2 nr. bloc de start
 (dacă există) MSB
 3 nr. bloc de start (dacă există)
 4 nr. bloc de start (dacă există)
 5 nr. bloc de start
 (dacă există) LSB
 6 rezervat
 7 lungime transfer (dacă există)
 MSB
 8 lungime transfer
 (dacă există) LSB
 9 octet de control

Bus-ul SCSI există în două variante: single-ended-SCSI, la care fiecare semnal e transmis ca tensiune față de o masă comună, și differential SCSI, la care semnalul este reprezentat de diferența de tensiune între două conductoare pereche. Specificația SCSI limitează lungimea unui single-ended-bus la maxim 6 metri, având în vedere mai ales conexiuni în interiorul unei carcase. Un differential-bus în schimb poate fi lung de pînă la 25 metri, fiind mai indicat din acest punct de vedere pentru conectarea unor echipamente aflate în amplasamente distincte. Din motive de siguranță, echipamente conectate prin cele două metode nu ar trebui amestecate pe același bus.

SCSI-1 sau SCSI-2 ?

Dacă SCSI-1 deja s-a afirmat, SCSI-2 de-abia "se naște". Care e diferența între ele? Citînd un membru al comitetului ANSI pentru SCSI: "SCSI promite flexibilitate și performanțe superioare, iar prin SCSI-2 acestea se obțin mai simplu". SCSI-1 este un standard definit (ANSI X3.131-1986), iar SCSI-2 încă se discută. (A nu se confunda cele două standarde cu cele două variante, single-ended și differential, pomenite mai sus - acestea apar ambele atît în SCSI-1 cît și în SCSI-2 !)

A existat o vreme în care se părea că ESDI răspunde tuturor cerințelor utilizatorilor, mai ales

controller-ele noi Super-ESDI cu rate de transfer de pînă la 24 Mbiți/secundă. Dar SCSI este standardul necontestat în lumea stațiilor de lucru Unix și a Macintosh-urilor și se bucură de o atenție din ce în ce mai mare și în lumea PC-urilor. Există motive tehnice clare în favoarea acestei opțiuni; totuși, utilizatorii trebuie să fie circumspecți. Căci în mediul PC-urilor există diverse piedici în calea implementării subsistemelor SCSI și sînt unele probleme de compatibilitate și performanță. Un avantaj al SCSI este viteza sa teoretică. Deja primele specificații SCSI permit rate de transfer de 4 Mocteți/s (32 Mbiți/s), ceea ce este mai mult decît oferă Super-ESDI. Cu toate acestea, deseori SCSI este "simțit" ca fiind mai încet. Acest lucru se datorează implementărilor actuale pentru PC-uri. Spre deosebire de ESDI, IDE sau ST-506, SCSI este un standard de bus, nu de echipament. În felul acesta SCSI oferă o interfață logică între calculator și periferia conectată, care face calculatorul independent de caracteristicile hard ale acesteia. Astfel, teoretic, orice harddisk SCSI poate fi conectat la un host-adapter, fără nici o grijă relativ la BIOS sau firmware-ul din controller. La fel de simplu se pot conecta alte echipamente SCSI: scannere, streamere, imprimante etc. În stațiile de lucru sau Macintosh-uri se poate astfel integra în sistem o paletă extrem de bogată de echipamente periferice variate, fără a ridica măcar capacul de pe carcasa calculatorului. Echipamentele se conectează pur și simplu printr-un cablu SCSI și gata, pot fi folosite la întreaga capacitate de către sistem, fără nici un efort suplimentar. Această flexibilitate se numără printre avantajele majore oferite de SCSI.

Și totuși, pe măsură ce tot mai mulți producători se orientează către SCSI, distanța dintre teorie și realitate crește. Multe din echipamentele SCSI ale trecutului erau, în realitate, incompatibile.

O altă problemă este "overhead"-ul condiționat de SCSI. In-

terfața logică are nevoie de timp de calcul, pentru a transforma comenzile logice în succesiunea corespunzătoare de comenzi fizice. Iar acest overhead se repetă la fiecare comandă SCSI.

La teste efectuate mai demult, PC-uri cu sisteme SCSI s-au dovedit a lucra de-abia la jumătatea vitezei celor de tip ESDI sau ST-506! Dar în prezent, piața oferă echipamente SCSI (host-adaptare și echipamente periferice) pentru toate variantele. Software și drivere sînt disponibile pentru DOS, OS/2 și Unix, harddisk-urile SCSI sînt la fel de rapide ca cele ESDI. Iar faptul că și IBM și DEC s-au orientat spre SCSI pare să asigure semnificația SCSI ca "standard al anilor '90". Marea majoritate a echipamentelor periferice disponibile corespund standardului SCSI-1. Dar produse de performanțe ridicate, comparabile cu perifericele mainframe-urilor, au început deja să apară, bazate pe SCSI-2. Caracteristicile cele mai importante ce definesc SCSI-2:

- rate de transfer de pînă la 10 Mocteți/s prin busul SCSI de 8 biți
- specificații pentru bus de date de 16 și respectiv 32 biți, cu rate de transfer posibile de pînă la 40 Mocteți/s

- corecția de erori la nivelul echipamentelor, pentru a păstra disponibilitatea unității centrale pentru alte activități

- cozi de așteptare pentru comenzi care permit echipamentelor SCSI să-și optimizeze singure performanțele

- nu în ultimul rînd: compatibilitate perfectă cu SCSI-1

Chiar dacă încă mai sînt probleme în atingerea vitezelor maxime de transfer permise de SCSI-2, toate elementele necesare pentru a face posibile aceste viteze de lucru sînt date. Ceea ce înseamnă că era SCSI a început.

ing. Attila Darvas, ing. Iosif Fettich

"Lupa de timp"

Deseori, ar putea fi utilă reducerea temporară a vitezei de execuție a PC-ului, pentru a urmări mai bine o succesiune de imagini grafice de exemplu. Programul "LUPA" permite "frnarea" PC-ului, cu ajutorul combinației de taste <Ctrl> + <Alt> + <Space>. Frna este activă atât timp cât se ține apăsată combinația de taste menționată. Temporizarea se acordează individual; valorile ce determină întârzierea sînt cele din rîndul 34 și respectiv 36 (cu cît produsul AX * CX este mai mare, cu atât întârzierea produsă este mai mare). Programul se creează cu TASM: TASM LUPA; TLINK LUPA; EXE2BIN LUPA.EXE LUPA.COM. Se poate intra direct și în DEBUG; (vezi programul din coloana alăturată); pentru a-i ajuta pe cei care preferă această variantă (sau nu au un macroasamblor disponibil) am dat și valorile corespunzătoare ale adreselor ca și comentarii.

