

**CALCULATOARE
PERSONALE**

1 / 92

(12)

125 Lei

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

ISSN 1220-1529



Periferice de intrare

Test: Balanțe contabile

Visual Basic

DR-DOS 6.0 vs MS-DOS5.0

Rețele locale

Memento:

Funcțiile Paradox 3.5

Ultima ora!

*S-a infiintat POLICOM SA., societate
comerciala multinationala cu capital privat,
avind principal actionar Ab Mod SNC*

*Noua firma va ofera un sortiment larg de
produse :*

- Calculatoare personale compatibile IBM*
- Imprimante matriciale*
- Dispozitive grafice*
- Floppy - dischete*
- Produse software originale*
- Copiatoare*
- Truse si accesorii pentru intretinere*

Nu uitati !

*POLICOM SA un partener potential pentru
a facerile dumneavoastra*

CLUJ : B-dul 22 Decembrie nr. 135 tel. 95-156350

if

revistă de Informatică
editată de firma Micro ATCI

Director: ing. Dumitru Dunca

Redacția:

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

La acest număr au colaborat:

ing. Vasile Brînduș
mat. Nicolae Galfi
ing. Răzvan Grigorescu,
mat. Vasile Man
ing. Bogdan Munteanu
fiz. Vera Radovici
mat. Eugen Rotaru
ing. Răzvan Sandu
stud. Socaciu Ioan-Tiberiu
ing. Marius Sturzoiu

Tiparul: tipografia Tîrgu Mureș

Revista apare lunar.

Preț: 125 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 manuscris pe adresa redacției, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază 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 revistei 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./fax 954/31660 (direct),
33612, 24158, 20057, 33511,
int. 134 sau 189
41417 (după ora 19)
telex 65354 (Intertours)

Ofuri

"If ist nicht tot, Umwertung aller Werte" ar zice Nietzsche dac-ar mai trăi. Dar el nu mai e și nice If nu se simte prea bine. Altfel ierarhiile se schimbă, pînă mai deunăzi un număr costa cam cît 2,5 l de benzină, iar azi ... Aceasta nu înseamnă însă că revista noastră s-ar fi ieftinit, deși ne-ar fi plăcut să fie așa

Ceea ce ne doare mai tare este faptul că nu reușim să impunem periodicitatea de apariție dorită. Anul trecut ne-am realizat doar 66% din obiective. Cine-i di vină? Sînt mulți vinovați, dar o parte din vină ne aparține și nouă. Avem sertarele pline, dar n-am fost suficient de atenți, în clasele primare, atunci cînd se explica principiul titirezului, așa că la cunoscutul (la noi) dans popular *Învîrtita* mai greșim pașii.

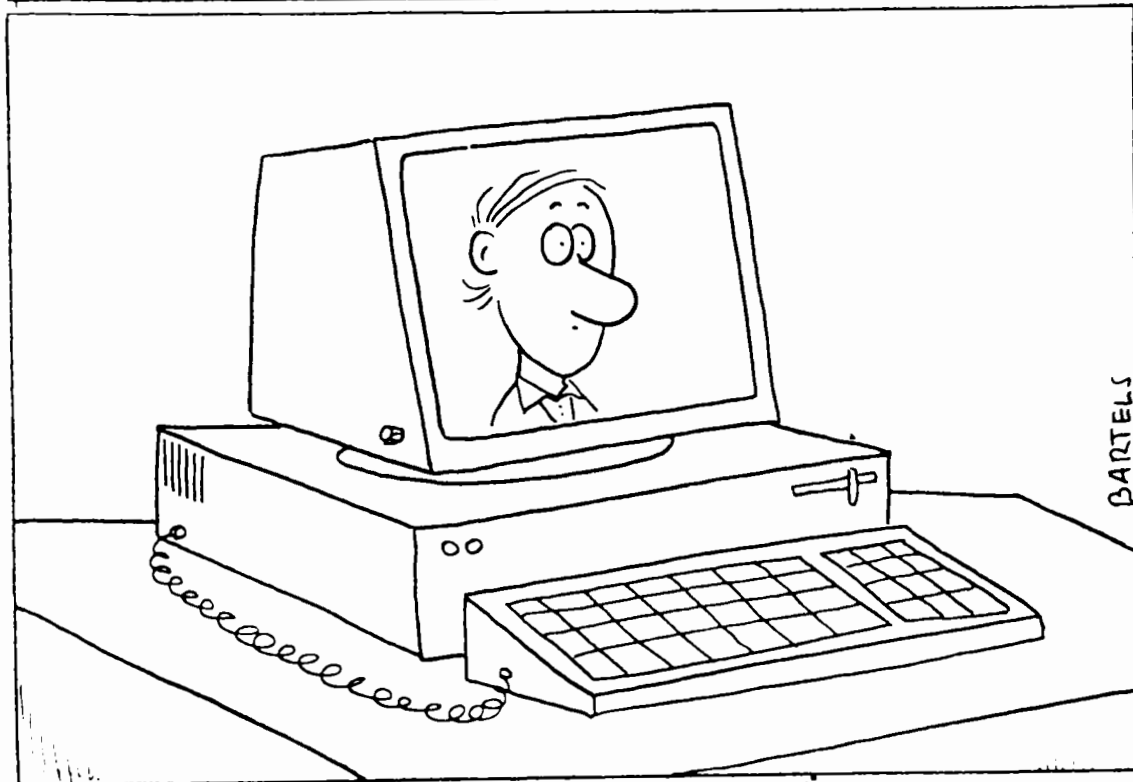
Începînd din acest număr vă prezentăm știri fierbinți (în momentul redactării), provenite direct de la sursă (aceasta este și explicația prezenței adreselor de la sfîrșitul lor). De aici derivă o rugăminte pentru cititori și o sugestie pentru întreprinzători. Mai întîi rugăminte: atunci cînd vă adresați acestor firme pentru informații suplimentare, sau cu comenzi, nu uitați să specificați: "...am citit în If că ...". Și sugestia: întreprinderile occidentale au, de regulă, o listă a publicațiilor de specialitate care se adresează acelei categorii de cititori căreia îi sînt destinate și produsele proprii. Publicațiilor aflate pe acea listă li se trimite prezentări ale tuturor produselor noi care se vor lansa pe piață (descrieri ale produselor, fotografii, dischete cu programe demonstrative, echipamente de test pentru o perioadă de 4-6 săptămîni, invitații la sediul întreprinderii pentru demonstrații, etc.). Dacă vă hotărîți să vă faceți o astfel de listă, vă rugăm să ne treceți și pe noi pe ea. Nu ne angajăm că vom publica toate prezentările trimise, dar materialele publicate vă vor face publicitate aproape gratis!

Lansăm cu această ocazie încă două concursuri: "Cea mai bună caricatură pe teme de informatică" și "Cea mai bună povestire care să conțină în ea și cuvîntul «calculator»". Caricatura ar fi bine să fie pe o coală de hîrtie format A4 și să nu fie lucrată în creion, iar povestirea să nu fie mai lungă de 2 pagini A4 dactilografiate la 2 rînduri. Cu ocazia fiecărei apariții a revistei va fi premiată cîte o singură lucrare de fiecare gen. (cele publicate în numărul respectiv), cu cîte un premiu în valoare de 2.000 lei. Lucrările nepublicate nu se restituie, dar se comunică autorilor acest fapt pentru a putea fi expediate altor redacții. Sperăm ca în numărul 2/92 să putem publica primele lucrări și să vă putem expedia și primele premii.

Sperăm că veți găsi suficiente subiecte interesante și în acest număr. Am dori să vă atragem în mod deosebit atenția asupra articolului "Balanțe contabile". Acest articol încearcă să vă prezinte, pentru prima dată la noi, dacă nu ne înșelăm, un test comparativ al unor produse soft autohtone. (Testul a fost făcut la începutul lunii noiembrie; este posibil ca pînă la această dată să existe versiuni mai noi ale acestor programe.) Sîntem la prima încercare de acest gen. Probabil că testul ar fi putut fi instrumentat mai bine. Sugestiile Dvs. ne vor fi de folos la viitoarele încercări, (sau poate ne vor determina să ne oprim). Repetăm, cu riscul de a fi catalogați drept pisălogi, că dorim să facem o revistă pentru Dvs., cu ajutorul Dvs. Hello... "Is there anybody out there?"

Ing. Romulus Maier

"Pentru intrarea în sistem,
apasă <Enter>"!...



BARTELS

Caroon Joachim Bartels

Cuprins

Expoziții

- Systems '91 pag. 4
- Comdex '91 pag. 4
- Info-Market '91 pag. 4

Știri

- FotoMan pag. 5
- CATD - Computer Added Textil Design pag. 6
- WordPerfect 1.0 pentru Windows pag. 6
- Listare color în format de ziar pag. 7
- Virus - Utilities pag. 7

Sisteme de operare

- DR-DOS 6.0 vs MS-DOS 5.0 pag. 8
- Novell cumpără Digital Research pag. 11

Profil

- Le roi est mort! Vive le roi! pag.10

Portret Philippe Kahn

Test

- Balanțe contabile pag.12
- BALPRO pag. 13
- BAL-CON pag. 13
- CONTAS pag. 14
- BALL pag. 14

Periferece de intrare

- Periferece de intrare pag. 16

Tastatura, lighpen, mouse, joystick, tableta grafică, trackball, touchmouse, scanner, ...

- Touch me, user! pag. 22

Cîte ceva despre cum lucrează un touchscreen

Vă prezentăm

- Visual Basic - programare intuitivă pag.24

Inteligența artificială

- Ce este bioinformatica? pag.25

Rețele locale

- Programele de educație Novell pag. 31
- Rețele locale sau mainframe ?
- o analiză a costurilor "la bani mărunți" pag. 32

Baze de date

- Servere de baze de date pag. 37
- Emerald Bay 2.27 (RSPI) pag. 37
- Ingres for OS/2, DOS (Ingres) pag. 37
- Microsoft SQL Server 1.1 (Microsoft) pag. 37
- NetWare SQL 2.11 (Novell) pag. 37
- Oracle Server 6.0 (Oracle Corp.) pag. 38
- SQL Base Server 4.1 (Gupta Tech.) pag. 38

Fundamente

- MS-DOS:
Sistemul de gestiune al fișierelor pag.39

Cursuri

- Turbo Prolog (partea a treia) pag.43
- OOP: C++ - (partea a doua) pag.46

Practică

- 1024 pag.52
- Parcarea harddisk-urilor pag.52
- Ce zi e astăzi? pag.52
- Program cu repetiție pag.53
- Subdirectoare pag.53

Rubrici

- Caseta redacției pag. 1
- Editorial pag. 1
- Colțul programatorului pag.29
- Poșta redacției pag.54
- Mica publicitate pag.54

DARLAN ROM SUISSE Srl

Distributor :

**ROM CONTROL DATA
KT TEHNOLOGY PTE
FOURTH SHIFT CORPORATION
RING COMPUTER**

Va pune la dispozitie programul de amnistiie :

FOXPRO 2.0

la urmatoarele preturi :

FOXPRO 2 0 MULTI USER(nelimitat)....	275.000	Lei
FOXPRO 2.0 SINGLE USER	215.000	Lei
DISTRIBUTION KIT (RUNTIME)	175.000	Lei
FOXGRAPH.....	185.000	Lei

Servicii hardware :

- calculatoare compatibile IBM PC*
- retele NOVELL*
- garantie, postgarantie*

Servicii software si consulting :

- analiza sitemului informational*
- implementarea sistemului integrat de gestiune
umarire si conducere a intreprinderii*

FOURTH SHIFT

- cursuri de initiere si perfectionare*

Relatii suplimentare la sediul firmei noastre :

CLUJ, OBSERVATORULUI 146/15

TEL : 95-123611 FAX : 95-124567

FotoMan - aparat de fotografiat alb-negru pentru PC-uri

FotoMan este o cameră digitală concepută în mod special pentru PC-uri. În comparație cu o cameră video analogică FotoMan nu necesită nici cartelă Frame-Grabber suplimentară și nici cartelă de digitalizare și deci nu are nevoie nici de o mufă suplimentară în PC deoarece FotoMan lucrează în mod digital. Imaginile în nuanțe de gri pot fi transferate direct în PC, prin interfața serială. Un chip de memorie RAM intern memorează pînă la 32 de imagini și le transmite apoi software-ului, imaginile rezultate fiind de foarte bună calitate. În acest caz nu mai apar pierderi de calitate ca în cazul transformării imaginilor analogice în imagini digitale.

Bateriile încorporate permit o funcționare independentă de rețea de 24 de ore. Ele se reîncarcă automat atunci cînd aparatul este așezat în unitatea de bază (prin intermediul căreia sînt trimise și imaginile în calculator). Încărcarea nu durează mai mult de 6 secunde și poate fi realizată și de la bateria mașinii.

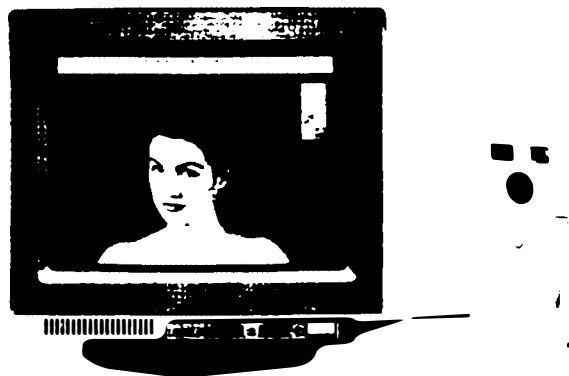
Semnale sonore specifice semnalizează din timp necesitatea reîncărcării bateriei, umplerea memoriei (cu cele 32 de poziții) sau faptul că aparatul este gata de lucru pentru următoarea fotografie.

După ce cele 32 de înregistrări au fost făcute, FotoMan poate fi cuplat printr-un cablu de legătură normal la interfața serială a PC-ului. Cu ajutorul soft-ului FotoTouch, dezvoltat de Logitech în mod special, imaginile pot fi încărcate în PC și pot fi memorate, prelucrate sau șterse. FotoTouch golește în mod automat memoria de imagini a camerei FotoMan, și ea devine din nou aptă pentru 32 de înregistrări.

Camera digitală poate fi cuplată prin interfața serială cu orice PC portabil, indiferent că e un Laptop sau Notebook. La cîteva secunde după înregistrarea imaginii ea poate fi vizualizată pe Notebook cu ajutorul programului FotoTouch, poate fi prelucrată sau poate fi expediată via Modem.

Un bliț înglobat garantează imagini iluminate optim și în încăperi întunecoase. Timpii de expunere sînt de 1/25 secunde cu bliț și de la 1/1000 pînă la 1/30 secunde în mediu luminos. Atunci cînd este necesar blițul se declanșează automat. Avînd o greutate de doar 280 g și dimensiuni reduse de 16,8 cm x 8,1 cm x 3 cm camera poate fi pusă comod într-un buzunar. Datorită unor adîncituri ergonomice pentru degete atît stîngacii cît și dreptacii pot fotografia cu FotoMan fără balansări, în momentul declanșării. Lentila standard corespunde unei lungimi de focalizare de 55 mm într-un format de 35 mm.

La o cameră normală aceasta ar corespunde la o diafragmă f4,5. Dacă FotoMan ar lucra cu filtru, atunci sensibilitatea acestuia ar trebui să fie de cca. ASA 200. Lentila este prevăzută cu un filet, care permite montarea unei alte lentile sau a unui filtru. Distanța la fotografiere cu FotoMan este de la 1 metru (cu bliț 1,2 metri) la infinit.



Rezoluția unei imagini este de 376x282 pixeli, ceea ce corespunde la o rezoluție de 75 dpi, calitatea pozelor fiind suficientă pentru ziare, broșuri, etc. O poză memorată necesită cca. 100 KB.

FotoTouch este o dezvoltare a software-ului Ansel conceput pentru scanner-ele Scanan 256 și permite prelucrarea imaginilor în 256 nuanțe de gri. FotoTouch recunoaște automat dacă FotoMan se găsește sau nu în unitatea de bază și selectează automat viteza maximă de transfer pentru preluarea fotografiilor. Fotografiile aflate în memoria lui FotoMan sînt afișate în așa-numitul mod previzualizare, în blocuri de cîte 5, de mărimea unor timbre. Cu ajutorul mouse-ului se selectează fotografiile dorite, acestea fiind apoi memorate în PC. Se poate specifica numele fișierului în care să se facă memorarea (max. 8 caractere), dacă se dorește se poate înregistra și numele fotografului. Imaginile pot fi memorate în format TIFF, PCX, BMP sau EPS.

Software-ul permite și luarea unor instantanee la "locul de muncă", putîndu-se specifica intervalul de timp după care să se facă declanșarea automată. FotoTouch permite și o prelucrare a imaginilor obținute, cum ar fi mărirea/micșorarea luminozității, mărirea/micșorarea imaginii, decupări, rotiri sau scalări ale acesteia, etc.

FotoMan și FotoTouch lucrează cu orice PC-uri de tip IBM 286, 386 sau 486, sisteme PS/2 și compatibile. Se presupune o memorie minimă de lucru de 1 MB RAM, dar se recomandă 2 MB. Software-ul lucrează sub Microsoft Windows 3.0. FotoMan poate fi utilizat cu orice cartelă grafică. EGA, VGA, Super VGA, Hercules și compatibile. Deoarece camera digitală se cuplează serial, la PC trebuie să existe o interfață RS-232 liberă.

FotoMan va putea fi obținut începînd cu sfîrșitul lunii ianuarie 1992 la toate magazinele de specialitate, prețul fiind de cca. 1649 DM (inclusiv impozitul pe valoarea adăugată).

Logitech SA

CH - 1122 Romanel/Morges, Switzerland

Tel: 41(0)218699656, Fax: 41(0)218699717

Computer Added Textil Design (CATD)

Încă de când primii oameni au început să picteze pereții grotelor, sculele lor și îmbrăcămintea lor, pornirea de a crea cu mijloace simple lucruri deosebite este de nestăvilit în noi toți.

Până acum nu a fost posibilă executarea unui design creativ în serie mică sau de unicat pe diferitele textile cu ajutorul ustensilelor existente; Dar acum există transferul creativ!

Finkenzeller a reușit să elaboreze o nouă tehnologie denumită Creative Transfer și să ajungă să producă design creativ și în serie mică, ieftin, repede și flexibil.

PC-ul oferă cale liberă creativității. Pot fi create și modificate ilustrații și desene de toate tipurile, fotografiile și inscripții cât și grafice în toate variațiunile.

Cu ajutorul imprimantei laser color cuplate la PC aceste proiecte și desene pot fi transpuse în doar câteva secunde pe hîrtie. Pagina astfel obținută, tratată cu un fluid special, se poziționează pe materialul textil pregătit în presa necesară. Toate graficele și toate imaginile vor fi impregnate în materialul textil la o temperatură de 210 grade și o presiune de peste 2 tone.

Dezvoltarea acestui sistem este practic nelimitată, fiind posibilă captarea imaginilor video, a diaporitivelor, a fotografiilor precum și prelucrarea grafică a tuturor informațiilor.

În timpul testului la care am asistat, în mai puțin de 5 minute o imagine color prelucrată pe PC a fost



transpusă pe hîrtie și de acolo pe un tricou de bumbac. Culoarea se impregnează în material și nu se aplică pe acesta, ceea ce înseamnă că la întindere imaginea nu crapă. Un material textil astfel impregnat poate fi spălat la o temperatură de 40-50 grade Celsius, fără ca imaginea să aibă de suferit. Este posibilă și impregnarea de imagini pe sticlă, materiale plastice, metale. O instalație completă costă în jur de 80.000 DM.

Carl Finkenzeller GmbH
Ingoistädter Str. 61n, 8000 München 45
Tel. 089/31818 - 103, Fax: 31818 - 138

WordPerfect 1.0 pentru Windows

La Systems '91 a fost prezentată prima versiune a procesorului de texte WordPerfect pentru Windows. Cu aceasta renumitul procesor de texte este disponibil nu doar pe DOS-PC-uri, Mac-uri, pe calculatoarele Unix și Next, pe VAX, calculatoarele mari IBM și pe sistemele Data General ci și sub Windows 3.0.

Paleta de funcții a lui WordPerfect 1.0 se bazează pe funcțiile versiunii DOS WordPerfect 5.1, fiind extinsă cu facilități suplimentare care utilizează avantajele lui Windows 3.0. Programul lucrează în modul WYSIWYG, ceea ce înseamnă că toate formatările de text, fonturile, graficele, tabelele, notele de subsol și formulele unui document vor avea o reprezentare identică pe ecran și la imprimantă.

Prin intermediul noilor funcții tabulatorii, marginile paginii, coloanele, tabelele și distanțele dintre rînduri pot fi definite direct pe ecran cu ajutorul mouse-ului. Toate funcțiile pot fi apelate, în funcție de preferințe, cu ajutorul mouse-ului, prin intermediul meniurilor pull-down, sau de la tastatură.

Cu totul nouă este funcția WYSBYGI (What You See Before You Get It). În modul previzualizare documentele dorite din director pot fi vizualizate într-o fereastră pe monitor și pot fi apoi încărcate. Această reprezen-

tare a formatării exacte cu toate atributele textului are loc la un simplu clic pe mouse.

Funcțiile utilizate mai frecvent și macrourele dintr-un meniu pot fi aranjate liber într-o listă de simboluri, denumită "Button Bar", reprezentate grafic și activate apoi printr-o tastare pe mouse.

Programul permite cuplarea și prelucrarea graficelelor, dispune de un editor de formule, de generatoare de tabele și de formulare, de stiluri ca și de gestiunea notelor de subsol și de sfîrșit. WordPerfect 1.0 pentru Windows este compatibil cu versiunea DOS WordPerfect 5.1, ceea ce înseamnă că între cele două versiuni pot fi schimbate texte fără probleme.

Prețul produsului este de cca. 1390 DM. Costul trecerii de la WordPerfect 5.1 la WordPerfect 1.0 pentru Windows este de cca. 350 DM.

Ca o curiozitate să amintim faptul că pînă în luna august a acestui an, numai în SUA au fost livrate mai mult de 2 milioane de versiuni. (În tot anul 1990 au fost livrate 2,35 milioane versiuni WordPerfect 5.1). Pentru comparație, în 1990 au fost vîndute 1,6 milioane de aplicații Windows.

WordPerfect Software GmbH
Frankfurter Strasse 21-25, 6236 Eschborn
Tel. (06196) 904 - 01, Fax (06196) 46003

Listare color în format de ziar

Producătorul de imprimante Tektronix a lansat pe piață prima imprimantă color de 300 dpi, numită "PhaserJet PXi" și care poate tipări în format de ziar aproape orice tip de hârtie.

Tipărirea pe o dimensiune a hârtiei de 297 x 446 mm dau utilizatorului posibilitatea de a face reprezentări în dimensiuni realiste. El poate alege dintr-un set de peste 16 milioane de nuanțe de culori.

Imprimanta PhaserJet PXi utilizează limbajul Postscript-Level 2 al firmei Adobe Systems care a îmbunătățit performanțele standardului Postscript cu câteva caracteristici și care oferă de exemplu o adaptare a culorilor independentă de aparatură.

Prelucrarea extrem de rapidă a fișierelor Postscript se datorează procesorului RISC de 24 MHz înglobat în PhaserJet PXi. Paginile color sînt listate în doar 2 minute, în timp ce cele alb-negru se pot lista în 40-60 secunde, în funcție de densitatea textului.

Configurația de bază a imprimantei PhaserJet PXi constă dintr-o memorie de 10 MByte, care poate fi extinsă, cu de două ori 4 MByte, pînă la 18 MByte. Prima extensie de 4 MByte oferă suficientă memorie pentru listarea unei imagini de dimensiuni maxime (297 x 446 mm), cea de-a doua extensie suplimentează memoria pentru buffer-ul de intrare, seturile de caractere și prelucrările de imagini în timpul listării.



Imprimanta PhaserJet PXi poate fi legată la mai multe calculatoare și în rețea. Furnitura de livrare conține și un conector Ethernet.

Preț: 37.000 DM. Imprimanta a fost expusă atît la TIB '91 (standul Xerox, pavilion F) cît și la SYSTEMS '91.

Tektronix GmbH
Colonia-Allee, 11
D-W 5000 Köln 80,
Tel: (0221) 96969 - 0

Virus - Utilities

Firma Ikarus Software a prezentat la Systems, în München trei noi produse: Virus - Utilities Advanced Edition, Virus - Utilities for Windows și programul interactiv de învățare ViruSchool.

Ceea ce ne-a determinat să vă prezentăm produsele acestei firme este maniera nouă de abordare a problemei: a fost elaborat un limbaj de descriere a virușilor, denumit VDL (Virus Description Language), care permite descrierea noilor soiuri de bestii apărute, preluarea acestora în baza de cunoștințe a sistemului, protecția putînd fi extinsă și asupra acestora. Avantajele ce derivă din folosirea unui limbaj de descriere a virușilor ni se par deosebite: cei interesați, "epidemiologii" calculatoarelor, nu vor mai trebui să schimbe între ei viruși adevărați, potențial periculoși, ci doar informații descriptive suficiente pentru combaterea eficientă a lor.

Versiunea 1.5. recunoaște peste 750 de viruși, permite calcularea sumelor de control, a supravegherii sistemului pentru detectarea activității unor posibili viruși și permite refacerea fișierelor infectate chiar și în cazul virușilor care rescriu informațiile din fișier - datele de configurație, sistem și secțiunile program pasibile de a fi rescrise putînd fi salvate, pentru siguranță, pe o dischetă. Protecția este asigurată și sub Windows. ViruSchool este un program școală care permite utilizatorului să învețe, conversînd cu calculatorul, ce este un virus, cum se poate preveni infectarea calculatorului și ce se poate face atunci cînd calculatorul s-a îmbolnăvit. Un pachet complet costă 565 DM.

Firma Ikarus Software editează lunar foaia informativă "Ikarus Hotline", în care sînt prezentate informații la zi din lumea virușilor și a protecției datelor, publicația fiind



distribuită gratuit utilizatorilor înregistrați ai programului.

Ikarus Software GmbH,
Franz Josef str. 7,
5700 Zell am See, Austria,
tel. 0043(0)6542/2941-0
fax: 0043(0)6542/2941-15

DR-DOS 6.0 vs MS -DOS 5.0

Cînd în iunie 1990 Digital Research a scos primul sistem de operare DOS cu un management de memorie îmbunătățit, Microsoft s-a grăbit să anunțe MS-DOS 5.0, care a leșit în lume în iulie '91. Dar prea mult timp de odihnă nu le rămînea soft-istilor din Redmond - cu DR-DOS 6.0, Digital Research încearcă din nou să se distanțeze.

MS-DOS folosește driver-ul "Himem.sys" pentru administrarea memoriei. Pe PC-uri cu 386 sau 486, "emm386.exe" are grijă de cererea de UMB-uri ("upper memory blocks") și, la dorință, de "transformarea" memoriei XMS în EMS (extinsă în expandată). În directorul rădăcină se mai găsește misteriosul, dar pentru Windows indispensabilul, fișier "wina20.386".

DR-DOS oferă "hidos.sys" pentru 286, respectiv "emm386.sys" pentru 386. Driverile oferite de DR-DOS sînt mai eficiente decît cele oferite de Microsoft: "hidos.sys" lucrează direct cu mai multe seturi de cipuri (NEAT, LEAP, SCAT) și permite străpungerea barierei celor 600 coțeți chiar și pe XT-uri dotate cu EMS (pe AT-uri 286, DR-DOS lasă disponibili maxim 628 coțeți, MS-DOS 621 coțeți), iar "emm 386.sys" știe să folosească și memoria video și să copieze ROM în RAM.

Ambele drivere 386 comută procesorul în modul virtual, ceea ce generează probleme în cazul unor programe (de exemplu, cele ce lucrează cu DOS-extenderul de la PharLap, cum ar fi "Mathematica"). Nici Windows nu se mai poate utiliza în modul standard - "emm386.sys" din DR-DOS are pentru aceasta comutatorul "/winstd", care permite apoi lansarea Windows în toate cele 3 moduri.

UMB-urile create se pot folosi prin intermediul comenzilor "devicehigh" în config.sys și "loadhigh" în auto-exec.bat. Astfel, drivere și TSR-uri sînt încărcate în UMB, nemalocupînd memorie "standard", convențională. La toate driverele și utilitarele care fac parte din furnitură, acest lucru funcționează fără probleme, dar la utilitare oferite de terți, instalarea poate deveni delicată. Căci TSR-uri care după lansare "se umflă" sau "slăbesc" (ca de exemplu Shell și Desktop din PC-Tools) nu se prea împacă cu modul de administrare al memoriei propus.

Memoria suplimentară, fie ea extinsă sau expandată, este folosită direct de unele programe, dar, de regulă, cel mai bun lucru iese ca ea să fie folosită de

un cache de harddisk. Dacă aici Microsoft-ul se prezintă cu deja cunoscutul Smartdrive, Digital Research vine cu unul din cele mai performante programe cache disponibile, Super PC-Kwik al firmei Multisoft. În timp ce Smartdrive se fixează în memoria extinsă și mai eliberează din memoria ocupată numai în folosul Windows-ului, PC-Kwik își "adună" singur memoria de lucru și, la cerere, o cedează altor programe. Un driver suplimentar asigură compatibilitatea cu modurile "standard" și "enhanced" pentru Windows. Ambele programe cache "grăbesc" mult harddisk-ul.

Din cele amintite pînă aici este evident că instalarea noilor sisteme de operare nu este o treabă banală. Acesta este și motivul pentru care ambele sisteme de operare sînt însoțite de programe de instalare puternice.

DR-DOS oferă 3 opțiuni: multă "funcționalitate", multă memorie, sau calea de mijloc. Cum "funcționalitatea" depinde însă prea ades de utilizarea particulară a PC-ului și nu întotdeauna se potrivește cu valorile implicite fixate de DR-DOS, nu poate fi evitată setarea individuală a parametrilor sistemului. Lucru care se întîmplă interactiv, prin intermediul unui help online. Din păcate, deseori programul de instalare nu urmărește setările făcute, astfel încît opțiuni care se exclud de fapt generează confuzii (ducînd, pînă la urmă, la consultarea manualului...).

MS-DOS lucrează altfel: trebuie răspuns la cîteva întrebări, după care programul de instalare construiește o configurație standard. Cine vrea să-și acordeze sistemul mai fin, e trimis direct la manualul de utilizare.

Ambele sisteme permit stocarea vechiului sistem de operare pe harddisk și revenirea la el, în cazul că se dorește acest lucru. DR-DOS are pentru aceasta comanda "uninstall", în timp ce MS-DOS creează la instalare o dischetă bootabilă ce rezolvă același lucru.

Ambele sisteme dispun de o suprafață utilizator grafică, așa-numitul shell: DOS-Shell la Microsoft, Viewmax la Digital Research. Chiar dacă ele simplifică mult viața utilizatorului, nici una din ele nu pune realmente accente noi.

Un taskswitcher este de găsît atît în MS-DOS 5.0, unde e complet integrat în Shell, cît și în DR-DOS 6.0, unde este un TSR numit Taskmax. Un taskswitcher (comutator de taskuri) nu are nici o legătură cu multitaskingul. El per-

mite doar păstrarea simultană în memorie a mai multor programe și comutarea dintr-unul într-altul. În timp ce unul lucrează, toate celelalte sînt "înghețate". Taskmax permite în plus și schimbul de date între programe. Cu ajutorul mouse-ului se selectează datele dintr-o aplicație, se comută și apoi se copiază în cealaltă. Pentru ca datele să fie corect interpretate, Taskmax oferă 3 formate.

O găselniță ce va bucura pe mulți este Superstor, pe care Digital Research l-a cumpărat de la firma Addstor, integrîndu-l în furnitura DR-DOS 6.0. Ca și alte programe de acest gen (de exemplu Stacker), Superstor promite dublarea capacității harddisk-ului, interpunîndu-se între controller și DOS pentru a scrie/citi datele pe disc totdeauna în formă comprimată. Timpul care se pierde prin comprimare este recuperat din numărul micșorat de accese la disc, astfel încît, din acest punct de vedere, diferențele sînt greu de sesizat. Programul "sstor.exe" divide harddisk-ul în două părți, din care una rămîne să găzduiască fișierele a căror comprimare nu se dorește. Restul sînt comprimate, iar la primul DIR utilizatorul se poate bucura descoperind o capacitate mărită a harddisk-ului. Dar nici Superstor nu face minuni: datele deja comprimate de "pkzip" sau "lharc" sînt stocate așa cum sînt, nu mai pot fi comprimate. Și fișierele cu grafică ale jocurilor, de regulă, nu mai pot fi comprimate. Cine are multe astfel de fișiere, va fi neplăcut surprins de cît de repede scade capacitatea așezată inițial.

Securitate mărită a datelor oferă ambele sisteme de operare. Microsoft a cumpărat "mirror", "unformat" și "undelete" de la Central Point Software (PC Tools), în timp ce Digital Research oferă "diskmap", "delwatch", "unformat" și "undelete". "Mirror" și "diskmap" salvează FAT-ul harddiskului, pentru a putea regăsi fișiere șterse din greșeală. Cît timp fișierele șterse nu au fost suprascrise, ele pot fi refăcute. "Delwatch" face chiar un pas mai departe, "simulînd" ștergerea înspre sistemul de operare - dar datele rămîn fizic disponibile. De abia cînd s-a umplut discul sau a fost atinsă limita specificată, ștergerea devine ireversibilă. În ambele cazuri, "unformat" face "nefăcută" o formatare. Spre deosebire de MS-DOS, DR-DOS oferă un "supliment" în ce privește securitatea și protecția datelor. Atît PC-ul cît și fișiere individuale pot fi

protejate prin parole. Comanda login permite numai persoanelor autorizate accesul la calculator. "Password" atribuie fișierelor sau directoarelor câte un cuvânt -cod și unul din trei niveluri de acces. În plus, astfel utilizatorul își definește și parola globală, utilizată în mod automat la accesarea fișierelor protejate. O afacere destul de complicată, care riscă să pună un utilizator distrat în situația de a-și pierde toate datele.

Protecția din DR-DOS nu este de netrecut (chiar dacă acest lucru nu mai e posibil prin simplă boot-are de pe o dischetă MS-DOS), dar oferă o protecție suficientă pentru "nespecialiști".

"Diskopt" este un alt utilitar inclus în DR-DOS 6.0 și care relevă grija Digital Research ca sistemul să cuprindă tot ce trebuie pentru o utilizare eficientă a PC-ului "Diskopt" încearcă să optimizeze accesul la harddisk, refăcând în porțiuni continue fișierele fragmentate.

"Filelink" permite o legătură între două calculatoare folosind interfața serială și un cablu standard.

Utilitarele "diskcopy" și "diskcompare", în DR-DOS 6.0, folosesc în mod automat memoria extinsă și memoria expandată sau un fișier ajutător pe harddisk. În fiecare din cazuri, discheta trebuie schimbată o singură dată - în MS-DOS, la un disc HD de 3,5" sînt necesare 7 schimbări de dischetă!. În DR-DOS, o dischetă-sursă poate fi copiată imediat de mai multe ori.

Ca editor, DR-DOS oferă un model ce folosește sintaxa Wordstar. MS-DOS-ul e mai atractiv la acest punct, utilizînd Quick-Basic-Editor.

O surpriză neplăcută cu "format a": în DR-DOS 6.0: dischete noi-nouțe, dar cu porțiuni defecte fizic, sînt formate fără nici un fel de mesaj de eroare. Dușul rece vine atunci cînd dischetele

urmează să fie folosite... MS-DOS cade în extrema calată: cum dă de un sector defect, blochează toată pista. De aceea, pentru formatări, mai degrabă se folosesc alte programe (NU, PC Tools) decît cele incluse în furnituri.

În ce privește help-ul online, ambele sisteme au făcut progrese: toate utilitarele și comenzile își dezvăluie scopul și modul de utilizare dacă liniei de comandă i se adaugă parametrul "?". MS-DOS oferă acest serviciu și prin intermediul comenzii "help <nume-comandă>"; DR-DOS are un help interactiv numit "dosbook", care printr-un sistem de meniuri și referințe încrucișate permite găsirea informației dorite. Digital Research a extins comenzile batch: "switch" permite preluarea unei taste de la utilizator, pentru a sări apoi la o etichetă specificată. Și "config.sys"-ul poate fi (aproape) programat: sînt permise comenzi switch, salturi și etichete. Astfel, PC-ul poate întreba la pornire ce configurație să folosească.

Digital Research furnizează un manual de utilizare de circa 700 de pagini, completat cu un volum mai mic cu indicații de folosire pentru Viewmax. Microsoft livrează un manual de utilizare de 800 de pagini, plus o scurtă introducere. Ambele manuale își îndeplinesc misiunea, oferind suficient ajutor mai ales la deciziile care privesc administrarea de memorie.

Mai rămîne problema compatibilității, pe care cel ce face testul o poate rezolva în două feluri: fie instalează toate softurile de care dispune, apăsă cîteva taste, mișcă un pic mouse-ul și încearcă să tragă niște concluzii, (din păcate, incompatibilitățile de regulă nu sînt vizibile decît în anumite cazuri speciale, greu de verificat), fie caută de la bun început un soft "problematic" (caz

în care riscul de a nu merge" este la fel de mare sub MS-DOS 5.0 ca și sub DR-DOS 6.0. Windows nu are nici un fel de probleme sub DR-DOS (cu condiția ca înainte instalării să se dea comanda "memmax -L -U"). Toate softurile încercate merg fără probleme. Ar mai fi de zis că între timp DR-DOS este așa de răspîndit încît producătorii de soft se vîd obligați să-și testeze noile produse sub ambele sisteme.

În concluzie, care ar fi DOS-ul mai bun? Digital Research oferă, fără nici un dubiu, tehnici mai rafinate și funcții mai multe. MS-DOS face o impresie mai compactă, deoarece toate utilitarele folosesc aceeași suprafață SAA cu mouse și meniuri pull-down. Dacă Microsoft se bizuie pe vechi cumpărători angroșiți, Digital Research caută să obțină contracte OEM (Original Equipment Manufacturer) și ofertanți de hardware ieftin.

