

CALCULATOARE PERSONALE

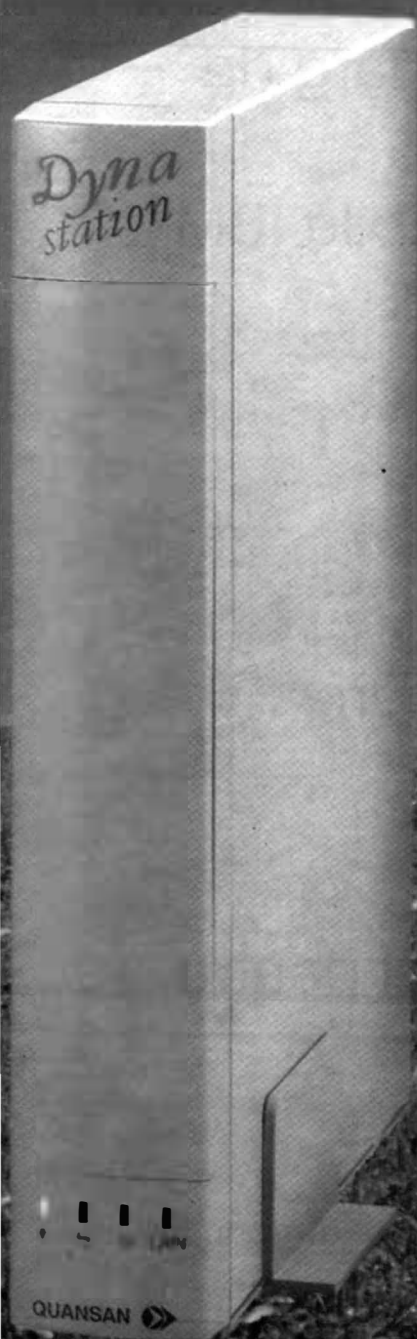
3 / 92

(14)

150 Lei

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

ISSN 1220-1529



Baze de date:

- Baze de date orientate obiect
- BLOB - Binary Large Objects
- Baze de date Client-Server

» Ce calculatoare putem cumpăra, de unde și la ce prețuri?

» PC-uri extensibile

» Instalarea rețelelor Novell

Cursuri:

- C++ (IV)
- Programarea Windows (I)

Ultima oră!

S-a înființat POLICOM S.A., societate comercială multinațională cu capital privat, avînd principal acționar Ab Mod S.N.C.

Noua firmă vă oferă un sortiment larg de produse:

- Calculatoare personale compatibile IBM
- Imprimante matriciale
- Dispozitive grafice
- Floppy - dischete
- Produse software originale
- Copiatoare Toshiba
- Truse și accesorii pentru întreținere

Nu uitați!

POLICOM S.A. un partener potențial pentru afacerile dumneavoastră.

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

Cuprins

Hotline

- Apple în România pag. 5
- IBM își extinde familiile de calculatoare PS/2 și PS/1 pag. 6
- ROMORGADATA '92 pag. 6
- CERF '92 pag. 6
- Puterea unei workstations la nivel DOS . . . pag. 8

News

- A picta ca van Gogh pag. 9
- Alpha - o nouă tehnologie a procesoarelor . . . pag. 10

Tehnologii

- Head Up Display pag. 12

Fundamente

- Comunicarea om-mașină pag. 14
- Sisteme multimedia. Metode de acces la text.
Metoda semnăturii. pag. 38

Vă prezentăm

- Construirea de sisteme multimedia pag. 20
- Noutăți la Guru 3.0 pag. 21
- Sisteme expert create cu Guru pag. 21

Baze de date

- Blob, blob... pag. 22
- Baze de date orientate obiect pag. 23
- Baze de date Client-Server pag. 25
- Gupta SQLBase Server pag. 27
- Informix SE și OnLine pag. 28
- Ingres SQL-Server pag. 28
- Oracle SQL-Server pag. 28
- Sybase SQL-Server pag. 29
- XDB-SQL-Server pag. 29
- Novell NetWare SQL pag. 29

Anchetă

- Piața românească de calculatoare pag. 34

Hardware

- PC-uri extensibile (2) pag. 36

Cursuri

- Stăpînind ierarhii, Curs C++ (IV) pag. 44
- Programarea Windows (I) pag. 49

Rețele

- Instalarea sistemului de operare
Netware 2.2 (1) pag. 54

Laborator

- Cîte ceva despre VGA pag. 58
- Organizarea memoriei grafice la VGA. pag. 58
- Programarea paletii de culori pentru
modurile cu 256 culori. pag. 58
- Modul 320x240x256 pag. 58
- De ce am ales modul 320x240x256 ? pag. 58
- Modul 360x480x256 pag. 58

Colțul programatorului

- Cine indexează mai bine ? pag. 60
- Clipper 5.01 pag. 60

Practica

- Trucuri Quattro Pro 3.0 pag. 61
- Corecții Borland în unit-ul »editors«
(Turbo Vision) pag. 62
- Cod autoreproducător pag. 62
- Here comes the music pag. 63
- Reversi - 64:0 pag. 63
- În contra virușilor pag. 64
- Un doctor pentru dischete pag. 64
- Print - o comandă ignorată? pag. 64

Rubrici

- Caseta redacției pag. 4
- Editorial pag. 4
- SHARE - if pag. 65

Așteptându-l pe Godot

Luna mai a fost o lună bogată în expoziții de tehnică de calcul: ROMOR-GADATA, la Pavilionul Expozițional, CERF '92, la Polivalentă, Expoziția Apple, la Sala AGIR. Expozițiile s-au constituit, toate în felul lor, în adevărate evenimente pentru împătimitii într-ale calculatoarelor, care-au avut ocazia să-și scalde cel puțin privirea, dacă punga nu le-a permis mai mult. Din păcate chiar și cei cu punga plină au rămas, de data aceasta, numai cu privitul. Ceea ce a caracterizat, mai mult sau mai puțin, toate aceste expoziții a fost așteptarea. Numai că cel așteptat nu era Godot ci Fixingul. Și fixingul nu venea... Nici din vina expozanților și nici din cea a organizatorilor expozițiilor. Vina era în altă parte și nu era o vină colectivă. Așa că a trebuit să ne mulțumim să privim, să mirosim și eventual să pipăim. Ne bucură în schimb faptul că multe produse interesante, lansate nu de mult pe alte piețe, au apărut și în țară. Aș enumera doar câteva: Windows 3.1, C++ 3.0, Object Vision 2.0, Ventura Publisher 4.0, 486/50, Screen Machine.

Expoziția de la Polivalentă a fost interesantă și prin încercarea de specializare a profilului, calculatoarele ne mai fiind expuse printre articole de lenjerie și sticle cu sucuri. Reușita a fost doar parțială nu atât pentru faptul că au fost expuse și produse electronice, fapt inclus de altfel în program, cât datorită faptului că multe firme importante au lipsit de la manifestare. Sperând că se va reuși trecerea peste relațiile de "familie", noi așteptăm în continuare primul Comdex sau CeBIT românesc. Oricum ne bucură faptul că, de această dată, sub cupola sălii, în loc să sclipască lămpașele garbage-collector-ilor Pieței Universității au sclipit monitoarele. Mulțumim!

Ne cerem scuze pentru ținuta numărului anterior. Purtând numărul 13, a avut ghinion. Încercăm să revenim cu o ținută mai decentă, chiar dacă prețul este mai ridicat. Este important conținutul, dar se mai întâmplă și ca haina să-l facă pe om.

Pentru cititorii care nu au reușit să facă cunoștință cu revista de la început, anunțăm că revista a intrat într-al treilea an, că au apărut 3 numere în 1990 (1-3/90), 8 numere în 1991 (1-8/91) și, cu acesta, 3 numere în 1992 (1-3/92). În stoc mai sînt disponibile reviste din toate numerele apărute în '91 și '92, mai puțin numărul 3/91. Din '90 nu mai avem în stoc nici o revistă.

În acest număr intenționăm să vă prezentăm un amplu material de sinteză despre piața românească de calculatoare personale și de dischete. Întrucît am primit foarte puține răspunsuri pînă la închiderea ediției, sinteza obținută este mai subțire. Deocamdată nu vă putem spune nimic despre dischete (la CERF o dischetă bună costa 500-600 lei). Pe măsură ce vom intra în posesia unor noi informații vom face și noi un "refresh". De asemenea îi rugăm pe toți cei care doresc să participe la anchetele pe care le vom iniția în viitor să ne scrie, specificînd domeniile de interes.

Anunțăm cititorii că nu am renunțat la nici unul din concursurile lansate. Așteptăm răspunsurile Dvs. Promitem ca de îndată ce vom intra în posesia cel puțin a unei bancnote de 5.000 lei, să mărim cel puțin premiul concursului "1024 de lei pentru cel mai bun program de cel mult 1024 de octeți".

În acest număr începem un nou curs: "Programarea Windows" și o serie de articole despre instalarea rețelelor Novell. De asemenea, am acordat un spațiu mai larg problematicei bazelor de date, prezentînd principalele S.G.B.D.-uri client-server. Chiar dacă acestea au pătruns încă prea puțin în țară, subiectul este interesant, și întreprinderile medii și mari nu se vor putea descurca în viitor fără un astfel de S.G.B.D. Așa că este bine să știm ce ne așteaptă, măcar din acest punct de vedere...

Romulus Maier

if

revistă de informatică
editată de firma Micro ATCI

Director: ing. Dumitru Dunca

Redacția:

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

Colaboratori externi:

ing. Caraiani Mitică
mat. Cozac Nelu
ing. Guțică Mirela
Hamor Soft SRL
ing. Malide Cristian
șef lucr. ing. Marinceu Dimitrie
corectura art. "Comunicarea
om-mașină" a fost făcută de dna.
prof. Sârbu Nastasia

cursul "Programarea Windows"
a fost revizuit de:

ing. Pop Adrian și
mat. Rotariu Eugen

Tiparul: Tiporex S.R.L.

Tiraj: 8.000 ex.

Preț: 150 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. Pen-
tru amănunte vă rugăm să luați
legătura cu redacția.

Cei care doresc să anexeze revis-
tei pliante publicitate tipărite în
regie proprie, sînt rugați și ei să se
adreseze redacției.

Adresa și telefonul redacției:

Micro ATCI, redacția "if",
RO-4300 Tirgu Mureș,
C.P. 172, O.P. 1,
tel./fax 954/31660,
41417 sau 46885 (după ora 19)
telex 65354 (Intertours)

□ **Peacock** - Monitoare Peacock Trinitron
Peacock Computer GmbH, Wünnenberg-Haaren, ofertant de soluții profesionale PC complete, are disponibilă o nouă generație de monitoare. Aparatele se disting prin utilizarea unor tuburi cinescopice Black-Trinitron, care oferă o rezoluție mai ridicată, contraste mai mari, înprospătarea mai rapidă a imaginii și culori excelente. Sînt livrabile monitoare de 14, 17 și 20 de inch, care în modul noninterlaced permit obținerea tuturor rezoluțiilor VGA, pînă la o rezoluție de 1280 X 1024 la monitoarele de 20 inch. Monitoarele color Peacock sînt realizate conform celor mai noi tehnologii în domeniu, iar radiațiile electromagnetice emise nu depășesc valorile stabilite de normele suedeze (MPR-II).

PEACOCK Computer GmbH,
Industriest. 14, D-4798 Wünnenberg-Haaren,
tel. 02957/79-0

□ **Dell** - Performanțele unui 486 la prețul unui 386

Firma Dell este prezentă pe piață cu o nouă serie de calculatoare interesante. Este vorba despre seria Dell 486P/50, P/33, P/25, P/20 și P/16. Calculatorul Dell 486P/16 - 16 MHz i486SX, 4 MB RAM, 120 MB 17 ms harddisk, unități de discetă de 3,5 și 5,25 inch, monitor Super VGA, 14 inch, 1024 X 768 interlaced, inclusiv MS-DOS 5, Microsoft Windows 3 și Microsoft mouse - costă doar 2.200\$ în timp ce un 386/33 costă cca. 2.150\$. De remarcat că viteza unității centrale a

calculatorului Dell 486P/16 este cu 40% mai mare decît a unui Dell 333P la 33 MHz (test Landmark)!

O altă ofertă tentantă a firmei Dell este un Notebook color, Dell 325NC. Acest portabil color dispune de un procesor i386 SL, la 25 MHz, 64 KB cache, 4 MB RAM, expandabil pînă la 12 MB, 80 MB harddisk, lector de dischete de 3,5 inch, monitorul are diagonala de 9,25 inch și poate afișa 16 culori în modul 640 X 480, și pînă la 256 de culori în modul 320 X 200. Bateriile NiMH permit o funcționare independentă de pînă la 3 ore. Preț informativ: 4.000\$.

System Plus International,
București, str. Zăgazului 10, ap. 25,
tel/fax 90-791391.

□ **Intel** - Sosește 80586!

După cum a anunțat firma Intel, la mijlocul lui 1992 vor fi disponibile primele procesoare 80586. Acesta este un procesor pe 64 de biți, care conține peste 3 milioane de tranzistoare, deci cu 60% mai mult decît un 486. Intel afirmă că procesorul va fi de la patru pînă la cinci ori mai rapid decît un 486/33 MHz, și va atinge 100 Dhrystone MIPS; pentru comparație amintim că un 486 atinge 41 Dhrystone MIPS. Structura internă a lui 586 corespunde cu tehnologia Superscalar RISC cu suport multiprocesor extins, astfel încît acest procesor va fi primul cip PC 100% compatibil RISC.

Intel GmbH, Domacher Straße 1, 8016 Feldkirchen.

Apple în România

Romanian Computer Systems și Delta Design au organizat în București, la sala AGIR, în perioada 11-15 mai, o interesantă expoziție Apple. Vă reamintim membrii familiei Apple:

Model	Procesor	RAM min./max.	Hard disk	Apple SuperDrive	Porturi SCSI/ADB	AppleTalk/LocalTalk	Alte caracteristici
Macintosh Classic	8 MHz 68000	1 MB - 4 MB	40 MB	da	da	incorporat,	
Macintosh Classic II	16 MHz 68030	2 MB - 10 MB	40 sau 80 MB	da	da	incorporat	intrare/ieșire sunet
Macintosh LC	16 MHz 68020	2 MB - 10 MB	40 sau 80 MB	da	da	incorporat	emulare Apple IIe, intrare/ieșire sunet
Macintosh IIfx	20 MHz 68030	3 MB - 17 MB	40 sau 80 MB	da	da	incorporat	opțional NuBus sau 030 Direct Slot cu 68882
Macintosh IIfx	25 MHz 68030/68882	5 MB - 32 MB	80 sau 160 MB	da	da	incorporat	trei sloturi NuBus, RAM cache
Macintosh IIfx	40 MHz 68030/68882	4 MB - 32 MB	80,160 sau 400 MB	1 - 2	da	incorporat	șase sloturi NuBus, RAM cache
Macintosh Quadra 700	25 MHz 68040	4 MB - 20 MB	80,160 sau 400 MB	da	da	incorporat	două sloturi NuBus, Ethernet, intrare/ieșire sunet
Macintosh Quadra 900	25 MHz 68040	4 MB - 64 MB	80,160 sau 400 MB	1 - 2	da	incorporat	cinci sloturi NuBus, Ethernet, intrare/ieșire sunet
PowerBook 100	16 MHz 68000	2 MB - 8 MB	20 MB intern	extern	da	incorporat,	
PowerBook 140	16 MHz 68030	2 MB - 8 MB	20 sau 40 MB intern	intern	da	incorporat	intrare/ieșire sunet
PowerBook 170	25 MHz 68030/68882	4 MB - 8 MB	40 MB	intern	da	incorporat	intrare/ieșire sunet, cartelă fax/modem

Lectorul de dischete SuperDrive poate citi și scrie afit dischete format Macintosh cit și MS-DOS. Bateriile modelelor portabile (ultimele 3) permit o funcționare independentă pentru 2,5-4 ore. Merită menționat și faptul că pînă în toamna acestui an calculatoarele Apple vor "învăța" limba română. Firma Apple este reprezentată în România de firma Romanian Computer Systems, str. Luca Stroici, nr. 2, sector 2, București.

IBM își extinde familiile de calculatoare PS/2 și PS/1

Viena, 28 aprilie 1992 - IBM Eastern Europe își extinde familia sa de calculatoare PS/2 cu noi modele desktop și notebook care utilizează puternicul procesor al IBM-ului 386SLC. IBM a anunțat de asemenea primele sisteme care vor utiliza noul microprocesor Intel 486DX2 și adăugarea unor noi modele familiei de calculatoare PS/1.

"Odată cu introducerea acestor noi modele PS/2 și PS/1 ne-am extins paleta de produse atât în familia sistemelor Micro Channel cât și în cea a celor AT bus," a spus Dilip Chandra, director general IBM Eastern Europe. "Credem că aceste sisteme oferă raporturi preț/performață excelente, la standardul de calitate IBM. Această tehnologie IBM oferă performanțe care nu pot fi depășite pe piața calculatoarelor 386SX."

Noile sisteme AT bus la 20 MHz, Personal System/2 model 35 SLC și 40 SLC caracterizează standardul 386SLC, oferind performanțe semnificativ îmbunătățite față de standardul 386SX, multe aplicații rulând cu până la 88% mai repede.

Puternicul cip 386SLC, proiectat și fabricat în Burlington, Vermont, SUA, dispune de 8 KB cache intern și de un cache controller intern. Performanțele acestuia cresc prin accederea datelor din cache-ul intern, de viteză foarte ridicată, în loc de accederea lor din memoria sistemului, ori de câte ori acest lucru este posibil, și prin optimizarea instrucțiunilor utilizate în mod curent. Ca rezultat, pe sistemele SLC rezultatele se vor observa mai rapid decât pe majoritatea sistemelor 386DX-25MHz.

PS/2 Model 35 SLC se livrează cu harddisk-uri de 40 MB sau de 80 MB, iar modelul 40 SLC cu disc de 80 MB. Toate trei modelele sînt livrate în mod standard cu un lector de dischete de 2,88 MB.

Sînt anunțate noi modele PS/1

IBM a anunțat, de asemenea, și modelul PS/1 Pro 386 SX care extinde familia de produse PS/1 cu sisteme mai puternice și cu mai multe facilități de extensie, care să satisfacă necesitățile utilizatorilor atât pentru munca la domiciliu cât și la birou.

Toate modelele PS/1 Pro includ un procesor 386SX/20 MHz, 2 MB memorie extensibilă pînă la 16 MB, trei slot-uri de extensie AT bus, grafică VGA pe 16 biți și monitor color 14 inch incorporate, tastatură cu 102 taste și mouse. Pe harddisk sînt livrate soft-urile: IBM DOS 5.0, Microsoft Works 2.0 și Microsoft Windows 3.0. PS/1 Pro este disponibil în trei variante: cu harddisk de 40 MB, 80 MB sau 129 MB, și capacități de extensie pentru un al doilea disc.

Un nou notebook SLC

IBM își extinde și producția de modele notebook, odată cu anunțarea sistemului PS/2 Model N51 SLC, primul portabil care profită de tehnologia SLC. Dezvoltat și fabricat de IBM, procesorul SLC/16 MHz este de pînă la 75% mai rapid decât un procesor standard 386SX/16 MHz, oferind o putere de calcul echivalentă cu cea a unui procesor 386SL/25 MHz.

PS/2 Model N51 SLC dispune de 2 MB RAM, expandabil pînă la 10 MB, un harddisk de 80 MB, un lector de dischete de 2,88 MB, un monitor LCD (Liquid Crystal Display) VGA cu 32 de nuanțe de gri, și două baterii, nichel metal hidrid, o tastatură cu 85 de taste, și un port de extensie Micro Channel.

Capacitatea mare de stocare și puterea procesorului recomandă utilizarea modelului N51 SLC sub sisteme de operare avansate, cum ar fi OS/2 sau pentru rularea unor aplicații care să beneficieze de viteză procesorului, cum ar fi aplicațiile numerice sau grafice.

Sînt disponibile noi sisteme 486DX2

IBM a anunțat, de asemenea, două noi modele 95 tower și două modele 90 desktop, care utilizează procesorul Intel 486DX2. Acest nou cip utilizează o tehnologie de "dublare a vitezei" care permite ca intern procesorul să lucreze la 50 MHz, în timp ce extern să se lucreze doar la 25 MHz. Rezultatul este un sistem cu performanțe cu pînă la 75% mai ridicate în mediul stațiilor de lucru, și cu performanțe cu pînă la 18% mai ridicate în domeniul fileserver-elor, comparate cu un procesor standard 486SX/25 MHz.

ROMORGADATA '92

Romexpo S.A. și IEG Solingen au organizat în perioada 5-8 mai 1992, la Complexul expozițional din București, expozițiile specializate ROMMEDICA, ROMPHARMA, ROMCONTROLA și ROMORGADATA. Deși s-a pus accentul în primul rînd pe aparatura medicală, farmacie și tehnică de măsură și control, firme mari (și) în tehnica de calcul avînd standuri numai de interes medical (Hewlett-Packard, Toshiba, Philips), și-a avut și secțiunea de tehnică de calcul și informatică ponderea ei, printre expozanți numărîndu-se: Adisan, B.F.D. Import-Export SRL, Electric Business SRL, ICI, Intercom SRL, Kontrax Birotica SRL, KT Technology Ltd., DHS Merger & Partner GmbH, Misaco SRL, Rom Team Solutions SRL, Scop, SINTA SA, SIS, System Plus SRL, Tektronix.

CERF '92

În perioada 11-15 mai, 1992, la Sala Polivalentă din București a avut loc expoziția de calculatoare și produse electronice CERF '92 (Computers and Electronics Romanian Faire), expoziție organizată de Comtek Expositions, Inc. SUA și sponsorizată de firma Computerland București. Expoziția, o primă încercare de rafinare a profilului, a reunit nume de marcă din domeniul tehnicii de calcul și informaticii: Alfa Bit SRL, Computerland București, DHS Merger & Partner GmbH, Trend SRL, ICE Felix Computers, Innovator SRL, Laser Computer Romania SA, Lecom SRL, Misaco, Rom Control Data, Romtrust Inc., Style Impex SRL, Sysgraph Computergraphik GmbH, Unisys, System Plus International, Westech Co. SRL, Universys SRL, Rom Team Solutions.

THE deal

in p.c. field

H A R D W A R E

S O F T W A R E

C A D - C A M

D E S K T O P

P U B L I S H I N G

T e c h n i c a l
s u p p o r t

T R A I N I N G

C o n s u l t i n g



A&C
INTERNATIONAL S.A.



Tel 40-0-53.53.15.

Puterea unei workstation la nivel DOS

Washington, D.C., 31 martie 1992: Quarterdeck Office Systems, Inc. (NASDAQ: QDEK) a făcut cunoscut, în cadrul expoziției Federal Office Systems Expo (FOSE), faptul că s-a început livrarea produsului DESQview/X. Produsul permite utilizarea unora dintre facilitățile existente până acum numai pe stațiile de lucru (workstations) atât pe DOS-PC-uri, cât și pe PC-uri 386 individuale sau cuplate în rețea. DESQview/X, al treilea produs Quarterdeck din seria de produse multitasking, este un program multitasking și multifereastră grafic. Printre caracteristicile sale se numără, printre altele, un desktop grafic, fonturi scalabile, macrouri de tastatură și meniuri adaptabile, transfer de date și remote computing. O caracteristică importantă a lui DESQview/X este facilitatea de a rula atât programe DOS în mod text, cât și programe grafice Microsoft Windows în ferestre mici sau "remote" pe un alt PC DESQview/X sau pe o X-Workstation.

DESQview/X este prima implementare client/server completă la nivel DOS a sistemului X-Window - un standard internațional pentru sisteme grafice cu tehnica ferestrelor și partajarea puterii de calcul. Pentru a putea atinge performanțele unei workstation pe lângă facilitatea de a rula simultan programe DOS și Microsoft Windows (multitasking), DESQview/X conține și sistemul grafic și protocoalele de rețea X-Window și, suplimentar, tehnologia "shared" DOS-Extender pe 16 și 32 de biți, a firmei Rational Systems.

Avantajele aduse de DESQview/X pentru PC-uri individuale sau cuplate în rețea:

- transfer de date - cut și paste pentru aplicații care rulează în diferite ferestre, indiferent dacă acestea lucrează REMOTE (la distanță) sau LOCAL;
- suprafață utilizator grafică cu "look and feel" tridimensional;
- meniuri și icon-uri ușor adaptabile;
- sistem avansat de gestiune a fișierelor și imprimantelor;

PACO SOFT

O surpriză plăcută la ROMORGADATA a constituit-o standul "Diviziei de informatică" a Palatului Copiilor. Sponsorizat de firmele Scop, Megalan și Logic, Palatul Copiilor a reușit ca de la resursele materiale și umane existente în septembrie 1990 (50 X HC 85, 2 X HC 85 extins, 2 X TIM S, soft Spectrum-Sinclair, 3 profesori de informatică) să crească până în septembrie 1991, resurselor hard inițiale adăugându-li-se o rețea Novell (11 PC-uri), o imprimantă laser HP Laserjet IIP și una matricială Epson FX 1050, iar celor soft adăugându-li-se produsele firmei Borland. Divizia, care este o asociație care nu-și propune realizarea de profit, lucrează cu cca. 1500 de copii, în diverse stadii de pregătire: Basic începători, Ba-

sic avansați, C începători, C avansați, rețele de calculatoare, cei 5 profesori de informatică și 8 colaboratori fiind insuficienți pentru acoperirea necesarului. În cadrul expoziției au fost prezentate mai multe aplicații în sfera rețelelor locale de calculatoare, programele (sursă și executabile) fiind distribuite gratuit celor care aveau la îndemână o dischetă goală, după terminarea expoziției aceste programe putând fi obținute de la firmele sponsorizatoare. Enumerăm câteva dintre aceste programe:

PACOMail, pachet de poștă electronică, autor: Șerbu Ciprian, clasă a VIII-a

PACOfact, utilitar care permite comunicarea între stații în regim conversațional, autor: Gălcă Mihnea, clasă a VIII-a

PACOprius, print server, autor: Fulger Dan, clasă a IV-a

La livrare DESQview/X mai este însoțit de:

- DESQview/X Application Manager, un desktop grafic și un manager de programe;
- DESQview/X File Manager, un sistem de gestiune a fișierelor locale sau remote, DOS sau neDOS;
- DESQview/X Icon Editor, un instrument grafic pentru crearea și editarea de icon-uri pixmap și bitmap;
- Adobe Type Manager (ATM) for DESQview/X;
 - un Outline Font Manager cu 13 fonturi Typ 1 pentru DESQview/X și programele grafice pentru sistemele X-Window;
 - permite și ferestre DOS scalabile pentru programe DOS în mod text.

După cum afirmă Gary Pope, vicepreședinte în domeniul proiectării și unul dintre membrii fondatori ai firmei Quarterdeck, până acum la nivel DOS lipsea un sistem integrat de ferestre conceput pentru comunicarea grafică. Prin portarea sistemului X-Window la nivel DOS, utilizatorii dintr-o rețea pot accede la întregul X-Software, care rulează pe workstations Unix sau VAX sau pe calculatoarele mainframe.

Împreună cu DESQview/X se mai livrează **bestseller-ul** Quarterdeck QEMM-386 (un manager de memorie), Manifest, un instrument de analiză a sistemului și memoriei și un manager de rețea pentru rețele Novell Netware și NetBIOS (575 DM). Opțional se mai livrează un manager de rețea pentru TCP/IP, DESQview/X OPEN LOOK Window Manager și DESQview/X OSF/Motif Window Manager.

Cerințe hard: PC 386 compatibil IBM sau mai mare; PC sau MS-DOS 3.3 sau mai mare sau DR DOS 6; 4 MB RAM sau mai mult, cartelă grafică EGA, VGA, SVGA sau DGIS cu 256 de culori; harddisk de cel puțin 40 MB. În prezent se lucrează și la o versiune pentru PC-uri 286.

Quarterdeck Office Systems GmbH, Willstätter Straße 15, D-4000 Düsseldorf 11, tel. (49) 211-59790-0, fax: (49) 211-594126.

PACOfix, utilitar care permite unui utilizator să afle toate directoarele în care se află fișiere care îi aparțin, autor: Romășcanu Vlad, clasă a VII-a

PACOfact, pachet de utilizare duală (profesor/elev) destinat procesului de instruire. Autor: Damian Ionuț, clasă a IX-a.

PACOfact, pachet dual de tip compile & link server care permite unui utilizator de la o stație mai puțin dotată (de ex., un XT) să lanseze o compilare și/sau o linkare dificilă pe o stație mai dotată (de ex., un 386 cu 4MB RAM), autori: Eșeanu Costin, clasă a VII-a și Băloiu Liviu, clasă a VII-a

PACOfact, generator de propoziții pornind de la o matrice de cuvinte, autor: Pope Sabin, clasă a IV-a

Palatul Copiilor - Divizia de informatică
B-dul Tineretului nr. 8-10, sector 4
tel. 754900, 754083, fax 757135, 150838

Fractal Design Painter 1.2b2

A picta ca van Gogh

Pînă acum a existat o diferență imensă între pictura pe pînză și cea pe ecran. Cu programul Painter al Corporației Fractal Design acest lucru se va schimba radical: programul simulează tehnicile naturale de pictare pe baza algoritmilor de fractali.



Painter for Windows

Prima dată Painter a fost prezentat la expoziția de toamnă Comdex de la Las Vegas, de atunci a fost găsit și un distribuitor german, și anume Letraset și apariția versiunii Windows a fost anunțată pentru următoarele săptămîni.

Suprafața și modul de descriere

Imediat după pornirea programului, acesta se prezintă printr-o suprafață care amintește de outfit-ul versiunii Macintosh: toate sculele și toți parametrii sînt grupați în toolbox-uri grafice foarte perfecționate, care nu posedă capetele de tabel tipice Windows, ci provin direct de la suprafața Mac. Aceste toolbox-uri pot fi dispuse liber pe ecran sau ele pot fi închise provizoriu.

Painter se deservește fie cu un mouse fie cu o tabletă grafică. În "pachetul" de livrare se află pe lîngă un driver special de mouse, care ajută la o transmitere mai precisă a mișcărilor mouse-ului, și drivere pentru digitizorul SD-510 și SD-42x al firmei Wacom și pentru tableta de desen Calcomps Drawing-board II. Primele încercări cu SD-510 arată că doar în legătură cu acest aparat de intrare programul își poate dezvolta întreaga capacitate creatoare. Utilizatorul lucrează cu un creion fără cablu și fără baterie, care permite pe de o parte o conducere mai naturală a pensulei decît cu un mouse. Pe de

altă parte acesta (creionul) este sensibil la presiune, grosimea culorii se schimbă în funcție de apăsarea, care se aplică de utilizator creionului.

Scule de pictat, hîrtie și culori

Toate funcțiile oferite de acest software folosesc simulării exacte a sculelor și tehnicilor naturale de pictat.

Elementul central este o colecție de scule, care conține printre altele: pensulă de ulei, marker de text, vîrfuri de cariocă și ceară, cretă, cărbune și chiar un airbrush. Există chiar și funcții care simulează stilul lui van Gogh, respectiv al lui Seurat. Toate sculele pot fi manevrate, avînd în vedere modul lor de funcționare, prin diferiți parametri și în cele mai multe cazuri ele reflectă o imagine foarte precisă a realității. De exemplu, se poate observa că un marker de text galben devine cu timpul tot mai murdar, dacă cu el se colorează peste culori mai închise. Painter face diferență între acuarele umede și uscate, el cunoaște procedeul de amestec și de acoperire a diferitelor tipuri de culoare și ține cont de structura hîrtiei folosite la aplicarea culorii.

Felul hîrtiei se alege în altă boxă de dialog. Utilizatorului îi stau la dispoziție diferite sortimente de pînză și hîrtie, dar se pot alege și materiale textile, tapete din fibră dură și alte materiale. Aceste setări se folosesc apoi de către

Painter la folosirea diferitelor scule, cum ar fi cărbune sau vîrf de ceară.

O imagine terminată poate fi acoperită complet cu o textură dorită de hîrtie cu ajutorul unei funcții speciale, iar programul mai are în vedere și direcția de incidență a luminii asupra tabloului.

Există și o dependență directă între sculele de pictat folosite și paleta de culori Painter. Cu toate că oricînd poate fi aleasă orice culoare din paleta de culori, în plus programul mai oferă și o altă paletă, din care se pot alege cîte 15 culori tipice pentru o sculă de pictat.

Alte funcțiuni

Painter nu se pretează doar pentru artiști, ci oferă și pictorului mai puțin versat unele funcții interesante. Există de exemplu posibilitatea de a încărca fișiere bitmap și de a le pune sub un fel de hîrtie pergament. Dacă apoi se pictează cu o așa-numită sculă de clonare deasupra acestei hîrtii de tracing, Painter combină funcționalitatea sculei cu coloristica imaginii încărcate, se dă astfel posibilitatea de a copia o imagine.

Se păstrează în acest timp întregul control asupra mișcării pensulei, și nu este nevoie să urmărim exact culoarea imaginii originale.

Auto-Cloning-ul merge cu un pas mai departe. Această funcție amortizează procesul descris mai sus, el preia deci și pictatul propriu-zis. Urmează ca în final imaginea să

fie pusă pe o textură de hîrtie corespunzătoare, și deja tabloul generat de calculator este terminat.

Mai există bineînțeles și alte funcții ajutătoare, cum ar fi o rutină de umplere, posibilitate de a genera șabloane și o lupă pentru lucrări mai precise.

În schimb, s-a renunțat la funcția text - aceasta nu a avut-o la dispoziție nici chiar van Gogh.

Concluzie

Painter este unul dintre programele de grafică cele mai inovative, elaborate în ultimii ani. Astfel, graficianului ce utilizează un PC îi stă la dispoziție un program autentic de pictare, după ce Deluxe Paint a murit de mult, iar produsele Rix și ZSoft s-au specializat în prelucrarea de imagini. Singura problemă: cel ce dorește să se bucure de Painter, acela are nevoie de un calculator destul de rapid cu o memorie de lucru disponibilă de cel puțin 6 MByte și de o cartelă Super-VGA.

Letraset Deutschland GmbH
Mergenthaler Straße 6
6000 Frankfurt 60
Tel. 069/4209940

Alpha - o nouă tehnologie a procesoarelor

Ultimul chip RISC prezentat de Digital Equipment nu poate fi întrecut încă din punct de vedere al vitezei, al scalabilității și a multitudinii sistemelor de operare care îl pot utiliza.

Digital Equipment a dezvoltat sub denumirea seacă de "21064" secretul păzit cu strășnicie timp de 12 ani de cercetare și dezvoltare: un procesor RISC de 64 de biți, care, la o frecvență de tact de 150 MHz, dezvoltă o putere maximă de 300 MIPS. Cei din branșă, mai ales cei de la firmele Mips și DEC, ambele membre ACE, au rămas surprinși. Nu de multă vreme cei din consorțiul ACE s-au pus de acord asupra unui singur procesor "R4000" al firmei Mips, pentru sistemele de operare ACE, "Windows NT" și "Open Desktop". Din punct de vedere al performanțelor și perspectivelor de viitor, CPU-ul firmei DEC este cu mult înaintea lui Mips.

Procesorul 21064, denumit pe scurt "Alpha" reprezintă prima implementare a unei arhitecturi care se poate afirma câțiva ani buni datorită unei adaptabilități crescute. Implementări ulterioare pot fi executate la frecvențe de tact mai ridicate. Din auzite, există deja un model de laborator care funcționează la 300 MHz.

Pentru fabricare lui Alpha, care s-a realizat printr-o tehnologie de 0,75 micrometri-CMOS-4 cu patru planuri de metalizare, a fost necesară construirea unei fabrici de 460 milioane de mărci în Scoția. Costurile de dezvoltare probabil au fost și mai ridicate.

Procesorul veritabil de 64 de biți, cu regiștri de 64 de biți, cu bus-uri de date de 64 de biți, cu adrese de 64 de biți și cu operații pe 64 de biți este primul model al unei noi familii de procesoare. În timp ce alte CPU-uri, care lucrează intern tot cu 100 MHz, reprezintă apusul unei familii, modelul firmei DEC la o frecvență de tact de 150 MHz are foarte bune perspective de viitor.

S-ar putea discuta despre paleta de aplicații posibilă, începând de la Palmtop pînă la Supercomputere, deoarece la o putere absorbită între 20 și 30 de wați orice producător de motherboard-uri pentru calculatoare Desktop ar fi tentat să-l utilizeze, nemaivorbind despre producătorul unui Palmtop. Pe o suprafață de 17x14 milimetri se găsesc 1,7 milioane de tranzistoare. Dintre acestea memoria Cache și alte memorii ocupă cca. 840.000 de tranzistoare. Pentru asigurarea logicii au mai fost necesare approxi-

mativ 800.000 de tranzistoare. Partea considerabilă de logică se opune unei miniaturizări mai mari a cipului.

Procesorul este astfel conceput încît programele Unix și OSF/1 pot fi folosite în același fel ca și software-ul VMS propriu.

Arhitectura are în vedere atît sisteme multiprocesor cît și sisteme de calculatoare paralele, care se bucură de o tot mai mare răspîndire.

Alpha este o arhitectură tipică Load/Store-RISC. Înseamnă că toate operațiile, cu excepția operațiilor Load/Store, se efectuează între regiștri. 32 de regiștri Integer și 32 de regiștri Floating-Point au fiecare lățimea de 64 de biți.

Toate comenzile au o lățime de 32 de biți tipice tuturor arhitecturilor RISC, avînd patru formate diferite și un cod operand de 6 biți.

O caracteristică care deosebește arhitectura Alpha de altele este instrucțiunea denumită PAL-Call. Este vorba de fapt de cîteva zeci de comenzi complexe care tratează întreruperile, stările de excepție, adresările virtuale sau task switching. Aceste subprograme pot fi adaptate foarte precis la un anumit sistem de operare. Se poate astfel rula VMS dacă se alege cea versiune PAL-Call, care este adaptată multora dintre posibilitățile sistemului de operare VAX. Același lucru este valabil și pentru OSF/1 sau Windows NT. Astfel Alpha devine interesant pentru multe sisteme de operare.

Pentru procesoarele care se orientează după arhitectura procesoarelor RISC, compilatorul folosit este de mare importanță, deoarece acesta trebuie să producă întotdeauna un șir de comenzi mașină prelucrabile.

DEC a dezvoltat un compilator care se compune dintr-o parte Front-End și una Back-End. Cu aceasta ar fi susținute chiar și sisteme tolerante la erori și platforme în timp real.

Cele mai importante blocuri de funcții ale procesorului Alpha sînt o unitate Integer de 64 de biți și o unitate Floating-Point tot de 64 biți. Prima dintre ele lucrează cu un Pipeline de 7 trepte iar cea de-a 2-a, chiar cu una de 10 trepte. Cipul este completat de o unitate Load/Store sau de adrese, care răspunde de handling-ul ramificațiilor programului și de salturi.

Datorită arhitecturii Superpipeline în mai multe trepte se asigură pe ciclu cîte o instrucțiune fiecărei funcții. Deoarece

în paralel fiecare Pipeline emite cîte o comandă/tact în total se execută 2 comenzi pe un tact. Doar înmulțirile Integer și împărțirile Floating Point nu sînt total sprijinite de Pipeline. Dintre cele patru tipuri de comenzi de bază întotdeauna se poate executa o pereche de comenzi: Load/Store, operații Integer, operații Floating-Point și operații de ramnificare/salt. Comenzile Load/Store pot prelucra atît 32 cît și 64 biți.

Unitatea Floating-Point susține atît tipuri de date standard VAX cît și cele ale standardului IEEE. Cu excepția împărțirii unitatea predă pe ciclu un rezultat pe 64 de biți.

Pe cip sînt integrate două memorii cache pentru date și pentru instrucțiuni, de cîte 8 KByte. Memoria cache pentru date este executată ca un cache "Write through" și "Direct mapped" cu blocuri de 32 biți. Și cache-ul de comenzi este "Direct mapped" și are aceeași dimensiune. Se susține un cache secundar printr-o interfață programabilă pînă la o mărime de 8 MByte.

Generatorul de tact integrat este responsabil de tactul relativ înalt al procesorului, dar și de frecvențele de tact programabile CPU tact/2 și CPU tact/8.

Lățimea busului de date poate fi aleasă între 64 și 128 biți la frecvențe de tact cuprinse între 75 și 18,5 MHz. Un numărător ON-CHIP măsoară mereu puterea CPU-ului și a sistemului. Spațiul fizic al adreselor măsoară în prezent 34 biți, cel virtual doar 43 biți, dar ambele au fost încercate la 64 biți. Motivul acestei restricții este numărul de conexiuni al cipului (de pini) de "doar" 431. Din acest punct de vedere următoarele versiuni vor oferi mai multe posibilități.

Dar și cu actualul mod de realizare memoria adresabilă de cîteva mii de GBytes este astronomic de mare și ea se dublează cu fiecare pin de adresă. Creșterea puterii de calcul nu se va mai putea exprima pe viitor doar prin frecvența de tact, trebuie avut în vedere, și numărul de instrucțiuni pe care le poate executa procesorul simultan. Procesorul 21064 poate executa 2 instrucțiuni pe ciclu, și se preconizează o rată de 8 instrucțiuni pe ciclu.

Primele chipuri de acest tip vor fi livrate în următoarele cîteva luni, dar cantități notabile pot fi așteptate doar la începutul anului 1993. El va costa 3.400\$ bucata. Angroșiștii îl pot cumpăra la un preț de aproximativ 1.500\$ bucata.

THE

right option

MX®
COMPUTERS

with

ATI Graphic Cards
MAG Monitors



in hardware

 **CalComp**

Panasonic

 **HEWLETT
PACKARD**



A&C
INTERNATIONAL S.A.



