

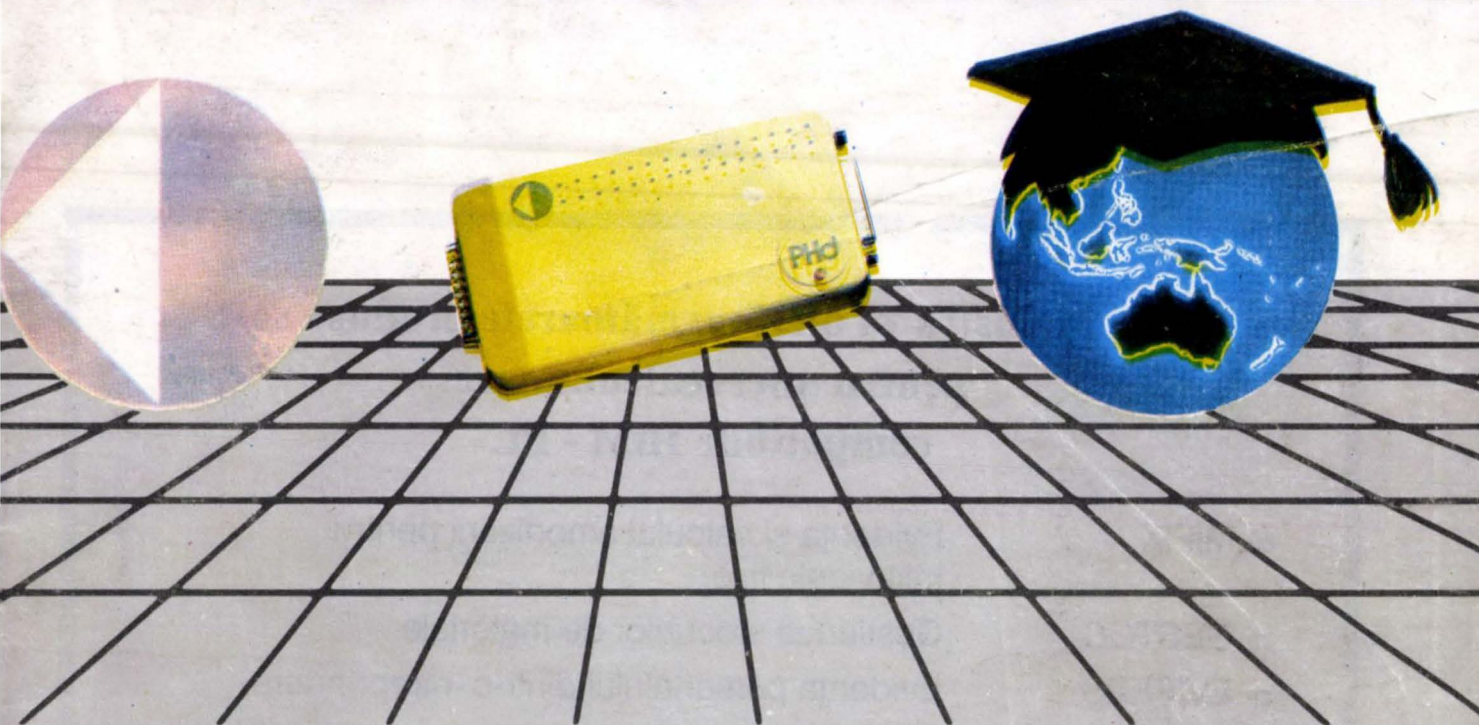
if

CALCULATOARE
PERSONALE

5 / 92
(16)
150 lei

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

ISSN 1220-1529



Pocket Hard Disk

- ★ Affordability
- ★ Portability
- ★ User Friendly
- ★ Security
- ★ Simplicity
- ★ 40 MByte to 120 MByte formatted capacity

...and able to withstand shocks of 150G"

PHD is easily connected to any personal computer parallel port, providing a trouble free, hard disk drive installation.

PHD performs as the next logical drive on the host system, allowing the user to use his own software and data anywhere a computer is available. Once stored, the disk can be removed ensuring absolute data security.

With **PHD**, managers, financiers, engineers and service personnel can carry all their computer files with them.

With **PHD** any computer screen will look like your own.



str. Gheorghe Doja 36

C.P. 64, O.P. 1

4300 Tîrgu-Mureş

Tel./Fax: 954 - 31660

**Firma noastră vă oferă următoarele produse soft
pentru microcalculatoare
compatibile IBM - PC**

- MIFIX Evidenţa şi calculul amortizării pentru mijloacele fixe
- GESTOC Gestiunea stocurilor de materiale
- EVIPERS Evidenţa personalului dintr-o întreprindere
- SALARII Calculul salariilor
- BALANŢA Balanţa contabilă şi "Cartea Mare"
- CASA Întocmirea registrului de casă
- BANCA Gestiunea operaţiunilor cu banca
- REPARAŢII Planificarea şi urmărirea reparaţiilor utilajelor dintr-o întreprindere
- METR Evidenţa verificării pentru aparate de măsură şi control
- ISCIR Evidenţa verificării pentru instalaţii sub incidenţa ISCIR

Produsele sînt realizate sub sistemul de gestiune a bazelor de date
PARADOX Versiunea 3.5

Cuprins

Editorial	5
Noutăți	6

OS/2

OS/2 2.0 A meritat așteptarea	10
De ce OS/2?	10
Ce este nou în OS/2 2.0?	10

OS/2 Versiunea 2.0

Lupta pentru locul la fereastră	14
---	----

C.A.C.

Mașina de minciuni PC	16
--	-----------

Interviu

INTERVIU Bill Gates	20
--------------------------------------	-----------

Fundamente

Metoda Backtracking	22
0. Pretext	22
1. Punerea problemei	22
2. Rezolvarea problemei	22
3. Algoritmul Backtracking	22
4. Probleme rezolvabile cu ajutorul metodei Backtracking	22

Rețele Neuronale

Ce este analiza senzitivă?	26
Programul exemplu	27
Analiza rezultatelor	28
Pentru dezvoltări ulterioare	29

Rețele

Administrarea rețelelor	37
Crearea de utilizatori și securitatea	37
Gestionarea aplicațiilor	37
Salvări	38
Listări	38

NetWare	38
LAN Manager	39
LAN Server	40
Care este cel mai bun?	41

ARTISOFT O revoluție în conectarea calculatoarelor	42
---	-----------

Baze de date

Un tabel de control pentru succes	45
Prima și a doua generație DMS	45
Problemele tipice ale generației a II-a a sistemelor DMS	46
Soluțiile oferite de cea de-a doua generație	47
Direcții de orientare: abordare practică	50
Rezumat	51

Hardware

Flash stîrnește senzație	52
O privire asupra procesării Over Drive	54

Laborator

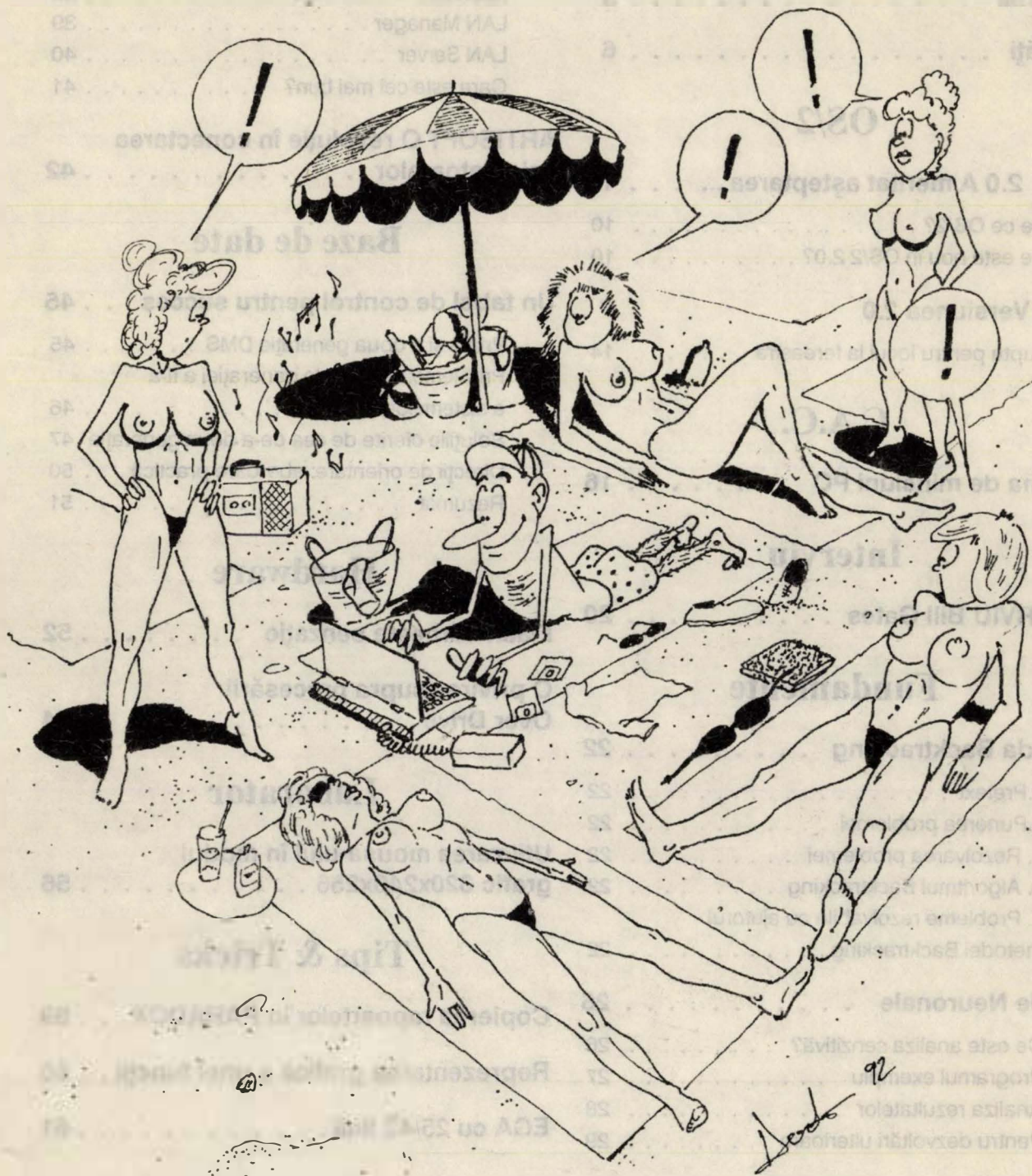
Utilizarea mouse-ului în modul grafic 320 x240 x256	56
--	-----------

Tips & Tricks

Copierea rapoartelor în PARADOX	59
Reprezentarea grafică a unei funcții	60
EGA cu 25/43 linii	61

Anunțuri

SHARE - if	62
PROPUNERE	62



92
Dede

if

revistă de informatică

editată de firma

Micro ATCI

Director:

ing. Dumitru Dunca

Redacția:

ing. Cristian Nagy
Doina Ceșa

Colaboratori externi:

stud. Vasile Antonescu
ing. Mihai Beer
ing. Cristian Cosma
mat. Ioan Cozac
ing. Octavian Cudalbu
mat. Eugen Rotaru
stud. Tiberiu Socaciu

Tiparul:

tipografia Tîrgu-Mureș

Tiraj: 4000 ex.
Preț: 150 lei

Manuscrisele originale sau listin-
gurile de programe sînt permise cu
plăcere cu condiția să nu fi fost pu-
blicate în altă parte. Prin expedierea
unui manuscris pe adresa redacției
autorul consimte implicit la publica-
rea materialului său în revistă. Ono-
rariul se negociază cu redacția.

Materialele republicate nu se
restituie.

În sfîrșit, un nou număr din "if"... Sper că, așa cum sugerează titlul unuia din articolele acestui număr, a "meritat așteptarea". Care (ea, așteptarea) a fost - trebuie să recunosc - destul de lungă.

Mulțumesc tuturor celor care nu ne-au uitat. Și, avînd în vedere că ochii care nu se văd se uită, mă voi strădui ca de acum înainte să ajung la o frecvență mai mare a aparițiilor. Eu mi-aș dori să apărem lunar. Vom face totul ...

Numărul de față insistă ceva mai mult asupra versiunii 2.0 a celebrului sistem de operare OS/2 elaborat de IBM. Presa spune că, deja, OS/2 nu mai are nevoie de "presă proastă". Cele două articole despre OS/2 intenționează să constituie numai un fel de prezentare a acestui sistem de operare pentru calculatoare personale. Deși nu este strict legat de tema OS/2, vă recomand să citiți incitantul și incisivul interviu cu William Gates - președintele firmei Microsoft. Veți vedea ce înseamnă "presă proastă" pentru OS/2.

Cele două articole grupate sub genericul "Fundamente" sînt foarte interesante, deși par a se adresa mai ales teoreticienilor. Totuși, poate vor interesa și pe alții.

Despre rețelele de calculatoare veți întîlni două articole. Primul face o comparație între serviciile oferite administratorului unei rețele (știi, ei, administratori de rețele cum poate fi viața uneori ...) de către trei importante sisteme de operare. Cel de-al doilea articol este o prezentare a firmei ARTISOFT și sistemului de operare pentru rețele locale numit LANtastic. Poate vă va stîrni curiozitatea.

Din "Laborator", colegul nostru Nelu Cozac continuă să vă dea informații despre modul X al interfeței VGA (320 x 240 x 256). Acum a trecut la șorice.

Țin să vă atrag atenția asupra caricaturii de pe pagina 4: este a doua dintr-o serie de desene foarte reușite care ne-au fost trimise de un viitor matematician: studentul Valentin Antonescu.

Poate că v-ar trezi interesul și o privire asupra anunțului de pe ultima pagina. Așteptăm cu nerăbdare reacțiile dumneavoastră - evident, sper că veți fi entuziasmați.

Dar cum orice lucru are un capăt (în afară de cîrnat care are două, după cum spune un proverb arab), închei această introducere cu un îndemn "mobilizator" primit de la unul din cititorii noștri: Go For "if" !!!

Cristian Nagy

Sistemul informațional al Olimpiadei Barcelona '92 bazat pe OS/2

În cazul unui eveniment ce se desfășoară la o scară atât de grandioasă, ca Jocurile Olimpice de vară, obținerea informațiilor în timp util este un serviciu de maximă importanță pentru sportivi, crainici, reporteri, spectatori și membrii de familie. Inima sistemului informațional olimpic de la Barcelona, 1992, a constituit-o sistemul de operare OS/2 al lui IBM.

Sistemul informațional pentru rezultate, sistemul pentru comentatori, sistemul informațional pentru familia olimpică și sistemul informațional pentru conducerea operațiunilor s-au bazat pe două calculatoare gazdă mainframe IBM 9021 și peste 4400 de stații de lucru PS/2 lucrând sub OS/2. De acest sistem informațional olimpic a depins satisfacerea cerințelor a peste 15000 de sportivi, 11000 de reprezentanți de presă, 40000 de membri ai "familiei olimpice" și 30000 de voluntari.

"Un sistem informațional de o asemenea mărime ridică probleme deosebite, inclusiv servirea a peste 96000 de utilizatori cu un grad de instruire în computere foarte diferit" spunea Fernand Sarrat, director adjunct cu probleme de dezvoltare a pieței de sisteme personale de la IBM.

OS/2-ul este proiectat pentru a rezolva aceste probleme și pentru a gestiona cereri suplimentare ale utilizatorilor, cum ar fi procesare locală, multitasking și conectivitate impecabilă între cele patru sisteme primare.

OS/2 integrează într-un mediu unic uneltele necesare dezvoltării aplicațiilor complexe ce compun sistemul informațional olimpic.

Sistemul informațional pentru rezultate, cu 400 de calculatoare PS/2 legate în rețea, a fost instalat în fiecare bază sportivă. El permite introducerea manuală sau automată a rezultatelor competițiilor și furnizează aceste rezultate în diferite formate, cum ar fi legătura directă cu calculatoarele agențiilor de presă sau afișări pentru crainicii TV.

Sistemul informațional și de comunicații olimpic, bazat pe 2000 de PS/2-uri legate în rețea, asigură ziariștilor, arbitrilor, sportivilor și voluntarilor accesul la informații generale cum ar fi rezultatele întrecerilor, programarea concursurilor, situația medaliilor, informațiile pentru presă și buletinele meteorologice. Acest sistem asigură un serviciu de poștă electronică (E-mail) pentru membrii familiei olimpice și pentru presă.

Sistemul pentru comentatori, bazat pe 1100 de PS/2-uri legate în rețea, permite comentatorilor de radio și televiziune să aibă acces la informații cu privire la evenimente sau la biografiile sportivilor prin intermediul unui touch-screen color. După îmbunătățire ca

viteză și volum- față de precedenta Olimpiadă, la care comentatorii aveau acces numai la câteva canale de comunicație simultan.

Sistemul informațional pentru conducerea operațiunilor, bazat pe un computer AS/400 de putere medie și 350 de PS/2-uri legate în rețea, este utilizat de funcționarii olimpici pentru a gestiona cazarea, rezervarea de bilete, transportul și serviciile medicale.

Inovațiile sistemului informațional olimpic includ:

- disponibilitatea instantanee a datelor și rezultate-lor de la toate locurile de concurs.
- transmiterea imediată a rezultatelor neoficiale.
- identificarea automată a limbii în care dorește utilizatorul să primească informațiile.

IBM Eastern Europe

Acorn pune un RISC într-un notebook

Acorn Computers a anunțat primul său notebook, realizat în jurul procesorului RISC de 32 de biți ARM

care rulează la viteza de 12 mips. Firma căreia mașina are aceeași putere ca și Power Book 170 al lui Apple, dar la jumătate de preț.

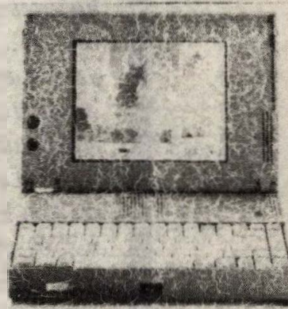
Modelul A4 de bază, care costa 1399 £ cu 2Mb RAM și disc flexibil de 2Mb și 1699 £ cu disc hard de 60Mb, se adresează utilizatorilor din învățământ și oa-

menilor de afaceri, precum și celor 200000 de utilizatori Acorn.

Notebook-ul utilizează sistemul de operare RISC OS3 al lui Acorn. Directorul de producție al companiei Heng Howarth, a confirmat că mașina va rula aplicații DOS sub emulare la o viteză care o face echivalentă cu un procesor 286.

"Mașina este capabilă să ruleze și aplicații Windows, dar performanțele ar putea fi mult mai bune", a adăugat el.

(Acorn)



HP pretinde că este pe primul loc

Un studiu - sponsorizat de HP - asupra siguranței în funcționare, efectuat în 1991, pretinde că PC-urile HP sînt de peste două ori mai sigure decît PC-urile IBM și de trei ori mai sigure decît cele Compaq. Studiul, efectuat în SUA de către International Data Corporation, a chestionat 500 de companii despre cinci furnizori de PC-uri, inclusiv AST și Dell.

Dell, clasat pe locul întii în ceea ce privește satisfacerea clienților într-un studiu efectuat de Data Pro/Computer Weekly în Marea Britanie și într-un studiu efectuat de JD Power în SUA, a fost sceptic cu privire la rezultate. "Oricît de departe ar merge studiile imparțiale și prestigioase despre satisfacția clienților, noi sîntem vîrf!" spune Maurice Cowley, directorul cu vînzările de la Dell.

Și IDC s-a interesat de satisfacerea clienților, realizînd un clasament cu șapte companii. Pe locul întii s-a situat HP cu punctajul 6,6 urmat de Compaq cu 6,3. Dell a venit pe ultimul loc cu 6,1. În schimb, Dell a ieșit pe primul loc la repararea PC-urilor cu o medie a ratei de returnare de 2,9 zile față de 3,6 zile pentru HP. Principala cauză a defectării a mașinilor o constituie harddiscul, 50% din reparațiile PC-urilor Compaq au fost generate de probleme cu harddiscul.

CA achiziționează Nantucket

Casa de soft Computer Associates International a anunțat că a încheiat achiziționarea lui Nantucket, firma care a elaborat sisteme de dezvoltare a aplicațiilor compatibile dBaSe/XBaSe.

Softul Nantucket va fi vîndut cu etichete CA. Computer Associates a spus că va lua măsuri de asigurare a continuității pentru calitatea asistenței tehnice, productivitatea salariaților și dezvoltarea produselor pe perioada integrării lui Nantucket în companie. Prețul pentru Clipper va rămîne 595 £.

(Computer Associates)

Criză de discuri flexibile

Un val imens de cerere de software a produs o criză de discuri flexibile de 3,5 in, criză ce ar putea să dureze și o bună parte a anului viitor.

Vina este dată pe lansarea simultană a lui Windows 3.1 și OS/2 2.0. Windows ocupă șapte dischete de 3,5 in, în timp ce OS/2 utilizează 21. În numai șase săptămîni s-au vîndut circa trei milioane de copii de Windows și o jumătate de milion de copii de OS/2.

Un purtător de cuvînt al lui Maxell, unul din cei mai mari furnizori de discuri flexibile, a confirmat criza, adăugînd că nu se așteaptă la o îmbunătățire a situației

generale pînă în anul viitor și că prețurile dischetelor vor crește semnificativ.

(PC Direct)

Cyrix lansează un procesor ce va face să scadă prețurile

Se estimează că prețurile PC-urilor se vor prăbuși odată cu lansarea clasei de procesoare Cyrix de 32 de biți pentru instrucțiuni 486 care vor constitui primul atac real la adresa dominării pieței de 486DX de către Intel.



Chip-ul 486DLC se poate instala pe o placă de bază standard pentru 386DX (cu cîteva modificări minore în BIOS și cîteva modificări hard care să permită 1K de memorie cache pentru procesor). Dacă producătorul dorește, se poate instala pe placa de bază și un soclu pentru un circuit de dublare a frecvenței de ceas. Noul 486DLC are bus de 32 de biți dar, spre deosebire de 486DX, nu are coprocesor matematic.

Procesorul Cyrix 486DLC de 25MHz și coprocesorul matematic însoțitor Cx87DLC se vor vinde cu 119\$, versiunile de 33MHz cu 155\$, iar versiunile 40MHz cu 199\$ (circa jumătate din prețul unui 486DX de 33MHz). Noul Cx486DLC lucrează la 3 Volți și este în întregime static.

Directorul cu marketingul pentru Europa de la Cyrix, Kelly Van De Hey, afirmă că chip-urile vor determina încheierea monopolului Intel asupra lui 486DX și-l vor determina pe gigant să-și reducă prețurile. El spune: "Între prețurile pentru 486SX și 486DX există o distanță prin care ar putea trece și un camion. Intel nu are de ales: fie scade prețurile, fie pierde o parte din piață".

Cyrix afirmă că Tondon, Tandy, Wyse, CompuAdd și Zeos intenționează să construiască mașini bazate pe noul Cx486DLC.

Borland aplică noua tehnologie WARP la Paradox 4.0

La conferința ținută în iunie la Marriot Hotel, în căldura toridă din Palm Desert, Philippe Kahn a făcut o demonstrație cu versiunea 4.0 a cunoscutului Paradox.

Încorporând ceea ce Borland numește noua tehnologie WARP (Wildly Accelerated Relational Performance), Paradox 4.0 este în prezent, produsul cu cea mai mare viteză de pe piață. În testele din cadrul demonstrației s-a dovedit a fi, în medie, de zece ori mai rapid decât predecesorul său (versiunea 3.5) și a reușit să depășească confortabil performanțele lui FoxPro 2.0, recenta achiziție a lui Microsoft.

Modificările cosmetice includ o interfață cu utilizatorul complet nouă, cu ferestre care pot fi dimensionate și poziționate în orice loc pe ecran. "Borland Desktop"-ul permite utilizatorului să vadă mai multe tabele de date simultan și, pentru prima dată, este incorporat suport complet pentru mouse.

Cu ajutorul mouse-ului utilizatorul poate apela meniuri pentru a vedea și selecta opțiuni ale programului, sau poate muta coloanele tabelelor.