Dar o trecere de la MS-DOS 5.0 la DR-DOS 6.0 (sau invers) probabil că nu merită (pentru cine trebuie să o plătească...). Cine pornește acum, este probabil mai cîștigat cumpărînd DR-DOS 6.0. Accente realmente noi nu sînt puse nici de Microsoft, nici de Digital Research. De fapt, au fost integrate în furnitura de livrare utilitare pe care producători independenți le oferă de mult. În ultimă instanță, sînt vindecate numai simptomele care fac ca DOS să fie "trecut" și inadecvat exploziei informaționale din ultimii ani: un "taskswitcher" înlocuiește un multitasking real, un optimizor de harddisk mai evită un pic slăbiciunile sistemului de fișiere bazat pe FAT și gîndit de fapt pentru dischete, driver-ele de memorie creează iluzia utilizării lineare a spațiului de adrese.

(I.F.)

Fabrica de zvonuri

Prima reacție la apariția DR-DOS 6.0

Microsoft-ul dezvăluie cîteva din "secretele" viitoarei versiuni. Patru sînt inovațiile decisive care urmează să propulseze Microsoft-ul în fruntea plutonului în cursa tehnologiilor: flat memory-model, noul sistem de fișiere, multitasking și funcții de rețea. Dacă asta vă amintește de OS/2, nu puteți greși. Într-adevăr, DOS 6.0 urmează să ștergă granițele între DOS, Windows și OS/2. Cel mai evident devine acest lucru în modul de administrare a memoriei: DOS folosește excesiv structura de segmente "prescrisă" de primele procesoare Intel. De aceea, memoria este divizată în bucățele mărunte, astfel încît programatorul trebuie să jongleze nu numai cu registrele de adresă, ci și cu alt segment. Cel tîrziu cu procesoarele 386 acest lucru a devenit necesar, dar, pînă acum, DOS era un sistem de operare axat pe 8088 în fața căruia toate procesoarele erau egale. DOS 6.0 urmează să schimbe această stare de fapt. Noua lozincă este "jos divizarea memoriei, jos bariera celor 640 koctei". Sigur, este o întrebare

dacă se va păstra compatibilitatea. Cum este sigur că numai posesorii de 386, 486 sau 586 vor putea să se bucure de noua administrare a memoriei. Tot ei, privilegiații, vor primi ca "supliment" un multitasking adevărat sub DOS. Nimic din toate acestea nu vor putea bucura posesorii de calculatoare cu 8088 sau 80286.

Nu vor exista însă diferențe în sistemul de gestiune al fișierelor. Noul DOS va ști să lucreze cu diverse sisteme de fișiere, printre ele HPFS din OS/2. Un lucru pare cert de pe acum: sistemul de fișiere actual poate fi îmbunătățit în așa fel încît să păstreze "aspectul" DOS, dar să lucreze de două ori mai rapid.

Ca termen de apariție pentru DOS 6.0 este planificat sfîrșitul anului 1992, chiar dacă acest lucru este la fel de nesigur ca și lista definitivă a caracteristicilor și numele "bebelușului". Încă este posibilă redenumirea sistemului de operare (MS-DOS/2 e o propunere), integrarea în Windows 3.1 (sau 3.2 sau 4.0) sau abandonarea lui completă, datorită unor probleme tehnice insurmontabile.

Le roi est mort! Vive le roi!



Ashton Tate: Firma americană de software care a proiectat S.G.B.D-ul dBase, cel mai utilizat program de gestiune a bazelor de date. Această definiție putea fi citită în 1984 într-un dicționar de informatică. În schimb între algebra booleană și Btx nu se găsea nici o definiție despre Borland și nici despre Kahn Philipp.



Probabil că această situație l-a supărat oarecum pe Philipp, născut în 1952, la Paris, într-o familie evreiasco-germană, în timp ce plonja în bazinul său de înot cu vedere spre plaja californiană. Probabil că șeful firmei Borland, destul de reținut în ceea ce privește informațiile despre viața sa personală - știm doar că nu doarme decât patru ore pe noapte, că urâște televiziunea, (nu știm dacă pe motiv de emanueli), că nu se uită niciodată la televizor și ca urmare câștigă o mulțime de timp, că este un reputat muzician de jazz și că este vegetarian convins - tocmai își consuma legumele pasteurizate la bordul propriului yacht de croazieră Dolphin Doncer. Probabil că nu dorea decât să se răzbune pe o instituție americană, ca Ashton Tate de pildă, pentru faptul că Biroul American de Imigrări l-a declarat în 1985 emigrant ilegal.



Oricare ar fi fost motivul, la 10 iulie '91, Philipp Kahn a înghițit firma Ashton Tate. Un pește mare într-adevăr, plăcerea de a face cumpărături costându-l 440 milioane dolari. Dar de unde a avut atîția bani, se întreabă cei din branșă? Ashton Tate a avut în anul trecut o cifră de afaceri de 230 milioane (pierderi: 18 milioane), Borland a avut o cifră de afaceri de 227 de milioane (cîștig: 27 milioane). Să fi fost firma IBM cea care i-a împrumutat sumița, doar pentru a înfige o sulă în



coasta firmei Microsoft? Asemănător modului în care în 1983, Philipp înființa în Scotts Valley firma Borland International, cu bani tapaji de la prieteni?

Omului i s-a mai încredințat o sarcină ingrată: cine nu-și amintește de acțiunea juridică deschisă de Ashton Tate împotriva firmei Fox Software, care a eșuat deoarece un judecător american a declarat dBase-ul, din cauza viciilor de formă strecurate în copyright, liber ca pasărea cerului? Judecătorul a făcut între timp recurs, din proprie inițiativă, așa că încă tot nu se știe dacă vicleana vulpe, cu al ei program Foxbase (între timp Foxpro 2.0), mai bun și mai rapid, a șterpelit sau nu ceva de la dBase.

Între timp s-a stabilit că firma Ashton Tate a produs singură produsul dBase, de la prima pînă la ultima construcțiune, și că nu a dezvoltat doar produsul Public - Domain elaborat de firma JPL pentru ministerul apărării.

Pînă la urmă nici Philipp Kahn nu se poate reține de a mai arunca cîte o privire și prin curțile vecinilor. Programul său de calcul tabelar de succes Quattro era destul de înrudit cu un alt program de calcul tabelar, care stăpînea piața și care se cheamă Lotus 1-2-3.

Se făceau deja speculații, avocații își frecau bucuruși minile, numai că la 11 octombrie '91 Departamentul Justiției (DOJ), din SUA, aprobînd achiziționarea firmei Ashton Tate de către firma Borland International, a pus și o condiție: Borland să nu se mai judece cu Fox Software Inc. pentru drepturile de copyright pentru familia de produse dBase. DOJ afirmă că drepturile de copyright pentru structura meniurilor și

limbajele de comandă "promovează monopolurile în detrimentul inovației".

Cînd valurile se vor liniști, Borland va deține un plus de 50% pe piața sistemelor de gestiune a bazelor de date. Adunînd segmentele de piață deținute în 1990 în SUA de Paradox și de dBase se ajunge la un procent de 74.3%. La vînzări noi în '90, după IDC, situația prezentîndu-se în modul următor: 64% dBase, 17% Paradox, 11% Superbase, 4% Super DB, 3% Dataease, 1% Foxpro. Avînd în vedere faptul că în lume există peste 3 milioane de utilizatori dBase, în viitor apropiat cele două produse vor coexista, așteptîndu-se versiunile Windows atît pentru dBase cît și pentru Paradox. Va exista un produs de migrare între cele două produse, care se va chema "Object dBase", denumit intern la Borland și Turbo-XBase, produs care va fi compilator și interpretor dBase în același timp, va fi compatibil dBase III și dBase IV și va avea o interfață Paradox.

Produsele Borland vor fi astfel concepute, după cum afirmă Philipp, încît operarea unui calculator să fie mai ușoară decât a unui casetofon. El afirmă că în viitor calculatoarele vor exista pentru a lucra așa cum lucrează oamenii și nu pentru a-i constrînge pe aceștia să lucreze așa cum lucrează ele.

Dar să vedem cum a receptat vestea tranzacției Eduard Altmann, redactor de peste 10 ani la revista economică DM din Germania:

»"Firma Ashton Tate cumpărată de firma Borland", pe cine n-a șocat în urmă cu cîteva săptămîni această știre? Ei bine, pe mine nu. Mărturisesc că sînt și eu vinovat de această tranzacție. Cel puțin moral. În mine am știut

că așa se va întâmpla încă de mult, de când testînd pachetul încă fierbinte al noului, pe atunci, dBase 4.0, și încercînd să sortez 300 de articole, am primit mesajul: "Not enough memory". Și aceasta cu toate că am eliminat din memorie tot ceea ce se putea. Mi-am spus atunci: această firmă este în declin, trebuie să existe o pedeapsă.

Dar au fost și momente frumoase cu dBase. Acestea au fost încă și mai înaintate, cînd am primit o copie de dBase II aproape fără greșeli. În sfîrșit datele de pe hard-disk-ul meu n-au mai fost trimise în nirvana de o putere necunoscută.

Îmi amintesc cu plăcere și de dBase III+. Deoarece datorită programului "Asistent" n-am mai fost neajutorat și am putut să-mi scriu programele linie cu linie, învășînd să plătească mulți bani pentru aceasta, în timp ce eu n-a trebuit decît să-mi sacrific o parte din timpul liber. Oricum ce altceva aș fi avut de făcut serile și la sfîrșit de săptămîină?

După cîțiva ani de studiu zadarnic a manualelor, mi-a pus cineva la inimă, mulțumescu-ți ție Philipp, un SGBD exotic denumit Paradox. La început neîncrederea a fost mare: o suprafață utili-

zator cu totul nouă, fișiere incompatibile și ceea ce era mai rău un limbaj de programare străin, numit PAL. Numai din masochism profesional am încercat cu Paradox cîteva calcule statistice și spre surprinderea mea în cîteva minute am obținut, din pustiul de date, rezultate pentru a căror obținere ar fi trebuit să lucrez la un program în dBase III zile întregi.

Începînd de atunci mi-a spus și "Borland ceva. Chiar dacă generatorul de programe a rămas pentru mine o carte ferecată cu șapte lacăte, nedepășit în filozofia sa secretă decît cel mult de PAL, și încă n-am învățat să-i

iubesc suprafața utilizator antică cu comenzi îmbricate totuși Paradox își face treaba fără blocaje și fără programe.

În bucătăria de software a lui Philipp sîrîie din nou ceva. Primele mirosoare ale unei noi mîncări numită Parabase răzbat înafară și produc apetit: posibilitatea de a lucra în rețea, tempoul Paradox-ului, flexibilitatea lui Object Vision, confortul în operare al lui Windows și utilizatorii lui dBase - miroase a inserare între algebra booleană și Btx.

(Romulus Maier)

De ce a cumpărat Novell Digital Research?

"Fuziunea este un pas absolut logic, căci produsele se completează" - declară un reprezentant Novell. Din fuziunea celor doi producători de sisteme de operare, care pînă acum vizau piețe de desfacere distincte, se naște un adversar serios pentru IBM, Microsoft, SCO, AT&T și Apple. Căci, dintr-o singură lovitură, Novell devine un ofertant de sisteme de operare atît pentru locuri de muncă individuale (stațiile client în rețea), cît și pentru server-e. Un rol decisiv joacă în acest cadru DR-DOS, apărut în versiunea 6.0. Netware se orientează după DOS, astfel încît pînă acum Novell depindea mult de Microsoft. Prin intermediul DR, Novell posedă acum o tehnologie DOS proprie și poate juca un rol mai activ. Cu alte cuvinte: DR-DOS urmează să fie extins spre a fi completarea ideală pentru Netware.

Deoarece liniile de produse ale celor două întreprinderi se completează în mare măsură, se naște întrebarea unor sinergii posibile (puterea comună să fie mai mare decît suma puterilor individuale). În cel puțin 6 domenii, cele două întreprinderi pot profita de experiența celeilalte.

Front-end grafic: suprafața utilizator din Netware se bazează în continuare pe DOS. Un front-end grafic, care să ușureze munca utilizatorilor și administratorilor de sistem, lipsește. Sisteme precum Windows conlucrează cu Netware, dar nu sprijină automat funcțiile Netware. Ar fi posibilă o suprafață utilizator comună pentru DOS și Netware, bazată pe GEM.

Funcții de rețea în DOS: protecție prin parole, blocarea de fișiere și articole (file and record locking) sînt cuvinte străine unui programator DOS obișnuit, motiv pentru care, de regulă, pentru utilizarea în rețea trebuiesc programate versiuni speciale. Cu integrarea unor astfel de funcții în DOS (și astfel și în Windows) administrarea rețelelor ar putea să se simplifice, pentru că toate programele DOS ar fi utilizabile și în rețea.

Multitasking cu Netware: DR are multă experiență în sisteme multitasking, iar în prezent sînt puține case de soft care oferă NLM-uri (Netware Loadable Modules), cu ajutorul cărora programe independente să poată fi executate pe server. Poate că în viitor va exista o cale mai simplă pentru a scrie aplicații client-server pentru Netware.

Rețele ieftine (low-end-LANS): Novell și-a manifestat deja dorința de a nu "abandona" piața rețelelor ieftine unor firme ca Artisoft (Lantastic). Cu Multi-user DOS, Digital Research posedă deja o soluție în această direcție.

Sisteme de operare în timp real: cu FlexOS, DR oferă un sistem de operare în timp real care se potrivește de minune pentru supravegherea stocurilor, facturări și urmărirea producției. O piață lucrativă pentru Netware.

Comenzii identice pentru DOS și Netware: ceea ce la Novell se cheamă NDIR, NCOPY și SALVAGE, se numește XDIR, XCOPY și UNDELETE în DR-DOS. Cît timp încă?

Conform declarațiilor făcute de reprezentanți ai celor două firme, planuri exacte încă nu s-au stabilit. Dar este sigur că nu va trece mult timp pînă să apară primele fructe ale acestei fuziuni.

(I.F.)

Balanțe contabile

Legea obligă toate întreprinderile, indiferent de forma lor de organizare, să țină o contabilitate în care să se regăsească toate tranzacțiile efectuate. Și cum spectrul Gărzii Financiare plutește deasupra tuturor întreprinderilor și întreprinzătorilor, fie că este vorba de magnați ai industriei gogoșilor, de proprietari de birt, de producători numărul 1 în și în, sau de directori de combinate, centrale, etc., atunci când acești șerifi ai spațiului contabilicesc fac o descindere, pe masă trebuie să apară și balanța contabilă. Și atunci când aceasta nu este curată, trebuie deschisă punga.

Există contabili și probabil că vor mai exista, care nu au încredere decât în situațiile făcute de mână. Dar există, și au existat contabili care au acceptat faptul că acestea s-ar putea face și altfel. Așa au apărut primele balanțe pe calculator, implementate mai întâi, cu mult entuziasm, pe așa-numitele mașini de facturat și contabilizat, apoi cu același entuziasm rămas neștirbit pe toată gama de calculatoare existentă în țară, de la TDF, M18, mini și până la Felix, entuziasm care a luat parcă o și mai mare amploare odată cu intrarea în țară a PC-urilor. Din acele vremuri a rămas celebra zicală: "Dați-mi un Cray și să vedeți ce balanță vă implementez!"

După datele pe care le avem există la această oră în țară sute de variante de balanțe contabile doar pe PC-uri, posibil chiar mii. Este mai ceva decât la chinezi, la cules de orez în luna mai, un colectiv de programatori, o balanță contabilă, ba chiar două. Numai că nu este suficient să dai viață unei aplicații, trebuie s-o întreții, nu poți spune s-a murdărit puradelu, nu-i nimic, facem altul. Ne imaginăm cu groază reacția provocată de o cât de mică modificare de legislație: toată floarea cea vestită a programatorilor făcând aceleași modificări în programele proprii, originale.

Sîntem de părere că înainte de a începe proiectarea unei aplicații este bine să aruncăm o privire în jur pentru a vedea dacă nu există deja ceva ce ne-ar putea satisface nevoile. Credem că și la noi va trebui să se creeze o industrie și o piață de software. Nu ne îndoim de faptul că nu peste mult timp vom putea cumpăra programele de care vom avea nevoie așa cum astăzi cumpărăm o carte sau o bucată de salam.

Desigur există și contraargumente: diversitatea întreprinderilor este mare și a cerințelor contabililor așisderea, amintim doar că unii contabili își țin contabilitatea cu ajutorul jurnalelor contabile, în timp ce alții folosesc fișele șah, nu toate documentele sînt tipizate și nu-i ușor să provoci o modificare de tabieturi. Din experiența trecutului știm că nu de puține ori întreprinderile care au preluat "de gata" anumite aplicații au trebuit pînă la urmă să le întrețină cu forțe proprii, uneori trebuind să depună mai mult efort decât dacă ar fi proiectat singure aplicațiile cu pricina. Tot din experiență mai știm că încercările de realizare a unor fluxuri informatice, nu de puține ori în prezența unor fluxurilor informaționale deficiente, au fost încununuate de succes foarte rar, reușita datorîndu-se în acele cazuri, în mod substanțial, contactului direct dintre proiectant și utilizator. Și totuși sîntem prea puțini pentru a ne putea permite să facem același lucru și încercînd să facem totul, nu realizăm pînă la urmă mare lucru.

Încercăm, prin acest articol, să lansăm o serie de teste comparative de produse soft autohtone. Pentru subiectul ales, balanțele contabile, am testat patru produse provenind din tot atîtea localități. Am dori să subliniem de la început faptul că toate cele patru produse sînt operaționale, au fost implementate cel puțin o dată și fiind întreținute în permanență, (așa

cum am fost asigurați), cu siguranță se vor mai perfecționa. Nu avem pretenția de a vă prezenta cele mai bune patru produse existente la această oră în țară și nici de a face o prezentare exhaustivă a acestui gen de produse. (Este posibilă existența unor produse mai bune, iar o prezentare exhaustivă ar fi imposibilă).

Mărturisim că am încercat contactarea mai multor producători de astfel de aplicații și că cei rămași în competiție au cel puțin un merit, acela de a fi acceptat, la solicitarea noastră, focul întrecerii.

Aplicațiile testate sînt foarte diferite atît în ceea ce privește aria de probleme acoperite cît și ca modalitate de prezentare, limbajul de programare folosit etc. Cea mai simplă se rezumă la realizarea calculului efectiv necesare în formula de bilanță de verificare cu 8 coloane astfel:

- pornind de la solduri inițiale sau total sume anterioare
- introducînd rulajele debitoare și creditoare pentru fiecare cont
- editează situații tip bilanță cu total sume cumulate și sold final calculat.

Extrema cealaltă caută să acopere pe cît se poate integral domeniul financiar contabil:

- se pornește de la documentele de bază care sînt utilizate în domeniul
- se realizează evidența analitică la nivelul unor conturi
- se generează automat informații tip notă contabilă
- se asigură interfața către aplicații tip GESTOC și MIFIX
- se editează rapoarte tip:
 - jurnale contabile
 - fișe tip șah
 - balanțe la nivel de conturi sintetice și analitice
 - editare bilanț contabil și anexe la bilanț.

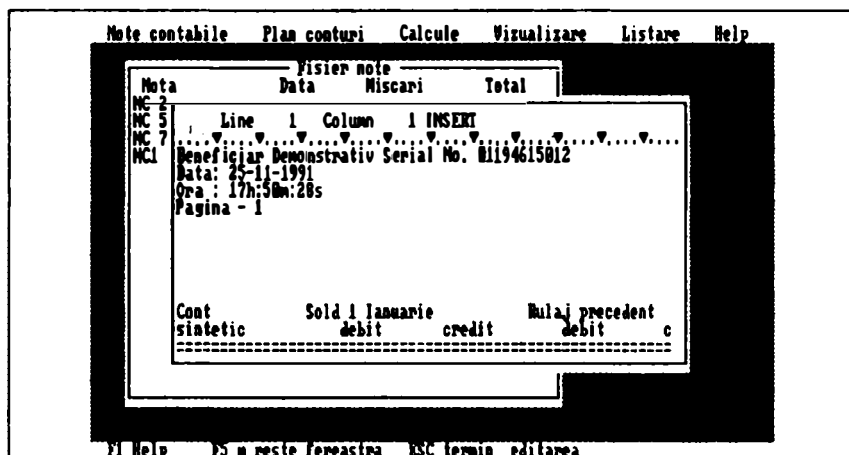
BALPRO

BALPRO este o balanță contabilă elaborată de firma Lotus-S.R.L. din Brăila. Lucrarea este disponibilă în mai multe medii de programare: Turbo Pascal, C, Quattro, Foxbase, dBase III și dBase IV.

Pornind de la solduri inițiale, nomenclator (plan) de conturi și note contabile, lucrarea permite vizualizarea și/sau listarea balanțelor analitice și sintetice, a fișelor șah, a notelor contabile încărcate și permite, de asemenea, și investigarea separată a operațiilor din fiecare lună a anului.

Este singura lucrare testată care permite o configurare proprie a listelor și care oferă utilizatorului, la nivel de help, și informații privind reglementările în vigoare din domeniul contabilității și un dicționar de termeni permițând familiarizarea cu domeniul și a utilizatorilor ne-specialiști.

Aplicația permite, de asemenea, importul/exportul unor fișiere în format SDF și dBase permițând preluarea datelor de balanță și în alte aplicații.



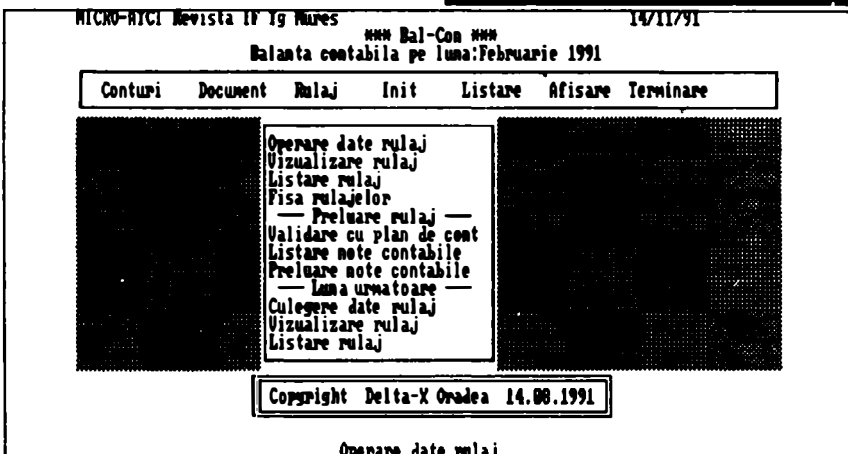
Firma Lotus S.R.L. mai pune la dispoziția utilizatorilor și aplicațiile FIXPRO (mijloace fixe) și SALPRO (salarizare/personal) și are în faza de testare o aplicație care va integra gestiunea magaziiilor, a recepțiilor de marfă, a vânzărilor către beneficiari, a facturărilor de marfă, urmărirea încasărilor, salarizare/personal, evidența mijloacelor fixe, balanță contabilă. (Aplicația va fi disponibilă după 1.01.1992).

BAL-CON

Bal-Con este o aplicație destinată evidenței contabile a unităților economice și este elaborată de firma Delta-X SRL din Oradea.

Aplicația realizată cu ajutorul produsului Clipper face parte dintr-un complex de lucrări din care fac parte și aplicațiile MIFIX și GES-TOC.

ăi mai fain !



Pornind de la documentele primare folosite pentru înregistrări contabile, note contabile, solduri inițiale și nomenclator de conturi, aplicația permite listarea ba-

posibilitatea înglobării în lucrare a unor evidențe analitice la nivel de cont contabil, cum ar fi evidența mijloacelor bănești, evidența debitori/creditori, etc. Se permite și realizarea balanțelor pe secții și centralizarea acestora la nivel de întreprindere.

BAL-CON este singura aplicație testată care permite definirea unor corespondențe de conturi și efectuează validarea on-line în momentul introducerii datelor contabile.

Documentația produsului poate fi vizualizată în orice moment. Având programul Norton Guide instalat rezident, la o simplă tastare a combinației de taste Shift-F1, se intră în documentație și se poate naviga în ea pentru lămurirea problemelor dorite.

Din furnitura de livrare face parte și un program de instalare care permite stabilirea parametrilor monitorului, a imprimantei, a parametrilor de implementare, inițializare fișiere, indexare fișiere, refacere rulaj cumulată. Pot fi utilizate imprimante de tipul: Epson FX - 1050/850, RCD 9335 Scamp, EP-2001, NX-1500 STAR sau alte tipuri de imprimante, putându-se specifi-

lanțelor de verificare la nivel de conturi analitice și/sau sintetice, a fișei sintetice șah, a extrasului de cont, a listei de venituri și cheltuieli, a defalcării cheltuielilor pe capitole, etc.

Lucrarea permite defalcarea conturilor sintetice pe patru nivele (max. 11 cifre) și ca urmare la intrare pot fi folosite atât note contabile cât și documente primare: carnete CEC, chitanțe, dispoziții de încasare/plată, etc., ceea ce are ca efect

ca numărul de linii/pagină, numărul de coloane/linie și secvențele de control pentru densitate mare, medie și normală. De asemenea se poate specifica portul la care este conectată imprimanta (LPT1, COM1, COM2).

Sugerăm proiectanților câteva îmbunătățiri la nivelul dialogului cu utilizatorul: considerăm că revenirea de pe un nivel inferior de meniuri, pe nivelul imediat superior se realizează mai comod prin apăsarea tastei Esc decât prin selectarea unui punct din meniu care să se numească "Terminare".

De asemenea în timpul prelucrării lor ar fi bine să se afișeze mesaje informative privind stadiul prelucrării (de ex. în programul de instalare, la indexare, singurul indicu al acceptării comenzii este pîlpitul LED-ului harddisk-ului, terminarea execuției comenzii fiind indicată doar de încetarea acelu pîlpit).

Firma Delta-X mai are realizate și aplicațiile Mi-Fi - evidența și calculul amortizării mijloacelor fixe și GE-MA - gestiunea și evidența contabilă a valorilor materiale, care împreună cu prezenta lucrare pot fi tratate ca un subsistem integrat. De asemenea în colaborare cu un colectiv de proiectare din Suceava este în curs de realizare o aplicație de personal-salarizare.

CONTAS

CONTAS - contabilitate analitică și sintetică, este o aplicație elaborată de întreprinderea ELINCO SRL din București, cu ajutorul produsului Foxpro.

Pomind de la nomenclatorul de conturi, solduri inițiale, note contabile, facturi, dispoziții de plată/incasare, se editează jurnalul înregistrărilor, balanțele maistru-șah debit/credit, fișa contului analitic/sintetic, balanțele de verificare analitice/sintetice. Aplicația are înglobate și gestionarea facturilor și a dispozițiilor de plată/incasare. De asemenea în suprafața utilizator a programului este înglobată și apelarea editorului de texte QEdit.

Produsul nu s-a comercializat pînă acum separat, fiind oferit gratuit întreprinderilor care au cumpărat calculatoare de la firma Elinco SRL

BALL

BALL - balanță expert, este o aplicație realizată de ing. Flaidăr Alin și comercializată de întreprinderea Micro ATCI Tîrgu Mureș.

Avînd la intrare nomenclatorul

Cursor: F117, Ctrl+F, Sterge? Del; Sterge la stînga: BS; Inserare: Ins; sus/jos: PgUp, PgDn

Nr. cont	1	Nume	Debit	Credit
100	0	MIJLOACE FIXE	823204.00	0.00
110	0	VALORI MIJL. FIXE	0.00	150420.00
123	0	INVESTITII IN CLUB	823204.00	823204.00
200	0	MATERIALE	1592553.69	1713586.32
200	1	MATERIALE	295488.00	1291978.00
200	4	MATERIALE	0.00	0.00
200	6	MATERIALE	0.00	0.00
200	7	MATERIALE	0.00	0.00
200	19	MATERIALE	0.00	0.00
201	0	OBIECTE DE INVENTAR	152275.52	243215.26
201	1	OBIECTE DE INV. IN DEP.	152275.52	235563.86
201	2	ECHIP. PROT. IN DEPOZIT	0.00	7651.40
202	0	OBIECTE DE INV. IN FOLOS.	603230.37	183168.47
202	1	OBIECTE DE INV. IN FOLOS.	217463.55	183168.47
202	1	OB. INV. UNIT. LOC.	0.00	0.00
202	2	ECHIP. PROT. IN FOLOS.	385766.82	0.00
202	2	ECHIP. PROT. UNIT. LOC.	0.00	0.00
203	0	VALORI OB. INV. SI ECHIPM.	153418.67	-351547.00
203	2	VALORI OB. INV. F. LOC.	0.00	0.00

F1-Ajutor F2-Sortare/salvare F3-Sterge F4-control F5-Salt la... F10-Exit

Editarea balanței este condiționată de existența egalității balanțiere.

O altă caracteristică importantă a acestui produs este posibilitatea realizării simultane a balanțelor contabile pentru mai multe întreprinderi. Un contabil care ține evidența contabilă a mai multor întreprinderi are astfel posibilitatea de a încheia balanțele contabile la timp, pe același calculator.

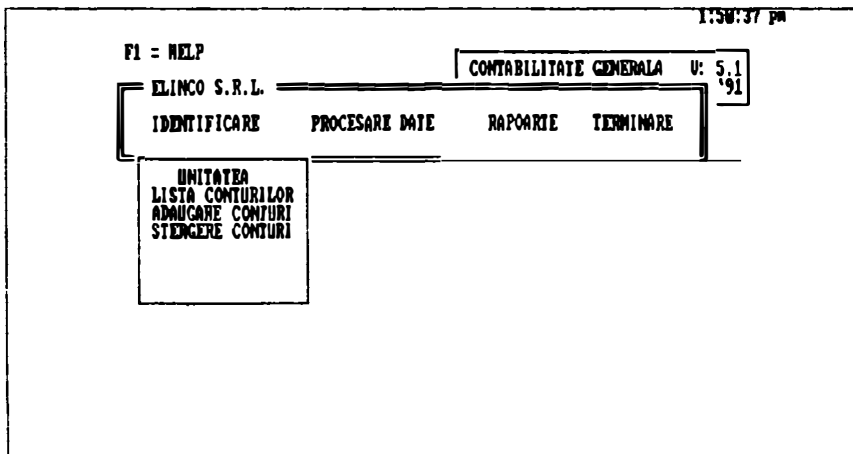
Vizualizarea situațiilor înainte de listare nu se poate realiza în mod WYSIWYG, nepermițându-se o defilare stînga-dreapta în situații, așa cum este posibil de pildă cu produsul BALPRO.

de conturi, soldurile inițiale și notele contabile se permite listarea balanțelor sintetice/analitice, a situației conturilor și a fișelor șah. Lucrarea permite și obținerea de rapoarte pe perioade mai mici de o lună, sau pentru secțiunile unei întreprinderi.

Realizată în Turbo Pascal 5.5 aplicația lucrează aproape exclusiv în memorie, ceea ce are ca efect timpi foarte mici de răspuns în prelucrări, dar și limitări; astfel se permit cel mult 7.000 de conturi și 10.000 de înregistrări contabile, ceea ce de regulă este suficient. Listarea se poate face pe imprimante compatibile Epson, sau pe Scamp-uri.

Este în curs de realizare o nouă versiune în Turbo Vision, care va efectua validarea conturilor introduse (ceea ce în versiunea actuală nu se realizează), va include tehnici de editare multidocument, listare în background cu PrintManager și o structurare mai avansată a informațiilor.

mat. Galfi Nicolae
ing. Maier Romulus



Balante contabile				
Produs	BALPRO	BAL-CON	CONTAS	BALL
Producător	Lotus S.R.L. Brăila Tel. 946/35544	Delta-X Oradea Tel. 991/34257	Elenco București Tel. 90/175300/175	Micro ATC Tg. Mureș Tel. 954/31660
Preț (lei)	20.000	55.000	-	30.000
Documente de intrare:				
Nomenclator (plan) de conturi	da	da	da	da
Solduri inițiale	da	da	da	da
Note contabile	da	da	da	da
Documente primare din contabilitate	nu	da	nu	nu
Facturi	nu	da	da	nu
Dispoziții de plată/Incasare	nu	da	da	nu
Situații care pot fi vizualizate/Imprimate*				
Balanța analitică	da	da	da	da
Balanța sintetică	da	da	da	da
Fișa șah	da	da	da	da
Note contabile	da	da	da	da
Extras de cont	nu	da	nu	da
Lista de venituri și cheltuieli	nu	da	nu	nu
Defalcarea cheltuielilor pe capitole	nu	da	nu	nu
Situația conturilor	nu	nu	nu	da
Rulaje pe conturi	nu	nu	da	nu
Fișa contului	nu	nu	da	nu
Grafic anual	da	nu	nu	nu
Documentație/help-uri, interfață utilizator				
Reglementări în vigoare	da	nu	nu	nu
Dicționar de termeni	da	nu	nu	nu
Îndrumar implementare/utilizare	da	da	da	da
Ferestre help generale	da	da	da	da
Help contextual	nu	parțial	nu	nu
Linie informativă (status)	da	da	da	da
Taste funcționale	da	da	da	da
Calculator de buzunar	nu	nu	nu	nu
Cerințe hardware (configurație minimă):				
Tip calculator	XT	XT	AT	XT
Memoria minimă necesară	nu/recomandat	nu/recomandat	da	nu/recomandat
Harddisk necesar ?	256 KB	512 KB	640 KB	256 KB
Spațiu disc	cca. 300 KB	cca. 1,2 MB	cca. 1,2 MB	cca. 240 KB
Alte caracteristici:				
Număr maxim conturi	nelimitat	nelimitat	nelimitat	7.000
Număr maxim operații contabile	nelimitat	nelimitat	nelimitat	10.000
Utilizabil în rețea	nu	nu	da	nu
Interfețe cu alte aplicații				
Mijloace fișe	da (SDF/dBase)	da	nu	nu
Gestiunea stocurilor	nu	da	nu	nu
Facturare	nu	nu	da	nu
Salarii	da (SDF/dBase)	în lucru	nu	nu
Altele				
Protecție	certificare	certificare	dischetă cheie , parolă	certificare
Mediu de programare	Turbo Pascal, C Quattro, Foxbase dBase III, dBase IV	Clipper	Foxpro	Turbo Pascal

* Notă: Unele dintre situațiile enumerate sînt asemănătoare principal, avînd doar nume diferite

Periferice de intrare

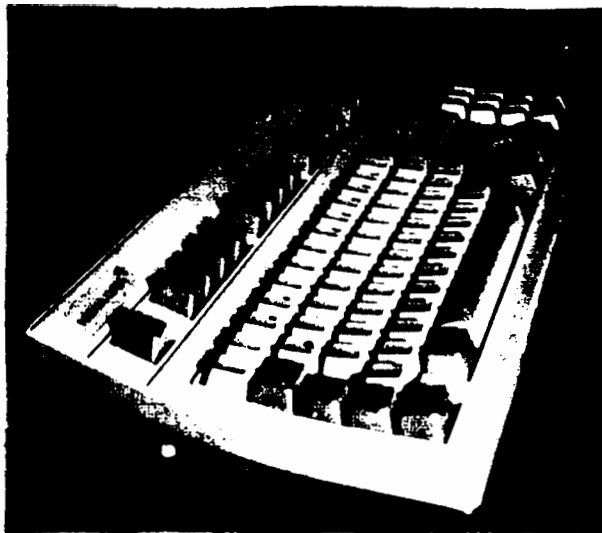
Articolul care urmează va prezenta perifericele de intrare care stau astăzi la dispoziția utilizatorului, modul lor de funcționare, care este utilitatea lor practică și care va fi viitorul acestor interfețe dintre om și mașină. Unele dintre perifericele care urmează au fost tratate mai detaliat în numere anterioare ale revistei "if", de altele ne vom mai ocupa în numerele viitoare. Ceea ce ne interesează în momentul de față este o privire de ansamblu asupra întregului.

Chiar și unul dintre cele mai bune calculatoare, un 80486/33 MHz cu procesor cache și cu alte accesorii care îi măresc performanțele poate fi doar atât de bun, cât îi permite soft-ul și modul în care este deservit. De multe ori, datele provenite de la alte calculatoare pot fi transferate automat, dar fără o introducere manuală a datelor un PC nu poate fi folosit nici în stadiul de astăzi al tehnicii.

Din păcate omul este foarte lent în comparație cu calculatorul, în plus el are nevoie de aparate (echipamente periferice) de intrare convenabile. De aceea se încearcă să se pună la dispoziția utilizatorului unui calculator un aparat de intrare cât mai necomplicat și ușor de manipulat. În acest sens este valabilă regula: cu cât sînt mai complexe datele de intrare ale unei anumite aplicații, cu atît trebuie să fie mai complex aparatul de intrare. Deci pentru intrări simple este nevoie de un aparat de intrare simplu.

De-a lungul ultimilor ani s-a dezvoltat combinația dintre mouse și tastatură. Ambele aparate de intrare combină, printr-un compromis, propriile lor avantaje și dezavantaje. Nu în ultimul rînd la acest lucru a contribuit și adaptarea software-ului. Introducerea datelor via tastatură și mouse este sprijinită de Windows 3.0 de pe calculatoarele MS-DOS, de suprafețele grafice de sub Unix, de exemplu Motif, Sun-

Tastatura - cel mai utilizat periferic de intrare



view sau Open Desktop cît și de Apple Macintosh. Dar mai sînt și alte aparate de intrare care ajută omul în introducerea rapidă, necomplicată și simplă a datelor într-un calculator.

Pentru preluarea textelor tipărite se folosește un scanner cu soft OCR (Optical Character Recognition) corespunzător. Cu acesta imaginile sînt digitalizate și prelucrate în calculator.

Un joc, un robot sau o simplă conducere (comandă) pot fi deservite foarte bine cu un joystick, iar pentru aplicațiile grafice există tablete digitizoare și mouse-uri. Pe lîngă acestea mai există în anumite domenii ecrane sensibile la atingere, creioane optice (lightpen) sau combinații ale tuturor mediilor de intrare. Există deja, deși încă în fașă, și introducerea vocală prin microfon.