(C.R.)

```

1  o|  title  #lupa.asm  |o
2  o|  extrn  afara: far  |o
3  o|  code  segment word  |o
4  o|  assume cs:code, ds:code  |o
5  o|  org    100h  |o
6  o|  start:  |o
7  o|      mov  ax, 3509h  |o
8  o|      int  21h  ;103  |o
9  o|      push es  ;105  |o
10 o|      push bx  ;106  |o
11 o|      pop  [word ptr x + 1] ;[144] ;107  |o
12 o|      pop  [word ptr x + 3] ;[146] ;10b  |o
13 o|      mov  dx, offset newint9 ;11e 10f  |o
14 o|      mov  ax, 2509h  ;112  |o
15 o|      int  21h  ;115  |o
16 o|      mov  dx, 15h  ;117  |o
17 o|      mov  ah, 31h  ;11a  |o
18 o|      int  21h  ;11c  |o
19 o|      ; noul int 9h  |o
20 o|      newint9: push ax  ;11e  |o
21 o|      push  ds  ;11f  |o
22 o|      push  cx  ;120  |o
23 o|      ; test hotkey  |o
24 o|      in   al, 60h  ;121  |o
25 o|      cmp  al, 39h  ;123  |o
26 o|      jnz  exit  ;142 ;125  |o
27 o|      xor  ax, ax  ;127  |o
28 o|      mov  ds, ax  ;129  |o
29 o|      mov  al, ds:[417h] ;12b  |o
30 o|      and  al, 0fh  ;12e  |o
31 o|      cmp  al, 0ch  ;130  |o
32 o|      jnz  exit  ;142 ;132  |o
33 o|      ; inceput bucla de intirziere  |o
34 o|      mov  ax, 4h  ;134  |o
35 o|      again: dec  ax  ;137  |o
36 o|      mov  cx, 2000h  ;138  |o
37 o|      bucla: loop bucla  ;13b ;13b  |o
38 o|      cmp  ax, 0  ;13d  |o
39 o|      jnz  again  ;137 ;140  |o
40 o|      ; sfirsit bucla  |o
41 o|      ; revenire in vechiul int9  |o
42 o|      exit:  pushf  ;142  |o
43 o|      x:    call  far ptr afara  ;143  |o
44 o|      pop  cx  ;148  |o
45 o|      pop  ds  ;149  |o
46 o|      pop  ax  ;14a  |o
47 o|      iret  ;14b  |o
48 o|      code  ends  |o
49 o|      end  start  |o
    
```

LUPA.ASM

Tabela ASCII confortabilă

Programul de mai jos, pe care îl puteți folosi ca atare ca intrare în SID (DR-DOS; SID < ASCII.DEB) afișează pe ecran o tabelă ASCII; Dacă în linia de comandă mai specificați caractere suplimentare, acestea apar în tabelă în mod blinking și în altă culoare. Pentru intrare în DEBUG, liniile 54-55 se înlocuiesc cu secvența:

```
54 rcx 55 74 56 nsec2.com 57 w 58 q
```

(C.R.)

```

1  o|  a100  |o
2  o|  mov  ah,f  |o
3  o|  int  10  |o
4  o|  mov  si,ff00  |o
5  o|  mov  ax,si  |o
6  o|  call 147  |o
7  o|  mov  al,1a  |o
8  o|  call 160  |o
9  o|  mov  ax,si  |o
10 o|  mov  bl,b  |o
11 o|  mov  di,80  |o
12 o|  inc  di  |o
13 o|  cmp  by[di],0  |o
14 o|  jz   124  |o
15 o|  cmp  [di],al  |o
16 o|  jnz  118  |o
17 o|  mov  bl,8c  |o
18 o|  call 162  |o
19 o|  mov  al,20  |o
20 o|  call 160  |o
21 o|  add  si,14  |o
22 o|  jnb  107  |o
23 o|  mov  ax,e0d  |o
24 o|  int  10  |o
25 o|  mov  ax,e0a  |o
26 o|  int  10  |o
27 o|  inc  by[105]  |o
28 o|  cmp  by[105],14  |o
29 o|  jb   104  |o
30 o|  ret  |o
31 o|  mov  cx,3003  |o
32 o|  aam  |o
33 o|  add  al,ch  |o
34 o|  push ax  |o
35 o|  cmp  ah,0  |o
36 o|  jnz  156  |o
37 o|  mov  ch,20  |o
38 o|  dec  cl  |o
39 o|  jz   15f  |o
40 o|  mov  al,ah  |o
41 o|  call 14a  |o
42 o|  pop  ax  |o
43 o|  mov  bl,7  |o
44 o|  mov  cx,1  |o
45 o|  mov  ah,9  |o
46 o|  int  10  |o
47 o|  mov  ah,3  |o
48 o|  int  10  |o
49 o|  inc  dl  |o
50 o|  mov  ah,2  |o
51 o|  int  10  |o
52 o|  ret  |o
53 o|  |o
54 o|  wasc2.com,100,174  |o
55 o|  q  |o
    
```

ASCII.DEB

Tabelă ASCII rezidentă

Programul ASCII.PAS creează un program rezident (TSR) care intră în acțiune pe "hotkey" (combinația de taste) <Ctrl> + <Alt> + <A>. Cât timp țineți apăsată cele trei taste simultan, pe ecran vă apare o tabelă ASCII extinsă (256 de caractere). Când eliberați tastele, tabela dispăre și puteți continua lucrul din punctul în care l-ați întrerupt. Funcție de cartela grafică existentă în sistemul Dvs..., s-ar putea să trebuiască să modificați adresa memoriei ecran (linia 5)

(C.R./I.F.)