Tel 40-0-53.53.15.

Head Up Display

Tehnologia de proiecție pentru afișajele militare montate pe cască (HMTV - Helmet Mounted TV) a evoluat continuu de-a lungul a peste 30 de ani, dar numai recent a fost adaptată cu succes la utilizări civile.

Acum, sistemele personale de afișaj prin proiecție, miniaturizate, ieftine, își găsesc aplicații într-o gamă de utilizări comerciale și industriale în pină expansiune.

Ecranele virtuale sînt sisteme de proiecție a imaginii ce permit utilizatorului să privească într-un dispozitiv mic, situat aproape de ochi și să vadă o imagine (a informațiilor de afișat) de dimensiune mult mai mare și situată la o distanță aparentă mai mare. Cu respectul dispozitiv de mici dimensiuni montat pe cască sau pe "ochelari", afișajul se află permanent în câmpul vizual personal al utilizatorului, care astfel este liber să efectueze alte sarcini.

Ecranul virtual montat pe cască, bazat pe un tub catodic (CRT) în miniatură pentru aplicații militare critice, a asigurat HUD ("afișaj cu capul sus") ce permite piloților și conducătorilor de tancuri să supravegheze instrumentele indicatoare și sistemele de achiziție a țintelor, fără a întrerupe contactul vizual direct cu scenele reale de pe câmpul de luptă.

Ce este HUD?

Head Up Display, "afișaj cu capul sus". Este un dispozitiv utilizat inițial la avioanele militare, dar înțlnit acum din ce în ce mai des în cockpit-ul avioanelor civile.

Permite afișarea prin proiecție a informațiilor vitale pentru zbor, care se găsesc astfel regrupate în fața ochilor pilotului, ceea ce evită aplecarea în permanență a privirii pentru consultarea tabloului cu instrumentele de bord.

La avioanele de luptă, se mai proiectează în plus diferite imagini reale sau sintetice (hartă electronică) ale peisajului survolat sau ale țintelor vizate (în infraroșu, radar).

Imaginea este proiectată deasupra bordului, în conturul câmpului vizual al pilotului, iar distanța aparentă de focalizare pentru ochi este infinită, astfel că nu solicită modificări ale acomodării ochilor (cum este cazul cînd se

schimbă privirea de la mediul ambiant la panoul de bord)

Primul ecran virtual practic, bazat pe tub catodic, a fost introdus de firma Hughes Aircraft în 1958. Aproape toate performanțele ulterioare s-au bazat doar pe îmbunătățirea afișajelor miniatură cu tub catodic, și-n consecință au fost limitate de caracteristicile lor inerente de cost, rezoluție și dimensiuni.

Au existat și sisteme prototip, nebazate pe tub catodic, cele mai multe cu cristale lichide microminiatură și sisteme optice de mărire. Dar acestea sînt dificil și costisitor de fabricat, iar performanțele lor n-au fost la fel de bune ca la cele cu tub catodic.

Un nou tip de dispozitiv de afișare prin proiecție.

În 1990, o firmă din New England a introdus un nou tip de dispozitiv, proiectat special pentru piața civilă, nu militară, și bazat pe o tehnică diferită. Denumit Private Eye, este fabricat de Reflection Technology Inc., din Waltham, Massachusetts și promite să înlocuiască afișajele cu tub catodic în toate aplicațiile de ecran virtual monocrom, cu excepția celor foarte critice.

În timp ce sistemele precedente utilizează un afișaj bidimensional și un sistem optic fix, noul dispozitiv folosește o rețea unidimensională (baretă de LED-uri) pentru afișaj și o optică cu baleiere.

Principial, dispozitivul se compune dintr-o baretă de LED-uri (diode electroluminescente) cu electronica de comandă aferentă, un obiectiv de proiecție și o oglindă mobilă, montată pe arcuri, ce reflectă lumina spre ochiul utilizatorului.

Imaginea de afișat este codată digital și trimisă la electronica de comandă printr-o interfață serie, sub forma unei hărți de biți (bit map). Harta de biți este stocată în memoria de ecran și organizată în format de coloane de către o rețea de porți la comandă (circuit specializat, cu rețea de porți logice interconectate discreționar, ce realizează algoritmul dorit).

Mecanismul de bază al afișajului constă dintr-o rețea liniară, monolitică de LED-uri - 280 în linie - un obiectiv ce

mărește imaginea și o oglindă oscilantă, cu contragreutate mobilă. Oglinda este comandată de un motor miniatură cu două înfășurări. Bobinele sînt controlate de un sistem de servo-reglaj în buclă închisă. Oglinda, montată pe arcuri, este echilibrată cu o contragreutate mobilă, pentru a absorbi cea mai mare parte a vibrațiilor mecanice parazite.

În fiecare moment utilizatorul, privind în oglindă, vede proiecția rețelei verticale de LED-uri, într-o poziție orizontală corespunzătoare unei coloane de date de afișat. Oglinda este balansată pe orizontală cu o frecvență fixă, de rezonanță, baleind un arc de 15 grade.

Controlerul de afișaj trimite datele de coloană unei perechi de cipuri de registre de decalaj CMOS, fiecare comandînd 140 de LED-uri. LED-urile, în sincronism cu mișcarea oglinzii oscilante, afișează coloana de date de imagine corespunzătoare poziției instantanee a oglinzii.

Pentru a sincroniza baretă de LED-uri cu poziția oglinzii sînt montate o sursă de lumină staționară și un fotoreceptor. O plăcuță (tab) de pe spatele oglinzii permite luminii de referință să ajungă la senzor atunci cînd oglinda se apropie de deflexia maximă, generînd astfel un puls de sincronizare (semnal de sfîrșit de cursă/linie).

Rata de baleiere pe orizontală a oglinzii, de 50 Hz, este destul de ridicată pentru ca utilizatorul să vadă, fără pîlpîire, o imagine virtuală a întregului afișaj bidimensional, cu o rezoluție de 720 (H) x 280 (V) pixeli.

În cazul afișării de texte, dispozitivul asigură afișarea a 25 de rînduri normale, a cîte 80 de caractere. Imaginea virtuală, roșie pe fond negru, are dimensiunea aparentă de 12" (31 cm) și apare la circa 2 ft (0,66 m) de ochiul utilizatorului.

Un reglaj surprinzător de important pentru o viziune subiectivă optimă este ajustarea fină a clarității imaginii. De fapt, nu-i vorba propriu-zis de claritate, ochiul adaptîndu-se automat, ci de distanța aparentă de proiecție. Se realizează printr-un reglaj culisant al focalizării: ansamblul lentilelor-obiectiv gîlusează pe o șină, și poate fi reglat de

utilizator astfel ca imaginea proiectată să apară oriunde între 9" (23 cm) și infinit, față de ochiul utilizatorului.

Astfel, imaginea afișajului poate fi plasată la aceeași distanță cu obiectele fizice (reale) din câmpul vizual al utilizatorului, evitând oboseala ochilor datorată permanentelor schimbări de focalizare (reglare a distanței de punere la punct).

Considerente de încapsulare

Încapsularea este un factor esențial în simplificarea producției și reducerea costurilor. Un singur circuit integrat hibrid încorporează rețeaua (bareta) de LED-uri și driver-ele de curent asociate. Acest circuit hibrid, împreună cu controller-ul buclei servo a motorului, memoria de ecran și totalitatea componentelor discrete, este asamblat în tehnologia montării pe suprafață pe o placă de cablaj flexibil, ce se pliază într-o cutie mică de plastic. Tot dispozitivul măsoară 2,5x2,5x7,6 cm și cântărește circa 70 g. Consumul de energie este de 0,33W la +5V. Dispozitivul poate fi produs în serie mare, la un preț de cost mult mai mic decât cel al ecranelor virtuale cu tub catodic miniatură.

Dimensiunile și greutatea reduse ale corpului deschid perspective pentru numeroase aplicații potențiale. Astfel, fabricanții de calculatoare portabile (laptop) intenționează să ofere modele noi, bazate pe o variantă de "Private Eye". Se întocmește afișajul clasic (cristale lichide sau panou cu plasmă), real, cu un afișaj virtual montat pe headset (benziță, inel de susținere pe frunte, ca cel pentru oglinda oftalmologică) la cîțiva cm de ochi.

Aceasta permite operatorului aflat în tranzit să lucreze mai bine într-un spațiu restrîns, mai ușor și mai confortabil. Imaginea de pe ecranul virtual este permanent în câmpul vizual personal (și privat - nu-l vede altcineva) al operatorului.

În domeniul servis/mentenanță/depanare echipamente, afișajul virtual este deja folosit ca o parte a manualului de servis cu diagnostic asistat de calculator, pentru a optimiza eficiența (pleonasm?) reparațiilor.

Aici, pagina din manual referitoare la subansamblul în curs de inspectare sau depanare este proiectată în fața tehni-

cianului care lucrează. În consecință, tehnicianul poate lucra cu ochii numai pe echipament și cu ambele mâini libere, minimizîndu-se astfel riscul de accidente și erorile costisitoare. Aceasta este o capacitate importantă și promițătoare, în special atunci cînd munca de depanare trebuie efectuată într-un mediu cu grad ridicat de pericolozitate sau precar (?).

În domeniul depanării asistate (dar nu numai), Reflective Technology intenționează să folosească discurile laser (Compact Disk) drept mediu prin intermediul căruia volumele arhivate de manuale de servis și "cărți de electronică" vor fi rediate pe ecranul virtual miniatură.

Notă: Cu vreo 10 ani în urmă se prezenta la Teleenciclopedia "cartea electronică". Informațiile erau stocate într-o memorie moartă (ROM, EPROM) și afișate pe un ecran cu cristale lichide. Eu făcusem socoteala că densitatea de informație este mai mică decât la o carte clasică, și-i evident mult mai scumpă. Pe atunci prețul memoriilor era destul de ridicat, iar capacitatea maximă, de 64 K. În schimb, varianta cu CD-ROM și ecran virtual "Private Eye" pare promițătoare, atît sub aspectul capacității (200.000 pagini pe disc de 5") cît și al costului (sub 3\$ pentru CD-urile audio sau video).

Fax fără hirtie

O firmă comercializează curent un receptor facsimil fără hirtie, folosind o variantă a lui "Private Eye". Acest echipament poate stoca pînă la 25 de pagini de date transmise, care pot fi apoi parcurse (scroll) de operator și vizualizate pe un ecran virtual cu beneficiul suplimentar al unei depline întimități (nu mai poate să îți se uite cineva peste umăr pe ecran).

Notă: Aceasta nu înseamnă că în curînd toate faxurile vor fi fără hirtie. Noțiunea de "imprimantă fără hirtie" nu are sens. Sînt destule situații în care ai nevoie de o copie permanentă a imaginii de pe ecran, unde capacitatea de hard-copy este importantă. Dar există, desigur, domenii în care nu-i necesară o înregistrare permanentă, informarea trebuie să fie rapidă și confidențială, scop în care faxul cu ecran virtual (eventual racordat la rețeaua telefonică fără fir) este indispensabil.

"Reflective Technology" oferă de asemenea o placă - driver care lucrează cu calculatoarele personale sub MS-DOS și asigură interfațarea acestor echipamente cu produsele tip ecran virtual ale firmei RT.

În următorii 5 ani, țelul firmei de a lansa pe piață un afișor cu scală de griuri, în culori, cu rezoluția de 1024x1024 (1M pixeli), într-o capsulă de dimensiuni mai mici decît cea curentă. Această versiune viitoare va avea grafică complet automată, la un preț suficient de scăzut pentru aplicații electronice de larg consum - esențialmente un televizor de înaltă definiție (HDTV) într-o cutie de chibrituri.

Notă: Da, se poate considera un breakthrough (străpungere tehnologică, pătrundere revoluționară). Pînă acum, afișajul - ecranul - constituia un obstacol major în calea miniaturizării receptoarelor TV. Ce folos că toate funcțiile aparatului sînt îndeplinite de un singur cip, dacă ecranul este mare și consumă mult?

Iar variantele comerciale de ecrane cu cristale lichide color, precum Watchman al firmei Sony, sînt costisitoare și incomod de privit, la o diagonală de 4-5 cm. Ecranul virtual color ar fi o soluție efecace.

După Allen Becker, președintele firmei RT și inventatorul lui Private Eye, îmbunătățirea rezoluției este iminentă, dar ecranul virtual în culori reale (full-colour) este încă de domeniul viitorului. Modelul ce va apare la finele lui 1991 va oferi o rezoluție de 640x480 pixeli utilizînd tehnica de încapsulare existentă. Cel în culori va fi realizat atunci cînd vor fi disponibile comercial rețelele de LED-uri verzi și albastre, la fel ca cele roșii din prezent.

Pentru primul model s-au ales LED-uri roșii tocmai deoarece sînt ușor de obținut în rețele liniare și au eficiența luminoasă maximă. Oglinda basculantă va putea suprapune vizual pixelii celor trei culori fundamentale (RGB) ai rețelelor individuale de LED-uri pentru a produce culori multiple. Problema este că rețelele de LED-uri verzi și albastre mai au cîțiva ani pînă să fie disponibile comercial, în variante cu eficiență mare, fiabile și ieftine.

(Cristian Malide, ICCE)

Comunicarea om-mașină

Nici în Evul High-Technology nu sînt realizate încă toate dorințele oamenilor. O astfel de dorință încă nerealizată este și mașina ca partener de comunicare. În lumea science-fiction-ului relația triunghiulară om-mașină-comunicare a fost rezolvată de mult. În realitate, oamenii de știință se ocupă de decenii cu această problemă, succesele fiind sohibbătoare. Cine dorește să afle care este starea de fapt, va putea arunca, prin intermediul acestui articol, o privire atît asupra fundamentelor cît și asupra stadiului cercetărilor.

În anul 1952, IBM, a prezentat prima mașină care era capabilă să recunoască limbajul uman. Calculatorul uriaș, prezentat atunci, era în stare să recunoască numerele de la 1 la 10, citite cu voce tare, cu o precizie care atingea 97%. Această știre de senzație a stîrnit o euforie enormă în rîndul cercetătorilor modului de comunicare om-mașină. Se credea că, în puțini ani, tastatura va putea fi înlocuită cu un microfon și că operarea calculatoarelor se va putea realiza integral prin vorbire.

Acestei euforii i-a urmat curînd dezluzia, atunci cînd s-a constatat că algoritmul pentru recunoașterea unor propoziții complexe este cu mult mai complicat decît cel pentru recunoașterea a zece numere și necesită un efort mult mai mare.

Cercetările în domeniul recunoașterii vorbirii au fost date, mai mult sau mai puțin, uitării. La sfîrșitul anilor '60, un studiu de la IBM a determinat guvernul SUA să susțină un proiect de cercetare, ceea ce a dat un nou avînt cercetărilor. Astăzi, o mulțime de institute și universități studiază acest domeniu.

Cu toate că puterea de calcul a sistemelor actuale a atins cote gigantice, comparativ cu cea a anilor de început și s-ar putea crede că problema comunicării nu mai are mult pînă la a fi rezolvată, experții apreciază posibilitățile concrete de realizare mai realist. Ei sînt de aceeași părere atunci cînd estimează ceea ce este realizabil într-un viitor previzibil și ceea ce nu va exista niciodată: de exemplu un calculator care să poată înlocui în întregime interlocutorul uman.

Fundamentele recunoașterii vorbirii

În prezent, comunicarea om-mașină se bazează pe aparatele de intrare: tastatură și mouse, și pe mijloacele vizuale de ieșire: monitor și imprimantă. Ultimele pot fi înlocuite, între timp, fără probleme cu o ieșire bazată pe vorbire. Cu ajutorul unei plăci de extensie speciale și al unui software specific, PC-ul poate vorbi pe orice ton dorit, de exemplu cu o voce joasă, bărbătească, sau cu una înaltă, feminină. Dacă prietenele PC-uri vorbeau încă monoton electronic, sistemele de ieșire moderne surprind prin accentul perfect și intonația care respectă semnele de punctuație. Totuși este mult mai simplu să se redea vorbirea, decît să fie recunoscută și înțeleasă electronic.

În aceasta constă primul pas - și astfel se ivește și prima problemă - pentru a putea recunoaște electronic vorbirea; aceasta trebuie memorată (înregistrată) sub formă de text, respectiv trebuie transformată în informație digitală comprehensibilă pentru calculator. În timp ce un om nu are, cel mai adesea, probleme în a înțelege diferiți vorbitori, calculatorului nu-i este la fel de ușor. Principal, limba vorbită nu este altceva decît o succesiune continuă de unde sonore de frecvențe diferite. Astfel un cuvînt rostit poate fi reprezentat grafic înregistrînd variația frecvenței undelor sonore în timp.

Probleme în cazul diferiților vorbitori

Comparînd fonogramele rezultate în urma înregistrării aceleiași cuvînt rostit de mai mulți vorbitori, se observă că există diferențe enorme. Acest lucru se datorează faptului că, de exemplu, femeile vorbesc folosind frecvențe mai înalte decît bărbaii. Un alt factor care determină diferențele dintre fonograme este viteza de exprimare a vorbitorului. Datorită fiziologiei aparatului de vorbire uman, chiar și în cazul în care același vorbitor rostește de mai multe ori același cuvînt, vor exista diferențe între fonograme.

În plus trebuie luate în considerare și zgomotul din mediul în care se vorbește, care și el ajunge în microfon. De regulă, o rată de recunoaștere excelentă, de peste 95%, nu poate fi obținută

decît în condiții de laborator, într-o încăpere izolată fonic.

Omul recunoaște o propoziție chiar și atunci cînd unele cuvinte sînt articulate neinteligibil. Datorită cunoștințelor generale pe care le posedă, ascultătorul poate înțelege, fără probleme, sensul propoziției. Dacă nu s-ar întîmpla așa, atunci doi conaționali vorbind dialecte diferite, nu s-ar putea înțelege.

Dacă va exista vreodată o mașină care să poată recunoaște, la perfecțiune, vorbirea, atunci ea va trebui să dispună de o bază de date cu cunoștințe generale. Ea este necesară, printre altele, și pentru a permite înțelegerea "omofonelor". Omofonele sînt cuvinte care se pronunță la fel, dar care au semnificații diferite, cum ar fi, de exemplu, "mai". Un partener de conversație uman deduce semnificația din context.

Două semnificații și jumătate per cuvînt

În exemplele care urmează, datele statistice se referă la limba germană. Nu dispunem de date statistice referitoare la limba română. Dacă, întîmplător, acest articol va fi citit și de un specialist care să ne poată pune la dispoziție date statistice referitoare la limba română, ne vom bucura să revenim aducînd completările de rigoare.

Un cuvînt are, în medie, mai multe semnificații. Astfel în limba germană, "Dichtung", de exemplu, poate să însemne și "poezie", dar și "garnitură de etanșare" (a unui robinet). În limba română "mai" poate să fie adverb (mai ales), poate desemna o lună a anului, ficalul sau bucata de lemn cu care se bat rufele cînd se spală.

În cele din urmă, trebuie să se facă distincție între vorbirea cursivă, numită și vorbire legată, și vorbirea cu pauze semnificative între cuvintele individuale, așa-numita "vorbire cuvînt cu cuvînt". Recunoașterea "vorbirii legate" poate fi realizată mult mai greu decît recunoașterea "vorbirii cuvînt cu cuvînt". În cazul vorbirii legate este greu pentru sistem să recunoască cuvintele compuse. Sistemele de recunoaștere a vorbirii "cuvînt cu cuvînt" cer vorbitorului respectarea unor pauze clare între cuvinte. Vorbitorul este constrîns astfel să-și adapteze, în mod nenatural, modul său de rostire necesităților mașinii.

Dialogul vorbit om-mașină este practic utilizabil numai dacă dialogul poate avea loc în limbajul uzual. Pe cît de

THE choice

in software

 **AUTODESK**

Microsoft

 **NOVELL**

Lotus

 **Fox**

BORLAND

Aldus

COREL

Symantec

 **A&C**
INTERNATIONAL S.A.



Tel 40-0-53.53.15.

complexe sînt problemele de recunoaștere a vorbirii, pe aînt de complexe sînt și soluțiile utilizate în prezent.

În acest loc trebuie să mai atragem atenția că între procesele de "recunoaștere a vorbirii" și de "înțelegere a vorbirii" există diferențe fundamentale, chiar dacă cele două domenii sînt strîns legate între ele. "A recunoaște" înseamnă a converti în scris, sub formă de text corect d.p.d.v. sintactic, limba vorbită, deci sunetele ei individuale captate prin intermediul microfonului și transmise calculatorului. "A înțelege" înseamnă a decela sensul și conținutul textului recunoscut, pentru a reacționa în mod corespunzător. În cele ce urmează ne vom referi mai întîi la recunoaștere și apoi la înțelegere.

Există trei tipuri de sisteme de recunoaștere a vorbirii: "sisteme dependente de vorbitor", "sisteme independente de vorbitor" și "sisteme adaptive la vorbitor".

De la unda sonoră la modelul digital

Un prototip reprezentativ pentru stadiul actual al recunoașterii vorbirii este sistemul de recunoaștere a vorbirii "Tangora", al lui IBM (denumit după Albert Tangora, campion mondial la scris cu mașina de scris). Tangora a fost prezentat pentru prima dată în SUA, în 1986. O echipă de cercetători IBM lucrează din 1988 în Heidelberg la o versiune germană, alte echipe lucrînd la versiuni pentru limbile italiană și franceză. Tangora este un sistem de recunoaștere a vorbirii, dependent de vorbitor, destinat recunoașterii propozițiilor în cazul cuvintelor rostite izolat. Sistemul este funcțional pe calculatoarele PS/2 și RS/6000 (sistem RISC cu sistem de operare UNIX - IBM), cu plăci de extensie speciale, procesoare de semnal și memorii rapide. Explicațiile care urmează se bazează pe principiul de funcționare al sistemului Tangora. Principiul de funcționare al hardware-ului de conversie a unui semnal analog (vorbirea) într-o formă inteligibilă pentru calculator este simplu: undele sonore corespunzînd fonogramei vorbitorului ajung prin intermediul unui microfon într-un convertor analog/digital. Vorbirea este recepțională de acesta ca o succesiune continuă de frecvențe diferite care pot fi reprezentate ca semnale digitale.

Eșantionare la 10 ms

Semnalul sonor este divizat, în acest convertor, în intervale de timp de 10 ms. Bucățile astfel obținute conțin un amestec de frecvențe, care pot fi reprezentate în urma unei transformări Fourier sub formă de coeficienți Fourier. Din acest spectru Fourier este prelevat un vector de caracteristici alcătuit din 20 de elemente, care este comparat cu 200 de vectori prototip, specifici vorbitorului. Simbolul care caracterizează prototipul cel mai apropiat este atașat semnalului sonor. Astfel semnalul acustic este transformat într-o succesiune de simboluri, pe baza cărora poate demara procesul de recunoaștere.

Recunoașterea limbii

Pentru ca un calculator să înțeleagă o limbă, trebuie să dispună de o bază de date în care să fie definite sunetele corespunzătoare. Memorarea fonogramelor tuturor cuvintelor individuale nu are sens atunci cînd se dorește elaborarea unui sistem de recunoaștere a vorbirii pentru un vocabular vast (cca. 10.000 pînă la 20.000 de cuvinte), sistem care să fie independent de vorbitor și adaptiv. Fiecare vorbitor ar trebui să rotească în fața sistemului, de mai multe ori, fiecare cuvînt al vocabularului - operație care ar necesita foarte mult timp.

Mai avantajoasă este memorarea fonogramelor "fonemelor" decît a fonogramelor cuvintelor întregi. Fonemele sînt cele mai mici unități sonore ale limbii. Există cca. 70 de foneme, în limba germană fiind prezente cca. 60 dintre ele.

Pentru transcripția limbii germane pentru sistemul de recunoaștere a vorbirii Tangora sînt utilizate 60 de foneme: trei foneme specifice sistemului: (Null-Phon, Liniște, Delimitator de propoziție), 25 de vocale, 30 de consoane, două "offglides" (așa sînt denumite sunetele diftongice "au", "eu", "er").

În cazul soluției cu memorarea setului de foneme apare problema coarticulării: sunetele individuale sînt puternic influențate de sunetele învecinate, ceea ce face ca recunoașterea lor să fie dificilă și nesigură.

Din acest motiv este de preferat memorarea unor unități sonore mai mari. În cazul unităților mai mari - în practică fragmente de silabe - pot fi cuprinse și efectele datorate coarticulării și se poate ține seama de ele. În baza de date, cuvintele care urmează a fi recunoscute sînt definite ca succesiuni de astfel de unități sonore. Definițiile sînt date cu

ajutorul așa-numitelor "modele Hidden-Markov". Descris simplificate, un model Hidden-Markov constă dintr-un număr de stări, între care sînt posibile tranzițiile. Pentru fiecare stare sînt definite tranzițiile posibile. În plus, pentru fiecare stare există o anumită probabilitate de ieșire, care modelează probabilistic succesiunea de unități sonore (foneme). Dacă vorbitorul rotește un cuvînt, sistemul recunoaște, mai întîi, acest cuvînt ca pe o succesiune probabilă de unități sonore (corespunzînd stărilor în modelul Hidden-Markov). În timpul analizei, este căutat, în baza de date, cuvîntul definit ca model Hidden-Markov, care se potrivește cel mai bine cu înregistrarea.

În baza de date vor fi găsite mai multe sute de cuvinte care vor candida pentru identificare, dintre acestea fiind ales acela care corespunde cuvîntului rostit cu cea mai mare probabilitate.

Fast-Match

Prima metodă de căutare, denumită "Fast-Match", constă din mai mulți pași de verificare, în care numărul de cuvinte este redus în urma unor examinări din ce în ce mai exacte. Atunci cînd nu mai rămîn decît cca. 100 de cuvinte posibile, urmează următoarea etapă de analiză. În cadrul acestei etape se accede o culegere de texte, în sistemul Tangora, gigantică. Această culegere de texte a fost realizată, de colectivul de cercetători care se ocupă de sistemul Tangora, prin citirea automată (scanner + software OCR) din reviste economice. Momentan, culegerea de texte Tangora cuprinde cca. 50 de milioane de cuvinte, și crește în fiecare lună cu cca. un milion de cuvinte.

Această culegere de texte a fost realizată în scopul determinării statistice a frecvenței de apariție a unor succesiuni de două sau de trei cuvinte. În acest fel numărul de cuvinte care rămîn în competiție este redus din nou.

Cuvintele rămase sînt supuse unei analize denumită Detailed-Match. Analiza Detailed-Match se asortă cu analiza Fast-Match, dar ea nu mai operează cu fonemele modelelor Hidden-Markov ci cu unități sonore artificiale, așa-numite "foneme". Cu ajutorul acestora se poate efectua o verificare acustică mai exactă. După parcurgerea analizei Detailed-Match, fraza este, în fine, recunoscută. Versiunea germană a sistemului Tangora recunoaște, în prezent, 13.000 de cuvinte cu o probabilitate de peste 95%. În alte limbi sînt înțelase 20.000 de cuvinte cu o probabilitate de peste 95%.

De la a recunoaște, la a înțelege

Recunoașterea corectă a vorbirii este doar unul dintre aspectele importante pentru comunicarea în mod natural dintre om și mașină. Mult mai important, și în același timp mai greu, este să aduci calculatorul să înțeleagă conținutul, sensul, a ceea ce a recunoscut.

Pentru o mașină care trebuie să scrie după dictare poate să fie suficient să recunoască și să transcrie sub formă de text ceea ce i se spune. O comandă nu poate fi executată însă decât dacă i se înțelege semnificația. Imaginați-vă un robot care să se poată deplasa asemănător omului. El dispune de ochi (cameră de luat vederi) și urechi (microfoane). Omul care îl comandă ține legătura cu el prin radio, dispune de un monitor pe care se afișează ceea ce "vede" robotul și îi comunică în limbaj natural ceea ce are de făcut. În cazul comenzii "Du-te trei metri spre nord", robotul nu va avea probleme. Și nici în cazul în care i se va spune "Du-te la masă", cu condiția ca robotul să se afle într-o încăpere în care să existe o masă, să dispună de o cameră de luat vederi, de un program de recunoaștere a formelor și în baza lui de date să fie definit modul în care arată o masă.

Să presupunem că robotul este folosit ca reporter și că este dotat cu un aparat de fotografiat. El se găsește într-o încăpere în care se află mai multe persoane și primește comanda. "Fotografiază bărbatul cu aparatul de fotografiat". Soft-ul cu care este dotat robotul analizează propoziția și încearcă să-i determine sensul. Comanda de bază, "fotografiază" este ușor de înțeles; tot la fel și obiectul la care se referă această comandă: "bărbatul". Sensul ultimei părți a propoziției "cu aparatul de fotografiat" nu poate fi stabilit exact. Dacă în încăpere se găsește un singur bărbat, atunci este clar că robotul trebuie să fotografieze acest bărbat cu aparatul de fotografiat.

Ce se întâmplă însă atunci când în încăpere se află mai mulți bărbați și unul dintre ei are atârnat la gât un aparat de fotografiat? Fără îndoială că, în acest caz, robotul trebuie să-l fotografieze pe acel bărbat. Pentru ca un robot să poată înțelege sensul unei instrucțiuni (comenzi), nu este deci suficient să fie în stare să descompună o propoziție în părțile sale componente. El mai are nevoie, în plus, de o bază de cunoștințe în care, pentru exemplul de mai sus, să fie definit ce este un bărbat și ce operație este desemnată ca fotografiere. In-

strucțiunea "fotografiază" ar putea fi descrisă printr-o succesiune de operații de genul următor: "ia un aparat de fotografiat, îndreaptă obiectivul în direcția obiectului de fotografiat, ține-l nemișcat și declanșează". Dar nici această bază de cunoștințe nu este încă suficientă pentru exemplul anterior. Robotul trebuie să mai știe că instrucțiunea "fotografiază obiect" este suficientă și că este de prisos să se facă specificarea "cu ce" doar întotdeauna se fotografiază cu un aparat de fotografiat și nu cu altceva. Dacă specificarea suplimentară este prezentă, ea nu poate servi decât la a descrie mai exact obiectul vizat (bărbatul care are un aparat de fotografiat). Pentru o înțelegere perfectă robotul ar trebui să dispună de o bază de date care să corespundă cunoștințelor umane - dar, cu capacitățile actuale de memorare, așa ceva nu este posibil. Și chiar dacă ar exista capacitate de memorare suficientă, cine ar putea defini baza de cunoștințe? Robotului anterior descris i-ar fi cu siguranță greu să execute comanda care i s-a dat. Vinovat de această situație este cel care a dat comanda. El ar fi trebuit să-și exprime dorința mai puțin ambiguu, de exemplu: "fotografiază bărbatul care are atârnat la gât un aparat de fotografiat".

Confuzii fără limite

Chiar și în cazul celor mai mari baze de cunoștințe, cel care dă comanda este forțat de mașină să-și formuleze comenzile cu o precizie pe care omul n-o folosește decât în cele mai rare cazuri. El poate să-și formuleze verbal comanda, dar trebuie să țină cont de limitările mașinii. În aceste condiții, comunicarea om-mașină, în limbaj natural, nu are sens. O soluție acceptabilă, în acest domeniu, va exista doar atunci când mașina va înțelege limbajul curent. Cât de repede se va realiza acest lucru? Asupra acestui punct de vedere, cercetătorii sînt de aceeași părere - vor trece decenii. Și probabil că nu va exista niciodată o soluție multumitoare. Realizabile sînt însă sistemele de comunicare în limbaj natural pentru anumite domenii de aplicare, pentru care să se memoreze cunoștințele lingvistice și conceptuale. Un exemplu în acest sens este sistemul de comandă în limbaj natural "Dos-Man" oferit de firma Transmodul Software din Saarbrücken, începînd din 1989 și care este perfecționat în continuare. DOS-Man este un DOS-Shell, cu ajutorul căruia puteți opera la calculator, dînd comenzi sistemului de operare în limbaj natural. În locul comenzii "dir *.txt", în cazul lui

Dos-Man trebuie dată comanda: "afișează toate fișierele cu extensia txt" (în germană). În plus, pot fi date și comenzi mai complexe, cum ar fi, de pildă, "afișează toate fișierele care au fost create în această săptămînă", sau "caută programul Word".

În continuare vom descrie modul de funcționare al lui Dos-Man. Analiza unei propoziții se face în cinci pași: "scănare", "analiză morfologică", "analiză sintactică", "analiză semantică" și "analiză pragmatică".

Scanare

Propozițiile rostite sînt atribuite mai întii unor variabile text. Scanner-ul desface o propoziție - care în acest moment pentru Dos-Man nu este decât o succesiune de caractere - în cuvinte individuale. Depistarea cuvintelor din propoziție se face prin căutarea spațiilor dintre caractere. În plus, sistemul examinează și caracterele care nu sînt litere. Pentru descrierile ulterioare să considerăm comanda: "afișează telexurile, mai vechi de o lună, în ordinea dimensiunii".

Analiza morfologică

În cadrul analizei morfologice, cuvintele obținute în urma scanării sînt aduse în forma lor normală, de exemplu verbele sînt puse la infinitiv, iar substantivele la nominativ singular. Pentru aceasta sistemul accede dictionarul și, pe baza terminațiilor, stabilește clasa cuvîntului și obține și alte informații lingvistice. Astfel "afișează" devine "a afișa" și i se adaugă informațiile suplimentare "verb, imperativ", iar "telexurile" devine "telex" cu informațiile suplimentare "substantiv, plural".

Deoarece în limba germană fiecare cuvînt poate avea în medie 2,5 semnificații și poate fi încadrat în tot atîtea clase, analiza este mult mai complicată decât pare la prima vedere. În urma analizei morfologice sistemul nu poate stabili încă, dacă, de exemplu, cuvîntul "unu" este un articol sau un număr.

Analiza sintactică

În cadrul analizei sintactice sînt determinate părțile de vorbire ale propoziției, cu ajutorul unei gramatici care conține o mulțime de reguli ce definesc structura limbii. Pentru exemplul anterior, în urma analizei sintactice va rezulta: "predicat: afișează, subiect: telexurile, atribut participial: în ordinea dimensiunii, atribut relativ: mai vechi de o lună".

Analiza sintactică returnează propozițiile incorecte sintactic, respectiv care

nu respectă regulile gramaticale, cum ar fi, de exemplu, "mai vechi de afișează și telexurile o lună".

Analiza semantică

Scopul analizei semantice este determinarea sensului unei propoziții. Pentru aceasta sînt necesare cunoștințe lingvistice. Pe de altă parte sînt necesare cunoștințe conceptuale. De exemplu, trebuie știut că fișierele care conțin telexuri au extensia ".tlx" sau că instrucțiunea "mai vechi de o lună" înseamnă că diferența dintre data de creare a fișierelor respective și data curentă trebuie să fie mai mare de o lună. Baza de cunoștințe conceptuale trebuie să permită, în plus, recunoașterea unor comenzi lipsite de sens; cum ar fi "șterge calculatorul". În baza de cunoștințe sînt înregistrate ca obiecte care pot fi șterse fișierele și directoarele, nu și calculatorul. Pentru exemplul anterior, în urma analizei semantice, se va obține următorul rezultat:

- afișează, cum: în ordinea dimensiunii
- ce: fișierele cu extensia ".tlx"
- care - data de creare < data curentă - o lună

Analiza pragmatică

În cadrul analizei pragmatice, structurile semantice sînt puse în corespondență cu comenzile sistemului de operare:

1. determină directorul curent
2. determină fișierele cu extensia ".tlx" din acest director
3. determină care dintre aceste fișiere au data de creare conform cerinței
4. sortează aceste fișiere după dimensiune
5. afișează lista de fișiere sortată.

În acest mod comenzile devin inteligibile și pentru calculator și pot fi executate pas cu pas.

Fundamentele și problematica metodelor de eșantionare

Un sistem care poate analiza instrucțiuni verbale este denumit analizor (parser). Procedura de analiză se compune din trei pași: analiza sintactică, analiza semantică și analiza pragmatică. În timpul analizei există două probleme dominante. Una dintre ele constă în descrierea formulării de propoziții cu reguli precise - gramatica cu care poate fi creată orice propoziție imaginabilă într-o anumită limbă. Cealaltă problemă constă în analiza propriu-zisă. Propozițiile simple, constînd din verb - adjectiv - substantiv, pot fi prelucrate ușor. Lucrurile se complică în cazul frazelor complicate, cum ar fi: "caută în dulapul

mic, din camera de zi, toate scrisorile care au fost puse acolo în urmă cu două săptămîni". În acest caz cuvîntul "acolo", care se poate referi atît la "dulap" cît și la "cameră", provoacă dificultăți. Un analizor (parser) nu poate prelucra, de la început, o frază în întregime. El este obligat să examineze părțile de propoziție în parte, una după alta. Dacă s-a prelucrat doar o parte a frazei, atunci analizorul poate înțelege acea parte, dar îi este greu să-și dea seama ce rol joacă această bucată pe asamblu și cît de importantă este pentru înțelesul final.

Intrucît o propoziție aflată la sfîrșitul frazei poate avea o influență decisivă asupra semnificației ei, soluția de a analiza o frază cuvînt cu cuvînt nu este neapărat cea mai bună. În prezent există analizoare care lucrează după următoarele principii: "descindere recursivă", "acceptor final" și "Noise-Disposal".

Descindere recursivă

În cazul acestei metode, analizorul examinează mai întîi elementele din care este alcătuită fraza și divide aceste elemente în continuare, pînă cînd se ajunge la elemente care nu mai pot fi divizate în continuare. Mai simplu spus, un astfel de analizor trebuie să știe că din punct de vedere gramatical, o propoziție constă dintr-o parte nominală și dintr-una verbală. În afară de aceasta, el trebuie să dispună de o bază de date în care să fie stocate semnificațiile gramaticale ale cuvintelor, de exemplu "un = articol", "hoț = substantiv". Alte reguli gramaticale vor stabili, de exemplu, că un articol poate însoți numai un substantiv vsau un adjectiv, pentru ca propoziția să aibă sens.

Acest mecanism funcționează totuși numai în cazul propozițiilor simple. Regulile gramaticale ale unei limbi sînt în mod firesc prea complexe pentru a putea fi descrise inteligibil pentru un analizor. În plus, nu este suficient să i se spună analizorului căreia părți de vorbire îi aparține fiecare cuvînt.

Ca exemplu se dă de obicei propoziția "Brînză mîncă șoarecele". Această propoziție este înțeleasă foarte ușor de către om. Unui om nu i-ar trece niciodată prin cap că șoarecele ar putea fi cel mîncat. Un analizor ar putea înțelege această propoziție numai cu condiția ca în baza sa de cunoștințe să existe niște definiții de genul următor:

- brînză (rigid, aliment)
- șoarece (ființă)
- a mîncă (a consuma alimente)

Ce s-ar întîmpla însă dacă analizorul ar trebui să înțeleagă că un burghiu mușcă cîte puțin din peretele de beton?

Acceptor final

Un astfel de analizor examinează mai întîi primul cuvînt din frază și cu ajutorul lexiconului de cuvinte îi determină semnificația gramaticală (de exemplu "un=articol"). Principiul acceptorului final constă în faptul că o stare inițială se transformă, prin intermediul unei anumite reprezentări, într-o stare finală posibilă. Stările posibile sînt definite de părțile de vorbire ale cuvintelor din propoziție.

În exemplul anterior, prima stare este un articol. Elementele posibile care îl pot succeda sînt definite cu ajutorul unui graf. În acest graf se va stabili, de exemplu, că un articol nu poate fi urmat decît de un adjectiv sau de un substantiv. Analizorul parcurge deci fraza cu ajutorul acestui graf. Problema care apare în cazul acestui tip de analizor este că analiza eșuează imediat, dacă o propoziție nu este corectă gramatical sau dacă gramatica a fost insuficient descrisă în graf. Astfel apare aceeași problemă ca și în cazul descinderii recursive, și anume necesitatea definirii exacte a unei gramatici.

Noise-Disposal

Analizoarele Noise-Disposal nu țin cont de gramatică, ci doar de conținutul unei propoziții. Toate cuvintele din propoziție pe care nu le înțeleg sînt privite ca "zgomot" (noise) și sînt "îndepărtate" (to dispose) pur și simplu din propoziție. Din cuvintele rămase se încearcă obținerea unui sens. Este clar astfel că principiul de eliminare a zgomotului nu se pretează pentru o comunicare exactă, în limbaj natural, între om și mașină. Principiul se poate dovedi totuși bun pentru comunicări simple, în limbaj natural, desigur, cu condiția ca analizorul să dispună de un tezaur de cuvinte suficient de mare pentru domeniul de aplicare vizat.

(R.M.)

Bibliografie:

- Michael Nickles:
Mensch-Maschine-Kommunikation, DOS, iulie 1991, DMV- Verlag, ISSN 0933-1557
- Reiner J. König:
Der Computer hört aufs Wort, Computer persönlich, 24/91, Markt & Technik Verlag, ISSN 0722-0987

We draw on your imagination

A Bold New Shape.
A Fast New Pace.

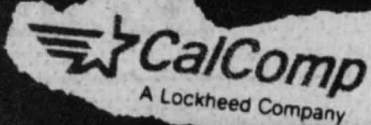
雅仕
A & C
INTERNATIONAL S.A.
AUTHORIZED DEALER

Constantin Tanase Str.
No. 15, Sect. 2,
Bucharest, 73299
Romania

Tel: 40-0-53.53.15
Service - "Hot Line": 40-0-12.77.73
Fax: 40-0-12.77.74
Telex: 10674 MAXIM R

...smooth. Our
...thanks to
...y and
...d

Take a bold first step to reshape plotting at your firm. Find out more today: Call 800-932-1212. In Canada, call 416-435-9010. We're expecting to hear from you.