Îmbunătățirea manipulării datelor include noile și mult doritele cîmpuri memo de lungime variabilă care pot conține pînă la 256Mb de informație pentru un

cîmp și maximum 4Gb pentru o tabelă. Cîmpurile memo pot fi create cu ușurință cu ajutorul unui editor încorporat, iar interogările și căutările se pot face pe toate cîmpurile memo.

Cei care folosesc procesoare de texte cum ar fi WordPerfect pentru crearea de documente se vor bucura aflînd că datele memo pot fi asociate unor programe externe. Fișierele create de alte programe pot fi folosite ca date, iar un dublu click pe datele aflate într-o tabelă încarcă și lansează editorul cu care au fost create.

Produsul poate manipula și noul tip de date cunoscut sub numele de Borland Blobs (Binary Large Objects) care poate conține orice tip de date binare cum ar fi imagini statice, animație și fișiere audio. Această caracteristică a fost demonstrată la conferință utilizînd imagini și sunet de la ultimele Jocuri Olimpice, smulgînd vii aplauze din partea publicului. Borland a

explicat că acestea au fost posibile datorită noii tehnologii Interbase, o nouă și puternică metodă de tratare a datelor complexe.

Facilitățile de lucru în rețea și de acces la date partajate au fost îmbunătățite cu o gestionare mai bună a încuierii fișierelor. Din nefericire, sistemul nu este compatibil cu versiunea precedentă.

Alte noi caracteristici include extinderi ale PAL-ului care permit utilizatorilor să creeze meniuri, ferestre de dialog, butoane de selecție și liste de selecție. Application Workshop permite crearea interactivă de meniuri și ecrane cu ajutorul mouse-ului.

Paradox 4.0 va fi disponibil la prețul 499,95\$. Un pachet LAN, care adaugă un utilizator suplimentar la rețea va costa 239,95\$. Utilizatorii înregistrați cu versiunea 2.0 sau superioară îl pot obține contra 129,95\$ iar cei cu "Borland Passport" cu 79,95\$. Versiunea runtime va costa 149,95\$.

La conferință a fost prezentat și un cod beta pentru Paradox pentru Windows care va fi disponibil în circa două luni.

(PC World - August 1992)

Dell oferă Novell 3.11

Dell a început să instaleze "din fabrică" Novel Netware 3.11 pe server-ele și pe stațiile sale de lucru utilizînd o tehnologie automatizată dezvoltată împreună cu gigantul softului de operare pentru rețele.

Capacitatea de integrare va reduce costurile suportate de clienți pentru integrarea în rețea și va îmbunătăți poziția vînzătorului direct în ceea ce privește furnizarea de soluții pentru file-server - este de părere Jeremy Welch, director de marketing pentru produse la Dell.

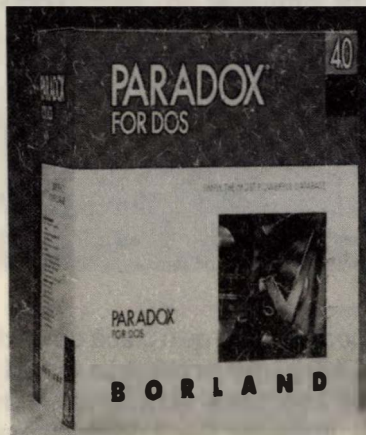
Aceasta este prima situație cînd NetWare este instalat automat, conform specificațiilor utilizatorului, înainte de livrare. "Acest fapt oferă o opțiune foarte atrăgătoare pentru utilizatori" a comentat Graeme Alan, director de marketing tehnic la Novell.

(PC Direct)

Viteză fără coprocesor

Firma Sigma Design a dezvoltat o nouă interfață VGA pentru calculatoare compatibile IBM PC. Noua placă sigma Legend GX este construită cu un nou chip accelerator S3-GUI care aduce pe ecran reprezentările grafice cu o viteză superioară circuitelor vga sau Super VGA. Astfel, noua interfață îmbunătățește performanțele aplicațiilor care utilizează GUI, cum ar fi Windows 3.0.

Interfața răspunde și cerințelor CAD oferind rezoluții de 1280x1024 și 1280x1024 puncte în 16 culori simultane.



Imprimantă Kyocera cu cilindru amorf

În sfârșit ceva nou în domeniul imprimantelor laser! Kyocer a elaborat o nouă metodă de construcție pentru cilindru unei imprimante, susținând că va reduce considerabil prețul listării unei pagini. Noua tehnologie pentru cilindru este ceramică - siliciu amorf, sau mai exact, a Si. Secretul costului scăzut per pagină constă



în faptul că un cilindru din a Si este extrem de durabil (durata sa de viață este aceeași cu a imprimantei).

ECOSYSFS-1500 (prima imprimantă care folosește a Si) are și alte avantaje: ea poate utiliza hârtie reciclată fără deteriorarea cilindrului și emite cantități foarte reduse de ozon.

Din nefericire, Kyocera nu va dezvălui tehnologia altor fabricanți înainte de a obține maximum de beneficiu. Ceea ce ar fi în regulă dacă FS-1500 ar fi o imprimantă grozavă; dar nu prea pare să fie.

Mai întâi părțile bune. Poate lista 10 pagini pe minut, ceea ce este rezonabil și suportă emulare HP LaserJet III, HP-GL2 și HP7475A. Are doi conectori JEIDA pentru adăugare de plăci cu fonturi. Are un port serial și un port paralel (care pot recepționa date simultan) și poate să aibă opțional-interfață Apphetalk, Ethernet sau token ring.

Partea proastă este că prețul de 3450\$ și un procesor relativ slab: 68000 la 16 MHz cu maximum 5Mb de RAM. Singura opțiune PostScript este o dezvoltare ROM bazată pe o versiune Kyocera de PostScript Level 1. Opțiunea este relativ ieftină (620\$), dar nu include RAM suplimentar pe placă sau procesor suplimentar.

De asemenea, tehnologia unicat pentru cilindru leagă utilizatorul de cartușele de toner Kyocera.

Ethernet într-un singur chip

AMD a anunțat o implementare Ethernet într-o singură capsulă care poate fi montată direct pe placa de bază a unui PC. El va fi disponibil la un preț sub 25\$ pentru comenzi peste 10000 de bucăți. Se așteaptă apariția mașinilor echipate cu acest chip în trimestrul al treilea al anului.

Circuitul este compatibil cu cunoscutele plăci NE2100 și NE1500T și cu softurile de rețea Novell, LAN Manager, Banyan Vines LANtastic.

Conform afirmațiilor AMD, circuitul are un consum mai mic decât soluția cu placă de interfațare și este cu 20-30% mai rapid.

(AMD)

IBM face o surpriză detractorilor lui OS/2

IBM i-a surprins pe observatorii care i-au considerat nule șansele de a avea succes cu mult întârziat sistem de operare OS/2 2.0 anunțând că se așteaptă să vândă între două și patru milioane de exemplare ale programului pînă la sfârșitul lui 1992.

După cum susține IBM, 90% din vânzările deja comandate prin linia directă permanentă de vânzări din SUA au mers la utilizatori de Windows.

(OS/2 User Group)

Unitate de disc flexibil categoria ușoară

Noua unitate de disc flexibil produsă de Anca Associates este mică și ușoară - ea măsoară numai 50,8x70x12,7 mm și cîntărește numai 75g. Capacitatea de stocare este de 42,6Mb cu un timp mediu de acces la sector de 5,5 ms și cu un consum de 0,15W în Sleep mode.

Unitatea poate fi instalată într-un notebook, într-un PC uzual, într-o imprimantă, într-un aparat medical, etc. Acest drive Single Disk de 1,8 inch costă circa 400\$.

(Gigasys)

OS/2 2.0

A meritat așteptarea ...

În 1987, IBM anunța OS/2 ca platformă de elită pentru calculatoarele personale ale sfârșitului deceniului '80 și mai departe. Ca parte a arhitecturii aplicațiilor pentru sisteme, OS/2 a devenit elementul cheie al strategiei software generale a lui IBM.

În 1992, la apariția lui OS/2 versiunea 2.0, OS/2 rămâne piatra unghiulară a strategiei de software pentru desktop a lui IBM. Și aceasta deoarece numai OS/2 răspunde celor mai vaste cerințe ale momentului și va corespunde noilor cerințe ce vor surveni către mijlocul deceniului '90 și ulterior.

Președintele IBM-ului, Jack Kuehler, spunea cu ocazia anunțării OS/2-ului:

"Nevoia de OS/2 de piață rămâne mai mare ca niciodată. Pur și simplu, clienții noștri au nevoie de un sistem de operare care să le permită să investească cu încredere în viitoarele lor aplicații. OS/2 continuă să fie strategia pe termen lung a IBM-ului".

De ce OS/2?

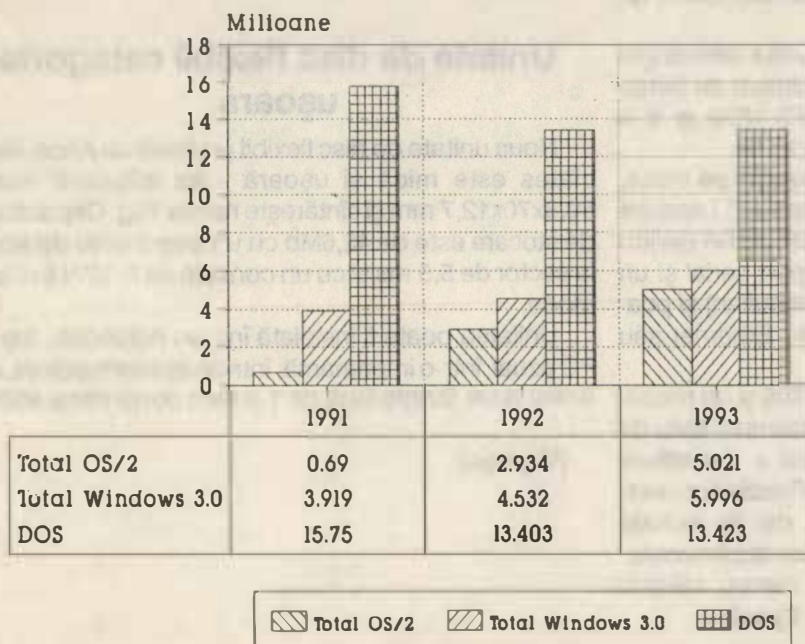
Cel mai popular sistem de operare pentru calculatoare personale din ziua de azi, DOS - care este în continuare un sistem de operare pe 16 biți, nu a fost proiectat să exploateze clasa de sisteme cu Intel 80386 și I486. DOS a fost proiectat pentru aplicații cu un singur utilizator și pentru interconectare limitată. Bazele proiectării sale nu mai corespund pentru următoarele motive:

- nu permite multitasking
- spațiu de memorie limitat
- nu are interfață grafică cu utilizatorul (fără Windows)
- integrarea în rețele locale și comunicarea cu calculatoare gazdă se bazează pe utilizări dincolo de posibilitățile sale.

MS-Windows versiunea 3.0 a extins funcțiile mediului DOS, depășindu-i unele limitări - mai ales în ceea ce privește asigurarea de memorie suplimentară și multitaskingul limitat pentru aplicații DOS. Totuși, Windows nu este un sistem de operare, ci o extensie construită deasupra lui DOS - deci are nevoie de DOS și este supus multor limitări ale acestuia. Deși Windows oferă un anumit grad de multitasking, aplicațiile DOS sau Windows care nu funcționează corect pot afecta alte aplicații aflate în curs de execuție sau chiar sistemul de operare. Aceasta poate impune o nouă lansare a sistemului adesea cu pierderea datelor.

Deci nici DOS, nici Windows - în forma lor actuală - nu pot răspunde cerințelor anilor '90. Este nevoie de proiectare nouă. Un sistem de operare modern are protecție, multitasking și interconexiune incluse de la început nu adăugate ulterior.

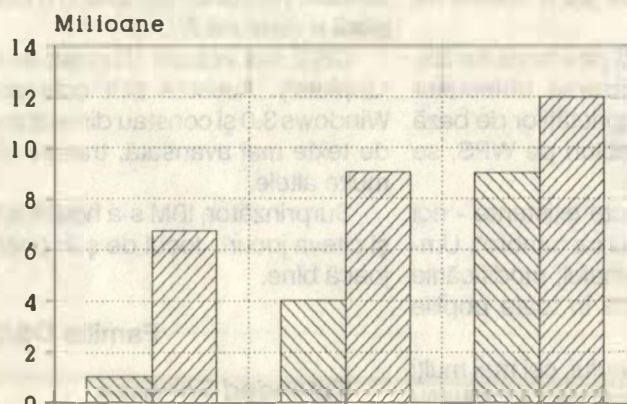
Vinzările de DOS, OS/2 și Windows 1991 - 1992



Ce este nou în OS/2 2.0?

Pentru început, s-ar părea că proiectanții lui OS/2 2.0 au ținut cont, cu adevărat, de nevoile bene-

Instalări OS/2 în lume față de Windows



	1991	1992	1993
OS/2 Instalat	1.082	4.016	9.037
Windows 3.x Instalat	6.719	9.064	11.992

▨ OS/2 Instalat ▨ Windows 3.x Instalat

ficiarilor. IBM a realizat aceasta elaborând versiuni preliminare ale produsului; acestea au fost disponibile aproape pentru oricine le-a vrut - în total peste 130 000. Nu s-a mai întâmplat pînă acum ca dezvoltarea unui produs să fie obiectul unui asemenea scrutin popular.

Pentru fiecare nouă versiune, IBM a "ascultat" reacția venită nu numai din sursele oficiale, ci și din rețele de poștă electronică cum ar fi CompuServe și Fidonet. În loc să încerce să convingă utilizatorii să încerce un nou sistem de operare cu soft complet nou, IBM a învățat să adauge OS/2-ului caracteristici care să protejeze investițiile în software DOS și Windows. Pentru a ușura opțiunea, unele aplicații (nu toate) vor rula mai repede și mai fiabil decît sub sistemul de operare original.

OS/2 se instalează peste aproape orice versiune de DOS sau OS/2. Pentru codul de bază este necesar un spațiu de minimum 15Mb pe disc, crescînd la 30Mb dacă se selectează toate opțiunile. Nu este necesară formatarea discului decît dacă doriți să beneficiați de avantajele oferite de High Performance File System (HPFS) sau de Boot Manager.

Boot Manager-ul ocupă o partiție de 1Mb pe hard disk și permite utilizatorului să aleagă orice partiție care poate fi încărcată. Chiar OS/2 însuși poate fi instalat pe o partiție secundară, permițînd utilizarea partiției primare pentru DOS, OS/1.x sau chiar Unix.

Dacă 30Mb vi se pare un spațiu enorm pentru rularea unui sistem de operare, amintiți-vă că acesta oferă echivalentul pentru DOS, Windows 3.0, OS/2, help foarte complex, un limbaj de programare proce-

dural, 20 de mini-aplicații ("applets"), un sistem de gestiune a fișierelor, un procesor de texte plus un program pentru construit diagrame.

Cu toate acestea, instalarea completă durează circa 30 de minute, timp în care vi se oferă posibilitatea rulării unei introduceri ce acoperă conceptele de bază.

Odată ce ați instalat complet OS/2 2.0 vi se oferă opțiunea de a migra aplicații existente către noul Work Place Shell (WPS) al OS/2-ului. Chiar din faza beta, sînt corect detectate și oferite spre selectare cele mai multe aplicații de succes.

Work Place Shell

Diferența evidentă între OS/2 2.0 și versiunile anterioare este Work Place Shell. Este extrem de important să i se stăpînească conceptele de bază înainte de a încerc-

pe să se lucreze serios.

Schimbarea constă în trecerea de la un spațiu de lucru orientat pe aplicație (ca vechiul Desktop Manager de sub OS/2) către unul orientat obiect.

WPS este una din primele aplicații care aderă la standardul IBM Common User Access, deși nu este implementat în întregime. Sub WPS, orice ce se găsește pe ecran este un obiect. Dacă doriți să faceți ceva cu un obiect, îl selectați, apoi WPS oferă o listă de acțiuni posibile. Fiecare tip de obiect are propriul set de acțiuni asociate.

De exemplu, pentru a edita un fișier sub vechea paradigmă, trebuia să găsiți aplicația de editare, să o lansați, să-i cereți apoi editorului să deschidă fișierul dorit. Sub WPS, trebuie să găsiți fișierul de editat, să activați lista cu acțiunile disponibile și să selectați Edit. Sau, fișierul poate fi "tras" peste un editor dacă se află cumva unul la îndemînă în acel moment.

Chiar și culorile și fonturile sînt tratate ca obiecte sub WPS. Dacă doriți să schimbați fontul unei ferestre WPS, se trece iconul obiectului font dorit peste obiectul dorit și este "scăpat" acolo.

Dacă unele obiecte din WPS nu mai sînt necesare, este ușor să fie trase grupat sau individual peste obiectul tocător. Greșelile pot fi ușor corectate.

O altă diferență importantă între WPS și vechiul Desktop Manager constă în modul de tratare a obiectelor minimizezate. Sub Desktop Manager, atunci cînd se minimizează un obiect, el cade pe "masa de lucru" a lui Presentation Manager sau pe cea a lui Windows și rămîne acolo pînă cînd este selectat sau închis.

Sub WPS, obiectele pot fi configurate individual pentru o anumită acțiune. Ele pot reveni la pliantul "inițial"; pot fi plasate în Minimized Windows Viewer (un pliant specializat care conține numai obiectele minimizezate); sau pot fi plasate în partea de jos a "mesei de lucru".

Se pot adăuga noi obiecte la WPS pe maimulte căi. Cea mai uzuală metodă este utilizarea obiectelor șablon. Acestea conțin definițiile proprietăților de bază ale obiectelor. Dacă se trage un șablon pe WPS, se poate crea un obiect din acea clasă.

În alt mod, se pot crea din obiecte existente - noi obiecte ca entități independente sau ca "umbre". Umbrele au aceleași proprietăți ca originalul; modificările făcute asupra originalului se reflectă în toate copiile umbră.

În pofida tuturor avantajelor WPS-ului, cei mai mulți utilizatori vor fi, probabil, mai interesați de capacitatea OS/2-ului de a rula aplicații DOS și Windows existente. Afirmația făcută de IBM că OS/2 2.0 este "DOS mai bun ca DOS, Windows mai bun ca Windows și OS/2 mai bun ca OS/2" stă în picioare și la o cercetare mai atentă.

OS/2 efectuează multitasking ca aplicații DOS și Windows (pentru mod real și standard) în sesiuni separate și protejate. Aplicațiile Windows pentru modul enhanced pot rula dacă sînt cu adevărat DPML (interfață cu modul DOS protejat).

Alte proprietăți ale modului Windows enhanced (multitasking DOS și paginarea memoriei) sînt asigurate direct de OS/2. Este asigurat serviciul standard DDE (schimb dinamic de date), deși numai între aplicații Windows.

Spre deosebire de Windows 3.0, chiar după încărcarea driver-elor pentru rețea, se pot pune la dispoziție 712Kb de memorie pentru fiecare sesiune DOS virtuală. Sesiunile DOS individuale au peste 40 de parametri care pot fi configurați de utilizator - acoperind toate aspectele privind memoria, driver-ele de dispozitiv, reîmprospătarea video, etc. De asemenea, fiecărei sesiuni i se oferă pînă la 32Mb EMS, 16Mb XMS și/sau 512Kb de DPML. Aplicațiile care rulează în ferestre DOS pot folosi moduri text sau moduri grafice pînă la rezoluția driver-ului de dispozitiv video.

Mai mult, în rarele cazuri cînd emularea DOS făcută de OS/2 nu este adecvată, utilizatorilor li se oferă posibilitatea de a crea imagini care să poată fi încărcate pentru orice sistem de operare pentru modul real. Este posibil să se ruleze simultan sesiuni separate cu DR DOS 6.0, CP/M-86, DOS cu Desquiew, DOS 3.3, DOS 4.0 și DOS 5.0. Se pot rula 240 de sesiuni DOS independente.

Suportul pentru DOS și Windows este mai mult decît satisfăcător pentru a putea considera OS/2 2.0 o platformă utilizabilă. Singurul aspect îngrijorător îl constituie viteza, mai ales cînd se rulează aplicații Windows.

Ceea ce este, poate, mai îngrijorător este faptul că mult-așteptatul instrument grafic pe 32 de biți pentru

OS/2 2.0 nu a apărut la această versiune. De această secțiune a OS/2-ului s-a ocupat partenerul strategic al lui IBM, Micrografx, care a avut întîzieri serioase. Avînd în vedere că conducerea IBM nu va mai accepta amînări, ne putem aștepta la o lansare destul de apropiată a versiunii 2.1.

OS/2 mai include și un set de mini-aplicații (numite "applets"). Acestea sînt echivalentul accesoriilor lui Windows 3.0 și constau din editare de texte, procesare de texte mai avansată, trasare de grafice, agende și multe altele.

Surprinzător, IBM s-a hotărît să adauge la standard și cîteva jocuri. Jocul de șah poate lucra și în rețea, și joacă bine.

Familia OS/2

Extended Services

Extended Services pentru OS/2 este un produs separat care oferă funcții de comunicație și de gestiune a bazelor de date. El include un spectru larg de protocoale (care pot fi toate active în același timp) și o puternică bază de date relațională client-server cu SQL. Extended Services poate lucra atît cu OS/2 Versiunea 1.3 (16 biți) cît și cu OS/2 versiunea 2.0 (32 biți).

În plus, Extended Services va rula pe calculatoare compatibile IBM PC, nu numai pe PS/2.

OS/2 LAN Server

OS/2 LAN Server este o platformă puternică ce asigură servicii LAN utilizatorilor de DOS, Windows și OS/2.

Ca și Extended Services, OS/2 LAN Server poate rula pe calculatoare compatibile IBM PC, ca și pe PS/2.

DACS/2

Distributed Database Connection Services/2 (DACS/2) este un produs complementar lui Extended Services. El oferă unui client OS/2 - conectarea Windows și OS/2 să aibă acces, lucrînd cu Database Manager, la baze de date de pe calculatoare gazdă în conformitate cu arhitectura bazelor de date distribuite (DRDA), care include DB2, SQL/DS și OS/400.

Provocarea OS/2

Pentru a face OS/2 să trăiască la adevăratele sale posibilități, IBM va trebui să umbrească nu numai strălucirea continuă a lui Microsoft Windows, ci și a altor rivali. La sfîrșitul lui ianuarie 1992, NeXT Computer anunța NeXTStep pentru calculatoare Intel 80486, SunSoft a anunțat Solaris iar membrii ACE Initiative se grupează în jurul lui Windows-NT. Constituie toate acestea o amenințare pentru OS/2? Examinînd alternativele anunțate, răspunsul este "NU" răsunător. Sistemul de operare NeXT va rula toate aplicațiile scrise pentru sistemele NeXT - NeXT Cube și NeXT Stations. Totuși, există foarte puține aplicații pentru NeXT care

să intereseze pe utilizatorii ne-tehnici. A doua mare problemă este că sistemul de operare NeXT costă de 8 ori mai mult ca OS/2. Sistemul de operare NeXT nu va fi disponibil înainte de trimestrul al 4-lea al lui 1992.

Sun Soft a intrat în luptă cu Solaris, un sistem de operare ce va rula atât pe Sun SPARC cât și pe procesoarele Intel de 32 de biți. Din nefericire, Sun a întârziat cu lansarea. În plus, Solaris este destinat vânzătorilor independenți de soft care caută un sistem de operare care să le permită să dezvolte atât pentru platformele RISC SPARC, cât și pentru cele CISC Intel de 32 de biți.