Economie de memorie RAM

Deseori, datorită multiplexelor programe rezidente încărcate, memoria RAM a PC-ului se dovedește neîncăpătoare. Nu se prea știe însă că, în anumite limite, spațiul folosit de acestea poate fi optimizat. Faptul că sistemul construiește șirul "environment" al unui program folosind toți parametri de "environment" ai programului apelant (de regulă, COMMAND.COM), duce la concluzia că procedura de urmat este ca întâi să se instaleze toate TSR-urile, iar abia apoi să se fixeze parametri "environment" prin intermediul comenzilor SET, PATH, DOS. Procedând în felul acesta, se pot economisi până în 100 octeți per program. O altă posibilitate este folosirea DEVICE-driverelor în locul unor programe ce se autoinstalează la

apel. Pentru mouse de ex., de regulă driverul este furnizat în două variante: ca program executabil (MOUSE.COM) sau ca device-driver (MOUSE.SYS). Înglobarea în sistem via CONFIG.SYS duce la o economie de 200 octeți față de instalarea prin apel DOS. Diferența provine din faptul că la instalarea unui program TSR, DOS mai folosește memorie, în afara codului efectiv, pentru șirul "environment", segmentul ce prefixează un program, blocurile de control a memoriei - toate acestea nu sînt necesare cînd instalarea se face prin CONFIG.SYS.

Cîtă memorie folosesc programele pe care le aveți încărcate la un moment dat, puteți urmări, de ex., cu comanda MEM /DEBUG (DOS 4.0)

(C.R./L.W.)

```

1  o|  {$M1024,,0,,0}
2  o|  uses Dos;
3  o|  type ecran = array[0..1999] of word;
4  o|  var shift : byte absolute $40:$17;
5  o|      SA : ecran absolute $b800:0; {eventual $b000:0, adresa memoriei ecran}
6  o|      SB : ecran;
7  o|      i, p, t: word;
8  o|  const H : String = 'Tabela ASCII:.';
9  o|      flag : boolean = false;
10 o|
11 o|  procedure Int9 (Flags, CS, IP, AX, BX, CX, DX, SI, DI, DS, ES, BP: Word);
12 o|  Interrupt;
13 o|  begin
14 o|      if flag then begin
15 o|          if (shift and 12 = 12) and (port[$60] = 158) then begin
16 o|              flag := false;
17 o|              SA := SB;
18 o|          end;
19 o|      end else begin
20 o|          if (shift and 12 = 12) and (port[$60] = 30) then begin
21 o|              flag := true;
22 o|              SB := SA;
23 o|              for i := 0 to 1999 do SA[i] := 0;
24 o|              for i := 1 to ord(h[0]) do
25 o|                  SA[i] := ord(h[i]) + 1792;
26 o|              for i := 0 to 255 do begin
27 o|                  p := (i mod 24*80 + 80) + (i div 24*7 + 2);
28 o|                  SA[p] := i + 1792;
29 o|                  SA[p+2] := i div 100 + 1840;
30 o|                  SA[p+3] := i mod 100 div 10 + 1840;
31 o|                  SA[p+4] := i mod 10 + 1840;
32 o|              end;
33 o|          end;
34 o|      end;
35 o|      Inline ($9c/$9a/0/0/0/0);
36 o|  end;
37 o|
38 o|  begin
39 o|      WriteLn ('Tabela ASCII rezidenta');
40 o|      WriteLn ('Se activeaza prin Ctrl-Alt-A');
41 o|      MemW[Seg(Int9):Ofs(Int9) + $1a4] := MemW [0:36];
42 o|      {pardon...Atentie daca modificati vreo linie de cod!
43 o|      mai treceti odata prin debug, sa vedeti unde pica..}
44 o|      MemW[Seg(Int9):Ofs(Int9) + $1a6] := MemW [0:38];
45 o|      SetIntVec($9, @Int9); Keep (0);
46 o|  end.
```

ASCII.PAS

Igiena zilnică

Pentru întreținerea harddisk-ului este necesar ca periodic să se efectueze anumite operații, ca de ex.:

- căutarea sectoarelor defecte;
- căutarea programelor infestate de viruși;
- salvarea informațiilor vitale (sectorul de boot, FAT-ul și structura de directoare) ;
- reformatarea nedistructivă

În prezent există o mulțime de produse-program care execută unul sau mai multe din sarcinile de mai sus. Execuția lor la comanda operatorului are deza-

vantajul că nu asigură ritmicitate. Pe de altă parte, execuția automată a acestor programe prin înscriserea lor în fișierul AUTOEXEC.BAT nu este recomandată datorită timpului consumat la inițializare.

Programul propus asigură execuția automată a acestor date, la anumite intervale de timp.

Utilitățile sînt împărțite în 3 grupe:

- utilitare ce se execută zilnic, la prima pornire a calculatorului;
- utilitare ce se execută săptămînal (în fiecare miercuri);
- utilitare ce se execută lunar (în prima zi a lunii);

Utilitățile din cele 3 categorii sînt cuprinse în fișierele EVERYDAY.BAT, WEEKLY.BAT și MONTHLY.BAT, ceea ce permite schimbarea lor fără recompilarea programului.