Principiul de funcționare al aparatelor uzuale de intrare, cine le utilizează și cît de eficace este interfața om-mașină pentru utilizator vor fi arătate în continuare:

Tastatura Ergonomia lasă de dorit

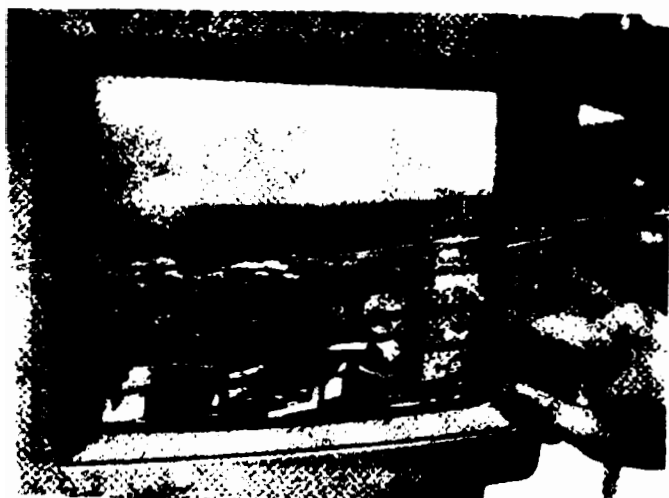
Nici un calculator nu este utilizabil fără tastatură. Tastatura este fără nici o discuție cel mai des folosit aparat de intrare pentru calcula-

toare. Aceasta se explică prin faptul că pentru elaborarea textelor s-a folosit mașina de scris încă înaintea erei calculatoarelor. Era deci de la sine înțeles să se preia sistemul mașinii de scris. Din punct de vedere ergonomic o tastatură este extrem de neprietenosă. Utilizatorul trebuie să-și țină mîinile într-o poziție care nu permite o muncă relaxată.

Deoarece arătătorul și degetul mijlociu ale mîinii drepte (la dreapta) sînt cele mai folosite, tastele cele mai des folosite ar trebui să se afle în dreapta acestora. De exemplu "J" nu face parte dintre acestea. Tastele cele mai des folosite se află în partea stîngă a rîndului superior al tastaturii.

Încercările de a obține o tastatură cu adevărat ergonomică au fost respinse de piață, fiind adoptată doar poziția tastaturii față de mîinile omenești, neținîndu-se cont de frecvența de apariție a fiecărei taste în parte. Și pe viitor tastatura va rămîne aparatul de introducere numărul 1. Pentru a nu încălca calculatorul cu administrarea tastaturii, aceasta este dotată cu un procesor propriu. Acesta transformă semnalele electrice datorate apăsării unei taste într-un semnal serial pe care îl transmite prin cablul tastaturii calculatorului.

Calculatorul este dotat cu o interfață serială pentru tastatură;



Touchscreen: introducere prin apăsarea unor cîmpuri



Lightpen - în vârful său se află o fotodioda

aceasta este interogată de sistemul de întreruperi iar informația tastaturii se transmite spre BIOS. Momentan există două tipuri, tehnic diferite, de tastaturi. Una este pentru PC-urile 8086 iar cealaltă pentru AT-urile 80286, 80386 și 80486. Aceste tastaturi nu sînt compatibile datorită diferențelor hardware. Deci o tastatură PC nu funcționează pe orice AT și invers. Dar există și tastaturi care posedă un comutator și care pot funcționa cu ambele tipuri de calculatoare. Deoarece la calculatoarele High-End, deci 386 și 486 cu frecvența de tact ridicată, tactul de bus este mărit, se poate întîmpla ca procesorul tastaturii să nu poată alimenta destul de rapid interfața serială. Rezultatul: semne greșite sau lipsă de semne. Deci la cumpărarea unui calculator rapid trebuie găsită și tastatura adecvată.

Cine-și dorește probleme de contact?

Pretențiile relative la mecanica tastaturii sînt foarte ridicate. Ea trebuie să suporte multe milioane de tastări fără ca să apară defecțiuni mecanice.

Există 2 tipuri uzuale de contacte electrice. Mai ieftine sînt contactele cu folii, care au în componență 3 folii. Cea de jos are rolul de contact electric, cea din mijloc are rol de distanțare, iar cea de deasupra reprezintă cel de-al doilea contact electric. În locul de contact, folia

mijlocie este prevăzută cu o gaură atît de mare încît folia superioară să nu o poată atinge pe cea inferioară decît la apariția unei forțe de apăsare. Mai scumpe și mai complicate sînt contactele cu lamele metalice. În acest caz apăsarea tastei apropie două lamele metalice și realizează astfel contactul. În funcție de modul de execuție lamelele sînt suflate cu argint și/sau aur, pentru a le prelungi perioada de viață și pentru a micșora rezistența de contact. În cazul tastaturilor din industrie foarte des folosite în locul contactelor se folosesc și cristale piezoelectrice. Acestea înregistrează tensiunile apărute la cele mai reduse presiuni și le valorifică. Aceste cristale sînt dispuse sub o placă de oțel, astfel încît tastaturile Heavy-Duty pot fi construite în stil antiexploziv și antiacvatic.

Există o problemă hard și soft și anume "blocarea" tastelor. Dacă o tastă este apăsată prea tare, ea poate sări înapoi și să facă în acel moment un al doilea contact. Aproape fiecare tastatură are altă bătaie a tastelor, datorită modului diferit al execuției mecanice. Importantă este definirea unui punct de presiune: unele tastaturi scot un sunet la apăsarea fiecărei taste care poate fi deranjant. Nu există nici o tastatură perfectă, și fiecare om are propria sa imagine despre cum ar trebui să fie o tastatură bună, de aceea înainte de cumpărare este bine să scrieți, de probă, un text mai lung și în final să lăsați să decidă degetele.

Ecranul - introducere directă

Deoarece ochiul este îndreptat spre ecran este de la sine înțeles să se facă "intrări" direct pe ecran. Cele mai importante două sisteme de intrare sînt ecranul sensibil la atingere (Touchscreen) și creionul optic (Lightpen). Ambele sisteme permit utilizatorului, prin simpla apăsare pe ecran, direct sau cu ajutorul lightpen-ului, să lanseze programe. Bineînțeles că software-ul trebuie să fie adaptat acestui lucru.

De regulă aceste sisteme sînt oferite azi la echipamentele și terminalele publice (de exemplu pe aeroporturi, în birouri de turism și în bănci), deoarece acest sistem economisește o tastatură și utilizatorul nu trebuie să execute de regulă intrări complicate. Astfel se oferă unele puncte de meniu, dintre care se poate face alegerea prin apăsare directă.

Cu degetele obosite nu se poate scrie

Nici Touchscreen și nici Lightpen nu pot fi folosite în mod util pentru intrări exacte și de mai lungă durată; deoarece ecranul este așezat vertical în fața utilizatorului, mîna trebuie să fie liberă și să nu se sprijine pe nimic. Acest lucru duce la o obosire rapidă.



Mouse - un periferic de intrare indispensabil

Joystick-ul nu este gândit doar pentru jocuri

Singura soluție ar fi integrarea ecranului în masa de scris. În cazul ecranului sensibil nu se pot obține introduceri foarte exacte datorită diferitelor presiuni exercitate de degetele omenești. Aici intrarea se reduce la atingerea de câmpuri mari. Pentru a face un ecran sensibil la tastare ei va fi acoperit cu mai multe straturi de folii. Aceste folii sînt formate din mai multe contacte extrem de subțiri, care sînt legate cu conductoare tot atît de subțiri. La atingere rezistența de contact a unui anumit domeniu se modifică, și astfel pot fi calculate coordonatele X și Y.

Lightpen-ul folosește el însuși semnalul de pe ecran pentru a-și determina poziția. În vîrfurile lightpen-ului este așezată o fotodiodă, care preia semnalul luminos avînd o frecvență de 50 Hz (sau chiar mai mult, în funcție de frecvența de repetiție a imaginii). Printr-o sincronizare exactă cu apariția în timp a semnalului de pe ecran, poziția lightpen-ului poate fi determinată foarte exact. În acest fel se obține, prin alegerea unei fotodiode extrem de mici, o rezoluție comparabilă cu rezoluția ecranului. Și în viitor touchscreen-ul și lightpen-ul vor fi folosite doar rar, în aplicații speciale.

Modul de utilizare pare în primul moment foarte plăcut, dar lucrul îndelungat este obositor și inexact. De aceea mouse-ul balistic va îndepărta aceste aparate de intrare orientate pe ecran.

Mouse-ul: sens și nonsens

Mouse-ul, (v. *if* 3/90), îi împarte pe utilizatori în două tabere. Mulți utilizatori care cunosc PC-urile încă de la începuturi cunosc anumite programe, cum ar fi Wordstar, încă de la primele versiuni. Au învățat pe dinafară combinațiile de taste și pot lucra fără să mai ia mîinile de pe tastatură. Și astăzi multe editoare de texte încă mai folosesc convențiile Wordstar. Ele au fost într-o anumită măsură un standard. Mouse-ul ar trebui să ușureze munca, nemaitrebuind memorate combinații lungi de taste. Dacă a reușit să fie prietenos, rămîne de văzut. Mouse-ul are sens doar dacă mîna nu trebuie să se miște mereu între mouse și tastatură.

Dacă dorim o salvare intermediară rapidă în timpul unei prelucrări de texte, în Word 5.0 trebuie să acționăm combinația de taste (Ctrl - F10). În timpul în care se apasă aceste taste un alt utilizator apasă o singură dată o tastă a mouse-ului. Pe de altă parte este de la sine înțeles că un software specializat pe utilizare prin mouse este foarte prietenos. Nu degeaba suprafața utilizator Windows 3.0, a lui Microsoft, are un asemenea succes.

Mouse-ul este aparatul de intrare ideal în cazul prelucrării datelor grafice, deci CAD sau DTP. Între timp au apărut cele mai diverse

forme de mouse. Baza se pune pe modul în care mouse-ul se adaptează mîinii. Dar producătorii uită de multe ori un lucru: există un număr destul de mare de stîngaci, ale căror nevoi ar trebui luate în seamă.

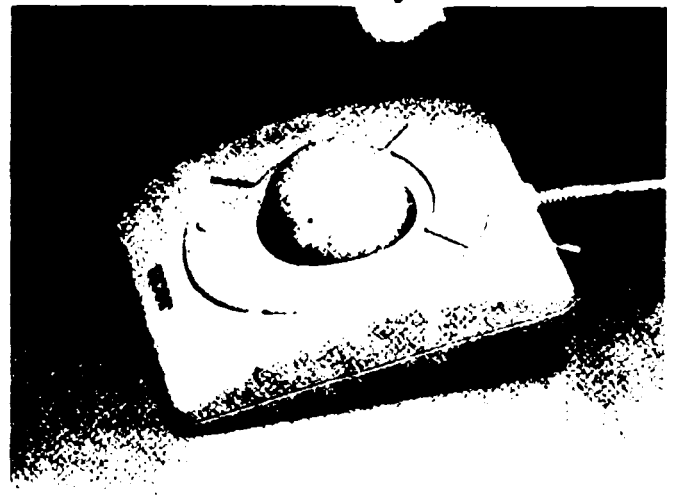
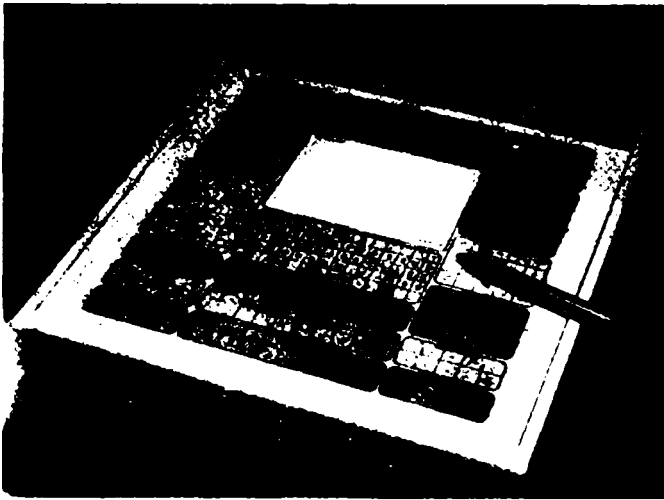
Casa șoricelului

Tehnica unui mouse este simplă. O sferă de cauciuc aflată pe partea inferioară a mouse-ului antrenează niște roțițe în direcțiile X și Y. Mai există și mouse-uri la care aceste două roțițe sînt antrenate direct, deci fără sfera de cauciuc. Pe roțițe sînt fixate cîte o plăcuță găurită. O unitate formată dintr-o fotocelulă și un LED transformă rotația în impulsuri electrice.

Transmiterea impulsurilor se face de regulă printr-un cablu care este conectat la interfața serială a calculatorului. Alți producători de mouse economisesc cablu și trimit impulsurile prin lumină infraroșie sau prin unde de calculatorului. Cablul nu mai incomodează, în acest caz, mișcarea mouse-ului, în schimb mouse-ul devine mai greu datorită electronicii mai complicate și a propriei alimentări cu curent prin baterie.

Mai există și așa-numitul busmouse. Acesta nu-și trimite datele prin interfața serială, ci direct în bus-ul de date al calculatorului.

Primele mouse-uri au avut o rezoluție slabă și o adresare liniară.



Tableta grafică - un periferic de intrare foarte precis

Trackball - un mouse răsturnat

Mouse-urile moderne au o rezoluție de pînă la 1200 dpi (Dots per Inch) și se adresează balistic. Aceasta înseamnă că mișcînd ușor mouse-ul cursorul poate fi plasat pe ecran exact. La o deplasare rapidă a mouse-ului cursorul se va deplasa mult pe ecran, se economisește timp și se folosește o suprafață mică pe mousepad (suprafața, pe care se mișcă mouse-ul).

Deoarece tot mai multe soft-uri se orientează pe operarea cu mouse, acestea devin din ce în ce mai utilizate. Rămîne de sperat, ca deservirea prin tastatură, care de multe ori este mai rapidă și mai confortabilă, să nu aibă de suferit din acest motiv.

Joystick-ul, nu doar pentru jucători

Joystick-ul, (v. "if" 5/91), a avut mare succes atunci cînd piața HC-urilor s-a extins foarte mult. El se pretează foarte bine la introducerea valorilor de comandă uni și bi-dimensionale. Multe jocuri au nevoie de un joystick. La îndrăgirea lui a condus și prețul. Un joystick simplu poate fi găsit la mai puțin de 30 DM.

Dar joystick-ul își are locul nu doar în domeniul jocurilor, ci și în comanda proceselor și la deservirea mașinilor moderne. Acolo unde pe timpuri o macara era deservită cu niște manete mari de comandă,

astăzi este folosit joystick-ul, deoarece comanda se face acum electronic.

Există două tipuri diferite de joystick-uri. Cel mai simplu, folosit în domeniul jocurilor, este "joystick-ul digital". Maneta joystick-ului este încastrată articulat și are pe fiecare din cele 4 părți din interiorul carcasei cîte un întrerupător cu contacte care se închid și se deschid în funcție de mișcarea manetei. Contactele pot fi realizate cu lamele metalice ieftine sau cu microîntrerupătoare scumpe. Bineînțeles că nici unui joystick nu-i lipsește un microîntrerupător suplimentar, denumit și tastă declanșatoare (trăgaci).

Cel de-al doilea tip constructiv este "joystick-ul analog". Contactele simple sînt înlocuite cu două potențiometre dispuse perpendicular. Astfel se pot introduce semnale proporționale. Aceste joystick-uri se folosesc la comanda de la distanță a modelelor. Semnalele de la joystick sînt transmise calculatorului prin intermediul unui gameport, deci a unei interfețe pentru jocuri. Deoarece jocurile pe calculator sînt foarte îndrăgite, joystick-urile vor rămîne și în continuare cele mai folosite medii de introducere.

Digitizorul: o concurență pentru mouse

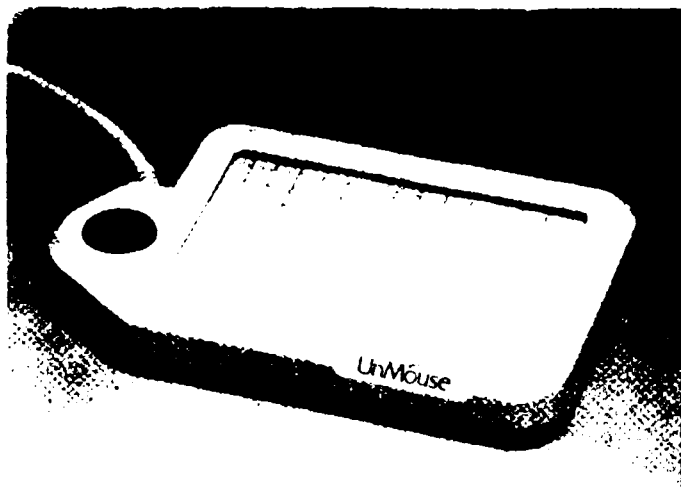
Nu numai datorită faptului că ele ajung să-și merite tot mai mult

prețul, digitizoarele, (v. "if" 3/91), își fac tot mai mult apariția pe piață. Pentru aplicații CAD aceste aparate de intrare sînt ca și predestinate. O tabletă grafică constă dintr-o suprafață plană din material sintetic sub care se află un tastator electronic și dintr-un creion special pentru introducerea și comanda punctelor din meniu.

Fiecare poziție a aparatului de intrare de pe tableta grafică se determină, în cele mai multe cazuri, inductiv sau capacitiv. Creionul emite unde electromagnetice, care sînt recepționate de tabletă. Un convertor A/D digitalizează această informație și o transmite mai departe calculatorului.

Unii producători oferă un protocol de transfer compatibil cu mouse-ul Microsoft, care dă posibilitatea utilizării lor și în cazul unor aplicații care recunosc doar mouse. Deoarece cele mai multe tablete lucrează prin metoda inductivă, ele produc un ușor cîmp magnetic. Dischetele care se ating cu acestea pot fi deci afectate. Alte tablete grafice lucrează cu ultrasunete. Creionul emite ultrasunete care sînt înregistrate de două microfoane. Pe baza diferenței de timp poate fi determinată poziția exactă a creionului.

Pentru o mai simplă utilizare a tabletelor grafice se livrează odată cu ele suporturi adaptate pentru aplicații speciale, cum ar fi Auto-CAD. Astfel se pune la dispoziția



Touchmouse - pentru introducere sînt necesare obiecte ascuțite

Scanner-ul - poate prelua atît texte cît și imagini

utilizatorului un mod de comandă care nu funcționează prin meniuri ecran ci prin atingerea unor simboluri de pe tabletă. Se pune întrebarea dacă acest mod de comandă are sens deoarece continua mutare a privirii de pe ecran pe tabletă și înapoi devine în timp foarte oboșitoare. Pentru posesorii de PC-uri Notebook sau a altor calculatoare portabile și pentru cei care nu au loc pentru un mouse sau o tabletă grafică pe biroul lor, se oferă alte două alternative de aparate de intrare: în primul rînd este vorba de Trackball, în principiu un mouse culcat pe spate și în al doilea rînd de așa-numitul Touchmouse, o tabletă grafică miniaturizată.

Pentru economisirea spațiului: Touchmouse și Trackball

Astfel utilizatorului i se dă posibilitatea deservirii unor programe, care necesită o conducere prin mouse, și fără suprafața plană.

Trackball constă dintr-o sferă înglobată într-o carcasă deschisă în partea superioară. La fel ca și la mouse mișcările sferei în direcțiile x și y sînt preluate de două roțițe. Se produce un semnal serial înregistrat cu ajutorul unei fante luminoase și a unei roțițe găurite, care va fi pus la dispoziția programului utilizator. Deservirea unui trackball este mai greoaie decît a unui mouse. Mișcările mîinii nu pot fi

traduse direct, ceea ce înseamnă că mîna trebuie să corecteze mereu, și deci vor exista erori de poziționare. De aceea introducerea datelor cu un trackball se folosește doar în aplicații speciale, cum ar fi comanda mașinilor și a sistemelor CAD speciale. Relativ nou pe piață este touchmouse-ul, o tabletă mică, și utilă, unde intrarea se face cu unghia sau cu alte obiecte ascuțite. Contactele integrate permit determinarea poziției locului apăsător, și software-ul de comandă calculează cu aceste date poziția mouse-ului pe ecran. Touchmouse-ul nu se pretează pentru o poziționare exactă, este însă o alternativă bună pentru cei care vor să utilizeze Windows 3.0 pe un laptop și care nu doresc să ducă cu ei și un mouse. Bineînțeles că trackball și touchmouse nu sînt singurele aparate de intrare care fac concurență mouse-ului. Unii producători au proiectat între timp o mulțime de înlocuitori. În schimb nu prea este de așteptat ca utilizatorii să le accepte cu ușurință, deoarece, de regulă, omul este supus inerției și este refractar la nou.

Convertorul A/D: o punte spre electronică

Fără îndoială că cea mai bună metodă de comunicație ar fi "calea directă" de la lumea exterioară către calculator. Dacă calculatorul ar înțelege vocea, ar putea prelua semnale de măsură și ar putea per-

cepe prin intermediul unei camere de luat vederi mediul înconjurător, multă muncă de introducere s-ar putea economisi. Aceste idei nu mai sînt de domeniul SF, căci cercetările în această direcție sînt foarte avansate, iar în cazuri speciale există deja soluții acceptabile. Un senzor oarecare, fie el microfon, cameră de luat vederi sau indicator de temperatură, înregistrează un semnal. Dar toate semnalele de origine naturală sînt analogice, valorile lor nu se modifică în trepte ci în mod continuu. Un sunet nu este deci puternic sau încet, ci el poate avea o infinitate de nuanțe, și în acest caz calculatorul are probleme. El înțelege de fapt doar valorile digitale, deci "cu curent" și "fără curent". Dacă un calculator urmează să prelucreze mărimi analogice, acest lucru presupune o digitalizare prealabilă. Pentru aceasta se folosește așa-numitul convertor A/D; acesta împarte un anumit domeniu de tensiune într-un număr de intervale. Fiecărui interval i se atribuie apoi în calculator o anumită valoare. Cu cît este mai mare numărul de intervale cu atît pot fi convertite mai precis valorile semnalului de intrare. Cele mai des întîlnite procedee de transformare a semnalelor analogice în semnale digitale sînt:

- procedeele integrării duble (Dual-Slope)
- procedeele aproximării succesive și

- procedeul conversiei fulger (Flash)

Principiul conversiei Dual-Slope constă în măsurarea timpului necesar unui condensator pentru a se încărca la tensiunea pe care trebuie s-o convertească. Cu cât tensiunea de intrare este mai mare cu atât mai lung este timpul de încărcare necesar. Acest procedeu este relativ lent și se pretează doar pentru măsurarea unei tensiuni cu un voltmetru digital, dar nu și pentru digitalizarea vocii sau a imaginii. În acest caz procedeul aproximării succesive este mult mai rapid; în acest caz, în principiu, se activează o tensiune de verificare în cadrul convertorului A/D, care se adaptează tensiunii efective și apoi are loc conversia. Această conversie se pretează deja pentru digitalizarea vocii.

Multimedia: un cuvânt magic

Semnalele video pot fi digitalizate prin procedeul flash. În acest caz tensiunea de intrare este aplicată în același timp la mai multe comparatoare de tensiune, al căror număr depinde de lățimea eșantionului. Viteza de conversie a acestui sistem este de circa 20 de nanosecunde. Prin comutări electronice rapide și exacte s-a ajuns, din punct de vedere tehnic, la proceduri de înregistrare digitală a sunetului pentru discuri CD și pentru discuri optice. Și cuvântul magic "Multimedia" este în ultimul timp foarte des folosit. Cu multimedia se pot prelucra în calculator în timp real imagini video, acest lucru fiind imposibil fără un convertor A/D rapid. Și în tehnica măsurărilor convertoarele A/D ocupă un loc important. Acolo unde înainte trebuiau introduse manual lungi șiruri de măsurători pentru a le valorifica prin calculator, astăzi se leagă un senzor la calculator prin intermediul unui convertor A/D, iar calculatorul preia atât colectarea cât și prelucrarea datelor.

Tehnica de producție și de automatizare nu mai pot fi imaginate fără astfel de sisteme. Căci tehnici-

le flexibile de automatizare se pot realiza doar prin controlul exact al desfășurării evenimentelor. Chiar și digitalizarea imaginii și a vocii se pot face doar printr-un convertor A/D. Cu toate că din punct de vedere tehnic s-a ajuns ca aceste informații să poată fi digitalizate prelucrarea lor, cum ar fi: recunoașterea vocii sau recunoașterea vizuală a unui obiect cu ajutorul calculatorului sînt încă în față. Deci va trebui să mai treacă încă puțin timp pînă cînd vom putea dicta un text calculatorului.

PC-ul ca cititor pasionat

Punerea problemei este simplă: un text scris pe hîrtie trebuie "introdus" într-un calculator, pentru a fi prelucrat în continuare. Există două soluții posibile. Prima ar fi copierea textului, dar această soluție necesită mult timp. A doua ar fi citirea unui astfel de text și aducerea lui cu un software adecvat în format text. Acest lucru îl face scanner-ul, el poate "introduce" atât imagini cât și grafice în calculator. Aceste aparate de citire descompun o imagine sau un text în puncte singulare digitale și trimit această informație PC-ului. Există mai multe metode de digitalizare a unei imagini. Cele mai importante sînt: procedeele Multiplier, Tehnic, Laser și CCD. Cel mai utilizat procedeu este CCD cu matrice linie și matrice bidimensională. Suprafața de scanat se vizualizează pe întreaga lățime a scanner-ului cu un tub luminiscent. Lumina reflectată va fi preluată apoi de o serie de fotodiode și diode laser. Ele înregistrează diferențele luminoase - întunecos și le convertesc într-o combinație binară. În funcție de densitatea elementelor CCD rezoluția scannerului este dinainte fixată. Scannerele normale pun la dispoziție o rezoluție de 300 dpi (Dots per Inch). La fel cum există mai multe procedee există și mai multe tipuri de scannere, cele de format mare fiind folosite mai ales în domeniul profesional iar cele de mîină în cel particular. În comparație cu tehnica avansată a scanner-elor recunoașterea scrisului

mai poate fi încă îmbunătățită. O imagine scanată este memorată în prima fază în format grafic de calculator. Pentru prelucrarea ulterioară într-un editor de texte ea trebuie transformată mai întîi într-un format text. Acest lucru se face cu un software OCR (Optical Character Recognition) care recunoaște optic semnele, (v. "if" 2/90). Tocmai în cazul recunoașterii scrisului se vede diferența dintre om și calculator. Un calculator, cu programele sale cu tot, nu poate să facă asociații. Astfel programele OCR simple nu pot recunoaște decît litere foarte clare. Literele care prezintă greșeli datorită unor spoturi calitativ inferioare îi dau multă bătaie de cap programului de recunoaștere. Dar și texte scante foarte curate necesită programe bune pentru a executa transformări neeronate. O metodă rapidă și sigură este așa-numita "Pattern Matching". Pentru fiecare literă în parte se memorează unul sau mai multe "modele normale". Un nou semn va fi apoi suprapus peste model și comparat prin înlănțuire logică. În funcție de cum au fost date toleranțele de comparație, se poate întîmpla, ca o literă de fapt clară să nu fie recunoscută de program sau un C să devină un E.

Tehnica prin suprapunere s-a dovedit foarte bună în cazul documentelor tipărite și scanate foarte bine, avînd o viteză mare de prelucrare și erori puține. Pentru documentele scrise de mîină sau cursiv procedeul Feature-Recognition este mai bun. Acesta se orientează după caracteristicile unei litere. Caracteristica tipică C este că nu conține nici o linie dreaptă și că spre dreapta este deschis. Algoritmul de urmărire a conturului conține caracteristicile tipice ale literei pe care le valorifică. Din păcate acest procedeu este foarte lent și nu merită folosit la textele tipărite.

(I.M.)

Touch me, user!

Noi și foarte variate alternative la tastaturile clasice au stîrnit în ultima vreme interesul utilizatorilor de PC-uri. Ceea ce în industrie a devenit deja ceva normal, a ajuns să fie oferit și utilizatorilor particulari din domeniul PC-urilor. Cuvîntul magic al viitorului este: atingere.

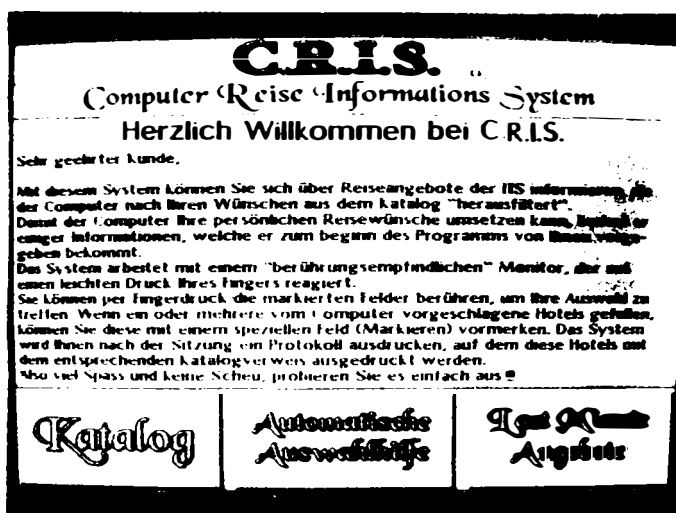
Timpurile în care oamenii le era frică să atingă calculatorul vor aparține, în curînd, definitiv trecutului. Mediul de introducere clasic, tastatura, încă mai domină clar - așa-numiții "Touches" înaintează. Ei nu vor mai putea fi ignorați, fie că sînt ecrane sensibile la contact (Touchscreen) fie că sînt cîmpuri de introducere date asemănătoare tastaturii (Touchpads). Avantajele sînt vizibile. Tehnica atingerii (Touch-Technik) oferă o deservire simplă cît și un schimb rapid de informații între om și calculator.

Atingerile nu sînt omise

"Touchpanels" este titulatura generală pentru acest mediu de introducere și este denumirea pentru o suprafață (din material sintetic, sticlă, folie) care prin atingere cu mîna, degetul sau un creion poate fi folosită pentru comandarea funcțiilor programului. Denumirea de Touchpanel vine din limba engleză și s-ar traduce prin cîmp de atingere sau cîmp de senzori (touch = a atinge, panel = cîmp, tablă de instrumente, tablă de comutație).

Dintre domeniile în care s-a folosit pînă acum putem aminti medicina și în primul rînd industria. În tot mai mare măsură această tehnică este folosită atît pentru schimbul general cît și pentru cel special de informații. Pe aeroporturi călătorii sînt informați prin Touchscreen despre rutele de zbor timpii de decolare și de aterizare și despre alte lucruri specifice, prin simpla atingere a unui cîmp din meniul de pe ecran, care dă posibilitatea prin subme-

În curînd în fiecare birou de turism: Informații via Touchscreen



niuri la o apropiere tot mai mare de informația dorită.

Nici birourile de turism nu-și vor mai informa clienții în viitorul apropiat cu ajutorul cataloagelor color despre ofertele de concediu, ci tot prin "Touchscreen" vor oferi meniuri pentru concedii, care vor cuprinde imagini de hoteluri, ștranduri și camere de hotel. Pe lîngă domeniul turistic vor mai fi dotate cu tehnica de atingere multe suprafețe utilizator din domeniul PC-urilor. Dar nu doar ecranele pot fi echipate cu această tehnică ci și tastaturile pot fi echipate cu senzori sensibili la atingere, caz în care tastele se înlocuiesc cu senzori, care sesizează o atingere foarte ușoară cu degetul.

Un semn cu degetul ajunge

Datorită tehnicii de touchscreen utilizatorul poate interveni pe instrumentul de introducere date. Monitorul îi deschide o nouă dimensiune a "introducerii interactive". Dar cum ajunge ecranul mat al monitorului să devină mediu de introducere?

Particularitățile suprafeței monitorului nu rezolvă problema. Componentele care răspund de aceasta sînt un ecran, care se află pe aceasta, sau deja implementat în monitor și un controller. Ecranul

sensibil la atingere (sticlă, material sintetic, folie) este prevăzut cu un strat transparent și bun conducător de electricitate introdus prin difuzie (un praf metalic foarte fin), iar pe conturul ecranului sînt prevăzuți electrozi care sînt montați între ecran (tub cinescopic, sau tip plasma) și rama ecranului.

Această sticlă stratificată și transparentă, cu conductori electrici, are electrozii amplasați liniar și alimentați cu o tensiune alternativă. Între acești electrozi se află suprafața activă, sensibilă la atingere.

Prin atingerea suprafeței active cu degetul ia naștere un curent de conducție care se scurge spre pămînt prin intermediul corpului uman. Această tehnică este denumită "analog-capacitivă". Curentul are valori de doar cîțiva nA.

Rezoluția senzorilor este de 1024 de puncte de atingere per axă, în interiorul cîmpului calibrat de pe ecran. Construcția unei astfel de suprafețe monitor, sensibilă la atingere, nu mai este o problemă. Există astfel de touchscreen pentru toate tipurile de monitoare, cu tuburi cinescopice, cu plasmă electroluminiscentă sau cu cristale lichide, cu suprafață bombată sau plată.

Inima propriu-zisă este așa-numitul "touchcontroller", care este montat fie direct în carcasa moni-

torului, fie poate fi introdus ca o cartelă de extensie (de 8 biți) în calculator. Controller-ul înregistrează atingerea ecranului, calculează poziția în care acesta a fost atins și transmite rezultatul, sub forma unor date seriale, calculatorului pentru a le prelucra mai departe. În cazul în care controller-ul este montat direct în carcasa monitorului comunicația cu calculatorul se realizează prin intermediul unei interfețe seriale RS232.

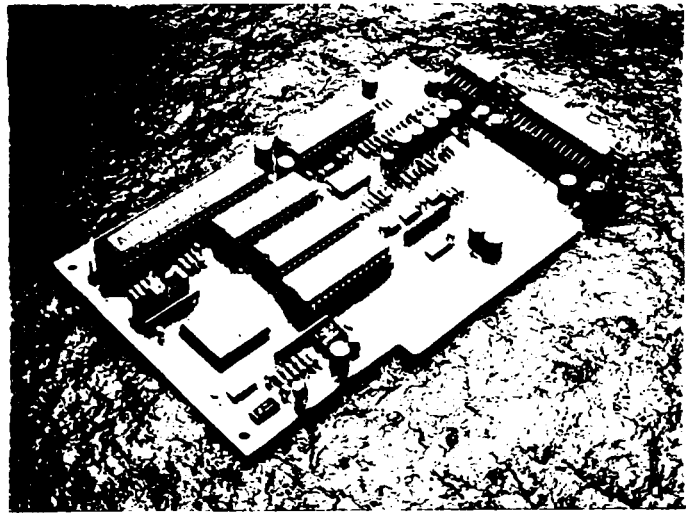
Touchcontroller-ul constă din două procesoare, procesorul analog și procesorul de comunicație. Partea analogă a touchcontroller-ului este constituită din nivelul de intrare al senzorilor. Cei patru senzori sînt măsurați continuu ceea ce duce la generarea unui curent între ecran și pămînt, la atingerea ecranului. Din curenții de conducție rezultați pentru fiecare electrod sînt calculate căderile de tensiune între electrod și punctul de atingere, tensiuni care sînt preluate în continuare. Pentru ca senzalul să fie cît mai puternic se face o filtrare a frecvențelor iar conductorii sînt ecranati împotriva încărcărilor electrostatice.

Procesorul senzorilor de atingere constă dintr-un convertor analog/digital, un interpretor de comenzi și o interfață spre procesorul de comunicație. Semnalele de măsură analogice sînt digitalizate și sînt transmise mai departe ca și coordonate x, y. Acestora le corespund coordonatele punctului de atingere, motiv pentru care întreaga suprafață activă trebuie calibrată. Aceste date calibrate sînt memorate într-un EPROM și sînt utilizate la calcularea coordonatelor.

Unitatea de comunicație controlează transmisia datelor prin interfață (fie interfața serială RS232 pe un bus de 8 biți, fie interfața externă V24). Ea trebuie să supravegheze în permanență întregul controller și să stabilească definiția mai multor cîmpuri de atingere.

Senzorii de atingere sînt plasați sub o suprafață care îi protejează

Inima: fără Touchcontroller nu funcționează nimic



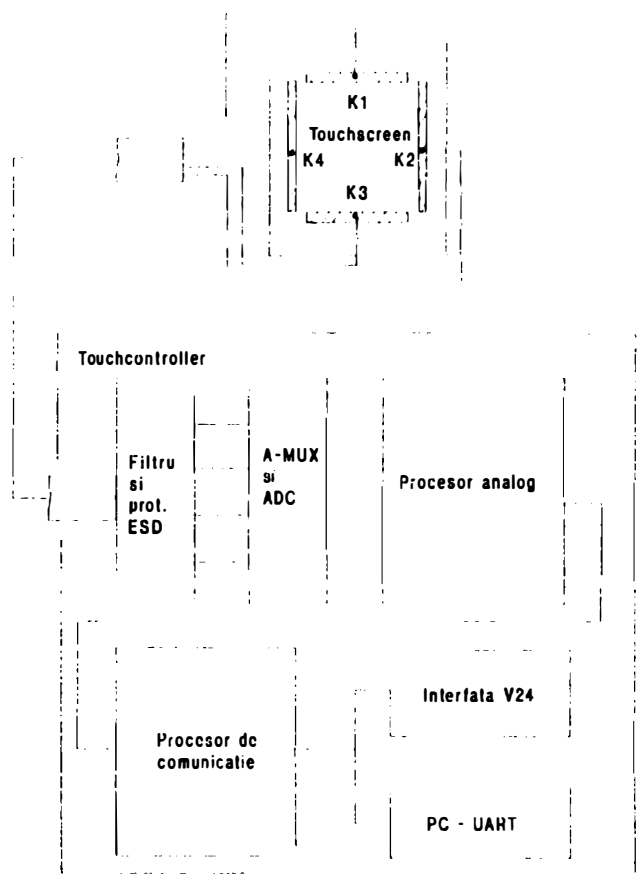
la zgîrieturi și care nu permite pătrunderea prafului sau a umezelii. Ca aparat de intrare în domeniile în care utilizatorii nu sînt specialiști în calculatoare, un "deget arătător" este un mijloc de comunicație indispensabil. Prin intermediul lui, practic oricine poate comunica cu calculatorul. Pentru ecranele sensibile la atingere au apărut între timp drivere speciale. Pentru utilizatorii din mediul MS-DOS (de exemplu pentru Windows 3.0) există deja drivere speciale cu

ajutorul cărora utilizatorul își poate defini comenzile de intrare.