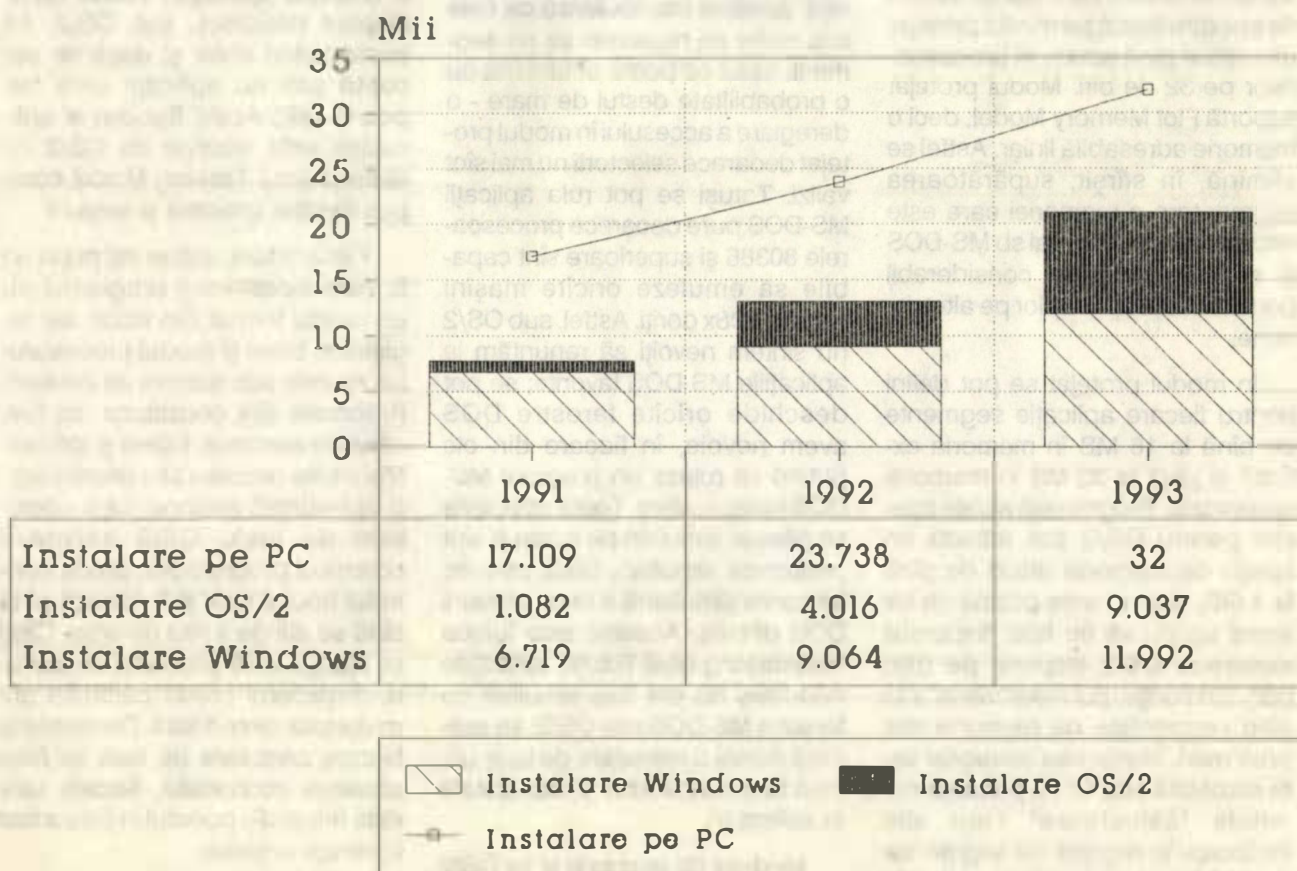
Cea de-a treia alternativă, inițiativa ACE cu Windows-NT și SCO, ar fi putut fi o amenințare formidabilă pentru OS/2, dar Digital Equipment a blocat progresul acestui grup. Când Digital a anunțat că arhitectura lor Alpha/RISC va înlocui sistemele VAX, majoritatea celor două mii de firme independente care dezvoltă soft pentru Digital și-au mutat interesul de la Windows-NT către OSF/1. Se pare că sistemul de operare strategic

pentru Digital și clienții săi nu este ACE și Windows-NT, ci arhitectura Alpha/RISC cu OSF/1.

Cu toată compatibilitatea aplicațiilor existente cu mediul OS/2 IBM trebuie să-i convingă pe vânzătorii independenți de soft să scrie aplicații pentru OS/2. Este interesant că unii din importanții vânzătorii de soft au promis suport. Printre aceștia se numără firme cunoscute ca Lotus, Micrografx, Borland, DeScribe. Pentru a urgenta aceste eforturi, IBM oferă "centre de portare" unde producătorii de soft intră cu aplicații DOS și/sau Windows și ies cu aplicații OS/2 2.0. Timpul mediu necesar pentru portare din DOS sau Windows a fost de trei săptămâni.

Cristian Nagy

Penetrația Windows și OS/2 în calculatoarele personale Intel pe 32 de biți 1991 - 1993



OS/2 Versiunea 2.0

Lupta pentru locul la fereastră

Versiunea 2.0 a sistemului de operare OS/2 produs de IBM a apărut în aprilie 1992. Această versiune pe 32 de biți oferă facilități de multitasking nemaiîntlnite în lumea PC-urilor. Lupta dintre IBM și Microsoft pentru sistemul de operare al viitorului continuă cu și mai multă violență.

OS/2 a fost dezvoltat de IBM pentru a satisface cerințele în continuă creștere ale hardware-ului. MS-DOS suferă de prea multe limitări - nu permite multitasking, nu suportă memorie peste limita de 1 MByte, etc. Pentru a le elimina, specialiștii de la IBM au creat un sistem de operare bazat pe modul protejat (neutilizat pînă acum) al procesoarelor pe 32 de biți. Modul protejat suportă Flat Memory Model, deci o memorie adresabilă liniar. Astfel se elimină, în sfîrșit, supărătoarea segmentare a memoriei care este necesară în modul real sb MS-DOS și se îmbunătățește considerabil portabilitatea aplicațiilor pe alte sisteme.

În modul protejat se pot defini pentru fiecare aplicație segmente de pînă la 16 MB în memoria extinsă și pînă la 32 MB în memoria expandată. Programele scrise special pentru OS/2 pot adresa un spațiu de memorie virtual de pînă la 4 GB, dar nu este posibil ca tot acest spațiu să fie fizic disponibil deoarece OS/2 depune pe disc părți din conținutul memoriei atunci cînd necesitățile de memorie sînt prea mari. Împărțirea memoriei este rezolvată sub OS/2 prin așa-numitele "selectoare" care sînt încărcate în regiștrii de segmentare. Selectoarele sînt puse în legătură cu o structură de date -

numită descriptor de segment - care conține informații despre segmente (adresa fizică, mărimea, scopul utilizării, nivelele de prioritate, ș.a.) Programele "văd" numai selectorul, deci nu știu, de exemplu, unde anume se află amplasate în memorie la un anumit moment. Descriptorii de segment dînt grupați în tabele de descriptori: cea globală, cea locală și cea de întrepreri. Acest procedeu permite sistemului de operare să mute, la nevoie, un program într-o altă zonă de memorie.

Aici apare o deosebire esențială față de aplicațiile MS-DOS în mod real: acestea manipulează de cele mai multe ori registrele lor de segment, ceea ce poate determina cu o probabilitate destul de mare - o dereglare a accesului în modul protejat deoarece selectorii nu mai sînt valizi. Totuși se pot rula aplicații MS-DOS pure deoarece procesoarele 80386 și superioare sînt capabile să emuleze oricîte mașini virtuale 808x doriți. Astfel, sub OS/2 nu sîntem nevoiți să renunțăm la aplicațiile MS-DOS favorite; se pot deschide oricîte ferestre DOS avem nevoie, în fiecare din ele putînd să ruleze un program MS-DOS independent. Toate aplicațiile se găsesc simultan pe ecran și sînt prelucrate simultan. OS/2 permite lansarea simultană a unor versiuni DOS diferite. Aceasta este lumea multitasking-ului! Totuși, aplicațiile Windows nu pot rula simultan cu ferestre MS-DOS sau OS/2; se execută numai o comutare de task-uri, însă fără cunoscutul "Eroare fatală în aplicație".

Modelul de memorie al lui OS/2 pune ordine și în dereglările de ac-

ces care apar prin concurența diverselor aplicații (cazul programelor TSR (Terminate and Stay Resident) de sub MS-DOS): permite fiecărei aplicații să aibă acces numai în zona de memorie care i-a fost alocată. Pentru platforme multitasking serioase acest lucru este indispensabil. Să presupunem că, în timp ce tipărim un document de sub un procesor de texte, un program TSR transmite - aceleași imprimante o altă sarcină de tipărire. Cum poate stabili imprimanta care date au fost trimise de o anumită tipărire. Cum poate stabili imprimanta care date au fost trimise de o anumită aplicație? Acest lucru dispare deoarece, sub OS/2, se poate hotărî chiar și dacă se acceptă sau nu aplicații care "se poartă urît". Acest Babilon al aplicațiilor este rezolvat de OS/2 cu ajutorul unui Tasking Model compus din fire, procese și sesiuni.

Fiecare task deține cel puțin un fir care înzestrează programul cu un mediu format din valori ale regiștrilor, stivei și modul procesorului, reunite sub numele de context. Procesele sînt constituite din fire, zone de memorie, fișiere și drivere. Mai multe procese sînt reunite într-o așa-numită sesiune. La o comutare de task, OS/2 salvează contextul programului, predă controlul noului task și îl rulează pînă cînd se atinge limita de timp. Cînd se ajunge la sfîrșitul listei de taskuri, dispecherul predă controlul primului task dintr-o listă. Deoarece la fiecare comutare de task se face slavarea contextului, fiecare task este reluat din punctul în care a fost întrerupt anterior.

Sincronizarea resurselor este realizată de OS/2 prin intermediul unui dispecer de eşantioane de timp bazat pe o listă de priorități, care atribuie tuturor firelor din sistem un eşantion de timp prestabilit; astfel este exclusă posibilitatea unui blocaj al sistemului (dead lock). Dacă - de exemplu - un fir actualizează, la un anumit moment, o structură de date, el nu poate fi întrerupt în acel moment.

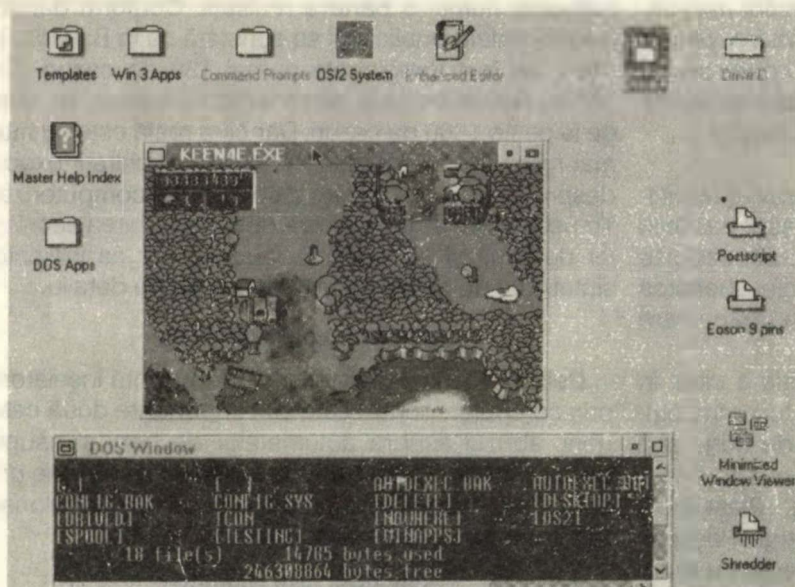
În acest scop, OS/2 atribuie o structură specială de date semafor - unui singur fir; semaforul este ce-

re/citare secvențială în/dintr-un alt proces) sau prin intermediul cozilor de așteptare (datele pot fi citite în ordinea dorită).

Ca orice sistem de operare, OS/2 trebuie, mai întâi, să citească programele de pe disc. Această funcție este realizată de o componentă a nucleului: încărcătorul. Atunci când încărcătorul primește un apel de funcție _EXEC, el generează un proces și construiește structurile de date pentru gestiunea memoriei. Spre deosebire de legătura statică, în care fișierul

izolare a taskurilor în modul protejat al procesoarelor 80X86. Date și coduri se află la baza unui model de priorități cu patru nivele. Cel mai înalt nivel de prioritate (nivelul 0) îl primesc, de obicei, numai funcțiile de nucleu. Astfel, ele au acces la toate datele din sistem. Pe nivelul 1 rulează componentele care asigură servicii de sistem, cum ar fi driver-e de dispozitiv, sistemul de gestiune a fișierelor, etc. Ele au acces la zonele utilizator, dar nu și la nucleu. Sub sisteme cum ar fi tastatura, mouse-ul sau subsistemul video rulează pe nivelul 2. În sfârșit, nivelul 3 rulează programele utilizator.

Din punct de vedere al aspectului, OS/2 oferă o suprafață utilizator grafică asemănătoare cu Presentation Manager, completată în versiunea 2.0 - cu o suprafață utilizator orientată obiect numită "Workspace Shell". Aceasta permite tipărirea unui document prin plasarea cursorului mouse-ului, cursorul este înlocuit cu simbolul ales. Ținând tasta apăsată, se "trage" simbolul documentului peste simbolul imprimantei și se eliberează tasta mouse-ului. Acum va începe tipărirea documentului. Acest procedeu este denumit "Drag and drop". La fel se poate proceda și pentru copierea sau ștergerea fișierelor.



Fereastră DOS în care rulează un joc pe VGA

rut de fir atunci când el face un acces la memorie. OS/2 permite accesul la memorie unui alt fir numai în momentul în care este eliberat semaforul.

Comunicarea între procese este asigurată prin fanioane (flags), cozi de așteptare (queues) și canale de comunicație (pipes). Cu ajutorul unui fanion un proces semnalizează producerea unui anumit eveniment. Alte procese pot să citească acest fanion cu ajutorul unei funcții OS/2. Procesele pot transmite date prin intermediul canalelor de comunicație (scrie-

.EXE trebuie să conțină valori pentru toți regiștrii segment, OS/2 permite referințe între segmente care sînt rezolvate numai în momentul execuției. Programul executabil conține numai o referință ; a modul care se găsește în Dynamic Link Library (DLL). Încărcătorul testează dacă această DLL se află deja în memorie și, în caz contrar, o încarcă.

Fiind un sistem de operare multitasking, OS/2 trebuie să împiedice "jenarea" reciprocă a taskurilor. Această cerință este rezolvată utilizînd proprietățile de protecție și

(Traducerea și adaptarea:

ing. Mihai BEER)

Este evident că, în viața noastră de zi cu zi, calculatoarele ocupă un loc din ce în ce mai important. În unele domenii activitatea nu mai poate fi concepută fără ele. Tot mai multe meserii și ramuri de activitate devin mai mult sau mai puțin "computer aided..." Și astfel am ajuns să avem CAD, CAM, CAE, CASE, etc.

De data aceasta vă propunem un domeniu ceva mai nou și care, mai ales în ultima vreme, începe să se bucure de atenție deosebită și din partea farurilor legislative. Este vorba de Computer Aided Crime (CAC) pentru care nu avem - deocamdată! - un echivalent în limba română. Ce ați spune de ceva în genul "delicvența asistată de calculator" ?

În legătură cu acest subiect vă prezentăm un articol publicat în numărul 8/1992 al revistei germane DOS International.

Mașina de minciuni PC

PC-uri pentru spargerea automatelor de jocuri, pentru falsificarea de dovezi și cifrarea spargerilor mafioate, pentru teletransmisie de date pentru tănuire, pentru pornografie infantilă, pentru comanda de organe pentru transplant, computere din sisteme de pază manipulate - toate acestea cer atenție din partea autorităților.

Datele săpate în plăci de piatră au supraviețuit mileniilor. Prelucrarea automată a datelor "mișcă" datele în jurul pământului. Oare manipulările și abuzurile săvârșite în acest domeniu pot să amenințe bunurile, libertatea individuală și statul de drept? Cum pot fi prinși cei care comit CAC?

Prin noua piesă a codului penal, justiția a intrat în posesia unui instrument deosebit pentru lupta împotriva abuzurilor prin computer. Dar, oricât de bune ar fi legile, ele ajută prea puțin dacă organele economice și cele de cercetare penală nu se pot impune. Piața liberă nu se prea supune prevederilor acestor legi. "Poșta de spărgători" le ia chiar în deridere: "spargerea cu ajutorul computerului este acum interzisă, ca și furtul prin copiere. Dar asta nu deranjează pe nimeni. Numai că pericolul nu constă în faptul că acum fiecare mic spărgător are în spate întreaga putere a statului concentrată ca într-un cartuș, ci în imposibilitatea aplicării legilor de către poliție care este slab pregătită și puțin solicitată..."

CAC într-un sens mai restrâns

În legile împotriva infracțiunilor săvârșite cu ajutorul computerului este vorba, în esență, de prevederi privind pătrunderea în baze de date, înșelătoria și sabotajul, precum și pirateria software care este reglementată de legea copy-right-ului. Dar aceste delictе, care caracterizează criminalitatea prin computer în sens restrâns, constituie partea cea mai redusă. Ca piraterie software se consideră folosirea nepermisă și utilizarea fără drept de autor a soft-ului. Cine ascultă plîngerile despre pagubele enorme provocate prin copiere ilegală ar putea crede că cifrele vor exploda. Dar

statistica criminalității redă, și în domeniul pirateriei software, numai o parte a realității. Conform datelor statistice date publicității se constată că în Bavaria, în 1991, au fost înregistrate numai 130 de cazuri. Extinzînd această cifră la întreg teritoriul federal, se ajunge la peste 1500 de cazuri. Dar cifra reală este cu mult mai mare. De asemenea, nu există declarații exacte despre aria de cuprindere a spionajului computerizat. Totuși statistica conține cifre despre trădarea secretelor de firmă și ale celor de afaceri, dar ca indicator sintetic ce nu poate permite exprimări de detaliu.

Înșelătoria prin computer

Delictul esențial din statistici îl reprezintă înșelătoria prin computer, pentru care sînt importante două categorii: abuzul asupra automatelor de bani și asupra celor de jocuri. Toate celelalte forme de înșelătorie prin computer, despre care circulă zvonuri spectaculoase, sînt conform cifrelor neimportante.

Sabotajul prin computer

Și statisticile despre sabotajele prin computer arată cît de mare este cifra "la negru". Dacă vom compara numai numărul mare de viruși răspîndiți cu delictеle de sabotaj prin computer, vom avea oglinda parțială a realității. Cauzele se regăsesc, desigur, și în lipsa de entuziasm a firmelor păgubite de a face publice informațiile.

În afara statisticilor oficiale asupra criminalității, cele mai multe și mai importante din aceste delictе apar și într-o statistică internă a "serviciului de informații asupra criminalității prin computer" al poliției.

CAC în sens mai larg

Începînd cu 1985, definiția CAC sună astfel: "Prin criminalitate prin computer se înțeleg toate faptele în care prelucrarea automată a datelor este un mijloc de acțiune și care motivează bănuirea unei fapte penale". Această definiție include atît delictеle criminalității prin calculator în sens restrâns cît și toate delictеle în care prelucrarea automată a datelor servește drept instru-

met - ca de exemplu tănuirea prin Mailbox-uri. Adversarii acestei definiții obiectează cum că, odată cu răspîndirea crescîndă a prelucrării automate a datelor, deliventa prin calculator ar putea deveni - conform acestei definiții - delictul cel mai frecvent înfîlînit în statisticile privind criminalitatea.

Dar exact această sentință va trebui să se impună! Aceasta cere funcționari criminaliști care să fie mai bine pregătiți în domeniul informaticii, precum și competențe polițienești în acest domeniu. Deci informatica ar trebui să devină materie de examen la facultățile care pregătesc cadre pentru poliție.

Profesor Klaus Brunnstein, Univesitatea din Hamburg specialist în informatică:

1. Societatea, economia și statul se bazează din ce în ce mai mult pe integrarea tehnicii de calcul în aplicații importante, dar fără a analiza riscurile. Între timp, prosperitatea sau ruina firmelor de stat sau particulare au devenit dependente de tehnica de calcul.

2. Sistemele de calcul actuale, în special PC-urile și sistemele UNIX, nu au fost proiectate avînd în vedere securitatea și controlul. Tineri de 16 pînă la 18 ani arată cum se poate abuza, în special prin PC-uri, sau cum pot fi induși în eroare utilizatorii prin viruși sau alte softuri. Răspîndirea sistemelor Mac-Intosh și Unix se confruntă, inevitabil, cu probleme asemănătoare.

3. În ciuda acestor pericole de "spargere" - chiar a rețelelor și a calculatoarelor mari - noțiunea de criminalitate asistată de calculator este nepotrivită, deoarece posibilitățile de abuz se găsesc, cel mult, în lipsa de securitate dată de chiar proiectarea tehnicii de calcul.

4. Cel puțin tinerii spărgători și autori de viruși nu au motive criminale: cum economia apreciază ca pozitivă construcția, programarea și introducerea PC-urilor, nu ne putem aștepta ca oameni tineri, fără etică sau cunoștințe de legislație, să-și considere preocupările "criminale".

5. În plus, sistemele de calcul răspîndite astăzi (PC-uri, rețele) nu conțin componente pentru păstrarea urmelor; sau, în cazul mainframe, ele nu sînt cuplate.

6. În sfîrșit, organele de cercetare (poliție, procuratură), ca și experții în drept (judecători, avocați) înțeleg foarte rar elementele criminale ale utilizării sistemelor de calcul. De exemplu, în procesul KGB-ului spărgătorii au fost nevoiți să explice judecătorului ce înseamnă "Email"; nu este de mirare că profesorul Stoll, un expert în domeniu, nu a putut să convingă completul de judecată cu expertiza sa.

Ținînd cont de nesocotința cu care statul, economia și societatea s-au făcut dependente de tehnică de calcul nesigură, abuzurile comise - în special de tineri - pot fi cu greu considerate drept criminale și, în plus, nu pot fi urmărite.

Cauzele diferențelor între cele două statistici se regăsesc în volumul mic de informații de care dispune poliția și în faptul că o mare parte a delictelor comise cu calculatorul nu sînt descoperite de poliție. Cel care analizează cifrele relativ scăzute din statisticile infracțiunilor asistate de calculator poate trage concluzia falsă că pericolul este supraapreciat. Aceasta deoare-

ce cea mai mare și mai periculoasă parte a acestor delikte, așa-numita criminalitate prin computer în sens larg, nu apare în statistici. Poliția - ca și publicul - nu sînt încă nici pe departe conștienți de pericol. Este vorba aici de traficul de stupefiante, de comerțul ilegal cu arme, de pornografia infantilă, de diverse forme de delikte economice și chiar de infracțiuni privind protecția mediului înconjurător. Din cauza acestui spectru foarte larg, termenul de "Computer Aided Crime" (CAC) a înlocuit - începînd cu sfîrșitul anului 1986 - denumirea de "criminalitate prin computer în sens redus și lărgit" pentru toate delicturile care folosesc prelucrarea automată a datelor ca instrument de acțiune. Abia după clarificarea noțiunilor se înlătură confuzia produsă de cele două statistici paralele ale poliției. Și aceasta deoarece pentru criminaliști și informaticieni CAC nu este altceva decît un domeniu particular de utilizare a calculatoarelor. În spiritul aceleiași viziuni par să acționeze și infractorii care înțeleg criminalitatea prin computer drept o "alternativă", cel mult ilegală, de folosire a prelucrării automate a datelor. Exemplele care urmează sînt atît de cunoscute încît direcționarea subiectului către aspectul ilegal al faptei scapă. Orice cunoscător a diversității lumii informaticii și, în plus, pe cea a criminalității și care mai adaugă și puțină fantezie își poate închipui diversitatea enormă a abuzurilor prin prelucrarea automată a datelor. Chiar "maniacii" calculatoarelor descoperă permanent ceva nou. Cei mai mulți dintre ei, chiar dacă nu iau în întregime în serios chestiunea furtului prin copiere a softului, sînt totuși cetățeni conștienți ai statului de drept și pregătiți pentru a sprijini poliția în lupta ei împotriva delicvenței chiar dacă este vorba de trafic de stupefiante, infracțiuni economice grave sau împotriva mediului înconjurător. Cea mai cunoscută aplicare a CAC este comerțul cu programe furate prin Mailbox. Numai în primii ani aceste programe au fost comercializate, în mare măsură, prin anunțuri în ziare. Cînd acest procedeu a devenit prea periculos, făptașii au început să folosească formulare pentru post-restant. În zilele noastre piraii deplasează softul prin Mailbox. Noua metodă se poate aplica și pentru bunuri furate, casete video cu pornografie infantilă sau casete video care glorifică violența. Transmisia de date la distanță prin Mailbox sau Btx reprezintă nu numai calea de comercializare a mărfurilor furate, ci și cele de comunicare între infractorii pe care poliția nu o mai poate controla. "Peștii mai mici" împart viruși prin Mailbox, unde cel care are box-ul poartă răspundea. Cît de devreme au abuzat infractorii de mediul Mailbox se poate vedea din faptul că, imediat după instalarea primelor automate de bani, prima indicație privind falsificarea Eurocheque-urilor s-a răspîndit printr-un Mailbox spune profesorul Brunnstein care susține că toate acestea ar putea duce - în viitor - la nenorociri neașteptate.