Guru 3.0

Construirea de sisteme multimedia

Cea mai nouă versiune 3.0 a pachetului de dezvoltare sisteme expert Guru oferă proiectanților posibilitatea de a-și defini funcții proprii și de a memora și prelucra fișiere cu imagini, grafică sau secvențe sonore cu ajutorul așa-numitelor BLOB (Binary-large-objects).

Guru a fost creat în octombrie 1985 și este unul din primele instrumente de dezvoltare sisteme expert. Produsul a fost, și a rămas, cu un pas înaintea timpului său.

Sistemul cuprinde modulele: bază de date, calcul tabelar, grafică de afaceri și editor de texte. Prin intermediul unei interfețe de comunicație datele pot fi extrase de pe un calculator mai mare sau, de ex., pot fi interogate cursurile de la bursă. Toate modulele sînt complet integrate, astfel încît din modulul de bază de date se poate lansa o comandă pentru producerea unui grafic, sau din cadrul programului de calcul tabelar se poate lansa o interogare a sistemului expert.

Guru dispune de generatoare de măști și de rapoarte flexibile; limbajul de interogare al bazei de date este compatibil SQL.

Proiectarea de aplicații se realizează prin intermediul unui limbaj de programare puternic, din categoria 4GL. O interfață așa-numită în limbaj natural permite comunicarea cu Guru în limbajul uzual, desigur doar în engleză.

Mașina de inferențe

Guru memorează cunoștințele expert în forma clasicelor reguli "Dacă - Atunci".

În dezvoltarea de sisteme expert shell-urile bazate pe reguli pierd tot mai mult teren. Din acest motiv li se reproșează adesea proiectanților mediului Guru, faptul că mizează atît de mult pe sistemele bazate pe reguli. Totuși Guru dă posibilitatea formulării unor reguli foarte complexe, care să permită o bună prelucrare a cunoștințelor experților - după ce acestea au fost strict formalizate.

Mașina de inferențe a shell-ului Guru prelucrează regulile înlănțuit înainte și înapoi. Modul de căutare poate fi influențat; cunoștințelor vași li se pot atașa factori de probabilitate și ele pot fi prelucrate cu ajutorul logicii Fuzzy.

O nouă situație pe piață

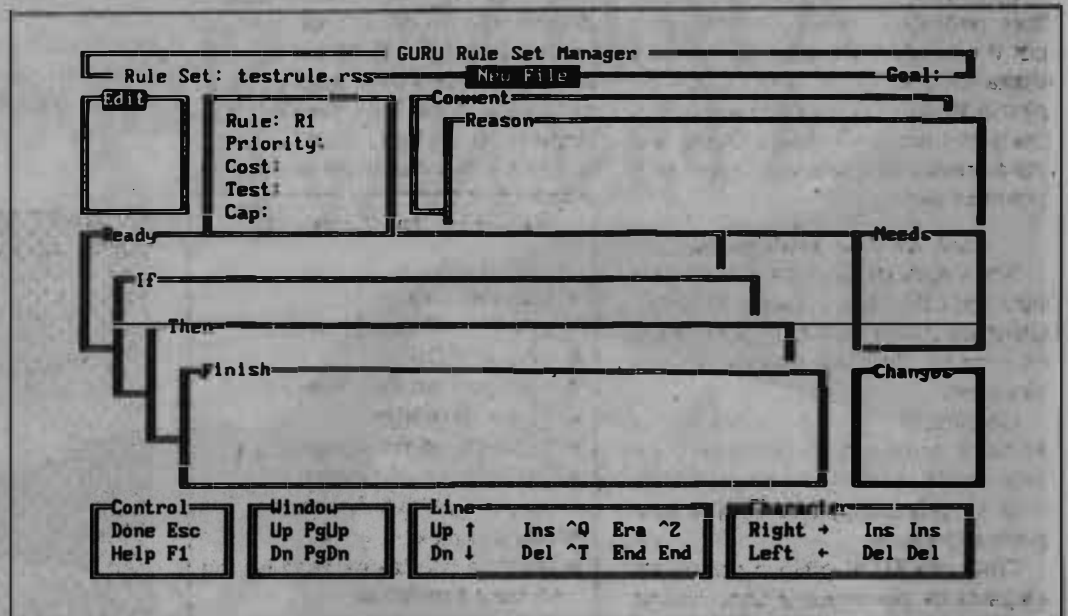
Guru este o extensie a lui "Knowledge Man" care este desemnat ca Decision-Support-Systems. Multă vreme

nici în cazul lui Knowledge Man nu a fost clar dacă era vorba despre un sistem destinat proiectanților sau utilizatorilor finali. Din această cauză Knowledge Man apărea uneori în teste comparative cu baze de date, alteori în teste comparative cu pachete integrate. Întotdeauna a fost lăudată puterea produsului, dar a fost criticat modul de operare complicat.

Asemănătoare a fost situația și în cazul lui Guru. Odată cu noua versiune 3.0, proiectantul - Micro Database Systems (MDBS) Inc. - definește explicit Guru ca instrument de dezvoltare. Acest lucru devine clar și prin prezentarea limbajului de programare de nivel 4GL, care acum se numește KGL (KnowledgeMan-Guru-Language). Datorită deplasării centrului de greutate spre programare, manualele au trebuit rescrise și restructurate. Documentația este prezentată sub forma unui manual de referință pentru KGL și a unei prezentări de ansamblu a modulelor. Prețioasă este o nouă cărțuie care se ocupă în mod special cu tehnica dezvoltării sistemelor expert.

Repoziționarea a implicat și pierderea suprafeței utilizator clasice. În locul ei a fost creată o nouă suprafață SAA

Suprafața utilizator Guru: masca de intrare pentru formularea de reguli pentru sistemul expert



(scrisă în KGL și disponibilă în cod sursă, pentru a servi de exemplu ca program bine scris în KGL).

Printre avantajele noii suprafețe se numără posibilitatea de a vedea oricând modul de ocupare al memoriei, sau de a interoga o bază de date prin metoda Query-by-example. Dezavantajul în constituie faptul că parcursul prin punctele de meniu nu mai este transformat, ca la versiunile anterioare, în comandă și nu mai este prezentat ca linie de comandă. Această abordare ușura învățarea limbajului de comandă.

În timpul testelor, modul de lucru cu meniuri a provocat de două ori pierderi de date. La ieșirea din modulul calcul tabelar nu s-a pus întrebară dacă datele trebuie salvate și în ce fișier; memorarea lor n-a decurs corect și un nou apel al tabelii tocmai prelucrate prin meniu n-a mai fost posibil. În modul comandă, în schimb Guru a reacționat corect întotdeauna.

Funcții definite de utilizator

Cu ajutorul comenzii "perform procedure" sînt apelate și prelucrate programe Guru stocate în fișiere. Pînă acum transferul valorilor (parametrilor) spre programe și între programe se făcea cu dificultate. Un tip de funcție așa-denumit "User-defined-functions" (UDF) permite așa- numitul "parameter-passing", așa cum este uzual în alte limbaje de programare.

UDF-urile sînt subprograme care, spre deosebire de programele Guru, pot fi compilate. În acest mod crește viteza de execuție, în unele cazuri de pînă la 10 ori. UDF-urile primesc un nume și sînt stocate în fișiere BLOB. Ele pot fi apelate ca o funcție în cadrul programelor Guru.

Bază de date multimedia

Noua versiune Guru permite tipul de date "BOLB" (Binary Large Objects). Obiectele binare mari sînt fișiere binare de orice tip, deci text, imagine, grafică sau sunet.

Un cîmp BLOB este o parte a unui articol. În acest mod în datele de bază de personal, de exemplu, se poate defini un cîmp BLOB care să conțină fotografia angajatului.

Cîmpurile BLOB oferă o posibilitate elegantă de prelucrare a unor volume mari de date nestructurate. Cu Guru pot fi create, de exemplu, arhive de imagini

Sisteme expert create cu Guru

- ❑ **Volkswagen AG, Wolfsburg:** Prognoză asupra disponibilității personalului
- ❑ **Pflanzenschutzamt Hannover:** Consultanță privind utilizarea substanțelor chimice pentru protecția plantelor
- ❑ **Instituto Químico de Sarria, Barcelona:** Sistem expert privind prelucrarea petrolului
- ❑ **Nasa Ames Research Center:** Support utilizator Cray pentru o mal bună coordonare a rețelei de supercalculatoare.
- ❑ **Army Secretary for Joint Actions:** Prelucrarea și regăsirea documentelor.
- ❑ **Unlon Bank Finland:** Sistem expert de consultanță în probleme de finanțare.

sau arhive muzicale. Dacă se mai adaugă și inteligența tehnologiei sistemului expert, Guru ar putea crea, de exemplu, o arhivă cu bucățile muzicale transmise de un anumit post de radio pentru un anumit profil al ascultătorului și/sau între anumite ore din zi.

Singura dificultate: este necesar un software corespunzător pentru reprezentarea graficii, textelor, etc., memorate într-un cîmp BLOB. De exemplu, cine prelucreează cu Guru fișiere cu grafică Harvard Graphics, trebuie să apeleze HG de sub Guru.

Harvard Graphics va afișa în acest caz imaginea dorită. Fișierele PIC pot fi afișate de Guru și fără ajutorul altui program.

Prelucrarea datelor cu cîmpuri BLOB mai pune o problemă: volumele mari de date.

Găsirea unei bucăți muzicale sau a unei imagini pe suportul de date se întâmplă în zecimi de secundă, numai că fișierele cu imagini sau sunete necomprimare au nevoie de mult spațiu pentru memorare, astfel încît harddisk-ul se umple rapid. În plus un fișier BLOB este cu cîțiva octeți mai mare decît fișierul

original, deoarece sînt adăugate informații de identificare. CD-ROM este suportul de memorare ideal pentru bazele de date multimedia create de Guru.

Mai rapid, mai puțin spațiu de memorie

Codul sursă al programului Guru a fost optimizat și comprimat în permanență de la versiunea 1.1 încoace. Deși se recomandă o memorie de 640 KByte, versiunea 3.0 - sub "Deskview" de exemplu - se mulțumește și cu 360 KByte.

Un sistem de consultanță creat cu versiunea 1.1 a fost rulat în versiunea 3.0 cam de două ori mai rapid. Guru este deplin funcțional, deci, chiar și pe un simplu XT. Pe harddisk ocupă cca. 5 MByte.

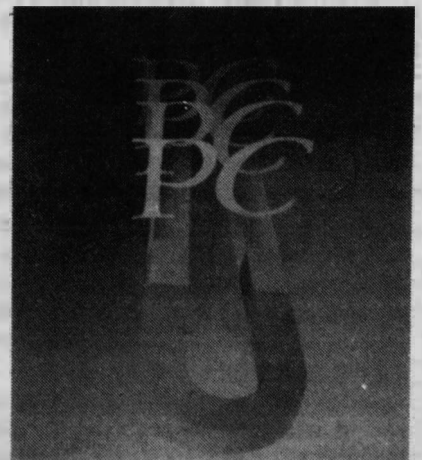
Mediul de dezvoltare este disponibil pentru OS/2, VAX, Sun și Unix.

Guru 3.0 costă 12.000 DM pentru un utilizator, 30.000 DM, versiunea LAN (număr nelimitat de utilizatori per Server). Există și un sistem de dezvoltare pentru începători la prețul de 3.195 DM. Modulul Runtime costă 1.600 DM.

(R.M.)

Noutăți la Guru 3.0

- suprafața SAA
- UDF (User Defined Functions)
- cîmpuri BLOB
- interogare bază de date QBE
- Report Templates
- limbaj de definire rapoarte
- SQL - Server - Frontend
- necesită mai puțină memorie
- este mai rapid
- permite analizarea modului de ocupare a memoriei



Blob, blob...

Un nou tip de câmp pentru bazele de date va schimba scena din temeli: "Binary Large Objects", denumite și BLOB, oferă utilizatorului de baze de date o mulțime de noi posibilități.

Lumea informaticii ar fi de neimaginat, astăzi, în absența bazelor de date relaționale. Multe sisteme utilizează limbaje de programare proprii, înglobate în sistem cum sînt de pildă limbajele de nivel 4GL, sau limbaje de programare tradiționale, cum ar fi C, Cobol și Fortran. O caracteristică distinctivă a bazelor de date o constituie organizarea datelor sub formă de tabele: datele sînt prelucrate pornind de la o structurare în rînduri (articole) și coloane (cîmpuri). De regulă se utilizează cîmpuri și articole de lungime fixă.

Ca tipuri de date se utilizează cîmpurile de tip șir, tip numeric, respectiv logic ca și tipurile derivate cum ar fi: dată, timp, cîmp întreg. Unele sisteme utilizează și cîmpuri speciale, în care se pot memora texte sau grafice de lungime variabilă. Doar în puține sisteme este prezent tipul de date, relativ nou, "Binary Large Objects", mai scurt denumit și BLOB.

BLOB-urile reprezintă mai mult decît cîmpurile tip Memo normale

BLOB-urile au o lungime fixă. Spre deosebire de cîmpurile de tip Memo sau de cîmpurile conținînd grafică, cîmpurile BLOB pot conține nu doar texte ASCII sau doar grafice. Ele permit stocarea, în contextul unui articol, a unui volum de date de orice gen. Astfel pot fi stocate imagini grafice, sunet (vorbire, zgomot, muzică), și desigur și texte.

Într-un cîmp de date care urmează a fi stocate, nu trebuie stabilit cu exactitate, BLOB-urile sînt universal utilizabile. Fizic, în mod normal, ele nu sînt stocate în aceeași structură de date cu celelalte cîmpuri ale articolelor bazei de date, ci în fișiere distincte. Schița pe care o prezentăm va lămurii această problemă. În articolul propriu-zis se stochează doar un pointer, care va indica poziția exactă în cadrul fișierului BLOB.

Din motive de performanță, pentru fiecare intrare într-un astfel de fișier se rezervă cîte un bloc de lungime fixă (de

ex. 512 Byte). Atunci cînd se actualizează conținutul unei anumite intrări BLOB și dimensiunea crește astfel încît blocul rezervat nu mai este suficient pentru stocare, fișierul BLOB trebuie reorganizat intern. Această reorganizare are loc în detrimentul vitezei de prelucrare.

Dacă dimensiunea standard a blocurilor utilizate poate fi liber stabilită, dimensiunea lor este determinată principial de necesarul de memorie mediu, sau maxim, pentru ceea ce trebuie memorat. Principal, totuși, conținutul unui BLOB nu este limitat în dimensiune și poate atinge domenii de ordinul gigaoctetelor.

Multimedia și BLOB

Progresul tehnic a făcut ca expresii ca "multimedia" sau "prelucrarea imaginilor" să fie pe buzele tuturor. În multe aplicații sînt integrate și imagini, prin referințe la numele corespunzătoare de fișiere. Un sistem care conține puține imagini sau grafice le poate trata în mod onorabil. Dacă biblioteca grafică crește în volum, atunci crește enorm și efortul depus pentru prelucrarea, organizarea și regăsirea lor.

Pentru prelucrarea și regăsirea de imagini și grafice cu ajutorul unei baze de date, tipul de date BLOB oferă premise excelente. Cu ajutorul lui nu este nici o problemă să se formuleze o întrebare în limbaj SQL (o comandă SELECT, de ex.) asupra unei biblioteci grafice. Condițiile impuse se pot referi, de exemplu, la texte stocate în alte cîmpuri ale articolului. Dar a rezuma posibilitățile de utilizare a cîmpurilor tip BLOB doar la aplicațiile multimedia nu ar fi corect.

Guru, Informix și KnowledgeMan

Printre aplicațiile comerciale care integrează deja cîmpurile de tip BLOB se numără, printre altele, și produsele elaborate de specialistul în baze de date MDBS: "Guru" și "KnowledgeMan" ca și "Informix - OnLine" al firmei Informix Software. La ambele produse MDBS pot fi integrate oricîte cîmpuri de tip BLOB în cadrul unei tabele. Limbajul de programare integrat "KGL" dispune de o mulțime de funcții care permit memora-

rea, respectiv citirea, de date din cîmpurile BLOB. Aceste date pot fi orice tip de fișiere externe (texte sau binare, incluzînd fișiere executabile tip EXE sau COM), sau și șiruri interne, matrici uni sau bidimensionale; formate grafice interne, tipuri de date definite de utilizator. În plus variabilelor li se pot atașa mai multe valori și anumiți coeficienți de probabilitate (variabile fuzzy). În SGDB-urile tradiționale aceste tipuri de date nu pot fi utilizate decît cu un efort considerabil. Cu ajutorul BLOB-urilor problema devine trivială. Ca exemplu ar putea servi următoarea comandă Guru/KnowledgeMan:

```
SELECT DISPLAY (PERSONAL.IMG)
FROM PERSONAL
WHERE SEX = 2 AND AGE <= 30
```

Această comandă SQL va determina afișarea pe monitor a fotografiilor tuturor angajaților înregistrate în tabela PERSONAL, care au vîrsta mai mică sau egală cu 30 de ani.

Există și posibilitatea stocării unor secvențe de program, anumite rutine, funcții de bibliotecă și a altor obiecte în cîmpurile BLOB, care să fie utilizate apoi în proiectarea de software sau la execuția unei aplicații.

Un efect colateral suplimentar îl constituie extinderea mecanismelor de protecție (drepturi de scriere/citire) specifice bazelor de date, și asupra acestor tipuri de cîmpuri. În acest mod și aceste date pot fi protejate împotriva unor acedări neautorizate.

Dacă SGBD-ul utilizat dispune de interfețele și de funcțiile necesare atunci nu mai există nici o predică în calea materializării ideilor utilizatorului.

Scena bazelor de date se modifică

Industria SGBD-urilor începe abia acum să comercializeze tehnologia BLOB. Ingres, în postura sa de lider, și mulți alții, și-au făcut publică intenția de a integra cîmpurile de tip BLOB în arhitecturile proprii. BLOB-urile vor influența în mod decisiv modul de proiectare al pachetelor software bazate pe baze de date și al sistemelor expert. Toți proiectanții de SGBD-uri relaționale vor trebui, mai curînd sau mai tîrziu, să includă în produsele proprii tipul de date BLOB.

(continuare în pag. 24)

Baze de date orientate obiect

După limbajele de programare orientate obiect vin la modă bazele de date orientate obiect. Ce sînt aceste baze de date și care sînt performanțele lor va fi exemplificat printr-un proiect care funcționează în cadrul Societății pentru Matematică și Prelucrarea datelor (GMO) din Germania, în momentul de față.

Sistemele de gestiune a bazelor de date (SGBD) joacă un rol foarte important în prelucrarea și administrarea unor cantități mari de date. Importantă nu este doar eficiența serviciilor caracteristice ale SGBD-urilor cum ar fi managementul tranzacțiilor, protecția la acces, siguranța datelor și integritatea lor ci și pretarea modelului de date ales la modelarea datelor. Să presupunem că avem o mulțime mare de date structurate în mod complex, de exemplu structuri de documente și formulare, de layout-uri sau/ și structuri geometrice complexe pe care trebuie să le memorăm într-o bază de date relațională. În acest caz structurile complexe vor fi modelate ca o mulțime de structuri plane sub forma unor relații. Deci unitățile legate între ele printr-o anumită logică vor fi descompuse în unități mai mici. Astfel diferitele programe utilizator vor trebui să compună în mod greoi, prin pași succesivi, de interconectare, (operații joint), din fiecare relație plană obiectele complexe sau să implementeze operațiile logice pe obiectele complexe ca și operații pe relații plane. Acest lucru nu este critic doar din punct de vedere al eficienței diferitelor programe utilizator, ci și în ceea ce privește interpretările și manipularile consistente asupra structurilor plane, care reprezintă de fapt obiecte complexe. SGBD-ui poate garanta doar consistența pe structurile plane cunoscute lui și în mai mică măsură consistența obiectelor structurate complex.

Un SGBD orientat obiect, cum ar fi cel din proiectul VODAK de la Institutul pentru Sisteme Integrate Informaționale și Publicistice, oferă concepte și mecanisme, care evită asemenea dezavantaje și limitări și care simplifică administrarea și prelucrarea unei cantități mari de date structurate. Acest lu-

cru se obține prin integrarea conceptelor orientate obiect cu tehnologiile convenționale ale bazelor de date.

Termenul de "orientare obiect" este folosit într-o varietate atât de mare de cazuri, încît chiar și experților le vine greu să dea un echivalent exact al acestei denumiri într-un context bine determinat. Termenul apare în diferite domenii și anume: limbaje de programare, baze de date, baze de cunoștințe, inteligență artificială și software.

La întrebarea: ce este de fapt un SGBD orientat obiect și care sînt avantajele sale față de sistemele convenționale nu poate fi dat nici un răspuns atîta timp cît nu se cunosc cel mai importanți termeni care se folosesc aici: obiectul, clasa, ereditatea, metodele și mesajele, fără a se cunoaște modelul care a stat la baza sa.

Modelul orientat obiect

Orientarea obiect poate fi considerată ca o tehnică de organizare a unui sistem în vederea tratării obiectelor care apar în el. Conceptul de bază este principiul de încapsulare: datele care descriu o unitate logică împreună cu caracteristicile lor și cu operațiile definite asupra acestor date vor fi încapsulate într-un obiect. Datele pot fi citite și modificate, asemănător cazurilor de tipuri de date abstracte, doar prin folosirea operațiilor definite în acest scop.

Astfel, modificări ale reprezentării structurale a datelor unui obiect nîu are efect direct asupra celorlalte obiecte.

Obiectele comunică schimbînd mesaje între ele. Un mesaj transmis unui obiect are ca efect executarea unei anumite operații asupra acestui obiect. Acest mecanism mai este denumit și Message Passing. În cadrul limbajelor de programare convenționale programatorul este răspunzător de alegerea corectă a operatorului. El identifică de pildă codul de program pentru adunarea a două numere complexe $A=6+3i$ și $B=3-2i$ cu numele funcției "Complex Add", atunci cînd scrie: `ComplexAdd(A,B)`. Dacă el dorește să adune cifrele binare $C=101101$ și $D=10011$, el trebuie să folosească funcții corespunzătoare, și anume `BinaryAdd(C,D)`. Deci utilizatorul funcțiilor `ComplexAdd`

și `BinaryAdd` identifică odată cu numele funcției și o parte a codului program, care va trebui executat, deci el decide cum va fi executată o operație. Modelul programării orientate obiect transferă responsabilitatea de la utilizatorul unei funcții la proiectantul acelei funcții.

Într-un sistem orientat obiect, programatorul poate scrie exemplele de mai sus în forma unor mesaje `A.Add(B)` și `C.Add(D)`, dacă operația `Add` a fost definită în prealabil atîta pentru numere complexe, cît și pentru cele binare. El va specifica doar ce trebuie executat. Care algoritim se folosește la adunare, se va determina automat, în funcție de obiectele `A`, `B`, `C` și `D` în cauză, deoarece la încapsulare a fost stabilit cu exactitate care dintre operații se pot efectua cu datele respective. Deci în cazul `A.Add(B)` va fi executat automat algoritmul de adunare a numerelor complexe, iar în cazul `C.Add(D)` cel al adunării numerelor binare. Acest mecanism poartă denumirea de polimorfism.

Obiectele de același fel se pot clasifica în clase de obiecte. Clasele folosesc ordonării ontologice ale obiectelor care apar și se prelucrează într-un sistem. Obiectele unei clase se numesc instanțe ale acelei clase. Astfel de exemplu numerele complexe `A` și `B` de mai sus sînt instanțe ale clasei `Complex Number`, iar `C` și `D`, instanțe ale clasei `Binary Number`.

Prin clasa unui obiect se determină atîta reprezentarea internă a datelor, deci structura unui obiect, cît și mulțimea operațiilor definite asupra obiectului.

Pentru a obține părți modulare de sisteme, definițiile de clase ar trebui să poată fi reutilizabile și la alte definiții de clase. Acest lucru se poate obține prin conceptul eredității. Definițiile unei clase pot fi derivate din alte definiții de clase, deci o clasă nouă se construiește pe baza unei clase existente, adăugînd acesteia noi definiții de metode și structuri interne de date. Procesul acesta poartă numele de specializare, deoarece clasele noi detaliază instanțele lor și le descriu mai specific, decît clasele mai generale folosite la compunerea lor.

Aceste principii sînt comune tuturor sistemelor orientate obiect, cu toate că

detaliile sînt realizate și definite în mod diferit. Orientarea obiect ușurează de multe ori, pe baza modelării naturale, înțelegerea sistemelor, a analizei lor, a proiectării, realizării și a întreținerii lor.

Sisteme de gestiune a bazelor de date orientate obiect

SGBD-urile orientate obiect se deosebesc prin integrarea funcționalității bazelor de date convenționale cu concepte de modelare orientate obiect. Scopul lor primar este de a separa reprezentarea și administrarea datelor de programele utilizator, care prelucrează aceste date, și de a permite o utilizare consistentă a datelor de către mai multe programe utilizator, astfel încît programele să devină independente de reprezentarea internă a datelor. Totuși responsabilitatea interpretării consistente și a manipulării datelor rămîne în sarcina programelor utilizator.

Gradul înalt de complexitate al cîmpurilor aplicației, cum ar fi, de ex., prelucrarea de documente puternic structurate, necesită transferul unei părți a acestei responsabilități de la programele utilizator către datele structurate și anume în SGBD. Cu alte cuvinte, programele utilizator nu trebuie doar să devină independente de reprezentarea internă a datelor, ci și de semantica datelor, și anume, de operațiile ce se execută asupra acestor date. Doar astfel se poate garanta independența de aplicațiile specifice a interpretării și manipulării corecte a informațiilor structurate complex, prin intermediul SGBD-urilor.

Adăugirea avută în vedere în cadrul proiectului VODAK pentru un SGBD deschis, orientat obiect, elimină următoarele îngrădiri ale sistemelor convenționale de baze de date:

- reprezentarea structurată: obiectele structurate complex pot fi depuse integral în baza de date. Ele nu trebuie descompuse în părți unitare plane, pentru a fi recompuse mai apoi de către programul utilizator. Acest lucru este posibil pe de o parte datorită construcțiilor tip predefinite cum ar fi Tuple, Set și altele, iar pe de altă parte prin posibilitatea de a putea defini clase de obiecte oricît de complexe.
- prelucrarea datelor: programele utilizator sînt separate atît de reprezentarea internă a datelor cît și, în mare parte, de semantica datelor.

Astfel, programele utilizator devin invariabile nu numai față de schimbarea formatului datelor, ci obțin în mare măsură și o interpretare și o manipulare consistentă a datelor conform semanticii lor, toate acestea fiind garantate de SGBD.

- model deschis, extensibil.

În comparație cu o serie dată de primitive de modelare, care stau la baza definirii unei "scheme" a unei baze de date într-un SGBD convențional mulțimea primitivelor de modelare poate fi lărgită cu primitive de modelare specializate pe o anumită aplicație.

Astfel modelul de date al SGBD-urilor orientate obiect poate fi ajustat în funcție de cerințele unei aplicații.

În cadrul VODAK în colaborare cu alte proiecte din Institutul pentru Sisteme Integrate Informaționale și Publicistice, se execută asemenea ajustări și extinderi de model în scopul modelării de documente hypertext, multimedia și pentru integrarea unor baze de date eterogene deja existente.

În cadrul integrării unor concepte de modelare orientate obiect și a funcționalității SGBD-urilor convenționale este necesară și o anumită adaptare a unor funcții fundamentale ale SGBD-ului.

Doar astfel se poate ajunge la o concordanță optimă a serviciilor caracteristice ale unei baze de date, cum ar fi de exemplu managementul de tranzacții, cu conceptele unui model orientat obiect.

Astfel în cadrul VODAK se exploatează de exemplu comutativitatea operațiilor asupra obiectelor, pentru a obține un paralelism mai ridicat în sistem. Acest fapt este important, deoarece structurile liber definite ale obiectelor unei aplicații pot deveni foarte mari și din acest motiv pot exista multe accedări paralele asupra aceluiași obiect.

De aceea, reducerea la minim a unor asemenea accedări este o premisă importantă în obținerea unor performanțe ridicate la un SGBD orientat obiect.

Deoarece o operație asupra unui obiect poate duce la apelarea altor operații asupra aceluiași obiect și a altor obiecte, tranzacția VODAK constă din mai multe subtranzacții.

Această "îmbricare" a tranzacțiilor are avantajul că fiecare operație se derulează ca o tranzacție. Deci operația va fi executată complet sau sistemul va fi

pus în mod automat în starea consistentă dinaintea executării operației. Dacă există subtranzacții care pot fi derulate cu succes, atunci rezultatele acestora devin instantaneu vizibile și utilizabile de către alte tranzacții, și nu doar atunci cînd întregul proces de tranzacții a fost încheiat.

Dacă o tranzacție globală trebuie întreruptă dintr-un motiv anume, atunci subtranzacțiile care deja au fost derulate cu succes vor fi repuse în starea lor inițială, cu ajutorul așa-numitelor tranzacții de compensare.

Deci datorită puterii mari de modelare și a integrării conceptelor orientate obiect cu funcționalitatea unei baze de date, SGBD-urile orientate obiect au în anumite aplicații avantaje mari față de sistemele convenționale.

Prototipuri ale unor astfel de sisteme, pe lîngă cel din Institutul pentru Sisteme Integrate Informaționale și Publicistice, se experimentează în toată lumea, în laboratoare de cercetare pentru a dezvolta în continuare tehnologia bazelor de date orientate obiect pînă în faza în care vor putea fi prezentate pe piață.

Componente ale unor astfel de sisteme sînt implementate deja în unele produse fără ca acestea să dispună însă de toate caracteristicile de performanță amintite.

(I.M.)

Blob, blob ...

(continuare din pag. 22)

Firma Data Access din Friedrichsdorf oferă deja o soluție. SGBD-ul "Dataflex", în versiunea 3.0 permite cîmpuri de tip BLOB cu o dimensiune maximă per cîmp de 16.384 octeți. Dacă un fișier binar depășește această dimensiune, atunci acesta poate fi divizat în mai multe cîmpuri. Teoretic, în acest mod, în fiecare tabelă a bazei de date, ar putea fi stocat un fișier binar avînd dimensiunea maximă de 16 milioane ori 16.000 octeți. O mărire a lungimii unui cîmp BLOB nu se are însă în vedere la Data Access.

Pentru tratarea unor cîmpuri de date mari, cum ar fi de pildă imaginile scanate, se oferă un program utilitar. Acesta preia afișarea imaginilor într-o fereastră, ca și memorarea și citirea datelor din fișier. Dataflex 3.0 rămîne răspunzător, în schimb, de manipularea datelor.

(R.M.)

Baze de date Client-Server

Care este diferența între un Sistem de Gestiune a Bazelor de Date (S.G.B.D) integrat și un model Client-Server? Ce avantaje prezintă stocarea centralizată a datelor și prelucrarea descentralizată a lor și ce implicații au în acest caz tehnologiile moderne de realizare a rețelelor? În articolul care urmează vom încerca să răspundem acestor întrebări.

Pentru a putea decide care S.G.B.D. este mai indicat pentru un anumit domeniu aplicativ, vor trebui lămurite mai întâi care sînt cerințele de calitate care se pretind sistemului: cît de mare va deveni baza de date? în ce scopuri vor fi manipulate datele și cărui cerc de utilizatori vor trebui să fie accesibile? Noțiuni ca redundanță, integritate, consistență sau actualitate vor fi alte criterii hotărîtoare în cazul unei evaluări.

S.G.B.D.-uri relaționale

S.G.B.D.-urile care pun în relație mai multe fișiere sînt denumite S.G.B.D.-uri relaționale. Într-un S.G.B.D. relațional fișierele sînt denumite tabele. Tabelele sînt înlănțuite prin intermediul unui cîmp comun, de exemplu numele. Să dăm un exemplu pentru clarificare: secția de reclamă a unui supermarket are nevoie de adresele clienților pentru a le putea expedia broșuri de reclamă. Secția de creditare în schimb, are nevoie în primul rînd de numele clienților și de datele referitoare la cont pentru a putea face înregistrările contabile. Dacă un client nu este capabil să returneze un credit acordat, atunci secția de creditare va trebui să scrie o somație de plată și s-o expedieze clientului. Pentru aceasta are nevoie de adresă. Prin intermediul cîmpului comun "nume" se poate face o legătură între tabela "adresă" și "conturi". Într-un SGBD nerelațional anumite date trebuie stocate în ambele fișiere, pentru a fi accesibile ambelor secții. Lipsa relației are ca dezavantaj faptul că datele trebuie stocate de mai multe ori. Acest fapt ridică anumite probleme pe care le vom descrie în continuare.

Date redundante

Se vorbește despre redundanță atunci cînd o bază de date conține date

echivalente stocate de mai multe ori. Dacă, de exemplu, adresa domnului Moraru este memorată atît la secția de desfacere cît și la contabilitate și la secția de creditare, aceleași date sînt stocate de trei ori. Această metodă implică ocuparea triplă a spațiului de memorare, și nu este economică. Două din cele trei adrese stocate sînt în plus, deci sînt redundante. În cazul stocării centralizate a datelor se reduce redundanța deoarece datele sînt memorate într-un singur loc, și cu toate acestea sînt accesibile tuturor secțiilor.

Integritatea datelor

Dacă adresa domnului Moraru trebuie modificată, din cauza unei modificări de domiciliu, de ex., în cazul unei stocări descentralizate această adresă trebuie modificată la toate secțiile (de trei ori în cazul nostru). Integritatea datelor - se vorbește adesea și despre consistența datelor - este periclitată dacă nu toate adresele sînt modificate cît mai rapid (dacă se poate simultan).

Actualitate

Pe actualitatea datelor se pune un preț foarte mare. Căci la ce-i servește secției contabilitate faptul că secția desfacere are fișierul de adrese ia zi dacă somațiile de plată proprii sînt returnate cu mențiunea "destinatar necunoscut"? De aici se poate vedea cît de strîns legate între ele sînt noțiunile redundanță, integritate și actualitate

Modelul Client - Server

Modelul Client - Server oferă o alternativă la modelul SGBD integrat. Într-un SGBD integrat baza de date propriu-zisă și suprafața utilizator alcătuiesc un tot unitar. În baza de date - așa-numitul Backend - sînt stocate fizic datele, în timp ce prin intermediul suprafeței utilizator - așa-numitul Frontend - datele sînt definite, manipulate și controlate. Această unitate concepțională implică înșă și o unitate fizică, aceasta înseamnă că baza de date este instalată pe un singur PC. Ca urmare datele vor fi accesibile doar utilizatorului aceluși PC.

O rețea, din contră, constă dintr-un calculator de comandă puternic - server-ul - și mai multe stații de lucru, așa-

numitele Workstations. Într-un SGBD care lucrează în rețea aceste două unități logice și fizice sînt denumite Server și Client. Astfel server-ul poate fi un PC puternic, de ex. un 386 sau 486, dar poate fi și un minicalculator sau microcalculator. Clienții sînt legați de server prin intermediul cartelelor de rețea și al cablului și nu trebuie să aibă aceeași configurație cu server-ul. În mod obișnuit sînt utilizate PC-uri 286 sau 386. Pe aceste PC-uri pot fi utilizate diferite pachete software, care însă să ofere o interfață unitară pentru comunicarea cu server-ul. Această interfață este realizată de un limbaj de interogare standardizat, așa-numitul Structured Query Language, cunoscut sub prescurtarea SQL. Posibilitatea de a utiliza pachete soft diferite permite o prelucrare și/sau o reprezentare corespunzătoare. Astfel un program de gestiune a stocurilor poate ușura examinarea rezervei de articole din stoc. Un procesor de texte cu funcție de scriere "scrisori-în-serie" va avea nevoie de adresele din baza de date și va constitui astfel un frontend adecvat pentru secretariat. În fine un program de calcul tabelar cu funcții grafice integrate va permite o reprezentare sub formă de diagrame a datelor.

Modelul client-server versus mainframe

Avantajele utilizării de PC-uri - și deci a modelului client-server - comparativ cu modelul calculatoarelor mari (mainframe) sînt imediate. Calculatoarele mari împreună cu terminalele neinteligente alcătuiesc un sistem. Terminalele neinteligente sînt aparate de monitorizare, care nu dispun de inteligență proprie și care reprezintă datele așa cum le primesc de la calculatorul mare. O reprezentare pretențioasă va implica solicitare suplimentară a calculatorului care în același timp trebuie să se ocupe și de gestiunea datelor. Situația se prezintă cu totul altfel în cazul utilizării rețelelor de PC-uri, în care sînt utilizate terminale inteligente, respectiv PC-uri, pentru a prelucra în continuare datele obținute în urma interogării. Pe lîngă posibilitatea utilizării unor pachete soft diferite și hardware-ul (capacitate hard-

disk, facilități grafice, putere procesor, etc.) poate fi adaptat necesităților concrete.

Rețele eterogene

Din fericire în cazul rețelelor de calculatoare moderne nu se mai impune trăsarea unei linii de demarcație între familiile de calculatoare. Cuvântul magic este rețele eterogene. Spre deosebire de o rețea omogenă, în care se pot cupla doar calculatoare de același tip (de ex. numai PC-uri compatibile IBM), într-o rețea eterogenă pot fi cuplate calculatoare din familii și cu platforme diferite. De exemplu datele pot fi stocate pe un calculator mare și suplimentar pot fi utilizate și mediile de memorare ale mini- sau microcalculatoarelor. Operarea simplă a PC-urilor permite o manipulare a lor (gestiune și reprezentare) pe stația de lucru. Pentru aceasta, în funcție de aplicație, pot fi utilizate atât calculatoare compatibile IBM (de ex. pentru calcul tabelar și prelucrare de texte), cât și Macintosh (de ex. pentru grafică) sau alte tipuri de calculatoare. În acest mod se obține un efect de sinergie, prin utilizarea caracteristicilor complementare ale calculatoarelor individuale.

Prelucrarea tranzacțiilor

Deoarece în cazul unui SGBD client-server se gestionează un sistem complex de date distribuite, trebuie să se acorde o atenție deosebită securității datelor. În centrul atenției stă în acest caz noțiunea de tranzacție. O tranzacție constă dintr-o succesiune de comenzi elementare care trebuie executate ca o singură operație. O cădere de sistem nu trebuie să ducă la întreruperea unei tranzacții în timpul prelucrării ei. Dar cum se

poate asigura acest lucru? Vom încerca să lămurim această situație prin exemplul următor:

Fie o tranzacție care să fie alcătuită din comenzile: citește articole din tabela A, copiază aceste articole în tabela B și șterge vechile articole din tabela A. Această procedură descrie deci o simplă mutare a articolelor din tabela A în tabela B. Dacă, de ex., comanda din mijloc nu este executată (de ex. din cauza unei erori de sintaxă), datele din tabela A nu vor mai fi copiate în tabela B și cu toate acestea ele vor fi șterse din A. În cazul unei tranzacții, în schimb procedura începe cu BEGIN WORK. Dacă nici una dintre instrucțiunile conținute în tranzacție nu produce mesaje de eroare și nu există nici un motiv care să necesite o reluare a operațiilor, atunci tranzacția va fi încheiată cu COMMIT WORK. Abia în acest moment modificările sînt scrise în baza de date cu caracter permanent. Dacă în schimb în timpul tranzacției a apărut o eroare, cu comanda ROLLBACK WORK se poate reface vechea stare, fără pierderi de date.

Restaurarea datelor

Prin metoda de prelucrare a tranzacțiilor și comanda ROLLBACK se asigură deci o consistență a datelor. Uneori se poate întâmpla, totuși, ca o cădere de sistem să strice consistența datelor. Pentru a putea reface starea de dinaintea căderii (pentru aceasta se utilizează termenul englezesc Recovery) SGBD-ul ar trebui să realizeze două lucruri: să înregistreze un protocol al tranzacțiilor (Journal) și să posede funcția ROLLFORWARD.

Pentru aceasta este citit mai întâi ultimul backup și este declarat apoi ca baza de date curentă. Cu ajutorul instrucțiunii ROLLFORWARD Bază_de_date toate tranzacțiile care fuseseră executate se execută din nou pentru noua bază de date (restaurată). În acest mod se reface din nou vechea stare a bazei de date.

Pe lângă protocolul tranzacțiilor, care se referă la operațiile care afectează întreaga bază de date, mai există și așa-numitul protocol Audit. Acesta din urmă se referă la modificările survenite în cadrul unei tabele. Pentru fiecare tabelă poate fi executat deci un protocol audit și pot fi efectuate astfel reconstrucții ale unei tabele anume. Deoarece în cazul unui SGBD relațional în cazul unei căderi de sistem, sau al unor instrucțiuni eronate, sînt afectate, de regulă, mai multe tabele înlănțuite, un protocol al tranzacțiilor este mai indicat și mai confortabil. Din motive de complexitate un protocol audit este executat de regulă la un SGBD integrat, în timp ce la SGBD-urile client-server se utilizează, de regulă, protocolul tranzacțiilor.

Mecanisme de blocare

Prin Locking se înțelege un mecanism automat de blocare care devine activ în cazul accesării simultane a mai multor utilizatori asupra aceleiași colecții de date. În cazul unui mediu multiuser (multiutilizator) se poate întâmpla ca mai mulți utilizatori să accedă simultan la aceeași tabelă. Operațiile pure de citire nu pun probleme. Dacă în schimb utilizatorul Moraru modifică date într-o tabelă (șterge, editează sau inserează articole) și în același timp utilizatorul Șulea face același lucru, atunci SGBD-ul nu poate decide care dintre modificări trebuie preluate. Mecanismul de blocare (locking) blochează execuția operațiilor unui utilizator. Deoarece în cadrul conceptului client-server operațiile sînt preluate ca tranzacții, o blocare nu provoacă o întrerupere ci emite doar un mesaj din care să reiasă că tranzacția n-a putut fi executată.