Tot ceea ce este posibil cu aceste ecrane sensibile la atingere va dovedi viitorul. Cu siguranță utilizatorul va trebui să devină mai "sensibil", pînă într-o zi în care calculatorul îi va spune: "Touch me, user" - dar te rog nu prea tare!

(R.M.)

Reprezentarea schematică a unui Touchcontroller și a unui Touchscreen. Touchcontroller-ul constă din două procesoare: procesorul analog și procesorul de comunicație.

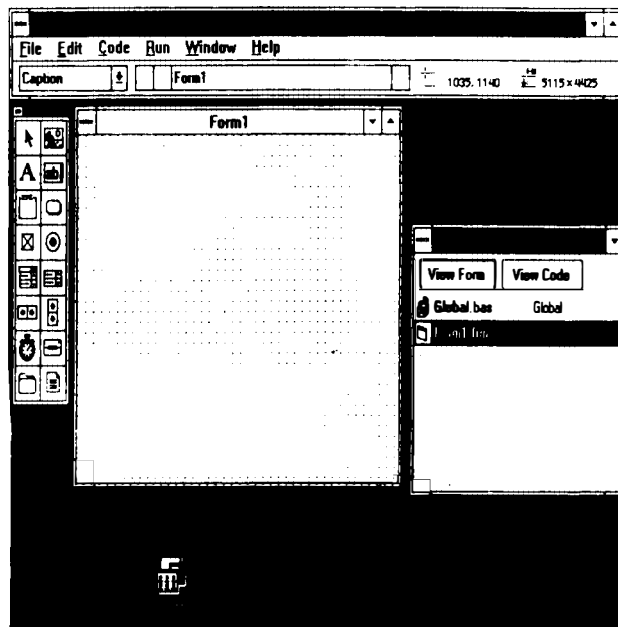


Visual Basic - programare intuitivă

Pînă acum, cine dorea să programeze în Windows 3.0 trebuia să parcurgă cele 600 de funcții ale Windows API (Application Program Interface). Ele fac legătura între o aplicație și mediul Windows propriu-zis și ușurează mult munca programatorului, dar acesta trebuie să pătrundă adînc în "secretele" sistemului. Înainte de a putea savura modul de lucru cu ferestre, programatorul trebuia să învețe modurile de apel și sintaxa acestora, ceea ce necesita un consum, de timp, efort pe care doar puțini programatori sînt dispuși să-l depună.

Prin lansarea Visual Basic-ului, Microsoft speră ca aceste dificultăți să aparțină definitiv trecutului, proiectarea unor aplicații Windows fiind mult ușurată. Deși nu vor putea fi realizate aplicații de genul Excel, de exemplu, totuși, întreaga concepție a noului mediu de dezvoltare pledează pentru depășirea barierelor ridicate de încurcatele funcții API. Comenzile simple și o structură flexibilă ușurează mult învățarea limbajului, astfel încît pînă și programatorii în C vechi se pot ușor acomoda.

Pentru a ușura cît mai mult intrarea în programarea Windows, la lansare, Visual Basic se prezintă cu un ecran ce amintește mai degrabă de un program de desenare decît de un mediu de programare. Se remarcă existența unei ferestre mari deasupra căreia scrie "Form 1". Aceasta reprezintă un formular ce corespunde unei ferestre din programul ce urmează a fi scris în Visual Basic. Existența în Visual Basic a unei suprafețe utilizator foarte comode și simple face ca verbul "a programa" să devină impropriu. Totul amintește mai degrabă de primii pași făcuți pentru pictarea unui tablou. Lîngă suprafața de lu-



cru se găsește "trusa de scule" (Toolbox) ce cuprinde toate elementele grafice cunoscute din Windows: butoane, cutii de control, cîmpuri de text, liste de fișiere, etc. Unele sînt, în parte, mici programe specializate. De exemplu, rubrica de listă de fișiere ne furnizează toate fișierele arhivate pe disc și lucrează așa cum sîntem deja obișnuiți din Windows. Trusa de scule este de altfel pasibilă de dezvoltări și diferenți producători au anunțat deja unelte ajutătoare (Add-On).

Proiectarea intuitivă a suprafeței utilizator

"Pictarea" suprafeței utilizator se realizează în cîteva secunde: se alege cu mouse-ul unealta dorită, se poziționează mouse-ul pe locul dorit din formular și se introduce, tot cu ajutorul mouse-ului, elementul ales. Poziția și mărimea elementelor poate fi oricînd modificată. Suprafața utilizator astfel realizată poate fi oricînd prezentată unui alt utilizator fără a scrie măcar o singură linie de program.

Modificările se pot realiza doar cu cîteva mișcări ale mouse-ului.

În Visual Basic elementele au denumirea de controale. Utilizatorul are libertate deplină în alegerea și aranjarea obiectelor, astfel încît același tip de control poate fi pus în același formular de mai multe ori. De exemplu, în aceeași fereastră pot exista mai multe rubrici de dialog sau butoane. Fiecare element va fi tratat ca un obiect independent cu caracteristici proprii (mărime, lungime, culoare). Aceste caracteristici vor fi denumite "proprietăți". Proprietățile unui anumit element pot fi vizualizate sau definite cu ajutorul cutiei de proprietăți, în afară de proprietăți, fiecare obiect este definit printr-un nume pentru a putea fi apelat în mod univoc. Acest fapt este esențial avînd în vedere că pe parcursul programului elementele definite (controalele) pot fi folosite ca și simple variabile. E indicat ca elementele definite să fie botezate cu nume ce amintește de funcția obiectelor (ex: "Pornirea ceasului"). În caz contrar Visual Basic-ul

denumește el însuși elementele cu nume tipice care, în cazul apariției multiple a aceluiași tip de control, vor fi numerotate crescător (ex: Button 1, Button 2, etc.). Pentru apelarea unui control din program este suficient să se dea numele controlului urmat de un punct, drept separator, și urmat de o proprietate. În acest fel, în caz de nevoie, se poate modifica o proprietate a unui control (ex: mărimea sau culoarea).

Sînt necesare cunoștințe de programe

Alfa și omega programării Windows o constituie tratarea așa-numitelor evenimente (events). Programele clasice funcționează după o schemă clar definită. Nu se poate spune același lucru despre programele Windows. Același program Windows poate fi folosit în mod diferit de utilizator, în funcție de diferitele aplicații. Înseamnă că programul trebuie să reacționeze flexibil la acțiunea utilizatorului. Nucleul Windows sesizează în timpul rulării unui program dacă survine un anumit eveniment. Un astfel de eveniment îl poate constitui selectarea cu un clic pe mouse a unui buton de comandă sau acționarea unei taste de către utilizator. Dacă survine un astfel de eveniment, el sensibilizează o rutină specifică evenimentului.

Visual Basic este altfel conceput. După ce s-a creat suprafața utilizator, va fi dat, pentru fiecare obiect, programul ce urmează a fi rulat la alegerea obiectului. La un dublu clic pe element se deschide o fereastră în care Visual Basic inserează de la început două instrucțiuni: începutul și sfîrșitul subrutinei.

(continuare în pag. 38)

Ce este bioinformatica?

O abordare biologică a dezvoltării de programe

Dezvoltarea de programe cu ajutorul metodelor biologiei, iată domeniul bioinformaticii. În ultimii ani au fost dezvoltate mai multe astfel de tehnici și tehnologii, unele dintre ele, deja demonstrate, au ieșit pe piață, altele se mai perfecționează încă în laboratoare. Toate oferă o mare promisiune programatorilor: de a putea depăși barierele actuale de complexitate și dimensiune a programelor și de a obține soluții simple pentru o mulțime de probleme spinoase, ca, de exemplu, recunoașterea formelor. Cel mai cunoscut succes al acestor tehnici sînt rețelele neuronale cunoscute sub mai multe denumiri, ca procesare distribuită paralelă sau conexiunism, sau neuroinformatică. Alte tehnologii ale bioinformaticii sînt algoritmi genetici, sistemele "L", logica fuzzy, sistemele fractale, automatele celulare, dinamicile haotice. Încercăm, credem, pentru prima dată să considerăm aceste tehnici împreună. Pe măsură ce învățăm mai multe despre ele, legăturile ne apar din ce în ce mai clare.

În unele cazuri, tehnicile sînt legate la nivel teoretic, de exemplu dinamicile haotice, fractalii, sistemele "L" și sistemele funcții iterate, au același suport matematic. În alte cazuri, tehnicile sînt distincte, dar păstrează totuși legătura cu lumea naturală, biologică.

Nevoia de bioinformatică

În anii '80 sistemele de rețele în timp real, GUI-urile, (suprafețele utilizator grafice), și aplicațiile distribuite au dominat piața, cu o complexitate și mărime a programelor extrem de variate. Mulți dintre noi am întîlnit în acești ani limitele metodelor actuale de dezvoltare a programelor.

De exemplu, cele zece milioane de linii de program scrise pentru naveta spațială au ajuns la un blocaj catastrofal în timpul numărătorii

inverse de la lansare, datorită lipsei unei virgule. În anul trecut, întreaga rețea de telefoane AT&T a fost imobilizată o zi întreagă datorită unei condiții anormale întîlnită de către software-ul multiprocesorului.

Sau, mai aproape de noi, multe dintre firmele producătoare de software pentru PC-uri au fost stînjinite de programe care rulau prea încet sau care aveau erori (de exemplu dBASE IV 1.0, LOTUS 1-2-3/G, OS/2, Windows, Macintosh System 7, etc.). Cu cît programele sînt mai mari, cu atît mai tare se blochează și cu atît mai mult timp se pierde cu aceste blocaje.

Deci, nu pare surprinzător că specialiștii în informatică au căutat metode de construcție a unor sisteme software mai puțin fragile, mai sigure, mai flexibile care să-și conserve nivelul înalt de funcționalitate. Au fost explorate multe direcții, de la programarea orientată pe obiecte la metodele inteligenței artificiale. Una dintre cele mai interesante abordări este bioinformatica.

Natura știe cel mai bine

Să nu fim înțeleși greșit, scopul bioinformaticii nu este să construiască programe mari. Cele mai multe implementări actuale sînt mici în dimensiuni, și consumă puține resurse. Dar cu cît numărul de linii ale unui program convențional se mărește, cu atît complexitatea lui se apropie de un bioorganism mic. Sistemele vii au structurile interioare construite din componente simple (celule) cu un grad mare de redundanță, toleranțe la greșeli și foarte adaptabile. Biostructuri de o incredibilă complexitate sînt construite cu ajutorul unei mulțimi foarte reduse de reguli. Specificațiile pentru o ființă umană de șase picioare înălțime (molecula genetică ADN) încap într-un container mai mic decît a milioane parte dintr-un club cu la-

tura de un inch, și totuși ființele umane sînt mult mai complexe decît un Boeing 747. Mai mult, spre deosebire de naveta spațială noi nu avem blocaje cînd o virgulă lipsește din secvența ADN (de obicei).

Hans Moravec, un teoretician de calculatoare și cercetător în robotică, a încercat să compare puterea de calcul a organismelor naturale, cu CPU-urile artificiale, după părerea lui un MIPS (milion de instrucțiuni pe secundă) este aproximativ echivalent cu 100.000 celule neuronale din creier. Altfel spus un Macintosh este echivalent cu un melc, iar un Cray2 este la nivelul unui rozător mic.

Putem extinde comparația hardware și în domeniul software-ului. Lanțul ADN al unei ființe umane, este aproximativ de șase miliarde biți lungime, și reprezintă specificațiile de construcție ale unui om. Pare a fi foarte multă informație, dar de fapt este surprinzător de puțină dacă o comparăm cu produsele software contemporane. ADN-ul este o secvență de cod și date doar de o mie de ori mai mare decît informația conținută în produse de tipul dBASE IV sau Lotus 1-2-3. Cu siguranță că programatorii naturii nu folosesc limbajul C și nici asamblorul!

De fapt informația utilă într-un ADN este mult mai mică pentru că doar 5% din ADN este folosit ca date de către corpul uman. Aceasta înseamnă aproximativ 40 Mbytes, mărimea unui hard disc de PC - AT. Celelalte procente, ADN-ul vechi, conțin informații neutilizate de sistemul țintă. O parte din aceste date pot fi văzute ca biți de stop sau de paritate, folosiți la corecția erorilor și protejarea semnalului genetic. Altă parte reprezintă informații depășite, valabile în trecut pe scara evoluției ființei umane, cu alte cuvinte o informație demodată. Cu siguranță, programatorii naturii știu să folosească ifdef-urile pentru

a comenta codul de care nu mai este nevoie.

Rețelele neuronale artificiale sînt bazate pe modul în care teoriile actuale descriu funcționarea creierului natural, adică prin interconectări de celule neuronale. În realitate funcționarea creierului mamiferelor continuă să rămînă necunoscută biologilor, dar acest fapt nu a stagnat dezvoltarea modelului de rețele neuronale.

Ariile de aplicabilitate ale rețelelor neuronale sînt surprinzător de extinse, în jurul principalului obiectiv de recunoaștere a formelor. Unul dintre primele modele de rețele neuronale, elementul linear adaptiv (ADALINE) al lui Bernard Widrow a fost folosit încă din 1959 ca filtru adaptiv hardware pentru eliminarea ecourilor pe liniile telefonice. În ultimii ani, rețelele neuronale artificiale au fost folosite pentru recunoașterea formelor, procesarea imaginilor, analiza limbajului natural, filtrarea zgomotelor, controlul roboților și modelarea financiară. Un avantaj cheie al rețelelor neuronale este modul în care acestea imită puterea creierului de a lua decizii și de a trage concluzii cînd este confruntat cu informații complexe, cu zgomote, irelevante și parțiale. Abilitatea rețelelor neuronale de a înțelege datele măsurabile ale lumii reale a devenit critică atunci cînd s-a încercat lucrul cu echipamente ținute în mîină cum ar fi prelucrarea scrisului de mîină sau a vocii.

Un alt avantaj este că rețelele neuronale nu necesită programe puternice ci mai degrabă sînt rezultatul unor antrenamente asupra unui set de date, rețeaua învățînd să scoată la ieșire rezultatele dorite. O dată ce a fost antrenată, rețeaua va fi capabilă să lucreze și cu intrări diferite față de cele cu care a fost antrenată, atîta timp cît aceste intrări nu sînt foarte diferite. Acesta este un mare avantaj față de software-ul clasic care trebuie construit special pentru a opera cu fiecare intrare particulară în parte. Aceasta ne va permite să eliminăm probabil catastrofa virgulei lipsă.

Fractali și sisteme de funcții iterate

Formele fractale sînt clase de obiecte care rezultă din evoluarea repetată a unei funcții simple, matematice, pentru a produce o formă complexă, cu un nivel de detaliu infinit. (O definiție completă ne oferă Mandelbrot: Un fractal este o mulțime a cărei dimensiune Hausdorff nu este întregă).

Descoperită de un matematician, tehnică fractalilor s-a dovedit neprețuită în construcția imaginilor pe calculator a obiectelor naturale, cu detalii și texturi ciudate. În ultimii ani, au ieșit la iveală legăturile adînci dintre fractali, sistemele de funcții iterate (IFS) și dinamica haotice, rezultatele fiind atît din domeniul biologiei cît și din cel al informaticii.

În științele naturale, metodele fractale sînt utile pentru vizualizarea dinamică haotice a sistemelor neliniare. Știința haosului, cunoscută și ca dinamică neliniară, este o tehnică de modelare matematică, folosită pentru reprezentarea unor stări complexe ale sistemelor cu buclă de reacție (feed back) de la inima umană pînă la sistemul vremii, interacțiunile dintre neuronii mamiferelor și mișcarea planetelor.

Una dintre cele mai utile aplicații ale fractalilor a fost, în industria PC-urilor, comprimarea imaginilor scante după teoria IFS-urile oferă o metodă compactă de a reprezenta o subclasă de fractali, care poate fi partiționată. Teoria lui Barnsley utilizează caracteristicile de partiționare iterativă ale IFS-urilor pentru a obține un factor de comprimare de 500:1 în unele cazuri, în altele de 10000:1.

Sisteme "L"

Chiar dacă sistemele fractale de ecuații pot produce imagini ale obiectelor naturale cum ar fi copacii sau munții, cu o stranie asemănare, este greu de depistat legătura directă între funcția $Z^2 + C \rightarrow Z$ și un organism natural. Descoperitorul fractalilor, Benoit Mandelbrot, nu s-a preocupat de

aceasta pentru că el n-a încercat să modeleze niciodată direct modul de lucru intern al proceselor biologice.

- În contrast, sistemele lui Lindenmayer, sau sistemele "L", similare fractalilor sub anumite aspecte, au fost create cu scopul explicit de a modela natura. Aristid Lindenmayer, un biolog, a dezvoltat o teorie matematică a creșterii plantelor în 1968. Un sistem "L" este o mulțime de reguli care specifică o secvență repetată de transformări ale unei forme inițiale. Regulile nu diferă față de cele BNF care specifică sintaxa limbajelor de programare, dar în loc de a analiza expresii, ele generează date. Dacă se dă o formă potrivită de pornire, și regulile, se pot genera imagini ale unor obiecte ca niște plante, care sînt incredibile de asemănătoare cu obiectul adevărat. Ca și la fractali, un volum mic de informație, poate produce un obiect detaliat. Se poate spune că un sistem "L" este o altă modalitate de a reprezenta o mulțime fractală. Dar spre deosebire de fractalii convenționali, sistemele "L" pot fi utilizate pentru a mima rezultatul final ca și diferitele stadii de creștere ale obiectului generat.

Rația mare de comprimare din sistemele "L" (și din IFS-uri) ne dă o idee asupra modului în care o porțiune de 10^8 biți ai ADN-ului poate specifica o rețea neuronală umană de 10^{11} elemente de procesare și 10^{14} conexiuni. Nici un biolog nu a găsit locul în interiorul unei plante unde regulile sistemului "L" ar putea fi memorate. Nici nu s-a dorit aceasta. Aceste sisteme "L" pot fi văzute mai repede ca un exercițiu formal pentru a înțelege cum poate fi modelat, specificat, prezis și controlat procesul natural.

Recent, oamenii ce se ocupă cu grafica pe calculator, ca de exemplu Przemyslaw Prusinkiewicz au folosit sistemele "L" ca pe o tehnică de zidire a unor imagini naturale.

Algoritmi genetici

O tehnică inspirată biologic ce nu se folosește în grafica pe calculator sînt algoritmi genetici. Invenți de John Holland în 1975. Pe

scurt, această abordare folosește programele pentru a găsi o soluție particulară într-un spațiu general de căutare, prin mimarea procesului natural de evoluție, mutație și selecție naturală. Deși algoritmi genetici sînt inspirați direct din procesele biologice în practică legătura e aproape pierdută. Algoritmi genetici sînt un model rafinat de evoluție așa cum o rețea neuronală artificială este un model al creierului nu neapărat adevărat.

Dar o imitație perfectă a naturii nu este necesară. Aceste tehnici de rezolvare a problemelor sînt valoroase în sine.

Există o altă tehnică de căutare semialeatoare numită "călire simulată", (echivalentă la bază cu algoritmi genetici), care mimează cristalizarea unui lichid în răcire (sau călirea unui metal încălzit și apoi răcit). Această tehnică este similară cu algoritmi genetici; prin generarea repetată a unei soluții de încercare testînd-o apoi cu țelul dorit, apoi se mută soluția semialeatar pentru a vedea dacă o soluție mai bună este disponibilă. Pe parcurs ce temperatura scade, programul converge spre o soluție a problemei aproape optimă. Echivalența dintre călirea simulată și algoritmi genetici vine din modul în care același proces se poate manifesta în același timp în sisteme biologice și nebiologice.

Algoritmi genetici ai lui Holland lucrează acum cu o tehnică îmbunătățită de căutare paralelă și optimizare numită "sisteme clasificatoare". Sistemele clasificatoare sînt similare cu sistemele "L" prin faptul că sînt bazate pe reguli. De asemenea, ele încorporează evoluția genetică și mutația. S-a demonstrat că ele sînt formal echivalente cu sistemele conexio-niste (rețele neuronale artificiale).

Alți cercetători au combinat algoritmi genetici cu alte tehnici - de exemplu au utilizat algoritmi genetici pentru a dezvolta modele noi de rețele neuronale. Sau au utilizat algoritmi genetici pentru a transforma programele convenționale - de exemplu programe de sortare aproape optimale pentru un anu-

mit set de date. Ca și rețelele neuronale, algoritmi genetici se potrivec foarte ușor cu hardware-ul paralel, ceea ce explică interesul unor producători mari de semiconductori (Intel, TI) pentru rețele neuronale și alte tehnici bioinformatic.

Alte tehnici bioinformatic

Logica fuzzy, sau teoria mulțimilor fuzzy este o tehnică inspirată din natură, fără a încerca să fie un model realist al unui obiect fizic. Inventată de Lofti Zadeh în 1965, logica fuzzy este o extensie a logicii matematice care permite valori noi, între valorile tari de adevărat și fals (1 și 0). Intenția este de a fi capabili să lucrăm într-un mod inteligibil cu noțiuni sau concepte imprecise, care nu au margini exacte. Aceasta nu mimează modul în care lucrează creierul uman, dar pare să se apropie de modul în care acționează gîndirea umană.

Logica fuzzy și-a găsit mulți adepți în companiile japoneze. Această tehnologie este acum un element cheie în produse ca: aparatele de fotografiat Cannon cu autofocalizare, mașinile de spălat Hitachi, transmisiile Nissan și Subaru, la fel ca și în recunoașterea scrisului de mînă la calculatorul Sony Palmtop.

Partea cealaltă a bioinformaticii

Pînă acum ne-am focalizat atenția pe modurile în care se răsfrînge biologia în programe, dar o discuție despre bioinformatică nu poate fi completă fără a menționa cealaltă parte, utilizarea calculatoarelor și a informaticii pentru a studia sistemele biologice, adică biologia computațională. Cîteva din tehnicile menționate anterior (rețele neuronale, sistemele "L") au pornit de la încercarea de înțelegere a sistemelor biologice. Dar tehnicile au devenit deja o tehnică de calcul depărtîndu-se de modelul biologic.

Cercetătorii rămîn să studieze în continuare bioorganismele ca unități de procesare a informației.

De exemplu, un autor în domeniul imunologiei face observația că sistemul imunologic, care este o rețea complexă de elemente interactive distribuite în corpul uman (așa ca filamentul de trei picioare lungime al ADN-ului uman), pot fi văzute ca procese cognitive, entități capabile de căutare paralelă distribuită, recunoașterea formelor și memorie asociativă, nu altfel decît o rețea neuronală artificială sau un sistem clasificant genetic.

Biologia computațională, a avut o influență largă în domeniul geniticii moleculare medicale, chiar dacă cercetătorii nu utilizează această frază în descrierea muncii lor. Știind că bioorganismele au rezultatul execuției de programare biochimice memorat pe o bandă codificată digital, (secvența ADN-ului genetic) cercetătorii medicali au descoperit că anumite boli sînt erori de transmisie de date, în semnalul genetic. De exemplu, fibroza cistică dezordonată ereditară, vine dintr-o singură schimbare în valorile datelor, exact ca și virgula lipsă din programele navei spațiale. Ca și cea virgulă, eroarea poate fi detectată, corectată, programul fiind însănoșit.

Natural, cele mai bune scule au fost construite pentru a îmbunătăți aceste metode: secvențiale de ADN controlate automat pe calculator (dezasamblare), mașini de reacții înlănțuite polimerice, (duplicator de benzi digitale), stații de lucru pentru proiectare moleculară, și așa mai departe. Există desigur aici și probleme bioetice care trebuie discutate, la fel ca și marile provocări tehnice. Și, cine știe, poate peste treizeci de ani, Borland sau Multiscope vor ieși pe piață cu un debugger interactiv pentru "forme-de-viață-dezvoltate-acasă", completat cu driver-e de citire/scriere pentru secvențe ADN.

(E.R.)

(Prelucrare după articolul "What Is BioComputing?",

Ray Valdés, Dr. Dobb's Journal #175, april 1991, p. 46)

CONSECO S.R.L.

cu sediul în Baia Mare , C.P.517, str. Siugariu 23/38, tel./fax: 40-94-36433

în colaborare cu

Societatea Comercială Româno - Canadiană

JM & CO. S.R.L.

din București, str. Miletin nr. 1, sectorul 3, tel.:20.33.24, tel./fax : 20.27.05

vă oferă :

Calculatorul compatibil IBM-PC, format A4 Notebook, tip FALCON 320 N

cu următoarele caracteristici:

- procesor 80386 SX, 20 MHz;
- soclu pentru 80387 SX;
- 2 MB RAM pe placă, extensibil la 8 MB;
- 1 x FDD 3,5" 1,44 MB (suportă și 720 KB);
- HDD 40 MB (cu posibilități de comandă pentru versiuni de 80 sau 120 MB);
- ecran LCD TSTM paper-white, 64 tonuri de gri, VGA 640x480;
- tastatură 82 taste cu 12 taste funcționale și tastatură numerică inclusă;
- porturi : 2 seriale, 1 paralel, conectori pentru FDD, monitor VGA color și tastatură normală exterioară;
- sistem de operare DR DOS 5;
- baterii (acumulatori) NiCd incluse, aprox. 2 ore funcționare continuă fără încărcare;
- alimentator pentru rețea c.a. 90 - 264 V;
- dimensiuni : 298 x 248 x 53mm.
- greutate (inclusiv bateriile): 3,2 Kg.

Prețul este de 2100 \$ SUA .

Toate prețurile includ atât vama cât și ICM., care sînt achitate de către furnizor.

Plata se efectuează în lei la cursul zilei plății.

Memento: Funcții Paradox 3.5

ABS	Valoarea absolută a unui număr ABS(Număr)	CURSOR-CHAR	Caracterul aflat în poziția în care se găsește cursorul CURSORCHAR ()
ACOS	Arccosinusul unui număr ACOS(Număr)	CURSORLINE	Textul de pe linia curentă a spațiului de lucru CURSORLINE ()
ARRAYSIZE	Dimensiunea unui tablou ARRAYSIZE(Tablou)	CVAR	Variația valorilor dintr-o coloană CVAR (NumTabelă, NumCîmp)
ASC	Returnează codul ASCII al unui caracter ASC(Caracter)	DATEVAL	Convertește un șir într-o dată DATEVAL (Șir)
ASIN	Arcsinusul unui număr ASIN (Număr)	DAY	Data unei zile ca și număr DAY (Dată)
ATAN	Arctangenta unui număr ATAN (Număr)	DIRECTORY	Numele directorului curent DIRECTORY ()
ATAN2	Arctangenta în patru cadrani ATAN2 (Număr1, Număr2)	DIREXISTS	Există directorul specificat? DIREXISTS (CaleDOS)
ATFIRST	Este articolul curent primul articol? ATFIRST ()	DOW	Returnează ziua săptămânii ca și șir DOW (Dată)
ATLAST	Este articolul curent ultimul articol? ATLAST ()	DRIVESPACE	Spațiul rămas pe disc DRIVESPACE (Șir)
BANDINFO	Linia curentă în generatorul de rapoarte BANDINFO ()	DRIVE-STATUS	Unitatea este pregătită? DRIVESTATUS (Șir)
BLANKDATE	Returnează o dată vidă BLANKDATE ()	EOT	Testează încercarea de poziționare după sfârșitul unei tabele EOT ()
BLANKNUM	Returnează un număr vid BLANKNUM ()	ERRORCODE	Codul corespunzător erorii de execuție celei mai recente ERRORCODE ()
BOT	Testează încercarea de poziționare înainte de începutul unei tabele BOT ()	ERROR-MESSAGE	Mesajul corespunzător erorii de execuție celei mai recente ERRORMESSAGE ()
CAVERAGE	Calculează media valorilor dintr-o coloană CAVERAGE (NumTabelă, NumCîmp)	ERRORUSER	Numele utilizatorului care a blocat un obiect ERRORUSER ()
CCOUNT	Numără valorile dintr-o coloană CCOUNT (NumTabelă, NumCîmp)	EXP	Exponențiala unui număr EXP (Număr)
CHAR-WAITING	Așteaptă un caracter în buffer-ul tastaturii? CHARWAITING ()	FAMILY-RIGHTS	Verifică drepturile de acces la familia unei tabele FAMILYRIGHTS (NumTabelă, Șir)
CHECKMARK-STATUS	Cîmpul curent este marcat în cerere? CHECKMARKSTATUS ()	FIELD	Numele cîmpului curent FIELD ()
CHR	Convertește un cod ASCII în caracterul corespunzător CHR (Număr)	FIELDINFO	Returnează indicatorul cîmpului curent în modul Report sau Form FIELDINFO ()
CMAX	Valoarea maximă dintr-o coloană CMAX (NumTabelă, NumCîmp)	FIELDNO	Poziția unui cîmp într-o tabelă FIELDNO (NumCîmp, NumTabelă)
CMIN	Valoarea minimă dintr-o coloană CMIN (NumTabelă, NumCîmp)	FIELDRIGHTS	Verifică drepturile de acces la un cîmp dintr-o tabelă FIELDRIGHTS (NumTabelă, NumCîmp, Șir)
CNPV	Valoarea netă a valorilor unei coloane CNPV (NumTabelă, NumCîmp, Număr)	FIELDSTR	Valoarea curentă a unui cîmp ca și șir FIELDSTR ()
COL	Coloana în care este poziționat cursorul pe ecranul utilizator PAL COL ()	FIELDTYPE	Tipul datelor din cîmpul curent FIELDTYPE ()
COLNO	Numărul coloanei curente din imaginea aflată în spațiul de lucru COLNO ()	FILESIZE	Dimensiunea unui fișier în bytes FILESIZE (NumFișier)
COS	Cosinusul unui unghi COS (Număr)	FILL	Returnează un șir de lungime specificată, format din aparițiile repetate ale unui caracter FILL (Expresie, Număr)
CSTD	Deviația standard a valorilor dintr-o coloană CSTD (NumTabelă, NumCîmp)		
CSUM	Suma valorilor dintr-o coloană CSUM (NumTabelă, NumCîmp)		

Memento: Paradox 3.5

FORM	Returnează numele formei curente pentru tabela curentă FORM()
FORMAT	Formatează o expresie FORMAT (SpecFormat, Șir)
FORMTYPE	Verifică tipul formei curente FORMTYPE (Șir)
FV	Valoarea următoare dintr-o serie de "cash flows" FV (Număr1, Număr2, Număr3)
GETCHAR	Citește un caracter de la tastatură GETCHAR ()
GRAPHTYPE	Tipul graficului curent GRAPHTYPE ()
HELPMODE	Tipul de help curent HELPMODE ()
IMAGEC-AVERAGE	Media valorilor din coloana curentă a imaginii curente IMAGEAVERAGE ()
IMAGEC-COUNT	Afișează numărul de valori din coloana curentă a imaginii curente IMAGECCOUNT ()
IMAGECMAX	Valoarea maximă din coloana curentă a imaginii curente IMAGECMAX ()
IMAGECMIN	Valoarea minimă din coloana curentă a imaginii curente IMAGECMIN ()
IMAGECSUM	Suma valorilor din coloana curentă a imaginii curente IMAGECSUM ()
IMAGENO	Poziția imaginii curente din spațiul de lucru IMAGENO ()
IMAGETYPE	Tipul imaginii curente IMAGETYPE ()
INT	Partea întreagă a unui număr INT (Număr)
ISASSIGNED	A fost asignată o variabilă sau un element de matrice? ISASSIGNED (Șir)
ISBLANK	Expresia este vidă? ISBLANK (Expresie)
ISBLANK-ZERO	Valorile vide vor fi tratate în calcule ca și zero? ISBLANKZERO ()
ISEMPTY	Tabela este goală? ISEMPTY (NumTabelă)
ISENCRYPTED	Tabela este protejată prin parolă? ISENCRYPTED (NumTabelă)
ISFIELDVIEW	Ne aflăm în modul Field View? ISFIELDVIEW ()
ISFILE	Există fișierul specificat? ISFILE (Șir)
ISFORMVIEW	Tabela curentă este afișată printr-o formă? ISFORMVIEW ()
ISINSERT-MODE	Paradox este în modul de inserare? ISINSERTMODE ()
ISLINK-LOCKED	Tabela curentă est blocată prin legare? ISLINKLOCKED ()
ISMULTI-FORM	Forma tabelii curente conține forme și pentru alte tabele? ISMULTIFORM (NumTabelă, NumFormă)
ISMULTI-REPORT	Raportul conține tabele legate? ISMULTIREPORT (NumTabelă, NumRaport)
ISRUNTIME	Programul este executat de versiunea Paradox Runtime? ISRUNTIME ()
ISSHARED	Tabela se află într-un director partajat? ISSHARED (NumTabelă)
ISTABLE	Există tabela? ISTABLE (NumTabelă)
ISVALID	Are fișierul un conținut valid? ISVALID ()
LEN	Lungimea unui șir LEN (Expresie)
LINKTYPE	Tipul de legătură dintre tabele aparținând unei forme multitabelare LINKTYPE ()
LN	Logaritmul natural al unui număr LN (Număr)
LOCKSTATUS	Cîte blocări de un anumit tip au fost plasate asupra tabelii curente? LOCKSTATUS (NumTabelă, TipBlocare)
LOG	Logaritmul în baza 10 al unui număr LOG (Număr)
LOWER	Convertește un șir în litere mici LOWER (Expresie)
MATCH	Compară un șir cu un șablon MATCH (Șir, Șir pattern [,NumListăVariabile])
MAX	Maximul dintre două numere MAX (Număr1, Număr2)
MEMLEFT	Memorie RAM disponibilă unei aplicații MAMLEFT ()
MENUCHOICE	Alegerea curentă din meniul curent Paradox MENUCHOICE ()
MIN	Minimul dintre două numere MIN (Număr1, Număr2)
MOD	Restul împărțirii unui număr cu altul MOD (Număr1, Număr2)
MONITOR	Tipul monitorului folosit MONITOR ()
MONTH	Luna unei date ca un număr MONTH (Dată)
MOY	Luna unei date ca șir MOY (Dată)
NETTYPE	Ce fel de rețea este folosită? NETTYPE ()
NFIELDS	Numărul cîmpurilor dintr-o tabelă NFIELDS (NumTabelă)
NIMAGERECORDS	Numărul de articole legate sau nu din imaginea curentă NIMAGERECORDS ()
NIMAGES	Numărul de imagini din spațiul de lucru NIMAGES ()
NKEYFIELDS	Numărul de cîmpuri cheie dintr-o tabelă NKEYFIELDS (NumTabelă)
NPAGES	Numărul de pagini dintr-un raport sau dintr-o formă NPAGES ()
NRECORDS	Numărul de articole dintr-o tabelă NRECORDS ()

NROWS	Numărul liniilor din imaginea sau raportul curent NROWS ()
NUMVAL	Convertește un șir într-un număr NUMVAL (Șir)
PAGENO	Numărul paginii curente dintr-un raport sau dintr-o formă PAGENO ()
PAGEWIDTH	Lățimea paginii din raportul curent PAGEWIDTH ()
PI	Returnează valoarea pi PI ()
PMT	Calculează rata de amortizare PMT (Număr1, Număr2, Număr3)
POW	Ridică un număr la o putere POW (Număr1, Număr2)
PRINTER-STATUS	Imprimanta este pregătită? PRINTERSTATUS ()
PRIVDIR	Numele directorului privat al rețelei utilizatorului PRIVDIR ()
PV	Valoarea actuală a unei serii de plăți PV (Număr1, Număr2, Număr3)
QUERY-ORDER	Ordinea implicită a câmpurilor din tabela de răspuns QUERYORDER ()
RAND	Returnează un număr aleator RAND ()
RECNO	Numărul articolului curent RECNO ()
RECORD-STATUS	Verifică starea articolului curent RECORDSTATUS (Șir)
RETRY-PERIO	Care este valoarea perioadei de reîncercare? RETRYPERIOD ()
RMEMLEFT	Returnează nivelul de memorie disponibil RMEMLEFT ()
ROUND	Rotunjește un număr ROUND (Număr1, Număr2)
ROW	Linia în care este poziționat cursorul pe ecranul utilizator PAL ROW ()
ROWNO	Afișează numărul curent al rândului dintr-o imagine, raport sau formă ROWNO ()
SDIR	Numele căii programului curent SDIR ()

SEARCH	Caută un subșir într-un șir SEARCH (Subșir, Șir)
SIN	Sinusul unui unghi SIN (Număr)
SORTORDER	Ce fel de ordine de sortare a fost instalată? SORTORDER ()
SPACES	Returnează un șir de spații SPACES (Număr)
SQRT	Rădăcina pătrată a unui număr SQRT (Număr)
STRVAL	Convertește o expresie într-un șir STRVAL (Expresie)
SUBSTR	Returnează un subșir dintr-un șir SUBSTR (Șir, Număr1, Număr2)
SYSCOLOR	Care sînt culorile curente ale sistemului Paradox? SYSCOLOR (Număr)
SYSMODE	Modul curent Paradox SYSMODE ()
TABLE	Numele tabelului curent TABLE ()
TABLE-RIGHTS	Verifică drepturile de acces la o tabelă Paradox TABLERIGHTS (NumTabelă, Șir)
TAN	Tangenta unui unghi TAN (Număr)
TIME	Ora zilei TIME ()
TODAY	Data zilei de astăzi TODAY ()
TYPE	Tipul unei expresii TYPE (expresie)
UPPER	Convertește un șir în caractere mari UPPER (Expresie)
USERNAME	Care este numele utilizatorului curent? USERNAME ()
VERSION	Afișează numărul de versiune Paradox folosit VERSION ()
WINDOW	Returnează un șir ce include conținutul ferestrei de mesaje a Paradox-ului WINDOW ()
YEAR	Anul unei date YEAR (Dată)

Taste funcționale

<F1>	Help
<F2>	Do-It
<F3>	Up Image
<F4>	Down Image
<F5>	Example
<F6>	Checkmark
<F7>	Form Toggle
<F8>	Clear Image
<F9>	Edit
<F10>	Menu (Menu)

<Alt>/<F3>	Instant Script Record
<Alt>/<F4>	Instant Script Play
<Alt>/<F5>	Field View
<Alt>/<F6>	Check Plus
<Alt>/<F7>	Instant Report
<Alt>/<F8>	Clear All
<Alt>/<F9>	Coedit
<Ctrl>/<F6>	Check Descending
<Ctrl>/<F7>	Instant Graph
<Shift>/<F6>	Group By

Combinății speciale de taste

<Alt>/<C>	Color Palette - Comută paleta de culori on/off în timpul proiectării unei forme
<Ctrl>/<D>	Ditto - La editare, sau introducerea date, copiază valoarea aceluiași câmp din articolul anterior (duplicare valoare)
<Ctrl>/<F>	Field View - Mută cursorul în interiorul unui câmp permițând editarea lui (la fel ca <Alt>/<F5>)
<Alt>/<K>	Key Viol - Vizualizează articolele la care apar situații conflictuale datorate unor duplicări de chei
<Ctrl>/<L>	Resync - Resincronizează articolele într-o formă multitabelară
<Alt>/<L>	Lock Toggle - Blochează sau deblochează un articol
<Ctrl>/<O> <Alt>/<O>	DOS - Suspendă temporar sesiunea Paradox și predă controlul la nivelul sistemului de operare DOS. Pentru revenire în Paradox trebuie tastat Exit
<Ctrl>/<R>	Rotate - Rotește coloanele unei tabelă
<Alt>/<R>	Refresh - Actualizează imaginea afișată pe ecran cu modificările efectuate în tabelă de utilizatorii din rețea
<Ctrl>/<U>	Undo - Reface incremental modificările făcute într-o tabelă într-unul din modurile Edit, Coedit sau DataEntry
<Ctrl>/<V>	Vertical Ruler Toggle - Afișează sau maschează contorul liniilor în timpul editării unui raport sau a scrierii unui program
<Alt>/<X>	CrossTab - Face referințele încrucișate pentru imaginea curentă
<Ctrl>/<Y>	Delete Line - Șterge linia curentă la editarea unui raport sau la scrierea unui program
<Ctrl>/<Z>	Zoom - Localizează prima apariție a unei valori într-o tabelă
<Alt>/<Z>	Zoom Next - Localizează următoarea apariție a unei valori într-o tabelă
<Ctrl>/<Backspace>	"Turbo" Backspace - Șterge conținutul câmpului curent în mod Edit sau DataEntry
<Tab>	Tab - Deplasare între câmpuri în mod vizualizare tabelă și între coloane în generatorul de rapoarte.