Oricum, virușii computerelor nu sabotează numai operarea calculatoarelor. Printr-o proiectare cores-

punzătoare a părții distructive pot fi realizate și delict de spionaj sau fapte ilegale de șantaj și constrângere.

Oricît s-ar proteja prelucrarea automată a datelor prin instrumente de securitate, publicul trebuie să se gîndească la faptul că, pe de altă parte, criminalii pot anula accesul poliției la datele care dovedesc efracția lor. Chiar criminalii dovedesc un interes mult mai mare pentru securitatea datelor decît firmele legale. Acest lucru este lesne de înțeles deoarece dacă un responsabil cu prelucrarea automată a datelor neglijează securitatea datelor riscă, în cel mai rău caz, o sancțiune profesională. Dimpotrivă, un infractor care nu-și protejează dovezile de ochii poliției socotește că va fi condamnat.

Credința în păstrarea adevărului prin documente scrise - "căci ceea ce deții scris negru pe alb poți să duci cu încredere acasă" - a fost zguduită de tehnica tipăririi. De cînd există PC-uri puternice, echipate cu scannere, imprimante laser și color și cu software grafic avansat, computerele pot produce "minciuni patentate". Aceasta deoarece documentelor falsificate - indiferent dacă sînt alb-negru sau color - le urmează fotografiile false. Pentru a obține astfel de succese, astăzi nu mai este necesar - ca înainte - un atelier de falsificat plus o pregătire specială. Prelucrarea numerică micșorează valoarea de dovadă a pozelor sau a negativelor. Digitizarea fotografiilor a două persoane cu ajutorul scannerului, schimbarea figurilor cu un editor grafic, depunerea rezultatului pe hîrtie sau pe dischetă și transmiterea rezultatului unui atelier pentru executarea unui diapozitiv sau a unui negativ color nu este o îndrăzneală chiar atît de mare. Un nou progres îl aduc aparatele foto cu schițare digitală.

Nu putem să închidem ochii în fața dezvoltărilor imaginabile. Computerul este un instrument de falsificare evident. După cum ne învață experiența, infractorii se inspiră fără rețineri din orice nouă dezvoltare hardware. Răufăcătorul care lucra, înainte, la un automat de jocuri cu sîrme, magneți sau burghie, își cumpără astăzi un laptop și, pentru sume între 5000 și 10000DM, programe de manipulare.

Jocul criminal de golire a automatelor arată foarte clar cum, prin computerizarea unei infracțiuni care înainte se făcea mecanic, se nasc noi pericole și cum PC-ul ușurează manipularea criminală. Deși pînă acum au fost prinși sute de făptași care golesc automatele, nu a fost încă prins nici un autor al softurilor de manipulare.

Exemplele de criminalitate împotriva mediului înconjurător arată pregnant multiplele fațete ale prelucrării automate a datelor. Pe de o parte, numai calculatoarele mai pot garanta controlul prescris al reglementărilor legale de supraveghere a aerului și apelor, în timp ce, pe de altă parte, deschid înșelăciunii toate căile. Este de la sine înțeles că aici lucrează interdisciplinar biologi, fizicieni, medici și informaticieni. Dar la funcționarii criminaliști care asigură probe efectuînd cercetări sau

la specialiști informaticieni care să le pregătească nu se gîndește aproape nimeni. Rămîne de netăgăduit faptul că nu există securitate șută la șută a prelucrării automate a datelor.

Gunter von Gravenreuth, avocat și inginer, specialist în procese de piraterie soft:

Chiar dacă pentru poliție s-au dus vremurile cînd dischetele erau găurite și apoi fixate ca să nu se piardă ca dovezi, totuși există și astăzi mari diferențe regionale în ceea ce privește clarificarea informatică a organelor de cercetare. Aici nu este vorba, precum s-ar crede, de diferența dintre sat și oraș. În timp ce procuratura din Munster, considerat a fi un oraș de provincie, obține rezultate de vîrf în ceea ce privește cercetările, procuratura din Frankfurt vorbea - chiar în octombrie 1990 - despre "un obiect plat de formă pătrată, încă nedescris exact, din material plastic tare", referindu-se la dischetele de 3,5 inch. CAC cuprinde astăzi toate domeniile vieții comerciale. Ea nu se amînestă numai în domeniul pirateriei soft. Astfel au fost, de exemplu, oferite măruri spre tănuire prin Mailbox. Accesibilitatea informaticii este folosită pentru șantaj. Înșelătorii prin computer sînt voalate; se dezvoltă un virus agresiv care devine activ numai pentru produsele concurenței.

Siguranța datelor este întotdeauna o problemă de compromis. Prin exagerarea măsurilor de siguranță cresc costurile care anulează avantajul introducerii tehnicii de calcul. În sens invers, trebuie să se țină cont cel puțin de condițiile de bază ale siguranței prelucrării automate a datelor. Din păcate, în ceea ce privește condițiile de bază, există goluri foarte mari. Nu-ți poți mustra secretara că își lipește parola sub tastatură dacă un mare comerciant de hardware din München livrează rețele la care parola supervisorului este tipărită pe etichete și beneficiarul nu este atenționat să o modifice.

Avea dreptate profesorul Nagel de la IBM care, acum 20 de ani, susținea că noțiunea de criminalitate prin computer este greșită pentru că discreditează informatica. Nici măcar inteligența artificială nu produce "computere criminale". Criminal poate fi numai omul, căci numai el convertește cele mai bune lucruri în arme.

Și totuși, nici un calculator nu garantează crima perfectă, dar unele ilegalități devin mai eficiente. În plus, computerul permite cifrarea informațiilor împiedicînd accesul organelor de cercetare la ele, posibilitate intens exploatată de infractorii.

Cei care, luînd în considerare statisticile, consideră infracțiunea asistată de calculator ca fiind inofensivă ar trebui să țină seama de faptul că, în continuare, computerul va preface toate formele de delicvență. Această evoluție nu afectează numai structura organelor de cercetare penală, ci atinge întreaga economie și chiar statul. Organizațiile criminale complexe nu pot fi imaginabile fără suportul informaticii. Poliția trebuie să țină cont de acest fapt în cercetările sale.

(Traducerea și adaptarea: ing. Mihai Beer)

**Trei nume cu rezonanta pe piata internationala,
reunite de un singur distribuitor:**

POLICOM S.A.

Produse oferite din stoc:

TOSHIBA

- copiatoare
- telefax-uri
- laptop-uri
- imprimante laser
- consumabile

stair 

- imprimante matriciale (A4 si A3)
- imprimante laser
- consumabile

Polaroid

- dischete 3,5" si 5,25"
- filtre pentru monitoare
- filme foto

Se asigura service in garantie si postgarantie.

Relatii suplimentare la magazinul din CLUJ

B-dul 22 decembrie nr. 135

tel. 95-156350 fax 95-152006 telex 31348

**Tot aici mai puteti gasi o gama completa de
calculatoare compatibile IBM-PC.**

INTERVIU

Bill Gates

Vă prezentăm câteva fragmente dintr-un interviu cu Bill Gates, președintele lui Microsoft Corporation. Interviul, publicat în numărul din aprilie 1992 al revistei "Upside", se referă la noul Windows NT, și conține interesante și incisive referiri la OS/2.

Rep.: Cum va fi posibil ca NT să nu "măcelărească" Windows 3.0 sau 4.1 sau orice altă versiune existentă la momentul apariției sale?

Gates: NT nu cere hardware suplimentar. Vom livra un NT care are nevoie de 8 megabytes. De fapt, ar putea rula cu 4, dar vrem să fim siguri că este loc suficient pentru ca cele mai puternice aplicații să poată rula. În plus, noi nu credem că NT va ajunge - la început - la un procent mare de utilizatori. Pentru cei mai mulți este suficient Windows peste DOS.

R: Cine vor fi primii utilizatori de NT?

G: În mod sigur, cei care lucrează pe server-e și mașini RISC. Cei care au cu adevărat nevoie de securitate. Mașinile multiprocesor se vor baza pe NT.

R: Ce scopuri urmăriți în legătură cu penetrarea lui NT pe piață?

R: Și apoi?

G: Apoi vom aplica strategia familiei din nou. NT va fi produsul "de jos" și vom avea lucruri uimitoare pe care le vom așeza "deasupra".

R: Microsoft ar putea dispune de o familie de sisteme de operare instalabile fără compromisuri pe o clasă în continuă diversificare de calculatoare, de la cele ce pot fi



William Gates III

ținute într-o mână la super-calculatoarele multiprocesor?

G: Ne-am hotărât să avem două implementări în ideea de a avea această elasticitate. Dacă vom avea nevoie de mai multe implementări, o vom putea face.

de obținerea de performanțe într-adevăr optimizate?

G: Nu, nu, nu! Scopul nostru este să nu existe nici un fel de renunțări. Atunci când introducem stil în Windows, cred că o facem mai bine decât Go. Băieții noștri sînt mai inventivi. Noi am făcut o treabă mai bună. În orice caz, noi beneficiem de influența lui "Hei, Windows este acest produs implementat masiv". Când un programator scrie o aplicație, are el vreun câștig? Eu cred că da. Trebuie să nu uităm că toate acestea sînt mijlocite de existența unor aplicații extraordinare și de economia softului.

R: Care ar fi aceasta?

G: Costuri fixe ridicate - 50 milioane \$ - pentru scrierea oricărei aplicații "mari". Apoi, mult mai mulți bani pentru creerea cererii și distribuției astfel încît să poată fi vîndută masiv.

R: Hai să vorbim despre sisteme de operare în general. Într-un interviu, Jim Manzi spunea că anii '90 vor fi cucerți cu aplicații, nu cu sisteme de operare. Are dreptate?

G: Nu știu exact ce a vrut să spună cu asta. Sistemele de operare joacă un rol mai important ca niciodată. De exemplu, facilitățile

Mașinile multiprocesor se vor baza pe NT

G: Pentru primul an, sau pentru primii doi, 15 sau 20 la sută din piață. În timp - dați-ne 4 ani - totul va trece către NT.

R: O formă a lui Windows pe orice... De fapt mizați pe faptul că beneficiarii vor fi tentați mai mult de protejarea investițiilor de soft decât

de transmitere a mesajelor vor fi incluse în sistemul de operare. Ideea de a coordona informații parțiate prin aplicații diferite nu se

poate regăsi decît în sistemul de operare.

R: Bine, dar Notes al lui Lotus face o mulțime din toate acestea, dar este o aplicație și nu un sistem de operare.

G: Așa este. Dar credeți că fiecare aplicație trebuie să inventeze tehnologii de securitate? Credeți că trebuie să existe seturi diferite de tehnologii de securitate administrate diferit pentru fiecare aplicație de pe sistem? Vreau să spun că, pentru o bună parte din ceea ce a făcut Notes, a trebuit să fie pionier într-o categorie de probleme pe care sistemul de operare nu le oferea. O mare parte din Notes - securitate, directoare și toate acestea vor fi incluse în sistemele noastre de operare și în cele ale lui Novell. Aceasta este schimbarea care vine. Pentru realizarea acestora, sistemul de operare este piesa centrală. Nu se poate ajunge la o acoperire mare a pieței pînă cînd nu vor fi incluse aceste facilități. Asemănător, uitați-vă la celelalte noutăți - suport audio, creioane, ecrane cu atingere, comunicație fără fir - toate sînt produse soft de sistem.

R: Ar putea deveni sistemele de operare cerințele anilor '90?

G: Dintotdeauna au fost nume-

bani. IBM a aruncat o grămadă de bani pe sisteme de operare.

R: Cam două miliarde pînă acum, nu?

G: IBM se situează într-o clasă aparte în ceea ce privește cheltuirea banilor în moduri ineficiente.

R: Revistele comerciale sînt pline de povești despre planul IBM-ului de a distribui OS/2 utilizatorilor de Windows și DOS. Crezi că IBM va încerca cu adevărat să o facă?

G: Deoarece ar trebui să-mi plătească un drept de autor, ar fi o manevră foarte scumpă - destul de stupid pentru ei și un chilipir pentru noi. Drepturile de autor ar trebui să fie semnificative. Dar chiar și așa, OS/2 este mare și lent și nu prea există aplicații pentru Presentation Manager.

R: Dar IBM trebuie să vadă o ocazie care merită să fie exploatată aproape la orice preț. Și anume ca OS/2 va ieși în martie 1992, iar NT numai către sfîrșitul acestui an.

G: Întotdeauna au avut o ocazie. Aceasta nu este prima versiune a lui OS/2; OS/2 a apărut cu mult timp în urmă. Toată lumea știe ce este OS/2. Credeți-mă nu este foarte deosebit. Cea mai mare parte a co-

G: Ceea ce utilizează oamenii astăzi este Windows. Aplicațiile se scriu pentru Windows. Și mai avem și alte aplicații, ceea ce constituie motivul pentru care se cumpără sisteme de operare. Cred că ar fi destul de bizar dacă OS/2 va dobîndi popularitate. Totuși, IBM îl poate băga pe gît unor indivizi.

R: Poate IBM să facă vreun fel de software?

G: Viitorul lor în soft este judecat cel mai bine de trecutul lor în soft. Trecutul lor în soft constă în TopView, OfficeVision, versiunea DOS 4.0 plină de erori și OS/2. Nu văd, în felul în care fac ei soft, nici o schimbare care să facă viitorul diferit de trecut.

Cred că ar fi destul de bizar dacă OS/2 va dobîndi popularitate. Totuși, IBM îl poate băga pe gît unor indivizi.

roase competiții. Ceea ce se întîmplă acum nu este o noutate. Oamenii tind să uite că, atunci cînd a fost introdus sistemul de operare, erau trei sisteme de operare. Și IBM era în spatele celor trei. Există o dinamică a unui sistem care tinde să o ia înainte. Este clar că aceștia sîntem noi. În mod sigur avem competitori care doresc să piardă mulți

dului este încă pe 16 biți. Nu este un sistem robust. Nu diferă chiar atît de mult de prima sa versiune. Dar au avut un avans ucigaș. OS/2 a fost anunțat în aprilie 1987. În curînd își va sărbători a cincea aniversare.

R: Există vreun scenariu IBM care să te îngrijoreze? Poate IBM să producă o erodare a pieței pentru voi?

Metoda Backtracking

0. Pretext

Deoarece multe probleme se pot rezolva elegant folosind metode și tehnici de programare adecvată, iar acestea nu sînt cunoscute de cei care se inițiază în programarea calculatoarelor prin prezentul articol doresc să pun la dispoziția cititorilor un material referitor la algoritmul Backtracking. De remarcat că mulți algoritmi recursivi pot fi transformați în algoritmi nerecursivi (uneori mai eficienți) rescriindu-i în spiritul acestei metode.

1. Punerea problemei

Fie n mulțimi finite oarecare M_1, \dots, M_n avînd p_1, \dots, p_n elemente:

$$(1) \quad |M_k| = p_k, k = 1, n.$$

Fără a restrînge generalitatea, putem considera că:

$$(2) \quad M_k = \{1, \dots, p_k\}, k = 1, n$$

deoarece putem eventual considera bijecțiile:

$$(3) \quad \Psi_k : \{1, \dots, p_k\} \rightarrow M_k, k = 1, n.$$

Vom nota cu:

$$(4) \quad M = M_1 X \dots X M_n$$

și în mod evident avem:

$$(5) \quad |M| = p_1 \dots p_n.$$

Vom numi soluție a problemei, un element al mulțimii

$$(6) \quad S = \{x \in M : \varphi(x) = 1\},$$

unde φ este un predicat n -ar:

$$(7) \quad \varphi : M \rightarrow \{0, 1\}.$$

Se pune problema determinării mulțimii S . O metodă simplă de determinare a mulțimii S este generarea tuturor elementelor mulțimii M (eventual cu ajutorul "principiului contorului") și verificarea pentru fiecare element a condiției $\varphi(x)$.

Această metodă însă nu convine, timpul de execuție fiind exponențial (de exemplu dacă $p_1 = \dots = p_n = 2$ algoritmul are timpul de lucru $O(2^n)$).

2. Rezolvarea problemei

Metoda Backtracking urmărește să evite neajunsurile expuse anterior. Pentru aceasta vom introduce pe lîngă condițiile interne legate prin predicatul φ și niște condiții de continuare φ_k astfel încît:

$$(8) \quad \varphi_k : M_1 X \dots X M_k \rightarrow \{0, 1\}$$

și:

$$(9) \quad \varphi_k(x_1, \dots, x_k) = 0 \text{ și } k < n \Rightarrow \\ \varphi_{k+1}(x_1, \dots, x_k, x) = 0$$

pentru $\forall x \in M_{k+1}$

adică φ_k este un predicat de aritate k , care are următoarele proprietăți: dacă k este fals pe un k -tuplu, atunci va fi fals pe orice $k+1$ -tuplu care are ca prime k componente pe componentele k -tuplului inițial.

În mod necesar avem următoarele condiții:

$$(10) \quad \varphi_n = \varphi$$

și:

$$(11) \quad \varphi_1 = 0$$

Să observăm că dacă

$$(12) \quad \varphi_k = 1, k = 1, n-1$$

vom genera mulțimea S prin parcurgerea întregii mulțimi M . De asemenea să observăm că dacă

$$(13) \quad \varphi = 1 \Leftrightarrow \varphi_k = 1 \forall k = 1, n$$

atunci mulțimea soluțiilor coincide cu mulțimea M , adică:

$$(14) \quad S = M \Leftrightarrow \varphi = 1$$

3. Algoritmul Backtracking

Predicatul φ_k introduse anterior ne sugerează construcția unei soluții

$$(15) \quad x \in S [x = (x_1, \dots, x_n), x_k \in M_k, k = 1, n]$$

prin determinarea succesivă a lor x_k . Dacă $\varphi_k(x_1, \dots, x_k) = 0$, atunci scădem pe k , altfel luăm la rînd acei $x_{k+1} \in M_{k+1}$ neverificați încă.

Dau în continuare sursa unui program Pascal și a unui C care calculează numărul de permutări și care determină permutările mulțimii $\{1, \dots, n\}$, unde $n \in \mathbb{N}^*$. De asemenea pentru amatorii de recursivitate ofer un al treilea listing care transformă acest algoritm într-un algoritm recursiv.

4. Probleme rezolvabile cu ajutorul metodei Backtracking

a. Generarea permutărilor mulțimii $\{1, \dots, n\}$.

Numim permutare a unei mulțimi

$$(16) \quad A_n = \{1, \dots, n\}$$

o aplicație bijectivă

$$(17) \quad \sigma : A_n \rightarrow A_n.$$

Teorema 1: $f : A \rightarrow A$, A finită, atunci:

$$(18) \quad f \text{ injectivă} \Leftrightarrow f \text{ surjectivă} \Leftrightarrow f \text{ bijectivă.}$$

Demonstrația: se găsește în [1].

Pe baza teoremei putem construi predicatul φ_k :

$$(19) \quad \varphi_k(x_1, \dots, x_k) =$$

$$= \bigwedge_{1 \leq i < j \leq k} (x_i < > x_j),$$

adică am pus condiția ca σ să fie injectivă. Se poate observa că:

$$(20) \quad \varphi_k(x_1, \dots, x_k) = \bigwedge_{i=1}^{k-1} (x_i < > x_{i+1})$$

Dar conform (9) putem simplifica predicatul φ_k :

$$(21) \quad \varphi_k(x_1, \dots, x_k) = \bigwedge_{i=1}^{k-1} (x_i < > x_k)$$

Putem scrie în Pascal o funcție de evaluare a predicatului:

function fi(k : integer; x:vector) : boolean;

```

var
  i:integer;
  b:boolean;
begin
  b := true;
  for i := 1 to k-1 do
    b := b and (x[k] < > x[i]);
  fi := b
end;
```

b. Problema celor n dame:

Să se așeze pe o tablă de șah cu latura de n pătrate, n dame astfel încât ele să nu se amenințe una pe alta.

Teorema 2: Numărul maxim de dame care se pot așeza este n.

Demonstrație: Presupunem că se pot așeza n+1 dame; tabla de șah avînd n linii, prin urmare există în mod evident o linie pe care se află cel puțin două dame, acestea amenințîndu-se reciproc. De fapt teorema este adevărată pentru n = 4.

Notînd cu x_k linia în care se afla dama (e numai una!) de pe coloana k putem scrie predicatul φ_k :

$$(22) \quad \varphi_k(x_1, \dots, x_k) = \bigwedge_{i=1}^{k-1} [(x_i < > x_k) \wedge (k-i < > |x_k - x_i|)]$$

Dar conform (9) putem rescrie predicatul φ_k :

$$(23) \quad \varphi_k(x_1, \dots, x_k) = \bigwedge_{i=1}^{k-1} [(x_i < > x_k) \wedge (k-i < > |x_k - x_i|)]$$

Am putea defini o funcție Pascal:

function fi(k:integer;x:vector):boolean;

```

var
  i:integer;
  b:boolean;
begin
  b := true;
  for i := 1 to k-1 do
    b := b and (x[k] < > x[i]) and
      (k-1 < > abs(x[k]-x[i]));
```

fi := b
end;

c. Colorarea grafurilor (vezi [2] cap. III, 2.3)

Numim graf neorientat

$$(24) \quad G = (X, U),$$

unde X se numește mulțimea vîrfurilor (nodurilor), iar

$$(25) \quad U \subseteq X^2$$

se numește mulțimea muchiilor. Pentru simplitate, dar fără a restrînge generalitatea putem considera:

$$(26) \quad X = \{1, \dots, n\},$$

deci:

$$(27) \quad |X| = n.$$

Se pune problema definirii unei funcții

$$(28) \quad f : X \rightarrow C$$

care atașează fiecărui nod $x \in X$ o culoare $c \in C$. Din nou fără a restrînge generalitatea putem considera:

$$(29) \quad C = \{1, \dots, m\}$$

și deci:

$$(30) \quad |C| = m.$$

Funcția f definită în (28) trebuie să satisfacă următoarea condiție:

$$(31) \quad (x, y) \in U \Rightarrow f(x) < > f(y)$$

adică două noduri legate printr-o muchie sînt colorate cu culori diferite. Numărul minim de culori cu care poate fi colorat un graf se numește număr cromatic și se notează cu $g(G)$, adică:

$$(32) \quad \gamma(G) = \inf\{m \in \mathbb{N}^+ \mid \exists f: X \rightarrow \{1, \dots, m\} \text{ și } (x, y) \in U \Rightarrow f(x) < > f(y)\}$$

În legătura cu colorarea grafurilor se cunoaște (vezi [3]) următoarea

Teorema 3: Pentru graf neorientat G care corespunde unei hărți avem

$$(33) \quad \gamma(G) \leq 4.$$

Demonstrația acestei teoreme este grea; de remarcă că pentru demonstrația completă s-a folosit și calculatorul.

Vom codifica o funcție-soluție în următorul mod:

$$(34) \quad f(i) = x_i,$$

deci o soluție se va prezenta sub forma unui n - tuplu $(x_1, \dots, x_n) \in C^n$.