Nivele de blocare

Mecanismele de blocare pot fi poziționate pe diferite nivele. Cea mai ieftină și cea mai simplă soluție o constituie blocarea unui întreg fișier atunci cînd doi utilizatori acced simultan la aceeași bază de date. Totuși în acest

Setul de comenzi SQL (standardul ANSI-86 și extensii)

Limbajul de manipulare a datelor - DML	Funcția: AVG	Predicat: SOME	CREATE INDEX ON
BEGIN DECLARE SECTION	Funcția: COUNT	Predicat: ANY	CREATE SYNONYM
CLOSE	Funcția: MAX	Predicat: ALL	CREATE TABLE
DECLARE CURSOR	Funcția: MIN	Predicat: operatori de comparație	CREATE VIEW
DELETE	Funcția: SUM	PREPARE	DROP INDEX
DELETE ... WHERE CURRENT OF	FROM	SELECT	DROP SYNONYM
DESCRIBE	GROUP BY	SELECT ... INTO	DROP TABLE
END DECLARE SECTION	HAVING	SUBSELECT	DROP VIEW
END-EXEC	INCLUDE	UNION	UPDATE
EXEC SQL	INSERT	UPDATE	STATISTICS
EXECUTE	INTERSECT	UPDATE ... WHERE CURRENT OF	Limbajul de control al datelor - DCL
EXECUTE IMMEDIATE	MINUS	WHENEVER	COMMIT
FETCH	OPEN	WHERE	GRANT
	ORDER BY	Limbajul de definire date - DDL	LOCK TABLE
	Predicat: BETWEEN	ALTER TABLE	REVOKE
	Predicat: EXISTS	COMMENT ON	ROLLBACK
	Predicat: IN		
	Predicat: LIKE		
	Predicat: NULL		

caz se poate bloca prelucrarea diferitelor tabele care aparțin aceleiași baze de date chiar și atunci când nu apar conflicte. În cazul unui număr mare de utilizatori aceasta ar putea conduce la o blocare permanentă. Utilizatorii trebuie să aștepte, în acest caz, mult timp pînă cînd baza redevine liberă. O astfel de soluție nu este deci economică.

O blocare la nivelul cîmpurilor individuale ar prezenta avantaje considerabile. Mai mulți utilizatori pot prelucra, în acest caz, aceeași tabelă fără să se blocheze între ei atît timp cît nu editează același cîmp. Totuși un astfel de mecanism de blocare este greu de realizat și presupune o reducere a performanțelor (vitezei) de prelucrare, deoarece tot timpul cîmpurile unui fișier trebuie verificate pentru a nu fi accesate simultan.

Deadlock

Un așa-numit deadlock poate să apară atunci cînd mai multe procese se blochează reciproc și pînă la urmă nu mai poate fi executat nici unul. Acest lucru se poate întîmpla, de exemplu, atunci cînd utilizatorul Moraru deschide tabela A, modifică date în această tabelă și vrea să copieze aceste date în tabela B, în timp ce utilizatorul Șulea tocmai a efectuat modificări în tabela B și vrea să le copieze în tabela A. Operațiile utilizatorilor Moraru și Șulea sînt executate ca tranzacții și pot fi încheiate doar după terminarea tuturor prelucrărilor. Dar întrucît tabelele A și B sînt blocate datorită accesării simultane, niciuna din cele două tranzacții nu poate fi încheiată și astfel se ajunge la un deadlock.

Un SGBD trebuie să observe apariția unui astfel de deadlock și să avertizeze utilizatorul asupra acestui lucru. Un deadlock poate fi evitat, de ex., dacă atunci cînd o tranzacție nu poate fi executată într-un anumit interval de timp este întreruptă. În acest caz trebuie executat un rollback automat pentru a reface starea inițială a sistemului.

Este ușor de înțeles astfel de ce noțiunea de tranzacție este atît de strîns legată de cea de mecanisme de blocare și ce influență au ele asupra calității securității datelor și deci a integrității.

SQL ca limbaj de interogare standard

Posibilitatea de a implementa baze de date distribuite conform conceptului

client-server, pe toate platformele hard, a impus necesitatea creării unui limbaj de interogare standard, a cărui sintaxă să fie aceeași pe toate calculatoarele. Structured Query Language, mai scurt SQL, a fost conceput la începutul anilor 80 ca limbaj de interogare pentru SGBD-ul IBM-ului DB2 disponibil pe calculatoare mari (mainframe). Ulterior American National Standards Institut (ANSI) și International Standards Organisation (ISO) au impus SQL ca limbaj de interogare standard pentru SGBD-urile de nivel ridicat. Practic toate SGBD-urile client-server dispun de o interfață SQL, al cărei set de comenzi corespunde cel puțin acestor norme și care adesea este extins cu comenzi suplimentare. Despre Embedded SQL se vorbește atunci cînd setul de comenzi SQL este imbricat într-un limbaj de programare, ca de ex. Cobol, Fortran, PL/1 sau C. SQL constă în principiu din trei componente: limbajul de definire a datelor (Data Definition Language - DDL), limbajul de manipulare a datelor (Data Manipulation Language - DML) și limbajul de control a datelor (Data Control Language - DCL).

Cu DDL se definește baza de date și tabelele conținute în ea. Tabelele sau cîmpurile pot fi deci inserate, șterse sau li se pot modifica atributele. De exemplu, tipul de cîmp char poate fi modificat în integer.

DML conține, ca elemente de limbaj, comenzi pentru modificarea datelor. Datele pot fi șterse, inserate, editate, însumate cu ajutorul unei funcții, poate fi calculată media aritmetică sau maximum unui cîmp. Tot aici sînt incluse toate tipurile de sortări, selecții și grupări. Așa-numitele instrucțiuni dinamice DML, care sînt utilizate în cadrul limbajelor de programare ca Embedded SQL, pot fi manipulate în timpul execuției programului sau li se pot varia argumentele. Deoarece aceste comenzi nu sînt încă standardizate, în funcție de implementare, pot să difere mult de la un limbaj de programare la altul.

DCL este foarte important mai ales în cazul sistemelor multiutilizator, deoarece el controlează securitatea sistemului și verifică drepturile de acces. În plus el permite o gestiune a tranzacțiilor și în caz de necesitate poate executa o refacere a bazei (rollback).

În principiu SQL oferă deci o interfață unitară între backend și diferitele frontend (programe de calcul tabelar, generatoare de rapoarte, prelucrare de texte etc.).

Deoarece odată cu dezvoltarea tehnologiei rețelelor, limitele dintre PC-uri, minicalculatoare și calculatoare mai dispar din ce în ce mai mult, conceptul client-server ca soluție integratoare cîștigă tot mai mult în însemnătate. SGBD-urile integrate rămîn interesante pentru o utilizare locală fără comunicații cu date externe, bazele de date gigant din lumea calculatoarelor mari se dovedesc a fi îmbătrînite și a nu mai fi moderne; fără a mai vorbi de costurile ridicate pe care le implică întreținerea unor astfel de sisteme.

Gupta SQL Base Server

SGBD-ul Gupta constă din mai multe componente: ca backend este utilizat SQL Base Server, care a fost realizat special pentru rețele locale LAN (Local Area Network). Din această cauză SQL Base Server suportă toate tipurile de rețele sub sistemele de operare DOS și OS/2. Se află în pregătire și o versiune Unix și în versiunea 5.0 SQL Base este disponibil și ca NLM (Netware Loadable Module) pentru Novell Netware 386. Pentru utilizarea în LAN, produsul a fost optimizat pentru OLTP (OnLine Transaction Processing) și pentru pregătirea și luarea de decizii (Decision Support). Aplicațiile OLTP cer pe de o parte actualizări multiple, concurente ale datelor, pe de altă parte reacții (răspunsuri) rapide. Aplicații clasice OLTP sînt, de ex., rezervarea de locuri la avion, sau aplicațiile agențiilor de asigurări. Cîteva caracteristici ne vor arăta cum reușește să obțină Gupta performanțe ridicate:

- procesorul SQL preia sarcinile sistemului de operare prin funcții proprii de Scheduling, Task Switching, Locking și Disk Caching.
- pentru securitatea datelor se ține un protocol combinat al activităților server-ului, în cazul în care mai mulți utilizatori își termină tranzacțiile aproape simultan cu COMMIT.
- buffer-ul de date permite acces mult-utilizator, astfel încît un utilizator poate accede și la datele utilizate simultan de altcineva.
- inserări și ștergeri multithreaded de linii din tabele.

Ca suport pentru luarea de decizii Gupta susține programe de calcul tabelar, interogări de baze de date și crearea de rapoarte. O funcție specială browse permite "răsfoirea" simultană în mulțimi de date diferite. Articolele pot fi actualizate oricând în succesiunea dorită.

Ca sistem de dezvoltare grafic pentru aplicațiile frontend este disponibil SQL Windows, care poate fi implementat sub Windows 3.0 pentru MS-DOS sau sub Presentation Manager pentru OS/2. Software-ul de management a bazei de date Quest permite utilizatorului final să accedă la date, să interogheze o bază de date SQL și să obțină rapoarte prin intermediul unei suprafețe utilizator grafice.

Informix SE și Online

Ca backend Informix oferă două server-e de baze de date: System-Engine (SE) și Online. Ambele respectă standardul de interfață ANSI-SQL/89 și pot lucra astfel cu toate programele frontend care respectă acest standard. Informix oferă următoarele programe utilitare: I-SQL, 4GL, 4 GL-Forms, Menus, ESQL/C, ESQL/Cobol și Storm. Prin intermediul componentei integrate Datalink, la date pot accede și programele Wingz (calcul tabelar și grafică de prezentare) și SmartWare (procesor de texte).

Informix SE și Informix-Online au fost concepute pentru domenii aplicative și piețe diferite. SE atinge cele mai bune performanțe la un număr mic până la mediu de utilizatori. Efortul de instalare și necesarul de memorie sînt mai reduse, gestiunea și dimensiunea bazei de date sînt adaptate unui număr mai mic de utilizatori.

Online este SGBD-ul uriaș care își relevă avantajele în cazul operării multiutilizator. Caracteristici semnificative sînt gestiunea unor volume mari de date și posibilitatea prelucrării distribuite (pot fi utilizate simultan mai multe baze de date). Pentru sporirea securității, Online permite oglindirea unor părți ale bazei de date, sau a întregii baze. Aceasta înseamnă că în permanență datele sînt memorate pe două suporturi distincte.

Informix Online este singurul SGBD, din cele prezentate în acest articol, care oferă și suport multimedia. El poate memora informații sub formă de text, gra-

fică, voce. Fiecare cîmp poate avea dimensiunea de pînă la 2 GB. Cu produsul suplimentar OnLine/Optical, datele pot fi stocate pe un disc optic.

SE și Online suportă baze de date distribuite, aceasta înseamnă că mai multe tabele ale aceleiași baze de date pot fi stocate pe medii de memorare și server-e diferite. Acest lucru permite o mai bună utilizare a resurselor într-o rețea locală.

Ingres SQL - Server

Caracteristicile principale ale SGBD-ului client-server Ingres sînt performanțe optime pentru OLTP (OnLine Transaction Processing) și suport pentru lucrul 24 ore din 24. Trei elemente fac din Ingres un SGBD inteligent: pe lîngă o gestiune simplă a datelor memorate (Data Management) mai sînt accesibile două facilități.

Gestiunea centralizată a regulilor unei întreprinderi prin așa-numitul Knowledge Management. Pînă acum aceste reguli trebuiau codificate în fiecare program de aplicație. Prin Knowledge Management oricît de complexe ar fi regulile ele pot fi gestionate central și deci trebuie definite o singură dată. Avantajele sînt următoarele:

- consistența datelor este asigurată
- programele utilizator devin mai simple și mai lizibile
- întreținerea regulilor este mai simplă
- administratorul bazei de date dispune de noi posibilități de acordare și revocare de drepturi de acces pentru utilizatori, grupe de utilizatori și aplicații.
- necesarul de resurse al utilizatorilor individuali poate fi supravegheat și gestionat central de administratorul sistemului.

Interogările utilizator care suprasolicită hardware-ul existent pot fi refuzate sau amîinate. Performanțele întregului sistem pot fi astfel mai ușor calculate.

Object Management este a doua facilitate care încearcă să facă din Ingres un SGBD inteligent. El permite gestiunea unor obiecte complexe, cum ar fi de exemplu coordonatele sau obiectele geografice. Pînă acum obiectele trebuiau definite în structuri de date simple. Object Management permite definirea unor noi tipuri de date, operatori și funcții care să fie gestionate la fel de rapid ca și tipurile de date standard.

Ingres oferă unele interactive pentru suportul unor suprafețe utilizator grafice (GUI), ca de exemplu pentru Windows 3.0. Cu ajutorul lui Ingres-Windows 4GL se pot crea baze de date într-un mediu de dezvoltare orientat obiect de generația a 4-a.

Pentru generarea automată de aplicații pentru terminale orientate caracter poate fi utilizat Ingres-Vision.

Oracle SQL-Server

OSE sau Open Software Environment este cuvîntul magic Oracle. Modelul OSE constă din șapte nivele:

1. Producători hardware: orice utilizator important trebuie să poată implementa un nou SGBD în mediul hard eterogen de care dispune. Oracle suportă hardware-ul tuturor producătorilor importanți, ca de ex., IBM, Nixdorf, Siemens, DEC, HP, Buii, Sun și Apple.

2. Arhitecturi hardware: IBM - Mainframe, sisteme multiprocesor VAX, stații de lucru RISC produse de Nixdorf sau arhitecturi SPARC produse de SUN sînt doar cîteva exemple din multitudinea de platforme existente azi în lumea calculatoarelor și pe care Oracle poate funcționa. OSE este de asemenea deschis pentru tehnologiile moderne din industria tehnicii de calcul cum ar fi multiprocesare dedicată sau simetrică, procesare paralelă, arhitecturi multi-computer sau diferitele realizări LAN.

3. Sisteme de operare: Această rubrică s-ar fi putut intitula "de la PC pînă la mainframe". MVS, VM, VSE pentru mainframe-uri IBM, System 6150 sub AIX, Siemens BS 2000, DEC/VAX cu VMS și Ultrix, peste 40 de derivate Unix, Macintosh, MS DOS și OS/2 sînt doar o parte din sistemele de operare suportate.

4. Standarde industriale. De fapt n-ar trebui să existe decît un standard, realitatea însă este alta. Pe lîngă interfața standard SQL și compatibilitatea cu DB2 al lui IBM, Oracle oferă interfețe și spre limbajele de programare, baze de date, aplicații și suprafețe grafice, utilitare frontend și interfețe de rețea și de comunicație. Implementarea SGBD-ului Oracle în mediul sistem propriu permite utilizatorului să păstreze mediul obișnuit de lucru. Cîteva dintre suprafețele utilizator posibile sînt enumerate în continuare: OSF/Motif, Open Look, Presentation Manager, Windows, DEC

Windows, suprafața utilizator Macintosh.

5. Procesare cooperativă: acesta este sloganul sub care Oracle integrează datele de bază situate în diferite locuri pe calculatoare provenind de la producători diferiți. În legătură cu arhitectura client-server, Oracle SQL Star comunică prin intermediul unor protocoale diferite cu lumea eterogenă a sistemelor de calcul. Colecțiile de date de pe PC-uri, mainframe-uri sau stații de lucru sînt gestionate central de către administrator, utilizatorul nu realizează de unde își extrage datele.

6. Resurse de informații: Bazele de date centrale, cum sînt bazele de date relaționale ale IBM-ului, DB2 și SQL/DS pot fi interogate prin intermediul lui Oracle SQL Connect din programe de aplicație care rulează pe un DEC/VAX sau pe un calculator Unix.

7. Modelele de integrare. Sistemul eterogen constituit din diferite platforme hard, S.O., rețele, baze de date este integrat într-un tot omogen.

Sybase SQL - Server

Arhitectura client-server Sybase constă din două familii de produse:

- SQL-Server preia gestiunea datelor și controlul tranzacțiilor independent de aplicațiile client și interfața utilizator
- SQL - Toolset cuprinde un set de instrumente prietenoase pentru dezvoltarea și operarea aplicațiilor.

Integritatea datelor și performanțe ridicate, aceasta este ceea ce Sybase dorește să realizeze cu SQL-Server. Pentru aceasta sînt pușe la dispoziție mai multe funcții. Așa-numitele Stored Procedures (obiecte create și compilate cu Transact SQL) împreună cu așa-numiții triggers (proceduri care sînt lansate automat în execuție atunci cînd cîmpurile respectă anumite condiții) sînt utilizați pentru asigurarea integrității datelor. Regulile valabile pentru întreaga întreprindere (business rules) sînt stocate central, pe server, și stau la dispoziția tuturor clienților. Acest concept permite înlăturarea codului redundant, ușurează întreținerea și asigură controlul central pentru protecția datelor întreprinderii.

Pentru atingerea unor performanțe ridicate este răspunzătoare arhitectura multi-threaded pe care se bazează SQL-Server. Ea cuprinde nucleul siste-

mului și un SQL-Task-Manager care preia funcțiile multiutilizator cum ar fi: scheduling, task switching, disk caching și locking. SGBD-ul este capabil ca în doar 1 MByte de memorie suplimentară să gestioneze 30 de utilizatori.

Prin intermediul așa-numitei arhitecturi Virtual-Server sînt utilizate la capacitate toate procesoarele dintr-un sistem multiproceosr.

Cu Open Server se încearcă utilizarea Sybase și într-un sistem eterogen. El permite atât clienților cît și server-ului să comunice cu SGBD-uri străine, baze de date nerelaționale, aplicații deja existente, sisteme în timp real, etc.

XDB-SQL-Server

XDB-SQL-Server este singurul SGBD prezentat aici, care în cea mai redusă formă a sa poate rula și pe un calculator cu doar 640 KB de memorie. Produsul bazat pe o arhitectură client-server este complet. El dispune de funcții GRANT/REVOKE pentru acordarea/revocarea de drepturi de acces, COMMIT, ROLLBACK pentru prelucrarea tranzacțiilor de mecanisme de blocare ca și de compatibilitate SQL deplină cu ANSI/ISO-DB2 și dialectele SAA-SQL.

Există două versiuni XDB-Server: o versiune MS-DOS, pentru care serverul are nevoie de protocolul Net BIOS și suportă pînă la 16 MB de memorie extinsă și o versiune OS/2 care are nevoie de cel puțin 2 MB memorie și de protocolul Net BIOS sau Named Pipes. Server-ul și stațiile de lucru pot funcționa sub sisteme de operare diferite.

Paleta de produse XDB mai conține pe lângă XDB-SQL-Server încă alte trei produse:

- XDB-SQL PLUS este un pachet single-user care conține XDB-SQL-Engine și SGBD-ul (DBMS), inclusiv Interactive SQL, Data Entry, Import/Export și Data Dictionary. Acest pachet mai conține și diferite utilitare, cum ar fi un generator de măști (forme), un generator de rapoarte și Procedural Language ca limbaj de programare.
- XDB-WORKBENCH permite crearea unor aplicații DB2 pentru mainframe-uri IBM și PC. Ca mediu de dezvoltare este utilizat limbajul Cobol.
- XDB-Tools sînt instrumente de dezvoltare pentru crearea de aplicații și versiuni runtime.

- XDB-EXT este o versiune single-user extinsă, care poate gestiona pînă la 16 MB de memorie extinsă.

- XDB-C este o interfață și o bibliotecă de utilitare pentru programe C care conține un precompilator C, Embedded SQL și API (Application Programming Interface) pentru comunicarea cu SQL Engine.

- XDB-Windows este un kit de dezvoltare pentru aplicații SQL sub Windows 3.0.

Novell Netware SQL

Și Novell oferă o serie de produse care ca și componente ale unui sistem de operare multiutilizator permit crearea unor baze de date client-server puternice. Ca backends sau SQL-Engines pot fi utilizate fie Netware Btrieve fie Netware SQL care în versiunea pentru AT 286 Netware 2.x pot fi cuplate static ca și VAP (Value Added Process) respectiv în versiunea 386 Netware 3.11 pot fi cuplate dinamic ca și NLM (Netware Loadable Module). Btrieve este un SGBD care a fost creat special pentru un File Handling rapid. Dintr-un limbaj de programare prin intermediul unor proceduri de sistem se poate accede secvențial la datele Btrieve.

Netware SQL este un SGBD cu interfață SQL standard prin care se poate comunica cu diferitele front-end. Pentru proiectanți, Netware SQL oferă o interfață compatibilă atât cu ANSI/SQL cît și cu specificațiile IBM SAA-SQL. Netware SQL poate fi utilizat în mai multe moduri:

- programele aplicative ca Lotus 1-2-3 pot accede direct la datele de pe backend prin intermediul Netware.
- Xtrieve PLUS al lui Novell este de asemenea un frontend bun dacă utilizatorul vrea să accedă la un produs deja integrat în Netware și nu vrea să cumpere alte pachete soft. Xtrieve dispune de un sistem de interogare condus prin meniuri (Query Manager) care permite și generarea de rapoarte.
- Pot fi create aplicații proprii utilizînd instrumente de dezvoltare. Un astfel de instrument este RaSQL al firmei Communication Horizon Clipper.
- Proiectanții pot scrie aplicații proprii utilizînd un limbaj de programare standard și interfața utilizator XQL.

(R.M.)

Baze de date

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Producător:	Gupta Technologies	Informix Software	Informix Software	Ingres	Oracle	Sybase	XDB Systems Inc.
Prețuri:							
Modul de bază	2.995 - 29.995 DM+ 14% TVA	la cerere	la cerere	la cerere	nec.	la cerere	peste 3900 DM
Extensii	nec.	Standard Engine (SE)	XA/TP, Optical OnLine	la cerere	nec.	la cerere	Precomp. MS-C 8.0, Cobol
Licență (Runtime)	1200DM+14%	la cerere	la cerere	la cerere	nec.	la cerere	la cerere
Date tehnice:							
Dimens. program în KB	600	700	1130	~ 50MB(Unix)	nec.	Server: 1500	7 MByte
Memoria rec. în MB	4 - 8	2	6	min. 6	nec.	16, (min.8)	peste 2
Versiune Windows 3.0	x	instr. suportă Windows 3.0	instr. suportă Windows 3.0	da, Ingres/ Windows4GL	nec.	ca Frontend	x
Expanded Memory	x	x	-	-	nec.	x	x
Extended Memory	-	x	x	x	-	x	-
Suport coprocesor	-	x	x	x	-	x	-
Sisteme de operare suportate:							
MS-DOS	x	x	-	da; Tools, ABF, Vision, ESQ, Win. 4GL, Net	x	numai Client	x
S.O. Macintosh	-	x	x	-	x	numai client	-
S.O. Next	-	x	planificat	-	x	x	-
Derivate UNIX	SUN-OS	x	x	x	>50 var. UNIX	x	încă nu
OS/2	x	x	-	x	x	x	x
DEC VMS	Connectivity și NetWare 386	-	-	x	x	x	-
Rețele suportate:							
Novell Netware 2.1	x	File-Sharing	-	x	x	-	x
Novell Netware 3.1	x	-	x	x	x	x	x
LAN-Manager	x	-	-	x	x	x	x
IBM PC-LAN	x	File-Sharing	File-Sharing	-	x	x	x
Banyan VINES 4.0	x	File-Sharing	File-Sharing	în pregătire	x	x	x
Altele	nec.	AT & T StarGROUP	AT & T StarGROUP	nec.	nec.	nec.	nec.
Protocoale suportate:							
Token Ring	x	x	x	func. de rețea	nec.	x	func. de rețea
CSMA/CD	x	x	x	func. de rețea	nec.	-	func. de rețea
Arcnet	x	x	x	func. de rețea	nec.	-	func. de rețea
Ethernet	x	x	x	func. de rețea	nec.	x	func. de rețea
StarLAN	x	x	x	func. de rețea	nec.	x	func. de rețea
TCP/IP	x	x	x	x	x	x	da
IOS TP4	x	-	-	func. de rețea	-	-	func. de rețea
IBM NetBEUI/DLC	x	-	-	func. de rețea	x	-	func. de rețea
Altele	nec.	nec.	nec.	nec.	SPX/IPX, DECnet, APPC, Named Pipes, și altele	nec.	Netbios, SPX, Named Pipes, APPC, Win. (loc.)
Volumul bazelor de date:							
Nr. max. tabele	nelimitat	nelimitat	14 milioane	nelimitat	nelimitat	32767	nelimitat
Nr. tabele desc. simultan	nelimitat	funcție de S.O.	32000	nelimitat	nelimitat	configurabil	funcție de S.O.
Nr. de articole per tabelă	nelimitat	2,1 miliarde	4,3 miliarde	nelimitat	nelimitat	2 miliarde	2 miliarde
Nr. cimpuri per articol	254	32767	32767	300	255	250	400
Lungime cimp. text în octei	254	32767	2 MB	2 KB	255	2 GB	4056
Nr. max. tabele înluite	nelimitat	funcție de S.O.	32000	30	orice	16	360
Cimpuri:							
Alfanumeric	x	x	x	x	x	x	x
Numeric	x	x	x	x	x	x	x

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Nr. zecimale	15	pin la 32	pin la 32	definibil	41	definibil	configurabil
Dat	x	x	x	x	x	x	configurabil
Timp	x	x	x	x	x	x	configurabil
Memo	x	x	x	-	x	x	x
Cîmpuri definite de utilizator	-	-	-	prin Object Management	-	-	-
Cuplare de imagini/grafic	x	-	BLOB	x	-	x	-
Calcul trigger	-	x	x	prin Knowledge Management	-	x	-
Gestione date:							
Ver. aut. a plauzabilității	x	x	x	x	x	x	x
Verificare duplicare articol	x	x	x	programabil	x	x	x
Cîmp obligatoriu	x	x	x	x	x	x	x
Conversie automată majuscule/minuscule	-	comand	comand	x	x	în instr. SELECT	x
Nr. de formate dat	definibil	orice	orice	orice	orice	nec.	5
Numerotare automată articole	-	x	x	x	x	la Stored Procedure	x
Preluare date din articolul anterior	-	x	x	x	x	x	x
Reorganizare automată/Reindexare	-	comand	comand	x	x	x	x
Import/Export:							
Nr. formate import/export	7	1	1	nec.	nec.	2	x
DIF	x	prin Wingz	prin Wingz	nec.	nec.	-	x
dBase III	x	prin Wingz	prin Wingz	nec.	numai import	-	x
dBase III Plus	x	prin Wingz	prin Wingz	nec.	numai import	-	x
dBase IV	x	prin Wingz	prin Wingz	nec.	nec.	-	x
Lotus	-	prin Wingz	prin Wingz	nec.	numai Import	-	x
Framework	-	prin Wingz	prin Wingz	nec.	nec.	-	nec.
Excel	-	prin Wingz	prin Wingz	nec.	nec.	-	nec.
Quattro	-	prin Wingz	prin Wingz	nec.	nec.	-	nec.
ASCII	x	x	x	x	x	x	fixed ASCII, free ASCII
DB2	x	cu conv. ASCII	OpenView	prin Gateway	nec.	-	DBMAUI, DSNTIAUL
Format definit de utilizator	-	x	x	x	x	x	-
Altele	nec.	x	x	nec.	nec.	binar	Wordperfect: Mail merge files, Multiplan: SYLK
Întreținere date:							
Ștergere simultană a mai multor articole	x	x	x	x	x	x	comandă SQL
Actualizare simultană a mai multor articole	x	x	x	x	x	x	comandă SQL
Recalculare automată	x	x	x	x	x	prin Trigger	nec.
Înlocuire automată	x	x	x	x	x	prin Trigger	nec.
Altele	nec.	nec.	nec.	nec.	nec.	nec.	Fetch Previous în ESQL, ISQL, Proceduri
Sortare date:							
În ordine crescătoare	x	x	x	x	x	x	max. 64 criterii
În ordine descrescătoare	x	x	x	x	x	x	max. 64 criterii
Alfanumeric	x	x	x	x	x	x	la alegere
Numeric	x	x	x	x	x	x	la alegere
Sortarea tuturor cîmpurilor	x	x	x	x	x	x	la alegere
Sortare doar a cheilor	x	x	x	x	x	nec.	la alegere
Nr. max. cîmp. sortate sim.	nelimitat	nr. cîmpurilor	nr. cîmpurilor	300	orice	16	64

Baze de date

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Indexare date:							
Nr. max. cimpuri indexabile	nelimitat	nelimitat	96	300	oricite	16	400
Nr. max. fisiere index deschise simultan	nelimitat	nelimitat	nelimitat	nelimitat	oricite	16	functie de S.O.
Act. aut. a fisierelelor index	x	x	x	x	x	x	x
Indexare aut. la import date	x	x	x	x	x	x	x
Altele	nec.	nec.	nec.	nec.	nec.	nec.	Clustered Index: 1 per tabel
Chel și indexare:							
Prime key	x	x	x	x	x	x	x
Foreign key	x	x	x	prin Knowledge Management	x	x	x
Integritate referențială	x	x	x	prin Knowledge Management	x	x	x
Index	x	x	x	x	x	x	simplic și unic
Query Optimizer	x	x	x	x	x	x	x
Altele	nec.	nec.	nec.	nec.	nec.	nec.	Compound Key: max. 20 comp.
Căutare date:							
Utilizare comenzi SQL	x	x	x	x	x	x	In Interactive SQL, Proceduri, Embedded SQL
Depistarea tuturor art. care satisfac condiția de căutare	x	x	x	x	x	x	
Cupare logică căutări	x	x	x	x	x	x	x
Căutare fonetică		x	x	-	x	x	-
Căutare în cimpuri memo	x	x	x	-	-	x	N/A
Generator de cereri	x	Utilitarul I-SQL	Util. dbaccess	x	x	x	-
Memorarea cererilor	x	x	x	x	x	x	x
Rapoarte:							
Rapoarte standard aut.	x	I-SQL,4GL	I-SQL,4GL	x	x	x	prin dialog și ca procedură batch
Rapoarte grafice	x	Storm,Wingz	Storm,Wingz	-	-	alți ofertanți	produse străine
Memorare rapoarte	x	x	x	x	x	x	x
Formulare raport	x	I-SQL,Storm	I-SQL,Storm	x	x	x	x
Macroui:		I-SQL,Storm, Wingz,4GL	I-SQL,Storm, Wingz,4GL				
Editor de macroui		x	x	4GL	nec.	-	x
Înregistrare automată	-	x	x	4GL	nec.	-	x
Macro la o apăsare de tastă		x	x	4GL	nec.	-	x
Limbaj de macroui propriu		x	x	4GL	nec.	-	SQL cu extensii
Protecție date:							
Parolă program	x	x	x	x	x	x	x
Parolă fișier	x	-	-	x	-	x	autorizare (DBA)
Parolă tabel	-	-	-	x	-	-	autorizare (DBA)
Parolă cimp				x	-	-	-
Protecție automată date	x	la nivel de S.O.	x	x	x	-	menu fig. LOG
Backup automat	x	la nivel de S.O.	x	x	x	-	menu fig. LOG
Criptare date	x	implicit	implicit	-	-	-	-
Funcții Recovery:							
Re-Index	x	x	x	x	x	-	x
Funcție Undo	x	-	-	x	x	-	x
Funcție Redo	x	-	-	x	x	-	x
Rollback	x	x	x	x	x	x	x
Rollforward	x	x	x	x	x	x	x
Fast Recovery	x	-	x	x	x	x	x

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Prelucrare tranzacții	x	x	x	x	x	x	da/nu la rulare
Tranzacții imbricate	x	-	-	-	-	x	-
Verificări la nivel cimp	x	x	x	x	x	x	Range/Defaults/Values
Verificare structurală	x	x	x	x	x	x	ver. tip și conv.
Verificare procedurală	-	x	x	x	x	x	via SQL în gen. de formulare
Cardinalitate	x	x	x	x	x	nec.	nec.
Commit în mai multe trepte	-	-	-	Commit 2 faze	x	x	-
Sincronizare	-	funcie de SGF	x	x	x	x	-
Moduri de blocare recunoscute:							
Blocare bază de date	nec.	x	x	x	x	-	prot. prin parolă
Blocare tabelă	nec.	x	x	x	x	x	Exclusive și Share Intentional Locks
Blocare articol	nec.	x	x	x	x	-	x
Blocare multiplă articole	nec.	x	x	x	x	-	x
Deed-Locks	x	atenuare prin Nowait	recunoaștere și rezolvare	recunoaștere și tratare	x	sînt depistate de server	rec. automată la Autocommit
Altele	Page-Level	Page-Level	Page-Level	citire fără blocare	-	blocare pag.	nec.
Concept de operare:							
Windows 3.0	Quest, SQLWindows	prin instr. apl.	prin instr. apl.	x	x	suport	versiune Windows
Suport mouse	x	prin instr. apl.	prin instr. apl.	x	x	x	nec.
Tehnica ferestrelor	x	prin instr. apl.	prin instr. apl.	x	x	x	la DOS și OS/2
Lucrul condus de meniuri	x	prin instr. apl.	prin instr. apl.	x	x	x	la DOS și OS/2
Programare:							
Limba de prog. propriu	x	4GL	4GL	4GL	PL/SQL	x	XDB PRO
Nr. limb. prog. suportate	nec.	5 + de la teri	5 + de la teri	6	nec.	6	3
Interfață C	x	x	x	x	x	x	MS-C6.0, API Interface și Embedded SQL-C
Interfață Cobol	x	x	x	x	x	x	Realia Cobol, Micro Focus Work-bench, Embedded SQL Cobol dinamic și static
Interfață Basic	x	-	-	x	-	x	-
Interfață Ada	-	sub Unix	sub Unix	x	-	x	-
Interfață Pascal	x	-	-	x	-	x	-
Interfață Fortran	-	sub Unix	sub Unix	x	x	x	-
Alte interfețe	interfață de prog. generică	interfață CALL	interfață CALL	-	-	nec.	XDB-Link la IBM Host DB2
Limba 4GL integrat	x	4GL	4GL	x	x	pentru Stored Procedures	x
Compilator integrat	x	compilator 4GL	compilator 4GL	x	-	x	pentru formulare și proceduri
Versiune Runtime	-	x	x	x	-	de la terți	la cerere
Instrumente grafice	SQLWindows, Quest, Chart Builder	Wingz	Wingz	x	x	-	produse străine
Generatoare:							
Generator de aplicații	x	4GL	4GL	x	x	x	în gen. de măști
Generator de măști	x	4GL Forms	4GL Forms	x	x	x	x
Generator de meniuri	x	Menus	Menus	x	x	x	x
Generator de rapoarte	x	x	x	x	x	x	x
Generator de etichete	x	x	x	x	x	-	nec.

Piața românească

a) Firma b) Telefon	a) Model b) Producător c) Procesor d) Preț informativ	a) Tip carcasă b) Prod. procesor c) Tact(MHz) d) Waitstates	a) RAM stand. (MB) b) RAM max. (MB)	a) Harddisk standard (MB) b) Producător c) Timp de acces (ms)/Interleave d) Harddisk max. (MB)	a) Interfață/ Tip controller b) Nr. floppy 5.25" c) Nr. floppy 3.5"	a) Standard grafic b) Rezoluția (pixeli) c) Nr. max. culori d) Monitor (inch)	a) Nr. interfețe paralele/seriale/mo use b) Arhitectură bus c) Sloturi d) Din care libere 8/16/32/64 biți 8/16/32/64	a) Sistem de operare b) Manuale c) Garanție d) Termen livrare
PC-uri AT286								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80286 d) 252.485 lei	a) - b) Intel c) 12 MHz d) -	a) 1 MB b) -	a) 40 MB (+ 75.175 lei) b) - c) 23 ms/- d) -	a) IDE b) 1 c) 1	a) MDA b) - c) - d) 14"	a) - b) - c) 18 luni d) -	a) - b) da c) 18 luni d) -
a) Elinco S.A. b) 90/12.20.30	a) Networks b) U.S.A. c) 80286 d) 375.000 lei	a) Baby b) Harris c) 20 MHz d) 0	a) 1 MB b) 16 MB	a) 40 MB b) Seagate c) 20 ms/1:3 d) oricit	a) IDE b) 1 c) 1	a) SVGA 512K b) 1024 x 768 c) 16 d) 14"	a) 1/2/0 b) ISA c) 2/5/0/0 d) 1/4/0/0	a) MS-DOS 5.0 b) da c) 12 luni d) 30 zile
a) System Plus SRL b) 90/79.13.91	a) A-Tronic b) - c) 80286 d) 350.322 lei	a) Slimline b) - c) 16 MHz d) -	a) 2 MB b) -	a) 80 MB b) - c) - d) -	a) IDE b) 1 c) 1	a) VGA 256K b) - c) monocrom (454.224 lei) d) 14"	a) - b) - c) - d) -	a) - b) - c) - d) -
PC-uri AT386SX								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80386SX d) 363.899 lei	a) intel b) - c) 20 MHz d) -	a) 1 MB b) -	a) 80 MB (+ 131.435 lei) b) - c) 19 ms/- d) -	a) IDE b) 1 c) 1	a) MDA b) - c) - d) 14"	a) - b) - c) - d) -	a) - b) da c) 18 luni d) -
a) Elinco S.A. b) 90/12.20.30	a) Ares b) U.S.A c) 80386SX d) 540.000 lei	a) Baby b) AMD c) 25 MHz d) 0	a) 2 MB b) 16 MB	a) 52 MB b) Quantum c) 14 ms/1:3 d) oricit	a) IDE b) 1 c) 1	a) SVGA 512K b) 1024 x 768 c) 16 d) 14"	a) 1/2/0 b) ISA c) 1/5/1/0 d) 1/3/1/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
a) MedComp DHS b) 90/12.30.37	a) Copam b) - c) 80386SX d) 2.555 DM	a) - b) Intel c) 8/20 MHz d) -	a) 2 MB b) 8 MB	a) 40 MB b) - c) 19 ms/- d) -	a) IDE b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) MS-DOS 5.0; Windows 3.0 b) - c) - d) -
a) System Plus SRL b) 90/79.13.91	a) Iverson b) U.S.A c) 80386SX d) 620.000 lei	a) - b) - c) 16 MHz d) -	a) 2 MB b) -	a) 80 MB b) - c) -/ d) -	a) IDE b) 1 c) 1	a) VGA b) - c) - d) 14"	a) - b) - c) - d) -	a) MS-DOS 5.0 b) da c) - d) -
PC-uri 386								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80386 d) 428.466 lei	a) intel b) - c) 25 MHz d) -	a) 1 MB b) -	a) 120 MB (+ 169.265 lei) b) - c) 19 ms/- d) -	a) IDE b) 1 c) 1	a) MDA b) - c) - d) 14"	a) - b) - c) - d) -	a) - b) - c) 18 luni d) -

a) Elinco S.A. b) 90/12.20.30	a) Sonic b) Ares USA c) 80386DX d) 940.000 lei	a) Mini Tower b) AMD/ Intel c) 40/ 33 MHz d) -	a) 4 MB b) 32 MB c) - d) -	a) 105 MB b) Quantum c) 14 ms/ 1:3 d) oricit	a) IDE/ SCSI b) 1 c) 1	a) SVGA 1024K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) ISA c) 0/6/1/0 d) 0/4/1/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
a) MedComp DHS b) 90/15.56.53	a) Copam b) - c) 80386DX d) 4.475 DM	a) Intel b) 8/33 MHz c) - d) -	a) 4 MB b) 16 MB c) - d) -	a) 105 MB b) - c) 15 ms d) -	a) IDE b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) AT-BUS c) - d) -	a) MS-DOS 5.0 b) da c) - d) -
a) System Plus SRL b) 90/79.13.91	a) A-Tronic b) - c) 80386 d) 865.646 lei	a) - b) 33 MHz c) - d) -	a) 4 MB b) - c) - d) -	a) 120 MB b) - c) - d) -	a) IDE b) 1 c) 1	a) VGA 256K b) - c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) - b) - c) - d) -
PC-uri 486SX								
a) Elinco S.A. b) 90/12.20.30	a) Ares b) USA c) 80486SX d) 1.150.000 lei	a) Full Tower b) Intel c) 20 MHz d) -	a) 4 MB b) 32 MB c) - d) -	a) 240 MB SCSI b) Quantum c) 13 ms/ 1:3 d) oricit	a) SCSI b) 1 c) 1)	a) SVGA 1024K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) ISA c) 0/6/2/0 d) 0/4/2/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
a) MedComp DHS b) 90/15.56.53	a) Copam b) Intel c) 80486SX d) 3.910 DM	a) Intel b) 8/20 MHz c) - d) -	a) 4 MB b) 16 MB c) - d) -	a) 105 MB b) - c) 15 ms/ - d) -	a) IDE b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) - d) 14"	a) 1/2/0 b) AT-BUS c) - d) -	a) MS-DOS 5.0 b) da c) - d) -
PC-uri 486								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80486 + coprocesor Weitek d) 4167	a) Tower Case b) Intel c) 33 MHz d) -	a) 16 MB b) - c) - d) -	a) 80 MB, 128 KB cache b) - c) - d) -	a) - b) 1 c) 1	a) SVGA b) 1024 x 768 c) - d) 14"	a) - b) - c) - d) -	a) - b) - c) 18 luni d) -
a) Elinco S.A. b) 90/12.20.30	a) Ares b) USA c) 80486DX d) 1.390.000 lei	a) Full Tower b) Intel c) 50 MHz d) -	a) 4 MB b) 32 MB c) - d) -	a) 340 MB SCSI b) Quantum c) 11 ms/ 1:3 d) -	a) SCSI b) 1 c) 1	a) SVGA 1024K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) ISA c) 0/6/2/0 d) 0/4/2/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
a) MedComp DHS b) 90/15.56.53	a) Copam b) - c) 80486DX d) 5.390 DM	a) Intel b) 8/33 MHz c) - d) -	a) 4 MB b) 16 MB c) - d) -	a) 105 MB b) - c) 15 ms/ - d) -	a) - b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) MS-DOS 5.0 b) da c) - d) -
a) System Plus SRL b) 90/79.13.91	a) A-Tronic b) - c) 80486 d) 1.132.383 lei	a) Full Tower b) Intel c) 33 MHz d) -	a) 4 MB, 64 KB cache b) - c) - d) -	a) 200 MB b) - c) - d) -	a) IDE b) 1 c) 1	a) VGA 256K b) - c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) MS-DOS 5.0 b) da c) - d) -

Nota redacției:

Deoarece pînă la închiderea ediției am primit foarte puține răspunsuri la ancheta inițiată, vom reveni cu completările de rigoare de îndată ce vom dispune de informații suplimentare. Rugăm firmele care doresc să ne ajute în prezentarea unei ogizi cît mai obiective a pieței românești să ne comunice tipurile de echipamente de tehnică de calcul, pachete software, consumabile, cursuri, servicii, etc., care fac parte din oferta firmei proprii, pentru a le putea expedia următoarele chestionare. Prețurile prezentate au un caracter pur informativ, fiind valabile în ziua în care noi am primit informațiile. Pentru informații la zi vă rugăm să vă adresați direct firmelor în cauză.