Obiecte Paradox

Extensie	Tipul obiectului
.DB	Tabelă
.F sau .F??	Formă
.R sau .R??	Format raport
.SET	Setări imagine
.VAL	Marcaje de validare
.PX	Index primar
.X??	Index secundar
.Y??	Index secundar
.SC	Script (program)
.G	Specificații grafic
Semnul ? înlocuiește un singur caracter în extensie.	

Specificații de format pentru leșire date

Tip format	Semnificație		Tip date
Dimensiune	Wn	Stabilește dim. permisă	toate
	Wn.m	Setează dim. și nr. poz. zec.	N,\$
Aliniere	AL	Aliniere la stînga	toate
	AR	Aliniere la dreapta	toate
	AC	Centrează valoarea	toate
Caractere	CU	Convertește în litere mari	toate
	CL	Convertește în litere mici	toate
	CC	Convertește în forma inițială	toate
Editare	E\$	Include semnul \$	N,S,\$
	EC	Include virgula ca separator	N,S,\$
	EZ	Afișează zerourile nesemnif.	N,S,\$
	EB	Folosește spațiu în locul zerourilor nesemnificative	N,S,\$
	ES	Afișează folosind notația științifică	N,S,\$
	E*	Folosește * pentru zerourile nesemnificative	N,S,\$
	EI	Folosește formatul internațional pentru numere	N,S,\$
Semn	S+	Afișează semnul + sau -	N,S,\$
	S-	Afișează semnul -	N,S,\$
	SP	Afișează val. negative în ()	N,S,\$
	SD	Afișează notația DB sau CR	N,S,\$
	SC	Afișează CR după valorile negative	N,S,\$
Date	D1	Format: ll/zz/aa	D
	D2	Format: Luna zz. aaaa	D
	D3	Format: ll/zz	D
	D4	Format: ll/aa	D
	D5	Format: zz-lll-aa	D
	D6	Format: ll zz	D
	D7	Format: zz-lll-aaaa	D
	D8	Format: ll/zz/aaaa	D
	D9	Format: zz.ll aa	D
	D10	Format: zz/ll/aa	D
	D11	Format: aa-ll-zz	D
Logic	LY	Utilizează Yes/No pentru True/False	L
	LO	Utilizează On/Off pentru True/False	L
Tip date: N = Numeric, S = numeric scurt, \$ = Valoare, D = Dată, L = Valoare logică			

Paradox este un S.G.B.D. al firmei:

Borland International Inc.

P.O. Box 660001, Scotts Valley, CA 95067-0001

Reprezentant autorizat în România:

S.C. Logic

- Sibiu, Calea Dumbrăvii 21

tel. (924) 46652, 45475, fax: (924) 46704

- București, b-dul Tineretului 8-10

tel. (90) 754900, fax: (90) 757135

Cine indexează mai bine ?

Nu, nu e vorba de salarii (din păcate...) ci de fișiere. Știut fiind că în general orice aplicație serioasă nu poate să se descurce fără fișiere secvențial-indexate (sau ceva similar), vă prezentăm un program - scris și testat sub Microfocus Cobol v1.2 - pe care vă rugăm să-l rescrieți în alt limbaj, într-o versiune care să se apropie cât mai mult de cea pe care v-o prezentăm. Scopul - obținerea unor cifre relative la viteza de execuție, care să permită o opțiune pentru cine pornește să construiască o aplicație.

Am decis să 'lansăm' ca model un program Cobol pe considerentul că este ușor de citit de către oricine. Desigur, acest test nu poate oferi date decisive pentru orice situație. Dar înfăptuiri pertinente sperăm să putem sintetiza, cu ajutorul dumneavoastră. Vă rugăm să ne remiteți nu numai programul sursă și executabil, ci și date relative la calculatorul pe care ați rulat și timpul obținut. Varianta prezentată, scrisă și rulată de Man Vasile, de la Oficiul de Calcul al D.Tc. Mureș, pe un 286/15 MHz, cu un harddisk de 40 Mo cu un timp mediu de acces de 28 ms,

cu 1 Mo RAM, lansată de sub XTGold cu buffers=35 și files=50 în CONFIG.SYS, fără vreun program cache activ, a avut nevoie de 1h30'58,44" pentru generare, 2'50,6" pentru prima consultare și 1'33,2" pentru a doua consultare. Fișierul de test ocupă 3,4 Mo, indexul 2,5 Mo iar programul executabil cca 111,5 ko. În aceleași condiții, dar sub Cobol 2.0, (care altfel se pare că uneori are ciudățenii) timpii sînt 8'54,65", 1'17,04" și 36,03", iar dimensiunile 3,6 Mo și respectiv 2,5 Mo.

Iosif Fettich

```

1 IDENTIFICATION DIVISION.
2 PROGRAM-ID. DEMCOBIF.
3 * REMARKS.  PROGRAM DEMCOBIF CREAZA UN FIS. SEC.V-INDEXAT
4 * IN ACCES DIRECT DE CCA 10000 ARTICOLE APOI INTEROGHEA-
5 * ZA UN ACCES DIRECT 1000 ARTICOLE IN DOUA MODALITATI.
6 ENVIRONMENT DIVISION.
7 CONFIGURATION SECTION.
8 INPUT-OUTPUT SECTION.
9 FILE-CONTROL.
10 SELECT DEMCOBIF ASSIGN TO "DEMCOB" ORGANIZATION INDEXED
11 ACCESS RANDOM RECORD KEY ABACT.
12 DATA DIVISION.
13 FILE SECTION.
14 FD DEMCOBIF LABEL RECORD STANDARD
15 BLOCK CONTAINS 8 RECORDS.
16 01 ARTACT.
17 02 ABACT.
18 03 AB1 PIC X(6).
19 03 AB2 PIC X(2).
20 02 ALTC1 PIC X(6).
21 02 ALTC2 PIC Z(5)9.
22 02 ALTC3 PIC X(6).
23 02 ALTC4 PIC Z(5)9.
24 WORKING-STORAGE SECTION.
25 77 I PIC X(6) COMP.
26 77 J PIC X(6) COMP.
27 77 K PIC X(6) COMP VALUE 0.
28 77 L PIC X(4) COMP VALUE 0.
29 77 REST PIC X(3) COMP.
30 77 SECVSF PIC X(6) COMP VALUE 10000.
31 77 SCHIMB PIC XX.
32 01 WDATA-EXEC.
33 02 MES1 PIC X(22) VALUE ' INDEXARE INCEPUTA ' .
34 02 FILLER PIC X(17) VALUE ' - (IHHMSS.) : ' .
35 02 HMS PIC 99B99B99B99.
36 01 WC11-GEN.
37 02 WC11-G1 PIC X(54) VALUE
38 'ABCDEFGHIJKLMNPOQRSTUVWXYZabcdeghij
39 'klmnopqrstuvwxyz .
40 01 WCH-G2 REDEFINES WC11-G1 PIC XX OCCURS 27.
41 02 WCH-CIT1 PIC X(8) OCCURS 1000.
42 PROCEDURE DIVISION.
43 INCEP.
44 OPEN OUTPUT DEMCOBIF
45 DISPLAY ' PROGRAM DEMCOBIF '
46 MOVE SPACES TO WCH-CIT1RE
47 MOVE 'INF.GR1' TO ALTC1
48 MOVE 'INF.GR2' TO ALTC3
49 ACCEPT HMS FROM TIME
50 DISPLAY WDATA-EXEC.
51 PERFORM GEN-FIS VARYING I FROM 1 BY 1 UNTIL I > 50000
52 ACCEPT HMS FROM TIME
53 MOVE ' INDEXARE TERMINATA ' TO MES1
54 DISPLAY WDATA-EXEC.
55 CLOSE DEMCOBIF
56 OPEN INPUT DEMCOBIF
57 ACCEPT HMS FROM TIME
58 MOVE 'INTEROGARE 1 INCEPUTA' TO MES1
59 DISPLAY WDATA-EXEC.
60 MOVE 1001 TO J
61 PERFORM CIT1-FIS VARYING I FROM 1 BY 1 UNTIL I > 500
62 ACCEPT HMS FROM TIME
63 MOVE 'INTEROGARE 1 TERMINATA' TO MES1
64 DISPLAY WDATA-EXEC.
65 ACCEPT HMS FROM TIME
66 MOVE 'INTEROGARE 2 INCEPUTA' TO MES1
67 DISPLAY WDATA-EXEC.
68 PERFORM CIT2-FIS VARYING I FROM 1 BY 1 UNTIL I > 1000
69 ACCEPT HMS FROM TIME
70 MOVE 'INTEROGARE 2 TERMINATA' TO MES1
71 DISPLAY WDATA-EXEC.
72 CLOSE DEMCOBIF
73 STOP RUN.
74 GEN-FIS.
75 SUBTRACT 1 FROM SECVSF
76 DIVIDE 26 INTO I GIVING K REMAINDER REST
77 PERFORM GEN-C11
78 MOVE I TO AB1 ALTC2 ALTC4
79 PERFORM SCR-FIS
80 DIVIDE 26 INTO SECVSF GIVING K REMAINDER REST
81 PERFORM GEN-CH
82 MOVE SECVSF TO AB1 ALTC2 ALTC4
83 PERFORM SCR-FIS.
84 GEN-CH.
85 ADD 1 TO REST
86 MOVE WCH-G2 (RES1) TO AB2
87 MOVE WCH-G2 (REST) TO SCHIMB
88 MOVE WCH-G2 (27) TO WCH-G2 (REST)
89 MOVE SCHIMB TO WCH-G2 (27).
90 SCR-FIS.
91 ADD 1 TO J
92 IF J > 95
93 MOVE 0 TO J
94 ADD 1 TO L
95 IF L < 1001
96 MOVE ABACT TO WCH-CIT1 (L).
97 WRITE ARTACT INVALID KEY
98 DISPLAY 'CHEIE DUBLA / NECRESCATOARE' ABACT.
99 CIT1-FIS.
100 MOVE WC11-CIT1 (I) TO ABACT
101 READ DEMCOBIF INVALID KEY
102 DISPLAY ' CHEIE NEGASITA ' ABACT.
103 SUBTRACT 1 FROM J
104 MOVE WCH-CIT1 (J) TO ABACT
105 READ DEMCOBIF INVALID KEY
106 DISPLAY ' CHEIE NEGASITA ' ABACT.
107 CIT2-FIS.
108 MOVE WCH-CIT1 (I) TO ABACT
109 READ DEMCOBIF INVALID KEY
110 DISPLAY ' CHEIE NEGASITA ' ABACT.

```

DEMCOBIF.CBL

Programele de educație Novell

Programul NAEC

NAEC reprezintă abrevierea de la Novell Authorized Education Center. Aceste centre primesc de la Novell informațiile tehnice necesare în procesul de pregătire a viitorilor experți în utilizarea produselor firmei.

Pentru ca un centru de pregătire să devină un NAEC el trebuie să îndeplinească o serie de cerințe extrem de exigente atât din punctul de vedere al dotării tehnice cât și din punctul de vedere al pregătirii instructorilor din centrul respectiv.

Astfel instructorii trebuie în prealabil să urmeze cursuri autorizate Novell fiind supuși apoi unei testări dure; doar după trecerea acestor teste candidatul devine CNI (Certified NetWare Instructor). În continuare proaspătul CNI trebuie să-și reconfirme statutul susținând periodic noi examene.

Prin intermediul programului NAEC Novell își susține programele de autorizare și certificare; calitatea pregătirii asigurate de NAEC este controlată în amănunt de firma Novell, fiind considerată la fel de importantă ca și calitatea celorlalte produse Novell.

Prin intermediul acestui program, Novell reunește distribuitorii, reseller-i, OEM (Original Equipment Manufacturer), consultanți centre de pregătire într-o alianță care are drept scop final asigurarea calității procesului de pregătire în utilizarea optimă a produselor Novell.

În cadrul NAEC se desfășoară două programe principale: CNE (Certified NetWare Engineer) și CNI (Certified NetWare Instructor).

Programul CNE se referă la certificarea Novell a unor ingineri din cadrul diferitelor organizații. O organizație care are în cadrul ei un CNE se bucură de un sprijin deosebit în domeniul tehnic și comercial din partea firmei Novell.

Certificarea CNE se obține prin examene care urmează cursurilor absolvite într-un NAEC. Certificarea CNE este și ea urmată de noi cursuri, de noi examene, pregătirea referindu-se de fiecare dată la cele mai noi evoluții în domeniul rețelelor de calculatoare, evoluții aflate

de o bună bucată de timp în strânsă legătură cu Novell.

Pregătirea în cadrul NAEC este asigurată în mod obligatoriu de CNI. Un CNI se specializează în unul sau mai multe din domeniile următoare:

- specialist NetWare
- specialist în comunicații
- specialist în gestiunea rețelei
- specialist în dezvoltare
- specialist în legături Unix - TCP/IP

Cu începere din luna decembrie firma SCOP va pune la dispoziția solicitanților primul NAEC din România, asigurând participanților la cursuri o pregătire conformă celor mai exigente standarde internaționale ale momentului.

Politica Novell în comerțul internațional

Novell are în vedere următoarele canale de distribuție pentru produsele sale:

- International Distributor. Reprezintă un reseller (revinzător) care comercializează și distribuie produse numai prin intermediul unui Novell Authorized Reseller.

- International Remarketer. Reprezintă un Internațional Distributor, International System Integrator sau International Value Added Reseller.

- International System Integrator. Reprezintă un reseller care oferă soluții integrate și care primește produsele Novell direct de la Novell.

- International Value Added Reseller. Reprezintă un reseller care oferă o soluție la cheie, adresată unei anumite nișe a pieței, soluția constând din produse software proprii la nivelul aplicațiilor. Soluția este implementată pe produse Novell.

- Novell Authorized Reseller. Reprezintă un reseller care cumpără produse de la International Remarketers și le revinde către End Users (utilizatorii finali).

Novell acordă unei firme, în baza atentei analize a posibilităților tehnice

dar și comerciale, unul din gradele enumerate mai sus.

În cazul României, firma SCOP a fost investită ca International Remarketer. Această înțelegere permite firmei SCOP să comercializeze produse Novell din cadrul unei arii bine definite și stabilite de obicei la nivelul frontierelor naționale. În cadrul acestei arii, comercializarea produselor Novell se face prin intermediul unor Novell Authorized Resellers sau direct, dar numai strict în aria stabilită. Este cu desăvârșire interzisă comercializarea de produse Novell utilizând agenți distribuitori, etc. În afara ariei stabilite. Dacă totuși are loc o astfel de comercializare, Novell nu mai are nici un fel de obligație referitoare la garanția produsului exportat, de asemenea se pierde licența pentru produsul respectiv, lucru care duce la pierderea oricărui suport tehnic din partea firmei Novell, ca și, aspect mult mai grav, la imposibilitatea ca ulterior să se obțină versiunea următoare a produsului în schimbul unei sume reduse de bani (așa-numitul upgrade).

În momentul de față, în România există mai multe firme care practică comercializarea unor produse Novell fără a avea dreptul să facă acest lucru; deci atunci când cumpărați un sistem de operare pentru rețeaua dvs. este foarte indicat să vă interesați de unde a obținut furnizorul dvs. produsul respectiv.

În cazul în care nu l-a obținut pe canalul Novell normal este preferabil să cîntăriți mai atent decizia dvs. deoarece veți pierde garanția produsului, suportul tehnic Novell ca și posibilitatea de upgrade. Mare atenție la produsele Novell pirat deoarece achiziția unui astfel de produs vă poate cauza în viitor mari neplăceri concretizate în amenzi foarte mari.

Și un ultim sfat: dacă furnizorul dvs. declară că este distribuitor autorizat Novell, cereți-i documentele doveditoare.

Oricum, firma SCOP vă stă cu plăcere la dispoziție pentru orice informație referitoare la produsele Novell; nu vă rămîne decît să ne contactați telefonic sau să ne oferiți plăcerea unei vizite.

ing. Sturzoiu Marius




NOVELL INTERNATIONAL AUTHORISED REMARKETER

Vă propune integrarea în sistemul autorizat de distribuție în România.
Veți beneficia astfel în mod oficial de:

- programul de upgrade al firmei NOVELL
- autorizarea firmei dvs. ca NOVELL AUTHORISED RESELLER
- suportul tehnic și de marketing necesar.



 **Sunmagraphics** / HOUSTON INSTRUMENT



AUTHORISED REMARKETER IN ROMANIA

Telefon: 11.74.21 / 11.92.48.

Fax: 11.73.74.

Noi sîntem soluția !

Rețelele - o analiză a costurilor "la bani mărunți"

Administratorii văd în primul rând avantajele rețelilor și consecința lor imediată, optimizarea proceselor de comunicație. Dar conducerea întreprinderilor și contabilii, de cele mai multe ori, nu au același punct de vedere. Ei decid în funcție de rentabilitate.

Fiecare întreprindere are colaboratori (angajați) care au idei nemăpomenite și un simț deosebit pentru tehnologii noi. Ei recunosc cu precizie strategiile actuale sau viitoare care ar putea optimiza procesul zilnic de lucru. Dar transpunerea acestor idei în practică eșuează de multe ori, deoarece acești angajați nu sînt în stare să evalueze costurile acestor noi tehnologii. Conducătorii întreprinderilor și contabilii, care decid dacă un proiect se realizează sau nu, se gîndesc doar la finanțe. Urmarea:

Deseori strategiile de perspectivă nu se materializează, ceea ce duce la pierderea competitivității întreprinderii. O "viziune" tehnică nu este de ajuns. Trebuie demonstrat avantajul în afaceri pe care îl aduce fiecare bănuț cheltuit.

În cazul introducerii unor înnoiri tehnologice, trebuie analizată influența acestui pas asupra întreprinderii. Trebuie făcută o analiză a costurilor.

Mulți administratori evită analiza costurilor, deoarece sînt de părere că aceasta ar fi prea com-

plicată. Totuși problema este mult mai simplă decît pare la prima vedere. Cel mai important lucru este interpretarea corectă a cifrelor. Mai este nevoie, în afară de un soft de calcul tabelar, de puțină înțelegere pentru terminologia folosită în lumea finanțelor (vezi cașeta).

Mijloace ajutătoare pentru analiza costurilor

După ce v-ați familiarizat cu termenii și concepțiile cele mai importante din "lumea investițiilor", puteți începe analiza. Există mai multe soluții pentru calcularea costurilor unui proiect, dintre care două sînt cele folosite mai des: analiza valorii actuale și analiza timpului de amortizare.

Valoarea curentă și valoarea netă actuală

Cu siguranță, una din cele mai răspindite metode este cea a analizei valorii curente, numită și "valoare în numerar". Aici se combină concepțiile taxei de scont, valorii banilor în timp și creșterea în timp a valorii banilor.

Intervalul de timp pentru care se efectuează analiza va fi de pildă 2 ani. S-a aflat de la contabilitate că taxa actuală de scont este de 10%. La începutul anului 1991 s-au investit 10 mărci și se așteaptă ca prin introducerea proiectului să se economisească pînă la sfîrșitul anului 1991 5

mărci și alte 5 mărci pînă la sfîrșitul anului 1992.

Deci analiza va arăta, mai întîi, astfel:

- Investiție 1991: 10 mărci
- Economisire 1991: 5 mărci
- Economisire 1992: 5 mărci.

Formulele folosite pentru calcularea valorii în numerar oferă o metodă pentru calcularea pe de o parte a puterii de cumpărare viitoare a banilor și pe de altă parte, a influenței acesteia asupra fluxului de capital.

Cele mai multe programe de calcul tabelar conțin deja formula corespunzătoare; aceasta este:

$$\text{Valoarea netă actuală} = FCn/(1+d)^a$$

- FCn = fluxul de capital în anul n (0 = anul în curs)
- d = taxa de scont
- a = numărul de ani pentru care se face calculul

Valoarea în numerar a investiției a fost de 10 mărci. Valoarea corespunzătoare a capitalului ce va intra sub formă de economii va fi de doar 8,68 mărci, cu toate că se economisesc în total 10 mărci pînă la sfîrșitul anului 1992. La această sumă se ajunge dacă economiile pronosticate pînă la sfîrșitul anului 1992 se convertesc în mărci "actuale", deci în valoare numerar.

Valoarea netă actuală se referă la diferența dintre întregul capital care intră și cel care iese. Această valoare trebuie să fie în orice caz pozitivă, deci mai mare decît

zero. Dacă valoarea netă actuală este negativă, proiectul nu este de dorit, deoarece el costă întreprinderea mai mult decît îi aduce.

Valoarea netă actuală din exemplul de mai sus este de 8,68 mărci minus 10 mărci investiția, deci un minus de 1,32 mărci. La aceasta se mai adaugă costul banilor, adică taxa de scont. Ar trebui deci adăugat încă o marcă, reprezentînd 10% din investiție. Prin urmare, proiectul ar fi respins, deoarece ar duce întreprinderea la un minus și nici măcar valoarea investiției nu ar putea fi acoperită.

Timpul de reîntoarcere a capitalului (timpul de amortizare)

Un alt element ajutător pentru efectuarea unei analize a costurilor este calculul timpului în care capitalul investit se reîntoarce.

Timpul de reîntoarcere, denumit și timp de amortizare, este numărul anilor necesari pentru recuperarea investițiilor dintr-un proiect. Această metodă este ușor de înțeles și implică calcule simple. Dacă de exemplu la începutul anului 1991 s-au cheltuit 300 de mărci și anual se recuperează 100 de mărci, atunci perioada de amortizare este de 3 ani.

Dacă această metodă este atît de simplă, se pune problema de ce să se mai folosească cea a valorii actuale. Metoda tim-

pului de amortizare are unele limitări. Ea nu ține cont de valoarea în timp a banilor și nici de fluxul de capital după prima investire. Cel mai bine se poate explica acest lucru cu un exemplu:

Proiect A Proiect B

Investiții:	300DM	300DM
Economie 1991:	100DM	0
Economie 1992:	100DM	0
Economie 1993:	100DM	300DM
Timpul de recuperare	3 ani	3 ani

Ambele proiecte necesită o investiție de 300 de mărci și 3 ani pentru a recupera acești bani. Analiza recuperării fondurilor arată că aceste două proiecte au aceeași valoare. Aceasta este însă o concluzie pripită. În cazul proiectului A banii se reîntorc mai repede decât în cazul proiectului B, și prin urmare ei pot fi reinvestiți. Din punctul de vedere contabil prima variantă este mai bună.

Calculul reîntoarcerii fondurilor cu siguranță nu este metoda care trebuie folosită în cazul proiectelor mari și de durată. Există însă proiecte simple, care au o perioadă de amortizare atât de scurtă, încât e superfluă analiza valorii actuale. La unele LAN-uri se ajunge la o perioadă de amortizare de doar câteva luni, folosirea lor ducând la o creștere foarte mare a productivității și la economii mari.

Dacă perioada de amortizare este mai mică de un an, atunci de regulă nu este necesară analiza detaliată a costurilor. Ea nu mai aduce informații importante, suplimentare.

Costuri de exploatare

Alternativa 1: LAN

An	Descriere	Costuri	Inflație în %	Costuri inclusiv inflație
1991	Școlarizare	10000	5	10000
	Întreținere	4000	5	4000
	Productivitate	0	5	0
1992	Școlarizare	5000	5	5250
	Întreținere	4000	5	4200
	Productivitate	0	5	0
1993	Școlarizare	0	5	0
	Întreținere	4000	5	4410
	Productivitate	0	5	0
1994	Școlarizare	0	5	0
	Întreținere	4000	5	4631
	Productivitate	0	5	0
1995	Școlarizare	0	5	0
	Întreținere	4000	5	4862
	Productivitate	0	5	0
1996	Școlarizare	0	5	0
	Întreținere	4000	5	5105
	Productivitate	0	5	0

Alternativa 2: Mainframe

An	Descriere	Costuri	Inflație în %	Costuri inclusiv inflație
1991	Școlarizare	0	5	0
	Întreținere	8000	5	8000
	Productivitate	20000	5	20000
1992	Școlarizare	0	5	0
	Întreținere	8000	5	8400
	Productivitate	30000	5	31500
1993	Școlarizare	0	5	0
	Întreținere	8000	5	8820
	Productivitate	30000	5	33075
1994	Școlarizare	0	5	0
	Întreținere	8000	5	9261
	Productivitate	30000	5	34729
1995	Școlarizare	0	5	0
	Întreținere	8000	5	9724
	Productivitate	30000	5	36465
1996	Școlarizare	0	5	0
	Întreținere	8000	5	10210
	Productivitate	30000	5	38288

În primele două tabele se arată costurile de exploatare în ambele variante. Dacă se menține soluția mainframe vor apărea pierderi de productivitate.

Reducerea costurilor prin "downsizing"

Procedeeul de "downsizing" este acum la modă. Într-un proiect tipic de downsizing trebuie analizate costurile pentru transferarea exploatarei unor aplicații de pe mainframe pe calculatoare mai ieftine, comparativ cu costurile menținerii modului actual de exploatare a aplicațiilor.

Trebuie calculate costurile și veniturile în ambele alternative. Pe de o parte este vorba de valori care sînt evidente și care apar o singură dată sau de mai multe ori. Pe de altă parte există costuri și venituri, care nu pot fi apreciate la prima vedere. Ele apar de mai multe ori.

Primul pas al analizei constă în estimarea valorilor costurilor și veniturilor. Pentru evaluarea financiară sînt necesare atât pricepere managerială cît și putere de imaginație.

Sînt mult mai mulți cei pe care mai degrabă această evaluare, și nu formulele matematice, îi duc la disperare. Dacă se uită un factor relevant de costuri sau anumite economisiri, întreaga analiză este incorectă, chiar dacă toate datele sînt confirmate prin zeci de pagini de documentație. Printr-o astfel de greșeală analiza, și prin ea proiectul, pot fi puse sub semnul întrebării. La acest pas ar trebui să existe disponibil timp suficient, care să permită și consultarea altor colaboratori, obiectivi.

Printre costurile care apar o singură dată sînt și următoarele:

- dezvoltarea aplicațiilor

Durata de reîntoarcere a capitalului
 Alternativa 1: Aplicația se rulează pe un LAN

An	Costuri de exploatare	Costuri de exploatare după impozitare	Amortizare	Amortizare după impozitare *	Investiție	Flux de capital după impozitare
1991	14000	9240	10000	3400	50000	55840
1992	9450	6237	16000	5440		797
1993	4410	2911	9600	3264		-353
1994	4631	3056	5750	1955		1101
1995	4862	3209	5750	1955		1254
1996	5105	3369	2500	986		2383
Total	42458	28022	50000	17000	50000	61022

Alternativa 2: Aplicația rămîne pe mainframe

An	Costuri de exploatare	Costuri de exploatare după impozitare	Amortizare	Amortizare după impozitare *	Investiție	Flux de capital după impozitare
1991	28000	18480	0	0	0	18480
1992	39900	26334	0	0	0	26334
1993	41895	27651	0	0	0	27651
1994	43990	29033	0	0	0	29033
1995	46189	30485	0	0	0	30485
1996	48499	32009	0	0	0	32009
Total	248473	163992	0	0	0	163992

* Pentru calculul din exemplu s-a presupus un procent de impozitare de 30%. Acest calcul determină perioada de reîntoarcere a capitalului. Și din acest exemplu se poate observa că un LAN este mai avantajos.

- hardware
 - software pentru sistemul de operare
 - cursuri de inițiere
 - realizarea conexiunilor inițiale
 - restructurarea platformei rețelei
 - instalarea etc.
- Printre costurile și economisirile palpabile, și care se repetă, sînt și:
- măsuri de perfecționare pentru utilizatori și manageri
 - întreținerea hard-ului și soft-ului
 - asistența aplicațiilor
 - modificări de cablaj
 - administrație locală și globală, la nivelul întregii întreprinderi
 - siguranță

- suport
 - distribuția soft-ului
 - telecomunicații
 - productivitate etc.
- Anumite costuri și economisiri nu sînt direct măsurabile, darele există totuși:
- flexibilitatea întreprinderii
 - mulțumirea colaboratorilor
 - "imagine" mai dinamică a întreprinderii
 - limite de timp mai puțin strînse pentru cei care au de luat decizii
 - ergonomia locului de muncă
 - administrarea rețelei (și la nivelul secțiilor) etc.
- Indiferent cît de mult ar fi detaliată o analiză, ea este doar atît de bună ca

și cifrele pe care le conține. Nu trebuie uitate costurile pentru colaborări. La multe proiecte acesta este cel mai important factor de costuri. De regulă, colaboratorii costă mult mai mult decît hard-ul și soft-ul. De aceea costurile cu personalul nu trebuie evaluate prea jos.

Deoarece costurile "invizibile" sînt foarte greu de calculat, de multe ori se preferă soluția, ale cărei costuri sînt mai transparente. Dar, de-seori, această soluție nu este alternativa optimă pentru întreprindere. Ar trebui măcar încercată estimarea rezonabilă a valorii costurilor invizibile.

Conducătorii unei întreprinderi acceptă mai ușor un proiect dacă este însoțit de cifre credibile.

Directorii și contabilii ezită de obicei la cumpărarea unor noi cal-

culatoare, deoarece nu li se pare o investiție necesară. Totuși, costul unui PC este doar cît salariul pentru cîteva zile al unui manager. Dacă în ce privește costurile și economisirile "invizibile" se insistă pe creșterea probabilității a productivității, atunci lupta este, de regulă, deja pe jumătate cîștigată.

Analiza costurilor pentru downsizing

Sînt puse față în față ambele alternative, atît soluția LAN cît și soluția mainframe. Costurile se calculează pe baza unor factori de productivitate, care includ costurile colaborărilor, inclusiv salarii și impozite.

Analiza din exemplu este făcută pentru o perioadă de șase ani. În această perioadă hard-ul de LAN se amortizează complet, de multe ori,

Valoarea actuală netă

Alternativa 1: Aplicația se rulează pe un LAN

An	Flux de capital după impozitare	Taxă de scont de 10%
1991	55840	55840
1992	797	725
1993	-353	-292
1994	1101	828
1995	1254	856
1996	2383	1480
Total	61022	59436

Valoarea actuală netă = 59436

Alternativa 2: Aplicația rămîne pe mainframe

An	Flux de capital după impozitare	Taxă de scont de 10%
1991	18480	18480
1992	26334	23940
1993	27651	22852
1994	29033	21813
1995	30485	20821
1996	32009	19875
Total	163992	127782

Valoarea actuală netă = 127782

Acest calcul corelează fluxul de capital după impozitare cu valoarea în timp a banilor.

amortizarea este terminată după trei ani. Costurile cu școlarizarea, cu întreținerea hard-ului și soft-ului au fost calculate pentru fiecare an în parte. Se pornește de la ideea că rata de inflație se ridică la 5% anual. Valoarea lui este inclusă, în exemplu, în costurile de exploatare. Acest lucru este necesar deoarece analiza se bazează pe valoarea actuală a mărcii. Ca și în toate celelalte domenii, costurile vor crește în următorii ani, și de

acest lucru trebuie să se țină seama în analiza costurilor.

Deși deseori utilizate, aceste tabele nu iau în considerare costurile care se pot economisi prin alte soluții. Ele permit doar o comparație a costurilor dintr-un interval de timp. Se pleacă de la premisa că una din alternative este obligatorie (de exemplu facturarea. Este evident că întreprinderea are nevoie pentru aceasta de o soluție PAD). Întrebarea este doar: ca-

re dintre alternative este cea mai rentabilă?

Stabilirea costurilor de exploatare este doar un pas din analiza integrală a costurilor. Cu această ocazie, se adună informații care vor reintra mai târziu în analiză. Cu analiza costurilor de exploatare se obține o primă impresie asupra rentabilității proiectului, dar ea nu reprezintă un instrument de analiză de sine stătător.

După ce au fost determinate costurile pentru

perioada de analizat atît pentru soluția LAN cît și pentru varianta mainframe, ele trebuie evaluate prin intermediul unui program de calcul tabelar. Pentru proiectul dat ca exemplu trebuie stabilite valoarea actuală netă și perioada de reîntoarcere a capitalului. Valorile de bază ale acestui spreadsheet provin din calculul costurilor de exploatare.

Aici apare pentru prima oară amortizarea. Valoarea finală din coloana

Din vocabularul financiar

La fel ca și tehnicienii, și experții în finanțe folosesc propria lor terminologie. Cine dorește să se înțeleagă cu ei, trebuie să le cunoască "limba". Explicăm în continuare cîțiva termeni, folosiți foarte des:

Fluxul de capital

Scopul cel mai important al unei analize a costurilor este de a constata cît de mult influențează un anumit proiect finanțele unei întreprinderi. Toate proiectele cer mai întîi niște investiții, se cheltuie deci niște bani. După realizarea lor se așteaptă ca banii să reintre în întreprindere, sub forma unor reduceri ale costurilor. Trebuie arătat, pentru fiecare interval de timp din plan cîți bani curg și în ce direcție. Aceste mișcări de bani se denumesc flux de capital.

Perioada de analiză

Aici se determină perioada în care proiectul va influența fluxul de capital al întreprinderii. Trebuie luate în seamă ciclurile de viață ale resurselor necesare: de regulă, pentru soft 3 ani, pentru hard 5 ani și pentru cablări 10-15 ani.

Este foarte dificilă o prognozare a fluxului de capital pe o perioadă mai mare de cinci ani, deoarece pot apărea mulți factori neașteptați. Acest lucru este valabil mai ales la proiecte care se bazează pe calculatoare. Are sens alegerea unui interval de timp cuprins între 3 și 5 ani.

Valoarea în timp a banilor sau bani numerar și discount

Atunci cînd se împrumută 100 de mărci nimănui nu-i este indiferent, dacă primește banii mîine înapoi sau abia peste un an. Dacă banii lipsesc un an, ei nu pot fi cheltuiți sau nu pot fi depuși la bancă ca să aducă o anumită dobîndă. Valoarea în timp a banilor se bazează pe principiul că "banii în mînă" valorează mai mult, decît banii care se vor obține în viitor.

Dacă, de exemplu costurile unui proiect însumează 10 mărci, și se așteaptă economisiri de 5 mărci în fiecare din următorii 2 ani, cele două valori se compensează. Din punctul de vedere al experților financiari însă, în acest caz este vorba despre un minus. Acest factor se numește

discount. Cu ajutorul lui se calculează valoarea în numerar a banilor la o dată ulterioară, în exemplul nostru fiind vorba de un an, respectiv de 2 ani.

Costurile de capital și dobînda

Noțiunile rată de discount și costuri de capital sînt interschimbabile. Costurile de capital se calculează din modul în care se finanțează o întreprindere. De obicei, acesta este o combinație între acțiuni, venituri și cheltuieli pentru bani împrumutați.

Acest calcul nu trebuie făcut de către administratorul analizei costurilor. El poate afla rata corespunzătoare de discount de la contabilitate.

Dobînda este în principiu un procentaj la împrumut. Ea se referă la reîntoarcerea capitalului din investiția inițială și la economisiri datorate proiectului respectiv în perioada de analiză.

Impozite

Statul obține cca. 50% din veniturile întreprinderii. Astfel, fiscul devine "partenerul" cel mai important al întreprinderii. Este însă posibilă scăderea din impozite

a costurilor unui proiect aducător de venituri ca și a cheltuielilor întreprinderii. Este și acesta un factor important al analizei costurilor.