În fișierul AUTOEXEC.BAT se va insera o linie cuprinzînd apelul programului:

```

1  o | program Daily;
2  o | {$M 8192,,0,0}
3  o |
4  o | uses Dos;
5  o |
6  o | var Year, Month, Day, DayNo : word;
7  o |   FlagFile : file of byte;
8  o |   FTime : longint;
9  o |   FileTime : DateTime;
10 o |
11 o | function FileExist (NumeFisier : String) : Boolean;
12 o |   var Fis : file of byte;
13 o |   begin {FileExist}
14 o |     {$I-}
15 o |     assign (Fis, NumeFisier);
16 o |     Reset (Fis);
17 o |     {$I+}
18 o |     if IOResult = 0 then FileExist := True
19 o |     else FileExist := False;
20 o |   end; {FileExist}
21 o |
22 o | procedure Execute (Nume : string);
23 o |   begin {Execute}
24 o |     if FileExist (Nume) then
25 o |       begin {executie fisier comenzi}
26 o |         SwapVectors;
27 o |         exec (GetEnv ('COMSPEC'), '/C' + Nume);
28 o |         if DosError < > 0 then begin
29 o |           writeln ('Eroare DOS: ', DosError, ' la executia fisierului ', Nume);
30 o |         end;
31 o |         SwapVectors;
32 o |       end
33 o |     else begin
34 o |       writeln ('Fisierul de comenzi ', Nume, ' nu exista. ');
35 o |     end;
36 o |   end; {Execute}
37 o |
38 o | begin {Daily}
39 o |   GetDate (Year, Month, Day, DayNo);
40 o |   assign (FlagFile, 'C:\DATE.DAT');
41 o |   if FileExist ('C:\DATE.DAT') then {obține data fisierului}
42 o |     begin
43 o |       reset (FlagFile);
44 o |       GetFTime (FlagFile, FTime);
45 o |       UnPackTime (FTime, FileTime);
46 o |     end
47 o |   else {initializeaza data}
48 o |     begin
49 o |       FileTime.Year := 1980;
50 o |       FileTime.Month := 1;
51 o |       FileTime.Day := 1;
52 o |     end;
53 o |   if (FileTime.Month < > Month) or (FileTime.Year < > Year)
54 o |     {s-a modificat luna}
55 o |     then Execute ('C:\MONTHLY.BAT');
56 o |   if DayNo = 3 {este miercuri}
57 o |     then Execute ('C:\WEEKLY.BAT');
58 o |   if (FileTime.Day < > Day) or (FileTime.Month < > Month) or (FileTime.Year < > Year)
59 o |     {s-a modificat ziua}
60 o |     then Execute ('C:\EVERYDAY.BAT');
61 o |   rewrite (FlagFile); {rescriere fisier, cu data curenta}
62 o | end. {Daily}

```

C:\DAILY

Pentru calculatoarele compatibile XT, care nu au incorporat in ceas intern, această linie trebuie să se găsească după comenzile de introducere de la tastatură a datei și orei (DATE, respectiv TIME).

Alăturat sînt prezentate listingurile programului și exemple de programe ce pot să apară în cele 3 fișiere de comenzi.

Brînduș Vasile,
str. Aurora nr.
25, bl. 10, apt.
32, Brașov

```

@echo off
rem Cautare semna-
turi virusi
c:\antivir\scan c:
rem Test harddisk
c:\norton\dt c:/d

```

EVERYDAY.BAT

```

@echo off
rem Cautare virusi
cu mai multe progra-
me
c:\antivir\vr c:
/q/w/t/s
c:\antivir\antivir
c:\antivir\doctor2
rem ... s.a.m.d.

```

WEEKLY.BAT

```

@echo off
rem Reducerea frag-
mentarii harddisk-
ului
c:\norton\sd c:
rem Stergerea zone-
lor alocare fisierelor
sterse
c:\norton\wipedisk
c:/e

```

MONTHLY.BAT

DAILY.PAS

Lista modemurilor avizate de Ministerul Comunicațiilor

Nr. crt.	Tipul modemului	Firma furnizoare	Țara	Debit binar (bit/s)
1.	GH 2052	ITT	USA	600/1200
2.	MD/6/12	ARE	Italia	600/1200
3.	IBM-3976	IBM	SUA	200
4.	IPA	IPA	România	600/1200
5.	MD-6-12	RACAL MILGO	Anglia	600/1200
6.	TAM-600	TERTA	Ungaria	600/1200
7.	FEA 200	FEA - București	România	200
8.	S 22581-F20	Siemens	Germania	600/1200
9.	TRADAN 1300	ITT	Anglia	2400
10.	GH 1151	ITT	Suedia	200
11.	GH 2054-M16	ITT	Suedia	1200/2400
12.	POTOK-AK	-	URSS	600/1200
13.	DS 9322-31A	NOKIA	Finlanda	1200/2400
14.	GH 2054C-16; C-20; F-26; K-16	ITT	Suedia	1200/2400
15.	GH 2058	ITT	Suedia	4800
16.	THC-598	THOMSON-CSF	Franța	300
17.	2200/24	RACAL MILGO	Anglia	2400
18.	2200/24	RACAL MILGO	Anglia	1200/2400
19.	TELEROM M18	IPA	România	600/1200
20.	MD 200-2	FEA	România	200
21.	TELEROM 3-M1	FEA	România	600/1200
22.	26 LSI	RACAL MILGO	SUA-Austria	2400/1200
23.	MD-101	ROBOTRON	Germania	200 Bd
24.	Cuplor acustic DC-2	Intrep. Er.hip. Periferice	România	200 Bd
25.	DARO-1641 SSW-GDB CUPLOR	RFT	Germania	9000 Bd
26.	MPS 48	RACAL MILGO	SUA-Anglia	4800 Bd
27.	COM-LINK II	RACAL MILGO	SUA	2400/4800
28.	TELEROM 3M0	FEA	România	300
29.	M 122	IEA	România	600/1200
30.	TELEROM 3M0	IEIA Cluj-Napoca	România	200/300
31.	TELEROM 3M1	IEIA Cluj-Napoca	România	600/1200
32.	MODEM 2084 ITT	ITT	Suedia	2400/1200
33.	MODEM 2082 ITT	ITT	Suedia	600/1200
34.	MODEM PENRIL 4800 D	PENRIL-CORP SUA	SUA	2400/4800
35.	MODEM 3M5	IEIA Cluj-Napoca	România	600/19200
36.	TAM 601	TERTA	Ungaria	600/1200
37.	TELEROM 3M2.1	IEIA Cluj-Napoca	România	2400/1200
38.	2400 P.A. FULL-DUPLEX MODEM	RACAL VADIC	SUA	2400/1200/300
39.	TELEROM M11	IPA Microproducție	România	600/1200 cu canal retur 75
40.	SE 1801/2/3	-	Anglia	1200
41.	MODEM SISTEM XB	ICPTTC-ICRET	România	600
42.	TELEROM M3	IEIA Cluj-Napoca	România	4800
43.	HAYES 1200B	-	-	1200
44.	ECM 4896M	-	-	4800/9600