PC-uri extensibile (2)

În prima parte a articolului, am prezentat conceptul de PC extensibil, PC conceput cu posibilitatea schimbării identității (tipul microprocesorului sau și frecvența tactului).

Ideea cea mai importantă în legătură cu PC-urile extensibile este aceea că, în majoritatea cazurilor, trăsătura de extensibilitate nu implică modificări esențiale în planul prețului și a performanțelor, ceea ce le face atrăgătoare și pentru cei care nu au de gând să-și extindă PC-ul în viitor.

Important este de asemenea și modul în care este implementată trăsătura de extensibilitate, deci tehnologia de extindere, care determină în mod hotărâtor restul aspectelor care caracterizează un PC extensibil: prețul configurației de bază, prețul realizării extinderilor disponibile, ergonomia realizării unei extinderi.

Cea mai veche tehnologie de extindere este cea care derivă din utilizarea sertarelor cu fund de sertar pasiv. În această tehnologie, placa procesor este un modul ca oricare altul, putând fi schimbată cu alta, cu condiția asigurării compatibilității cu bus-ul. Sertarul cu fund pasiv a fost forma constructivă cea mai puțin utilizată. Creșterea gradului de integrare a condus la posibilitatea realizării unor produse monoplacă (single board computer) sau a înglobării principalelor funcții ale unui sistem într-o placă de bază (placă sistem) care apare ca un fund de sertar activ pentru un sertar în care se introduc adaptoarele și extensiile. Creșterea vitezei de lucru a componentelor a scos în evidență din ce în ce mai mult limitările de viteză impuse de fundul de sertar pasiv (8-12Mhz) ceea ce a dus la alegerea de noi soluții constructive. Atât timp cât o soluție monoplacă nu este, în general, realizabilă și nici dorită (duce la scăderea flexibilității sistemului), soluțiile similare fundului de sertar sînt un rău necesar. Esențial este ca sistemul să fie în așa fel croit încît la nivelul interconexiunilor dintre module, comunicația sa să se poată face și la o viteză mai mică fără a fi afectate performanțele sistemului.

În cazul PC-urilor, bus-ul sistem standard (AT BUS) care era satisfăcător

ca și viteză în momentul apariției PC AT-urilor, a devenit între timp prea "îngust". Ca atare, mișcarea IBM-ului cu bus-ul MCA, poate părea prea bruscă, dar este justificată.

Revenind la tehnologia sertarului cu fund de sertar pasiv, trebuie spus că aceasta a fost promițătoare în cazul mașinilor CP/M de 8 biți, dar a căzut în dizgrație o dată cu apariția PC-urilor, fiind cunoscute foarte puține PC-uri în care să fie utilizată: Wang Professional PC în 1985, Tyse și Zenith în 1987 și, de ce nu, TELEROM PC fabricat cîndva la IEIA Cluj.

O altă tehnologie de extindere, utilizată efectiv, se bazează pe ideea "spargerii" plăcii de bază în două părți: una cu componentele independente de tipul și viteza microprocesorului și alta cu componentele specifice printre care microprocesorul însuși. În realitate, nu se poate face o separare atît de strictă între ceea ce este dependent și ceea ce este independent de tipul și viteza microprocesorului, trebuind să fie făcute compromisuri legate de realizarea unui raport preț/performanță dorit.

Prima parte este noua placă de bază. A doua este modulul de extensie sau placa procesor. Modulul de extensie se introduce într-un slot special, ale cărui caracteristici depind de felul în care s-a făcut "spargerea". Principala problemă o constituie relația dintre procesor și memorie. Soluția punerii memoriei pe modulul de extensie, deci implicit, cînd este cazul, și a cache-ului, este mai scumpă dar, în cazul unor fabricanți, se poate obține o reducere de preț prin recuperarea de către aceștia a unor componente.

Dacă memoria rămîne pe placa de bază, atunci conexiunea trebuie să fie de mare viteză și cu o lățime a bus-ului la nivelul microprocesorului celui mai puternic. Dacă nu se respectă aceste condiții, performanțele sistemului vor fi afectate drastic. Existența unui cache pe placa procesor (la 486 este chiar pe chip), atenuează scăderea de performanță datorată unei conexiuni procesor-memorie necorespunzătoare.

În cazul extinderii de la 386 la 486, unii producători au ales soluția înlocuirii

microprocesorului 386 cu un modul care conține microprocesorul 486 și cîteva circuite de adaptare care fac ca modulul să fie compatibil la nivelul pinilor cu un 386. Acest modul se introduce direct în soclul microprocesorului 386.

Schimbarea microprocesorului poate conduce la necesitatea unor schimbări în BIOS. Problema este rezolvată în mod diferit de la un fabricant la altul. Soluțiile cele mai comode pentru utilizator sînt cele care nu-i cer acestuia nici o operație suplimentară în afara schimbării modulului procesor. Vă prezentăm două dintre acestea: ROM-urile de BIOS sînt pe modulul de extensie sau BIOS-ul configurației inițiale se poate adapta tuturor procesoarelor din extensiile posibile. O soluție interesantă este cea cu EPROM care cere utilizatorului să ruleze un program de modificare a conținutului EPROM-ului funcție de tipul microprocesorului instalat. În fine, se utilizează și soluția banală a schimbării efective a ROM-urilor pe placa de bază.

În continuare vom prezenta cîteva tipuri de PC-uri extensibile. Ne vom referi în special la caracteristicile legate de extensibilitate. În general, PC-urile prezentate sînt fabricate de firme cu un bun renume pe piața occidentală.

1. ALR Business VEISA

Este produs de firma ALR (Advanced Logic Research). În configurația de bază conține o placă procesor 386/33. Sînt disponibile două module de extensie cu 486/25, respectiv 486/33. ALR practică o reducere de preț pentru cei care trimit firmei vechiul modul. Pentru extensii, PC-ul este prevăzut cu trei slot-uri speciale. Unul este ocupat de placa procesor. În celelalte două se pot introduce opțional un modul de cache și un modul de extensie de memorie.

2. ALR Modular Processor System

Acest tip de PC are opțiuni în toate trei arhitecturile: ISA, EISA și MCA. Sînt disponibile module procesor pentru 386/25, 386/33, 486/25 și 486/33. De curînd a apărut și un modul cu 486SX. Pe placa procesor exista procesul, soclu pentru coprocesor și, la 386/33, so-

ciu pentru chip-ul Intel Smartcache. Și în cazul acestor PC-uri, ALR practică reduceri de preț în cazul recuperării procesorului vechi.

3. ALR SX Power Flex

Acest tip de PC extensibil, produs de firma ALR, oferă extensibilitate, prin intermediul modulelor procesor, de la 286 la 486SX trecînd prin 386SX/16 și 386SX/20. Datorită faptului că memoria este pe placa de bază, iar legătura cu modulul procesor se face printr-un bus de 16 biți (configurația cu 486DX/25 are performanțe foarte slabe în raport cu PC-urile 486DX/25 adevărate, fiind comparabile cu cele ale unui 386DX/33. Cu excepția microprocesorului 286, restul sînt recuperate contra cost de ALR în momentul cumpărării unui modul de extensie.

Toate PC-urile extensibile produse de ALR se caracterizează prin modul simplu de realizare a extensiei. Singurele operații necesare sînt scoaterea modului vechi și introducerea celui nou.

4. Amkly 386/25e

Produs de firma Amkly Systems, 386/25e este un PC extensibil cu bus EISA. Are două opțiuni: 386DX/25 și 486DX/25. Extensia se realizează prin plăci procesor care conțin pe lângă microprocesor, patru slot-uri pentru SIMM-uri și controller-ul de cache Intel 82359. Modulul 386 conține și un soclu pentru coprocesor, prețul extinderii poate fi redus prin trimiterea la firmă a vechiului modul 386DX/25 de pe care au fost mutate SIMM-urile pe noul modul, 386/25e este proiectat în așa fel încît va putea găzdui, în viitor, și extensii mai rapide de 33, 50 sau 66 Mhz.

5. AST Premium II

Produse de firma AST Research, aceste PC-uri extensibile oferă performanțe peste medie (cu excepția adaptorului video) pentru următoarele module procesor: 386SX/20, 386DX/33, 486SX/20 și 486DX/33. Modulele procesor, numite FAST board, se introduc în oricare din cele trei slot-uri ISA extinse la 32 biți (arhitectura CUPID, proprietatea firmei AST). Pe placa procesor se pot pune pînă la 16 Mocteji de RAM. În plus se pot pune plăci de memorie în cele două slot-uri CUPID libere. Extinderea constă în simpla schimbare a

plăcii procesor. BIOS-ul detectează tipul FASTboard-ului instalat, luînd măsurile care decurg din modificarea tipului/vitezei microprocesorului.

6. Copam B Serie

Firma Copam oferă una dintre cele mai largi game de extensii: 386SX/16, 386DX/25, 386DX/33, 486DX/25 și promite 386SX/20, 386DX/40 și 486DX/50. Tehnologia de extindere este cea a modulelor procesor care se introduc într-un slot special. Pe modulul de extensie se află, în afară de microprocesor, un soclu pentru coprocesor, memoria RAM, BIOS-ul și CMOS-ul, practic toate circuitele a căror funcționalitate trebuie să țină pasul cu microprocesorul. Este remarcabil faptul că prețul modulelor de extensie nu este cu mult mai mic decît cel al firmelor care pun pe modulul de extensie numai procesorul și soclul coprocesorului.

7. Dell System 333P

Produs de firma Dell Computer Corp., 333P conține, în configurația de bază, un 386DX/33 și realizează performanțe modeste. Cu un modul suplimentar de cache, performanțele cresc spectaculos, ducînd sistemul între mașinile bune. Microprocesorul 386DX este înglobat în placa de bază. Pentru extindere la 486DX/33, trebuie achiziționată o placă procesor care se introduce într-un slot al plăcii de bază. Odată instalat, microprocesorul 486 va înhiba microprocesorul 386 de pe placa de bază, nefiind practic nevoie ca acesta să fie scos. Dacă sistemul inițial conține cache, acesta devine inutil ba chiar trebuie scos pentru a face loc modulului procesor 486. Proiectanții au mizat pe cache-ul intern din 486. Trebuie scos de asemenea și coprocesorul 387 dacă acesta există. În plus, trebuie schimbat și BIOS-ul. Se observă că, la acest tip de PC-uri, realizarea extinderii este ceva mai laborioasă și presupune scoaterea din sistem a unor componente pe care Dell nu le recuperează contra cost.

8. Everex STEP386 cu InSTEP486

Firma Everex Systems produce PC-ul STEP386 în versiunile 20, 25 și 33 Mhz. Aceste PC-uri pot fi extinse la 486 de frecvență corespunzătoare. Extinderea se face prin introducerea plăcii InSTEP486 în soclul microprocesorului

386. Această operație este foarte dificilă și cu un mare risc de distrugere a modului de extensie care costă "numai" 2.999\$. În plus, trebuie schimbat și BIOS-ul. Cine poate trece cu vederea aceste dificultăți, va beneficia de un 386 și apoi de un 486 de foarte bună calitate.

9. Tandon Option

Tandon oferă o serie completă de module procesor: 286/12, 386SX/20 fără și cu cache, 486SX/20 și 486DX/33. PC-urile extensibile Tandon se remarcă prin procedura extrem de simplă de schimbare a plăcii procesor. Modulul procesor este închis într-o cutie metalică, schimbarea fiind la fel de simplă ca și schimbarea unei dischete în drive. Deoarece memoria este situată pe placa de bază și bus-ul de date este de 16 biți, configurația 486/33 va avea performanțe mai scăzute în lucrul cu memoria.

10. Wearnes-seria Boldline

PC-urile Wearnes sînt realizate pe baza unor proiecte ALR. Seria Boldline constă din PC-uri extensibile și neextensibile. În configurația de bază, PC-urile extensibile vin cu 286/12.5, 286/16, 386SX/16, 386SX/20, 386SX/25 sau 486SX/20. Placa de bază este prevăzută cu un slot dedicat de 104 contacte în care se pot introduce module procesor cu 386SX/16, 386SX/20, 386SX/25, 486SX/20 și 486SX/25. Evident că pentru o anumită configurație de bază au sens numai extensiile mai puternice.

Firma DARIAN ROM SUISSE SRL va comercializa în viitor calculatoarele Wearnes.

*Mitică Caraiani,
DARIAN ROM SUISSE SRL,
tel. 95/123611, fax. 95/124567*

Sisteme multimedia.

Metode de acces la text.

Metoda semnăturii.

În urma acordului de colaborare științifică și culturală semnat în anul 1990 de Germania și România, fundația *Deutscher Akademischer Austauschdienst (DAAD)* cu sediul la Bonn, a acordat României burse de specializare în Germania în anii 1990-1992. Pentru a fi acceptată candidatura, era necesară o invitație din partea unui institut sau a unei universități cu indicarea temei și a utilității la întoarcerea în țară. În urma unei corespondențe cu diferite universități vest-germane, am primit din partea *Prof. Dr. H.J. Appelrath* invitația de a participa timp de două luni la proiectele colectivului Sisteme informatice al *Facultății de Informatică a Universității Oldenburg*.

Tema de cercetare la care am participat poartă numele de **Multimedia Archiving and Retrieval System (MARS)** și este parte a proiectului **A Multimedia Filing System (MULTOS)**, proiect **ESPRIT Nr. 28**. În ciuda timpului scurt, programul de instruire la care am participat a fost deosebit de consistent.

Universitatea Oldenburg este o universitate de capacitate medie având facultăți cu specificul Științele Naturii: pictură, muzică, pedagogie, sociologie, filologie (literatură germană și engleză), istorie, geografie, biologie, informatică, religie etc.

Dotările pentru studenți sînt deosebite: piscină, terenuri de sport acoperite, cantină, bufet, librărie și o bibliotecă extraordinară desfășurată pe mai multe etaje cu posibilitatea computerizată de consultare, camere de proiecție și colecții de reviste și rapoarte la consfătuiri științifice.

În cadrul Facultății de Informatică sînt mai multe colective de cercetare în domeniile informatică teoretică și aplicativă.

Proiectul **MARS** urmărește construirea de modele pentru diferite structuri semantice ale documentelor multime-

dia, realizînd administrarea automată a datelor.

Conținutul unui document multimedia depinde de aplicație din mai multe puncte de vedere:

- referitor la prezentarea sa în pagină: text, imagine etc. (de exemplu standarde Post Script sau ODA);
- referitor la structura sintactică: forma paragrafelor, capitolelor, titlurilor etc. (de exemplu standardele de structuri logice SGML sau ODA);
- referitor la interpretarea semnificativă a sa.

Documentele multimedia în aplicațiile moderne de birotică necesită o modalitate flexibilă care să permită accesul rapid la informație, abstractizarea conținutului într-o formă cât mai apropiată de limbajul natural. O scrisoare spre exemplu are de obicei o prezentare standard în pagină: dimensiunea paginii, tipul font-urilor, o structură sintactică standard: antet, persoana căreia îi este adresată, formula de adresare, conținut, semnătură și o structură semantică standard în cazul în care este o ofertă, un refuz, o felicitare etc.

Proiectul **MARS** urmărește tipizarea reprezentării documentelor pe diferite clase (documente de afaceri, oferte, scrisori) și în cadrul fiecărei clase pe diferite scheme, toate aceste structuri aflîndu-se într-un catalog din care pot fi extrase modelele dorite. Pentru crearea unui nou document (o scrisoare de răspuns de exemplu) este ales un model corespunzător completat cu "valori" extrase din alte documente (pentru exemplul de mai sus: scrisoarea cerere, baza de date cu informațiile necesare etc.). Bineînțeles acest program este în strînsă legătură cu sisteme de baze de date pe care le interoghează pentru a extrage informații și pe care le completează cu noi date. Modul de regăsire este clasic, cu ajutorul unui limbaj de interogare în care se adresează cererea. Pentru regăsirea rapidă a datelor este posibilă indicarea valorilor atașate con-

ceptelor. Prin gruparea și generalizarea structurilor se poate organiza o structură ierarhică ce permite regăsirea eficientă.

Problema ce se pune este cea a optimizării lucrului: timp de răspuns scurt și în special posibilități de arhivare. În multe situații nu este suficientă o bază de date la care regăsirea să se facă după dicționar sau cuvînte cheie. În cazul documentelor multimedia ce sînt integral arhivate problema regăsirilor se complică datorită dimensiunii lor mari și a faptului că trebuie construite strategii diferite pentru text, imagine, sunet

Programul **MARS** se găsește la Universitatea Oldenburg în faza de lucru. Programarea se face în **MODULA2** și **C++** pe stații de lucru **SUN** sub sistemul de operare **UNIX**. Pentru eficiența implementării sub **MODULA2**, în cadrul colectivului condus de prof. dr. H.J. Appelrath, a fost realizată o bibliotecă de module pentru **MODULA2** numită **OM2Lib** (Oldenburg Modula2 Library), portabilă pe diferite echipamente. Biblioteca **OM2Lib** conține modulul matematic, module pentru operații de I/O, lucrul cu terminalele standard sau ne-standard, erori, lucrul cu memoria (alocare, dealocare), cu șiruri, liste, mulțimi, fișiere, etc. Pentru că este de curînd dată în exploatare (9.10.1991) și deci încă în faza de testare, am ajutat la testare descoperind cu ocazia implementării erori în modulele **OM2Lib**.

Prof. dr. H.J. Appelrath împreună cu responsabilul proiectului **MARS**, dr. informatician H. Eirund, mi-au propus spre rezolvare problema regăsirilor de text prin metoda semnăturii (Signature Methode).

În ultima decadă au apărut numeroase situații (biblioteci computerizate foarte vaste, sisteme informatice de birou conținînd documente multimedia) care pun serioase probleme regăsirii manuale a textului.

Fox Pro 2.0

DARIAN ROM SUISSE SRL

3400 CLUJ-NAPOCA

str. Observatorului nr. 146/15

tel. : 95-123611

fax: 95-124567

Pe o bază de date de 200MB, avînd milioane de articole, FOXPRO pe PC este mai rapid de 4 ori decît DB2/IBM pe un mainframe, iar diferența față de alte sisteme de gestiune ca Paradox sau DBase este enormă. Toate acestea datorită noii tehnologii de acces la date RUSHMORE.

Utilitățile sale de generația a patra, orientate pe obiecte, (generator de ecrane, de rapoarte, de meniuri și proiecte) cresc extraordinar viteza de dezvoltare a aplicațiilor. Și toate acestea generează programe sursă FOX, accesibile utilizatorilor !

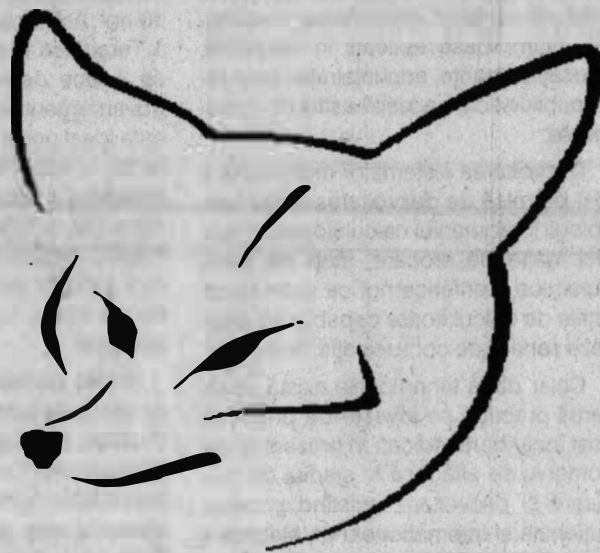
Ce este nou în FOXPRO 2.0 ?

Noul sistem de gestiune a memoriei asigură o exploatare optimă, astfel încît viteza crește atît la sisteme mono- cît și multiuser. Volumul indexării compuse reprezintă un sfert din cel al sistemelor similare și totuși este mult mai rapid.

RQBE / Relational Query-By-Example / asigură o interfață simplă cu ajutorul căreia, doar cu cîteva mișcări de mouse, obineți informațiile de care aveți nevoie. FOXPRO va genera instrucțiunea SELECT/SQL corespunzătoare, care împreună cu celelalte comenzi SQL: INSERT și CREATE TABLE pot fi folosite în aplicațiile FOX.

FOXPRO 2.0 a fost dotat cu peste o sută de comenzi și funcții noi, ca de exemplu relația "one-to-many-to-many", noi metode de sortare, sau comanda BROWSE FOR care ar putea să fie ea singură baza unei noi aplicații.

Dezvoltarea proiectelor se realizează simplu și eficient cu noul "project-manager", care urmărește fișierele folosite, înglobează în sistem toate modificările sau reconstruiește întreaga aplicație. Generarea programelor EXE și distribuirea produselor realizate este posibilă prin modulul opțional Distribution Kit.



Este posibil importul datelor dintr-o multitudine de formate : PARADOX, MULTIPLAN, LOTUS, SYMPHONY, EXCEL etc.

Prin produsele opționale ale familiei FOX : Fox Graph, Distribution Kit sau Application Program Interface, Fox Pro 2.0 devine un instrument eficient și complet pentru dezvoltarea aplicațiilor.

Trimiteți-mi dischetele demonstrative FOXPRO 2.0

NUME

FUNCȚIA

FIRMA

ADRESA

TELEFON

FAX

Format dischetă 3,5" |_| 5,25" |_|

Dischetele demonstrative pot fi obținute gratuit numai în schimbul acestui talon!

Înainte de a descrie lucrarea, socotesc că este foarte util să prezint câteva cuvinte despre suportul teoretic al acesteia.

Sistemele multimedia se definesc ca sisteme capabile să combine și să alăture o mare varietate de tipuri de informație: text, grafică, voce, imagine, audio și video iar documentele ce rezultă se numesc documente multimedia. Numeroase aplicații în medicină, educație, finanțe, administrație, timp real, publicistică, necesită astfel de documente.

Dezvoltarea sistemelor multimedia a fost permisă de dezvoltarea noilor tehnologii în domeniul calculatoarelor: tehnici înalte de stocare, stații de lucru puternice, periferice noi ca scanner-ul, rețele de calculatoare capabile să realizeze servicii de comunicație de calitate.

Chiar dacă tehnologiile există, realizarea practică se lovește de prețul de cost încă foarte ridicat. În prezent acest domeniu se află încă în stadiul de cercetare și dezvoltare, existând proiecte naționale și internaționale de elaborare a produselor software și a unor standardelor internaționale pentru tehnologiile și comunicații.

O primă problemă care s-a pus a fost cea legată de existența unei tehnologii de stocare capabilă să rețină un volum mare de date. Începând cu anul 1980, discurile optice au reprezentat o soluție a acestei probleme, ținând cont de faptul că pe un disc optic cu diametrul de 12" pot fi stocați câțiva gigabytes.

Se realizează discuri optice în trei variante: discuri optice read only (Compact Disc Read Only Memory CD-ROM), discuri optice Write Once, Read Many Times (WORM) și discuri optice mixte write/read.

CD-ROM reprezintă o memorie externă de mare capacitate și preț scăzut (10 \$ la nivelul anului 1987). Timpul de acces este de 200 msec putând stoca de la unu pînă la câțiva gigabytes. Este folosit pentru stocarea de baze de date care nu necesită actualizări. Pentru prima dată a fost utilizat în publicistică. Mecanismul optic de citire este la distanță de disc, care este îmbrăcat într-o folie de plastic sau sticlă, astfel încît informația este protejată la deteriorare fizică. ISO (International Standards Or-

ganisation) a adoptat un standard pentru această tehnologie.

WORM reprezintă un mediu sigur pentru arhivare. Permite scrierea doar o singură dată avînd acces cvasi direct și are un timp de întrebuițare foarte lung. Nu permite actualizări, ștergeri, deci discul nu poate fi reutilizat. Multe firme garantează acest disc pentru mai mult de 10 ani. Are o capacitate de mai mult de 1 Terabytes (1 Million de Mbytes) ceea ce îl face deosebit de util în aplicații guvernamentale sau de afaceri. WROM este ideal pentru memorarea informațiilor ce nu trebuie actualizate. Viteza de accesare a informației, nu este foarte mare (de la 10 la 13 biți în mai puțin de 5 sec.), dezavantajul unor tipuri de discuri WROM fiind legat de faptul că ocupă spațiu (conțin 12 discuri) și sînt scumpe.

Există diverse tehnologii de scriere pe discurile optice. Una dintre acestea (bubble forming process) realizează scrierea prin încălzirea cu raza laser a suprafeței. Aceasta formează în substraturile sale gaz care are suficientă forță să o deformeze. Raza de citire detectează deformările suprafeței. O altă tehnică topește suprafața de telurium cu raza laser, citirea făcîndu-se prin detectarea schimbării reflexivității suprafeței.

Discurile optice reutilizabile (rewritable) sînt cele mai recent apărute. În mod obișnuit au o capacitate de 650 Mbytes, unele tipuri ajungînd pînă la 30-50 Gbytes. Au o dimensiune de 5,25 inci. Tehnologia de realizare este fie prin impurificarea teluriumului astfel încît ștergerea se realizează prin crearea de curenți în suprafața de citire-scriere ce o nivelează, fie prin tehnici combinate optice și magnetice. ISO și ANSI (American National Standard Information) lucrează la standarduri pentru WROM.

Un prim mare avantaj al discurilor optice este capacitatea mare de stocare. Al doilea avantaj este legat de faptul că se pot deteriora mai greu decît cele magnetice. Domeniul de cercetare în această privință este legat de realizarea de discuri optice reutilizabile care să permită rescrierea directă fără ștergere prealabilă. Apariția tehnologiilor pentru discuri optice a permis dezvoltarea sistemelor informatice multimedia.

Pentru o mare varietate de aplicații, utilizarea principală a calculatorului este în domeniul stocării și manipulării unui mare volum de informații. Calculatoarele sînt dotate cu sisteme ce permit utilizatorilor să stocheze, regăsească și să șteargă informația. În mod obișnuit sistemele de operare sub care se lucrează permit stocarea datelor într-o structură ierarhică formată din cataloage (directory) și fișiere. Într-un fișier se poate afla informația reprezentată de diferite medii (text, imagine, sunet). Clasic, regăsirea se face la nivel de fișier prin parcurgerea liniară a cataloagelor. În ultimii ani, noile tehnologii de stocare și stațiile de lucru puternice au permis extinderea acestei regășiri liniare. În anii 1960 a apărut noțiunea de **hypertext** definită de *Ted Nelson* ca "o combinație între text în limbaj natural și capacitățile calculatorului pentru lucrul interactiv, display dinamic a unui text neliniar care nu poate fi tipărit convențional pe o pagină convențională".

Hypertext poate fi definit (*Jef Conklin*) ca o reprezentare schematică formată dintr-o rețea de obiecte dintr-o bază de date și o metodă de acces la baza de date. Pe de altă parte, metodele de acces presupun un sistem de ferestre și imagini pe ecran asociate obiectelor din baza de date.

În momentul în care au apărut calculatoare capabile de a stoca și alte tipuri de informație, problema regășirilor a fost pusă pentru diferite medii, apărînd și dezvoltîndu-se sistemele hypermedia. Calculatoarele bazate pe sisteme hypermedia sînt definite ca o extensie a hypertextului ce încorporează alte medii.

Sistemele hypermedia permit utilizatorilor crearea de informație ce să includă text, voce, imagine, grafică, video, sunet.

În momentul actual există mai multe centre în lume se utilizează astfel de sisteme: administrație, în biblioteci, sisteme de instruire etc.

Tehnologiile pentru sistemele de gestiune a bazelor de date curente permit modelarea datelor și transferul către zona CAD-CAM sau aplicații medicale. Aceste tipuri de aplicații necesită lucrul cu mai multe tipuri de date. De aceea au fost proiectate sisteme de gestiune a

bazelor de date multimedia (**MDBMS - Multimedia Data Basis Management Systems**). MDBMS permite nu numai stocarea de date de diverse tipuri dar și operații sofisticate de regăsire.

În proiectarea MDBMS s-a pornit de la baze de date relaționale care au fost reorganizate pentru a permite stocarea diferitelor tipuri de date. Altă modalitate de proiectare este construirea de MDBMS pornind de la baze de date orientate pe obiect (**Object Oriented Databases**). Un model unic nu a fost încă creat, acest domeniu aflându-se în faza de cercetare.

Privind implementările actuale, se observă că problema ce se ridică este distribuția datelor multimedia.

Indiferent de tipul datelor, caracteristic MDBMS sînt dimensiunile foarte mari ale datelor. Pentru regăsiri se folosesc strategii adecvate mediului, diferite pentru text, imagine sau sunet, toate urmărind însă o regăsire rapidă și eficientă. Cerința formulată pentru regăsire conține valori ale diverselor atribute (cuvinte cheie, informații despre pacient, numele cîmpurilor etc.) și combină operatori Booleani AND, OR, NOT.

Multe dintre metodele de acces la informație au fost deja propuse în literatura sistemelor de gestiune a bazelor de date. În general, **metodele de regăsire a textului** se clasifică în patru categorii: parcurgerea (scanning) întregului text, inversiunea, metode bazate pe multiatribut și metode bazate pe clustering.

Metodele de parcurgere integrală a textului sînt ușor de implementat dar bineînțeles, sînt cele mai lente. Problema care se pune este regăsirea unui șir de caractere (string).

Dacă întrebarea este o expresie Booleană complicată, trebuie introdus un nou pas în algoritmul de rezolvare, pas care să determine care dintre termenii găsiți satisfac expresia Booleană. Avantajele metodelor de parcurgere completă a textului este că nu necesită spațiu de memorie și nu pun probleme în cazul inserărilor sau actualizărilor.

Datorită timpului de răspuns mare, aceste metode nu sînt recomandate în cazul bazelor de date mari.

Inversiunea termenilor presupune reprezentarea fiecărui document ca o listă de cuvinte (chel) care descriu

conținutul documentului. Aceste chei sînt stocate într-un fișier atașat documentului numit fișier invers, cheile fiind stocate alfabetic. Dezavantajele sînt spațiul mare de stocare (50-300% din fișierul original) și problemele legate de actualizări sau reorganizări. Avantajele sînt implementarea facilă și rapiditatea regăsirilor. Din aceste motive, multe sisteme de gestiune a bazelor de date au adoptat această metodă (DIALOG, MEDLARS, ORBIT, STAIRS, CDSISIS).

Metodele de regăsire cu multiatribut se bazează pe metoda semnăturii. Să presupunem că avem o înregistrare formată din trei atribute (chei): "parcurerea", "integrală", "textului". Fiecare atribut este codificat cu ajutorul unei funcții într-un șir de F bits, dintre care m bits sînt 1, rezultînd "semnătura" cuvintelor.

Aceste semnături sînt stocate într-un fișier separat numit "fișierul semnătură". Nu există nici o restricție în legătură cu dimensiunea documentului ce trebuie arhivat. Dezavantajul acestor metode este timpul de răspuns lent în cazul fișierelor mari, de aceea trebuie alese variante optime pentru implementare. Avantajul este eficiența în cazul actualizărilor, abilitatea de a formula întrebări doar cu părți din cuvînt (trunchieri ale atributelor), abilitatea de a suporta creșterea dimensiunii fișierelor, precizie la regăsire.

Metodele bazate pe clustering au la bază ideea grupării documentelor similare pentru a forma clusters (ciorchini). Ipoteza de la care se pornește este aceea că documentele ce au un conținut sau o structură similară răspund în general aceluiași cereri. Gruparea poate fi făcută și termenilor din cadrul unuia document. Fiecare document este reprezentat ca un vector tridimensional și anumite cuvinte cheie îi sînt asociate cu ajutorul unei funcții de Indexare. Numărul cuvintelor cheie dă dimensiunea vectorului.

Accesul la document se face prin indirectarea într-un spațiu t-dimensional. Următorul pas este gruparea vectorilor în clusters, indirectarea făcîndu-se în aceste grupe. Regăsirile sînt de obicei accelerate prin gruparea în clusters. Dezavantajul este legat de faptul că nu există un algoritm matematic pentru determinarea numărului de clusters (de-

terminarea este făcută după o metodă empirică).

Recent au fost adoptate **soluții hardware** pentru regăsirea textului. Acestea sînt mult mai atractive, în special în condițiile descreșterii prețului de cost la echipamentele hardware. În momentul de față, nu există pentru aceste metode performanțe legate de timpul de răspuns.

Tema de cercetare propusă, **Regăsirea textului prin metoda semnăturii**, urmărește rezolvarea problemei regăsirilor de text prin metoda cu multiatribut a semnăturii și a fost dezvoltată în MODULA2 pe sistem UNIX, utilizînd modulele bibliotecii MODULA2, OM2Lib. Programul a fost realizat ca un tip de date abstracte în trei module: Signature, Basis și Hash.

Modulul Signature este de fapt tipul de date abstracte pentru acces și extracție de cuvinte dintr-un fișier semnătură, celelalte două fiind module ajutătoare de lucru: pentru constante și calcule.

Utilizînd modulul Signature, este posibilă crearea de fișiere semnătură cu un număr nelimitat de semnături, inserarea documentelor în fișiere semnătură (arhive), actualizări și bineînțeles regăsiri de text.

S-a pornit de la ideea de a avea arhivat întregul text semnificativ al unui document, excluzînd de la arhivare prepoziții, pronume, articole etc.

Există mai multe metode de obținere a semnăturii cunoscute în literatura de specialitate prin inițialele cuvintelor ce le descriu:

- WS - Word signature,
- SC - superimposed coding,
- BC - bit block compression,
- RL - run length encoding,
- EN - entropy-based bounds,
- VBC - variable bit-block - compression.

Descrierea acestor metode necesită o disertație teoretică consistentă, chiar o lucrare separată cu acest subiect. Fără a arăta cum s-a ajuns la această concluzie, trebuie spus că cea mai eficientă metodă din punctul de vedere al timpului de răspuns este SC, iar cea mai economică din punctul de vedere al spațiului utilizat RL. BC și VBC realizează un compromis între SC și RL.

Din aceste metode, am ales pentru implementare metoda SC.

Proiectarea și programarea tipului de date abstracte Signature a ridicat mai multe probleme interesante: alegerea algoritmului, programarea cât mai independentă de mașină, soluții optime de implementare.

La alegerea algoritmului am fost ajutată de existența bibliotecii universității unde am găsit material documentar foarte interesant. Pentru obținerea semnăturii am aplicat o metodă după modelul propus de *Christodoulakis și Faloutsos* (1984) bazat pe ideea suprapunerii codului a lui *Mooers* (1949). Problemei timpului de acces și a reducerii spațiului de arhivare i-am găsit ca soluție utilizarea unui fișier listă cu numele documentelor, atașat fișierului semnătură. În cazul inserării sau al eliminării trebui întâi verificat dacă respectivul document se găsește sau nu arhivat în fișierul semnătură și, pentru că parcurgerea acestuia din urmă necesită prea mult timp, verificarea se face în fișierul listă. De asemenea, fiecărei semnături îi este atașat un cod ce reprezintă documentul a căreia îi aparține și legătura cod - nume document se face în fișierul listă. Altfel, atașarea la semnătură chiar a numelui fișierului ar însemna creșterea substanțială a dimensiunilor arhivei.

Asupra fiecărui cuvânt al documentului, cu excepția cuvintelor excluse de la arhivare, se aplică o hash-funcție ce dă semnătura cuvântului. Metoda SC presupune obținerea de semnături nu pe cuvânt ci pe blocuri de text prin suprapunerea cu o funcție Booleană OR a semnăturilor fiecărui cuvânt, obținându-se pentru arhivare o semnătură pentru întregul bloc. În felul acesta se rezolvă atât problema spațiului cât și a timpului de acces. Lucrarea urmează să fie continuată pentru metoda BC.

Procedurile conținute în tipul de date abstracte Signature sînt:

Init - crează un fișier cu semnături conținând în prima înregistrare metoda de lucru (SC sau BC), numărul F de bits din tiparul semnăturii, numărul m de 1 bits și numărul D de cuvinte a căror semnătură se suprapune pentru a forma semnătura blocului. Prin procedura Init este creat și un fișier cu numele

nume-fișier-signature. Ist care conține numele documentelor arhivate în fișierul semnătură, legătura între cele două făcîndu-se printr-un cîmp de cod atașat fiecărei semnături a documentului cât și numelui din listă.

Insert - inserează un document în fișierul semnătură și numele lui în fișierul listă.

Remove - elimină din fișierul semnătură un fișier document.

open - deschide atât fișierul semnătură cât și fișierul listă de documente.

Close - închide fișierele deschise.

Get Candidate Simple - este procedura care la formularea unei cerințe (un cuvînt) dă o listă cu documentele conținînd această cerință.

Write Result Proc - este o procedură utilizată de Get Candidate Simple pentru afișarea listei de fișiere document.

Rezolvînd această temă, am avut de cîștigat foarte mult atât în domeniul teoretic cât și în cel al proiectării și programării.

Ceea ce este prezentat în acest raport este doar un rezumat al activității desfășurate la Oldenburg, cât și o mică parte a suportului teoretic care stă la baza metodelor moderne de regăsire a textului.

Timpul fiind foarte scurt (doar 2 luni dintre care primele două săptămîni au însemnat adaptarea la lucrul cu MODULA2 și sub sistem UNIX), ceea ce am realizat este doar o parte a temei de cercetare. Aceasta se poate dezvolta cu adăugarea unei proceduri complete pentru cerințe cu operatori Booleani AND, OR și NOT, a unei proceduri de eliminare a pașilor falși și completarea procedurilor existente pentru alte metode (BC și RL).

Data fiind lipsa de documentație în acest domeniu, prezentarea detaliată a metodelor de acces pentru text ar fi de un mare interes.

De asemenea, seminarii sau informări cu tema "Documente multimedia" ar fi utile țînînd seama că viitorul sistemelor informatice de birou, al bibliotecilor și al instituțiilor de informare-documentare este legat de documentele multimedia, fiind trecută în

prezent atât bariera tehnologică (există acum discuri optice de capacitate mare și timp de acces rapid, sisteme de comunicație etc.) cât și cea a lipsei aplicațiilor.

11.01.1992

Dipl. Ing. Mirela Guțică

Bibliografie:

- » 1. Archivierung multimediales Dokumente H. J. Appellrath, H. Eirund - 1990
- » 2. Acces Methodes for Text - Christos Faloutsos - 1985
- » 3. Multimedia Services and Applications Luis Orozco Barbosa, Nicolas D. Georganas - Telecommunications Research Institute of Ontario Department of Electrical Engineering.
- » 4. Design of Office Informations Systems C. A. Ellis, N. Naffah - Springer-Verlag Berlin Heidelberg New York - 1987
- » 5. Signature - Abstract Data Type for Archiving and Acces of the Text - Mirela Guțică - Oldenburg 1991
- » 6. Bericht über das Thema "Textsignaturen" - Mirela Guțică - Oldenburg 1991

* Gratis * Gratis * Gratis *

SokySoft vă oferă gratuit pachetul de rutine

DERIVATA 1.0

pentru derivarea simbolică a funcțiilor. Cei care doresc să intre în posesia acestor rutine nu trebuie să achite decât

contravaloarea unei dischete și a cheltuielilor de expediție, sau să trimită o dischetă proprie, goală, pe adresa:

Socaciu Ioan Tiberiu,
str. Arieșului, nr. 102,
bl. B2, sc. II, ap. 55,
3400 Cluj - Napoca
tel. 95/140047

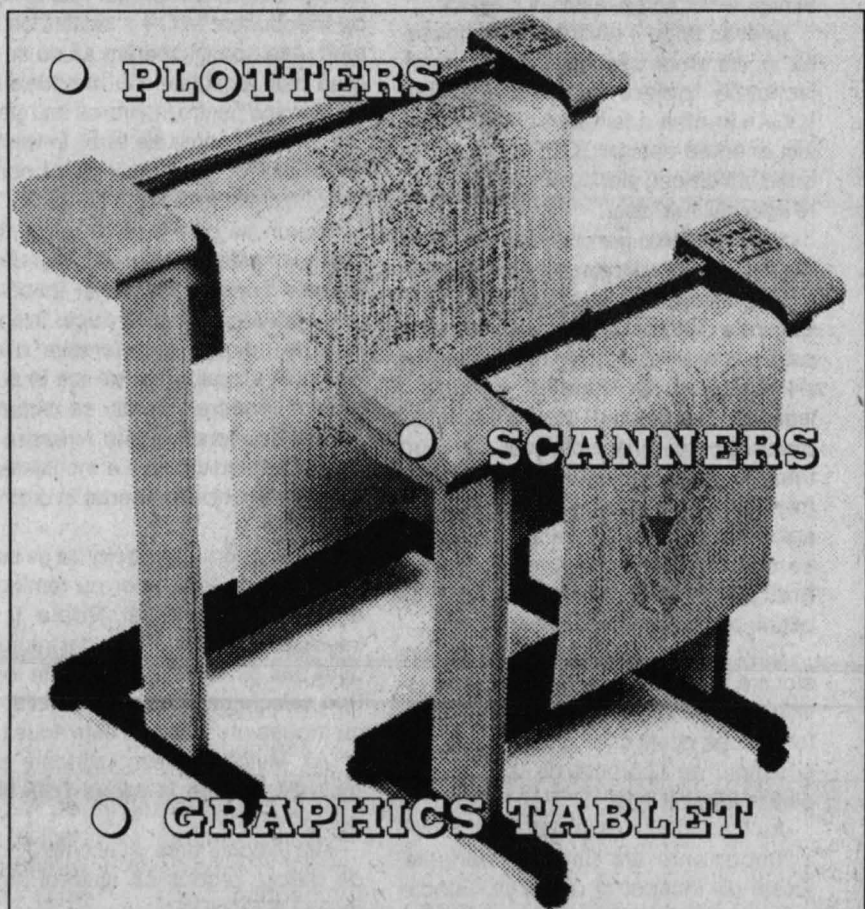
* Gratis * Gratis * Gratis *

scop

**DISTRIBUITORUL AUTORIZAT
ÎN ROMÂNIA
DE PRODUSE ALE FIRMEI**

S Summagraphics / HOUSTON INSTRUMENT

*Noi
sîntem
soluția*



TELEFON: 11.74.21 / 11.92.48. FAX: 11.73.74.