Amortizarea

Se investește atît în calculatoare cît și în alte resurse ale întreprinderii. Un 486, care rezolvă o anumită problemă, nu va rămîne pe veci în exploatare. Odată și o odată tot vine rîndul unei noi investiții.

Legea permite scăderea costurilor din impozit pe o anumită perioadă de timp. Se presupune că aceste aparate sînt "consumabile" și în plus duc pe termen lung la venituri crescute, impozabile. Costurile de amortizare reduc însă pe moment venitul total impozabil. Cu cît amortizarea este mai mare, cu atît impozitele ce sînt de plătit sînt mai mici.

De cele mai multe ori calculul amortizării este foarte ușor de efectuat. Programele de calcul tabelar conțin de regulă funcții corespunzătoare, care calculează automat amortizarea corectă. Și în acest caz contabilul devine partener: el poate explica cum este aplicată amortizarea de către întreprindere.

"fluxul de capital după impozitare" se calculează din costurile de exploatare după impozitare la care se adaugă investițiile în calculatoare și se scade amortizarea după impozitare. Dacă se folosește pentru asemenea analize un program de calcul tabelar uzual cum ar fi "Lotus 1-2-3", calculul se face automat. Doar amortizarea și investiția trebuie introduse.

Valoarea actuală, reîntoarcerea capitalului și dobânzile

Pasul următor este calcularea valorii actuale. Și în acest caz, programele de calcul tabelar dispun de funcții proprii. Aceste formule pot fi elaborate și individual, dar este mult mai complicat, și în cele mai multe cazuri, se va evita acest lucru. Urmează calculul VAN (valorii actuale nete), care se obține scăzând taxa de scânt (în exemplu ea este de 10%).

Pentru exemplul de mai sus s-a calculat VAN de 59.436 mărci. VAN pentru soluția mainframe este de 127.782 mărci. Deci alternativa LAN este cu 68.346 mărci mai ieftină. În mod concret rezultă de aici că proiectul preconizat este mult mai rentabil dacă funcționează pe LAN decât pe un calculator mare, cu toate că acesta există deja în întreprindere.

În acest punct, analiza ar fi de fapt terminată. Calculația valorii actuală este atât de clară încât aduce destule argumente pentru proiect. Totuși uneori se dorește și efectuarea unei analize a reîntoarcerii capitalului. Se pune întrebarea: cât timp va dura până când investiția va intra din nou în întreprindere? Intrarea

Dobînda limită

An	Flux de capital LAN după impozitare	Flux de capital mainframe după impozitare	Diferența	Taxa de scânt 69%
1991	55840	18480	-37360	-37360
1992	797	26334	25537	15292
1993	-353	27651	28004	10041
1994	1101	29033	27932	5997
1995	1254	30485	29231	3758
1996	2383	32009	29626	2281
Total	61022	163992	102970	9

Dobînda limită corectă se calculează cu o formulă. Această valoare oferă informații despre taxa de scânt, în cazul nostru ea fiind de 67%

capitalului rezultă din calculul economisirilor în cadrul costurilor generale ale unui LAN față de cele ale unui mainframe.

Acest calcul este foarte simplu. Trebuie doar scăzute costurile anuale pentru LAN din cele corespunzătoare pentru mainframe. În exemplul nostru se vede că în primul an diferența de 37.360 mărci este în favoarea mainframe-ului. Motivul este investiția de cumpărare a LAN-ului, care este de 50.000 de mărci. În următorii ani balanța se înclină în favoarea LAN rețeaua devenind mult mai avantajoasă decât mainframe-ul. În acest caz, capitalul se reîntoarce în întregime după doi ani, iar economiile aduse în următorii ani aduc argumente inatacabile în favoarea trecerii la o rețea locală.

În unele cazuri, în locul sau pe lângă calculul VAN se calculează "dobînda". Trebuie găsită o dobînda limită care duce la o valoare nulă de capital, deci numerarul de venituri și cheltuieli să fie la fel de mare. Această dobînda procentuală internă se determină cu o formulă cunoscută în contabilitate

. Astfel se obține plata efectivă a dobînzilor unui proiect de investiție înainte de scăderea plății dobînzilor. În proiectul din exemplu valoarea finală a acestui calcul este de 9 mărci și deci foarte aproape de valoarea ideală 0. Această valoare a fost atinsă prin aplicarea a diferite dobînzii limită. Ce ne spune o dobînda limită de 67%? Dacă întreprinderea trebuie să împrumute capitalul necesar de la bancă la o dobînda de 15%, înseamnă că dobînda limită minimă trebuie să fie de cel puțin 15%. Deoarece ea a fost stabilită la 67% înseamnă că proiectul va fi cu siguranță acceptat.

Proiectele care sînt calculate la o dobînda limită ridicată sînt puse la îndoială de contabili. Ei sînt înclinați să verifice în amănunt aceste cifre. De aceea, analizele care prezintă dobînzii limită ridicate, nu vor fi prezentate în această formă.

Documentarea succesului

Analiza costurilor a convins atât conducerea cât și contabilii întreprinderii. Proiectul a fost re-

alizat și s-a dovedit eficient și productiv. Nu trebuie uitată acum documentarea succesului, cu toate economiile și avantajele lui. Mai ales proiectele "downsizing" provoacă neliniște, deoarece pentru întreprindere ele înseamnă mari modificări organizatorice.

Chiar dacă proiectul întrece toate așteptările, vor exista mereu persoane care nu se vor putea acomoda cu noile structuri organizatorice. Pentru a veni în întîmpinarea argumentelor lor, este utilă elaborarea unei documentații cuprinzătoare.

Este indicat ca de la bun început contabilitatea să fie implicată în proiect. Oricum la început sînt necesare informații în legătură cu amortizarea, impozitele și fluxul capitalului; informații ce pot fi obținute numai de la acești specialiști.

În afară de aceasta, se poate pleca de la premisa că aceștia vor efectua o analiză de control a proiectului după ce acesta a fost pus în practică. Ei sînt cu siguranță mai maleabili dacă sînt implicați de la bun început și dacă trebuie să controleze propriile lor cifre.

Chiar dacă, în mod normal, contabilitatea va prelua cea mai mare parte a muncii pentru analiza costurilor, este important să se cunoască măcar în mare procedeele. Astfel se pot înțelege cifrele și motivele care determină o anumită decizie. Dacă analiza este făcută independent și li se prezintă experților financiari în "limba" lor, se poate conta pe acceptarea proiectului.

(I.M)

Servere de baze de date

În cele ce urmează vom încerca o trecere în revistă a SGBD-urilor care implementează modelul client-server în rețele locale de calculatoare (Local Area Network). Produsele care urmează a fi prezentate fac parte din categoria serverelor de baze de date (database server) și sînt proiectate pentru a rula pe un LAN-server uzual, înzestrat cu un sistem de operare în rețea solid și pentru a furniza servicii de gestiune a datelor stațiilor de lucru-client.

Serverele de baze de date incluse în rețele locale de calculatoare au început să preia sarcini odinioară îndeplinite numai de SGBD-uri operînd pe minicalculatoare sau sisteme mari, inclusiv aplicații on-line de prelucrare a tranzacțiilor. De asemenea, serverele de baze de date permit accesarea partajată a informației, sigură și simultană, unei largi varietăți de aplicații utilizate în mod curent și posedînd o interfață-utilizator prietenoasă cum sînt, de pildă, aplicațiile de calcul tabelar sau editoarele de texte.

Softul de server de baze de date destinat microcalculatoarelor rulează - după cum lesne se poate închipui - pe serverul rețelei. Cînd o aplicație-client are nevoie de regăsirea sau actualizarea unei informații formulează o solicitare, de obicei în SQL (Structured Query Language), către serverul de baze de date. Acesta prelucrează cererea, regăsește și actualizează înregistrările și, fie că le returnează pe acestea, fie că un cod de stare este returnat aplicației.

În ceea ce privește mai răspînditul model al serverului de fișiere (file server), se poate spune că acesta folosește serverul rețelei ca pe o unitate de hard disk și nimic mai mult. În consecință, fișiere întregi uneori sînt transmise prin rețea pentru a fi prelucrate de stația de lucru-client. Cu alte cuvinte, transferul de date care are loc, într-un SGBD monoutilizator, între unitatea de disc fix și UC, pe un bus relativ larg, trebuie să se desfășoare, în modelul serverului de fișiere, pe mediul de comunicare al rețelei, evident mai strîmt și, în plus, folosit simultan și de alți clienți din rețea. Modelul client-server evită acest neajuns.

Însă serverele de baze de date pot face mai mult decît să descongeseze mediul de comunicare al rețelei. Caracteristici specifice le permit să asigure integritatea datelor (în unele cazuri, chiar integritatea referențială a acestora), să sprijine gestiunea tranzacțiilor și să recupereze toate tranzacțiile în situația unei defecțiuni a sistemului. Unele servere de baze de

date prezintă extensii către limbajul SQL care le oferă posibilitatea de a impune la nivel central anumite condiții sau reguli cum ar fi, de pildă: "nu accep-ta o comandă de la un client insolubil" sau "adaugă la valoarea mărții 5%, re-prezentînd ICM-ul".

Nu este necesară implementarea acestor reguli în fiecare aplicație care accedează datele, fiind suficientă o singură implementare. Dacă o regulă se schimbă, de exemplu dacă ICM-ul crește la 10% atunci ne putem consola cu faptul că modificarea corespunzătoare trebuie operată într-un singur loc. Calculatoarele-client care vin în contact cu o arhitectură client-server prin intermediul unui soft front-end au avantajul că nu trebuie să îndeplinească toate funcțiile de prelucrare a datelor, o parte dintre acestea revenind serverului de baze de date. Astfel, calculatoarele-client mai puțin performante se comportă mai bine decît ar putea să o facă într-un model "file server". Și un alt avantaj major al arhitecturilor client-server: problema creșterii performanțelor întregului sistem se reduce la creșterea performanțelor calculatorului pe care lucrează serverul de baze de date în timp ce o operație similară asupra unui server de fișiere nu reprezintă o soluție, atît timp cît ea nu se extinde și asupra stațiilor de lucru.

Înainte de a trece la prezentarea succintă a citorva dintre cele mai cunoscute servere de baze de date, subliniem rolul important pe care îl au în arhitecturile client-server instrumentele front-end, de legătură între serverul de baze de date (back-end) și utilizator. Cîteva dintre cele mai răspîndite asemenea instrumente sînt: Lotus 1-2-3, Paradox, dBASE IV, DataEase, ObjectVision, R&R Report Writer, SQL Windows ș.a. În final, două mențiuni: între paranteze este indicat producătorul iar prețurile sînt informative.

Emerald Bay 2.27 (RSPI)

Emerald Bay este un SGBD multiutilizator, destinat în special programatorilor, care oferă o arhitectură client-server pentru PC-uri. Vulcan, o adaptare a limbajului Xbase pentru Emerald Bay, oferă un interpretor, un compilator EXE și facilități care privesc funcții definite de utilizator, operarea cu ajutorul mouse-ului, salvarea și restaurarea imaginilor ecranului și tablouri multidimensionale. C Developer's Toolkit, o bibliotecă de funcții C care accedează bazele de date Emerald Bay dintr-un program C, conține biblioteci

de imagini ecran care permit includerea formatelor de ecran generate cu Forms Design Utility în programe scrise în C. Și pentru Pascal a fost prevăzută o "trusă", Pascal Developer's Toolkit cu ajutorul căreia se pot accesa baze de date EB din programe Pascal. Fiecare produs sau "trusă" Emerald Bay include un Engine pentru prelucrarea bazelor de date EB din programe scrise în C sau Pascal, un Database Administrator, funcții de import/export (Rosetta), editoare de machete de ecran (Forms Designer/Screen Painter), un editor de rapoarte (Report Writer), unul de texte (Text Editor) și o versiune minimală, pentru 2 utilizatori, a serverului. Vulcan și C Developer's Toolkit costă cîte 395\$ iar Pascal Developer's Toolkit, 199\$. Prețul serverului EB este de 199\$ pentru 4 utilizatori, 399\$ pentru 8 și de 599\$ pentru un număr nelimitat de utilizatori.

Ingres for OS/2, DOS (Ingres)

Ingres Server for OS/2 și Ingres Tools for DOS sînt instrumente care realizează conectarea PC-urilor-client la SGBD-ul relațional Ingres rufînd pe un server sub UNIX sau VAX/VMS, prin intermediul sistemelor de operare în rețea LAN Manager (Microsoft), NetWare (Novell) sau 3+ Open (3Com). Acestea lucrează pe PC-uri unde, după ce traduc cererea în limbaj SQL, o transmit prin rețea către server. Acesta optimizează interogarea și efectuează gestiunea datelor, returnînd sau memorînd informația corespunzătoare. Ingres Server for OS/2 se vinde la prețul de 1.995\$.

Microsoft SQL Server 1.1 (Microsoft)

Microsoft SQL Server este un sistem de gestiune a bazelor de date relaționale (SGBDR) pentru rețele de PC-uri care oferă o gestiune centralizată a datelor, cu accent pe securitatea și integritatea acestora precum și pe controlul utilizatorilor. Este sprijinit de o gamă largă de aplicații front-end. SQL Server, împreună cu LAN Manager și DCA Microsoft Comm Server alcătuiesc o familie de produse Microsoft proiectată să rezolve orice problemă care poate fi ridicată de funcționarea unei rețele de calculatoare. Probabil că în funcție de numărul de probleme rezolvate variază și prețul: între 995 și 3995\$.

NetWare SQL 2.11 (Novell)

NetWare SQL 386 este implementat ca un modul încărcabil NetWare (Net-

Ware Loadable Module) ceea ce înseamnă că este puternic integrat în sistemul de operare în rețea NetWare 386, spre deosebire de un VAP (Value Added Process) care rulează "deasupra" sistemului de operare, fără să aibă vreo interferență cu acesta. Acest fapt îl face mai robust și mai performant. Poate executa simultan prelucrări vizând mai multe baze de date. Aplicațiile front-end pot acceda ambele versiuni ale sistemului, 286 și 386, fără nici o modificare. NetWare SQL folosește protocoalele NetWare uzuale (Internetwork Packet Exchange și Sequenced Packet Exchange) pentru a acceda resursele serverului. Funcțiile de securitate și de administrare a bazei de date sînt centralizate la nivel de server. Cu toate că NetWare SQL respectă prevederile referitoare la comenzi SQL, conținute în standardul ANSI Level 1, oferă și funcții suplimentare.

Oracle Server 6.0 (Oracle Corp.)

Oracle Server permite rularea de aplicații multiutilizator în arhitecturi client-server sau pe minisisteme și utilizarea simultană a aceleiași baze de date. Utilizatorii pot accesa datele direct de la stații de lucru DOS, OS/2, UNIX sau Macintosh. Poate folosi atât protocoalele SPX/IPX pentru conectarea rețelelor funcționînd sub Novell NetWare cît și protocolul TCP/IP pentru UNIX, precum și protocoale asincrone. Realizează prelucrarea tranzacțiilor și recuperarea automată a sistemului. Versiuni pentru OS/2, SCO UNIX, XENIX și AT&T V/386. Preț: 51.00\$ pentru 8 utilizatori în configurații 386/486.

SQLBase Server 4.1 (Gupta Technologies)

SQLBase este un server de baze de date care furnizează o implementare completă de SQL rulînd atît sub DOS cît și sub OS/2. Ambele versiuni transformă un PC/AT într-un server de baze

de date care poate fi accesat de mai multe PC-uri integrate într-o rețea compatibilă NetBios. De asemenea, ambele includ o interfață de programare în C a aplicațiilor pentru Microsoft Windows și sînt în curs de elaborare instrumente pentru OS/2 Presentation Manager. Setul de dezvoltare multiutilizator costă 1995\$/server iar versiunea monoutilizator, 955\$. Instrumentul front-end SQLWindows, produs tot de Gupta Technologies, conține un editor grafic de formate ecran tip Microsoft Windows. Limbajul procedural intern SQLWindows, numit SAL, poate include instrucțiuni SQL imbricate pentru accesarea bazei de date, poate utiliza obiecte create cu editorul de formate și este sprijinit de un editor intern SQLWindows. SQLNetwork, al aceleiași firme, permite conexiuni transparente între aplicațiile SQLBase și alte baze de date SQL existente pe minicalculatoare sau sisteme mari.

(R.G.)

Visual Basic - programare intuitivă

(continuare din pag. 24)

Între aceste 2 linii trebuie inserate instrucțiunile ce vor trebui executate atunci cînd se tastează pe mouse. Un lucru e clar: Visual Basic este, spre deosebire de alte produse ca Object Vision al firmei Borland sau Toolbook al firmei Asymetrix, gîndit, înainte de toate, ca un soft pentru că, în orice caz trebuie făcute programe. Ceea ce înseamnă că, în ciuda existenței unui "Help" foarte bun, cu multe exemple, este absolut necesară cunoașterea instrucțiunilor, lucru ce nu poate fi cerut unui profan.

În ajutorul programatorilor

Fereastra de instrucțiuni, (de cod), arată ca un editor de texte. Pe marginea superioară sînt 2 rubrici. Cea din stînga conține o listă a controalelor formularului activ. Din această listă se poate alege cărui control să i se asocieze mecanismul de desfășurare al unui eveniment. În cea din dreapta apar procedurile care, pentru controlul ales, pot fi sau sînt deja programate. Cînd nu este absolut necesar să se cuprîndă toate evenimentele în pro-

gram, evenimentele codificate deja vor fi vizualizate.

Astfel, Visual Basic, spre deosebire de Windows, îi economisește programatorului o mulțime de timp: el nu trebuie să se mai ocupe de mulțimea de mesaje pe care nucleul Windows le trimite programului activ, și el se poate concentra asupra interacțiunii cu utilizatorul, restul rezolvă Visual Basic în locul lui.

La introducerea liniilor de program, Visual Basic vechiază asupra corectitudinii sintaxei. Astfel, greșelile de tastare sînt recunoscute imediat și pot fi corectate. Acest mod de lucru este cunoscut deja din Quick-Basic și se bazează pe transformarea codului sursă într-un cod intermediar (Threaded-P-Code). Acesta nu corespunde formei finale a programului, dar poate fi oricînd lansat din Visual Basic. Execuția codului P nu este așa de rapidă ca cea a codului final, dar programul poate fi oricînd testat sub această formă și eventualele greșeli pot fi eliminate foarte repede fără a mai pierde timp cu compilarea.

Pentru depanarea programelor, Visual Basic oferă tot ajutorul ce se așteaptă de la un limbaj modern de programare. Astfel, pot fi fixate puncte de întrerupere (Breakpoint) care provoacă întreruperea programului și permit vizualizarea și chiar modificarea conținutului unor variabile. Programul poate fi modificat și în timpul unei întreruperi și apoi se poate continua rularea lui. Deci, subliniem, rularea programului nu trebuie reluată de la început în cazul efectuării unor modificări. După ce programul a fost testat "la sînge" el poate fi compilat cu compilatorul Visual Basic-ului rezultatul fiind un fișier EXE. Programul executabil astfel obținut poate fi rulat independent de Visual Basic și poate fi vîndut sau dat fără taxă de licență.

Dezvoltări utile în perspectivă

Prin apariția Visual Basic-ului, Microsoftul a făcut un salt în viitor. Odată cu apariția acestui pachet programarea aplicațiilor Windows nu se mai limitează la cercul programatorilor ce sînt familiarizați cu Windows - API. Întrucît

Visual Basic permite și schimbul dinamic de date (DDE), el devine o platformă potrivită pentru integrarea aplicațiilor existente. Nu trebuie renunțat sub nici o formă la rutinele C sau assembler existente; ele pot fi folosite prin intermediul DLL-ului (Dynamic Link Libraries). Premisa reușitei este bineînțeles un compilator corespunzător, altfel programele existente nu pot fi rulate sub Visual Basic.

În curînd Microsoft va scoate și un Control Development Kit care va face posibilă definirea proprie a unor controale folosibile apoi în formulare. Visual Basic-ul funcționează pe orice calculator pe care poate funcționa Windows-ul. Totuși, se recomandă următoarea configurație minimă: un calculator 386SX (16MHz) cu o memorie internă de 2 MBytes, un hard-disk de 40 MBytes, un monitor VGA și un mouse. Prețul acestui pachet este de 600 DM.

(Radovici Vera)

Sistemul de gestiune al fișierelor MS-DOS

Atît în cazul intrărilor, fie ele de la tastatură sau de la mouse, în cazul ieșirilor, fie la imprimantă sau pe ecran, cît și în cazul multor altor operații, programatorii se comportă de multe ori ca și cum nu ar exista MS-DOS. Ei folosesc BIOS-ul sau adresează direct hardware-ul. Un singur domeniu al programării rămîne întotdeauna în atenția sistemului de operare: manipularea fișierelor și a directoarelor se face doar prin intermediul funcțiilor MS-DOS. Există motive întemeiate în acest sens.

Fișierele și directoarele formează partea vizibilă a unei componente complexe și foarte importante a MS-DOS și anume sistemul de gestiune al fișierelor (S.G.F.). Acesta este cel care dă dischetelor și harddisk-urilor structura lor și face posibil accesul foarte simplu al utilizatorilor de PC-uri la conținutul acestor purtători de date. Dacă vă gîndiți care sînt acțiunile care se repetă zilnic și care depind de accesul la fișiere, atunci veți ajunge la concluzia că sistemul de operare MS-DOS nu s-ar putea închipui fără S.G.F.

Închipuiți-vă de exemplu, ce se întîmplă cînd apelați un program. În primul rînd MS-DOS va căuta programul (el citește directoarele căutînd în conținutul numelui programului). Dacă l-a găsit, atunci îl va încărca (se citește fișierul program și conținutul va fi copiat în memoria de lucru). În fine, MS-DOS lansează programul, care în cele mai multe cazuri execută din nou accesarea asupra fișierelor, pentru a citi datele necesare funcționării sale. Fiecare dintre aceste accesări indiferent că ele sînt lansate de MS-DOS sau de alt program, sînt derulate de S.G.F. Vedeți deci că fără S.G.F. nimic nu merge.

S.G.F. stă ca bază pentru structura fizică a dischetelor sau a hard-

disk-urilor. Deoarece aceste două tipuri de memorii de masă sînt structurate aproape identic, în continuare vom utiliza noțiunea *disc*, atunci cînd deosebirile dintre dischetă și harddisk sînt nesemnificative. Structura fizică a acestor medii de memorare este cît se poate de simplă. Întregul disc este divizat în domenii de aceeași dimensiune, denumite *sectoare*. Sectoarele au aceeași dimensiune și se deosebesc între ele doar prin poziția fizică de pe disc. Din această cauză ele mai sînt denumite și *sectoare fizice*. Un sector poate fi identificat în două moduri. Controller-ul harddisk-ului utilizează adresa fizică a fiecărui sector, în timp ce MS-DOS procedează mai simplu: sectoarele sînt numerotate crescător începînd de la 0. La această numerotare se face referire atunci cînd

este vorba despre *numărul sectorului*.

Structura prezentată este produsă în urma procesului de *formatare*. Din păcate, în cazul sistemului de operare MS-DOS, această noțiune are două sensuri, și aceasta din cauza unor motive istorice: atunci cînd a fost creat sistemul de operare MS-DOS, PC-urile nu dispuneau decît de lectoare de dischete. Pentru ca o dischetă să poată fi utilizată pentru scopul pentru care era destinată, deci ca suport de date, a fost creată comanda *format*, care rezolva două probleme. Pe de o parte această comandă realiza împărțirea dischetei în sectoare, iar pe de altă parte, comanda *format* crea și structura logică de date de care avea nevoie MS-DOS pentru a putea memora date pe dischetă. Și în mod natural aceste două ope-

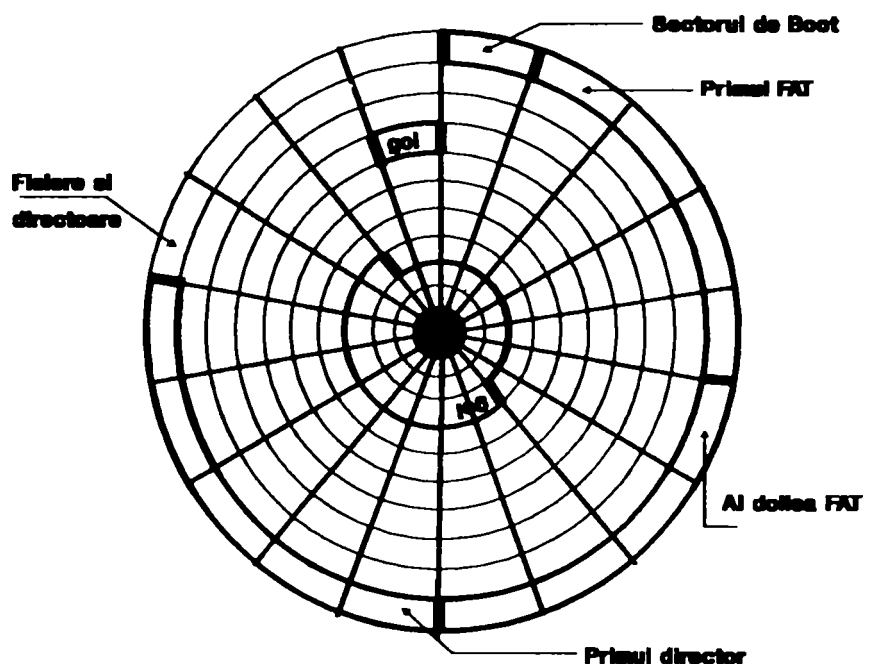


Fig. 1: Vedere de ansamblu asupra structurii de date create de sistemul de operare MS-DOS în urma formatarei pe un disc

D	O	S	S	W	A	P	E	X	E										
0							8			22	24	26	28						
Nume fisier (8 Byte)							Extensie (3 Byte)			Atribut (1 Byte) neutilizabili (10 Byte)			Data (2 Byte)	Țimpul (2 Byte)	Primul cluster (2 Byte)	Dimensiune (4 Byte)			

Fig. 2: Exemplu de intrare într-un director (în acest caz pentru fișierul DOSSWAP.EXE)

rații au fost cuprinse în aceeași noțiune de *formatare*.

De la apariția primului harddisk situația s-a schimbat. Împărțirea harddisk-ului în sectoare este realizată, de regulă, de producător, încă înainte de livrare. În sarcina utilizatorului mai rămâne așadar doar crearea structurii logice pentru MS-DOS pentru a putea utiliza harddisk-ul. Această operație este denumită *formatare*, nu în ultimul rând, deoarece comanda cu care este realizată este denumită, ca și înainte *format*. Dar cum s-ar denumi atunci prima operație de pregătire? Producătorii îi spun - cum este și de așteptat - *formatare*.

Pentru a mai atenua confuzia împărțirea în sectoare mai este denumită adesea și *formatare low-level*, sau *formatare fizică*. Deoarece comanda MS-DOS *format* nu poate realiza această operație, atunci când unitatea desemnată este un harddisk, pentru a o realiza este necesar un program special sau o procedură BIOS. În orice caz nici una nici cealaltă metodă nu sînt normale. Din fericire această operație trebuie executată foarte rar de utilizatori, majoritatea dintre ei nefiind puși practic niciodată în situația de a-și formata fizic harddisk-ul.

Dacă împărțirea în sectoare nu este realizată cu comanda *format*, este de presupus că această comandă realizează "celălalt fel" de *formatare*, creînd structurii logice

necesară pentru sistemul de operare MS-DOS. Tocmai acesta este cazul. Structura prezentată în fig. 1 constituie baza gestiunii fișierelor pe un disc. Aproape auxiliară este posibilitatea ca sistemul de operare MS-DOS să se autoîncarce de pe disc.

Un sector important

Cheia întregului disc se află în primul sector. Acest sector este denumit *sectorul de boot*, denumire care vine de la termenul englezesc *to boot* care desemnează pornirea sistemului de operare. (De aici și barbarismele *boot-area*, *boot-abil* ș.a.m.d.). Dar chiar și în cazul în care discul nu conține nici un sistem de operare, și deci calculatorul nu se poate "boot-a" de pe el, acest sector este foarte important.

Motivul pentru care acest sector este atît de important poate fi observat din tabela 1. Toate datele care descriu formatul discului, și cu ajutorul cărora sistemul de operare MS-DOS poate utiliza discul, sînt stocate în *sectorul de boot*. Datele se ascund în spatele prescurtării *DPB* (Disk Parameter Block). Este vorba despre descrierea structurii logice a discului. Numai cu ajutorul acestor date sistemul de operare poate găsi *primul director* de pe disc, și numai prin accederea primului director este în stare MS-DOS să găsească fișierele. După

cum se poate observa, sectorul de boot este cheia întregului disc.

În sectorul de boot se găsesc însă pe lîngă *DPB*, și alte date importante. Importantă este și mica procedură de încărcare, ce servește la pornirea sistemului de operare. Pe harddisk-uri mai există în sectorul de boot și o altă structură de date, *tabela de partiționare*. Dacă ați utilizat un harddisk cu capacitatea mai mare de 32 MByte, cu un sistem de operare anterior lui MS-DOS versiunea 4.0, cuvîntul *partiție* desemnează cu siguranță o noțiune cunoscută.

Cu versiunile mai vechi de MS-DOS, un harddisk nu putea avea o capacitate de memorare mai mare de 32 MByte. Dar întru-cît multe harddisk-uri sînt mai mari, MS-DOS oferă o cale ocolitoare pentru a putea utiliza întreaga capacitate a discului. Harddisk-ul este împărțit

Adresa	Lungime	Conținut
0	3	Comandă de salt la începutul procedurii de boot
3	8	N u m e l e producătorului și numărul de versiune
11	2	Numărul de bytes dintr-un sector
13	1	Numărul de sectoare dintr-un cluster
14	2	Numărul de sectoare rezervate
16	1	Numărul de FAT-uri
17	2	Numărul de intrări în primul director
19	2	Numărul de sectoare de pe disc
21	1	Tipul discului
22	2	Numărul de sectoare din FAT
24	2	Numărul de sectoare per pistă
26	2	Numărul capetelor de scriere/citire
28	2	Numărul de sectoare ascunse
30	416	Procedura de boot
446	16	Tabela de partiționare (numai la harddisk-uri)
462	50	Extensie la procedura de boot

Tabela 1: Structura sectorului de boot

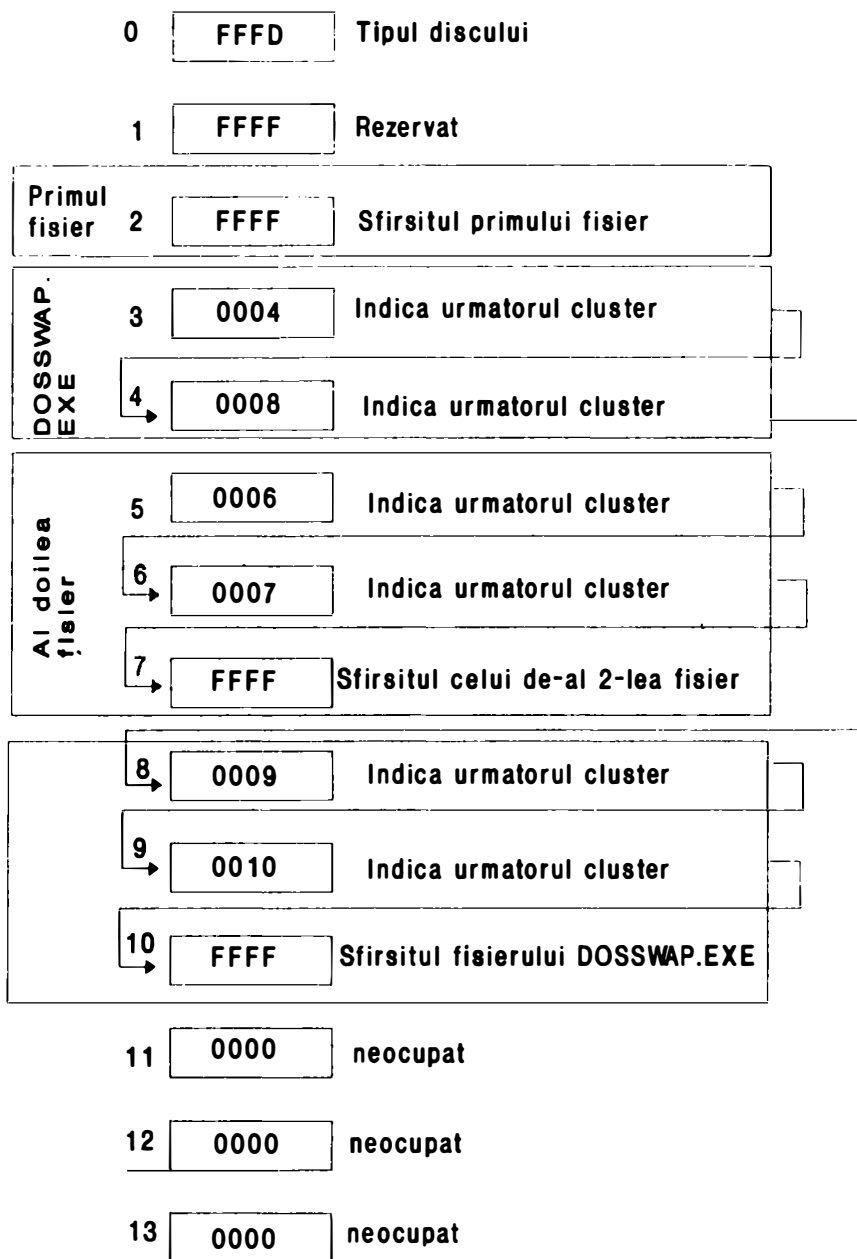


Fig. 3: Exemplu de structură a unui FAT (File Allocation Table)

în mai multe unități logice, fiecare unitate putînd avea capacitatea maximă de 32 MByte. Împărțirea unui harddisk în unități logice este denumită *partiționare*. În tabela 2 este prezentată structura de date, din sectorul de boot, care stabilește dimensiunea și poziția partițiilor.

S-a putut observa cum este împărțit un harddisk și cum poate găsi sistemul de operare primul director. Acest prim director este denumit adesea și *root* (rădăcină) deoarece constituie baza structurii arborescente a directorilor de pe

disc. Legăturile dintre directoare și dintre directoare și fișiere sînt conținute în cuprinsul directorilor. Fig. 2 arată cum este construită o astfel de intrare în director.

La începutul intrării se află numele directorului, respectiv al fișierului. Acest nume este constituit din 11 caractere, fără punct între nume și extensie. După nume urmează așa-numitul *attribute byte*, care conține o informație importantă: cu ajutorul unui bit din acest byte se stabilește dacă intrarea reprezintă un director sau un fișier.

Tot în această intrare sînt stocate date referitoare la dimensiunea fișierului și la data și ora creării. Aceste date nu sînt necesare S.G.F.-ului, ele le memorează dar nu le utilizează. Dimpotrivă, cîmpul *Primul Cluster* este deosebit de important pentru S.G.F. MS-DOS.

Un *cluster* este o unitate logică care constă dintr-unul sau mai multe sectoare fizice învecinate. Aceasta înseamnă că sectoarele sînt grupate în mici entități denumite *cluster*. În mod obișnuit un cluster conține un sector pe o dischetă și patru sectoare pe un harddisk.

Există două motive care justifică această grupare: pe de o parte timpul este mai eficient utilizat dacă S.G.F. scrie mai multe sectoare odată. Acest mod de lucru conduce însă la o risipă de spațiu de memorare. Astfel, de exemplu, un fișier cu dimensiunea de doar 10 Byte ocupă un cluster, și deci 4 sectoare.

Al doilea motiv care justifică gruparea cîntărește mai greu: MS-DOS nu poate trata mai mult de cca. 65.000 de unități (sectoare, respectiv *cluster*) pe un harddisk. Cu aceasta - la o dimensiune tipică de sector de 512 Byte dimensiunea maximă a unui harddisk este limitată la aproximativ 32 MByte. Dacă sectoarele sînt grupate în unități mai mari, atunci se extinde și această valoare limită. Cu patru sectoare per cluster dimensiunea maximă a unui harddisk devine 128 MByte. Această limită era neînchipuit de mare în urmă cu cîțiva ani, astăzi însă ea poate fi atinsă cu ușurință. Sîntem curioși dacă noile versiuni MS-DOS vor ridica această limită.

Dar să ne întoarcem la cîmpul *Primul Cluster*. Valoarea acestui cîmp stabilește poziția de început pe harddisk a fișierului. Cutoate că de regulă un fișier ocupă mai multe *cluster*-e, în director nu există memorat decît primul cluster. Cum se regăsesc celelalte *cluster*-e ale fișierului?

Răspunsul este dat de structura de date concepută special pentru acest scop. Ea este denumită *File Allocation Table* (tabela de alocare a fișierelor), mai scurt *FAT*. Această structură este atât de importantă încît pe fiecare disc ea este stocată în dublu exemplar.

Sistemul de operare MS-DOS se îngrijește ca ambele copii să se găsească în aceeași stare, fapt care poate fi verificat cu comanda *Chkdsk*. În sarcina acestui program cade verificarea conținutului celor două FAT-uri, atît una față de cealaltă cît și în sine. Mesajele de eroare date de *Chkdsk* relevă date inconsistente, sau eronate, în una sau în ambele copii FAT.

În fig. 3 este dat un mic exemplu al unui FAT. Așa cum se poate observa, pentru fiecare cluster există cîte o intrare în FAT. Numărul cluster-ului servește ca index în FAT. În funcție de faptul dacă, și cum, un cluster este ocupat, intrarea ia diferite valori.

Valoarea 0 desemnează un cluster liber, care poate fi utilizat la crearea unui nou fișier, respectiv a unui nou director. Valorile de la FFF0h pînă la FFF6h sînt rezervate, iar valoarea FFF7h desemnează un cluster defect, în care nu pot fi memorate date. Valorile de la FFF8h pînă la FFFFh specifică faptul că este vorba despre ultimul cluster al unui fișier. Toate celelalte valori au o semnificație simplă: conțin numărul următorului cluster al fișierului.