Poșta redacției

Mă cheamă Mugur, sînt elev în clasa a X-a la liceu "Spiru Haret", secția informatică. Lucrez de 4 ani pe o Cobra, mai mult în BASIC și în ultimul an și în CP/M. În curînd voi avea un IBM AT/286. E bine, deși am o mare dragoste pentru informatică și "inghit" zilnic o mulțime de informații, nu cred că revista "IT" mă va ajuta prea mult în învățarea lucrului cu P.C.-ul. De aceea, VĂ ROG, încercați să rezumați în câteva pagini la stîlul Dvs. "academic" și rezervați-ne și nouă, începătorilor, un spațiu cît de mic.

Nu mi vă lua cineva în întrebări fiindcă folosesc un produs Borland fără autorizație? Poate publicați ceva în legătură cu situația de la noi (prezentă și viitoare) privind la drepturile de autor în informatică.

Ing. Sava Stas, Buzău

Aș vrea să învăț în primul rînd Pascal și C și de ce nu și limbajul de asamblare MACRO, dar pentru aceasta îmi trebuie ceva documentație, pe care nu am de unde s-o iau. Dacă cumva puteți să mă ajutați în acest sens, sînt dispus să cumpăr absolut orice în acest domeniu.

Bogdan Dragomir, str. Conductiv nr. 45, bl. O13, apt. 39, sect. 6, 77886 București

Referitor la câteva articole și scrisori de la "Poșta redacției" vă aduc câteva completări.

1. În numerule 3 și 4/91 există două articole referitoare la formatele grafice și program-e de conversie a fișierelor grafice. Pentru cei interesați aș mai adăuga câteva exemple (fișiere mici o ganăși înăsa de a epuiza întregul fișier existent).

SCD - precizarea pentru SCODL, în programe care se ocupă de tratarea dispozitivelor (Corel Draw, Diapo Agro Matrix).

AMP - precizarea de la Aldus Metafile folosit în Page Maker, Pitești 2.

LGO - precizarea de la LOGO este formatul utilizat de programul de creare/analizare (formulare) JetForm.

Fuși Paint care alături de Mac Paint este utilizat de programul de tratament de teste Quartz XPress.

IMA al firmei Zenographics pentru Mirage.

WMP - pentru Docile Album

ASC - pentru ZZ - 2D

DEF - pentru Dyna CADD

CUT - pentru Dr. Halo

GERBER - pentru (re)imprimarea

Doi completări la programul Auto CAD, pe fișiere formatele DXF, SLD, ADI mai există și RND; iar formatul PIC poate fi folosit și în programe mai noi ca: Video Show, Draw Plus, Pitești, Ami Pro.

În domeniul shareware există numeroase programe care permit vizualizarea, conversia fișierelor grafice cît și a imaginilor, imprimarea. Exemple:

VGAZE/GAV 4.0 - convertește în imagini EGA (GIF, 16 culori) toate imaginile GIF sau VGA/MCGA în format.BLD.

VGAZCGA V 2.0 - convertește imaginile din format GIF în imagini CGA 4 sau 16 culori în format GIF sau CGX.

SQZGIF - convertește imaginile din format GIF (rezoluție 2048x2048x256) în format VGA/MCGA și le salvă în standard Blood sau PLT.

MACZGIF - convertește imaginile Mac în imaginile GIF sau format Blood.

GIFPUB - convertește imaginile color GIF în imaginile alb/negru în format PCX pentru a le încorpora în programe de publicistică stil Ventura.

EGAZVGA V 3.0 - convertește imaginile EGA în VGA înaltă rezoluție sau în imaginile MCGA (2048x2048x256 cm) format GIF sau CGX.

GIFTOPCX - convertește fișiere GIF în PCX și inversează MVGAVGA V 1.0b - Afișează pe ecran imaginile GIF, BLD și PLT avînd rezoluția maximă de 2048 x 2048 x 256 adaptabilă la automat la rezoluția monitorului a cunoscutului (BitMap60).

GRAPHIC WORKSHOP V 4.3 - permite vizualizarea, conversia, manipulara, imprimarea imaginilor în format Mac Paint, PCX, GEM, IMG, TIFF, GIF, EPS și MS-Paint. Împăn se poate face pe imprimantele emulatoare compatibile FX-80, pe HP Laser Jet sau Post Script. Memoria EMS sau extensibilă, dacă este prezentă, este recunoscută. Programul recunoaște interfețele grafice de la standardul CGA pînă la Super VGA.

3. Deși nu este un mouse propriu-zis (5/91), Ballpoint (trackball) este ultimul produs al firmei Microsoft în domeniu. Este rezervat calculatoarelor portabile și se poate atînga lateral (altăna sau dreapta, la dorința utilizatorului) cu ajutorul a două cleme metalice. În partea exterioară are patru butoane care pot fi apăsate altă cu degetele cît și cu podul palmei. Este livrat cu un nou program Mouse.com v. 8.0, compatibil cu toate celelalte mouse-uri și programe Microsoft. El permite modificarea vitezei cursorului, iar sub Windows se pot modifica și funcțiile tastelor și azele de deplasare.