Stăpînind ierarhii

În precedentele 3 părți ați putut afla cele mai importante lucruri relative la C++. Vom consolida acum aceste cunoștințe, pentru a le putea utiliza cu succes în practică.

Desigur, după ce ați trecut prin primele 3 părți ale acestui curs, nu mai este greu să definiți clase proprii, împreună cu metodele adecvate. Funcție de "antrenament", va fi posibilă curînd și manevrarea lejeră a obiectelor dinamice. Dar, cu acestea, de-abia se stăpînesc unelte "normale". Ca și în viața reală, importantă este acordarea reciprocă a unor clase diferite, cu cele mai diverse funcții element și interacțiuni. Iar acest acord ar trebui obținut fără constrîngerii, fără forțări asupra claselor deja existente. De aceea relațiile între diferitele clase derivate trebuie analizate deja la formarea ierarhiei claselor.

Acestei teme îi revine un loc deosebit în literatura informatică mai nouă. Noțiunile "object oriented analysis" (OOA=analiză orientată obiect) și "object oriented design" (OOD=proiectare orientată obiect) sînt cuvintele cheie care apar cel mai des.

Vă prezentăm pentru început un mic exemplu, care utilizează metodele descrise în literatură. Listingul LISTA1.CPP cuprinde cele mai importante elemente pentru liste dublu înlănțuite. Într-o astfel de listă, fiecare element conține o legătură - un pointer - înspre elementul precedent și una spre cel care urmează. În afară de aceasta, fiecare element este prevăzut cu un atribut, care stabilește o legătură între elementele ce se conformează unui anumit criteriu. Prin intermediul acestui atribut se obține o sortare a elementelor listei.

Cînd datele unei liste înlănțuite sînt stocate într-o colecție de date, este relativ ușor posibilă tipărirea datelor existente după acest criteriu de sortare. Pur și simplu se tipărește de la începutul pînă la sfîrșitul listei.

Au fost numite aici două caracteristici importante ale listelor înlănțuite. Există un început și un sfîrșit. Funcție de care capăt al listei este vorba, lip-

sește legătura spre predecesor sau succesor. Lipsa acestor legături este marcată de pointerul vid (NULL). Astfel de pointeri marchează faptul că legătura nu este "reală", că datele la care se referă nu există de fapt. La adresa "NULL" nu se pot găsi date, căci prima adresă începe la "1".

Obiectele »A1« pînă la »A5« sînt create dinamic. Deoarece în momentul creerii încă nu există succesori, la inițializarea obiectului s-a renunțat la conținutul corect al legăturilor spre predecesor și succesori. În acest exemplu, nu acesta este obiectivul principal. Menționăm însă aici că metode precum adăugarea, ștergerea și respectiv inserarea unui element în listă sînt indispensabile pentru o administrare confortabilă a listelor înlănțuite. Pentru metode care merg puțin mai departe, facem trimitere în primul rînd la cartea de introducere în C++ editată de Borland. Alte apariții sperăm să nu se lase prea mult așteptate. Cu metodele descrise acolo pentru inserarea sau ștergerea unor elemente ale listei, în listing ar fi posibilă dotarea cu legături corecte ale obiectelor încă de la generare.

"Luxul" pe care îl oferă listele dublu înlănțuite este că ele pot fi parcurse în ambele sensuri, înainte și înapoi. Să presupunem că o listă dublu înlănțuită conține pointeri spre ferestre diferite. Atunci la vizualizarea "de jos în sus" a tuturor ferestrelor care se suprapun (parțial sau total) fiecare fereastră ar fi "deschisă" individual. La închiderea ferestrelor ar trebui procedat în ordine inversă.

Dar, în cadrul ferestrelor ce se suprapun, ordinea ferestrelor nu rămîne totdeauna cea inițială. Poate fi mai interesantă la un moment dat informația dintr-una din ferestrele aflate "în teanc". Prin selectarea acesteia din taste - sau cu mouse-ul - fereastră este adusă sus, în vîrf. Metoda corespunzătoare a fost realizată, în mare, în »Move-To()« (listingul Liste2.CPP).

Iată în continuare un exemplu posibil de dialog, produs cu ajutorul programului LISTE2.EXE:

```

1 2 3 4 5
Tastati un numar între 1 si 5: 3
3 1 2 4 5
Tastati un numar între 1 si 5: 4
4 3 1 2 5
Tastati un numar între 1 si 5: 2
2 4 3 1 5
Tastati un numar între 1 si 5: 5
5 2 4 3 1
Tastati un numar între 1 si 5: 5
5 2 4 3 1
Tastati un numar între 1 si 5: 1
1 5 2 4 3
Tastati un numar între 1 si 5: 0
1 5 2 4 3

```

Cifra cea mai din stînga reprezintă fereastra cea mai de sus. Succesiunea ferestrelor nu se schimbă dacă, din întîmplare, se selectează acea fereastră care este deja în vîrf. În exemplu, a fost selectată de două ori la rînd fereastra 5. Deși în metoda »Move-To()« succesiunea se schimbă, în implementarea de față nu s-a făcut nici o modificare a legăturilor spre succesori sau predecesori. Din contră, au fost interschimbate numai conținuturile ferestrelor. Acest lucru se întîmplă prin intermediul metodei »Schimb()«. Menționăm aici că prin aceasta uzul metodei se restricționează destul de mult. Explicităm puțin dificultățile realizării schimbului:

- al treilea caz se autoelimină prin simplitate, căci dacă fereastra cea mai de sus este mutată sus, nu se întîmplă absolut nimic

- în primul caz se modifică legăturile predecesorului, ale fostului element aflat sus și ale noului element aflat în vîrf

- iar al doilea caz implică modificarea a chiar patru elemente diferite.

În listingul LISTE2.CPP se tot schimbă două conținuturi învecinate între ele, pînă cînd elementul dorit se află în vîrf. Această metodă are dezavantajul că, la liste lungi, acest lucru va dura desigur foarte mult.

În listingul FEREAȘT1.CPP sînt realizate metodele fundamentale pentru reprezentarea unei ferestre simple. Aceste metode se aseamănă cu cele pe care le-am prezentat în listingurile

anterioare, relative la dreptunghiuri și pătrate. La acele listinguri ne putem imagina și funcții element care să influențeze dimensiunile și poziția ferestrei. De aceea, memoria tampon pentru o fereastră a fost dimensionată astfel încât - la operațiuni "zoom" de exemplu - să fie posibilă reprezentarea ferestrei pe întregul ecran.

Pentru fiecare fereastră, coordonatele poziției, lățimea, înălțimea precum și textul corespunzător pot fi ușor accesate. Fiecare obiect fereastră poate fi vizualizat pe ecran cu ajutorul numărului său intern.

Listingul FEREAST1.CPP nu urmărește poziția relativă a diverselor ferestre între ele. Nu este posibil aici un »refresh« (o reîmprospătare) completă a ferestrelor "de jos în sus". Pe ecran, obiecte tridimensionale sînt reprezentate într-o suprafață bidimensională. Să presupunem că o fereastră oarecare se translatează sau este redimensionată de așa manieră încît devine necesar un "refresh" al reprezentării. Atunci, din păcate, colecția de metode din listingul FEREAST1.CPP se dovedește insuficientă. De aceea, a fost realizat listingul FEREAST2.CPP.

Dar, înainte, să menționăm și o altă posibilitate de rezolvare. Proprietățile clasei »Lista« în listingul LISTE1.CPP și metodele clasei FEREAST1.CPP sînt de fapt suficiente pentru a realiza o legătură prin tehnici de programare uzuale. Succesiunea ferestrelor s-ar păstra în clasa »Lista«. Ferestrele însele

sînt conținute în clasa »Fereast«. Cu funcția »Get-Elem()« s-ar putea returna o valoare întreagă care să specifice poziția în clasa ferestrelor. Dar, în spiritul încapsulării datelor, cu noile metode este mai avantajoasă derivarea unei clase noi, din clasele »Fereast« și »Lista«. În listingul »Fereast2.CPP«, această clasă derivată se numește »Ve-

spre deosebire de listingurile LISTE1.CPP și LISTE2.CPP, aici s-a folosit un al 3-lea pointer spre elementul corespunzător. Acest pointer este mai ușor modificabil decît o cantitate mai mare de elemente-date, cum apăreau în fosta clasă »Fereastr«.

În figura 2, partea superioară provine din clasa »Liste« (3 pointeri), iar partea inferioară este analogă clasei »Fereastr«. În metoda »Move-Top()«, care are ca efect deplasarea celei de-a n-a ferestre (numărînd de sus), nu trebuie schimbate părțile inferioare ale instanțelor. Este suficient să se interschimbă pointerii spre ferestrele corespunzătoare. Este mult mai rapidă modificarea unui pointer decît mutarea unui domeniu la care face referință acest pointer. În ultimul caz, domeniul de memorie care ar trebui mutat este semnificativ mai mare. Și în sensul unei valabilități generale, este mai bine să se acționeze numai asupra pointerilor, deoarece atunci aceștia pot face referire la obiecte de dimensiuni diferite.

Metoda »Fereastr::Afis()« face uz de datele din partea inferioară fără a folosi pointerul spre elementul corespunzător. În schimb metoda »View::Afis()« folosește pointerul spre fereastră eventual modificată a unei instanțe învecinate a clasei »View«. Din acest motiv a fost necesară formularea funcțiilor element specifice clasei »Fereastr« ca funcții virtuale. În programul FEREAST2.EXE, la tastarea valorii »1« vederea spre ferestre nu se schimbă în nici un fel, deoarece fereastră cea mai de sus rămîne sus. Dacă se alege de cîteva ori la rînd cifra »2«, apar alternativ cele două ferestre aflate sus (se modifică aspectul). Acest lucru este reprezentat schematic în figura 2. La alegerea cifrei »5«, fereastră cea mai de jos (a 5-a numărînd de sus) va fi adusă în vîrf. Fosta fereastră cu numărul 4 devine noua fereastră 5 și deci se va găsi la baza "stivei" de ferestre.

Listingul FEREAST2.CPP e gîndit ca o sugestie pentru integrarea altor metode pentru ferestre suprapuse. Intenționat am renunțat la o colecție cuprinzătoare de metode deja reprezentate. Programarea orientată obiect poate fi învățată aproape în joacă, prin încercarea de multiple combinații a unor clase diferite și ale metodelor lor. Cititorului îi va face cu siguranță mai

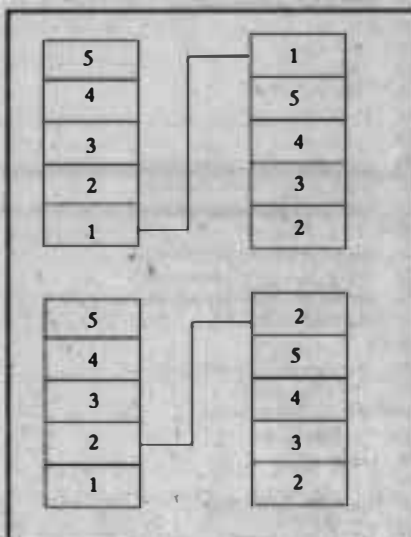
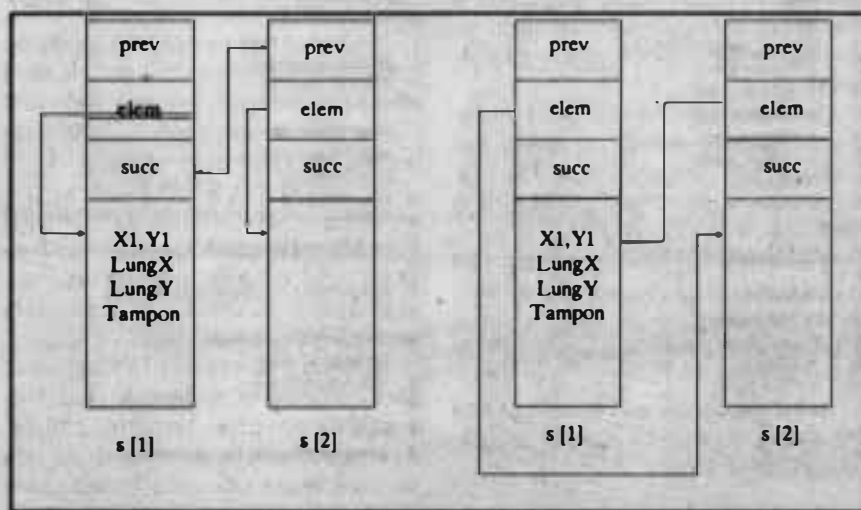


Figura 1. Două cazuri la translația ferestrelor suprapuse

dere«. În clasa »Vedere« sînt reunite metodele ambelor clase. O instanță a acestei clase are în antet 3 pointeri. Cei doi pointeri »prev« și »succ« sînt legăturile cunoscute spre predecesor și succesor în cadrul listei. Dar, în locul unei valori numerice a elementului listei,

Figura 2. Doua instanțe schematice ale clasei »Vedere«, în starea inițială și după interschimbarea celor două ferestre superioare



multă plăcere să creeze o metodă nouă, decât să tasteze pînă la epuizare, introducînd un listing fără sfîrșit. OOP-ul este și mai ușor înțeles în acest fel. În acest sens: mult succes în dezvoltarea unor metode noi!

(J.S./I.F.)

```

1 // Lista1.cpp
2 // lista inlantuita
3
4 #include <conio.h>
5 #include <iostream.h>
6
7 class Lista
8 {
9     Lista *prev;
10    int element;
11    Lista *succ;
12
13 public:
14    Lista (Lista* p, int e, Lista* s)
15    {
16        prev = p;
17        element = e;
18        succ = s;
19    };
20
21    Lista* get_prev() { return (prev); }
22    int get_elem() { return (element); }
23    Lista* get_succ() { return (succ); }
24
25    void set_prev(Lista* p) { prev = p; }
26    void set_elem (int e) { element = e; }
27    void set_succ (Lista* p) { succ = p; }
28 };
29
30 int main()
31 {
32    clrscr();
33
34    Lista* A1 = new Lista (NULL, 1, NULL);
35    Lista* A2 = new Lista (A1, 2, NULL);
36    A1->set_succ (A2);
37    Lista* A3 = new Lista (A2, 3, NULL);
38    A2->set_succ (A3);
39    Lista* A4 = new Lista (A3, 4, NULL);
40    A3->set_succ (A4);
41    Lista* A5 = new Lista (A4, 5, NULL);
42    A4->set_succ (A5);
43
44    Lista* A0 = new Lista (*A1);
45    cout << "A1: " << A0->get_elem() << "\n";
46
47    A0 = A0->get_succ();
48    cout << "A2: " << A0->get_elem() << "\n";
49
50    A0 = A0->get_succ();
51    cout << "A3: " << A0->get_elem() << "\n";
52
53    A0 = A0->get_succ();
54    cout << "A4: " << A0->get_elem() << "\n";
55
56    A0 = A0->get_succ();
57    cout << "A5: " << A0->get_elem() << "\n";
58 }

```

LISTA1.CPP

```

1 // Lista2.cpp
2 // lista inlantuita
3
4 #include <conio.h>
5 #include <iostream.h>
6
7 class Lista
8 {
9     Lista *prev;
10    int element;
11    Lista *succ;
12 public:
13    Lista (Lista* p, int e, Lista* s)
14    {
15        prev = p;
16        element = e;
17        succ = s;
18    }
19
20    Lista* get_prev() { return (prev); }
21    int get_elem() { return (element); }
22    Lista* get_succ() { return (succ); }
23
24    void set_prev (Lista* p) { prev = p; }
25    void set_elem (int e) { element = e; }
26    void set_succ (Lista* p) { succ = p; }
27    Lista* find_top (Lista* l)
28    {
29        if (l->get_prev() != NULL)
30            l = l->get_prev();
31        if (l->get_prev() != NULL)
32            l = find_top (l);
33        return (l);
34    }
35
36    Lista* find_end (Lista* l)
37    {
38        if (l->get_succ() != NULL)
39            l = l->get_succ();
40        if (l->get_succ() != NULL)
41            l = find_end (l);
42        return (l);
43    }
44
45    Lista* find_next (int e, Lista* l)
46    {
47        if (l->get_elem() != e)
48            l = l->get_succ();
49        if (l->get_elem() != e)
50            l = find_next (e, l);
51        return (l);
52    }
53
54    void Schimba (Lista* l)
55    {
56        if (l->get_prev() != NULL)
57            l = l->get_prev();
58        m = l->get_elem();
59        l = l->get_succ();
60        l->set_elem(m);
61        l = l->get_prev();
62        Schimba (l);
63    }
64
65    Lista* move_top (int e, Lista* l)
66    {
67        l = find_top (l);
68        l = find_next (e, l);
69        if (e == l->get_elem()) {
70            Schimba (l);
71            l = find_top (l);
72            l->set_elem (e);
73        }
74        return (l);

```

```

75    }
76 void Afisare (Lista* l)
77 {
78     cout << " " << l->get_elem() << " ";
79     if (l->get_succ() != NULL) {
80         l = l->get_succ();
81         Afisare (l);
82     }
83     else cout << "\n";
84 }
85 void Prezinta (Lista* l)
86 {
87     l = find_top (l);
88     Afisare (l);
89 }
90 };
91
92 main()
93 {
94     int t;
95
96     clrscr();
97     Lista* A1 = new Lista (NULL, 1, NULL);
98     Lista* A2 = new Lista (A1, 2, NULL);
99     A1->set_succ (A2);
100    Lista* A3 = new Lista (A2, 3, NULL);
101    A2->set_succ (A3);
102    Lista* A4 = new Lista (A3, 4, NULL);
103    A3->set_succ (A4);
104    Lista* A5 = new Lista (A4, 5, NULL);
105    A4->set_succ (A5);
106
107    Lista* A0 = new Lista (*A1);
108    A0->Prezinta (A0);
109
110    t = 1;
111    while (t > 0) {
112        cout << "Testați un numar între 1..5 (0 termina): ";
113        cin >> t;
114        if (t <= 5) {
115            A0 = A0->move_top (t, A0);
116            A0->Prezinta (A0);
117        }
118    }
119 }

```

LISTA2.CPP

```

1 // Fereast1.cpp
2 // ferestre dinamice
3
4 #include <conio.h>
5 #include <stdio.h>
6
7 const XMax = 80;
8 const YMax = 25;
9 const XMin = 1;
10 const YMin = 2;
11
12 enum Boolean {false, true};
13
14 class Fereastr // definiteaza clasa Fereastr
15 {
16     int x, int y; // coordonatele
17     int lungx, int lungy;
18     char tampon [4096];
19
20 public:
21     Fereastr::Fereastr (int inix, int inity, \

```

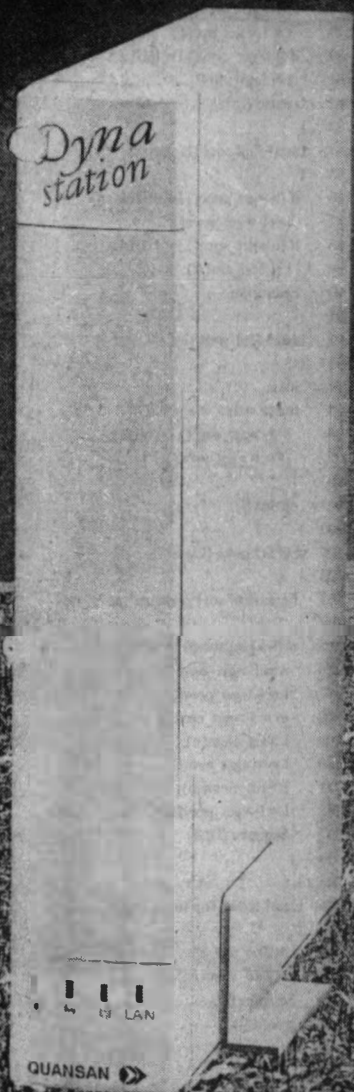
QUANSAN

*Dyna
station*

An Elegant

Book-sized Workstation

- ALL-IN-ONE 386SX-20/25
- INTERNAL 40W SWITCHING POWER SUPPLY
- SUPER VGA DISPLAY
- WEIGHING JUST 2.5KGs
- 1 x FDD & LAN CARD (OPTION : HDD)



DYNA-STATION 386DS20/25

CPU	80386SX-20/25
Viteza	20 sau 25 MHz
RAM (standard-max.)	1 MB - 8 MB, SIMM
BIOS	64 KB
Coprocessor (opțional)	80387SX
Porturi seriale/paralele	1 serial/ 1 paralel
Conector extern FDD	da (opț. al 2-lea port serial sau game port)
Port mouse PS/2	da
Controller floppy disc	1,44 Mb/ 720 kb/ 1,2 Mb/ 360 Kb
Controller hard disk	IDE
Lector de dischete 3,5"	1 x 1,44 Mb
Hard disk 3,5" (opțional)	40/ 80/ 120/ 210 Mb
Sloturi de extensie, 16 biți	1 slot (dacă nu este instalat HDD)
Versiuni VGA	256 Kb video RAM: 640x480x16, 800x600x16 culori (Trident TVGA 9000) 512 Kb video RAM: 640x480x256 (non interlaced), 800x600x256 (non-interlaced), 1024x768x16 culori
Dimensiuni	28,5 x 5 x 22 cm
Greutate	2,5 Kg
Sisteme de operare	MS-DOS 5.0 sau 4.01, OS/2, Windows 3.0, Unix, Novell Netware, OS/2 LAN Manager
LECOM S.R.L., Calea Victoriei nr. 126, București, tel. 506420, fax: 129558	

```

22         int lx, int ly)
23 {
24     x = inix; lungx = lx - x;
25     y = inity; lungy = ly - y;
26 }
27 //functii element publice
28 int Fereastră::GetX1() { return x; }
29 int Fereastră::GetY1() { return y; }
30 int Fereastră::GetX2() { return (x+lunx); }
31 int Fereastră::GetY2() { return (y+luny); }
32 int Fereastră::GetX() { return lungx; }
33 int Fereastră::GetY() { return lungy; }
34 void Fereastră::Citire()
35 {
36     gettex (GetX1(), GetY1(), GetX2(), \
37             GetY2(), &tampon);
38 }
39 void Fereastră::Afsare()
40 {
41     puttex (GetX1(), GetY1(), GetX2(), \
42            GetY2(), &tampon);
43 }
44 void Fereastră::Umplere (char valoare, char atr)
45 {
46     int i;
47     for (i = 0; i < 4096; i = i+2) {
48         tampon[i] = valoare; tampon[i+1] = atr;
49     }
50 }
51 }
52 };
53
54 int main()
55 {
56     char tasta;
57     Boolean sfirsit;
58
59     clrscr();
60     Fereastră *F1 = new Fereastră (XMin, YMin, \
61                                   XMax, YMax);
62     F1->Umplere ('1', 0x16);
63     F1->Afsare();
64
65     Fereastră *F2 = new Fereastră (5, 3, \
66                                   10, 10);
67     F2->Umplere ('2', 0x2f);
68     F2->Afsare();
69
70     Fereastră *F3 = new Fereastră (8, 5, 10, 20);
71     F3->Umplere ('3', 0x38);
72     F3->Afsare();
73
74     Fereastră *F4 = new Fereastră (16, 12, 26, 14);
75     F4->Umplere ('4', 0x40);
76     F4->Afsare();
77
78     Fereastră *F5 = new Fereastră (36, 18, 76, 22);
79     F5->Umplere ('5', 0x51);
80     F5->Afsare();
81
82     Fereastră *F6 = new Fereastră (50, 15, 62, 19);
83     F6->Umplere ('6', 0x6a);
84     F6->Afsare();
85
86     printf ("Alegeti un numar intre 1..6: ");
87     while (1) {
88         tasta = getch ();
89         sfirsit = false;
90         switch (tasta)
91         {
92             case '1': F1->Afsare();
93                     break;
94             case '2': F2->Afsare();
95                     break;
96             case '3': F3->Afsare();

```

```

97         break;
98         case '4': F4->Afsare();
99                 break;
100        case '5': F5->Afsare();
101                break;
102        case '6': F6->Afsare();
103                break;
104        default: sfirsit = true;
105                };
106        } while (!sfirsit);
107 }

```

FEREAST1.CPP

```

1 // Fereast2.cpp
2 // fereastre dinamice in liste inlantuite
3
4 #include <conio.h>
5 #include <stdio.h>
6
7 const XMax = 80;
8 const YMax = 25;
9 const XMin = 1;
10 const YMin = 2;
11
12 enum Boolean {false, true};
13
14 class Fereastră // defineste class Fereastră
15 {
16     int x; int y; // coordonatele
17     int lungx; int lungy;
18     char tampon [4096];
19
20 public:
21     Fereastră (int inix, int inity, \
22              int lx, int ly)
23     {
24         x = inix; lungx = lx - x;
25         y = inity; lungy = ly - y;
26     }
27     //functii element publice
28     int Fereastră::GetX1() { return x; }
29     int Fereastră::GetY1() { return y; }
30     int Fereastră::GetX2() { return (x+lunx); }
31     int Fereastră::GetY2() { return (y+luny); }
32     int Fereastră::GetX() { return lungx; }
33     int Fereastră::GetY() { return lungy; }
34     char* Fereastră::GetBuf() { return tampon; }
35     void Fereastră::Citire()
36     {
37         gettex (GetX1(), GetY1(), GetX2(), \
38                GetY2(), &tampon);
39     }
40     virtual void Fereastră::Afsare()
41     {
42         puttex (GetX1(), GetY1(), GetX2(), \
43                GetY2(), &tampon);
44     }
45     virtual
46     void Fereastră::Umplere (char valoare, char atr)
47     {
48         int i;
49         for (i = 0; i < 4096; i = i+2) {
50             tampon[i] = valoare; tampon[i+1] = atr;
51         }
52     }
53 }
54 };
55
56 class Lista
57 {
58     Lista *prev;
59     Fereastră *element;
60     Lista *succ;
61 public:
62     Lista (Lista *p = NULL, \
63           Fereastră *e = NULL, \
64           Lista *s = NULL.)
65     { prev = p; element = e; succ = s; }
66
67     Lista* get_prev() { return (prev); }
68     Fereastră* get_elem() { return (element); }
69     Lista* get_succ() { return (succ); }
70
71     void set_prev (Lista* p) { prev = p; }
72     void set_elem (Fereastră* e) { element = e; }
73     void set_succ (Lista* p) { succ = p; }
74     void set_list (Lista* p, Fereastră* e, Lista* s)
75     { prev = p; element = e; succ = s; }
76     Lista* find_top (Lista* l)
77     {
78         if (l->get_prev() != NULL)
79             l = l->get_prev();
80         if (l->get_prev() != NULL)
81             l = find_top (l);
82         return (l);
83     }
84     Lista* find_end (Lista* l)
85     {
86         if (l->get_succ() != NULL)
87             l = l->get_succ();
88         if (l->get_succ() != NULL)
89             l = find_end (l);
90         return (l);
91     }
92     Lista* find_next (int e, Lista* l)
93     {
94         int z;
95         for (z = 1; z < e; z++) {
96             if (l->get_succ() != NULL)
97                 l = l->get_succ();
98         }
99         return (l);
100     }
101     void Schimba (Lista* l)
102     {
103         Fereastră* m; Fereastră* n;
104
105         if (l->get_prev() != NULL) {
106             n = l->get_elem();
107             l = l->get_prev();
108             m = l->get_elem();
109             l->set_elem(n);
110             l = l->get_succ();
111             l->set_elem(m);
112             l = l->get_prev();
113             Schimba (l);
114         }
115     }
116     Lista* move_top (int e, Lista* l)
117     {
118         l = find_top (l);
119         l = find_next (e, l);
120         Schimba (l);
121         return (l);
122     }
123 };
124
125 class View : public Fereastră, public Lista
126 {
127     int vizibil;

```

(continuare în pag. 62)

Programarea Windows (I)

Ca mediu de programare pentru Windows s-ar fi putut folosi mai multe compilatoare C, dar s-a depins de SDK și WRT. Acest lucru s-a schimbat radical odată cu apariția noilor produse. Această serie de articole va arăta cum funcționează Windows și cum se poate programa: simplu sau complicat.

Pentru utilizatorii DOS, Windows 3.0 este o suprafață utilizator grafică foarte frumoasă. Pentru impunerea sa pe piață hotărâtoare a fost disponibilizarea rapidă a aplicațiilor și rolul conducător al firmei Microsoft pe piață. Împotriva acestora GEOWORKS Ensemble nu a putut concura cu toate că este o suprafață mai bună și mai rapidă.

La acestea se mai adaugă faptul că Microsoft însăși a extras unele produse din versiunea 2 și le-a împins cu forță în regimul versiunii superioare. Gigantul software și-a dat seama că Windows se va putea vinde mai bine dacă se va face multă vîlvă în jurul produsului, de aceea mai întîi se oferă cantitate, abia apoi urmîndu-se a se ajunge și la calitate.

Se pare că Borland a cîștigat prima rundă în lupta pentru limbajele de programare aferente Windows. Producătorul de software, proaspăt "căsătorit" cu Ashton Tate nu a făcut, nici mai mult nici mai puțin, decît să-și redezvolte toate produsele cunoscute de pînă acum, dar rezultatul a fost mai bun calitativ decît SDK-ul sau WRT-ul (Whitewater Resource Toolkit) lui Microsoft. Pe lîngă acest fapt produsele Borland sînt mult mai ieftine. (Concurența este mult mai înrăită în SUA!). Firma Borland oferă produse calitativ superioare la jumătatea prețului celor de la Microsoft.

Făcînd o comparație între MS-C 6.0, (între timp a apărut versiunea 7.0), și SDK cu C++ și Object Windows Library ale firmei Borland, se ajunge repede la concluzia, că programarea orientată obiect (OOP) este tehnica cea mai comodă de scriere a aplicațiilor Windows.

Pentru a putea trage asemenea concluzii este necesară cunoașterea și înțelegerea mecanismelor interne Windows.

Faceți cunoștință cu Windows

Programele "clasice" sînt interpretate secvențial: Interpretorul de comenzi poziționează contorul de comenzi al procesorului la începutul programului, iar apoi acesta prelucrează comandă după comandă. Există și un sfîrșit definit. Derularea programului este definită de structura programului. Cei care cunosc această structură o denumesc "utilă" iar cei mai conservatori o consideră singura cale spre o adevărată programare.

Programele Windows sînt puse în mișcare de informații, deci ele funcționează cu totul diferit decît programele secvențiale. Fiecare aplicație este compusă dintr-o multitudine de componente singulare, fiecare dintre acestea reprezentînd o funcție bine definită. Aici se crează informații fie printr-o funcție parțială a programului, fie printr-un eveniment din afară (clic pe mouse, apăsarea pe o tastă, starea imprimantei etc.). Acestea trebuie să provoace o anumită reacție a funcțiilor de care aparține. De aceea nu se poate determina cu exactitate modul de derulare al unui astfel de program.

Dacă, spre exemplu cursorul mouse-ului s-a poziționat undeva pe ecran și a fost apăsată o tastă a mouse-ului, un handler de eveniment determină cînd și unde s-a întîmplat, găsește funcția de care aparține și îi predă mesajul corespunzător. Funcția reacționează la informație în mod individual și transmite programului părinte un mesaj de terminare (rezolvare). Acest mecanism se repetă într-o buclă infinită.

leșirile pe ecran generate de toate programele Windows sînt de natură grafică.

Acest fapt cere un cu totul alt mod de gîndire, mai ales pentru cei care au programat pînă acum orientat caracter. leșirile grafice pot fi puncte singulare, cercuri, dreptunghiuri, linii sau poligoane. Dar și afișarea textelor se face în mod grafic. Astfel s-a ușurat folosirea comună în aplicații a textului cu grafica. Tratarea textului pierde din confort, dar avantajele obținute merită osteneala.

Pentru ca utilizatorul să se ocupe cît mai puțin de detaliile interne ale ieșirilor grafice, Windows îi pune la dispoziție un GDI (Graphic Development Interface). Cu aceasta pot fi reprezentate elemen-

tele grafice cele mai utilizate. Graficele obținute pot fi dependente sau independente de contextul ieșirii. Deci cu același apel al funcției "Rectangle" se poate afișa un dreptunghi atît pe ecran cît și la imprimantă. Din punctul de vedere al utilizatorului ieșirea pe ecran poate fi identică cu cea de la imprimantă (WYSIWYG).

Pentru a putea realiza această unitate Windows necesită unități inteligente, care pot reprezenta toate elementele susținute de GDI. Aceste unități se denumesc driver-e. Ele declanșează ieșirea grafică propriu-zisă. GDI-ul recunoaște patru unități: ecranul, aparate hardcopy, cum ar fi imprimanta și plotter-ul, bit-map-uri și metafișiere.

Ecranele și imprimantele sînt echipamente fizice, care redau imaginile în mod optic. Celelalte două: bitmap-urile și metafișierele sînt pseudounități. Ele permit memorarea imaginilor în memoria de lucru sau pe harddisk.

Oricărei aplicații de sub Windows îi este subordonată o fereastră grafică. Ea constă din două zone distincte: zona non-client, care este gestionată de obicei de sistem și poate cuprinde un che-nar activ (border resizeble), care prin "tragere" rescalează fereastra, precum și un antet pe care e afișat de obicei numele aplicației și de care dacă se "trage" se mută fereastra aplicației oriunde pe ecran. Acest antet mai poate cuprinde trei controale specifice - Sistem meniu (un meniu scurt cu funcții de interfață cu Windows), buton de maximizare (fereastra va acoperi tot ecranul) și buton de minimizare (fereastra revine la mărimea normală dacă era maximizată, sau devine iconă - o reprezentare grafică restrînsă a aplicației). Restul suprafeței ferestrei se numește zonă client și aici se manifestă cu precădere funcțiile specifice aplicației. Originea sistemului de coordonate este definit astfel încît în colțul din stînga sus să se afle domeniul client. Concomitent un mecanism de siguranță veghează, ca fereastra să nu încalce o altă fereastră și invers. Înseamnă că obiectele grafice care depășesc marginea ferestrei vor fi tăiate automat.

```

1  #include <Windows.h>
2
3  long FAR PASCAL FirstWndProc (HWND, unsigned, WORD, LONG);
4
5  int PASCAL WinMain (HANDLE hInstance,
6                    HANDLE hPrevInstance,
7                    LPSTR lpszCmdLine,
8                    int cmdShow)
9
10 HWND hwn;
11 MSG msg;
12 WNDCLASS wndclass;
13
14 if (!hPrevInstance)
15 {
16     wndclass.lpszClassName = "FIRST:MAIN";
17     wndclass.hInstance = hInstance;
18     wndclass.lpfnWndProc = FirstWndProc;
19     wndclass.hCursor = LoadCursor(hInstance, "Sageata");
20     wndclass.hIcon = LoadIcon(hInstance, "Icon_1");
21     wndclass.lpszMenuName = NULL;
22     wndclass.hbrBackground = COLOR_WINDOW + 1;
23     wndclass.style = NULL;
24     wndclass.cbClsExtra = 0;
25     wndclass.cbWndExtra = 0;
26     RegisterClass(&wndclass);
27 }
28 hwn = CreateWindow ("FIRST:MAIN",
29                   "Meniu principal",
30                   WS_OVERLAPPEDWINDOW,
31                   CW_USEDEFAULT,
32                   0,
33                   CW_USEDEFAULT,
34                   NULL,
35                   NULL,
36                   hInstance,
37                   NULL);
38 ShowWindow (hwn, cmdShow);
39 while (GetMessage (&msg, 0, 0, 0))
40 {
41     TranslateMessage (&msg);
42     DispatchMessage (&msg);
43 }
44 return 0;
45
46 long FAR PASCAL FirstWndProc (HWND hwn,
47                             unsigned msg,
48                             WORD wParam,
49                             LONG lParam)
50 {
51     switch (msg)
52     {
53     case WM_DESTROY:
54         PostQuitMessage(0);
55         break;
56     default:
57         return (DefWindowProc(hwn, msg, wParam, lParam));
58     }
59 }
60 return 0L;
61 }
62

```

Fişierul FIRST.C - Primul program C pentru Windows

```

NAME FIRST
EXETYPE WINDOWS
DESCRIPTION 'Primul Inceput'
CODE MOVEABLE DISCARDABLE
DATA MOVEABLE MULTIPLE
HEAPSIZE 512
STACKSIZE 5000
EXPORTS
    FirstWndProc

```

Fişierul FIRST.DEF - Fişierul de definire a modulelor

```

first.exe: first.obj first.def first.res
link first,first/allgn:16,first/map,mllbcw llbw;/NOD/NOE/CO, min.def
rc min.res

```

```

first.res: first.rc first.crs first.ico
rc -r first.rc

```

```

first.obj: first.c
cl -AM -c -Gaw -Od -W2 -Zpl first.c

```

Fişierul FIRST.MAK - Fişierul MAKE

```

icon_1 icon first.ico
Sageata cursor first.crs

```

Fişierul FIRST.RC - Fişierul de resurse.

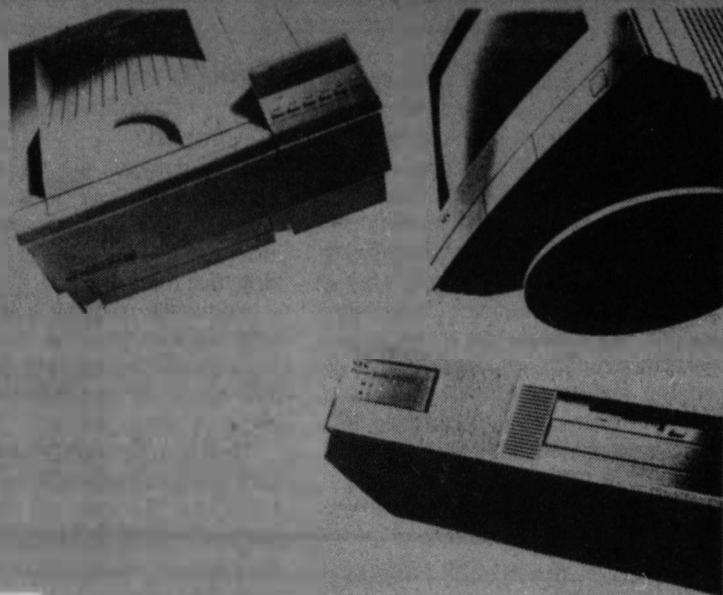
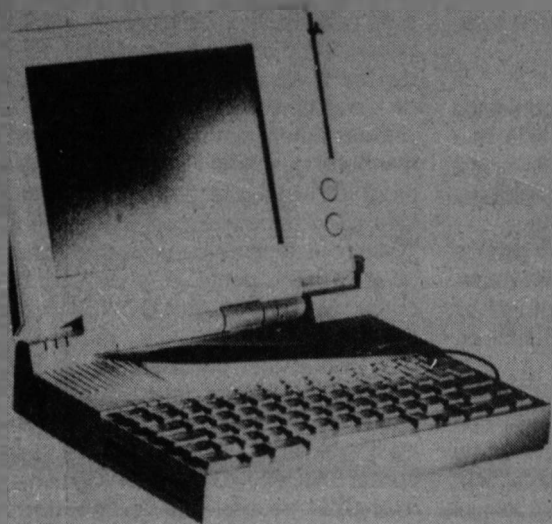
Listing 1.1. Scheletul de bază al unui program Windows scris în limbajul MS-C.

```

1 Program First;
2 {SR a3.res}
3 Uses WObjects;
4
5 type
6
7 MyApp = Object (TApplication)
8     Procedure InitMainWindow;Virtual;
9 End;
10
11 Procedure MyApp.InitMainWindow;
12 Begin
13     MainWindow := New(PMenuWindow, Init);
14 End;
15
16 Var MyA : MyApp;
17
18 Begin
19     MyA.Init('First');
20     MyA.Run;
21     MyA.Done;
22 End.

```

Listing 1.2. Orientat obiect cu Turbo-Pascal for Windows.