Vom putea defini predicatul φ_k astfel:

$$(35) \quad \varphi_k(x_1, \dots, x_k) = \bigwedge_{1 \leq i < j \leq k} (x_i < > x_j) \wedge (x_i, x_j) \in U$$

sau ținînd cont de (9):

Fundamente

$$(36) \quad \varphi_k(x_1, \dots, x_k) = \bigwedge_{i=1}^{k-1} (x_i < > x_k) \\ (x,y) \in U$$

Pentru relația (36) putem scrie o funcție Pascal:

```
function fi(k:integer;x:vector):Boolean;
var
  i:integer;
  b:boolean;
begin
  b := true;
  for i := 1 to k-1 do
    if u(i,k) then
      b := b and (x[k] < > x[i]);
  fi := b
end;
```

Lăsăm pe seama cititorului găsirea unor predicate φ_k pentru următoarele probleme:

a. Realizarea tuturor circuitelor hamiltoniene cu calul pe tabla de șah.

b. Să se construiască toate turnurile formate din h cuburi (discuri) de latură (rază) li și de culoare ci astfel încât să nu stea un cub (disc) cu latura (raza) mai mare pe un cub (disc) cu latura (raza) mai mică și să nu stea alăturate două cuburi (discuri) cu o aceeași culoare.

c. Să se determine toate submulțimile M ale unei mulțimi finite a de numere reale strict pozitive ale elemente au o sumă dată S:

$$(37) \quad S = \sum_{x \in M} x$$

d. Să se genereze toate combinațiile și toate aranjamentele unei mulțimi finite $\{1, \dots, n\}$

5. Bibliografie:

- [1] Ministerul Educației și Învățământului - "Algebra. Manual pentru clasa a IX-a"
- [2] Leon Livovschi & Horia Georgescu - "Sinteza și analiza algoritmilor"
- [3] Ioan Tomescu - "Introducere în combinatorică"
- [4] Socaciu Ioan - Tiberiu - "Lucrări de laborator" (manuscris)

Tiberiu Socaciu

```
/* sursa in C */
/* program 1 */
#include <stdio.h>
#define max 10
int n,i,nr,x[max];
void prelucreaza(x)
int x[max];
{
  int i;
  nr ++;
  for (i = 1; i <= n; i++)
    printf("%d",x[i]);
  printf("\n");
}
int fi(k,x)
int k,x[max];
{
  int i,b;
  b = 1;
  for (i = 1; i < k; i++) b = b && (x[k] != x[i]);
return b;
}
void bt(void)
{
  void prelucreaza(int x[max]);
  int fi(int k,int x[max]);
  int inainte,k;
  k = 1;
  x[1] = 0;
  while (k > 0)
  {
    inainte = 0;
    while ((! inainte) && (x[k] < p[k]))
    {
      x[k] ++;
      inainte = fi(k,x);
    };
    if (inainte) if (k == n) prelucreaza(x);
    else
    {
      k ++;
      x[k] = 0;
    }
    else k--;
  }
}
void main(void)
{
  printf("\ndati valoarea lui n:");
  scanf("%d",&n);
  printf("\nvaloarea lui n este:%d\n",n);
  nr = 0;
  for (i = 1; i <= n; i++) p[i] = n;
  bt();
  printf("\nnumarul de permutari este:%d",nr);
}
```

```

{sursa in Pascal}
{ program 2 }
type
  vector = array [1..100] of integer;
var
  n,i:integer;
  x,p:vector;
  nr:integer;
procedure prelucreaza(x:vector);
var
  i:integer;
begin
  for i:= 1 to n do write(x[i],',');
  writeln;
  nr:= nr + 1
end;
procedure bt;
var
  inainte:boolean;
  k :integer;
function fi(k:integer;x:vector):boolean;
var
  i:integer;
  b:boolean;
begin
  b:= true;
  for i:= 1 to k-1 do
    b:= b and (x[k] < > x[i]);
    fi:= b
  end;
begin
  k:= 1;
  x[1]:= 0;
  while k > 0 do
    begin
      inainte:= false;
      while not inainte and (x[k] < > p[k]) do
        begin
          x[k]:= x[k] + 1;
          inainte:= fi(k,x)
        end;
      if inainte then if k = n then prelucreaza(x)
      else begin
        k:= k + 1;
        x[k]:= 0
      end
      else k:= k-1
    end
  end;
begin
  read(n);
  nr:= 0;
  for i:= 1 to n do p[i]:= n;

```

```

  bt;
  writeln(nr)
end.

[sursa in Pascal a versiunii recursive)
type
  vector = array [1..100] of integer;
var
  n,i:integer;
  x,p:vector;
  nr:integer;
procedure prelucreaza(x:vector);
var
  i:integer;
begin
  for i:= 1 to n do write(x[i],',');
  writeln;
  nr:= nr = 1
end;
procedure btr(k:integer;var x:vector);
var
  i:integer;
function fi(k:integer;x:vector):boolean;
var
  i:integer;
  b:boolean;
begin
  b:= true;
  for i:= 1 to k-1 do
    b:= b and (x[k] < > x[i]);
    fi:= b
  end;
begin
  if fi(k,x) then
    if k = n
      then prelucreaza(x)
      else for i:= 1 to p[k + 1] do
        begin
          x[k + 1]:= 1;
          btr(k + 1,x)
        end
    end;
begin
  read(i);
  nr:= 0;
  for i:= 1 to n do p[i]:= n;
  for i:= 1 to p[1] do
    begin
      x[1]:= i;
      btr(1,x)
    end;
  writeln(nr)
end.

```

Rețele Neuronale

O tehnică de explicare a procesului de luare a deciziilor într-o rețea neuronală.

Unul dintre punctele cheie în care rețelele neuronale diferă față de sistemele expert este modul în care acestea își structurează intern informația conținută. În sistemele expert tradiționale, informația este ținută sub formă de reguli. Când un sistem expert ajunge la o concluzie, se pot regăsi oricând în colecția lui de reguli acele reguli care au fost activate pentru a ajunge la concluzie, și în ce ordine. În cazul unei rețele neuronale însă, această informație nu este la fel de ușor de regăsit, din cauză că rețelele neuronale memorează informația sub formă de "accentuări" sau de "întărituri sinaptice". Accentuările într-o rețea au valori numerice în loc de simbolice, făcând astfel extrem de dificilă extragerea unei explicații simbolice.

Totuși, este posibil să cunoaștem mai mult despre modul cum afectează diverse intrări ieșirile unei rețele neuronale. Acest lucru este posibil prin aplicarea unei tehnici din domeniul analizei statistice neparametrizate, numită "analiză senzitivă".

Acest articol descrie analiza senzitivă și cum se aplică ea pentru a descrie "gîndirea" într-o rețea neuronală. Este descris în amănunțime un exemplu însoțit de programe scrise în "C". Exemplul este un procesor complet de rețele neuronale, implementînd algoritmul de propagare înapoi. Acest articol presupune că sînteți familiarizați cu rețelele neuronale și cu algoritmul de propagare înapoi (vezi articolele începute în numărul 3(6)/91 al revistei "if", despre rețele neuronale).

Ce este analiza senzitivă?

Analiza senzitivă urmărește la fiecare intrare modul în care o mică schimbare a ei se răsfrînge în ieșire. Dacă o mică schimbare a unei anumite intrări provoacă mari schimbări unei anumite ieșiri acea intrare poate fi considerată unul dintre factorii cheie ai rezultatelor curente.

Procesul de aplicare a analizei senzitive constă în pornirea cu prima intrare a rețelei și adăugarea la ea a unei mici cantități, recalculînd ieșirile, apoi scăderea unei mici cantități din intrarea rețelei și recalcularea ieșirilor, apoi considerarea diferenței dintre cele două ieșiri și raportarea ei la cantitatea totală de schimbare a intrării. De aici rezultă M valori, una pentru fiecare ieșire. Acest proces este repetat pentru fiecare dintre intrările rețelei. Rezultatul este o matrice $M \times N$ unde M

este numărul ieșirilor iar N numărul intrărilor. Mica modificare a intrării o vom numi "perturbație".

În plus, această tehnică de bază este utilizată intern în multe dintre rețelele neuronale pentru a determina care dintre elementele procesului trebuie modificat.

În particular, ea este utilizat în unele variante ale produsului MADALINE și în cîteva tehnici de optimizare a rețelelor.

Ca un exemplu despre cum acționează analiza senzitivă, să presupunem că o rețea neuronală a fost antrenată să determine credibilitatea unui individ. Intrările acestei rețele sînt informații de la o aplicație de credit incluzînd cîștigurile, cheltuielile, numărul de cărți de credit, raportul proprietar/chiriaș. Pentru individul

Intrare	Noua ieșire	Modificarea ieșirii
Venit +10%	0,60	+0,30
Venit -10%	0,15	-0,15
Cheltuieli -10%	0,25	-0,05
Cheltuieli + 10%	0,40	+0,10
Închiriere → proprietate	0,31	+0,01

Tabela 1: Efectul schimbării diferitelor variabile de intrare asupra ieșirii curente a unei rețele neuronale antrenată pentru aprobarea creditelor. Ieșirea nominală pentru setul curent de intrări este 0.3.

Scop	Cost în \$	Nivel de cumpărare	pentru 10% schimbare în preț	Senzitivitate
Educație	99	1.0	0.00	0.0
Educație	150	1.0	-0.20	-2.0
Educație	200	0.5	-0.10	-1.0
Educație	500	0.3	-0.05	-0.5
Educație	1895	0.1	-0.01	-0.1
Aplicație	99	0.1	+0.10	+1.0
Aplicație	500	0.5	+0.10	+1.0
Aplicație	1895	0.8	-0.01	-0.1
Aplicație	3000	0.7	-0.01	-0.1

Tabela 2: Schimbări în nivelul de cumpărare. Cît de mult doriți să vă cheltuiți banii depinde de scop și de preț. Este estimată schimbarea în nivelul de cumpărare. Observați că sensibilitatea la o intrare particulară (preț) depinde de alte intrări (scop). Senzitivitatea este procentul de schimbare în ieșire împărțit la procentul de schimbare în intrare.

curent, să presupunem că ieșirea rețelei este 0.3 (o valoare mai mică decât 0,5 înseamnă respingerea acordării creditului). Tabela 1 arată efectele unei mici schimbări în fiecare intrare asupra ieșirilor. Care dintre intrări va trebui să le schimbați puțin pentru a transforma respingerea în acceptare? Cea mai ușoară metodă va fi să cerem individului să-și crească câștigul cu 10% rezultând astfel o acceptare (0.6). Probabil că principala cauză a respingerii este nivelul prea mic de venit al solicitantului. Din tabela 1 rezultă că nu este aproape nici o diferență dacă persoana are o casă sau stă în chirie. Pe de altă parte motivul respingerii ar putea fi acela ca individul are cheltuieli prea mari. Dacă aceste cheltuieli ar fi reduse cu 20-30% acest lucru ar putea fi suficient pentru acceptare (ieșirea evaluatorului s-ar ridica peste pragul de 0.5.)

Este foarte important de văzut că modul în care fiecare dintre aceste intrări variabile afectează ieșirea este dependent de valorile lor curente. Pentru un alt individ, cel mai important factor ar putea fi statutul de proprietar sau chirieș.

În mai multe feluri, aceasta este analog cu modul în care indivizii iau decizii. Când e dificil de luat o decizie între două moduri de derulare a acțiunii, de obicei există un factor particular care este chiar la limita de separație (sau mai mulți). Dacă factorul ar înclina puțin într-o parte sau alta, decizia ar fi clară.

De exemplu, compania mea oferă un pachet soft introductiv pentru 99\$ iar un pachet soft sofisticat pentru 1895\$. Veți cheltui 99\$ pentru un pachet de învățare a rețelelor neuronale? Veți cumpăra fără să gândiți prea mult (nivel de cumpărare 1.0). Dar veți cheltui 1895\$? Aproape sigur nu, în afară de cazul în care aveți o aplicație specifică, și vreți să fiți sigur că aveți cele mai bune scule pentru a avea succesul asigurat (nivel de cumpărare educațional 0.1, nivel de cumpărare aplicațional 0.8). Ați cheltui 200\$? Poate (nivel de cumpărare 0.5). Dar 500\$? Nu, decât în cazul în care vi dorii foarte mult (nivel de cumpărare 0.3).

Astfel confruntat cu decizia despre cât de mult ați plăti pentru un pachet soft de învățare a rețelelor neuronale, cu cât este prețul mai mare, cu atât mai puțin vreți să îl aveți. Sub 150\$ îl cumpărați fără să vă gândiți. Peste această sumă aveți nevoie de motivații suplimentare. Dar dacă aveți o aplicație anume în minte, veți plăti 1895\$ bazat pe capacitățile dorite. Dar 3000\$? Veți plăti, bazat pe aceleași capacități (nivel de cumpărare 0.7). Dar 500\$? S-ar putea, dar v-ați uita mai atent la prezentarea produsului (nivel de cumpărare 0.5). De ce sînteți mai liniștiți cumpărînd un produs de 2000-3000\$ pentru a rezolva o anume aplicație? Pentru că știți din experiență că sculele bune costă bani, sculele ieftine cerînd de obicei un efort inutil mult mai mare decât banii economisiți (cu câteva excepții notabile).

Priviți acum cu atenție tabela 2. Observați că un produs educativ este foarte sensibil la preț cînd prețul urcă peste 150\$. Cînd ai nevoie de o sculă pentru

dezvoltarea unei aplicații, peste o anumită limită de preț, alți factori devin mai importanți decât prețul. Observația cheie din această tabelă este că modul în care prețul afectează dorința de a cumpăra este dependent de motivul pentru care vreau să cumpăr produsul. Acesta este adevărul analizei senzitive în general. Modificările rezultate în ieșire sînt adevărate numai pentru setul curent de intrări. Pentru că o mică schimbare într-o anumită intrare poate să nu aibă nici un efect la ieșire, nu putem trage concluzia că acea intrare este neimportantă. Combinată cu alte seturi de intrări, o mică schimbare în această intrare poate avea o influență puternică asupra ieșirii.

Programul exemplu

Listingul 1 conține un program pentru o rețea neuronală, proiectat pentru a antrena o rețea cu propagare înapoi și apoi să o poată explica folosind analiza senzitivă. O rutină principală selectează funcția dorită și permite ca rețeaua și explicațiile să fie solveate într-un fișier pentru o analiză ulterioară. Pentru că spațiul este critic rețeaua completă nu este dată împreună cu programul, ci trebuie antrenată. Programul a fost scris sub Turbo C versiunea 2.0 dar trebuie să meargă fără modificări și sub Microsoft C și sub Zortech C.

Programul Exemplu

Iată pașii pentru rularea programului exemplu. Următoarea secvență va crea network.exe și va lista tabelele arătate în acest articol.

1. Compilați sursa "network.C" (am utilizat Turbo C 2.0 dar veți putea utiliza Microsoft C sau Zortech C la fel de bine)

C > tca network.C

Dacă aveți coprocesor de virgulă mobilă, s-ar putea să fie mici diferențe față de acest articol datorită rotunjirilor diferite. Exemplul a fost creat inițial fără coprocesor.

2. Rulați programul Cnetwork.

3. Creați o nouă rețea: Ce doriți să alegeți? C rețeaua a fost creată.

4. Antrenați rețeaua (asta a luat 5 ore pe un Laser 386). Ce doriți să alegeți? t se afișează mesaje relative la pasul curent.

5. Salvați rețeaua în fișierul "network.net". Ce doriți să alegeți?

6. Tipăriți accentuările rețelei: Ce doriți să alegeți? pnetwork.est

7. Analizați cum este senzitivă rețeaua la intrări diverse: Ce doriți să alegeți? e network.exp

8. Leșiți: Ce doriți să alegeți? X

Programul este organizat astfel: Rutinele utilitare pentru încărcare, salvare, ștergere, creare și listare rețea sînt la început. Acestea se pot chema direct din programul principal. Rutina Recall() este utilizată pen-

tru a executa rețeaua o singură dată, pentru determinarea răspunsului său la un anumit set de intrări. Funcția Learn() execută o singură actualizare a accentuărilor din rețea. Pentru antrenarea accentuărilor se folosește metoda standard cu delta și moment (vezi cartea lui Rummelhart și McClelland *Parallel Distributed Processing*, vol 1, cap 8).

Funcția Train() aplică funcția Learn() în mod repetat, dîndu-i cîte un exemplu din setul de antrenament, test E (). Pentru a fi siguri că sînt folosite toate exemplele, am folosit o anumită strategie de amestecare, implementată în funcție NextTestN(), care creează o listă amestecată de exemple, o trimite spre programul se învățare și o reamestecă dacă stiva este goală. (vezi textul încasetat pentru detalii de rulare a programului. Au fost luate precauțiile necesare pentru ca programul să fie ușor de compilat sub alte compilatoare C).

Analiza rezultatelor

Rețeaua a fost antrenată pînă cînd eroarea a fost mai mică decît 0.001. Tabelul 4 (a) ne arată ieșirea rețelei peste intervalul intrărilor 1 și 2. Aceasta corespunde foarte bine antrenamentelor pe intrările test E(). Setul de antrenament a fost conceput astfel încît intrarea 3 să fie aleatoare. Dacă ea ar fi complet aleatoare, accentuările conectate la ea ar trebui să fie toate aproape de zero. Tabelul 3 este o listare a rețelei. Motivați că mărimea înregistrărilor asociate cu intrarea 3, subliniate în tabel, sînt aproape toate foarte mici în comparație cu accentuările celorlalte intrări la fiecare element de procesare (PE). Aceasta înseamnă că intrarea 3 va avea de obicei un mic efect în ieșirea rețelei. Elementul de procesare 0 este așa numitul "bias" și ieșirea lui este întotdeauna 1.0.

Tabelele 4(b)(c) și (d) ne arată sensibilitatea ieșirilor la schimbări mici în prima și a doua variabilă respectiv. Tabelele barează intrările posibile (intrarea 1 pe axa x și intrarea 2 pe y). Notați că ieșirea este cea mai sensibilă cînd o variabilă este mai aproape de margine. Dacă vă gîndiți un moment, asta are sens. Într-o regiune mai îndepărtată de margine, mici schimbări în intrări vor avea efect mic în ieșire. Doar aproape de margine, cînd ieșirea trece dintr-o stare în alta, o intrare potrivită va avea un efect puternic. Aceasta sugerează unele modificări în program. Anume, dacă o mică schimbare în fiecare intrare, produce puține, sau nici o schimbare a ieșirilor, să mărim schimbarea pînă efectele sînt semnificative.

În tabela 4(b), care privește sensibilitatea ieșirii la schimbări în intrarea 1, observați că ieșirea este mai sensibilă cînd intrarea 1 este în domeniul 0.45-0.55 și intrarea 2 este în domeniul 0.7-1.0. Asta este în locurile unde ieșirea rețelei face tranziția între 0 și 1. La fel în tabela 4(c) ieșirea este mai sensibilă la intrarea 2 cînd aceasta este între 0.25 - 0.35 și 0.65 - 0.8. Aici rețeaua

Layer 0 with 3 PEs						
1	labb:5b12	PE	Output=0.0	Error=0.0	WorkR=0.0	NConns=0
2	labb:5b32	PE	Output=0.0	Error=0.0	WorkR=0.0	NConns=0
3	labb:5b52	PE	Output=0.0	Error=0.0	WorkR=0.0	NConns=0

Layer 1 with 5 PEs						
4	labb:5b72	PE	Output=0.0	Error=0.0	WorkR=0.0	NConns=4
	Src=0	labb:00d6	Weight=	1.591	Delta Wt=	0.0
	Src=1	labb:5b12	Weight=	3.830	Delta Wt=	0.0
	Src=2	labb:5b32	Weight=	-16.104	Delta Wt=	0.0
	Src=3	labb:5b52	Weight=	0.486	Delta Wt=	0.0
5	labb:5bba	PE	Output=0.0	Error=0.0	Workr=0.0	NConns=4
	Src=0	labb:00d6	Weight=	14.082	Delta Wt=	0.0
	Src=0	labb:5b12	Weight=	-15.446	Delta Wt=	0.0
	Src=0	labb:5b32	Weight=	-16.720	Delta Wt=	0.0
	Src=0	labb:5b52	Weight=	2.472	Delta Wt=	0.0
6	labb:5c02	PE	Output=0.0	Error=0.0	Workr=0.0	NConns=4
	Src=0	labb:00d6	Weight=	3.883	Delta Wt=	0.0
	Src=0	labb:5b12	Weight=	2.332	Delta Wt=	0.0
	Src=0	labb:5b32	Weight=	-20.745	Delta Wt=	0.0
	Src=0	labb:5b52	Weight=	1.366	Delta Wt=	0.0
7	labb:5c4a	PE	Output=0.0	Error=0.0	Workr=0.0	NConns=4
	Src=0	labb:00d6	Weight=	-1.377	Delta Wt=	0.0
	Src=0	labb:5b12	Weight=	-3.519	Delta Wt=	0.0
	Src=0	labb:5b32	Weight=	-3.751	Delta Wt=	0.0
	Src=0	labb:5b52	Weight=	0.545	Delta Wt=	0.0
8	labb:5c92	PE	Output=0.0	Error=0.0	Workr=0.0	NConns=4
	Src=0	labb:00d6	Weight=	4.869	Delta Wt=	0.0
	Src=0	labb:5b12	Weight=	-10.058	Delta Wt=	0.0
	Src=0	labb:5b32	Weight=	-8.766	Delta Wt=	0.0
	Src=0	labb:5b52	Weight=	3.108	Delta Wt=	0.0

Layer 2 with 2 PEs						
9	labb:5cda	PE	Output=0.0	Error=0.0	Workr=0.0	NConns=9
	Src=0	labb:00d6	Weight=	-20.434	Delta Wt=	0.0
	Src=1	labb:5b12	Weight=	46.298	Delta Wt=	0.0
	Src=2	labb:5b32	Weight=	-2.895	Delta Wt=	0.0
	Src=3	labb:5b52	Weight=	-0.492	Delta Wt=	0.0
	Src=4	labb:5b72	Weight=	-21.351	Delta Wt=	0.0
	Src=5	labb:5bba	Weight=	20.966	Delta Wt=	0.0
	Src=6	labb:5c02	Weight=	-26.267	Delta Wt=	0.0
	Src=7	labb:5c4a	Weight=	1.304	Delta Wt=	0.0
	Src=8	labb:5c92	Weight=	8.782	Delta Wt=	0.0
10	labb:5d54	PE	Output=0.0	Error=0.0	Workr=0.0	NConns=9
	Src=0	labb:00d6	Weight=	20.434	Delta Wt=	0.0
	Src=1	labb:5b12	Weight=	-46.298	Delta Wt=	0.0
	Src=2	labb:5b32	Weight=	2.895	Delta Wt=	0.0
	Src=3	labb:5b52	Weight=	0.492	Delta Wt=	0.0
	Src=4	labb:5b72	Weight=	21.351	Delta Wt=	0.0
	Src=5	labb:5bba	Weight=	-20.866	Delta Wt=	0.0
	Src=6	labb:5c02	Weight=	26.287	Delta Wt=	0.0
	Src=7	labb:5c4a	Weight=	-1.292	Delta Wt=	0.0
	Src=8	labb:5c92	Weight=	-8.783	Delta Wt=	0.0

Tabela3: Accentuările în rețeaua antrenată. Rîndurile subliniate sînt accentuările corespunzătoare intrării 3. Intrarea 3 a fost proiectată mai mult sau mai puțin aleatoare. Ca un rezultat, accentuările asociate cu ea sînt în general mici relativ la celelalte accentuări pentru elementele de procesare (PE).