Dacă urmăriți intrarea în director din fig. 2, veți observa că în cîmpul Primul Cluster este memorată valoarea 0003h. În a treia intrare în FAT (fig.3) veți găsi valoarea 0004h. Aceasta înseamnă că al doilea cluster al fișierului *dos-swap.exe* este cluster-ului numărul 4. În a patra intrare în FAT veți găsi valoarea 0008h, deci cluster-ul numărul 8 este al treilea cluster al fișierului. FAT-ul este parcurs în continuare în acest mod pînă cînd este întîlnită valoarea FFFFh. Aceasta înseamnă că ultimul cluster al fișierului este cluster-ul

numărul 10. Deci fișierul *dos-swap.exe* constă din cinci clustere, cu numerele 3, 4, 8, 9 și 10, în chiar această ordine.

Acum se poate trage o concluzie definitivă asupra modului în care MS-DOS citește un fișier. Mai întîi sistemul de operare caută în director o intrare care să conțină numele fișierului dorit. Din această intrare este citit numărul primului cluster. Apoi este citită tabela de alocare a fișierelor (FAT). Numărul primului cluster este utilizat ca index pentru citirea unei valori din tabelă. Dacă această valoare este mai mare decît FFF7h înseamnă că fișierul a fost citit în întregime. Altfel valoarea citită dă numărul următorului cluster care trebuie citit. Acest număr va servi din nou ca index pentru următoarea valoare din FAT, și se continuă pînă cînd este găsit ultimul cluster.

Caracteristici speciale

S.G.F. MS-DOS are două caracteristici speciale: un fișier poate fi șters foarte ușor și poate fi recuperat la fel de ușor. Pentru aceasta nu trebuie modificate decît FAT-ul și intrarea în director. Datele propriuzise nu sînt distruse. Pentru a restaura un fișier șters nu trebuie decît să se găsească vechea intrare în director. Numărul primului cluster se află încă nemodificat în această intrare, cu toate că intrarea este marcată ca "ștearsă".

Pornind de la acest prim cluster, cu ajutorul FAT-ului pot fi regăsite toate celelalte cluster-e. Cu singura condiție ca după ștergere niciunul dintre aceste cluster-e să nu fi fost alocat altui fișier. Aceasta este tehnica utilizată de utilitarele care recuperează fișierele șterse.

A doua caracteristică a sistemului de fișiere este mai puțin avantajoasă. Deoarece datele nu trebuie stocate neapărat grupat, în anumite condiții, ele sînt "împrăștiate" în cluster-ele de pe disc. Citirea unui astfel de fișier necesită relativ mult timp, deoarece capetele de citire trebuie să se miște des și departe,

Adresă	Lungime	Conținut
0	1	Stare (0 = inactiv, 80h = activ)
1	3	Primul sector (cap, cilindru și sector)
4	1	Tip (1 =DOS 12-bit, 4=DOS 16-bit, 5=DOS extins)
5	3	Ultimul sector (cap, cilindru și sector)
8	4	Primul sector (numărul logic al sectorului)
12	4	Numărul de sectoare din această partiție

Tabela 2 :Structura unei tabele de partiționare la un harddisk

pentru a ajunge la toate cluster-ele. Datorită înlănțuirii cluster-elor în FAT este posibilă reorganizarea datelor, astfel încît acestea să fie stocate în cluster-e învecinate. Pentru aceasta conținutul cluster-elor va trebui reîmpărțit (copiat) și apoi vor trebui modificate corespunzător intrările în FAT. Tocmai acestea sînt operațiile pe care le execută *Disk Optimizer* pentru a accelera accesarea datelor.

Viitorul S.G.F. MS-DOS

În viitor, S.G.F. se va modifica destul de puțin. Pe de o parte acest S.G.F. este suficient de bun pentru discurile de capacitate medie. Pe de altă parte există multe programe care utilizează anumite caracteristici ale acestui sistem, și dacă sistemul ar fi modificat aceste programe nu ar mai fi funcționale. Aceasta nu înseamnă însă că nu este de așteptat nici o noutate.

Astfel există, de exemplu, o componentă MS-DOS mai puțin cunoscută, care se numește *Network Redirector* și care permite cuplarea unui S.G.F. dorit în MS-DOS. Există deja cîteva domenii în care această facilitare este utilizată: unitățile CD, unele rețele și diferite alte periferice pot fi adresate prin această interfață. În anii următori vor fi cu siguranță noutăți în acest domeniu.

(R.M.)

La nivelul cel mai de jos

Turbo-Prolog - Partea a treia

În cea de a treia și ultima parte a cursului ne vom ocupa de modul în care se poate programa apropiat de codul mașină și de cuplarea unor proceduri externe într-un program Prolog.

Și în Turbo-Prolog sînt posibile programarea Low-Level (la nivelul de jos, apropiat mașină) și accesul la nivelul sistemului de operare. Ca și în celelalte limbaje de programare Turbo, există o serie de predicate (funcții) care permit accesul la nivel DOS și la nivelul hard. Pentru apelul unor comenzi MS-DOS poate fi utilizat predicatul "system". Pentru manipularea datei și orei sistem pot fi utilizate predicatele "date" și "time". În plus, începînd din versiunea 2.0, cu ajutorul predicatelor "envsymbol" și "comline" pot fi preluate variabilele de mediu și parametrii liniei de comandă. Predicatul "system" oferă următoarele două posibilități: "system (Comandă DOS)" și "system (Comanda DOS, Reset Video, Errorlevel)".

În cazul primei variante comanda DOS este transmisă predicatului ca șir, de exemplu: "system ("del *.*")". Dacă acest șir este vid, atunci se încarcă o copie a fișierului COMMAND.COM din care se poa-

te reveni în Prolog dacă se tastează "exit".

În cazul variantei extinse pe lângă comanda DOS se poate stabili prin intermediul argumentului "Reset Video" dacă se dorește, sau nu, refacerea imaginii curente ecran la terminarea execuției comenzii DOS. "Reset Video = 1" va determina refacerea ecranului, lucru care nu se va întîmpla dacă "ResetVideo = 0". În plus codul de retur al comenzii apelate este încărcat în variabila "Errorlevel" și poate fi prelucrat mai departe.

Cu ajutorul predicatelor "time" și "date" pot fi manipulate cu ușurință timpul și data sistemului. Sintaxa de apel este: "time (Ora, Minut, Secunda, Sutine)" sau "date (An, Luna, Zi)". Variabilele utilizate trebuie fie să fie toate legate, fie să fie toate libere. Dacă variabilele sînt libere, atunci în ele vor fi încărcate valorile curente din sistem; dacă variabilele sînt legate atunci se va modifica timpul sau data sistemului.

Predicatul "envsymbol" permite preluarea valorii unei variabile din mediul actual al programului. Sintaxa predicatului "envsymbol" este următoarea: "envsymbol (EnvVar, Valoare)". "EnvVar" stabilește numele variabilei de mediu a cărei valoare va fi returnată ca șir în variabila "Valoare". În cazul în care

variabila specificată nu există, predicatul "envsymbol" eșuează.

Pentru a putea accede la parametrii liniei de comandă poate fi utilizat predicatul "comline", care poate fi apelat cu sintaxa "comline (Parametri)".

În urma apelului parametrului liniei de comandă sînt legați de variabila șir "Parametri". Acest predicat a fost compilat și transformat în program executabil (.EXE), deoarece în Turbo-Prolog parametrul nu pot fi transmiși la fel ca și în Turbo-Pascal, de exemplu.

Programul de exemplu din fig. 1, arată modul în care pot fi utilizate predicatele prezentate.

Programul citește linia de comandă și interpretează parametrul acesteia ca pe o comandă DOS care trebuie executată. Dacă în linia de comandă nu se specifică nici un parametru atunci se încarcă o copie a lui "COMMAND.COM". Programul cronometrează durata de timp pentru care copia a fost încărcată și la revenire afișează acest interval de timp pe monitor.

Suplimentar se afișează valoarea variabilei de mediu PATH. Pentru ca programul să poată prelua parametrul, el trebuie compilat anterior și transformat în fișier executabil (.EXE).

```

predicates
  gettime (real)
  write_command (string)
  showpath
  do_DOS_Command

clauses
  gettime(Sec):-
    Time (H,M,S,S100),
    Sec = 3.6E2*H + 60*M + S + S100/100.

  showpath:-
    envsymbol("PATH",Path),
    write("\n\n Variabila de mediu PATH"),
    write("\n cste pozitionata pe", Path,"\n\n").

  write_command("):-
    write_command("COMMAND.COM").
  write_command(X):-
    write (X," a fost executata!\n\n").

do_DOS_Command:-
  gettime(SecStart),
  comline(ComandaDOS),
  system (ComandaDOS,1,_),
  write_command(ComandaDOS),
  gettime (SecEnd),
  Secunde = SecEnd-SecStart,
  writef("DOS a fost activ %-1.0f secunde!",Secunde),
  showpath,
  write("Terminati apasind o tasta ..."),
  readchar(_).

goal
  do_DOS_Command.

```

Fig. 1

Turbo-Prolog permite programatorului și accelerarea directă a întreruperilor calculatorului. Cu ajutorul predicatului "bios" pot fi executate toate întreruperile calculatorului. Acest predicat poate fi folosit cu una din sintaxele următoare: "bios (Interruptnummer, Input Register, OutputRegister)" sau "bios (Interruptnummer, InputRegister, OutputRegister, Flags)". "Interruptnummer" desemnează numărul întreruperii de executat. "InputRegister" și "OutputRegister" sînt date structurate de tipul "regdom". Acest tip de date va fi definit în felul următor:

```
domains
regdom = reg (integer, integer, integer,
integer, integer, integer, integer, integer)
```

Astfel nu mai este necesară o redefinire a lui în "domains".

Cu "reg (...)" sînt desemnați regiștrii individuali ai procesorului principal din care se citește sau în care se scrie.

```
/* Registri de intrare */
reg (AXi, BXi, CXi, DXi, Sli, Dli, DSi, ESi)
/* Registri de iesire */
reg (AXo, BXo, CXo, DXo, Slo, Dlo,
DSo, ESo)
```

Un apel al lui "bios" ar putea arăta în modul următor:

```
bios (Interruptnummer,
reg (AXi, BXi, CXi, DXi, Sli, Dli,
DSi, ESi),
reg (AXo, BXo, CXo, DXo, Slo, Dlo,
DSo, ESo)).
```

"AXi", "BXi", "...", desemnează valorile de intrare pentru execuția întreruperii. Valorile registrelor "AXo", "BXo", "...", sînt returnate de predicat după apelul întreruperii. Trebuie avută grijă ca toate variabilele "?Xi" să fie legate și respectiv toate variabilele "?Xo" să fie libere.

Opțional predicatul "bios" mai poate conține și un al patrulea parametru "flag" prin intermediul căruia pot fi returnate flag-urile procesorului. Figura 2 arată modul în care sînt împachetate flag-urile

■	■	■	■	U	R	I	E	V	N	■	H	■	P	■	C
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Fig. 2

procesorului într-o variabilă întregă. Prescurtările utilizate au următoarea semnificație:

- U = flag de depășire
- R = flag de direcție
- I = flag de întrerupere
- E = flag de pas cu pas
- V = flag de semn
- N = flag de nul
- H = carry auxiliar
- P = paritate
- C = carry

Care flag-uri sînt poziționate și după care întrerupere, se poate citi dintr-un manual DOS.

Așa cum a fost descris, în Turbo-Prolog există posibilitatea apelării unei întreruperi. În plus programatorul poate utiliza următoarele predicate pentru programarea low-level:

```
memword (Segment, Offset, Word)
membyte (Segment, Offset, Byte)
portbyte (PortNumber, Byte)
ptr-dword (StringVar, Segment, Offset)
```

"membyte" și "memword" permit accesul la octeți individuali. Dacă al treilea argument al predicatului este liber încă în momentul apelului, atunci celula de memorie specificată prin "Segment: Offset" este citită și valoarea ei este legată de această variabilă, altfel conținutul variabilei este scris în respectiva celulă de memorie.

Predicatul "portbyte" scrie, respectiv citește, un byte la portul specificat.

Cu ajutorul predicatului "ptr-dword" poate fi stabilită adresa absolută a unei variabile șir. Trebuie avută grijă în acest caz deoarece terminatorul de șir nu este caracterul null, la fel ca în Turbo-Pascal. Dacă variabila "StringVar" este liberă atunci de ea este legată valoarea "Segment: Offset", altfel Segment și Offset sînt legate de valorile corespunzătoare. Adresa absolută a celulei de memorie se

poate obține apoi cu relația: "Segment * 16 + Offset".

Programul din fig. 3 demonstrează modul în care pot fi utilizate predicatele low-level. Programul de exemplu comută mai întâi, printr-un acces direct la memorie, tasta "PrtScr". Apoi calculatorul poate fi accelerat sau frînat între anumite limite prin modificarea ratei de refresh. În cele din urmă cu ajutorul funcției "43h" a întreruperilor DOS este modificat "Hidden-Flag"-ul unui fișier. Dacă fișierul specificat este vizibil în director, atunci este ascuns și invers.

Turbo-Prolog pune la dispoziția programatorilor un mare număr de funcții high-level. Uneori este necesar totuși să se insereze într-un program și segmente realizate cu alte limbaje de programare procedurale. Pentru acest scop compilatorul dispune de interfețe pentru Assembler, C, Fortran și Pascal. În continuare va fi clarificat modul de utilizare al unor proceduri externe

```
predicates
ptrscroff
changerefresh
changehideflag

goal
ptrscroff,
changerefresh,
changehideflag.

clauses

ptrscroff:-
membyte(0,1280,1),
write("PtrScr - Tasta dezactivata!\n").

changerefresh:-
write("Noua rata de refresh (> =3):"),
readln(X),
port_byte(67,84),
port_byte(65,X).

changehideflag:-
write("Modificare hiddenflag !\n"),
write("Nume fișier: "),
readln(S),
ptr_dword(S,DSi,DXi),
bios(33,reg($4300,0,0,DXi,0,0,DSi,0),
reg(_,CX,_,_,_)),
bitxor(CX,2,CXi),
bios(33,reg($4301,0,CXi,DXi,0,0,DSi,0),
reg(_____,_)).
```

Fig. 3

```
IDIV_PROC SEGMENT BYTE
    PUBLIC 'CODE'
    ASSUME CS:IDIV_PROC
    PUBLIC idiv_0
```

```
idiv_0 proc far
    push bp
    mov bp,sp
    xor dx,dx
    mov ax,[bp] + 16
    mov cx,[bp] + 14
    idiv cx
    lds si,dword ptr [bp] + 10
    mov [si],ax
    lds si,dword ptr [bp] + 6
    mov [si],dx
    pop bp
    ret 12
idiv_0 cndp
```

```
IDIV_PROC ENDS
END
```

Fig. 4

prin intermediul a trei mici exemple în Assembler și C.

Toate procedurile care sînt formulate într-un alt limbaj, sînt apelate în Turbo-Prolog ca și predicate globale. La declararea unui modul extern trebuie specificat explicit limbajul folosit prin intermediul cuvîntului "language". Definiția unui predicat global va arăta în modul următor:

```
global predicates
test (integer, real)-
    (i,o), (o,i) language asm
```

Variantele de apel admisibile trebuie specificate înaintea cuvîntului cheie "language", "i" este utilizat pentru Input și "o" pentru "Output". Un predicat poate avea în Turbo-Prolog mai multe variante de tip și de apel. Din această cauză diferitele proceduri pentru același predicat trebuie să conțină un nume, urmat de o linie de subliniere ('-') și apoi să fie numerotate de la 0 în sus; de exemplu pentru predicatul "test" vor exista "test-0" și "test-1". Pentru adresarea memoriei Turbo-Prolog utilizează pointeri de 32 de biți și din această cauză și în procedurile externe apelurile și returnările din subprograme trebuie să fie de tipul "FAR".

Următorul exemplu prezintă o rutină în Assembler care efectuează o împărțire întreagă cu valorile parametrilor furnizați și

```
global predicates
idiv(integer, integer, integer, integer)
- (i,i,o,o) language asm
goal
write("Deimpartit: "),
readint(Deimpartit),
write("Impartitor: "),
readint(Impartitor),
idiv(Deimpartit, Impartitor, Rezultat, Rest),
write("Rezultat: ", Rezultat, " Rest: ", Rest),
nl.
```

Fig. 5

livrează ca parametri de ieșire cîțul și restul împărțirii. În fig. 4 este prezentat programul corespunzător scris în Assembler.

Modul de construire a stivei pentru rutina "idiv" este prezentat în fig. 5. În programul Prolog din fig. 6 este arătat modul de utilizare al modulului extern. Pentru a cupla cele două bucăți de program sursele "IDIV.ASM" și "ASM-PRO.PRO" vor trebui mai întîi compilate pentru a se obține fișierele obiect (.OBJ).

După compilare sînt disponibile fișierele "IDIV.OBJ", "ASM_PRO.OBJ" și "ASM_PRO.SYM". Cu comanda: "TLWK INIT ASM_PRO IDIV ASM_PRO.SYM. ASMMIXED,, PROLOG" se realizează editarea de legături și se obține programul executabil "ASMMIXED.EXE".

Turbo-Prolog poate fi completat și cu Turbo-C un mediu de programare extrem de puternic. Prin utilizarea de rutine Turbo-C facilitățile limbajului Prolog pot fi mult extinse. Două exemple (fig. 7 și 8) lămuresc modul de conlucrare al celor două limbaje de programare.

În primul exemplu se arată, cu ajutorul unui modul C simplu, cum poate fi implementat un predicat cu mai multe forme de apel. Pentru aceasta pentru fiecare din cele trei forme de apel "i,i,o", "i,o,i" și "o,i,i" este implementată cite o rutină C. Programul apelant Turbo-Prolog utilizează apoi predicatul global cu diferitele forme de apel.

La compilarea modulelor trebuie avute în vedere următoarele aspecte:

```
sub_0(int A, int B, int *C) /* (i,i,o) */
{
    *C = A - B;
}
sub_1(int A, int *B, int C) /* (i,o,i) */
{
    *B = A - C;
}
sub_2(int *A, int B, int C) /* (o,i,i) */
{
    *A = B + C;
}
```

Fig. 6

- 1) Numele funcțiilor C, care vor fi apelate de Turbo-Prolog trebuie să se termine cu o liniuță de subliniere și un număr de secvență.
- 2) Modulele corespunzătoare Turbo-C trebuie compilate cu modelul de memorie "large".
- 3) Dacă se utilizează Turbo-Prolog Version 1.1 un modul C cu variabilele registru nu poate fi compilat, aceasta înseamnă că trebuie poziționat comutatorul "use registers variables", al compilatorului, pe OFF.
- 4) Comutatorul compilatorului "Generate underbars" trebuie poziționat pe OFF.

La editarea de legături trebuie avute în vedere următoarele aspecte:

- 1) Modulul de inițializare Turbo-Prolog "INIT.OBJ" trebuie specificat ca prim parametru.
- 2) Dacă se utilizează funcții de bibliotecă Turbo-C, la editarea de legături trebuie dată și bibliotecă "CL.LIB". Pentru operații cu numere reale sînt necesare și bibliotecile EMU.LIB și MATH.LIB.

Apelul editorului de legături va arăta în modul următor:

```
TLWK INIT <modul principal TP>
<fis.obiect>
<modul principal TP.SYM>,
[nume fis. EXE],,
[bibliotecij PROLOG +
[EMU + MATHIL] + CL.
```

(continuare în pag. 51)

C++, partea a II-a

Diversitate virtuală

Pe lângă moștenire și încapsulare, polimorfismul este cea mai importantă caracteristică de bază a obiectelor. Acest articol arată cum se utilizează.

Înainte de a analiza mai amănunțit avantajele polimorfismului urmează un scurt paragraf relativ la funcții virtuale.

Prin introducerea funcțiilor virtuale, listingul EX5.CPP din articolul precedent se simplifică. În exemplu sînt declarate metodele pentru două clase diferite. Prima clasă "Semn" nu memorează informații despre fond. Clasa derivată "Semn2" memorează conținutul și atributul ecran din poziția X,Y, înainte de a afișa acolo un nou caracter.

Deoarece clasa derivată moștenește metodele clasei de bază, sînt imediat disponibile toate metodele pentru clasa derivată "Semn2". Dar, pentru acest obiect modificat, nu toate metodele lucrează. Acest lucru este evidențiat mai cu seamă de metoda "Ștergere", care funcționează diferit de la o clasă la alta. Clasa "Semn" nu ține seama de conținutul precedent al ecranului. Metoda Semn2::Ștergere() în schimb reafixează vechiul conținut al ecranului în poziția x,y. Deoarece și alte metode (ca de exemplu: "Mișcare()") se bazează pe metoda "Ștergere" corespunzătoare, în Listingul EX5 nu a fost posibilă preluarea nerestricționată a metodelor din clasa "Semn". În cadrul clasei "Semn2" a fost nevoie să se recurgă fie la metoda din "Semn::Citire()", "Semn::Ștergere()" și "Semn::Mișcare()" au fost definite ca metode virtuale. Acest lucru înseamnă că în cadrul clasei "Semn2" aceste metode sînt înlocuite cu metode noi. Ele nu folosesc metodele clasei de bază. Prin aceasta, listingul devine mult mai clar și mai scurt. Obiectele și metodele pot fi comparate cu propoziții ce conțin subiect și predicat. Obiectele (clasele) corespund subiectului iar metodele predicatului. Să considerăm propozițiile:

"Apa curge", "Nasul curge", "Timpul curge". Se constată cu ușurință că metoda "curge()" este aplicabilă unor obiecte diferite. Abia în legătură cu o clasă apare un mesaj inteligibil. Doar este o mare diferență între faptul că apa curge sau nasul curge. În schimb mesajul "Ora curge" se exprimă asemănător cu obiectul timp. Această prezentare plastică ajută mult la înțelegerea polimorfiei.

```

1 // Listing exemplu nr. 6
2 // funcții virtuale
3
4 #include <conio.h>
5 #include <stdio.h>
6
7 class Semn // definiția clasei Semn
8 {
9     int X, Y; // coordonate
10    public:
11    char valoare; // valoare alfanumerica
12    char attr; // atribut ecran
13
14    Semn(int InitX, int InitY)
15    {
16        X = InitX;
17        Y = InitY;
18    }
19
20    // Funcții-element publice
21
22    int Semn::GetX() { return X; }
23    int Semn::GetY() { return Y; }
24    void Semn::Position(int PosX, int PosY)
25    {
26        X = PosX; Y = PosY;
27    }
28    void Semn::Citire()
29    {
30        getch(X, Y, X, Y, &valoare);
31    }
32    void Semn::Scriere(char Sign)
33    {
34        Citire();
35        valoare = Sign;
36        puttext(X, Y, X, Y, &valoare);
37    }
38    void Semn::Modificare(char Sign)
39    {
40        valoare = Sign;
41        puttext(X, Y, X, Y, &valoare);
42    }
43    void Semn::Indicare()
44    {
45        puttext(X, Y, X, Y, &valoare);
46    }
47    virtual void Semn::Ștergere()
48    {
49        valoare = '';
50        Indicare();
51    }
52    virtual void Semn::Mișcare(int NewX, int NewY)
53    {
54        char memo_valoare;
55        memo_valoare = valoare;
56        Ștergere();
57        valoare = memo_valoare;
58        Position(NewX, NewY);
59        Indicare();
60    }
61    void Semn::Copiere(int NewX, int NewY)
62    {
63        Position(NewX, NewY);
64        Indicare();
65    }
66    void Semn::Seteaza_attr(char attr_nou)
67    {
68        attr = attr_nou;
69        Indicare();
70    }
71 };
72
73 // definiția clasei Semn2
74
75 class Semn2 : public Semn
76 {
77     protected:
78     char valoare_veche;
79     char attr_vechi;
80
81     public:
82     Semn2(int InitX, int InitY) : Semn(InitX, InitY)
83     {
84         valoare_veche = 0x0;
85         attr_vechi = 0x0;
86     }
87
88     // Funcții-element publice
89
90     void Semn2::Citire()
91     {
92         getch(GetX(), GetY(), \
93             GetX(), GetY(), &valoare_veche);
94     }
95     void Semn2::Ștergere()
96     {
97         valoare = valoare_veche;
98         attr = attr_vechi;
99         Indicare();
100    }
101    void Semn2::Mișcare(int NewX, int NewY)
102    {
103        char memo_valoare;
104        char memo_attr;
105
106        memo_valoare = valoare; valoare = valoare_veche;
107        memo_attr = attr; attr = attr_vechi;
108        Indicare();
109        valoare = memo_valoare;
110        attr = memo_attr;
111        Position(NewX, NewY);
112        Citire();
113        Indicare();
114    }
115 };
116
117 int main()
118 {
119     int X1, Y1, s, z;
120
121     X1 = 1; Y1 = 1;
122     Semn BSemn(X1, Y1);
123     BSemn.Scriere('+');
124     BSemn.Seteaza_attr(0x101);
125     for (s = 0; s < 25; s++)
126         for (z = 0; z < 80; z++)
127             BSemn.Copiere(X1+z, Y1+s);
128     X1 = 4; Y1 = 10;
129     Semn2 ASemn(X1, Y1);
130     ASemn.Scriere('A');
131     ASemn.Seteaza_attr(0x101);
132     printf("\n Apasati o tasta ... ");
133     getch();
134     ASemn.Mișcare(X1 - 1, Y1 - 4);
135     getch();
136     ASemn.Modificare('B');
137     getch();
138     ASemn.Ștergere();
139     getch(); clrscr();
140     return 0;
141 }

```

EX6.CPP

Transparență prin diversitate de forme

Asupra unor obiecte diverse pot fi aplicate aceleași metode. Sigur că nu este o mare diferență între deplasarea pe ecran a unui cerc sau a unui dreptunghi. Metoda "Deplasare()" poate fi aplicată unor metode diverse.

Din punct de vedere al conținutului, listingurile EX71.CPP până la EX73.CPP sînt legate. Primul listing este gândit ca o introducere în tema polimorfism. În listingul EX71.CPP sînt prezentate cele mai importante metode pentru obiectele "Semn", "Linie", "Pătrat" și "Dreptunghi", și anume "Ștergere()", "Indicare()", "Mișcare()", "Clasa "Linie" este o clasă derivată a clasei "Semn". Această clasă este o linie orizontală, ce se întinde pe direcția x. Clasa "Linie2" este o linie verticală. Această moștenește toate metodele clasei "Linie". Numai metoda "Indicare()" se deosebește de metoda "Linie::Indicare()". Clasele "Pătrat" și "Dreptunghi" sînt și ele clase derivate. Deși clasa "Pătrat" nu se prezintă ca un pătrat pe ecran, lungimea ei pe direcția x coincide cu lungimea pe direcția y. Pentru clasa "Pătrat" a trebuit definită numai metoda "Indicare". De remarcat că nu este necesar nici un atribut suplimentar. Pentru clasa "Dreptunghi" trebuie fixate lungimile pe direcția x și pe direcția y.

În programul principal se exemplifică apelul metodelor pentru cele 3 clase numite. Pentru toate clasele se fac aceleași lucruri, în aceeași succesiune. Acesta este polimorfism.

Ștergerea obiectelor se preia și pentru prim-plan. Atributul ecran are următoarea structură: biții 0-3: culoarea prim-planului, biții 4-6: culoarea fundului și bitul 7: blinking (luminare intermitentă).

Cu metoda "Semn::BackColor()" cei 3 biți ai culorii de fond sînt deplasați. Această metodă este parcursă la orice ștergere, indiferent de clasa derivată la care tocmai se aplică. În cadrul metodei "Semn::Ștergere()" se face trimiterea la metoda "Indicare()", declarată virtuală. Funcția Expand() furnizează un șir de caractere de lungime L, cu poziții succesive pentru "valoare" și "atribut".

Funcția "gettext(x1,y1,x2,y2,buffer)" citește o fereastră ecran dintre colțul stînga sus (de coordonate x1,y1) și colțul dreapta jos (x2,y2). În "buffer" se găsește o porțiune a memoriei de reîmprospătare a ecranului și anume (y2 - y1 + 1) linii și (x2 - x1 + 1) coloane; lungimea totală trebuie înmulțită cu 2, deoarece pentru fiecare poziție x,y pe ecran se stochează o valoare (1 octet) și un atribut de culoare (1 octet).

```

1 // Exemplul 7.1
2 // obiecte pluriforme
3
4 #include <conio.h>
5 #include <stdio.h>
6 #include <stdlib.h>
7
8 void Expand (char w, char a, int l, char *t)
9 {
10     int i;
11
12     for (i=0; i<=2*l; i=i+2) {
13         t[i] = w; t[i+1] = a;
14     }
15 }
16
17 class Semn // definește classa Semn
18 {
19     int x, y; // coordonatele
20
21     public:
22     char valoare; // valoare alfanumerică
23     char attr; // atribut ecran
24     Semn::Semn (int IniX, int IniY)
25     {
26         x = IniX;
27         y = IniY;
28     }
29     int Semn::GetX () (return x);
30     int Semn::GetY () (return y);
31     void Semn::Position (int PosX, int PosY)
32     {
33         x = PosX; y = PosY;
34     }
35     void Semn::Citire ()
36     {
37         gettext (x, y, x, y, &valoare);
38     }
39     virtual void Semn::Indicare ()
40     {
41         puttext (x, y, x, y, &valoare);
42     }
43     void Semn::Scriere (char Sign)
44     {
45         valoare = Sign;
46         Indicare ();
47     }
48     void Semn::Modificare (char Sign)
49     {
50         valoare = Sign;
51         Semn::Indicare ();
52     }
53     void Semn::Back Color ()
54     {
55         int Color;
56
57         if (attr > 128) {
58             Color = attr - 128;
59             Color = _rotr (Color, 4);
60             attr = Color + 16 * Color + BLINK;
61         }
62         else {
63             attr = _rotr (attr, 1);
64             attr = attr + 16 * attr;
65         }
66     }
67     void Semn::Ștergere ()
68     {
69         BackColor ();
70         valoare = ' ';
71         Indicare ();
72     }
73     void Semn::Mișcare (int NewX, int NewY)
74     {
75         Ștergere ();
76         Position (NewX, NewY);
77         Indicare ();
78     }
79     void Semn::Copiere (int NewX, int NewY)
80     {
81         Position (NewX, NewY);
82         Indicare ();
83     }
84     void Semn::Set_attr (char fattr, char battr = 0,
85                         char Bblink = 0)
86     {
87         if (battr < 8)
88             attr = fattr + 16 * battr + Bblink;
89         else
90             attr = fattr + Bblink;
91     }
92 };
93
94 class Linie : public Semn
95 {
96     protected:
97     int lungime;
98     public:
99     Linie::Linie (int IniX, int IniY) \
100         : Semn (IniX, IniY)
101     {
102         lungime = 0;
103     }
104     int Linie::GetL () (return lungime);
105     void Linie::Indicare ()
106     {
107         char temp[4096];
108         Expand (valoare, attr, lungime, temp);
109         puttext (GetX(), GetY(), \
110                 GetX()+lungime, GetY(), &temp);
111     }
112     void Linie::Fix_lungime (int L)
113     {
114         lungime = L;
115     }
116 };
117
118 class Linie2 : public Linie
119 {
120     public:
121     Linie2::Linie2 (int IniX, int IniY)
122         : Linie (IniX, IniY)
123     {
124         Linie::Fix_lungime (0);
125     }
126     void Linie2::Indicare ()
127     {
128         char temp[4096];
129         Expand (valoare, attr, GetL()*GetL(), temp);
130         puttext (GetX(), GetY(), \
131                 GetX()+GetL()-1, &temp);
132     }
133 };
134
135 class Patrat : public Linie
136 {
137     public:
138     Patrat::Patrat (int IniX, int IniY)
139         : Linie (IniX, IniY)
140     {
141         Linie::Fix_lungime (0);
142     }
143     void Patrat::Indicare ()
144     {
145         char temp [4096];
146
147         Expand (valoare, attr, GetL()*GetL(), temp);
148         puttext (GetX(), GetY(), \
149                 GetX()+GetL()-1, GetY()+GetL()-1, &temp);
150     }
151 };
152
153 class Dreptunghi : public Semn
154 {
155     int LungX;
156     int LungY;
157
158     public:
159     Dreptunghi::Dreptunghi (int IniX, int IniY) \
160         : Semn (IniX, IniY)

```

```

161 {
162     LungX = 0;
163     LungY = 0;
164 }
165 void Dreptunghi::Indicare ()
166 {
167     char temp [4096];
168
169     Expand (valoare, attr, LungX*LungY, temp);
170     puttext (GetX(), GetY(), \
171         GetX()+LungX-1, GetY()+LungY-1, &temp);
172 }
173 void Dreptunghi::Fix_lungime (int lx, int ly)
174 {
175     LungX = lx;
176     LungY = ly;
177 }
178 };
179
180 int main ()
181 {
182     int x1, y1, s, z;
183     // fond cu semn albastru
184     x1 = 1; y1 = 1;
185     Semn BSemn (x1, y1);
186     BSemn.Scriere (219);
187     BSemn.Set_attr (BLUE);
188     for (s=0; s<25; s++)
189         for (z=0; z<80; z++)
190             BSemn.Copiere (x1+z, y1+s);
191     printf ("Apasati o tasta ...");
192     getch ();
193     // Reprezinta si apoi sterge o linie
194     x1 = 10; y1 = 10;
195     Linie LSemn (x1, y1);
196     LSemn.Set_attr (RED, BLUE);
197     LSemn.Fix_lungime (60);
198     LSemn.Scriere ('-');
199     getch ();
200     LSemn.Stergere ();
201     getch ();
202     // Reprezinta si apoi sterge un patrat
203     x1 = 10; y1 = 2;
204     Patrat PSemn (x1, y1);
205     PSemn.Set_attr (GREEN, BLUE);
206     PSemn.Fix_lungime (20);
207     PSemn.Scriere ('=');
208     getch ();
209     PSemn.Stergere ();
210     getch ();
211     // Reprezinta si apoi sterge un dreptunghi
212     x1 = 10; y1 = 2;
213     Dreptunghi DSemn (x1, y1);
214     DSemn.Set_attr (CYAN, BLUE);
215     DSemn.Fix_lungime (40, 20);
216     DSemn.Scriere (219);
217     getch ();
218     DSemn.Stergere ();
219     getch ();
220     clrscr ();
221     return 0;
222 }

```

EX71.CPP

Desigur că e nevoie de un timp de acomodare pentru obișnuirea cu noile metode de lucru, orientate obiect; mai ales pentru programatorii ce au "crescut" cu limbajele de programare clasice. Dar este, sigur, mai simplu decât pare. Acest lucru ar vrea să-l demonstreze cele două listinguri care urmează.

Este de asemenea adevărat că cei odată "prinși" greu renunță la comoditățile obținute. Programele sînt pur și simplu mai transparente. Chiar dacă este necesară o anumită experiență în lucrul cu clasele și metodele, pentru a

recunoaște care funcții pot fi formulate virtual, pentru evitarea codului inutil.

În cele două listinguri care urmează, EX72.CPP și EX73.CPP metoda "Semn::Indicare()" este declarată ca funcție virtuală. Pentru că afișarea unei linii sau unui dreptunghi se deosebește totuși puțin de afișarea unui caracter individual, o primă diferență este deja numărul de atribute folosite pentru determinarea fără echivoc a obiectului definit.

Cele mai multe metode există pentru clasa de bază "Semn". Toate clasele derivate moștenesc aceste metode. Nicăieri nu este definită vreo metodă "Linie::Modificare()" sau "Linie::Copiere()". Nici nu este necesar. Cu ajutorul funcțiilor virtuale trebuie descrise numai excepțiile specifice claselor, și deja există toate metodele corecte pentru urmași. Că acest lucru funcționează, o ilustrează programul principal. Acolo aceleași metode se aplică claselor înrudite "Linie" și "Dreptunghi". Nu ar trebui să existe rețineri în definirea și testarea unor metode noi.