Tot pentru amatori de sisteme cu bilă, firma Future Soft a lansat pentru Macintosh, PC și portabile un nou produs Thumbstick (1945) care este o mini-mușcă trackball ce se ține în mîni.

O inovație în domeniul deșirer firmei americane Apoint prin noul produs Mousepen. El a constituit obiectul a numeroase brevete, în special pentru sistemul miniatural de fricțiune în punctul de contact. Sub forma de stilou el conține în vîrf o bilă mică, dură și rezistentă. Pe partea din față, două butoane sînt apăsate la nivelul indexului.

În greutate de 90g, cu un consum electric minim, cu un spațiu de lucru necesar de doar cinci cm el poate lucra pe orice material sau suprafață. Spre deosebire de mouse-urile clasice, Mousepen nu are nevoie de nici o întrebare de conectare. Se poate conecta direct la portul AD B a unui Macintosh, la interfața seriea unui PC sau a unui portabil (RS-232C/1200 bds). Pentru Macintosh nu este necesar nici un program. Pentru PC, Mousepen dispune de un pilot de reglare a vitezei. Este compatibil cu mouse-urile Microsoft și cu modulul Windows. În versiunea Pro, Mousepen prezintă în plus un al treilea buton care permite ajustarea în timpul execuției unei aplicații - tratament de texte, desen de mare precizie - a finetăii rezoluției virfului. Prețul pentru PC este 295\$ iar pentru Macintosh 193\$.

4. Versiunea "beta" DOS 5.0 (5/91) a fost testată de mai mult de 7000 de utilizatori ospunind nici o problemă de "etanșitate". Secretul, altfel bine păstrat, a fost de multa ori decompilat nu numai de articolele și notele din presa americană ci și de "caperși", previziunile și responsabilități micro din numeroase mari întreprinderi. De la versiunile "beta" 333 până la 409c niciodată nu au circulat un așa numit mare de "transmitări nedelicate".

DOS 5.0 ca și DR-DOS a devenit mai "prietenos" permînd alfabeta sintaxei comenzilor. În lucrul cu harddisk-ul programul cache Smartdrive a fost înlocuit cu un program puternic Fast al firmei Future Computer. Suprafața utilizator Donnell-semigrafică, poate genera ecrane VGA avînd aceeași capacitate de gestionare a fișierelor cît și Windows în ceea ce privește deplasarea, copierea sau gtergerea documentelor fiind un concurent puternic al lui Norton Commander. Este permisă, pentru prima dată, formatarea dischetelor de 3 1/2" la 2,88 Mo. (Sînt așteptate, pe piață, noile lecturări de discuri de 3 1/2" - 2,88 Mo HD).

Se poate spune despre DOS 5.0 că este: o versiune 3.31 încorporată în DOS Extender, ca și Qemul al firmei Quarterdeck; permite alfabeta sintaxei comenzilor, ca și DR-DOS al firmei Digital Research, are un Donnell, ca și Program sub Windows 3.0 sau ca și Norton Commander; are o instalare automată, ca toate celelalte programe; are în alfabeta utilizare de manipulare a fișierelor, ca și PC Tools al Norton Point Software - de fapt, aceleași: Microsoft a cumpărat licența de la CPS.

5. Pentru iubitorii familiei de calculatoare Macintosh și pentru amatori de date exacte am corectat și completat datele articolului din ultimul număr referitoare la acesta.

Clasic are frecvența de ceas 7,8336 MHz, iar SE/30 și Portable au frecvența 15,6672MHz. SE/30 are 2560 de memorie cache; //ci și //fx au 256 o de memorie cache pentru date și 256 o de memorie cache pentru instrucțiuni. Ultimele două datorită RAM-ului cu timpul de acces de 80 ns permit pagina-re rapidă ("burst mode"). Nu numai //fx cît și celelalte două calculatoare din familia constructivă modulară //ci și //cc cît și LC au 512 ko ROM care conțin printre altele și programul Quick Draw pe 324 ceea ce le permite alfabeta simultană a 16,8

milioane de culori.

//ci, //ci și //fx pot lucra opțional cu tastatura extinsă de 106 taste. Opțional primăre două pot utiliza un harddisk de 80 Mo iar //fx unul de 160 Mo.

//ci are un conector de RAM-cache 120 de contacte. //fx are 32 ko de memorie cache integrată - SRAM (RAM static) cu timpul de acces de 25 ns ce utilizează Fast Memory Controller.

Tot //fx are un conector cu 120 de contacte care permite accesul direct în bus-ul microprocesorului putîndu-se astfel accesa viteza calculatoarelor prin interfețele de mare viteză.

Clasic, LC și Portable dispun de un circuit sonor ASC (Apple Sound Chip) ce poate genera pentru cîi pe 8 biți la 22 kHz; la celelalte ASC conține un demodulator stereo pe 8 biți la 44,1 kHz și un sintetizator quadrofonic permițînd utilizarea cîștilor sau a difuzorilor stereo.

Clasic cu are intrare audio cî doar ieșire mono.

SE/30 - port ieșire stereo

LC - port intrare mono, port ieșire mono

//ci - port intrare mono, port ieșire stereo

//ci și //fx - doar port ieșire stereo

Portable este identic cu Clasic. El mai poate fi livrat cu taste numerice în loc de "trackball".

Referitor la memoria acestor calculatoare cît și la testorul SuperDrive se poate consulta Poșta redacției - if 4/91.

6. Punctul acesta cu are nimic de a face cu vîriun articol din revistă. M-am hotărît să-l scriu doar după emisiaunea "Uni-versul conșterii" de ieri, miercuri 17.07.91 ora 22.15.