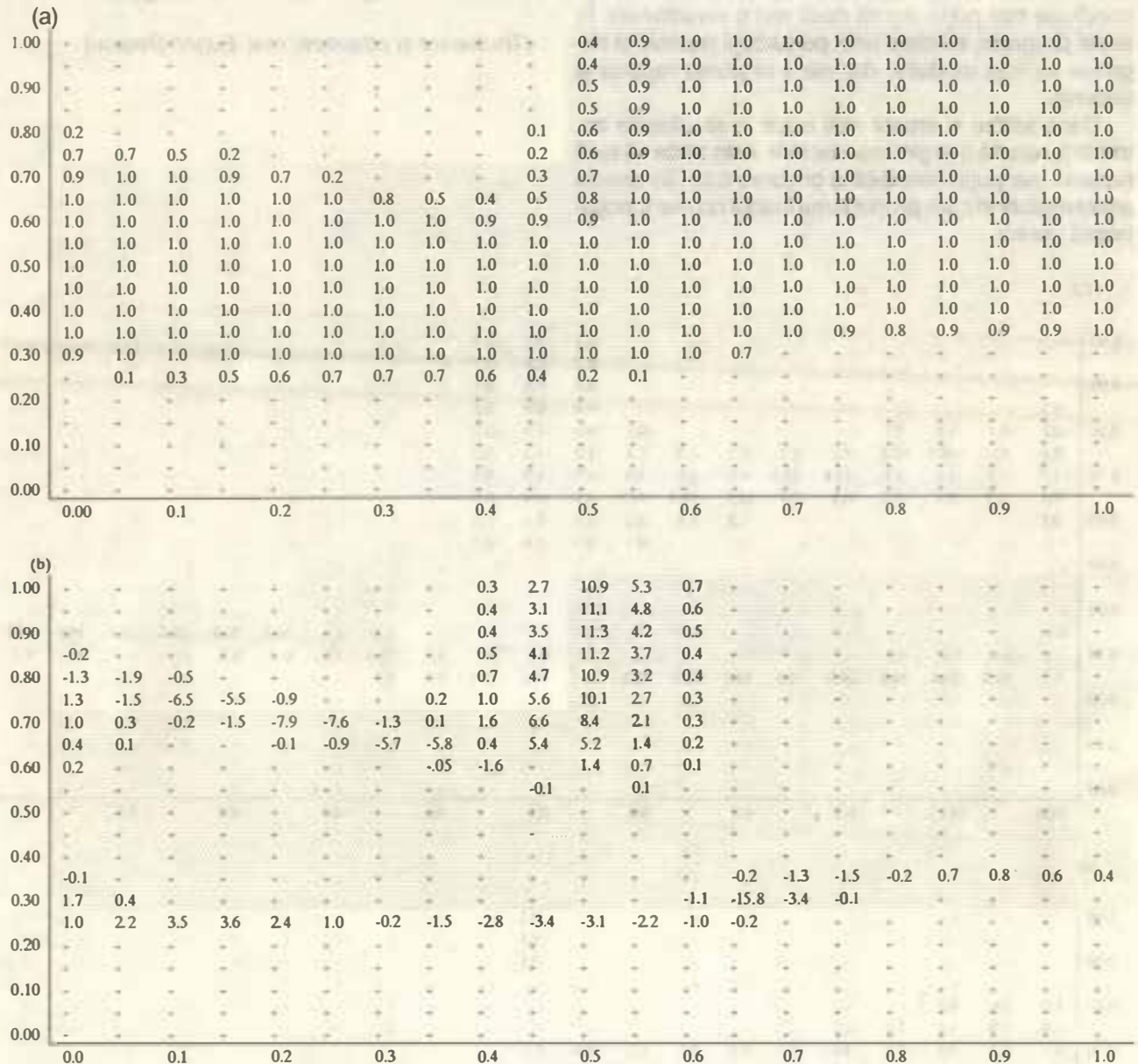
face tranziția între 0 și 1 respectiv 1 și 0, de-a lungul axei pentru intrarea 2.

Uitîndu-ne la tabelele 4(b)(c) și (d) observăm că ieșirea este întotdeauna sensibilă la intrare cînd la intrare se întîmplă o tranziție. De altfel exemplul are margini foarte clare și o singură variabilă se poate schimba la fiecare dintre ele. Asta vă poate face să suspectați un anumit fel de probleme. Problema apare datorită modului în care o rețea cu propagare înapoi își crează ieșirea. Ea ia o serie de ridicături și le potrivește

împreună pentru a construi o anumită stare a intrărilor și ieșirilor.

Priviți din nou spre exemplul despre acceptarea

situații e important să variem domeniul de intrare și să măsurăm schimbările în ieșire.



(Tabela 4 se continuă pe pagina următoare)

Tabela 4: (a) Ieșirea rețelei ca funcție de intrările 1 și 2 cu intrarea 3 ținută constant la 0.5. Observați că rețeaua a învățat configurația dorită foarte bine. (b) Rezultatul aplicării analizei senzitive pentru intrarea 1 peste domeniul valorilor pentru intrarea 1 și 2. Ieșirea este sensibilă la tranziție când intrarea 1 merge de la 0.45 la 0.55 (și ieșirea de la 0 la 1). (c) Rezultatul aplicării analizei senzitive pentru intrarea 2 peste domeniul valorilor intrărilor 1 și 2. Ieșirea este foarte sensibilă la două tranziții. Prima când intrarea 2 este între 0.25 și 0.35 și a doua când intrarea 2 este între 0.65 și 0.75. În primul caz sensibilitatea este negativă când ieșirea face o tranziție între 0 și 1. În al doilea caz (pentru intrarea 1 mai mică decât 0.5) sensibilitatea este negativă când ieșirea face o tranziție în direcția negativă de la 1 la 0. (d) Deși ieșirea este sensibilă la intrările 3, valorile sînt întotdeauna mai mici. Acesta era de așteptat de la o intrare aleatoare.

creditor. Schimbările în variabilele de intrare au fost procentaje din intrări neprelucrate. În funcție de cum au fost modificate aceste variabile de intrare, asta poate însemna o schimbare mai mare sau mai mică în una sau mai multe dintre ieșirile rețelei. În astfel de

Pentru dezvoltări ulterioare

Programul conține mai multe posibilități de dezvoltare ulterioară. Cum am descris mai înainte, făcînd


```

#include <stdio.h> /* pentru lucru cu fisiere */
#include <stdlib.h> /* malloc, rand, RAND_MAX, etc. */
#include <math.h> /* exp(), sqrt() */
#include <dos.h> /* FP_SEG(), FP_OFF() */

#define MAX_LAYER 4 /* numarul maxim de nivele */
#define MAX_PES 50 /* numarul maxim de elemente de procesare */

#define WORD(x) (((short *)(&x))[0])
#define MAX(x,y) ((x)>(y)?(x):(y))
typedef float WREAL; /* real pentru lucru */

/* structura de conectare */
typedef struct _conn {
    struct _pe *SourceP; /* pointer sursa */
    WREAL WtValR; /* valoarea ingrosarii */
    WREAL DWtValR; /* delta ingrosare */
} CONN;

/* element de procesare */
typedef struct _pe {
    struct _pe *NextPEP; /* urmatorul PE pe nivel */
    WREAL OutputR; /* iesirea PE-ului */
    WREAL ErrorR; /* zona de lucru pentru eroare */
    WREAL WorkR; /* zona de lucru pentru functia de
    explicatie */
    int *PEIndexN; /* indexul PE-ului in PEIndexP */
    int MaxConnsN; /* numarul maxim de conexiuni */
    int NConnsN; /* numarul de conexiuni utilizate */
    CONN ConnsS[1]; /* lista conexiunilor acestui PE */
} PE;

PE *LayerTP[MAX_LAYER+1] = {0}; /* adresele PE-urilor de pe
nivel */
int LayerNI[MAX_LAYER+1] = {0}; /* numarul PE-urilor de pe
fiecare nivel */

PE *PEIndexP[MAX_PES] = {0}; /* adresele PE-urilor */
int NextPEXN = {0}; /* indexul urmatorului PE liber */
PE PEBias = {0, 1.0, 0.0, 0};

/***** RRandR() - calculeaza o valoare aleatoare intr-un interval */
double RRandR( vR )
double vR; /* marimea intervalului */
{
    double rvR; /* valoarea returnata */

    /* valoarea aleatoare intre 0 si 1 */
    rvR = ((double)rand()) / (double)RAND_MAX;
    /* transformata intre -vR si vR */
    rvR = vR * (rvR + rvR - 1.0);
    return( rvR );
}

/***** AllocPE() - aloca dinamic un PE */
PE *AllocPE( peXN, MaxConnsN )
int peXN; /* indexul PE-ului (0=automat) */
int MaxConnsN; /* numarul maxim de conexiuni */
{
    PE *peP;
    int AlcSize;

    if ( NextPEXN == 0 ) {
        PEIndexP[0] = &PEBias;
        NextPEXN++;
    }
    if ( peXN == 0 )
        peXN = NextPEXN++;
    else if ( peXN >= NextPEXN )
        NextPEXN = peXN + 1;

    if ( peXN < 0 || MAX_PES <= peXN ) {

```

```

        printf( "Valoare ilegala a indexului PE-ului de alocat: %d\n",
                peXN );
        exit( 1 );
    }
    if ( PEIndexP[peXN] != (PE *)0 ) {
        printf( "PE-ul numarul %d este deja utilizat\n", peXN );
        exit( 1 );
    }
    AlcSize = sizeof( PE ) + MaxConnsN * sizeof( CONN );
    peP = (PE *) malloc( AlcSize );
    if ( peP == (PE *)0 ) {
        printf( "Nu pot aloca %d octeti pentru PE-ul numarul %d\n",
                AlcSize, peXN );
        exit( 1 );
    }
    memset( (char *) peP, 0, AlcSize );
    peP->MaxConnsN = MaxConnsN + 1; /* numarul maxim
    de conexiuni */
    peP->PEIndexN = peXN; /* index folosit pentru incarca-
    re/salvare */
    PEIndexP[peXN] = peP; /* index pentru mai tirziu */
    return( peP );
}

/***** AllocLayer() - Aloca dinamic PE-urile de pe un nivel */
int AllocLayer( LayN, NPESN, NConnPPEN )
int LayN; /* numarul nivelului */
int NPESN; /* numarul de PE-uri la pe nivel */
int NConnPPEN; /* numarul de conexiuni pentru fiecare PE */
{
    PE *peP;
    PE *apeP;
    int wxN;

    /* verificare de siguranta */
    if ( LayN < 0 || sizeof( LayerTP ) / sizeof( LayerTP[0] ) <= LayN )
    {
        printf( "Numar de nivel incorect (%d)\n", LayN );
        exit( 1 );
    }
    /* aloca PE-urile din nivel si le leaga impreuna */
    LayerNI[LayN] = NPESN;
    for( wxN = 0; wxN < NPESN; wxN++ , apeP = peP ) {
        peP = AllocPE( 0, NConnPPEN + 1 ); /* aloca urmatorul PE */
        if ( LayerTP[LayN] == (PE *)0 )
            LayerTP[LayN] = peP;
        else
            apeP->NextPEP = peP;
    }
    return( 0 );
}

/***** FreeNet() - elibereaza memoria ocupata de retea *****/
int FreeNet ()
{
    int wxN;
    char *P;

    for( wxN = 1; wxN < MAX_PES; wxN++ ) {
        if ( (P = (char *) PEIndexP[wxN]) != (char *)0 )
            free( P );
        PEIndexP[wxN] = (PE *)0;
    }
    NextPEXN = 0;
    for( wxN = 0; wxN < MAX_LAYER; wxN++ ) {
        LayerTP[wxN] = (PE *)0;
        LayerNI[wxN] = 0;
    }
    return( 0 );
}

/***** SaveNet() - salveaza retea intr-un fisier */

```

```

int SaveNet( fnP )
char *fnP; /* numele fisierului in care se listeaza */
{
    int wxN;
    FILE *fP;
    PE *peP;
    CONN *cP;
    int ConnXN;

    if ( NextPEXN <= 1 )
        return( 0 );
    if ( ( fP = fopen( fnP, "w" ) ) == ( FILE * ) 0 ) {
        printf( "Nu pot deschide fisierul de iesire %s", fnP );
        return( -1 );
    }
    /* salveaza toate PE-urile */
    for( wxN = 1; wxN < NextPEXN; wxN++ ) {
        peP = PEIndexP[wxN];
        if ( peP == ( PE * ) 0 ) {
            fprintf( fP, "%d 0 PE\n", wxN );
            continue;
        }
        fprintf( fP, "%d %d PE\n", wxN, peP->NConnsN );
        for( ConnXN = 0; ConnXN < peP->NConnsN; ConnXN++ ) {
            cP = &peP->ConnsS[ConnXN];
            fprintf( fP, "%d %d %d\n",
                cP->SourceP->PEIndexN, cP->WtValR, cP->DwtValR );
        }
        fprintf( fP, "%d %d Sfiritul PE-urilor\n", -1, 0 );
        /* salveaza modul in care sint legate PE-urile */
        for( wxN = 0; wxN < MAX_LAYER; wxN++ ) {
            if ( ( peP = LayerIP[wxN] ) == ( PE * ) 0 )
                continue;
            fprintf( fP, "%d NIVEL\n", wxN );
            do {
                fprintf( fP, "%d\n", peP->PEIndexN );
                peP = peP->NextPEP;
            } while ( peP != ( PE * ) 0 );
            fprintf( fP, "-1 Sfiritul Nivelului\n" ); /* sfiritul nivelului */
        }
        fprintf( fP, "-1\n" ); /* Sfiritul tuturor nivelelor */
        fclose( fP );
        return( 0 );
    }

    /***** LoadNet() - incarca o retea dintr-un fisier */
    int LoadNet( fnP )
    char *fnP; /* numele fisierului de unde se incarca */
    {
        int wxN;
        FILE *fP;
        PE *peP;
        PE *lpeP;
        int LayN;
        CONN *cP;
        int ConnXN;
        int PEN;
        int PENConnsN;
        int StateN;
        WREAL WtR, DWtR;
        char BufC[80];

        fP = ( FILE * ) 0;
        if ( ( fP = fopen( fnP, "r" ) ) == ( FILE * ) 0 ) {
            printf( "Nu pot deschide fisierul de intrare %s", fnP );
            return( -1 );
        }
        FreeNet(); /* renunta la retea deja existenta */
        StateN = 0;
        while( fgets( BufC, sizeof( BufC ) - 1, fP ) != ( char * ) 0 ) {
            switch ( StateN ) {

```

```

                case 0: /* PE-uri */
                    sscanf( BufC, "%d %d", &PEN, &PENConnsN );
                    if ( PEN < 0 ) {
                        StateN = 2;
                        break;
                    }
                    peP = AllocPE( PEN, PENConnsN );
                    cP = &peP->ConnsS[0];
                    ConnXN = PENConnsN;
                    if ( ConnXN > 0 ) {
                        StateN = 1;
                    }
                    break;
                case 1: /* Conexiuni PE-uri */
                    sscanf( BufC, "%d %f %f", &PEN, &WtR, &DWtR );
                    WORD( cP->SourceP ) = PEN;
                    cP->WtValR = WtR;
                    cP->DwtValR = DWtR;
                    cP++;
                    peP->NConnsN++;
                    if ( --ConnXN <= 0 )
                        StateN = 0;
                    break;
                case 2: /* Date despre nivele */
                    sscanf( BufC, "%d", &LayN );
                    StateN = 3;
                    if ( LayN < 0 )
                        goto Done;
                    lpeP = ( PE * ) &LayerIP[LayN];
                    break;
                case 3: /* Nivelele */
                    sscanf( BufC, "%d", &PEN );
                    if ( PEN < 0 ) {
                        StateN = 2;
                        break;
                    }
                    LayerNI[LayN]++; /* numarul de PE-uri */
                    peP = PEIndexP[PEN]; /* adresa PE-ului */
                    lpeP->NextPEP = peP;
                    lpeP = peP;
                    break;
            }
        }
    }
}

Done:
/* converteste indexurile tuturor PE-urilor in pointeri */
for( wxN = 1; wxN < MAX_PES; wxN++ ) {
    if ( ( peP = PEIndexP[wxN] ) == ( PE * ) 0 )
        continue;
    for( ConnXN = peP->NConnsN, cP = &peP->ConnsS[0];
        --ConnXN >= 0; cP++ ) {
        cP->SourceP = PEIndexP[ WORD( cP->SourceP ) ];
    }
}

if ( fP )
    fclose( fP );
return( 0 );
}

ErrExit:
if ( fP )
    fclose( fP );
FreeNet();
return( -1 );
}

/***** PrintNet() - Tipareste retea */
int PrintNet( fnP )
char *fnP; /* numele fisierului */
{
    FILE *fP;
    PE *dpeP;
    int layerXN;
    CONN *cP;
    int ConnXN;

```

```

if ( *fnP == '\0' ) {
    fP = stdout;
}
else {
    if ( ( fP = fopen( fnP, "a" ) ) == ( FILE * ) 0 ) {
        printf ( "Nu pot deschide fisierul de iesire <%s>\n", fnP );
        return( -1 );
    }
}
for( layerXN = 0; ( dpcP = LayerTP[layerXN] ) != ( PE * ) 0; layerXN++ ) {
    fprintf( fP, "\nNivelul %d cu %d PE-uri\n",
        layerXN, LayerN[layerXN] );
    for( ; dpcP != ( PE * ) 0; dpcP = dpcP->NextPEP ) {
        fprintf( fP,
            "%2d %04x: %04x PE Iesire = %6.3f Eroare = %6.3f\n",
            Variabila de lucru = %6.3f Numar conexiuni = %d\n",
            dpcP->PEIndexN, FP_SEG(dpcP), FP_OFF(dpcP),
            dpcP->OutputR, dpcP->ErrorR, dpcP->WorkR,
            dpcP->NConnsN );
        for( ConnXN=0; ConnXN < dpcP->NConnsN; ConnXN++ ) {
            cP = &dpcP->ConnsS[ConnXN];
            fprintf( fP,
                " Sursa: %2d %04x: %04x Ingresare = %7.3f\n",
                Delta Ingr. = %6.3f\n",
                cP->SourceP->PEIndexN, FP_SEG(cP->SourceP),
                FP_OFF(cP->SourceP), cP->WtValR, cP->DWtValR );
        }
    }
}
if ( fP != stdout )
    fclose( fP );
return( 0 );

/***** FullyConn() - conecteaza total un nivel sursa la o destinatie */
int FullyConn( DLayN, SLayN, RangeR )
int DLayN; /* nivelul destinatie */
int SLayN; /* nivelul sursa */
double RangeR; /* intervalul de marime */
{
    CONN *cP;
    PE *peP;
    PE *dpcP;

    /* pentru toate PE-urile de pe nivelul destinatie */
    for( dpcP = LayerTP[DLayN]; dpcP != ( PE * ) 0;
        dpcP = dpcP->NextPEP ) {
        cP = &dpcP->ConnsS[dpcP->NConnsN];
        if ( dpcP->NConnsN == 0 ) {
            cP->SourceP = &PEBias;
            cP->WtValR = RRandR( RangeR );
            cP->DWtValR = 0.0;
            cP++;
            dpcP->NConnsN++;
        }
        /* pentru toate PE-urile de pe nivelul sursa */
        for( speP = LayerTP[SLayN]; speP != ( PE * ) 0;
            speP = speP->NextPEP ) {
            cP->SourceP = speP;
            cP->WtValR = RRandR( RangeR );
            cP->DWtValR = 0.0;
            cP++;
            dpcP->NConnsN++;
        }
    }
}
return( 0 );

/***** BuildNet() - creaza toate structurile necesare pentru retea */
int BuildNet( NInpN, NHid1N, NHid2N, NOutN, ConnPrevF )
int NInpN; /* numarul de PE-uri de intrare */

```

```

int NHid1N; /* numarul de PE-uri pe nivelul intermediar 1 */
int NHid2N; /* numarul de PE-uri pe nivelul intermediar 2 */
int NOutN; /* numarul de PE-uri de iesire */
int ConnPrevF; /* 1 = conecteaza la toate nivelele anterioare */
{
    int ReqdPEsN;
    int LayerXN;
    int SLayN, DLayN;

    if ( NInpN <= 0 || NOutN <= 0 )
        return( -1 );
    FreeNet();
    LayerXN = 0;
    AllocLayer( LayerXN, NInpN, 0 );
    if ( NHid1N > 0 ) {
        LayerXN++;
        AllocLayer( LayerXN, NHid1N, NInpN );
        if ( NHid2N > 0 ) {
            LayerXN++;
            AllocLayer( LayerXN, NHid2N, NHid1N + ( ConnPrevF ?
                NInpN : 0 ) );
        }
        LayerXN++;
        AllocLayer( LayerXN, NOutN, ConnPrevF ? ( NInpN + NHid1N
            + NHid2N ) : NHid2N );
        for( DLayN = 1; LayerTP[DLayN] != ( PE * ) 0; DLayN++ ) {
            for( SLayN = ConnPrevF ? 0 : ( DLayN - 1 ); SLayN < DLayN;
                SLayN++ )
                FullyConn( DLayN, SLayN, 0.2 );
        }
    }
    return( 0 );

    /***** Recall() - executa un pas in retea */
    int Recall( ivRP, ovRP )
    WREAL *ivRP; /* vectorul de intrare */
    WREAL *ovRP; /* vectorul de iesire */
    {
        int DLayN;
        PE *peP;
        CONN *cP;
        int ConnC;
        double SumR;

        for( DLayN = 0; ( peP = LayerTP[DLayN] ) != ( PE * ) 0;
            DLayN++ ) {
            for( peP != ( PE * ) 0; peP = peP->NextPEP ) {
                if ( DLayN == 0 ) {
                    peP->OutputR = ivRP[0];
                    peP->ErrorR = 0.0;
                    ivRP++;
                }
                else {
                    ConnC = peP->NConnsN;
                    cP = &peP->ConnsS[0];
                    SumR = 0.0;
                    for( --ConnC >= 0; cP++ )
                        SumR += cP->SourceP->OutputR * cP->WtValR;
                    peP->OutputR = 1.0 / ( 1.0 + exp( - SumR ) );
                    peP->ErrorR = 0.0;
                }
                if ( LayerTP[DLayN+1] == ( PE * ) 0 ) {
                    ovRP[0] = peP->OutputR;
                    ovRP++;
                }
            }
        }
    }
    return( 0 );

    /***** Learn() - un pas elementar in procesul de invatare */

```

Fundamente

```

double Learn(ivRP, ovRP, doRP, LearnR, MomR)
WREAL *ivRP; /* vector de intrare */
WREAL *ovRP; /* vector de iesire */
WREAL *doRP; /* vector dorit */
double LearnR; /* rata de invatare */
double MomR; /* momentul */
{
    double ErrorR = 0.0;
    double LErrorR;
    PE *speP;
    int DLayN;
    PE *peP;
    CONN *cP;
    int ConnC;

    Recall( ivRP, ovRP );

    for( DLayN = 0; ( LayerTP[DLayN+1] ) != ( PE *) 0; )
        DLayN ++;
    /* calculeaza eroarea, propagare inapoi, actualizare ingrosari */
    for( ; DLayN > 0; DLayN-- ) {
        for( peP = LayerTP[DLayN]; peP != ( PE *) 0;
            peP = peP->NextPEP ) {
            if ( LayerTP[DLayN+1] == ( PE *) 0 ) {
                peP->ErrorR = ( doRP[0] - peP->OutputR );
                ErrorR += ( peP->ErrorR * peP->ErrorR );
                doRP ++;
            }
            peP->ErrorR * = peP->OutputR * ( 1.0 - peP->OutputR );
            ConnC = peP->NConnsN;
            cP = &peP->ConnsS[0];
            LErrorR = peP->ErrorR;
            for( ; --ConnC >= 0; cP ++ ) {
                speP = cP->SourceP;
                /* propaga eroarea */
                speP->ErrorR + = LErrorR * cP->WtValR;
                cP->DWtValR = LearnR * LErrorR * speP->OutputR +
                    MomR * cP->DWtValR;
                cP->WtValR + = cP->DWtValR; /* actualizeaza ingrosari */
            }
        }
    }
    return( ErrorR );
}

/***** Explain() - calculeaza iesirile corespunzatoare
modificarilor in intrare */
int Explain( ivRP, ovRP, evRP, DitherR )
WREAL *ivRP; /* vector de intrare */
WREAL *ovRP; /* vector de iesire */
WREAL *evRP; /* vector de explicatie */
double DitherR; /* perturbatie */
{
    PE *speP;
    int speXN;
    PE *dpeP;
    int dpeXN;
    int OutLXN;

    for( OutLXN = 0; LayerTP[OutLXN+1] != ( PE *) 0; )
        OutLXN ++;
    Recall( ivRP, ovRP );
    for( dpeP = LayerTP[OutLXN]; dpeP != ( PE *) 0;
        dpeP = dpeP->NextPEP )
        dpeP->WorkR = dpeP->OutputR;
    for( speXN = 0, speP = LayerTP[0]; speP != ( PE *) 0;
        speXN ++, speP = speP->NextPEP ) {
        /* perturbatie pozitiva */
        ivRP[speXN] + = DitherR;
        Recall( ivRP, ovRP );

```

```

for( dpeXN = 0, dpeP = LayerTP[OutLXN]; dpeP != ( PE *) 0;
    dpeXN ++, dpeP = dpeP->NextPEP )
    evRP[dpeXN] = 0.5 * ( dpeP->OutputR - dpeP->WorkR ) /
        DitherR;
    /* perturbatie negativa */
    ivRP[speXN] - = ( DitherR + DitherR );
    Recall( ivRP, ovRP );
    for( dpeXN = 0, dpeP = LayerTP[OutLXN]; dpeP != ( PE *) 0;
        dpeXN ++, dpeP = dpeP->NextPEP )
        evRP[dpeXN] - = 0.5 * ( dpeP->OutputR - dpeP->WorkR ) /
            DitherR;
    evRP + = dpeXN;
    ivRP[speXN] + = DitherR;
}
return( 0 );
}