Modul de lucru al compilatorului Borland C++ 2.0 simplifică mult acest lucru. El oferă un debugger integrat, care sprijină mult parcurgerea pe pași a programului. Programul EX72.CPP ar trebui neapărat parcurs în modul trace (), pentru a vedea efectul metodelor moștenite și felul în care se "sare" la funcțiile specifice clasei. Desigur, ceva asemănător este adevărat și pentru compilatorul Zortech. Între timp există și un compilator shareware pentru C++, cu care se pot face primele experiențe.

```

1 // Exemplul 7.2
2 // metode pentru obiecte pluriforme
3
4 #include <conio.h>
5 #include <stdio.h>
6 #include <stdlib.h>
7
8 void Expand (char w, char a, int l, char *t)
9 {
10     int i;
11
12     for (i=0; i<=2*l; i=i+2) {
13         t[i] = w; t[i+1] = a;
14     }
15 }
16
17 class Semn // definește clasa Semn
18 {
19     int x, y; // coordonatele
20
21 public:
22     char valoare; // valoare alfanumerică
23     char attr; // atribut ecran
24     Semn::Semn (int InitX, int InitY)
25     {
26         x = InitX;
27         y = InitY;
28     }
29     int Semn::GetX () { return x; }
30     int Semn::GetY () { return y; }
31     void Semn::Position (int PosX, int PosY)
32     {

```

```

33         x = PosX; y = PosY;
34     }
35     void Semn::Citire ()
36     {
37         gettext (x, y, x, y, &valoare);
38     }
39     virtual void Semn::Indicare ()
40     {
41         puttext (x, y, x, y, &valoare);
42     }
43     void Semn::Scriere (char Sign)
44     {
45         valoare = Sign;
46         Indicare ();
47     }
48     void Semn::Modificare (char Sign)
49     {
50         valoare = Sign;
51         Indicare ();
52     }
53     void Semn::BackColor ()
54     {
55         int Color;
56
57         if (attr > 128) {
58             Color = _rotr - 128;
59             Color = _rotr (Color, 4);
60             attr = Color + 16 * Color + BLINK;
61         }
62         else {
63             attr = _rotr (attr, 4);
64             attr = attr + 16 * attr;
65         }
66     }
67     void Semn::Stergere ()
68     {
69         char Mvaloare, Mattr;
70         // dupa stergere, in obiect ramin nemodificate
71         // datele pentru valoare si atribut !!!
72
73         Mvaloare = valoare; Mattr = attr;
74         BackColor ();
75         valoare = ' ';
76         Indicare ();
77         valoare = Mvaloare; attr = Mattr;
78     }
79     void Semn::Miscare (int NewX, int NewY)
80     {
81         Stergere ();
82         Position (NewX, NewY);
83         Indicare ();
84     }
85     void Semn::Copiere (int NewX, int NewY)
86     {
87         Position (NewX, NewY);
88         Indicare ();
89     }
90     void Semn::Set_attr (char fattr, char battr = 0, \
91         char Blink = 0)
92     {
93         if (battr < 8)
94             attr = fattr + 16 * battr + Blink;
95         else
96             attr = fattr + Blink;
97     }
98 };
99
100 class Linie : public Semn
101 {
102 protected:
103     int lungime;
104 public:
105     Linie::Linie (int InitX, int InitY)
106         : Semn (InitX, InitY)
107     {
108         lungime = 0;
109     }
110     int Linie::GetL () { return lungime; }
111     void Linie::Indicare ()

```

```

112 (
113     char temp[4096];
114     Expand (valoare, attr, lungime, temp);
115     puttext (GetX(), GetY(), \
116         GetX()+lungime, GetY(), &temp);
117 )
118 void Linie::Fix_lungime (int L)
119 (
120     lungime = L;
121 )
122 );
123
124 class Dreptunghi : public Semn
125 {
126     int LungX;
127     int LungY;
128
129     public:
130     Dreptunghi::Dreptunghi (int InitX, int InitY)
131         : Semn (InitX, InitY)
132     {
133         LungX = 0;
134         LungY = 0;
135     }
136     void Dreptunghi::Indicare ()
137     {
138         char temp [4096];
139
140         Expand (valoare, attr, LungX*LungY, temp);
141         puttext (GetX(), GetY(), \
142             GetX()+LungX-1, GetY()+LungY-1, &temp);
143     }
144     void Dreptunghi::Fix_lungime (int lx, int ly)
145     {
146         LungX = lx;
147         LungY = ly;
148     }
149 };
150
151 int main()
152 (
153     int x1, y1, a, z;
154     // fond cu semn albastru
155     x1 = 1; y1 = 1;
156     Semn BSemn (x1, y1);
157     BSemn.Scriere (219);
158     BSemn.Set_attr (BLUE);
159     for (a=0; a<25; a++)
160         for (z=0; z<80; z++)
161             BSemn.Copiere (x1+z, y1+a);
162     printf ("Apasati o tasta ...");
163     getch ();
164     // Reprezinta si apoi modifica o linie
165     x1 = 10; y1 = 10;
166     Linie LSemn (x1, y1);
167     LSemn.Set_attr (RED, BLUE);
168     LSemn.Fix_lungime (60);
169     LSemn.Scriere ('-');
170     getch ();
171     LSemn.Miscare (x1+5, y1+2);
172     getch ();
173     LSemn.Stergere ();
174     getch ();
175     // reprezinta si apoi modifica un dreptunghi
176     x1 = 10; y1 = 2;
177     Dreptunghi DSemn (x1, y1);
178     DSemn.Set_attr (CYAN, BLUE);
179     DSemn.Fix_lungime (40, 20);
180     DSemn.Scriere (219);
181     getch ();
182     DSemn.Miscare (x1+5, y1+2);
183     getch ();
184     DSemn.Stergere ();
185     getch ();
186     clrscr ();
187     return 0;
188 )

```

EX72.CPP

În listingul EX73.CPP atributul "lungime" a fost modificat în "Lungx", astfel încît pentru dreptunghi a mai trebuit declarat numai atributul suplimentar "Lung Y".

Cu noua funcție element "Deplasare ()" devine necesară modificarea la afișare a claselor "Linie" și "Dreptunghi". Metoda "Deplasare()" face posibilă modificarea la afișare a claselor "Linie" și "Dreptunghi". Metoda "Deplasare()" face posibilă modificarea poziției cu +/-1 pe direcția x sau y. În contrast, metoda "Mișcare()" transferă obiectul într-o poziție nouă absolută.

În total există șase cazuri distincte de afișare a unei linii în raport cu fereastra ecran.

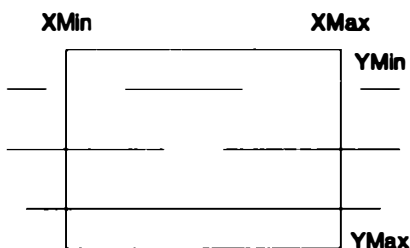


Fig. 1 Cazuri de pozitionare linii

Fie x_1 punctul din stînga al unei linii, x_2 punctul din dreapta (ele definind astfel segmentul reprezentat).

- 1) $x_1 \geq x_2 \geq X_{Min}$
- 2) $X_{Min} < x_1 < x_2 < X_{Max}$
- 3) $X_{Max} < x_1 < x_2$
- 4) $x_1 = X_{Min}, x_2 < X_{Max}$
- 5) $X_{Min} < x_1 = X_{Max}, x_2 < X_{Max}$
- 6) $x_1 = X_{Min}, x_2 = X_{Max}$

Motivat de fereastra ecran, există linii care sînt afișate în întregime (cazul 2), linii "ciuntite" în stînga și/sau în dreapta (cazurile 4, 5 și 6). Cazurile 1 și 3 nu pot fi afișate, deoarece linia iese cu totul din fereastră. Pentru liniile afișabile, rezultă următoarea reprezentare.

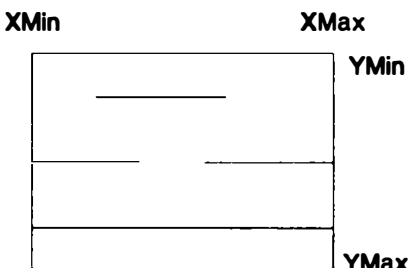


Fig. 2 Reprezentarea liniilor din fig.1

Aceleași cazuri există, prin analogie, la Dreptunghi.

La cele două exemple, în timpul ghidării pe ecran, nu se face nici o explicitare a tastelor cursor pentru deplasare, respectiv pentru dilatare sau comprimare. Dar pentru cel care-și introduce singuri programele, aceasta nu

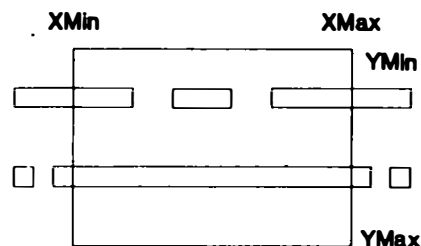


Fig. 3 Cazuri de pozitionare dreptunghiuri

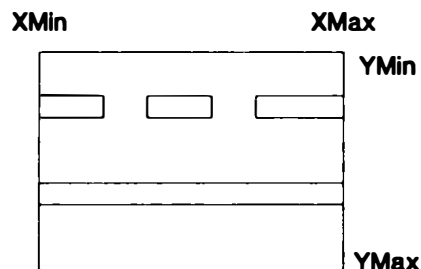


Fig. 4 Reprezentarea dreptunghiurilor din fig. 3

e o problemă. Pentru "Linie" a fost prevăzută dilatarea și comprimarea pe direcția x prin intermediul tastelor cursor - stînga și respectiv cursor - dreapta. Celelalte două taste servesc la deplasarea în sus sau în jos.

Din motive de spațiu s-a renunțat la deplasarea spre stînga sau dreapta. Desigur că nu va fi nici o dificultate pentru cei ce doresc să completeze.

La dreptunghi în schimb s-a implementat în întregime deplasarea în toate cele 4 direcții. După terminarea deplasărilor (printr-o tastă "necursor") se trece în regimul dilatare - comprimare.

Problemele "laterale" ilustrate în fig. 1 și 3 sînt rezolvate de funcțiile element "Stînga ()" și "Dreapta ()", ele returnînd punctele de început și sfîrșit pentru porțiunea ce "se vede" în fereastră. Pentru "testul celor două funcții" "Dreptunghiul" ar trebui deplasat lateral pînă ce iese din cîmpul vizual, apoi în sus sau în jos și apoi readus în fereastră.

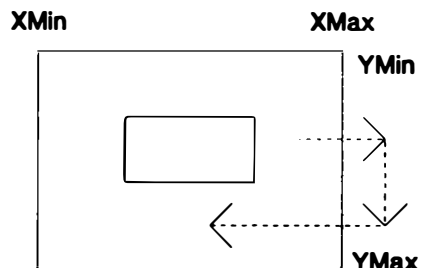


Fig.5 Propunere de test pentru EX73.CPP

După ultimele două listinguri, EX72.CPP și EX73.CPP, nu va fi o problemă integrarea metodelor "Mișcare()", "Copiere()", "Ștergere()", "Deplasa-

re()”, "Dilatare()" și "Comprimare()" într-un listing comun, pentru a avea o clasă completă cu toate metodele.

```

1 // Exemplul 7.3
2 // metode suplimentare
3
4 #include <conio.h>
5 #include <stdio.h>
6 #include <stdlib.h>
7
8 const XMax = 80;
9 const YMax = 24;
10 const XMin = 1;
11 const YMin = 1;
12 enum Boolean {false, true};
13
14 void Expand (charw, char a, int l, char *t)
15 {
16     int i;
17
18     for (i=0; i<=2*l; i=i+2) {
19         t[i] = w; t[i+1] = a;
20     }
21 }
22
23 class Semn // definate clasa Semn
24 {
25     int x, y; // coordonate
26
27     public:
28     char valoare; // valoare alfanumerica
29     char attr; // atribut ecran
30     Semn::Semn (int InitX, int InitY)
31     {
32         x = InitX;
33         y = InitY;
34     }
35     int Semn::GetX () {return x;}
36     int Semn::GetY () {return y;}
37     void Semn::Position (int PosX, int PosY)
38     {
39         x = PosX; y = PosY;
40     }
41     void Semn::Citire()
42     {
43         gettext (x, y, x, y, &valoare);
44     }
45     virtual void Semn::Indicare ()
46     {
47         putchar (x, y, x, y, &valoare);
48     }
49     void Semn::Scriere (char Sign)
50     {
51         valoare = Sign;
52         Indicare ();
53     }
54     void Semn::Modificare (char Sign)
55     {
56         valoare = Sign;
57         Semn::Indicare ();
58     }
59     void Semn::BackColor ()
60     {
61         int Color;
62
63         if (attr > 128) {
64             Color = attr - 128;
65             Color = _rotr (Color, 4);
66             attr = Color + 16 * Color + BLINK;
67         }
68         else {
69             attr = _rotr (attr, 4);
70             attr = attr + 16 * attr;
71         }
72     }
73     void Semn::Stergere ()
74     {
75         char Mvaloare, Mattr;
76         // dupa stergere, in obiect ramin nemodificate
77         // datele pentru valoare si atribut !!!
78
79         Mvaloare = valoare; Mattr = attr;
80         BackColor ();
81         valoare = ' ';
82         Indicare ();
83         valoare = Mvaloare; attr = Mattr;
84     }
85     void Semn::Miscare (int NewX, int NewY)
86     {
87         Stergere ();
88         Position (NewX, NewY);
89         Indicare ();
90     }
91     void Semn::Deplasare (int difx, int dify)
92     {
93         Stergere ();
94         Position (x+difx, y+dify);
95         Indicare ();
96     }
97     void Semn::Copiere (int NewX, int NewY)
98     {
99         Position (NewX, NewY);
100        Indicare ();
101    }
102    void Semn::Set_attr (char fattr, char battr = 0, \
103                        char Blink = 0)
104    {
105        if (battr < 8)
106            attr = fattr + 16 * battr + Blink;
107        else
108            attr = fattr + Blink;
109    }
110 };
111
112 class Linie : public Semn
113 {
114     protected:
115     int LungX;
116     public:
117     Linie::Linie (int InitX, int InitY)
118         : Semn (InitX, InitY)
119     {
120         LungX = 0;
121     }
122     int Linie::GetL () { return LungX; }
123     int Linie::Stinga ()
124     {
125         int x1;
126         x1 = GetX() - XMin + 1;
127         if (GetX() < XMin)
128             {
129                 if (GetX()+LungX > XMin)
130                     x1 = XMin;
131                 else
132                     x1 = 0;
133             }
134         return (x1);
135     }
136     int Linie::Dreapta ()
137     {
138         int x2;
139         x2 = GetX() + LungX - XMin;
140         if (GetX()+LungX > XMax)
141             {
142                 if (GetX() <= XMax)
143                     x2 = XMax;
144                 else
145                     x2 = 0;
146             }
147         return (x2);
148     }
149     void Linie::Indicare ()
150     {
151         char temp{4096};
152         int x1, x2;
153
154         x1 = Stinga();
155         x2 = Dreapta();
156         Expand (valoare, attr, x2-x1, temp);
157         if (x1 != 0)
158             if (x2 != 0)
159                 putchar (x1, GetY(), x2, GetY(), &temp);
160     }
161     void Linie::Fix_lungime (int L)
162     {
163         LungX = L;
164     }
165     void Linie::Dilatare ()
166     {
167         Stergere ();
168         LungX += 1;
169         Indicare ();
170     }
171     void Linie::Comprimare ()
172     {
173         Stergere ();
174         if (LungX > 0) LungX -= 1;
175         Indicare ();
176     }
177 };
178
179 class Dreptunghi : public Linie
180 {
181     int LungY;
182
183     public:
184     Dreptunghi::Dreptunghi (int InitX, int InitY)
185         : Linie (InitX, InitY)
186     {
187         LungX = 0;
188         LungY = 0;
189     }
190     void Dreptunghi::Indicare ()
191     {
192         char temp {4096};
193         int x1, x2;
194
195         x1 = Stinga();
196         x2 = Dreapta();
197         Expand (valoare, attr, (x2-x1+1)*LungY, temp);
198         if (x1 != 0)
199             if (x2 != 0)
200                 putchar (x1, GetY(), x2, \
201                         GetY()+LungY-YMin, &temp);
202     }
203     void Dreptunghi::Fix_lungime (int lx, int ly)
204     {
205         LungX = lx;
206         LungY = ly;
207     }
208     void Dreptunghi::Dilatare (int x1, int y1)
209     {
210         Stergere ();
211         LungX += x1;
212         LungY += y1;
213         Indicare ();
214     }
215     void Dreptunghi::Comprimare (int x1, int y1)
216     {
217         Stergere ();
218         if (LungX > x1) LungX = LungX - x1;
219         if (LungY > y1) LungY = LungY - y1;
220         Indicare ();
221     }
222 };
223
224 int main()
225 {
226     int x1, y1, a, z;
227     char taste;
228     Boolean sfirsit;
229     // fond cu semn albastru
230     x1 = 1; y1 = 1;
231     Semn BSemn (x1, y1);
232     BSemn.Scriere (219);
233     BSemn.Set_attr (BLUE);

```

```

234 for (s=0; s<25; s++)
235   for (z=0; z<80; z++)
236     BSemn.Copiere (x1+z,y1+s);
237 printf ("Apasati o tasta _");
238 getch ();
239 // Reprezentata si apoi modifica o linie
240 x1 = 10; y1 = 10;
241 Linie LSemn (x1, y1);
242 LSemn.Set_attr (RED, BLUE);
243 LSemn.Fix_lungime (60);
244 LSemn.Scriere (".");
245 do
246 {
247
248 sfirnit = true;
249 tasta = getch ();
250 if (tasta == 0)
251 {
252   sfirnit = false;
253   tasta = getch ();
254   switch (tasta)
255   {
256     case 72: LSemn.Deplasare (0, -1);
257             break;
258     case 80: LSemn.Deplasare (0, 1);
259             break;
260     case 75: LSemn.Comprimare ();
261             break;
262     case 77: LSemn.Dilatare ();
263             break;
264     // altceva docit tasta cursor
265     default: sfirnit = true;
266   };

```

```

267   };
268 } while (!sfirnit);
269 LSemn.Stergere ();
270 getch ();
271 // Reprezinta si egal modifica un dreptunghi
272 x1 = 10; y1 = 2;
273 Dreptunghi DSemn (x1, y1);
274 DSemn.Set_attr (CYAN, BLUE);
275 DSemn.Fix_lungime (15, 10);
276 DSemn.Scriere (219);
277 do
278 {
279   sfirnit = true;
280   tasta = getch ();
281   if (tasta == 0)
282   {
283     sfirnit = false;
284     tasta = getch ();
285     switch (tasta)
286     {
287       case 72: DSemn.Deplasare (0, -1);
288               break;
289       case 80: DSemn.Deplasare (0, 1);
290               break;
291       case 75: DSemn.Deplasare (-1, 0);
292               break;
293       case 77: DSemn.Deplasare (1, 0);
294               break;
295       // nu e tasta cursor
296       default: sfirnit = true;
297     };
298   };
299 } while (!sfirnit);

```

```

300 DSemn.Scriere ("=");
301 do
302 {
303   sfirnit = true;
304   tasta = getch ();
305   if (tasta == 0)
306   {
307     sfirnit = false;
308     tasta = getch ();
309     switch (tasta)
310     {
311       case 72: DSemn.Comprimare (0, 1);
312               break;
313       case 80: DSemn.Dilatare (0, 1);
314               break;
315       case 75: DSemn.Comprimare (1, 0);
316               break;
317       case 77: DSemn.Dilatare (1, 0);
318               break;
319       // nu e tasta cursor
320       default: sfirnit = true;
321     };
322   };
323 } while (!sfirnit);
324 DSemn.Stergere ();
325 getch ();
326 clrscr ();
327 return 0;
328 }

```

EX73.CPF

Cît ar fi durat elaborarea unui program, cu funcțiunile prezentate, într-un alt limbaj de programare, fără suportul orientării obiect?

Toate exemplele tratate pînă în momentul de față au în comun faptul că au lucrat exclusiv cu obiecte statice. Ceea ce înseamnă că necesarul de memorie e stabilit încă din momentul compilării

și este disponibil pe toată durata execuției. La obiecte mari, acest lucru poate să ducă la probleme cu memoria. (Iosif Fetich)

Turbo-Prolog partea a III-a

(continuare din pag. 45)

După ce ambele listinguri au fost compilate și s-au obținut fișierele obiect SUB.OBJ și C_PRO.OBJ, pentru cazul nostru, comanda de editare a legăturilor va fi:

```

TLINK INIT C_PRO SUB
C_PRO.SYM, CMIXED,, PROLOG.

```

În al doilea exemplu este cuplată cu un program Prolog tot o rutină C. În aceasta se lucrează cu numere reale, așa că la editarea de legături vor fi necesare bibliotecile EMU.LIB, MATH.LIB și CL.LIB. Modulul C calculează facto-

```

global predicates
sub (integer, integer, integer)
- (i,i,o),(i,o,i),(o,i,i) language c
goal
sub (100,150,X),
write("100-150 = ",X,"\n"),
sub (1000,Y,200),
write("1000-*,Y,* = 200\n"),
sub(Z,150,600),
write (Z,"-150 = 600\n").

```

Fig. 7

La nivelul cel mai de jos

rialul unei variabile întregi, rezultatul fiind de tip real (fig. 9).

După ce au fost create modulele obiect C_FAC.OBJ și FACT.OBJ, cu comanda de editare a legăturilor: TLINK INLT FACT C_FAC FACT.SYM, FMIXED,, PROLOG + FMIXED.EXE.

```

global predicates
factorial(integer,real) - (i,o) language c
goal
write("Calculul factorialului cu ajutorul"),
write(" unui modul extern scris in Turbo-C:\n"),
write("Factorial de "),
readint(N),
factorial(N,X),
write("Rezultat: ",X,"\n").

```

```

void factorial_0(int n, double *fac)
{
double help;

if (n <= 0)
{
*fac = 0.0;
return;
}

for (help = 2.0, *fac = 1.0; n > 1; n--, help + = 1.0)
*fac * = help;
}

```

Fig. 8

La utilizarea unor rutine Turbo-C mai trebuie să se țină seama de:

1) cînd sînt apelate funcții de bibliotecă în C acestea trebuie precedate de o linie de subliniere. Deci "test" în loc de "test". Acest lucru este necesar deoarece prin mascarea comutatorului compilatorului liniile de subliniere nu mai sînt generate în mod automat.

2) funcțiile apelate din Turbo-Prolog nu trebuie să returneze nici o valoare și din această cauză trebuie definite cu "void".

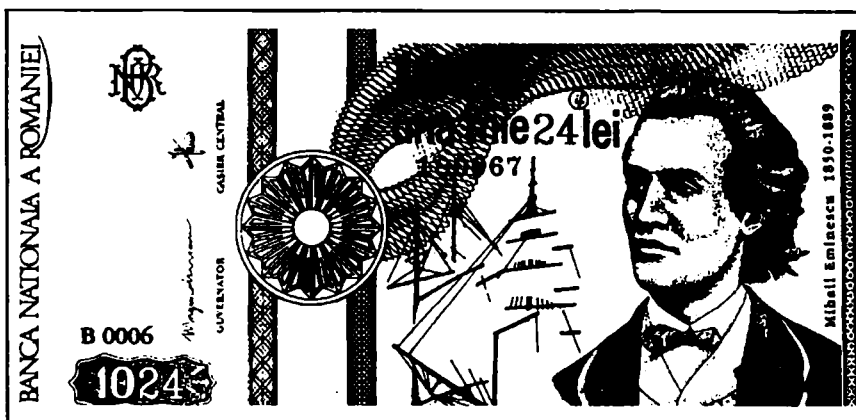
3) Funcțiile de ieșire "print" și "putc" nu funcționează atunci cînd Turbo-Prolog și Turbo-C sînt utilizate împreună.

În numărul viitor vă vom prezenta modul de folosire a listelor în Prolog. Așa cum s-a văzut Turbo-Prolog oferă posibilități de "programare logică". Programele date ca exemple au fost testate cu următoarele compilatoare: Turbo-Prolog 2.0, Turbo-Assembler 1.0 și Turbo-C 1.5.

(R.M.)

1024

Conform promisiunii din numărul 7 al revistei, vom premia - în fiecare număr al revistei - cel mai interesant program mai mic de 1024 octeți (în sursă) pe care îl primim spre publicare. De data aceasta, sînt doi cîștigători, : **Vasile Brînduș și Bogdan Munteanu** din Brașov. Felicitări !



```

1  program Park;
2
3  uses Dos,Crt;
4
5  var Regs : Registers;
6
7  procedure DPark (Drive : byte);
8  begin { DPark }
9  Regs.AH := 8;
10 Regs.DL := Drive;
11 intr ($13,Regs); { obținere parametri HD }
12 if Regs.AH = 0 then { drive-ul exista }
13 begin
14 inc (Regs.CH);
15 if Regs.CH = 0 then { depășire }
16 Regs.CL := Regs.CL+$40;
17 Regs.AH := $0c;
18 Regs.DL := Drive;
19 intr ($13,Regs); { cautare pista }
20 write ('Hard-discul nr. ',chr(Drive-$80+ord('0')),' parcat ');
21 if Regs.AH = 0 then
22 writeln ('corect !')
23 else
24 writeln ('cu eroare ...',Regs.AH);
25 end;
26 end; { DPark }
27
28 begin { Park }
29 ClrScr;
30 Regs.AH := 0;
31 Regs.DL := $80;
32 intr ($13,Regs); { Resetare controller HD }
33 DPark ($80); { Parcare primul HD }
34 DPark ($81); { Parcare al doilea HD }
35 write ('Acum puteti opri calculatorul ...',^G^G);
36 inline ($FA/$F4); { blocare calculator cu cli / hlt }
37 repeat until false { bucla infinita }
38 end. { Park }
    
```

PARK.PAS

Ce zi e astăzi ?

Programul ce urmează, scris în FORTRAN-77, tipărește data, ora și ziua săptămîinii pe display, pe imprimantă sau într-un fișier disc. Menționez că ziua săptămîinii se stabilește automat, pe baza datei setate în sistem.

Sandu Răzvan,
Bv. Aerogării 12, 71547 București I

Parcarea harddisk-urilor

Partea cea mai costisitoare și, în același timp, cea mai sensibilă a unui PC este harddisk-ul. Este deci evidentă necesitatea acordării unei griji deosebite exploatarea acestor componente.

O mare importanță o are protejarea suprafeței magnetice la transportul calculatorului și la pornirea/oprirea lui. În ultimul caz, datorită fenomenelor electrice tranzitorii ce se petrec în aceste condiții, este posibil ca prin capetele de scriere ale harddisk-ului să apară un curent electric accidental care să ducă la pierderea informației înscrise anterior pe pista respectivă. Dacă ultima citire sau scriere s-a făcut pe pista ce cuprinde FAT-ul sau directorul rădăcină, atunci pagubele sînt considerabil mai mari.

Programul propus realizează deplasarea capetelor harddisk-ului dincolo de ultima pistă folosită și blocarea calculatorului. În acest fel, la oprirea calculatorului capetele se vor găsi în dreptul unei piste nefolosite (deci care nu cuprinde informație).

Ing. Vasile Brînduș,
str. Aurora nr. 25 bl. 10 ap. 32 2200 Brașov
Ing. Bogdan Munteanu,
str. Operetel nr.52 C ap. 12, 2200 Brașov

```

1  SUBROUTINE WEEK(NL,N)
2  C  Aceasta subrutina scrie ziua saptaminii,
3  C  data si ora pe unitatea logica NL,
4  C  alinate cu N spatii
5  CHARACTER *9 SAPT(0:6),D,T
6  INTEGER ZI, ANLU(12), AN
7  DATA SAPT/'DUMINICA','LUNI','MARTI',
8  '-MIERCURI','JOI','VINERI','SIMBATA'/
9  DATA ANLU/0,3,3,6,1,4,6,2,5,0,3,5/
10 CALL IDATE(LUNA,ZI,AN)
11 CALL DATE(D)
12 CALL TIME(T)
13 A = A/4
14 M = INT(A)
15 J = M + AN + ANLU(LUNA) + ZI
16 K = MOD(J,7)
17 IF(N.NE.0) THEN
18 WRITE (NL,15) SAPT(K),D,T
19 ELSE
20 WRITE (NL,18) SAPT(K),D,T
21 ENDIF
22 RETURN
23 15 FORMAT (' ',<N>X,A9,2X,A9', '<N>X,A8)
24 18 FORMAT (' ',A9,2X,A9',A8)
25 END
    
```

WEEK.F77

Mica publicitate

Schimb programe Basic pentru compatibile Spectrum. Tel. 929/48591, Mihai

Vind calculator "CIP", 64k, compatibil ZX-Spectrum, color înregistrez dischete 5,25" sau 2,5" cu diferite programe (JOCURI, UTILITARE, COMPILATOARE, SISTEME). Tel. 90/16.78.13

Am 15 ani și învăț la Liceul Teoretic "Avram Iancu" din Cluj, clasa a X-a, secția de informatică... Aș dori ca prin intermediul revistei să pot face cunoștință cu tineri de vârsta mea și cu preocupări apropiate. Grama Radu, str. Gorunului, nr. 1, bl. D12, sc. B, ap. 20, 3400 Cluj Napoca.

REDUCERE !!

Începând cu nr. 292 oferim tuturor celor interesați posibilitatea de a se abona la revista noastră la un preț mai convenabil: oricine solicită un număr de cel puțin 10 exemplare, va avea o reducere de 10%, dacă va achita în avans contravaloarea revistelor cerute (din care se scade 10% din valoarea lor). Deoarece prețul poate varia de la un număr la altul, vă rugăm fie să "deschideți cont" la noi, fie să luați legătura cu redacția.

Pentru cei care doresc numere mai vechi ale revistei: mai sînt disponibile o parte din revistele editate anul trecut, și anume numerele 2-891. Acestea pot fi achiziționate în aceleași condiții ca mai sus.

Poșta redacției

De curînd, am intrat în posesia a două mouse-uri (unul de tip "Microsoft" iar celălalt de tip "Leonard") și aș dori să știu ceva mai amănunțit cum funcționează și cum pot fi utilizate; de asemenea aș mai fi interesat și de programele care fac posibilă utilizarea mouse-urilor. Dacă nu veți avea suficient spațiu, pentru a-mi satisface cerințele, la rubrica "Poșta redacției", pe care o rezervați pentru corespondență, v-aș fi recunoscător dacă mi-ați trimite datele solicitate (bineînțeles dacă este posibil) pe adresă:

Cartier Calea București, Bloc B5, Scara 2, Etaj 3, Apartament 7, Tel. 45344, cod 1100, Localitatea Craiova, Județul Dolj.

Vă mulțumesc!

Blaghi Sebastian Paul

Cîteva informații veți găsi poate în if nr.4/91 (7).

În revista dumneavoastră am găsit și o serie de articole extrem de interesante despre programe ce pot prelucra și interpreta imagini și m-am gîndit ca împreună cu un profesor să fac un program ce să interpreteze amprentele digitale, iar în urma prelucrării imaginii furnizate să pună un diagnostic cît mai corect asupra bolii de care suferă pacientul. Am acces la un scanner cu ajutorul căruia să pot încărca imaginea, dar ceea ce îmi lipsește este un

program care să o analizeze. V-aș ruga, dacă puteți, să-mi furnizați cîteva denumiri de programe care să realizeze asemenea lucruri, precum și informații de unde le-aș putea procura. Aș mai dori să-mi recomandați și bibliografie de citit despre programe și despre domeniul respectiv.

Pentru a fi mai explicit aș dori un utilitar care să reușească să recunoască și să analizeze o dreaptă, o intersecție de drepte, arce de cerc, de parabolă, bucle, etc. și să-mi furnizeze coordonatele caracteristice fiecărui element al imaginii. Știu că pare cam mult ceea ce aș dori eu să realizeze un singur program, dar poate că reușesc să combin între ele acțiunile mai multor programe.

Aștept cu nerăbdare răspunsul dumneavoastră pe adresa: București, str. Ștefan cel Mare, nr. 2, bl. 13, sc. A, ap. 16, sect. 1, Tel: 11.09.28.

Cu mulțumiri,

Ionuț Ciubotaru,
student medicină anul III

Pe toți corespondenții noștri îi rugăm să se uite și peste talonul din aceșt număr și serviciul Share-if pe care îl lansăm. Îi rugăm pe toți cei care ne trimit materiale spre publicare să ne specifice în scrisoarea însoțitoare dacă sînt sau nu de acord ca materialele trimise să intre în baza pentru share-if, după, înainte sau în loc de publicarea în "if".

Iosif Fettich

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:

Alba Iulia

» Computer 5 S.R.L.
tel. 22173

Arad

» BB Computer
Boros Ladislau
tel. 17583, 31778

Bacău

» Pătru Liviu
tel. 44242

Baia Mare

» Rîțiu Traian
tel. 25230

» Omni Computing
tel. 11794

» SINTEC S.R.L.
tel. 15720 - 171

Brașov

» Pummi S.R.L.
tel. 22018
Taifun S.R.L.
tel. 87368

» West Side Computers
tel. 49151

Brăila

» Teodorescu Constantin
tel. 47203

București

» Dracea Tudor
tel. 210340

» Bronislaw
Telecom S.R.L.

tel. 382136, după ora 17

Buzău

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

Cluj Napoca

» S.C. "AbMod" S.N.C.
—Magazin
Hardware-Software
b-dul 22 Decembrie
nr. 135

—Sediul central

tel. 111090

» Costin Suciu Libris

tel. 145526

» Roșca S.N.C.

tel. 138773

Constanța

» Ionescu Niculae
tel. 655077

Craiova

» Stătescu Paul
tel. 48117

Deva

» Toma Tudor
tel. 10144, 10234

Drobeta Turnu-Severin

» Bazavan Constantin
tel. 23666 - 143

Iași

» Buleu Daniel
Complex stud.
T. Vladimirescu
cam. T13 cam. 106

» KEYSYS S.R.L.
tel. 44499

Onești

» Brumă Cornel
tel. 24222-236

Oradea

» S.C. "AbMod" S.N.C.
tel. 60278, 44476

Petroșani

» S.C. Infotel-Pub-
Service S.R.L.
tel. 43100, 45537

Pitești

» Sorel S.R.L.
tel. 23600

Rădăuți

» Burciu Florin
tel. 63483

Satu Mare

» Viorel Santoi
Micro 17, str. Aniversării
bl. VA1, ap.6

Sibiu

» Pop Ioan
tel. 32011 / 208

Sighișoara

» Beer Mihai
tel. 73638

Suceava

» S.S.I. Suceava
Cazan Maria
tel. 15133

Timișoara

» Tomoroga Mircea
tel. 77422

Căutăm difuzori/distribuitori autorizați pentru difuzarea revistei "if" !. Condiții avantajoase !

Rugăm persoanele interesate să ia legătura cu redacția.

Revista "if" vă ofera spațiu pentru publicitate și reclamă la următoarele tarife:

o pagină alb-negru

50.000 lei

Pentru anunțurile care ocupă numai o parte din suprafața paginii, prețul este de 100 lei/cm²

Un nou serviciu pentru cititorii noștri:

SHARE - If

Pentru o mai bună circulație a informației între ofertanți și solicitanți de programe (și nu numai ...), vă oferim posibilitatea să:

- economisiți timpul necesar introducerii programelor mai lungi, al căror listing a apărut în "If".
- preluați pe dischetă, pentru eventuale multiplicări, textul articolele din "If" pe care ați vrea să le aveți disponibile și în această formă
- preluați acele programe și texte de articole care ne-au fost trimise spre publicare, dar pe care, din variate motive (cel mai adesea, lipsa de spațiu) - nu ajungem să le publicăm. (Desigur, aceste materiale le difuzăm numai atunci când există acordul autorului - încă nu am găsit o variantă de recompensare a autorului pentru această variantă de difuzare) O listă incompletă: criptare fișiere, completare formulare, o parolă solidă, nucleu de multitasking, "catalog" de dischete, etc.
- faceți cunoștință cu acele programe din care ni s-au trimis demo-uri, respectiv **să difuzați programele dumneavoastră demonstrative** prin intermedierea noastră. Acești din urmă sperăm să intereseze în mod deosebit pe toți autorii noștri de programe, cărora le este destul de greu să-și facă publicitatea de care au, evident, nevoie.

Cei dornici să includem demo-urile lor în oferta noastră sînt rugați să achite, odată cu expedierea dischetei sursă, suma de 500 lei în contul nostru: Cont nr. 40729-96010402, Banca Română pt. Dezvoltare Mureș.

(Cei care trimit materiale informativ/didactice, și nu publicitare, nu au de achitat nimic.) Prețul pe care îl practicăm pentru o dischetă este, din păcate, greu de definit în avans, din motive cunoscute de toată lumea. De aceea vă rugăm să vă interesați - telefonic sau în scris - în momentul lansării comenzii. În principiu, prețurile sînt identice cu prețurile cu care, la rîndul nostru, am achiziționat dischetele, la care se adaugă 50 lei pentru munca noastră și, suplimentar, taxa de expediere. Desigur, cei care ne trimit dischetele lor, au de achitat doar taxa de copiere și expediere, iar cei care trec sau trimit pe cineva pe la redacție, vor avea de achitat doar cei 50 lei, taxa de copiere.

**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

MicroInf - software profesionist pentru neprofesioniști!

CERBER

- protecția programelor și datelor împotriva virușilor și accesul persoanelor neautorizate

MICROLINK

- comunicarea între sucursale în mod rapid și eficient

μ UTILITY

- confort în întreținerea fișierelor

FACT

- o informare rapidă asupra tranzacțiilor efectuate

CONFIG

- configurator rapid și flexibil

MIDAS

- achiziționarea de date din procesul de producție

Distributor autorizat: **MBL Computers SRL**

București, Calea Dorobanților 152, sector 1, tel.333227,333706



str. Gheorghe Doja 36
Tîrgu-Mureș ROMÂNIA
Adresa poștală: C.P. 64, O.P. 1
4300 Tîrgu-Mureș
ROMÂNIA
Telefon/Telefax: 004-954-31660

AUTHORIZED REMARKETER - IBM PERSONAL COMPUTER PRODUCTS

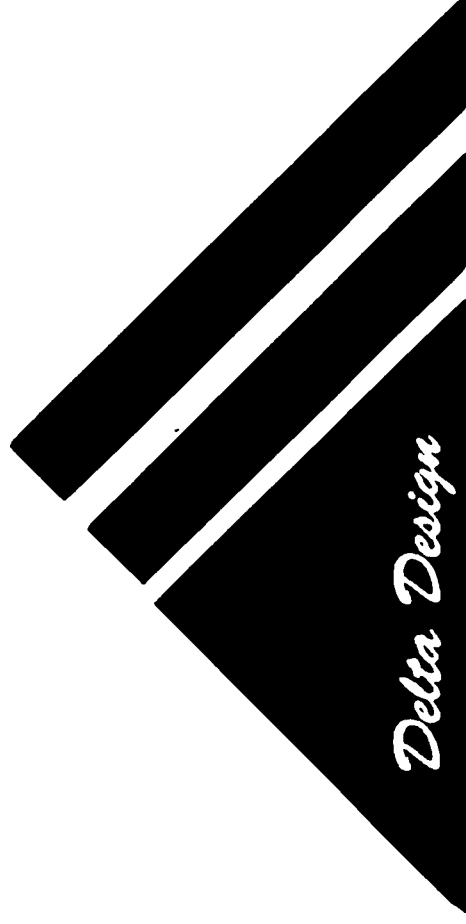
AUTHORIZED REMARKETER - IBM PERSONAL SYSTEM/1 COMPUTER

vă oferă

CALCULATOARE PERSONALE

din gama PS/1 și PS/2

Raportul preț/performanță este în favoarea Dvs. !



Delta Design S.A., Hotel București, 63-81 Victoria Street,
Wing D - Agencies, Stair E 2, 4th Floor, Sect. 1, Bucharest,
Phones: 15.42.26 / 40 015.45.80 Ext.2404 / Fax: 40 13.60.40.

99.94% of the users fully satisfied of the delivered products

Authorised distributors for the following companies:

Tulip Computers, Apple Computer Inc., Microsoft, Autodesk,
Ashton Tate, Claris, Corel, Delta Design International etc.