În debutul emisiaunilor, un reporter care a mai lucrat "chidva" în informatică pe prezenta un calculator "specificat" în sisteme vocale. Eu cîd sîd vorbitu-se de documentul respectiv imediat mi ghidnce la un laborator de cercetare și la niște prepari pe măsură. Nici vorbă de așa ceva. Cred că la fel ar fi vorbit și un profan în domeniul calculatoarelor care vizită un PC cu un program de grafică iar fi declarat "specifica-t" în domeniul. Este regretabil ca într-o emisiaune care se vrea de informare științifică, un reporter "sî vîdă" tele-repor-torilor tot ceea ce i se oferă, fără nici un discernămint.

Totuși să vedem despre ce este vorba.

SOUND BLASTER este o interfață sonoră care reunete 11 cî FM de sintetizare (compatibilă ALDIB) și un canal de digitizare vocală; are integrat un amplificator de putere, y obms, y wriis pe canal cu posibilitatea reglării volumului printr-un potențiometru exterior; are de asemenea integrat un amplificator pentru digitizarea analogică pe 8 biți (5 - 13 KHz/mod direct sau transfer DMA) o intrare micro (600 oms); are cîte două cî cu un port joystick care poate fi folosit și pentru o interfață MIDI și cu o ieșire cable numerizată (stereo) pentru boxe, cîcși sau amplificator stereo.

SOUND BLASTER se poate instala în 16 oriicare din slot-uri de extensie ale unui PC, XT, AT, PS/2. Configurația necesară funcționării: minim 512 ko RAM, DOS 2.0 sau o versiune ulterioară, orice interfață video MG, CGA, EGA sau VGA.

Interfața este livrată cu o dichetă demonstrativă cu efecte sonore, un cablu pentru amplificator și: nici programe. (Doi au pînă fi urmările la TV). Prețu 499 DM.

TALKING PARROT - reprezintă un papagal care nu suportă tastatura răspunzînd prin strigăte; acceptă doar microfonul prin care utilizatorul poate comunica cu el, imitîndu-l vocea; imaginile și sunetul pot fi modificate. Pentru o mai bună calitate este de preferat un AT.

FM ORGAN - prin intermediul acestui program se poate chita sau învîta să se chite pe tastatura PC-ului. Este un sintetizator monoofonic care oferă la alegere 20 de instrumente, 80 de acompaniamente pre-registrate, posibilitatea modificării tempo-ului, ritmurilor. Se pot utiliza și instrumente MIDI, care se conectează prin intermediul unei interfețe MIDI (optional) la portul joystick al interfeței.

VOXKIT - permite înregistrarea, stocarea și ascultarea oricărui sunet, cît și compresia și decompresia fișierelor.

SB TALKER - este un sintetizator text-voce care face posibilă introducerea unui text care apoi va fi citit de calculator cu pronunție americană.

DR SBAITSO - este un program de inteligență artificială cu care se poate conversa în engleză asupra diferitelor subiecte.

Opțional există o interfață MIDI și softul aferent care permite utilizarea instrumentelor MIDI (orgă, sintetizator...) în direct cu calculatorul (registrări, resuscitări...).

Tot opțional există boxa "style walkman" care se conectează direct în priză jack de ieșire. Sînt livrate de asemenea două chip-uri C/MS care permit exploatarea jocurilor și programelor e/MS și GAME BLASTER. Ele se conectează în cele două socluri libere de pe interfață. Prețul lor este de 39,95 DM.

Trebuie menționat că există numeroase jocuri care utilizează posibilitățile sonore ale interfeței SOUND BLASTER. Ex: Battle of Britain, Indiana Jones & the Lost Crusade, The secret of monkey island, Prince of Persia, Lords of the rings, Bard's tale III, Ghostmaster II, Time of Lore, Budakan, etc.

7. Sper să le fac o bucurie iubitorilor de Macintosh prin următoarea adresă:

Apple Computer France
12, avenue d'Océanie
Z.A. de Courtaux-Cedex
91956 Les Wis Coeur

Voicu Manole, str. Măcișilor 1, 1900 Timișoara, tel. 961/35437

Mica publicitate

Firmă particulară cu experiență în domeniu, execută programe de evidență financiară, contabilitate, pe calculatoare compatibile IBM - PC. Telefon 90/17.80.95

Ofetim dezvoltări software : simulări procese Industriale, auto-testări cunoștințe, interfețe Interactive. G.Anescu, ICN, C.P.78, 0300 Pitești

Vind calculator personal CIP, 80k, compatibil ZX - Spectrum, în stare de funcționare. Preț: 15.000, tel. 90.16.78.13 (seara).

Caut sursă program sinteză vocală, orice compatibilitate (preferabil IBM-PC). Tel. 916/88660.

Cumpăr originale sau xerox-uri din domeniile: grafică prin raytracing; formate grafice pt. fișiere. Doresc descriere carte sau cuprins. Ciocîrlan Florin, Brazda lui Novac, F9, 1, 19, 1100 Craiova, Tel. 941/54988.

Și Dumneavoastră puteți contribui la îmbunătățirea conținutului revistei !
Ajutați-ne, completând chestionarul alăturat, și revista se va apropia mai
mult de ceea ce așteptați de la o revistă destinată utilizatorilor de PC-uri !
Vă mulțumim !