```

```

/***** date pentru antrenarea retelei:
+-----+ 1.0
I | zero | | |
N | | | one | |
T +-----+ | | 0.7
R | | | | |
A | | | | |
R +-----+ | | 0.3
E | | | | |
A | | | zero | |
2 +-----+ | | 0.0
0.0 0.5 1.0
INTRAREA 1

```

Intrarea 3 este "zgomot".
*/

```

typedef struct _example {
    WREAL InVecR[3]; /* vector de intrare */
    WREAL DoVecR[2]; /* vector de iesire dorit */
} EXAMPLE;

```

```

#define NTEST ( sizeof( testE ) / sizeof( testE[0] ) )

```

```

EXAMPLE testE[] = {
    { 0.0, 0.0, 0.0, 0.0, 1.0 },
    { 0.0, 0.2, 0.6, 0.0, 1.0 },
    { 0.0, 0.4, 0.1, 1.0, 0.0 },
    { 0.0, 0.6, 0.7, 1.0, 0.0 },
    { 0.0, 0.8, 0.2, 0.0, 1.0 },
    { 0.0, 1.0, 0.8, 0.0, 1.0 },

    { 0.2, 0.0, 0.9, 0.0, 1.0 },
    { 0.2, 0.2, 0.3, 0.0, 1.0 },
    { 0.2, 0.4, 0.8, 1.0, 0.0 },
    { 0.2, 0.6, 0.2, 1.0, 0.0 },
    { 0.2, 0.8, 0.7, 0.0, 1.0 },
    { 0.2, 1.0, 0.1, 0.0, 1.0 },

    { 0.4, 0.0, 0.0, 0.0, 1.0 },
    { 0.4, 0.2, 0.6, 0.0, 1.0 },
    { 0.4, 0.4, 0.1, 1.0, 0.0 },
    { 0.4, 0.6, 0.7, 1.0, 0.0 },
    { 0.4, 0.8, 0.2, 0.0, 1.0 },
    { 0.4, 1.0, 0.8, 0.0, 1.0 },

    { 0.6, 0.0, 0.9, 0.0, 1.0 },
    { 0.6, 0.2, 0.3, 0.0, 1.0 },
    { 0.6, 0.4, 0.8, 1.0, 0.0 },
    { 0.6, 0.6, 0.2, 1.0, 0.0 },
    { 0.6, 0.8, 0.7, 1.0, 0.0 },
    { 0.6, 1.0, 0.1, 1.0, 0.0 },

    { 0.8, 0.0, 0.4, 0.0, 1.0 },
    { 0.8, 0.2, 0.6, 0.0, 1.0 },

```

```

{ 0.8, 0.4, 0.1, 1.0, 0.0 },
{ 0.8, 0.6, 0.7, 1.0, 0.0 },
{ 0.8, 0.8, 0.2, 1.0, 0.0 },
{ 0.8, 1.0, 0.8, 1.0, 0.0 },

{ 1.0, 0.0, 1.0, 0.0, 1.0 },
{ 1.0, 0.2, 0.3, 0.0, 1.0 },
{ 1.0, 0.4, 0.8, 1.0, 0.0 },
{ 1.0, 0.6, 0.2, 1.0, 0.0 },
{ 1.0, 0.8, 0.7, 1.0, 0.0 },
{ 1.0, 1.0, 0.0, 1.0, 0.0 }
};

int TestShuffleN[NTEST] = {0}; /* tablou de amestecare */
int TestSXN = {NTEST + 1};

/**** NextTestN() - amesteca */
int NextTestN()
{
    int HitsN;
    int wxN;
    int xN, yN;

    if (TestSXN >= NTEST) {
        for( wxN = 0; wxN < NTEST; wxN++ )
            TestShuffleN[wxN] = wxN;
        for( HitsN = 0; HitsN < NTEST + NTEST/2; HitsN++ ) {
            xN = rand() % NTEST;
            yN = rand() % NTEST;
            wxN = TestShuffleN[xN];
            TestShuffleN[xN] = TestShuffleN[yN];
            TestShuffleN[yN] = wxN;
        }
        TestSXN = 0;
    }
    return( TestShuffleN[TestSXN + +] );
}

/**** TrainNet() - antreneaza retea */
int TrainNet( ErrLvlR, MaxPassN )
double ErrLvlR; /* nivelul de eroare de atins */
long MaxPassN; /* numarul maxim de pasi */
{
    WREAL ivR[MAX_PES];
    double lsErrR;
    int CurTestN;
    int HitsN;
    long PassN;
    int ExampleN;

    HitsN = 0;
    CurTestN = 0;
    lsErrR = 0.0;
    PassN = 0;
    for(;;) {
        ExampleN = NextTestN();
        lsErrR += Learn( &testE[ExampleN].InVecr[0], &ivR[0],
            &testE[ExampleN].DoVecr[0], 0.9, 0.5 );

        CurTestN++;
        if ( CurTestN >= NTEST ) {
            PassN++;
            lsErrR = sqrt( lsErrR ) / (double) NTEST;
            if ( lsErrR < ErrLvlR )
                HitsN++;
            else
                HitsN = 0;
            printf( "Pass %3ld Error = %3f Hits = %d\n", PassN, lsErrR,
                HitsN );
            if ( PassN > MaxPassN || HitsN > 3 )
                break;
            CurTestN = 0;
            lsErrR = 0.0;
        }
    }
}

}
return( 0 );
}

/**** ExplainNet() - executa explicatia si o listeaza */
int ExplainNet( fnP, DitherR )
char *fnP; /* numele fisierului de iesire */
double DitherR; /* perturbatia */
{
    FILE *fP;
    int wxN;
    int xN, yN;
    int axN;
    WREAL *wFP;
    static WREAL ivR[MAX_PES] = {0};
    static WREAL ovR[MAX_PES] = {0};
    static WREAL evR[MAX_PES * MAX_PES] = {0};

    if ( *fnP == '\0' )
        fP = stdout;
    else
        if ( ( fP = fopen( fnP, "a" ) ) == (FILE *) 0 ) {
            printf( "Nu pot deschide fisierul de iesire <%s> \n", fnP );
            return( -1 );
        }
    fprintf( fP, "\n
*** Iesirea retelei ca functie de intrare: 1 & 2 ***\n\n" );
    ivR[2] = 0.5;
    for( yN = 20; yN >= 0; yN-- ) {
        if ( ( yN % 2 ) == 0 )
            fprintf( fP, "%6.2f | ", yN / 20. );
        else
            fprintf( fP, " | " );
        for( xN = 0; xN <= 20; xN++ ) {
            ivR[0] = xN / 20.;
            ivR[1] = yN / 20.;
            Recall( &ivR[0], &ovR[0] );
            if ( fabs( ovR[0] ) < .1 )
                fprintf( fP, " - " );
            else
                fprintf( fP, "%5.1f", ovR[0] );
        }
        fprintf( fP, "\n" );
    }
    fprintf( fP, " +. - " );
    for( xN = 0; xN <= 20; xN++ )
        fprintf( fP, "-----" );
    fprintf( fP, "\n " );
    for( xN = 0; xN <= 20; xN++ )
        fprintf( fP, ( xN % 2 ) == 0 ? "%5.1f" : " ", xN / 20. );
    fprintf( fP, "\n" );
    fprintf( fP, "\n
*** Functia de explicatie pentru intrarea 1
peste domeniul de intrare ***\n\n" );
    ivR[2] = 0.5;
    for( yN = 20; yN >= 0; yN-- ) {
        if ( ( yN % 2 ) == 0 )
            fprintf( fP, "%6.2f | ", yN / 20. );
        else
            fprintf( fP, " | " );
        for( xN = 0; xN <= 20; xN++ ) {
            ivR[0] = xN / 20.;
            ivR[1] = yN / 20.;
            Explain( &ivR[0], &ovR[0], &evR[0], DitherR );
            if ( fabs( evR[0] ) < .1 )
                fprintf( fP, " - " );
            else
                fprintf( fP, "%5.1f", evR[0] );
        }
        fprintf( fP, "\n" );
    }
}
}

```

```

fprintf( fP, " +-" );
for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, "-----" );
fprintf( fP, "\n          " );
for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, ( xN % 2 ) == 0 ? "%5.1f" : "    ", xN/20. );
fprintf( fP, "\n" );
fprintf( fP, "\f\n
*** Functia de explicatie pentru intrarea
2 peste domeniul de intrare ***\n\n" );
ivR[2] = 0.5;
for( yN = 20; yN >= 0; yN-- ) {
if ( ( yN % 2 ) == 0 )
    fprintf( fP, "%6.2f | ", yN / 20. );
else
    fprintf( fP, " | " );
for( xN = 0; xN <= 20; xN++ ) {
ivR[0] = xN / 20.;
ivR[1] = yN / 20.;
Explain( &ivR[0], &ovR[0], &cvR[0], DitherR );
if ( fabs( cvR[2] ) < .1 )
    fprintf( fP, " -" );
else
    fprintf( fP, "%5.1f", cvR[2] );
}
fprintf( fP, "\n" );
}
fprintf( fP, " +-" );
for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, "-----" );
fprintf( fP, "\n          " );
for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, ( xN % 2 ) == 0 ? "%5.1f" : "    ", xN/20. );
fprintf( fP, "\n" );
fprintf( fP, "\f\n
*** Functia de explicatie pentru intrarea 3
peste domeniul de intrare ***\n\n" );
ivR[2] = 0.5;
for( yN = 20; yN >= 0; yN-- ) {
if ( ( yN % 2 ) == 0 )
    fprintf( fP, "%6.2f | ", yN / 20. );
else
    fprintf( fP, " | " );
for( xN = 0; xN <= 20; xN++ ) {
ivR[0] = xN / 20.;
ivR[1] = yN / 20.;
Explain( &ivR[0], &ovR[0], &cvR[0], DitherR );
if ( fabs( cvR[4] ) < .1 )
    fprintf( fP, " -" );
else
    fprintf( fP, "%5.1f", cvR[4] );
}
fprintf( fP, "\n" );
}
}
fprintf( fP, " +-" );
for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, "-----" );
fprintf( fP, "\n          " );
for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, ( xN % 2 ) == 0 ? "%5.1f" : "    ", xN/20. );
fprintf( fP, "\n" );
if ( fP != stdout )
    fclose( fP );
return( 0 );
}

```

/* functia principala */

```

int main()
{
    int ActionN;
    char *sP;
    char *aP;

```

```

char BufC[ 80];

printf( "\nProgram de explicare a concluziilor
unei retele neuronale\n\n" );
for( ;; ) {
    printf( "\n
C - creare retea\n\
L [fname] - incarcare retea antrenata\n\
S [fname] - salvare retea\n\
P [fname] - listare retea\n\
F - stergere retea din memorie\n\
T - antrenare retea\n\
E [fname] - explicare retea\n\
X - iesire din program\n\n" );
    Ce doriti sa faceti ? " );
    fflush( stdout );
    sP = fgets( BufC, sizeof( BufC ) - 1, stdin );
    if ( sP == ( char * ) 0 )
        break;
    while( *sP != 0 && *sP <= ' ' )
        sP++;
    ActionN = *sP;
    if ( 'A' <= ActionN && ActionN <= 'Z' )
        ActionN -= 'A' - 'a';
    sP++;
    for( aP = sP; *aP <= ' '; )
        aP++;
    *aP = '\0';
    switch( ActionN ) {
        case 'c':
            BuildNet( 3, 5, 0, 2, 1 );
            break;
        case 'l':
            if ( *sP == '\0' )
                sP = "network.net";
            LoadNet( sP );
            break;
        case 's':
            if ( *sP == '\0' )
                sP = "network.net";
            SaveNet( sP );
            break;
        case 'p':
            PrintNet( sP );
            break;
        case 'f':
            FreeNet();
            break;
        case 't':
            TrainNet( 0.003, 100000L );
            break;
        case 'e':
            ExplainNet( sP, .01 );
            break;
        case 'x':
            goto Done;
        default:
            break;
    }
}
Done:
return( 0 );
}

```

Administrarea rețelelor

Indiferent ce sistem de operare folosește rețeaua, un administrator de LAN are mereu aceleași lucruri de bază de făcut: să creeze noi numere de identificare pentru utilizatori, să întrețină structura de directori și securitatea file server-ului, să șteargă fișierele vechi, să instaleze și să actualizeze aplicațiile, să întrețină cozile de listare, să facă salvări, și multe altele.

"Sculele" disponibile pentru aceste operațiuni determină situația muncii sale: dacă este vorba de câteva minute "de iad" de schimbări sau dacă este un efort herculean. Totuși, atunci când o firmă se decide asupra sistemului de operare ce va fi utilizat pentru rețea, adesea sînt neglijate timpul și efortul cerute de administrarea rețelei.

În acest articol vom examina fiecare din aceste sarcini de administrare și vom compara modul în care diferă, în ceea ce privește susținerea funcției de administrator, cele mai importante sisteme de operare pentru rețele locale - NetWare al lui Novell Inc. LAN Manager al lui Microsoft și LAN Server al lui IBM.

Crearea de utilizatori și securitatea

Administratorul unui LAN trebuie să creeze noi numere de identificare pentru utilizatori ori de cîte ori cineva se leagă la rețea și să șteargă utilizatorii atunci cînd cineva părăsește rețeaua. Identificatorii pentru utilizatori trebuie să fie adăugați în fișierele "batch" la prima instalare a sistemului și la adăugarea de noi departamente la rețea. Administratorii rețelelor de instruire sînt nevoiți să creeze și să șteargă identificatori de utilizatori pentru fiecare clasă, în fiecare trimestru. După instalarea inițială, administratorul mai trebuie să adauge utilizatori unul cîte unul atunci cînd sînt angajați oameni noi sau cînd sînt transferați.

Crearea unui identificator pentru un utilizator nu înseamnă numai asigurarea unui șir de caractere ca nume pentru fiecare persoană care se conectează la rețea. Trebuie să se stabilească și căror grupuri le va aparține utilizatorul. Grupurile nu se folosesc numai pentru asigurări la departamente și grupuri de lucru, ci și pentru utilizarea aplicațiilor și a disponibilităților de listare. Dincolo de abilitatea de a gestiona utilizatorii ca grupuri se situează o altă posibilitate valoroasă: gruparea file server-elor și a altor resurse (de exemplu server-ele de listare) în "domenii". Sistemele de operare pentru rețele care funcționează în acest mod permit administratorului să stabilească identificatorul utilizatorului o singură dată - pentru domeniul, în loc să fie nevoie să o facă pentru fiecare server în parte. Unele sisteme

de acordare de nume extind domeniul "global" (pentru toată întreprinderea).

Un sistem de operare pentru rețea trebuie să pună la dispoziție un mecanism pentru a crea și manipula identificatorii de utilizator unul cîte unul cu ajutorul unui utilitar bazat pe meniuri, precum și un mecanism de gestionare a acestor identificatori pe pachete, pe măsură ce se adaugă noi grupuri de utilizatori sau este nevoie să se facă modificări ce afectează mai mulți utilizatori. Adăugarea unei noi aplicații la 100 de utilizatori - unul cîte unul - poate deveni foarte plictisitoare.

Una din cele mai delicate sarcini ale administratorului unei rețele este stabilirea și păstrarea controlului asupra accesului utilizatorilor la resursele rețelei. Sub oricare din principalele sisteme de operare pentru rețele, administratorul de sistem poate asigura utilizatorilor privilegii diferite asupra directoarelor, și uneori chiar asupra anumitor fișiere, de pe file server.

Deoarece toate sistemele au prima linie de apărare bazată pe parole de intrare, este foarte important modul în care sînt protejate fișierele care conțin parolele. Sistemul trebuie să permită un administrator de sistem care să poată citi toate datele de pe server (dacă nu pentru alt motiv, măcar pentru salvări), precum și un sub-administrator care poate crea utilizatori dar nu le poate citi toate datele. Parolele trebuie să fie memorate în formă criptată, cu verificarea lungimii minime și a vârstei maxime ale parolei. De asemenea, sistemul trebuie să dispună de o metodă de limitare a accesului unui utilizator ca stație sau ca timp.

Cu cît accesul la parola de intrare este mai strict protejat, cu atît mai eficientă va trebui să fie "scula" cu care poate gestiona administratorul acest sistem de securitate. Nici unul dintre sisteme nu asigură același nivel de protecție ca securitatea de pe mainframe, dar unele pot asigura o bună securitate a parolei.

Gestionarea aplicațiilor

Sistemul de operare pentru rețele ideale ar trebui să aibă și un mecanism simplu de instalare a aplicațiilor noi. El ar trebui să creeze directoare ale utilizatorilor pentru informațiile de configurare, să întrețină un sistem de meniuri administrat central - dar nu sufocant și un sistem de bariere care limitează numărul utilizatorilor care rulează Wondercale la numărul maxim specificat în licență.

Un sistem de meniuri administrat central permite administratorului să adauge o aplicație în meniurile a 20 de utilizatori fără a-i deranja, și fără a fi nevoie să-și părăsească biroul. Atunci cînd utilizatorul "intră" în rețea, trebuie stabilit mediul în care intră. Trebuie să se

asigneze câte o literă de "drive" fiecărui director de pe file server. Unul sau mai multe din aceste directoare trebuie să fie în "căile" utilizatorului, astfel încât acesta să poată găsi Wonderdraw de pe discul P: fără a trebui să schimbe directorul. Ar putea fi necesar să se lanseze automat, la conectarea în rețea, un sistem de meniuri să se încarce câteva programe TSR (terminate-and-stay-resident) pentru rețea, cum ar fi un program de poștă electronică sau un controlor de listare.

Odată ce a fost definit acest "mediu" pentru un utilizator, ar trebui să existe o cale de a-l copia la câțiva utilizatori deodată. Modificările în fișierele cu procedurile de conectare sau în fișierul de comenzi nu ar trebui să îl oblige pe administrator să se "plimbe" pe la stația de lucru a fiecărui utilizator, nici nu ar trebui să impună utilizarea unui sistem de meniuri specific.

Salvări

Cea mai importantă sarcină a administratorului de rețea este salvarea datelor. Sistemul de operare pentru rețea poate să includă un utilitar de salvare pentru controlul unui dispozitiv de salvare, iar în caz contrar ar trebui să se ferească atât cât este posibil de soft de la alte firme. De asemenea, trebuie să existe posibilitatea ca un utilizator care nu este administratorul sistemului să poată salva toate datele de pe sistem.

Listări

Imprimantele pot să-și facă de cap pe cont propriu. Dacă o listare care are nevoie de o imprimantă LaserJet cu cartuș PostScript este trimisă din greșeală către o imprimantă LaserJet III fără cartuș PostScript, nu există nici un motiv ca imprimanta să "listeze" 127 de pagini goale, just? Și totuși așa ceva se poate întâmpla în orice rețea.

După stabilirea inițială a cozilor și a imprimantelor, cele mai multe din responsabilitățile cu privire la listări ale administratorului se concentrează asupra gestionării cozilor de listare. Un administrator de sistem trebuie să aibă posibilitatea de a șterge un "job" de listare (ca cel menționat mai sus) din coada de listare înainte de a se produce stricăciuni prea mari, ca și posibilitatea de a muta listările urgente în vârful cozii și de a redirecta coada de la imprimanta laser defectă de la etajul doi către cea de la etajul trei care funcționează.

Un "printer spooler" ideal ar trebui să alerteze automat utilizatorii că este necesar să-și trimită din nou listările. De asemenea, ar trebui să-l anunțe pe administrator când are nevoie de hârtie.

De asemenea, administratorul de sistem va dori să poată transmite unui manager de listări toată complicația cu administrarea listărilor fără a compromite securitatea.

NetWare

Crearea de utilizatori și securitatea

În sistemul NetWare, instrumentul principal al administratorului pentru gestionarea utilizatorilor este utilitarul axat pe meniuri SYSCON. El poate fi lansat de la orice stație de lucru DOS din rețea. Cu SYSCON, administratorul poate să creeze utilizatori și grupuri de lucru, să administreze drepturile lor asupra directoarelor și să controleze accesul utilizatorilor la sistem. NetWare îi permite administratorului de sistem să limiteze accesul unui utilizator într-o anumită perioadă a zilei, sau la anumite stații de lucru. De asemenea, administratorul poate fixa cerințele pentru parole, inclusiv durata maximă și lungimea minimă, precum și numărul maxim de stații de lucru cu care utilizatorul se poate conecta în același timp. Toate acestea se pot face utilizator cu utilizator.

Administratorul poate crea mai mulți utilizatori odată și poate stabili restricții cu utilitarele MAKEUSER și USERDEF descrise mai jos. Din nefericire, odată ce utilizatorii au fost creați, nu există o metodă de a schimba pe loturi restricțiile de timp și de parolă. A face modificări pentru 30 de utilizatori, unul câte unul, pentru a nu le permite accesul în rețea duminică este o cale bună de a-ți irosi dimineața. Deși SYSCON permite selectarea mai multor utilizatori folosind tasta F5, dacă se selectează mai mult de un utilizator, SYSCON nu permite modificări semnificative.

Sub NetWare 2.1x, numai utilizatorul special "supervisor" - sau echivalentul său din punct de vedere al securității - poate crea noi identificatori de utilizator. Deoarece supervisor-ul are drepturi nelimitate asupra tuturor datelor de pe server, numai un număr limitat de utilizatori pot crea alți utilizatori fără riscuri în ceea ce privește securitatea.

NetWare 3.1 și NetWare 2.2 adaugă tipul de utilizator "șef de grup de lucru". Șeful unui grup de lucru nu are acces la toate informațiile de pe server, dar poate să creeze noi utilizatori care au drepturi egale, cel mult, cu ale lui.

Șefii grupurilor de lucru pot să șteargă numai acei utilizatori pe care i-au creat sau care le-au fost asignați de către supervisor. Aceasta permite administratorului de sistem să delege sarcina de creare a utilizatorilor șefilor de departamente.

Sub NetWare 2.1x administrarea securității se face director cu director. Pentru gestionarea securității la nivel de director există două utilitare: FILER și SYSCON, care folosesc meniuri. Există și alternative cu linie de comandă pentru cei care cred că meniurile țin numai de restaurante. Și NetWare 3.1 și 2.2 permit restricții de acces la nivel de director. SYSCON permite gestionarea securității pe grupuri de utilizatori, dar grupurile trebuie create ținând cont de considerentele de securitate.

NetWare nu oferă posibilitatea de a controla accesul utilizatorului la anumite fișiere sau directoare. Unele produse soft, cum ar fi LT Auditor al lui Blue Lance sau LANtight al lui Network Management Inc., oferă această facilitate prin intermediul unor programe TSR pe stația de lucru.

Utilitarul MAKEUSER produs de NOVELL oferă comenzi pentru configurarea grupurilor de utilizatori. La instalarea unui nou sistem, utilizarea lui SYSCON pentru configurarea a 30 de utilizatori, cinci grupuri și a tuturor drepturilor lor de acces este o misiune consumatoare de timp. MAKEUSER permite administratorului de sistem să scrie un fișier de comenzi pentru a crea utilizatori, pentru a le crea directoarele, pentru a copia un fișier existent în fișierul lor de comenzi pentru conectare, pentru a controla apartenența la grupuri și pentru a modifica parametri de control ca lungimea parolei și dreptul de acces la directoarele de pe server.

Cu MAKEUSER, o simplă căutare și înlocuire poate transforma un fișier de comenzi MAKEUSER care crează noi utilizatori într-unul care șterge utilizatori - sau date din directoarele lor, dacă este necesar. MAKEUSER este util pentru centre de pregătire și pentru școli care trebuie să creeze și să șteargă conturile elevilor la începutul și sfârșitul perioadelor de învățare.

Deși MAKEUSER este un instrument prețios, el are câteva limitări importante. Cu ajutorul lui se pot adăuga utilizatori grupurilor, dar nu se pot crea grupuri și nici nu se pot adăuga drepturi grupurilor. Deși se poate crea o bibliotecă de fișiere de comenzi, nu există facilitățile de a memora configurații standard pentru utilizatori, iar scrierea fișierelor de comenzi este mult prea apropiată de programare pentru administratorii de sistem care nu au o formație tehnică.