NEC - UltraLite SL/20P

Procesor: Intel 386SL-20, 20 MHz, comutabil la 10 MHz; soclu pentru coprocesor Intel 80387SX; ROM 128 KB; memoria de lucru 2 MB (80 ns), extensibilă cu 2, sau 6 MB, pînă la max. 8 MB; cache 16 KB; Power Management: în microprocesor; monitor: LCD Touch Screen pentru operare cu creion, diagonală 9,4", 25 x 80 caractere, super-twisted nematic (STN), cold cathode fluorescent (CCFT), sidelit LCD; Video-RAM: VGA 640 x 480 puncte, 256 KB video-RAM, reprezentare în 16/32 nuanțe de gri; monitor extern: port VGA pentru cuplarea unui monitor extern cu o rezoluție de 1024 x 768; hard disk 80 MB (IDE 19ms); floppy: 3,5" (1,44 MB) intern; interfețe: conector video VGA, 1 serială (RS232C), 1 paralelă (Centronics), IDE pe motherboard, floppy extern sau Docking Station, încărcare baterie sau baterie externă, conector pentru bloc numeric extern, conector pentru tastatură și mouse, Pen port; slot pentru cartelă fax/modem; tastatura: 79 taste, 12 taste funcționale; Dimensiuni: 292x55x230 mm; Masa: 3,1 kg.



NEC Maus optional

NEC - ProSpeed 486SX/C

Procesor: Intel i486SX-20, 8 KB cache intern; 20 MHz; soclu pentru coprocesor i487SX; ROM: 64 KB sistem/32 KB video; memoria de lucru: 2MB (80 ns) extensibilă pînă la max. 20/32 MB; monitor: 10,4" TFT color monitor cu matrice activă, VGA 640x480 puncte, 256 din 4096 de culori; video RAM: 512 KB; port pentru monitor VGA extern; hard disk: 120 MB (IDE, 19ms); floppy: 3,5" (1,44MB) intern; interfețe: un slot 8/16/32 biți EISA, IDE pe motherboard, conector video VGA, 1 serială, 1 paralelă, floppy extern, conector mouse; slot pentru cartelă fax/modem; tastatura: 92 taste, 12 taste funcționale; bloc numeric separat; dimensiuni: 375x106x396 mm, masa: 7,6 kg.

Lecom

Distribuitor autorizat NEC:
LECOM S.R.L.
Calea Victoriei nr. 126
70179 București
tel.: 50.64.20; 50.79.94; 12.95.58
fax: 12.95.58

Obiectele interfeței utilizator

Este vorba despre obiecte care marchează pe ecran un domeniu din care utilizatorul poate să-și rezolve intrările. Ele pot fi: ferestre, meniuri, icon-uri și boxe de dialog. Windows sprijină programatorul cu funcțiuni predefinite, astfel încât efortul pentru producerea și întreținerea lor să fie minim. Trebuie însă știut modul în care au fost implementate obiectele interfeței utilizator. În plus, importantă mai este și cunoașterea asocierii informațiilor cu obiectele și a anumitor rutine de bibliotecă Windows cu controlul obiectelor.

Aplicațiile Windows pun programatorilor următoarele probleme:

- 1. Aplicațiile trebuie să fie comandate prin evenimente
- 2. Windows cere ca toate ieșirile spre periferice (ecran, imprimantă) să fie de natură grafică
- 3. Folosire consecventă a obiectelor interfeței utilizator

Limbaajul este important

În trecut SDK era singura bază a programării eficiente Windows. Apoi s-au mai folosit și compilatoarele C compatibile Windows. În acest scop se pretează pe lângă produsul Microsoft și compilatoarele firmelor Jensen Partner MetaWare și Zortech.

Problema în cazul programării Windows îl constituie faptul că Windows funcționează comandat de evenimente, prin schimbul de informații dintre diferitele obiecte, el însuși nefiind însă programat orientat obiect.

De aceea programatorii refuză să introducă compilatorul C++ compatibil Windows în speranța unei simplificări a scrierii de aplicații.

Motivul este clar. Proiectanții firmei Microsoft au prezentat o cale de mijloc între o arhitectură "precedentă" și una comandată prin evenimente, deoarece compilatorul C este capabil să lucreze cu arhitectura modernă, doar cu un efort foarte mare. Conflictul apărut între C++ și Windows este comparabil cu contrastul dintre soluții simple și probleme complicate.

Borland vrea să înlăture această anomalie. Cele două vîrfuri ale sale, C++ și Pascal orientat obiect, se livrează deja pentru utilizare sub Windows. Cu ajutorul unei aplicații minimale se va putea arăta că OOP simplifică cu mult programarea. O bibliotecă de clase cu structurile sale tipice de ereditate poate fi

manipulată mult mai ușor decît o culegere de 600 de funcții API.

Convenții

Sub Windows s-a impus convenția de nume maghiară. Scopul ei este de a face codul de program mai clar, astfel întreținerea ulterioară a programelor devine mult mai simplă.

Convenția maghiară, (după țara de origine a inventatorului Charles Simonyi), este un compromis între denumiri cît mai scurte ale variabilelor și funcțiilor, dar și cît mai explicite.

Tabelul 1 ne arată o serie de prefixe care apar destul de des. Astfel prin "ach Name 14" s-a denumit un sector de semn, constînd din 14 elemente, iar "ich" se folosește pentru denumirea indexului unui cîmp grafic.

Pentru identificarea obiectelor, Windows folosește cifre de tip Integer fără semn, avînd o lungime de 16 biți. Un astfel de număr se mai numește și Handle. Singurul lor scop este identificare, iar o conversie în pointer nu dă nici o informație utilă. Unele funcții API returnează un Handle. Față de DOS un NULL-Handle sub Windows este un Handle anulat, iar un Handle de fișier anulat are valoarea "-1", deoarece DOS distribuie Handle-re de fișiere, și nu suprafața Windows.

Fișierul de header Windows

Orice aplicație Windows necesită o referință asupra fișierului WINDOWS.H. Acest fișier este foarte mare (110 KByte)

și nu conține decît definițiile constante-lor simbolice ale celor mai diferite tipuri de date și ale așa-numitelor prototipuri de funcții bibliotecă.

Constantele simbolice folosesc unei mai bune descifrări și întrețineri a unui program. În cazul lor se înlocuiesc niște cifre cu o denumire corespunzătoare. Toate constantele simbolice apar scrise cu litere mari, pentru a le deosebi de nume de variabile (WINDOWS.H definește în jur de 1500).

WINDOWS.H definește și o serie întregă de tipuri de date. Ele au menirea de a mări volumul de tipuri de date ale limbajului C. Pe lângă tipurile uzuale cum ar fi "BOOL", "BYTE", "DWORD" sau "HANDLE" se mai află și definiții de structuri, de ex: "tag RECT". Ele ușurează referirea la diferitele tipuri de date, care aparțin unui obiect (coordonatele colțurilor unui dreptunghi).

Au fost definite prototipuri de funcții de header în fișierul Windows pentru a putea observa deja în timpul compilării, dacă rutinele au fost apelate cu un număr eronat de parametri, cu tipuri greșite de parametri sau cu valori de retur eronate. Ele indică cum ar trebui să arate apelul unei rutine.

Mesaje

S-a mai spus că programele Windows folosesc un model de programare condus prin mesaje. Deci nu există o ordine prescrisă de derulare a unui pro-

Prefix	Tip date	Definiție WINDOWS.H
a	Array (cîmp)	-
ch	Caracter (character)	-
cb	Număr de octeți (count of Bytes)	-
dw	double word (unsigned long)	DWORD
h	Handle	-
hdc	Handle la contextul unității	-
hwnd	Handle la o fereastră	-
i	Index (întotdeauna compus)	-
l	long Integer	LONG
lp	long Pointer (tip compus far)	-
n	Integer	-
np	near Pointer	-
pt	Point (Structură Punct - x,y)	POINT
r	Rectangle (Structură dreptunghiulară)	RECT
sz	Null terminated string	-
w	word (unsigned integer)	WORD

Tabela 1. Prefixele cele mai frecvent utilizate în programarea Windows

gram, mesajele structurează procesul de derulare.

Mesajele pot fi de diferite naturi. Ele pot arăta că a avut loc un eveniment de mouse (deplasare sau clic), că s-a apăsât "tasta" sau că fereastra a fost modificată.

Nu trebuie să ne închipuim că un mesaj ar fi ceva deosebit. În principiu el nu este altceva decât un număr fără semn pe 16 biți. Bineînțeles că în programarea Windows se folosesc niște prescurtări mai pe înțeles decât acele cifre reci. Toate mesajele încep cu "WM" (Windows Message).

Prelucrarea evenimentelor

O aplicație scrisă în C, care utilizează SDK-ul, trebuie să conțină apelul funcției "WinMain". Această rutină reprezintă de fapt punctul de intrare în programul propriu-zis. Declarația arată astfel:

```
int PASCAL WinMain (HANDLE hInstance,
HANDLE hPrevInstance, LPSTR
lpzCmdLine, int cmdShow)
```

"WinMain" trece prin două faze. Prima este așa-numita inițializare de program, cea de-a doua poartă numele de buclă de mesaje. Complexitatea lui "WinMain" constă în inițializare, care constă la rândul ei în executarea a trei rutine de bibliotecă Windows și anume: "RegisterClass", "Create Window" și "Show-Window".

Prin parametrul "Instance" programului i se comunică numele său. Acest nume este necesar identificării. Unele rutine de bibliotecă Windows necesită acest Handle.

Windows caută numele modulului ascuns în fișierul executabil. Acesta poate fi adus într-un format inteligibil utilizatorului cu ajutorul utilitatului "EXEHDR". Dacă se găsește numele unui modul deja executat, atunci se pornește o altă instanță program. Prin parametrul "hPrevInstance" programul află că mai are o "tudă". Programele înrudite folosesc același cod de program și aceleași resurse (nu și în mod obligatoriu), în schimb datele diferă între ele.

Linia de comandă "lpzCmdLine" conține convenția de apel a programului.

Ultimul parametru a lui "WinMain" este "cmdShow". Aceasta informează programele asupra următoarei operații ce va avea loc asupra ferestrei principale.

Bucia de mesaje dă posibilitatea unei întrebări continue a mesajelor. Trebuie avută în vedere mereu menținerea co-

Obiectele Interfeței utilizator

Fereastra: este o parte a aplicației. Ea folosește structura celorlalte obiecte și răspunde de schimbul de mesaje din sistem.

Icon: este un simbol, care caracterizează o parte a ferestrei ca domeniu de acces asupra unei anumite funcții. Astfel: meniul sistem, icon-ul de minimizare și de maximizare sînt domenii de pe ecran, pe care un clic de mouse provoacă o anumită reacție.

Meniuri: Windows oferă cinci tipuri de meniuri: de sistem, de bare, tip Pull-down, imbricate și Tear-off. Ele conțin comenzi și opțiuni de program. Cel mai simplu este meniul sistem, deoarece Windows se îngrijăște ca funcționarea comenzilor să fie unitară în întregul sistem.

Cursor: Cursorul este un bitmap și se deplasează pe ecran corespunzător: cu deplasarea unui mouse sau a unei alte unități de indicare pe ecran. Aspectul cursorului poate fi modificat (săgeată, stop-hand, clepsidră sau altele). Astfel cursorul poate fi adaptat cerințelor aplicației și ușurează perceperea opțiunilor a detaliilor unei aplicații (de ex.: clepsidra la transferul datelor pe harddisk).

Caret: este denumirea cursorului tastaturii, deoarece prin cursor a fost deja numit arătătorul mouse-ului. Fereastra căreia îi este arondată tastatura poate folosi un Caret pentru a arăta că sînt posibile transferi de date prin intermediul tastaturii. Interfețele utilizator Windows nu permit decât folosirea unui singur caret. Însămună că atunci cînd o fereastră primește controlul tastaturii, ea trebuie să producă un caret, iar atunci cînd ea pierde controlul, va trebui să-l distruagă.

Scrollbar: Nu întotdeauna fereastra poate cuprinde întregul domeniu al obiectelor de date. Afișarea barelor de defilare exemplifică această stare de fapt. Cu ajutorul scrollbar-elor utilizatorul poate modifica domeniul de afișare al obiectului de date.

Boxe de dialog: Aceste boxe reprezintă posibilități standardizate de introducere date de către utilizator. Ele sînt necesare, de exemplu, atunci cînd după lansarea unei comenzi mai sînt necesare informații suplimentare pentru executarea ei corectă. Acestea pot fi cerute și respectiv corectate prin intermediul unei boxe de dialog.

municației între părțile individuale ale programului și alte aplicații deschise.

O aplicație minimă Windows constă dintr-o fereastră, un meniu sistem, butoanele de minimizare și de maximizare precum și din domeniul Client. Toate caracteristicile (deplasare, modificarea mărimilor) sînt preluate de către rutinele bibliotecii Windows. Utilizatorul poate stabili cu ajutorul WRT forma cursorului și simbolul pentru aplicație, pe care le poate integra ulterior.

Listingul 1.1 prezintă o aplicație Windows minimală, în care FIRST.C reprezintă programul propriu-zis, în timp ce FIRST.RC, FIRST.ICO și FIRST.CRS conține resurse care se integrează în timpul creării fișierului EXE. Împlicat link-editor-ul Microsoft mai necesită încă un fișier de definire a modulelor, care descrie structura și construcția unui program. Este foarte avantajoasă generarea unui fișier MAKE. Astfel se poate economisi scrierea ardevoasă a liniilor necesare de comandă în cazul unor compilări repetate. Această minimă structură a aplicației trebuie folosită în toate cazurile în care SDK-ul își găsește utilitate.

Listingul 1.2 rezolvă aceeași soluție minimală. Biblioteca de clase "Object Windows" ușurează procesul creării unui program. Necesară este doar declararea unui urmaș al lui "TApplication". Orică metodă, care depășește volumul minim (al aplicației) trebuie definită din nou.

Partea a II-a va trata în principal prelucrarea mesajelor. Miezul oricărui program Windows constă din cel puțin o funcție care poate recepționa și prelucra mesaje. De fapt tratarea mesajelor determină modul de comportare al programului.

Se va vedea însă, că programatorul va trebui să se ocupe doar de mesajele care sînt relevante pentru aplicația sa. Celelalte mesaje pot fi prelucrate de către "Default Windows Procedure", dacă comportamentul standard al ferestrei rămîne neschimbat. În caz contrar, mesajele care trebuie prelucrate diferit, vor trebui captate.

(I.M.)

Instalarea sistemului de operare Netware 2.2

Conceperea și instalarea rețelelor sub Novell Netware ca și actualizarea de la Netware 286 la 386 stau în centrul acestei noi serii de articole. În prima parte ne vom ocupa de instalarea lui Netware 2.2.

Netware 2.2 (denumit adesea și "Netware 286") este urmașul lui Netware 2.1. Versiunea 2.2 înlocuiește toate versiunile anterioare 2.1x și este disponibilă pentru configurații diferite. Ea este oferită pentru 5, 10, 50 și 100 de utilizatori. În ceea ce privește numărul maxim de utilizatori care pot lucra simultan nu este însă nici o diferență între diferitele configurații. Procedurile diferite de instalare dispar.

Caracteristicile de putere ale lui Netware 2.2 le cuprind pe cele conținute până acum de SFT Netware și includ și drepturi de acces și utilizare din Netware 3.11. Novell s-a adaptat astfel evoluțiilor de pe piață și a simplificat mai ales procedura de instalare. Cu versiunea 2.2, Netware a ajuns la a 9-a generație și Novell și-a ținut astfel cuvântul, atunci când a promis ca pe lângă versiunea 3 să dezvolte în continuare și Netware versiunea 2.

Netware, a noua

Nu întotdeauna trebuie utilizat Netware 3.11. Mai ales în cazul rețelelor mici și atunci când sînteți la prima experiență privind lucrul în rețea se recomandă să utilizați Netware 2.2 deoarece cerințele și deci și costurile nu sînt tot atît de ridicate.

Cîteva observații preliminare - pentru instalarea lui Netware 2.2 este necesar un calculator cu statut special pe care Netware 2.2 să ruleze ca sistem de operare și care să preia controlul și comanda rețelei. Acest calculator va fi desemnat ca "Fileserver".

Fileserver și Workstation

Fileserver-ul lucrează așadar sub un sistem de operare special care se cheamă "Netware" și care nu se bazează pe sistemul de operare DOS. Din această cauză Netware nu este supus limitărilor S.O. DOS (de exemplu limitarea la maxim 640 KByte RAM care să poată fi adresată direct).

Celelalte calculatoare care vor funcționa ca posturi de lucru în rețea vor fi desemnate ca "Workstations". În fiecare stație de lucru, ca și în fileserver, se va introduce cîte o cartelă de rețea, și aceste cartele vor fi legate între ele prin cablu. În ceea ce privește cablajul nu vom intra în detalii în acest moment. Deocamdată este suficientă o construcție schematică (fig. 1).

Abia după ce legătura fizică între calculatoare este realizată este posibilă comunicația. Aceste legături presupun anumite limitări, de exemplu încăperile unei secții, sau mai tîrziu clădirea firmei. De aceea se vorbește despre o rețea "LAN" (Local Area Network), deci despre o rețea care se limitează la un domeniu local.

Într-o rețea resursele (datele programelor sau fișierele aplicației, imprimantele, ș.a.m.d.) pot fi partajate între utilizatori. Pentru aceasta este necesară o anumită logică de comandă. Pe lângă stațiile de lucru sînt necesare deci și calculatoare care să preia anumite sarcini de comandă ("Server"). Unul dintre acestea a fost deja amintit - fileserver-ul. Acest calculator reprezintă nucleul rețelei. Resursele care sînt cuplate la fileserver stau, mai mult sau mai puțin, la dispoziția tuturor utilizatorilor. Acestea sînt în special harddisk-urile și imprimantele instalate ca atare în rețea.

Fileserver-ul trebuie să îndeplinească anumite cerințe: el ar trebui să fie cel puțin un calculator compatibil 100% 286. Procesoarele din generații mai noi sînt mai indicate.

Fileserver-ul - un calculator nobil

Un calculator AT cu tactul de 6 MHz nu satisface necesitățile de viteză ale unei rețele. Este mai indicat un AT cu tactul de 16 MHz sau și mai bine un 386.

Acest calculator trebuie să aibă memoria de cel puțin 1,5 MByte RAM, se recomandă 2,5 MByte. Adesea memoria unui calculator poate fi crescută doar în pași de MByte întregi, de aceea este bine dacă fileserver-ul dispune de 4 MByte RAM.

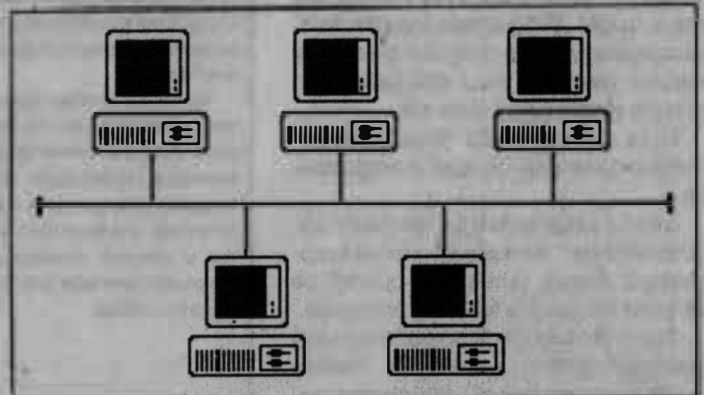
Dacă în rețea se instalează un harddisk cu o capacitate mai mare de 70 MByte, pentru fiecare 100 MByte suplimentari fileserver-ul ar trebui să aibă cîte un Mbyte RAM în plus. Dacă pe server se vor instala și alte aplicații sub forma unor așa-numite "VAP" (Value Added Processes) va trebui să se țină seama și de necesarul de memorie al acestora.

Pe fileserver Netware poate fi instalat în două moduri: în mod "Dedicated", caz în care server-ul stă în exclusivitate la dispoziția sistemului de operare Netware; în modul "Non-Dedicated" pe lângă Netware mai rămîne disponibil spațiu și pentru un proces DOS, astfel încît calculatorul poate fi utilizat și ca Workstation. Din cauza măsurilor de siguranță care trebuie luate, modul Non-Dedicated "înghite" încă un MByte de memorie.

Dedicat sau nededicat, aceasta e întrebarea

Acum urmează să se instaleze harddisk-ul și cartela de rețea în Fileserver.

Fig. 1. Construcția schematică a unei rețele



După ce instalarea hardware-ului este încheiată se poate începe instalarea sistemului de operare Netware.

Porniți copierea! Nu, nu a discheteilor. Înainte de a face copii de siguranță după dischetele originale citiți capitolul 1 din cartea "Installing/Maintaining the Network". Acolo veți găsi mai multe pagini cu "Worksheets", care trebuie copiate. Important: în Worksheets-urile cu pricina trebuie scrise valorile setate, pentru a avea mai târziu o documentație a instalării și a vă putea lămurii rapid problemele care ar putea să apară.

După etapa Worksheets puteți începe copierea dischetelor. În pachet, în afară de Netware-Box veți găsi un alt pachetel cu inscripția "SERIALIZED DISKETTE", care conține o dischetă cu numele "OSOBJ". Aceasta stabilește configurația - deci dacă în rețea se vor putea cupla 5, 10, 50 sau 100 de utilizatori. În plus aceasta conține numărul de serie Netware.

Comparativ cu Netware 2.1x numărul de dischete a scăzut simțitor. Acest lucru s-a realizat prin livrarea produsului doar pe dischete High-Density (1,2 respectiv 1,44 MByte). În varianta 5 1/4 inch, furnitura conține 18 dischete.

Nu toate aceste dischete trebuie copiate. Chiar mai mult, este suficient să se copieze dischetele cu inscripțiile "SYSTEM-1", "SYSTEM-2", "OSEXE" și "OSOBJ" (utilizați comanda DISK-COPY).

În continuare va trebui să luați o altă hotărâre care va influența în mod hotărâtor durata instalării. Instalarea poate decurge în întregime pe fileserver. Pentru aceasta se va porni instalarea de pe discheta "SISTEM-1", lansând în execuție programul "INSTALL". Dischetele care vor fi necesare în continuare vor fi cerute de programul de instalare. Cu această metodă ajungi curînd să te simți ca un diskjockey.

Dacă dispuneți deja de o rețea cu un Fileserver Netware și nu doriți un upgrade (o actualizare), atunci există o metodă mai elegantă și care necesită mai puțin timp. Pentru aceasta este necesară o stație de lucru care să dispună de un lector de dischete capabil să citească dischetele din furnitura Netware 2.2. Această stație de lucru poate fi chiar viitorul fileserver, care este "agățat" în rețeaua existentă ca post de lucru, dacă acest lucru este posibil (de exemplu prin

înlocuirea unei stații de lucru) și care dispune de shell-ul Netware pentru viitorul Fileserver (mai ales de fișierul IPX.COM).

În acest caz trebuie să deschideți o sesiune de lucru pe fileserver-ul disponibil (să vă logați?!), dar trebuie să aveți grijă să aveți drepturi de scriere într-un anumit director. Acest lucru poate fi verificat cu ajutorul comenzii "RIGHTS" (RIGHTS director). În continuare aveți nevoie de cca. 18 MByte spațiu liber pe un harddisk al fileserver-ului. Pentru a verifica acest criteriu puteți utiliza una dintre comenzile "CHKVOL" sau "VOL-INFO". Urmează să realizați un Drive Mapping pe volumul, corespunzător, de ex. cu comanda: "MAP N: =SYS:". Și pe această unitate, deci pe N, trebuie apoi să treceți.

Ca alternativă la utilizarea unei stații de lucru se poate utiliza și un calculator DOS cu un spațiu de memorare liber de cel puțin 18 MByte. Lucrul pe harddisk are avantajul că modificările se pot face foarte repede, fără a mai trebui să se introducă discheta potrivită.

Odată ce vă aflați pe o unitate cu capacitate de memorare suficientă va trebui să creați un nou director "NETWARE" și să vă mutați în acesta. Pe discheta "SYSTEM-1" se găsește un program

cu numele "UPLOAD" care poate fi apelat cu comanda "A:UPLOAD". La "destination path" va trebui să specificați directorul NETWARE tocmai creat, specificînd și unitatea. Apoi vi se va cere să introduceți dischetele Netware într-o anumită succesiune.

Upload creează în directorul "NETWARE" o structură de directoare care corespunde dischetelor individuale. Apoi poate începe instalarea propriuzisă. Din directorul "NETWARE" se va apela programul "INSTALL".

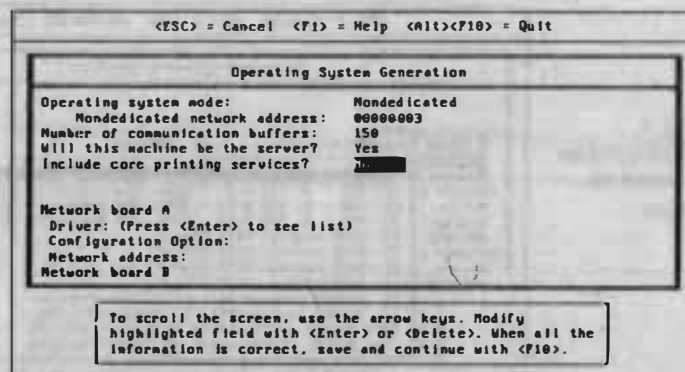
Așa cum se poate vedea din fig. 2, instalarea cu programul INSTALL constă din patru module diferite. Meniul principal oferă opțiunile "Basic Installation", "Advanced Installation", "Maintain existing System", respectiv "Upgrade from Netware v 2.x". Primele două opțiuni se ocupă cu instalarea simplă sau avansată, a treia permite modificarea unui server deja existent, ultima opțiune servește actualizării (upgrade) unei versiuni deja existente, dar mai vechi, pe fileserver la versiunea Netware 2.2.

Opțiunea "Basic Installation" impune instalarea cu parametrii de bază. Opțiunea "Advanced" permite modificarea oricărui parametru. Ecranul cu mesajul de bun venit poate fi părăsit tastînd <ENTER>. În acest moment pe moni-

Nr. modul	Nume modul	Executabil pe	Parametru pentru INSTALL
1	Operating System Generation	viitorul fileserver, harddisk-ul altui calc.	Basic: niciunul Adv. : -E
2	Linking and Configuring	viitorul fileserver, harddisk-ul altui calc.	niciunul
3	ZTEST (Track Zero Test)	viitorul fileserver	-L
4	File Server Definition	viitorul fileserver	-F

Fig. 2. Parametrii de apel pentru "INSTALL"

Fig. 3. Modulul "Operating System Generation"



tor va apare primul modul al instalării: "Operating System Generation" (asemănător cu fig. 3).

Primul cîmp determină modul de lucru al fileserver-ului. "Dedicated" va avea ca efect un server dedicat pe care nu va putea rula decît Netware. "Non-Dedicated" permite simultan și un proces DOS, astfel încît server-ul poate fi utilizat și ca stație de lucru.

Avantajul celei de-a doua variante, economisirea costurilor pentru o stație de lucru implică însă și dezavantaje semnificative: pe de o parte pe Fileserver nu pot fi rulate programe care utilizează memoria extinsă (de exemplu Windows 3.0). Pe de altă parte un program care rulează ca proces DOS se poate bloca și poate determina astfel căderea server-ului și deci a întregii rețele.

Procesul DOS al unui server nededicat necesită o adresă în rețea, care este văzută ca o adresă logică de cartelă de rețea. Această "Nondedicated Network Address" trebuie să fie diferită deci de toate adresele cartelelor de rețea instalate!

Operation Buffer

Numărul buffer-elor de comunicație (Communication Buffer) specifică numărul blocurilor rezervate pentru buffer-e pe fileserver pentru interogări din partea stațiilor de lucru. Conținutul buffer-elor este prelucrat de acesta cu urgența maximă. Un bloc ocupă 512 Byte de memorie. Dacă buffer-ul este plin și nu mai poate prelua alte interogări, stațiile de lucru vor trebui să-și repete cererea.

O valoare prea mică influențează deci negativ viteza stațiilor de lucru, o valoare prea mare determină o su-

praîncărcare inutilă a memoriei fileserver-ului. Se pot da valori cuprinse între 40 și 1000. Valoarea standard de 150 de buffer-e este o bună valoare de pornire.

Dependent de faptul dacă instalarea s-a lansat de pe harddisk sau nu, va trebui răspuns la întrebarea: "Will this machine be the server?". Dacă se răspunde cu "No" vor fi rulate numai primele două module de instalare. Apoi dischetele vor trebui trecute printr-o operație "download" și cu aceste dischete procesul de instalare va continua, pornind de la al treilea modul, pe viitorul fileserver. Dacă instalarea se face de pe harddisk-ul unui calculator DOS sau de pe o stație de lucru va trebui să răspun-deți "No".

"Yes", dimpotrivă, determină continuarea instalării, cu urmarea că după parcurgerea celui de-al treilea modul veți pierde toate datele care le aveți pe harddisk-ul fileserver-ului! Dacă instalarea se execută de pe un server existent - viitorul fileserver fiind momentan legat ca stație de lucru - va trebui să răspun-deți "Yes".

La "Include Core Printing" se specifică dacă tratarea imprimantelor trebuie integrată în rețea ("Yes"). Acest răspuns are ca efect posibilitatea cuplării imprimantelor din rețea la fileserver (max. 5 buc.). Dacă în rețea ar trebui integrate și imprimante cuplate la stații de lucru, noțiunea Netware "Printservers" vorbește de la sine și deci aici nu poate fi luat în considerare decît un răspuns "No". Despre Printserver vom vorbi mai târziu.

Urmează să stabiliți specificațiile pentru cartela de rețea (Network board) și harddisk-ul (Channel) fileserver-ului. În acesta pot fi instalate pînă la 4 cartele de rețea, care vor fi denumite de la "LAN

A" pînă la "LAN D". LAN A, respectiv "Networkboard A" este obligatorie, toate celelalte sînt opționale. Un sfat: procesul DOS al unui server nededicat îl puteți denumi "LAN E".

Tipul cartelei de rețea îl puteți specifica în cadrul opțiunii "Driver", după ce ați tastat <ENTER>. Va trebui să selectați acel driver care corespunde cartelei de rețea disponibile. Dacă utilizați o cartelă Ethernet, acestea sînt adesea compatibile cu cartelele Novell NE-2000, al căror driver-e există deja în furnitura Netware. Cartelele Arcnet pot utiliza adesea driver-ul "Netware-RX-Net" care este de asemenea în furnitură.

Dacă aveți dubii va trebui să parcurgeți manualul cartelei de rețea (majoritatea suportă Netware). Dacă nu găsiți nici un driver potrivit, atunci e de sperat că odată cu cartela de rețea vi s-a livrat și o dischetă cu un driver pentru Netware. Adesea pe aceste dischete există doar drivere pentru Netware 2.1x. Aceste drivere pot fi utilizate totuși și pentru Netware 2.2. După ce ați tastat <Insert> și ați introdus discheta cu pricina puteți selecta driver-ul suplimentar.

Cartelele de rețea utilizează configurații diferite pentru întreruperi. Configurația instalată trebuie specificată în cadrul opțiunii "Configuration Option" (fig. 4).

Opțiunea 0 corespunde de regulă configurației de lucru. Valorile standard pentru Ethernet sînt Interrupt 3 și I/O Adresă 300, pentru Arcnet Interrupt 2, I/O - Adresă 2E0 respectiv D000 ca buffer RAM. Aceste valori depind desigur de hardware, ca de pildă de interfețele seriale și paralele, cartela grafică, etc. Este indicat să verificați datele pentru cartela de rețea.

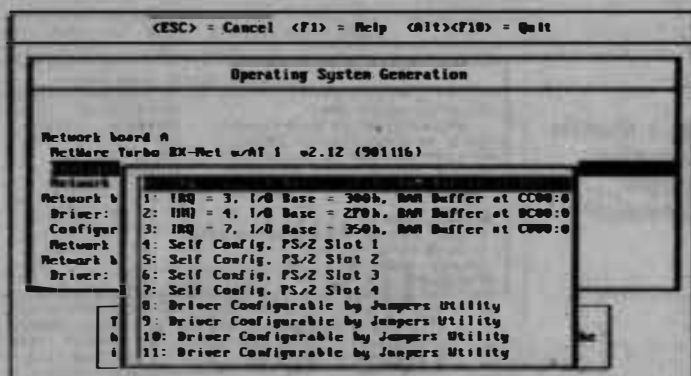
Fiecare cartelă de rețea necesită o adresă de rețea neambiguă (Network Address). Adresa de rețea este un număr hexazecimal cu opt poziții (0 pînă la F).

Această procedură (Driver, Configuration Option, Network Address) trebuie repetată pentru toate celelalte cartele de rețea. Toate valorile date pînă în acest punct va trebui să le copiați întocmai în copile worksheet.

(continuarea în numărul următor)

(R.M.)

Fig. 4. Configurarea cartelei de rețea





micro
ATCI S.R.L.

str. Gheorghe Doja 36
Tîrgu-Mureş ROMÂNIA
Adresa poştală: C.P. 64, O.P. 1
4300 Tîrgu-Mureş
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. !

Cîte ceva despre VGA

Relativ puține informații sînt de găsit în ce privește programarea cartelelor grafice VGA - nu în ultimul rînd datorită lipsei unui standard acceptat.

Organizarea memoriei grafice la VGA.

Memoria grafică este organizată pe patru plane, fiecare plan avînd capacitatea de 65536 octeți (64 Ko). Planele sînt numerotate de la 0 la 3, și sînt adresate paralel, astfel că este necesar un spațiu de adresare de doar 64 Ko (de la A000:0000 la A000:FFFF).

Cum se adresează memoria ecran?

a) Scrierea. Porturile "Sequence Controller Index" se programează astfel:

-scriere în memoria ecran:

Port{\$3C4} = 2; (Map Mask Register)

- precizarea planului:

Port{\$3C5} = valoare;

(cuprinsă între 1 și 15)

Bitul i (i între 0 și 3) al valorii înscrise la portul \$3C5 are următoarea semnificație: informația va fi înscrisă în planul i dacă bitul i are valoarea 1. Pot fi puși pe 1 unul sau mai mulți biți; în acest al doilea caz informația va fi înscrisă simultan în mai multe plane. Exemplu:

Port{\$3C4} = 2;

Port{\$3C5} = 5; Mem[\$A000:5] = 201;

Valoarea 201 va fi înscrisă simultan în planele 0 și 2, la adresa relativă 5.

b) Citirea. Porturile "Graphics Controller Index" se programează astfel:

- citire din memoria ecran:

Port{\$3CE} = 4; (Read Map Register)

- precizarea planului:

Port{\$3CF} = valoare;

(cuprinsă între 0 și 3)

Valoarea înscrisă la portul \$3CF dă numărul planului de unde se va citi informația. Evident, la un moment dat putem citi dintr-un singur plan.

Exemplu:

Port{\$3CE} = 4;

Port{\$3CF} = 1; B = Mem[\$A000:9];

Variabila B (de tip byte) va primi valoarea aflată în planul 1 la adresa relativă 9.

Programarea paletelor de culori pentru modurile cu 256 culori.

Cartela VGA poate afișa simultan 256 culori dintr-o paletă de 262144 (2^{18}) culori.

Pentru programarea paletelor de culori

se apelează întreruperea BIOS \$10 cu următoarele valori în regiștri:

ah = \$10; al = \$10; bh = 0;

bl = index-culoare;

ch = i-green; cl = i-blue; dh = i-red;

Semnificația valorilor din regiștri:

index-culoare = valoare între 0 și 255;

i-red, i-green, i-blue = intensitatea co-

respunzătoare componentelor Roșu, Verde și Albastru;

Modul 320x240x256

În acest mod ecranul se prezintă astfel:

P(0, 0) P(1, 0) ... P(319, 0);

P(0, 1) P(1, 1) ... P(319, 1)

...

P(0,239) P(1,239) ... P(319,239)

Cele patru plane de memorie sînt organizate astfel:

Planul 0:

P(0, 0) P(4, 0) ... P(316, 0);

P(0, 1) P(4, 1) ... P(316, 1)

...

P(0,239) P(4,239) ... P(316,239)

Planul 1:

P(1, 0) P(5, 0) ... P(317, 0);

P(1, 1) P(5, 1) ... P(317, 1)

...

P(1,239) P(5,239) ... P(317,239)

Planul 2:

P(2, 0) P(6, 0) ... P(318, 0);

P(2, 1) P(6, 1) ... P(318, 1)

...

P(2,239) P(6,239) ... P(318,239)

Planul 3:

P(3, 0) P(7, 0) ... P(319, 0);

P(3, 1) P(7, 1) ... P(319, 1)

...

P(3,239) P(7,239) ... P(319,239)

Am indicat între paranteze coordonatele x și y ale pixelilor.

Programarea modului 320x240x256 se realizează astfel:

- se apelează funcția BIOS pentru programarea în modul standard, 320x200x256;

- se programează cîteva regiștri ai controlorului VGA pentru a obține 480 linii pe verticală și apoi pentru a scana de două ori aceeași linie (astfel se obține o rezoluție de 240 linii pe verticală); rezoluția pe orizontală se păstrează (320 pixeli pe linie);

- se programează de asemenea controlorul VGA pentru a folosi toate cele patru plane de memorie, similar cu mo-

dul 640x480x16 (deoarece modul 320x200x256 folosește un singur plan de memorie).

De ce am ales modul 320x240x256 ?

Acest mod prezintă mai multe avantaje față de celelalte moduri:

- fiecărui pixel îi corespunde un octet în memoria ecran, de unde rezultă o viteză de prelucrare de cîteva ori mai mare față de modurile cu 16 culori, unde operațiile la nivel de pixel sînt destul de laborioase;

- acest mod funcționează peoricetip de VGA;

- față de celelalte moduri cu 256 culori, acest mod prezintă de asemenea avantaje:

- densitatea pixelilor este aceeași pe orizontală ca și pe verticală, deci dacă se desenează un cerc sau un pătrat, de exemplu, nu trebuie făcute operații suplimentare de ajustare a coordonatelor x și y;

- este posibilă paginarea memoriei ecran.

Modul 360x480x256

Acest mod diferă de cel prezentat anterior prin faptul că cele patru plane de memorie ecran sînt organizate fiecare ca o matrice de dimensiuni 480x90. Programarea acestui mod diferă de asemenea prin faptul că se programează atît rezoluția pe orizontală (360 pixeli pe linie, deci 90x4), cît și cea pe verticală (480 linii, deci aici fiecare linie va fi scanată o singură dată).

În articolul de față sînt prezentate cîteva programe scrise în Turbo Pascal care ilustrează folosirea modului 320x240x256 (Xmod + Xgrafica) și a modului 360x480x256 (Cmod + Cgrafica). Precizăm că programele au fost testate pe mai multe calculatoare compatibile IBM-PC, cu VGA de tip Oti's și OAK.

Într-un articol viitor vom prezenta un program de animație folosind tehnicile de paginare a memoriei ecran. Programul va folosi modul 320x240x256 datorită avantajelor pe care le prezintă față de alte moduri grafice.

mat. Nelu Cozac

```

1 uses cgrafica,crt;
2 var i,j:integer;
3 begin
4   csetgraph;
5   csetcolor(5,63,0,63);
6   csetcolor(9,0,0,63);
7   for i:=0 to 479 do
8     begin
9       cwriteln(0,i,5);
10      cwriteln(1,i,5);
11    end;
12   for i:=0 to 479 do
13     begin
14       cwriteln(359,i,5);
15       cwriteln(358,i,5);
16     end;
17   for j:=0 to 359 do
18     begin
19       cwriteln(j,0,5);
20       cwriteln(j,1,5);
21     end;

```

```

1 uses xgrafica,crt;
2 var i,j:integer;
3 begin
4   xsetgraph;
5   xsetcolor(5,63,0,63);
6   xsetcolor(9,0,0,63);
7   for i:=0 to 239 do
8     begin
9       xwriteln(0,i,5);
10      xwriteln(1,i,5);
11    end;
12   for i:=0 to 239 do
13     begin
14       xwriteln(319,i,5);
15       xwriteln(318,i,5);
16     end;
17   for j:=0 to 319 do
18     begin

```

```

1 unit xgrafica; ($D-,L-,S-)
2
3 interface uses dos;
4
5 procedure xsetgraph;
6 procedure xwriteln(x,y:word; p:byte);
7 function xreadpixel(x,y:word):byte;
8 procedure xsetcolor(c,r,g,b:byte);
9 procedure xterminate;
10
11 implementation
12
13 var regs:registers; i,j,k:word;
14   me:array[0..239,0..79] of byte absolute $a000:0;
15
16 const valori:array[1..10] of word=
17   ($0d06,$3e07,$4109,$ea10,$ac11,
18   $d12,$0014,$c715,$0616,$e317);
19
20 procedure xsetgraph;
21 begin
22   regs.ax:=19; intr(16,regs);
23   portw[$3c4]:=$604;
24   portw[$3c4]:=$100;
25   port[$3c2]:=$e3;
26   portw[$3c4]:=$300;
27   port[$3d4]:=$11;
28   port[$3d5]:=port[$3d5] and $7f;
29   for i:=1 to 10 do portw[$3d4]:=valori[i];
30   portw[$3c4]:=$f02;
31   fillchar(me,240*80,#0);
32 end;
33
34 procedure calcula(x,y:word);
35 begin

```

```

22   for j:=0 to 359 do
23     begin
24       cwriteln(j,479,5);
25       cwriteln(j,478,5);
26     end;
27   for i:=2 to 477 do
28     for j:=2 to 357 do
29       cwriteln(j,i,9);
30     delay(2000);
31     port[$3c4]:=2; port[$3c5]:=1; delay(20);
32     fillchar(mem[$a000:0],480*90,#0);
33     port[$3c4]:=2; port[$3c5]:=2; delay(20);
34     fillchar(mem[$a000:0],480*90,#0);
35     port[$3c4]:=2; port[$3c5]:=4; delay(20);
36     fillchar(mem[$a000:0],480*90,#0);
37     port[$3c4]:=2; port[$3c5]:=8; delay(20);
38     fillchar(mem[$a000:0],480*90,#0);
39   cterminate;
40 end.