Chestionar

1. Cel mai mult mi-au plăcut următoarele articole:
1. _____
2. _____
3. _____
2. Cel mai puțin mi-au plăcut următoarele articole:
1. _____
2. _____
3. _____
3. În numerele următoare aș dori să fie incluse articole care să trateze:
1. _____
2. _____
3. _____
4. Utilizez un calculator:
 acasă la birou
5. Tipul calculatorului utilizat:

6. Lucrez mai frecvent cu următoarele pachete software:
1. _____
2. _____
3. _____
7. P.C.-urile mă interesează ca:
 profesionist hobby
 student elev
 utilizator posibil utilizator
8. Citec revista "if":
 regulat ocazional
9. Ce-ar mai fi de spus ? _____

Data: _____ Semnătura: _____ # 7 / 91

**Informații la zi din
lumea calculatoarelor
personale puteți
obține numai citind
regulat revista
"if" !**

Micro ATCI

C.P. 64, O.P. 1

RO - 4300 Tîrgu Mureș

**Aveți ceva de vîndut?
Doriți să
cumpărați ceva ?
Vreți să vă oferiți
serviciile sau aveți
nevoie de ajutor într-o
problemă ?
Folosiți mica publicitate
de specialitate din
"if" !**

Micro ATCI

C.P. 172, O.P. 1

RO - 4300 Tîrgu Mureș

Important !

Doriți să aveți garanția că nu veți pierde nici un număr din "IF" ?
Încheiați un abonament folosindu-vă de talonul alăturat,
sau adresați-vă difuzorilor noștri autorizați:

Arad

Boros Ladislau
tel. 73412, 31778

Bacău

Pătru Liviu
str. 9 Mai, bl. 36, sc. B, ap. 15

Baia Mare

Omni Computer
Cudalbu Octavian
tel. 36433

Brăila

Teodorescu Constantin
tel. 47203

București

Dracea Tudor
tel. 210340

Buzău

S.S.I. Infobuz - Buzău S.A.
ing. Ursu Ioan
tel. 31844

Cluj Napoca

AbMod
● Magazin
Hardware-Software
b-dul 22 Decembrie nr.
135
● Sediul central
tel. 111090
Costin Suciul Libris
tel. 145526

Craiova

Stătescu Paul
tel. 48117

Deva

Toma Tudor
tel. 10144, 10234

Drobeta Turnu-Severin

Bazavan Constantin
tel. 23666 - 143

Onești

Brumă Cornel
tel. 24222-236

Oradea

AbMod
tel. 60278, 44476

Rădăuți

Burciu Florin
tel. 63483

Sighișoara

Beer Mihai
tel. 73638

Timișoara

Tomoroga Mircea
tel. 77422

Căutăm difuzori/distribuitori autorizați
pentru difuzarea revistei "IF" !.

Condiții avantajoase !

Rugăm persoanele interesate să ia legătura cu redacția.



Timișoara
str. Gheorghe Doja nr. 36
tel. 954/31660

MenuPdx 1.0

Biblioteca de meniuri pentru Paradox 3.5

- Cuprinde:

- 6 modele de meniuri Pull Down
- 8 modele de meniuri Pop Up

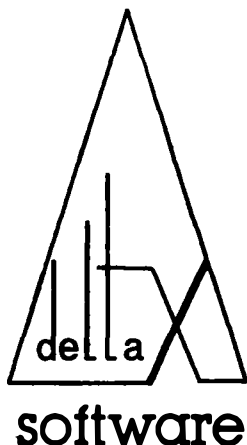
scrise în PAL și apelabile direct din aplicația D-voastră.

- Realizează înlănțuirea automată a meniurilor și lansarea procedurilor din aplicația D-voastră;
tot ce vă rămâne de făcut este crearea unei/unor tabele de meniuri, (eventual help-uri) și să scrieți
procedurile care vă rezolvă problema.

- Demo disponibil.

- Preț informativ: 4895 lei.

**TIMPUL DUMNEAVOASTRĂ ESTE PREȚIOS !
CONTACTAȚI-NE ȘI VEȚI SCĂPA DE MUNCA DE RUTINĂ !**



Societatea de Informatică **DELTA-X S.R.L.**
Oradea, P-ța Independenței Nr.47 A3/23
Telefon: 991-34257, 51804

Vă oferă la prețuri avantajoase următoarele produse program generalizabile pe calculatoare compatibile IBM PC XT/AT:

- 1. Bal-Con** - evidența contabilă generală
- 2. MI-FI** - evidența și calculul amortizării mijloacelor fixe
- 3. Ge-Ma** - gestiunea și evidența contabilă a valorilor materiale (materiale, obiecte de inventar, produse finite).

Produsele enumerate, sînt puternic parametrizate avînd un înalt grad de generalizare și au strîns legături între ele, utilizînd colecții comune de date, transferînd informații între ele. Evidența contabilă Bal-Con, reprezintă "vîrfurile piramidei" care pune la dispoziția celorlalte produse planul de conturi contabile, respectiv primește date (note contabile generate automat) din partea acestora. Sistemul este deschis, nefiind obligatorie utilizarea produselor program enumerate împreună, respectiv este facilitată conectarea altor aplicații la evidența contabilă (salarii, desfacere, etc.). Utilizarea și operarea este facilă prin meniurile, mesajele explicative concrete apelabile în toate fazele de prelucrare, și autodocumentare.

Prin programele de instalare livrate, utilizatorul poate configura produsele la caracteristicile echipamentului utilizat.

Versiunile finalizate ale celor trei produse menționate reflectă legislația economică și reglementările metodologice în vigoare și vor fi aliniate și în viitor la eventualele schimbări legislative sau metodologice care vor apare.

- 4. dAccess** - accesare multiuser a bazelor de date compatibile dBASE.

Permite culegerea, prelucrarea și vizualizarea datelor în regim multiuser, de la 1 la 8 terminale, conectate pe căile seriale COM1 - COMB și de la tastatura/monitorul calculatorului.

Oferă în regim multiuser, prin taskuri independente, majoritatea prelucrărilor permise de sistemele tip dBASE, definirea acestor taskuri făcîndu-se într-un mod simplu, familiar utilizatorilor.

Avînd în vedere, că majoritatea utilizatorilor de tehnică de calcul din economia noastră, dispun de terminale simple în bună stare de funcționare (VDT-52S, etc.), prin utilizarea acestui produs achiziționarea unor calculatoare IBM-PC compatibile în configurația obișnuită (2 căi seriale) nu exclude utilizarea eficientă a acestor terminale devenite disponibile.