```

CMOD.PAS

```

19   xwriteln(j,0,5);
20   xwriteln(j,1,5);
21 end;
22 for j:=0 to 319 do
23   begin
24     xwriteln(j,239,5);
25     xwriteln(j,238,5);
26   end;
27   for i:=2 to 237 do
28     for j:=2 to 317 do
29       xwriteln(j,i,9);
30     delay(2000);
31     xsetcolor(5,0,63,0);
32     xsetcolor(9,63,63,0);
33     delay(2000);
34   xterminate;
35 end.

```

XMOD.PAS

```

36   k:=x and 3; i:=y; j:=x div 4;
37 end;
38
39 procedure xwriteln(x,y:word; p:byte);
40 begin
41   calcula(x,y);
42   port[$3c4]:=2; port[$3c5]:=1 shl k;
43   me[i,j]:=p;
44 end;
45
46 function xreadpixel(x,y:word):byte;
47 begin
48   calcula(x,y);
49   port[$3ce]:=4; port[$3c5]:=k;
50   xreadpixel:=me[i,j];
51 end;
52
53 procedure xsetcolor(c,r,g,b:byte);
54 begin
55   with regs do
56     begin
57       ah:=16; al:=16; bh:=0; bl:=c;
58       ch:=g; cl:=b; dh:=r;
59     end;
60   intr(16,regs);
61 end;
62
63 procedure xterminate;
64 begin
65   regs.ax:=3; intr(16,regs);
66 end;
67
68 end.

```

XGRAFICA.PAS

```

1 unit cgrafica; ($D-,L-,S-)
2
3 interface uses dos;
4
5 procedure csetgraph;
6 procedure cwriteln(x,y:word; p:byte);
7 function creadpixel(x,y:word):byte;
8 procedure csetcolor(c,r,g,b:byte);
9 procedure cterminate;
10
11 implementation
12
13 var regs:registers; i,j,k:word;
14   me:array[0..479,0..89] of byte absolute $a000:0;
15
16 const valori:array[0..9] of byte=
17   ($6d,$59,$5a,$8e,$62,$8b,$0b,$3e,0,$40);
18
19 const valorj:array[0..7] of byte=
20   ($ea,$8c,$df,$2d,0,$e7,$04,$c3);
21
22 procedure csetgraph;
23 begin
24   regs.ax:=19; intr(16,regs);
25   portw[$3c4]:=$604;
26   portw[$3c4]:=$100;
27   port[$3c2]:=$e3;
28   portw[$3c4]:=$300;
29   port[$3d4]:=$11;
30   port[$3d5]:=port[$3d5] and $7f;
31   for i:=0 to 9 do
32     begin
33       port[$3d4]:=i; port[$3d5]:=valori[i];
34     end;
35   for i:=0 to 7 do
36     begin
37       port[$3d4]:=i+16; port[$3d5]:=valorj[i];
38     end;
39   portw[$3c4]:=$f02;
40   fillchar(me,480*90,#0);
41 end;
42
43 procedure calcula(x,y:word);
44 begin
45   k:=x and 3; i:=y; j:=x div 4;
46 end;
47
48 procedure cwriteln(x,y:word; p:byte);
49 begin
50   calcula(x,y);
51   port[$3c4]:=2; port[$3c5]:=1 shl k;
52   me[i,j]:=p;
53 end;
54
55 function creadpixel(x,y:word):byte;
56 begin
57   calcula(x,y);
58   port[$3ce]:=4; port[$3c5]:=k;
59   creadpixel:=me[i,j];
60 end;
61
62 procedure csetcolor(c,r,g,b:byte);
63 begin
64   with regs do
65     begin
66       ah:=16; al:=16; bh:=0; bl:=c;
67       ch:=g; cl:=b; dh:=r;
68     end;
69   intr(16,regs);
70 end;
71
72 procedure cterminate;
73 begin
74   regs.ax:=3; intr(16,regs);
75 end;
76
77 end.

```

CGRAFICA.PAS

Cine indexează mai bine ?

Testul	486 /25MHz/16Mb 200Mb/16ms	286/15 MHz/1Mb 40Mb/16ms	286/12MHz /1Mb 40Mb/28ms
Creare si indexare	7'55"	23'28"	32'20"
Interogare 1	32"	55"	1'37"
Interogare 2	9"	20"	30"

```

1  *
2  * CLIPPER 5.0/5.01
3  *
4  * DEMCLIP creeaza un fis. secv.indexat in acces
5  * direct de 100000 articole, apoi interogheaza in
6  * acces direct 1000 de articole in doua modalitati
7  *
8  * compilare: > clipper demclip /m/n/w/a
9  * linkeditare: > rlink fi demclip
10 * /
11
12 STATIC i, j, l := 0
13 STATIC rest
14 STATIC secvf
15 STATIC Wch_Cit1 [1000]
16
17 STATIC m_ab1
18 STATIC m_ab2
19 STATIC m_altc1 := "INF.GR1"
20 STATIC m_altc2
21 STATIC m_altc3 := "INF.GR2"
22 STATIC m_altc4
23
24 FIELD AB1 // N, 6
25 FIELD AB2 // C, 2
26 FIELDALTC1 // C, 6
27 FIELDALTC2 // N, 6
28 FIELDALTC3 // C, 6
29 FIELDALTC4 // N, 6
30
31 PROCEDURE main()
32
33 USE DEMCLIP
34 ZAP
35 INDEX ON STR(AB1,6)+AB2 TO DEMCLIP
36 ? * PROGRAM DEMCLIP"
37
38 ? *INDEXARE INCEPUTA: *+ TIME()
39 j := 0
40 secvf := 100000
41 FOR i := 1 TO 50000
42   Gen_Fis()
43 NEXT
44 ? *INDEXARE TERMINATA: *+ TIME()
45
46 USE DEMCLIP READONLY
47 SET INDEX TO DEMCLIP
48
49 ? *INTEROGARE 1 INCEPUTA: *+ TIME()
50 j := 1001
51 FOR i := 1 TO 500
52   Cit1_Fis()
53 NEXT
54 ? *INTEROGARE 1 TERMINATA: *+ TIME()
55
56 ? *INTEROGARE 2 INCEPUTA: *+ TIME()
57 FOR i := 1 TO 1000
58   Cit2_Fis()
59 NEXT
60 ? *INTEROGARE 2 TERMINATA: *+ TIME()
61 USE
62
63 QUIT
64
65 // -----

```

```

66 PROCEDURE Gen_Fis
67
68 secvf--
69
70 rest := i % 26
71 Gen_Ch()
72 m_ab1 := m_altc2 := m_altc4 := i
73 Scr_Fis()
74
75 rest := secvf % 26
76 Gen_Ch()
77 m_ab1 := m_altc2 := m_altc4 := secvf
78 Scr_Fis()
79
80 RETURN
81
82 // -----
83 PROCEDURE Gen_Ch
84
85 LOCAL schimb
86 STATIC Wch_G2 := { "AB", "CD", "EF", "GH", "U";
87   "KL", "MN", "OP", "QR", "ST";
88   "UV", "WX", "YZ", "ab", "cd";
89   "ef", "gh", "ij", "kl", "mn";
90   "op", "qr", "st", "uv", "wx";
91   "yz", "" }
92 rest ++
93 m_ab2 := Wch_G2[rest]
94
95 schimb := Wch_G2[rest]
96 Wch_G2[rest] := Wch_G2[27]
97 Wch_G2[27] := schimb
98
99 RETURN
100
101 // -----
102 PROCEDURE Scr_Fis
103
104 j ++
105 IF j > 95
106   j := 0
107   l ++
108   IF l < 1001
109     Wch_Cit1 [l] := STR(m_ab1,6)+m_ab2
110   ENDIF
111 ENDIF
112
113 SEEK STR(m_ab1,6)+m_ab2
114 IF FOUND ()
115   ? *CHEIE DUBLA: *+STR(m_ab1,6)+m_ab2
116 ELSE
117   APPEND BLANK
118   AB1 := m_ab1
119   AB2 := m_ab2
120   ALTC1 := m_altc1
121   ALTC2 := m_altc2
122   ALTC3 := m_altc3
123   ALTC4 := m_altc4
124 ENDIF
125
126 RETURN
127
128 // -----
129 PROCEDURE Cit1_Fis
130

```

Clipper 5.01

Programul a fost compilat cu CLIPPER 5.01 si executat pe 3 calculatoare, cu FILES=50 si BUFFERS=35 in CONFIG.SYS.

Dimens.programului executabil: 223 Ko
Dimens. fișierului de date (dbf): 3,3 Mo

Dimens.fișierului index (ntx): 3,8 Mo
(dupa reindexare: 1,9 Mo)

Observatii:

1. Fișierul creat nu are o dispersie uniformă.

2. Testul 1 este relevant în prelucrări de tip "pe loturi", nu și în prelucrări conversaționale, specifice calculatoarelor personale.

În schimb lipsesc teste ca:

- parcurgere secvențială,
- parcurgere indexată,
- adăugarea, modificarea, ștergerea unui numar de articole cu cod aleator.

3. Fișierul index rezultat după o reindexare se micșorează la jumătate. Porțiunile micșorării se datorează dispersiei neuniforme. Cu un fișier index reorganizat, evident scad și timpii de parcurgere indexată și unele interogări. Timpii de interogare din tabel au fost obținuți cu fișierul index nereorganizat.

HAMOR Soft SRL

4000 - Sfintu Gheorghe c.p. 140
tel: 923/12139, 15648, fax: 12139

```

131 SEEK Wch_Cit1 [i]
132 IF ! FOUND ()
133   ? *CHEIE NEGASITA: *+Wch_Cit1 [i]
134 ENDF
135 j--
136 SEEK Wch_Cit1 [j]
137 IF ! FOUND ()
138   ? *CHEIE NEGASITA: *+Wch_Cit1 [j]
139 ENDF
140
141 RETURN
142
143 // -----
144 PROCEDURE Cit2_Fis
145
146 SEEK Wch_Cit1 [i]
147 IF ! FOUND ()
148   ? *CHEIE NEGASITA: *+Wch_Cit1 [i]
149 ENDF
150
151 RETURN
152
153 // -----

```

DEMCLIP.PRG

Quattro Pro

Crearea de copii multiple

Dacă într-o tabelă doriți să copiați un domeniu de mai multe ori, în mod normal, trebuie să marcați acel domeniu de fiecare dată și apoi să-l inserați în locul dorit. Trucul următor vă ajută să rezolvați această problemă mai simplu.

- 1. Selectați meniurile *Edit-Names-Create* și dați litera "K", de exemplu, ca nume pentru domeniul sursă.
- 2. Marcați domeniul care trebuie copiat de mai multe

ori și apoi tastați <Enter>.

- 3. Mutăți cursorul în colțul din stînga sus al domeniului în care doriți să inserați prima copie.
- 4. Tastați <Ctrl> + <C> pentru a apela meniul *Edit-Copy*. În loc să marcați din nou domeniul care trebuie copiat, tastați <K>, după care apăsați de două ori tasta <Enter>. În acest mod puteți copia un domeniu de ori cîte ori doriți.

Salvare fișier la un simplu clic pe mouse

Dacă doriți să memorați un fișier pe disc, apelînd meniurile *File-Save*, și există deja un fișier pe hard disk cu nume identic, o să primiți un mesaj de avertizare, care vă va semnala acest lucru. Va trebui să selectați *Replace* deși nu doreați altceva decît să faceți o nouă copie de siguranță.

Paleta de funcții selectabile cu ajutorul mouse-ului, aflată în marginea din dreapta a ecranului, vă poate oferi o soluție mai simplă:

- 1. Selectați meniurile *Options-Mouse-Palette*.
- 2. Selectați cîmpul paletei mouse căruia doriți să-i încredințați funcția de salvare. Pentru aceasta puteți folosi cea de-a cincea

tastă - *5th Button* - care în mod standard poartă inscripția "5" și determină apelarea unui macro pentru redarea unui sunet.

- 3. În cadrul submeniului *Text*, dați o denumire proprie cîmpului, denumire care să nu fie mai lungă de 3 caractere.
- 4. Selectați submeniul *Macro* și tastați următoarea succesiune de caractere:

Mac

- 5. Dacă selectați apoi *Quit*, veți vedea noul cîmp apărînd în meniul din marginea din dreapta a ecranului. În viitor este suficient să poziționați cursorul mouse-ului pe acest cîmp și să dați un clic-mouse.

Revenire rapidă

Quattro Pro dispune de o structură de meniuri ierarhică, cu ajutorul căreia se selectează o anumită funcție sau se stabilesc anumiți parametri. De exemplu pentru a selecta o diagramă de tipul 3D, trebuie pătruns pînă pe al treilea nivel de meniuri: *Graph-Graph Type-3D*.

Adesea însă se selectează un astfel de meniu doar

pentru a vedea care sînt opțiunile disponibile. Pentru a reveni în tabelă, în acest caz, trebuie apăsată de trei ori tasta <Esc>. De aici și trucul: dacă se tastează <Ctrl> + <Break> se sare peste toate nivelele intermediare și se revine în tabelă.

Cuplare rapidă tabele

Ca și Lotus 1-2-3 și Quattro Pro oferă posibilitatea introducerii într-o tabelă a unor așa-numite formule de cuplare prin intermediul cărora se stabilește o legătură cu anumite cîmpuri, sau domenii, din alte tabele. Pentru aceasta trebuie făcută o referire externă, specificîndu-se numele tabelii dorite și al cîmpului, sau al domeniului, ca de ex.:

```
+1,14*[P:ApplVOT.WQ1]A35
```

Această posibilitate este foarte puternică, dar referința implică tastarea multor caractere. Cu ajutorul unui truc, o formulă care să se refere la mai multe tabele se poate scrie mult mai simplu. Să presupunem că există trei fișiere: BRASOV.WQ1, CLUJ.WQ1 și ARAD.WQ1, care în cîmpul E17 conțin numărul alegătorilor partici-

panți la vot. Pentru a aduna aceste valori într-o a patra tabelă ar trebui tastată următoarea formulă:

```
@SUM([CA:QPRO\BRASOV.WQ1]E17; [CA:QPRO\CLUJ.WQ1]E17; [CA:QPRO\ARAD.WQ1]E17)
```

În loc să tastați o formulă atât de lungă puteți să rezolvați problema mult mai simplu executînd următorii pași:

- 1. Încărcați toate tabelele care trebuie cuplate.
- 2. Treceți în tabela în care doriți însumarea.
- 3. Tastați formula: @SUM(!*E17)
- 4. După ce ați tastat <Enter> această formulă este transformată în formula anterior descrisă. Motivul: asterix-ul va determina cuplarea tuturor fișierelor deschise în acel moment.

Marcare rapidă

De regulă Quattro Pro este compatibil cu Lotus 1-2-3, dar există și câteva deosebiri semnificative. Una dintre acestea este de exemplu suprafața *Pull-Down* și operaerea cu mouse. Tocmai la marcarea unui domeniu cu mouse-ul se poate folosi un truc interesant. Atunci cînd se dorește marcarea unui domeniu mare se poziționează cursorul în colțul din stînga sus al domeniului, se apasă tasta din stînga a mouse-ului și ținîndu-se apăsată se extinde marcajul pînă în colțul din dreapta jos al domeniului.

Mai simplu se poate rezolva problema în modul următor:

Vă rugăm să ne scrieți părerea dumneavoastră despre această nouă rubrică și să specificați și pa-chetele software pentru care ați dori să prezentăm astfel de trucuri.

Nu uitați să ne trimiteți propriile descoperiri!

Corecții Borland în unit-ul »editors« (Turbo Vislon)

Din păcate, în unit-ul »editors« din TurboVision s-a strecurat o greșeală. Puteți opera însă și D-voastră corciturile; trebuie schimbate numai câteva linii în fișierul »editors.pas«, după cum urmează.

Secvența

```
LODSW
CMP     BL,DL
```

va trebui să înlocuiască secvența

```
LODSW
OR      BL,BL
JE      @@2
CMP     BL,DL
```

din rutina ScanKeyMap. Secvența

```
JE      @@4
CMP     BH,DH
JE      @@4
```

înlocuiește secvența

```
JE      @@4
@2:    CMP     BH,DH
JE      @@4
```

din rutina ScanKeyMap. Secvența

```
var
  ShiftState: Byte absolute $40:$17;
  Key: Word;
```

înlocuiește secvența

```
var
  Key: Word;
  din rutina ConvertEvent; În fine, secvența
```

```
begin
  if (ShiftState and $03 <> 0) and
    (Event.ScanCode >= $47) and
    (Event.ScanCode <= $51) then
    Event.CharCode := #0;
    Key := Event.KeyCode;
```

înlocuiește secvența

```
begin
  Key := Event.KeyCode;
  din rutina ConvertEvent. Deoarece dischetele de referință erau deja produse iar unit-ul este disponibil în format sursă, Borland s-a decis pentru acest mod de operare a corecțiilor.
```

Borland

Cod autoreproducător

»Clona.c« este un program care se »autoreproduce«. Spre deosebire însă de cei mai mulți reprezentanți ai acestei »specii«, care datorită unor virulente »oi negre« se bucură de o nemeritată faimă proastă, programul »Clona« nu duplică codul executabil ci produce o imagine exactă a propriei sale surse. (Desigur, pentru acest lucru nu se folosește sursa existentă, ca original pentru o copie banală; nu se face nici un fel de referință spre un fișier »clona.c« existent.) De fapt, »Clona.C« nu face uz de nici un fel de comenzi explicite pentru ieșire în fișier: duplicarea se obține prin faptul că output-ul programului este redirecat spre un fișier cu același nume. Programul se lansează cu sintaxa:

```
clona > clona.c
```

Soluționarea problemei a început cu căutarea unei instrucțiuni (de ieșire) care să se autoreproducă; mai exact, care să se conțină pe ea însăși ca argument și să-l »scoată« pe acesta de două ori. Experimentarea unor rutine de construcție proprie nu a fost (din fericire) necesară, deoarece rutina »printf« din biblioteca standard C știe tot ce trebuie pentru acest lucru. Trucul constă în a »împacheta« întregul cod al programului în șirul de formare și de a-l folosi suplimentar pe acesta ca argument-șir. Simplificând, schema de lucru se prezintă de maniera »printf(X,X)«, unde

```
X = "X = %s printf(X,X)"
```

Referința înapoi în această structură evită o recursie infinită după modelul: <instrucțiune-de-ieșire> ("<instrucțiune-de-ieșire>")<instrucțiune-de-ieșire>

Ultimul obstacol - ghilimelele care acționează ca delimitatori de șir și formatarea textului (fragmentarea lui în linii distincte - se trece folosind constante ASCII. (vezi și if 8/91 (mov), pag. 29)

(G. H./I.F.)

```
1: #include <stdio.h>
2: char a[ ] = "#include <stdio.h> %c%cchar
   a[ ] = %c%c%c%c; %cvoid main() {printf
   (a,10,34,a,34,10);}";
3: void main() {printf (a,10,34,a,34,10);}
```

Clona.C

Curs C++ (IV)

(continuare din pag. 48)

```
128public:
129 View::View (int x1, int y1, int x2, int y2, \
130 Lista* p = NULL, Fereastra* e = NULL,\
131 Lista* s = NULL)
132 : Lista (p, e, s),
133 Fereastra (x1,y1,x2,y2)
134 {
135 vizibil = 1;
136 }
137 void View::Afisare()
138 {
139 puttext (get_elem() -> GetX1());
140 get_elem() -> GetY1();
141 get_elem() -> GetX2();
142 get_elem() -> GetY2();
143 get_elem() -> GetBu();
144 }
145);
146
147int main()
148{
149 char tasta;
150 Boolean sfirsit;
151 Fereastra* ff;
152 View* v[5];
153 int z;
154
155 clrscr();
156 ff = new Fereastra (XMin, YMin, XMax, YMax);
157 ff->Umplere ('x',0x16);
158 ff->Afisare();
159
160 v[1] = new View (10, 3, 46, 15);
161 v[1]->Umplere ('A',0x12);
162
163 v[2] = new View (14, 5, 50, 17);
164 v[2]->Umplere ('B', 0x2f);
165
166 v[3] = new View (18, 7, 54, 19);
167 v[3]->Umplere ('C', 0x40);
168
169 v[4] = new View (22, 9, 58, 21);
170 v[4]->Umplere ('D', 0x38);
171
172 v[5] = new View (26, 11, 62, 23);
173 v[5]->Umplere ('E', 0x51);
174
175 v[1]->set_list (NULL, v[1], v[2]);
176 v[2]->set_list (v[1], v[2], v[3]);
177 v[3]->set_list (v[2], v[3], v[4]);
178 v[4]->set_list (v[3], v[4], v[5]);
179 v[5]->set_list (v[4], v[5], NULL);
180
181 for (z = 5; z > 0; z--)
182 v[z]->Afisare();
183
184 printf ("Alegeti un numar între 1..5: ");
185 sfirsit = false;
186 do {
187 tasta = getch();
188 z = tasta - 0x30;
189 v[1]->move_top (z, v[1]);
190 v[1]->Afisare();
191 if (tasta == '0') sfirsit = true;
192 } while (!sfirsit);
193)
FEREAST2.CPP
```

Here comes the music ...

1. INTRODUCERE

În zilele premergătoare Sfințelor Sărbători de Paști, m-am plimbat cu fiica mea prin centrul orașului nostru (Timișoara).

Moștenitoarea mea, cu spiritul ei de observație deosebit, (moștenit!), a sesizat în magazine, prezența unor instrumente jucării electronice, gen "minlorgă", impunând "lepurașului", drept răsplată pentru notele bune de la școală, procurarea acestor jucării... Numai că, "lepurașul" nostru, lucrează din greu la o instituție bugetară, la care salariul multiplu "liberalizat", de abia satisface nevoile curente ale familiei.....

Dar problema tratată, trebuia cumva rezolvată. Fiind pasionat de calculatoare, mi-am adus aminte că, în "limbajul" GW-BASIC, pe care subsemnatul îl stăpânește cel mai bine, există comanda:

SOUND <frecvență>,<durată>, care

produce sunete de frecvență <frecvență> (valori între 37-32767 Hz) și durata <durată> (valori între 0-65535 "tacte" (1/18,2 părți dintr-o secundă)).

De aici, ideea programului ce urmează....

2. PROGRAMUL REALIZAT

M-am gândit să transform tastatura PC-ului meu ("rablă", fără HD, cumpărat cu banii de mașină, spre disperarea soției! (dar în fond, oricum, în viața aceasta eu nu-mi voi putea cumpăra DACIA!)), într-o "miniorgă" electronică, care să satisfacă dorințele Deliei, (fiica mea!). A rezultat programul MUZICABAS, prezentat alături.

3. CONCLUZII.

"lepurașul", anul acesta, a fost "grozav". Deși ideea acestui program este extrem de simplă, la îndemâna oricui, economiile realizate de "lepuraș" au fost substanțiale, iar bucuria Deliei uriașă.

Unde mai punem la socoteală faptul că fiica mea se "joacă" permanent cu calculatorul "rablă" al tăticii....

Atenție tăticii! N-o să ajungeți prea des să lucrați la calculatorul Dvs., căci el va fi "acaparată" de copil!

Programul "merge", evident și pe TMS-S, și pe..., cu modificări minore.

În încheiere, tasteți "melodia":

QQWQQRE QRRQQTR...

Distracție plăcută!

Ș.l. ing. Marințeu Dimitrie M.Gh.
U.T. Timișoara, Fac. Mecanica

Reversi - 64:0

Este imposibil să fi lansat Windows fără să fi încercat măcar o partidă de "Reversi". Iar dacă ați început să pierdeți mai mult decât să câștigați, de la un anumit nivel în sus, aveți acum ocazia să vă "răzbuțuiți" crunt, administrându-i calculatorului o "bătăuță" cum n-a mai văzut.

Să nu mai pomenim de impresia pe care o lăsați asupra prietenilor (care încă nu s-au abonat la "if") dacă rețineți pe de rost succesiunea de mișcări care vă asigură o victorie de 64:0 (!!).

Nu uitați să alegeți nivelul de dificultate dorit; soluțiile prezentate mai jos presupun că vă imaginați tabla de joc cam ca o tablă de șah; colțul din stînga sus este A1, cel din dreapta jos H8. La începutul jocului, cîmpurile D4 și E5 sînt acoperite de piesele roșii.

Michael Kett/Bernd Nies

c5 - c6; f3 - d2; b1 - a7; g5 - pass;
c7 - b5; h3 - b6; a8 - b7; e7 - g7;
a5 - c4; d1 - c2; b8 - e6; h8 - g8;
d3 - b4; a6 - f5; e8 - g4; f8 - pass;
b3 - a4; c1 - f2; h4 - h5; f7 - You
a3 - c8; e1 - d8; h6 - h2; Won by
f4 - c3; e2 - f1; h1 - pass;64;
d6 - e3; g1 - a2; g6 - f6;
d7 - g3; a1 - b2; g2 - h7;

Solutia pentru "Novice"

c5 - e8; g8 - f7; f2 - f1; b4 - a4;
f5 - g4; d8 - g6; g1 - d2; h8 - a2;
f4 - f3; h5 - h6; b1 - c3; a1 - pass;
h4 - h3; h7 - g2; c1 - e1; b5 - pass;
h2 - c4; d7 - e3; d1 - pass; a3 - pass;
g3 - c6; c7 - c8; b7 - b6; a4 - You
g5 - f6; b8 - pass; a6 - a7; Won by
d6 - e7; d3 - c2; a8 - g7; 64;
e8 - f8; h1 - e2; b3 - b2;

Solutia pentru "Expert"

```

10 REM MENU ALEGERE NUMAR OCTAVE
20 CLS : PRINT : PRINT " Program realizat de iepuras pentru Delia"
30 PRINT "....."
40 PRINT " MENU ALEGERE OCTAVE"
50 PRINT "....."
60 PRINT " 1-0 OCTAVA"
70 PRINT " 2-DOUA OCTAVE"
80 PRINT " 3-PATRU OCTAVE"
90 PRINT " 4-SFIRSIT"
100 PRINT "....."
110 INPUT "ALEGEREA TA: "; N%
120 IF (N% 1 OR N% 4) THEN PRINT "GRESEALA!": GOTO 20
130 ON N% GOTO 140, 500, 800, 1000
140 REM PRIMA "SUBROUTINA" MUZICALA!
150 CLS : PRINT : PRINT " CIND DORESTI SA TE OPRESTI, APASA "; 0; "!"
160 PRINT " TASTE MUZICALE:q,w,e,r,t,z,u,i,Q,W,E,R,T,Z,U,I!"
170 PRINT " PENTRU DURATE DE SUNET DOUBLE, LITERE MARI!"
180 a$ = INKEY$
190 IF a$ = "q" THEN SOUND 130, 2
200 IF a$ = "w" THEN SOUND 147, 2
210 IF a$ = "e" THEN SOUND 165, 2
220 IF a$ = "r" THEN SOUND 174, 2
230 IF a$ = "t" THEN SOUND 196, 2
240 IF a$ = "z" THEN SOUND 220, 2
250 IF a$ = "u" THEN SOUND 245, 2
260 IF a$ = "i" THEN SOUND 262, 2
270 IF a$ = "Q" THEN SOUND 130, 4
280 IF a$ = "W" THEN SOUND 147, 4
290 IF a$ = "E" THEN SOUND 165, 4
300 IF a$ = "R" THEN SOUND 174, 4
310 IF a$ = "T" THEN SOUND 196, 4
320 IF a$ = "Z" THEN SOUND 220, 4
330 IF a$ = "U" THEN SOUND 245, 4
340 IF a$ = "I" THEN SOUND 262, 4
350 IF a$ = "0" THEN GOTO 20
355 goto 180
500 REM .....SIMILAR....DAR MAI MULTE TASTE ACTIVE!
800 REM .....SIMILAR....DAR SI MAI MULTE TASTE ACTIVE!
1000 END
    
```

MUZICA.BAS

În contra virusurilor

Dacă folosești în mod curent programul SCAN (McAfee) pentru controlul antivirus (versiunea cea mai nouă: SCAN99) adăugați următoarele semnături în SCAN.EXT:

```

*FA 2E 8C 1E 72 05 BC C8 07*   EDCL
*61 24 FC E6 61 2E C6 06 1F*   Eddie-1800 (Dark Avenger xx?)
*FA 2E 8C 1E 8D 0E BC 7C 15*   Gluc (Lipici, Timișoara, ...)
*50 1E 06 B4 25 B0 01 0E 1F*   Dracula
    
```

Iar dacă vă interesează colaborări serioase în ce privește organizarea unei "rețele" naționale pentru protecția sistemelor și prevenirea/combaterea efectelor virusilor, iată o adresă care vă va fi fără îndoială de folos: HotSoft S.R.L., C.P.172, O.P.1, 4300 Tirgu Mureș.

```

1 {program: doctor.pas
2 autor: Helmut Schuhmacher}
3
4 uses Cri, Dos;
5 type P = array[0..511] of Byte;
6 var Z: array[0..39,1..9,0..1] of ^P;
7     N: array[0..39,1..9,0..1] of Boolean;
8     q,w,u,k,o: Byte;
9     R: Registers; c,d: Char;
10    J: Integer;
11    A: Boolean;
12 function Y: Byte;
13 begin
14   Rah:= 1; Intr(19,R); Y:= R.ah;
15 end;
16 procedure X(q,w,u,k,o: Byte; var B: P);
17 begin
18   Rah:= q;
19   R.al:= 1;
20   R.dl:= w;
21   R.dh:= o;
22   R.ch:= u;
23   R.cl:= k;
24   R.bx:= Ofc(B);
25   R.cs:= Seg(B); Intr(19,R)
26 end;
    
```

```

27
28 begin
29 d:= 'r'; A:= True; ClrScr;
30 Write('Discdoctor: 720 sect. = 360 ko');
31 GotoXY(1,3);
32 for u:= 0 to 39 do
33 for k:= 1 to 9 do
34 for o:= 0 to 1 do
35 begin
36   N[u,k,o]:= false;
37   Write(Ord(N[u,k,o]));
38 end;
39 repeat
40 begin
41   A:= d='w';
42 A:= NOT A; q:= 3-Ord(A);
43           GotoXY(1,15);
44   WriteLn(d,' Drive(0=A, 1=B)? ');
45   ReadLn(c); Val(c,w,J);
46   GotoXY(1,3);
47 for J:= 0 to 32767 do Port[$3F2]:= 48 + w;
48 end;
49 begin
50 for u:= 0 to 39 do
51 for k:= 1 to 9 do
52 for o:= 0 to 1 do
53 begin
54   if N[u,k,o]= NOT A then
55     begin
56       if A then New(Z[u,k,o]);
57       X(q,w,u,k,o,Z[u,k,o]^);
58       if Y= 0 then N[u,k,o]:= A
59       else begin
60         Dispose(Z[u,k,o]);
61         X(0,w,u,k,o,Z[u,k,o]^);
62       end;
63       Write(Ord(N[u,k,o]));
64     end;
65     WriteLn;
66   end;
67   GotoXY(1,18);
68   WriteLn('DISK: read, write, end');
69   ReadLn(d) until d='e'
70 end.
71
    
```

Print - o comandă ignorată?

Deși este una din comenzile vechi ale sistemului DOS, surprinzător de puțină lume știe să se folosească eficient de această comandă. Ea permite listarea în "background" la imprimantă, utilizatorul PC-ului fiind liber să facă altceva în timp ce "curge" listarea. Mai ales la listările care duresc mult - vezi orice listare care merge în regim grafic pe o imprimantă matriceală - folosirea acestei posibilități se recomandă cu prisosință. Chiar dacă acest lucru înseamnă un mic efort suplimentar - cum ar fi de exemplu listarea prealabilă a unui fișier mai mare în mai multe fișiere mici pe disc, pentru a putea relua convenabil listarea în cazul în care se întâmplă ceva cu vreo pagină.

Un doctor pentru dischete

Programul alăturat servește la regenerarea unor dischete greu citibile, care inițial au fost scrise corect dar apoi au ajuns să provoace deseori erori la citire, fie datorită unor manipulari greșite, fie datorită unor zgârieturi, cimpuri magnetice, căldurii, frigului, îndoirii sau pur și simplu datorită descălfirării unității floppy. Deasemenea, programul poate fi util la citirea unor dischete care au fost presate și care nu pot fi salvate decât după scoaterea lor din învelișul deteriorat și introducerea lor într-un alt înveliș, în bună stare, care să permită rotirea dischetei. Cu ajutorul acestui program veți putea, în multe cazuri, să citiți corect dischete pe care poate că altfel le-ați abandonat. "Doctorul" funcționează după următoarea metodă:

discheta cu probleme poate fi citită de oricâte ori se dorește. Sectoarele citite odată corect rămân în memoria de lucru a calculatorului și nu trebuie recitite. La repetarea citirii nu vor fi citite decât acele sectoare a căror citire corectă încă nu a reușit; se repetă acest lucru de cîte ori este nevoie pentru a reuși citirea dischetei.

Discheta problematică poate fi între timp plimbată și pe cealaltă unitate (dacă dispuneți de două unități floppy). După fiecare tură, programul arată câte sectoare au fost citite corect. Dacă toate sectoarele au fost citite corect sau a "n"-a citire nu mai aduce nici un fel de îmbunătățire, puteți scrie toate sectoarele citite pe o altă dischetă, după care aveți o copie bună a dischetei originale. Indicații de utilizare: lansați "doctorul" apelînd programul. Pe ecran veți putea urmări 720 de zerouri - câte unul pentru fiecare sector. Fiecare 0 semnifică starea unui sector de pe dischetă, și anume faptul că (încă) nu a putut fi citit corect. La citirea corectă a unui sector, zero-ul corespunzător lui se preschimbă-n "1". După prima parcurgere a dischetei, se repetă întrebarea asupra următoarei acțiuni: (r- citire, w- scriere, e- ieșire). Pentru scriere se recomandă o dischetă pe care ați formatat-o în prealabil fără erori. De data aceasta, trecerea citirilor afișate din 1 în 0 semnifică faptul că sectorul corespunzător a putut fi scris fără erori. Desigur, și scrierea u puteți repeta de un număr oarecare de ori, dacă este necesar. Deoarece programul - unul din candidații la premiul pentru programele ce ocupă cel mult 1024 ko în sursă, la colegii noștri de la DOS International - a renunțat la orice întrebare suplimentară de siguranță ("Sigur vreți să scrieți pe discheta aceasta?"), este recomandabil să vă feriți de posibilități de greșală, protejînd la scriere discheta pe care doriți să o refacți.

DOCTOR.PAS

(Helmut Schuhmacher/I.F.)

Cod	Nume pachet	Nr. dischete	Format	Capacitate	Limba
Sisteme de operare					
29	DR - DOS 6.0	1	3,5"	1,4M	E
Proiectare circuite electronice					
41	Addi-Data: Signal Tools	1	5,25"	1,2M	E
42	MicroSim Corporation: Design Center Demo	1	5,25"	1,2M	E
43	ORCAD	1	5,25"	1,2M	E
44	ORCAD: PC Board Layout Tools	1	5,25"	360K	E
Aplicații					
12	CAPP	1	5,25"	1,2M	D
2	Buch, Trio	2	5,25"	360K	D
39	BALPRO - balanță contabilă	1	5,25"	360K	R
40	UNO - completare formulare	1	5,25"	360	R
45	Hamor Soft: hMISS - secretariat, hCONT - contabilitate, hGEN&hUTIL - dezvolt. aplicații	1	5,25"	360	R
Utilitare, bibliotecă					
14	Borland Windows Tools	1	5,25"	360K	D
17	KAO BTX-Dekoder + Demo	1	5,25"	360K	D
36	QuickStep Tools	2	5,25"	1,2M	D
6	Spindrift Libraries Demo	1	5,25"	360K	E
7	BKS Demo	1	5,25"	360K	D
38	List v 3.0	1	5,25"	360K	R
Baze de date					
10	Paradox 3.0	1	5,25"	360K	D
13	Foxpro 1.02	2	5,25"	360K	D
31	Foxpro 2.0	2	3,5"	720K	D
18	SOS - Software Tools Datenbank	1	5,25"	360K	D
24	Advanced Revelation	1	3,5"	720K	E
25	Rbase 3.1	1	3,5"	720K	D
32	WindowBase	1	5,25"	1,2M	D
33	Address One	1	5,25"	360K	D
37	Superbase 4	2	3,5"	720K	D
Grafică, DTP, CAD					
11	Time Works Publisher	2	5,25"	360K	D
15	MEGA CAD	1	5,25"	360K	D
27	Aldus Page Maker 4.0	1	3,5"	720K	E
30	Draw Perfect 1.1 si Office 3.0	1	3,5"	1,4M	D
9	Micrografx	1	5,25"	360K	D
Cursuri					
5	DOS - Curs IBM	1	5,25"	360K	D
Procesoare de texte, OCR					
28	Word Perfect for Windows	1	3,5"	1,4M	D
35	Scout OCR	1	5,25"	360K	D
40	Ekta Editor V1.5	1	5,25"	360K	E
Măsură și control					
3	Labtech Notebook Demo	1	5,25"	1,2M	E
4	Labtech Control	1	5,25"	1,2M	E
34	Scope	1	5,25"	360K	D
Jocuri, diverse					
8	Pearl Agency	1	5,25"	360K	D

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. Acest din urmă punct 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.

Vă rugăm ca pe comanda pe care ne-o trimiteți să specificați pe lîngă numele pachetului și capacitatea și formatul dischetelor pe care doriți să vă furnizăm produsul.

SYSTEM PLUS

este distribuitor autorizat pentru toata gama produselor



si



NR.1 IN PC-URI DIN SUA SI EUROPA

NR. 1 IN FIABILITATE

In acelasi timp va oferim si solutia unui furnizor unic pentru:
RETELE NOVELL si UNIX, SOFTWARE, PERIFERICE, ACCESORII, FAX, UPS
de la cele mai renumite firme americane si japoneze:

**EPSON, MICROTEK, SUMMAGRAPHICS, HOUSTON
INSTRUMENT, MICROSOFT, BORLAND, CLARIS, ALDUS s.a.**

Asistenta tehnica, scolarizarea si sevice-ul in colaborare cu
LAB. DE ELECTRONICA FUNCTIONALA, FACULTATEA DE ELECTRONICA, I.P.B.

tel/fax (90)791391, (90)108342

Talon de comandă-abonament

Subsemnat(a/ul).....
doresc să-mi trimiteți | _____ | exemplare din fiecare număr al revistei *if*,
începînd cu numărul | _____ |, pe adresa:

Telefon: acasă..... la servici.....
Pentru aceasta am expediat prin mandat postal/virament/cec.....lei în
contul nr. 40729-96010402 deschis la Banca Română pt. Dezvoltare Mureș.
Abonamentul funcționează pînă la epuizarea sumei; în cazul că banii pe care
i-am trimis nu mai ajung pentru un alt număr, voi fi anunțat de acest lucru în timp
util, astfel încît nu risc să pierd nici un număr. Dacă mă răzgîndesc și renunț la
abonament, mi se returnează toată suma rămasă.

if 3 / 92 **Data:** **Semnătura**

Important !

În curînd va apare cartea

Programarea în limbajul Pascal
editată de Micro ATCI Tîrgu-Mureș.

Manualul este o traducere din limba
germană și este destinat atît celor care
vin în contact pentru prima dată cu
acest limbaj de programare cît și celor
care doresc să se perfecționeze
învățînd noi metode și tehnici de pro-
gramare. Volumul este bogat în exem-
ple, toate programele prezentate fiind
încercate și testate cu ajutorul compl-
latorului Turbo Pascal 6.0 al firmei Bor-
land.

Tirajul fiind limitat, asigurați-vă că
nu veți pierde lucrarea expediind o
comandă fermă pe adresa:

Micro ATCI, C.P. 64, 4300 Tîrgu-Mureș

*"if" nr. 4/92 va apare pe piață în jurul
datei de 1 iulie 1992!*



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.

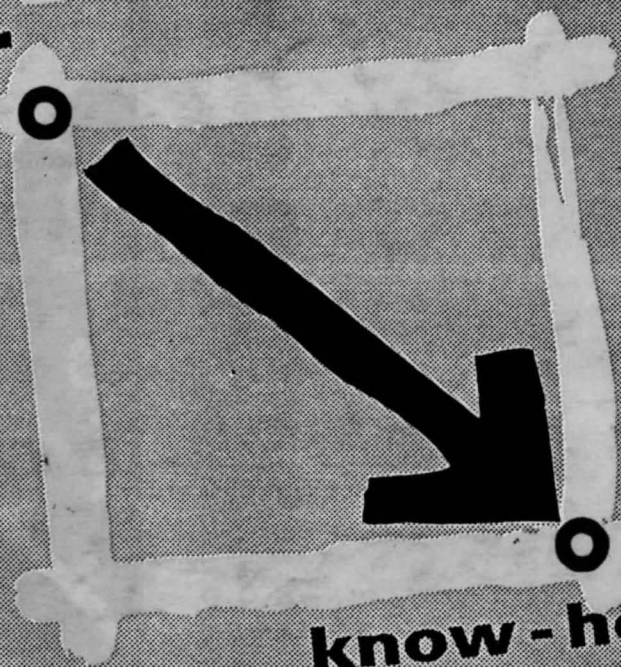
99.94% of the users fully satisfied of the delivered products

Authorised distributors for the following companies:

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

*The shortest
way between*

CAD
user



the
know-how in
AUTOCAD

is passing through



A & C
INTERNATIONAL S.A.

AUTODESK AUTHORISED DEALER FOR ROMANIA



Tel: 53.53.15
Fax: 12.77.74 Telex: 10674 MAXIM R
Technical Support - Service "Hot Line": 12.77.73