Utilitarul USERDEF de sub NetWare 3.1 extinde posibilitățile lui MAKEUSER. El este un program bazat pe meniuri care generează fișiere de comenzi MAKEUSER și care rulează MAKEUSER. Cu ajutorul lui USERDEF, administratorul de sistem poate construi modele pentru diferite tipuri de utilizatori. De asemenea, USERDEF copiază baza de date cu configurația job-urilor de listare a lui PRINTCON pentru utilizatorii definiți cu ajutorul lui. Utilitarul Origen produs de Preferred Systems Inc. duce mai departe conceptul MAKEUSER, permițând unui administrator să construiască o instalare standard de server completă, cu grupuri și cu structura de directoare agreată de administrator.

NetWare oferă servicii de contabilizare, permițând administratorului să modifice timpul de conectare, spațiul de memorare pe disc și alte resurse ale sistemului. Din nefericire, contabilitatea lui NetWare nu permite urmărirea numărului de pagini listate - costul cel mai ușor de urmărit într-o rețea și nu oferă raportări utile.

Gestionarea aplicațiilor și a lansării

NetWare include un sistem de creere de meniuri pentru facilitarea instalării aplicațiilor. Sistemul este destul de primitiv, fără facilități de combinare a meniurilor standard cu cele ale utilizatorului sau pentru administrare centrală.

Mediul de conectare al utilizatorului este controlat de o serie de fișiere de comenzi. Sistemul execută automat un fișier de comenzi general - pentru întregul sistem, apoi unul pentru utilizatorul respectiv. Fișierele de comenzi de conectare se aseamănă cu fișierele de comenzi DOS, și ele pot asigura utilizatorului unități de disc și imprimante. Fișierele de comenzi pentru conectare se editează cu utilitarul SYSCON. Când crează noi utilizatori, administratorul poate să copieze fișierul de comenzi de la un utilizator existent, dar utilitarul nu oferă posibilitatea de a scrie un fișier de comenzi pentru un grup.

Salvări

NetWare oferă câteva utilitare pentru salvări. Dar termenul "utilitare" poate fi greșit folosit, deoarece ele au o utilizare limitată. NetWare Loadable Module pentru salvări pe bandă suportă un set limitat de dispozitive și sînt rar utilizate de administratorii de NetWare.

Listări

Cozile de listare sînt gestionate cu utilitarul PCONSOLE, care permite administratorului să controleze job-urile din coada de listare. El nu permite administratorului să controleze imprimantele decît dacă este utilizat programul server de listare PSERVER. Dacă se rulează NetWare 2.1x fără PSERVER, multe din comenzile de control ale imprimantelor trebuie să fie date de la consola file server-ului.

Utilizatorii pot și ei să ruleze PCONSOLE pentru a-și gestiona propriile job-uri de listare.

LAN Manager

Crearea de utilizatori și securitatea

O rețea LAN Manager poate fi administrată de la o stație de lucru OS/2 sau de la o stație de lucru DOS care rulează un software avansat pentru stații. (Stațiile de lucru DOS pot fi instalate fie ca stații de bază, fie ca stații avansate. Opțiunea pentru stație avansată permite administrarea rețelei, dar consumă mai multă memorie).

LAN Manager oferă un utilitar bazat pe meniuri pentru crearea de utilizatori și grupuri de utilizatori. Administratorul poate stabili orele de conectare, stațiile de lucru permise, directoarele de lucru și data expirării contului atunci cînd crează utilizatori.

Pot fi stabilite la nivel de server vîrsta maximă și minimă a parolei, numărul de parole vechi pe care

utilizatorul nu le poate utiliza și lungimea minimă pentru parolă.

Programul permite administratorului să multiplice un utilizator, copiind toate privilegiile și proprietățile pentru un nou utilizator - ceea ce constituie o caracteristică neprețuită pentru crearea grupurilor. Pentru toate aceste funcții sînt disponibile și utilitare care pot fi apelate prin linii de comandă, astfel încît se pot scrie fișiere de comenzi pentru a crea loturi de utilizatori, dar nu există un utilitar destinat acestui scop.

Gestionarea aplicațiilor și a lansării

LAN Manager nu oferă un sistem de meniuri administrat central. Fixarea căii pentru fișierele unui utilizator trebuie prin intermediul fișierelor standard de comenzi DOS, iar modificările la o cale existentă se pot face numai la începutul sau la sfîrșitul ei (dar întreaga cale poate fi înlocuită cu o nouă declarație de cale).

Un server LAN Manager poate fi instalat fie pentru securitate la nivel de utilizator, fie pentru securitate la nivel de porțiune. Securitatea la nivel de utilizator din LAN Manager funcționează asemănător cu cea din NetWare. Un utilizator poate avea una sau toate din permisiunile definite (vezi tabela) pentru un director dat.

Securitatea la nivel de porțiune permite asigurarea parolelor și a permisiunilor pentru fiecare director. Un utilizator care dorește să asigneze o unitate logică de pe stația sa de lucru unui director de pe server cu ajutorul comenzii NET USE (similară cu comanda MAP din NetWare) va trebui să introducă parola pentru acel director. Securitatea este asigurată numai prin parola la nivel de director, fără a restrînge accesul la date pentru fiecare utilizator.

LAN Manager cîteva funcții de control care permit să se supravegheze conectările, deconectările, sau momentele în care utilizatorii asignează o unitate unui director cu comanda NET USE. Nu este supravegheat accesul la nivel de fișier. Un administrator dorește, de multe ori, să afle cine a lucrat cu un anumit fișier și cînd. LAN Manager poate urmări cînd a fost asigantă o unitate unui director, dar nu indică cînd sau dacă - a fost utilizat un fișier din acel director; de obicei, asignările de directoare se fac în momentul în care utilizatorul se conectează la rețea.

Salvări

LAN Manager se livrează acum cu o copie a softului pentru salvare pe bandă Sytos Plus produs de Sytron - Sytos Plus suportă multe tipuri de unități de bandă de pe file server.

LAN Manager-ul lui Microsoft, ca și varianta LAN Server a lui IBM, mai include și un interesant serviciu de duplicare. Pentru un file server se poate alege ca, de cîte ori se modifică un fișier dintr-un director specificat, acel fișier să fie copiat în unul sau mai multe

directoare de salvare. Directoarele destinație pot fi pe un alt file server sau pe o stație de lucru OS/2. (Pentru rețelele NetWare, o soluție similară este oferită de produsul LAN shadow furnizat de NMI, care cere o stație de lucru dedicată).

Serviciile de duplicare asigură un fel de semi-toleranță la erori. Ele sînt utile acolo unde pierderea datelor, chiar și de pe durata unei ore, este inacceptabilă. O limitare o constituie faptul că nu se pot copia fișiere deschise, astfel încît nu se pot proteja baze de date care sînt menținute deschise pe toată durata zilei de lucru; dar dacă fișierul este închis la un moment dat, serviciul de duplicare îl va copia în directorul de salvare. Serviciul de duplicare poate fi o cale utilă de a distribui actualizări sau alte date de la un server la altul (fișierele noi - sau fișierele existente care au fost modificate - vor fi copiate automat).

Listări

Interfața bazată pe meniuri a lui LAN Manager permite administratorului să controleze cozile de listare, să șteargă job-uri și, în general, să gestioneze cozile. Utilizatorii și administratorii pot să oprească temporar, să șteargă sau să elibereze job-uri de listare. LAN Manager îi permite administratorului să protejeze cozile de listare prin parole și să delege gestionarea cozilor.

De asemenea, LAN Manager oferă administratorului posibilitatea de a proiecta pagini de separare care vor fi listate înaintea fiecui job pentru identificarea utilizatorului. Paginile de separare pot include comenzi PostScript sau fișiere grafice.

LAN Manager suportă imprimante situate la distanță și imprimantepartajate legate la stațiile de lucru ale utilizatorilor astfel încît administratorul nu este nevoit să se deplaseze la file server pentru gestionarea listărilor.

LAN Server

LAN Server este o variantă a lui LAN Manager, dar interfața cu utilizatorul și caracteristicile de administrare sînt diferite.

O rețea LAN Server este administrată de la consola file server-ului sau de la o stație de lucru OS/2 din rețea. Nu sînt disponibile funcții de administrare de la stațiile de lucru DOS. După cum este de așteptat de la un produs IBM, LAN Server oferă un control mai centralizat decît NetWare sau LAN Manager 2.0, punînd la dispoziție utilitare de gestionare a meniurilor și un sistem de denumire pe domenii.

Crearea de utilizatori și securitate

Identificatorii de utilizatori sînt creați cu ajutorul programului EE User Profile Maintenance de sub OS/2. Un utilitar se poate conecta numai de la o singură mașină la un moment dat, astfel încît este posibil ca unii utilizatori să aibă nevoie de mai multe conturi. Grupurile

de utilizatori sînt suportate ca în NetWare sau LAN Manager.

Administratorul unei rețele LAN Server asignează fiecărui director o "parcelă". Dacă un utilizator dorește să asigneze o unitate de disc acelui director, el utilizează comanda NET USE, făcînd referire la "poreclă". Aceasta face utilizarea rețelei mai transparentă, deoarece utilizatorii nu sînt nevoiți să urmărească unde (pe care server) se află fiecare director.

Totuși, utilizarea "parcelelor" poate crea necazuri administratorului. De exemplu, dacă un server ajunge la limita capacității sale de memorare, ar putea fi necesar să se mute unele directoare pe un alt server. Ar fi bine să existe posibilitatea modificării unei "porecle" existente, pentru a informa LAN Server-ul că acum directorul se află pe un alt server. Din nefericire, nu există posibilitatea modificării numelui server-ului pentru o "poreclă" existentă. Administratorul va trebui să șteargă "porecla" și să creeze alta cu același nume.

Programul LAN Requester, de sub OS/2, este unealta de bază pentru administratorul LAN Server. Cu acest utilitar administratorul poate stabili permisiuni ale utilizatorilor și ale grupurilor pentru "porecle" sau directoare.

Gestionarea aplicațiilor și a lansării

Administratorul poate limita numărul utilizatorilor care își asignează unități la o "poreclă". Dacă există o licență pentru patru utilizatori de Wondercalc, se poate limita la patru numărul utilizatorilor pentru directorul Wondercalc. Pentru ca aceasta să aibă efect pentru un produs care are licență pentru utilizatori simultani, va trebui ca utilizatorii să-și asigneze o unitate chiar înainte de a utiliza produsul și să șteargă asignarea cînd termină.

Programul User Profile Maintenance permite administratorului să stabilească mediul utilizatorului conectat stabilind imprimanta și asignările de unități în momentul conectării și controlînd ce elemente apar în meniul de selectare a aplicațiilor al utilizatorului. Din nefericire, administratorul nu poate să selecteze cinci utilizatori și să le adauge tuturor o asignare de unitate.

Modulul DOS LAN Requester din LAN Server este livrat cu un sistem de gestionare a meniurilor, numit selector de aplicații, care seamănă cu DOSSHLL din DOS 4.0. Primul meniu acoperă aplicațiile de pe server - acele aplicații care se găsesc pe file server, spre deosebire de cele de pe discul hard al utilizatorului. Acest meniu poate fi controlat de administratorul de sistem. Atunci cînd se adaugă o aplicație se pot specifica numai asignările de unități care se folosesc în timpul rulării acelei aplicații. De asemenea se poate stabili de pe ce unitate se va rula aplicația.

Pentru fiecare utilizator se construiește un fișier PROFILE.BAT care se va executa atunci cînd utilizatorul se conectează la rețea. Calea pentru fișierele unui utilizator poate fi stabilită în PROFILE, dar ea se pierde

dacă utilizatorul părăsește sistemul de meniuri Application Selector. LAN Server nu oferă nici o cale de limitare a folosirii spațiului pe disc pentru utilizatori sau pentru directoare.

Salvări

LAN Server nu are facilități de salvare în afara serviciului de duplicare și a programelor de salvare și restaurare din OS/2. Pentru salvări pe bandă, se folosește aproape peste tot produsul Sytos Plus.

Listări

Gestionarea cozilor de listare se face prin intermediul OS/2-ului. Un administrator de la o stație OS/2 poate să șteargă job-uri de listare, să întrerupă temporar imprimante sau cozi și, în general, să gestioneze procesul de listare. Stațiile de lucru DOS pot să sisteze temporar, să șteargă și să elibereze job-uri, dar nu pot să oprească cozile de listare.

Care este cel mai bun?

Alegerea unui sistem de operare pentru rețea are un impact teribil asupra vieții de zi cu zi a administratorului rețelei. Desigur, facilitățile de administrare nu sînt singurele aspecte ale unei rețele care trebuie luate în considerare. Cel puțin la fel de importante sînt viteza, securitatea și posibilitatea de utilizare a softului provenit din alte surse. Dar acest articol dorește să se oprească puțin asupra unui aspect care are, de obicei, prioritate minimă: operarea cotidiană a unei rețele.

În ceea ce privește facilitățile de administrare nu există un "cîștigător" clar deoarece nu există un beneficiar standard. Dar iată o comparație:

- NetWare are cele mai bune "scule" - de exemplu utilitarul MAKEUSER axat pe fișiere de comenzi. Pe de altă parte, lipsa serviciului de denumire globală și limitările lui Naming Service din NetWare, transformă administrarea unei rețele vaste într-o corvoadă zilnică.
- LAN Manager are cea mai bine gîndită interfață cu utilizatorul. Ea este ușor de folosit și constituie condue bine spre epoca GUI. Ca și la NetWare, serviciul de numiri este slab; mai ales în domeniul accesului independent de amplasare a datelor și programelor.
- LAN Server are cele mai bune facilități de administrare generală, în special din cauza serviciilor sale de numire. Inconvenientul lui LAN Server este că cere o stație de lucru OS/2 pentru administrarea rețelei.

Howard Marks

(Traducere și adaptare:Cristian Nagy)

ARTISOFT

O revoluție în conectarea calculatoarelor

În antiteza față de rețelele "clasice" tip client-server (având ca cel mai cunoscut și bine situat pe piață firma NOVELL, cu produsul NetWare), unde exista un centru unic în jurul căruia gravitează toate stațiile de lucru, rețelele de tip "egal-la-egal" sînt mult mai democratice, fiecare utilizator putînd deveni, în extremis, un server care să servească pe alții.

de mică, prețul deosebit de scăzut și extrema simplitate în utilizare.

Aceste tipuri de rețele se pretează foarte bine pentru un număr mic spre mediu de utilizatori. Inițial rețele locale modeste, pe zi ce trece dezvoltă facilități care le apropie de rețelele tip client-server (în privința performanțelor tehnice) și de rețelele de arie largă WAN (în ce privește facilitățile de comunicație).

pus de președintele ei: saltul de la statutul singular de "producător al rețelelor de tip LANtastic" la cel de producător de mijloace de comunicație (inclusiv multimedia).

Are însă acest succes financiar și un suport tehnic real, sau se bazează numai pe o "modă LANtastic"?

Avantaje

Ce avantaje ar oferi acest sistem radical diferit?

În primul rînd, se elimină necesitatea achiziționării obligatorii, de la început, a unui calculator server dedicat foarte performant din punct de vedere al vitezei, memoriei RAM și discului hard, ce este - implicit - și foarte scump. În al doilea rînd, o stație de lucru poate să-și aleagă serverul pe care dorește (după criteriile de performanță, dotări cu anumite periferice sau aplicații soft, spațiu de manevră pe disc, etc) să-și execute lucrarea la un moment dat, după cum poate să lanseze în execuție mai multe lucrări simultan pe servere diferite.

În același timp, fără a dezafecta nici măcar temporar rețeaua, se poate transforma chiar stația de lucru respectivă în server sau invers, după cum server poate deveni chiar și un AT286 "normal", ce poate îndeplini simultan și funcții de stație de lucru. Iată deci că unul din punctele forte ale rețelelor "egal-la-egal" este extrema flexibilitate.

Alte cîteva puncte de rezistență sînt memoria RAM necesară extrem

Firma

Cel mai dinamic reprezentant al producătorilor de rețele "egal-la-egal" și care se pare că a reușit să impună aproape un standard în domeniu este firma americană ARTISOFT, al cărei președinte (acționar principal, căci de un an de zile a devenit companie publică, cotate la bursă), Jack Schoof II, avînd deja aproape 35 de ani (!), a reușit să crească încasările firmei, bazate în principal pe vînzarea produsului lor LANtastic, numai între sfîrșitul lui 89 și prima jumătate a lui 91 de la 6,6 milioane \$ la peste 41 milioane \$, anunțînd închiderea anului fiscal 92 cu peste 73 milioane \$ încasări!

Cu o rată de aproape 10% din venituri dedicată publicității și reprezentării în fața clienților, cu fonduri foarte mari puse la dispoziția cercetării și proiectării, cu permanente contacte (inclusiv achiziționări de firme sau cooperări "bombă") în vederea străpunerii de zone în premieră absolută (ultima realizare fiind "puntea" între rețelele de tip DOS și cele LocalTalk de pe calculatoarele Apple), firma dovedește practic că luptă în mod real pentru atingerea scopului pro-

Cine ești dumneata, domnule Schoof ?

După ce la 14 ani a devenit (conform unor mărturisiri făcute de apropiați) programator profesionist, s-a apucat de montat calculatoare compatibile IBM PC ("clones") pentru prieteni, afacere ce s-a dovedit rentabilă, banii investindu-i în angajarea unui coleg de școală, Alex Karahalios (programator de excepție, actualmente vicepreședinte însărcinat cu dezvoltarea și cercetarea) și pregătirea începuturilor firmei. ARTISOFT s-a născut în 1982, iar primul lor produs a fost SpiderNet, un produs de paratajare a utilizării imprimantelor de către mai multe calculatoare. Produsul era revoluționar, născut puțin prea devreme, și cei doi au muncit destul ca să-l impună pe piață. Dar, recunosc și ei, nu a fost un succes comercial deosebit. Dar, avînd această experiență, începe munca deosebit de tenace de creare a rețelei LANtastic. În 1983, Schoof ia (după propria sa mărturisire) cea mai riscantă decizie din carieră, afectînd 10% din capitalul firmei pentru a plăti o pagină de publicitate în revista BYTE.

Din acest moment începe aventura LANtastic. În 1989, versiunea

3.0 aduce deja aproape 5 milioane de dolari încasări și se iau decizii mari: dezvoltarea versiunii 4.0, precum și producerea de componente hard proprii (adaptoare Ethernet și vocale) cu fiabilitate deosebită (garanția dată de ei este 5 ani!).

Rezultatul se poate vedea azi pe piețele întregii lumi, din Japonia și pînă în SUA, trecînd prin bătrînul continent.

LANtastic

Ce știe să facă LANtastic 4.0 (care este cel mai răspîndit, cel mai nou și cu 50% mai rapid fiind însă 4.1)?

Utilizînd circa 12 kB RAM pentru stațiile de lucru și 40 kB pentru server, este cel mai "mic" nucleu de rețea produs. El se încarcă direct peste sistemul de operare DOS, nu necesită formatarea dedicată a hard-discului și se poate dezactiva, fără resetarea calculatorului.

Avînd un sistem propriu de cache realizat soft, reușește să obțină performanțe de viteză foarte bune cu transmisie printr-o rețea Ethernet cu fir subțire de 10 MB/s, cu pînă la 300 utilizatori (primit de la început; nu există upgrade-uri!). Atît de bune încît a fost posibilă crearea unei versiuni sensibil mai ieftine, cu plăci adaptoare dedicate și transmisie prin fir toradat, viteza fiind de numai 2MB/s și pînă la 32 de utilizatori.

Sistemul de operare se încarcă de pe o singură dischetă (atît în cazul serverului, cît și a stației de lucru), instalare și "reglajul fin" al rețelei fiind simplificate la maximum și asistate de meniuri pop-up și HELP-uri (deocamdată, în limba engleză).

Există și posibilitatea creării de stații de lucru fără unități de discuri, încărcarea softului de rețea făcîndu-se în mod automat la pornirea calculatorului cuplat în rețea (prin adăugarea unui PROM special pe placa de adaptare): Se pot

astfel utiliza, fără rabaturi sesizabile pentru utilizatori, stații de lucru foarte ieftine la performanțe similare cu un PC complet.

Softul de rețea permite cele mai importante operații specifice unor rețele, de la transferul de fișiere și pînă la lucrul în mod multiutilizator, poșta electronică și chiar mesagerie vocală (direct sau prin înregistrări digitizate și stocate pe disc), realizate cu un adaptor special și un receptor telefonic livrat alături.

De asemenea, un lucru ce nu poate lipsi într-o rețea este securitatea datelor și a accesului la fișiere și aplicații. În acest doemniu, pe de o parte s-a dezvoltat softul necesar comunicării (prin portul serial) cu un UPS, în privința accesului putîndu-se face toate protecțiile cunoscute prin parolă, prin ora (perioada) de lucru și zona la care se permite accesul.

Există de asemenea aplicația care face o supraveghere globală a activității în întreaga rețea, putîndu-se ajunge chiar la contorizarea timpului de calculator consumat de fiecare utilizator, în vederea taxării.

Așa cum aminteam la început, utilizatorii pot alege (dacă li s-a permis accesul în zona respectivă) rularea aplicațiilor lor pe un server care le asigură cei mai buni parametri urmăriți (în general viteza de procesare).

Programul spooler de partajare a lucrărilor la imprimante asigură accesul și execuția lucrărilor la mai multe imprimante simultan.

Un punct de performanță deosebită este lucrul cu unități de tip CD-ROM, se pare cel mai "curat" utilizat pînă acum în rețele. Dacă unitatea este atașată unui server, ea poate fi accesată din orice punct al rețelei prin apelare ca un driver obișnuit.

Există compatibilitate pentru rularea de sub Windows ca aplicație DOS, dar se produce și o versiune ce apare ca aplicație Windows și

are toate avantajele ce derivă din aceasta. Lucrul sub Windwos se face atît în modul standard, cît și în cel real și în cel extins ale procesoarelor 286 și respectiv 386.

Activitatea de întreținere a rețelei este extrem de simplă și nu necesită nici pe departe angajarea expresă a unui inginer de sistem.

Manualul este bine gîndit și asigură găsirea rapidă a informației dorite, fără a încărcă memoria cu detalii inutile. Distribuitorii pot asigura școlarizări, dar de cele mai multe ori LANtastic se învață prin studiu individual și mai ales prin practica la calculator.

Transformarea într-un server dedicat se face printr-o simplă comandă DOS ("ALONE"), fără a fi necesară decuplarea din rețea sau resetarea întregii rețele. Un server poate gestiona pînă la 5100 de fișiere deschise simultan.

Ca o curiozitate, pe un server se pot lansa pînă la 4 rețele separate simultan!

Dacă se utilizează adaptoare AE-3 (compatibile NE-2000), se pot construi brațe de rețea cablate diferit (coaxial sau torsadat), fără ca, funcțional, rețeaua să sufere cu ceva.

Compatibilitatea pornește de DOS 3.1, soft-ul de rețea asigurînd blocarea la nivel de înregistrare în cazul accesului multiplu la fișiere.

Este perfect compatibil Net-BIOS, deci orice aplicație scrisă pentru rețea și care respectă acest standard va funcționa sub LANtastic fără nici un fel de probleme.

Un punct puternic este costul, prețul de vînzare (fără taxe) unanim aplicat în lume fiind de 669\$ pentru un starter kit ce conține softul pentru 300 de utilizatori, două plăci adaptoare Ethernet compatibile NE-1000, un cablu subțire gata mufat, în lungime de 8 metri, doi terminatori pentru capetele rețelei, două mufe T, manuale, dischete cu soft. Totul se poate monta în cîteva zeci

de minute (pentru cei foarte neîndemânatici), iar pornirea rețelei durează vreo oră pentru cei ce știu un pic de engleză și nu au văzut sau "pipăit" niciodată vreo rețea.

Distribuitorii de LANtastic au inițiat o ofertă mult mai bogată de versiuni, componente și accesorii de rețea.

În ce privește partea de conectare oferta ARTISOFT pornește de la sistemul propriu de 2 MB/s, continuarea cu sistemul clasic Ethernet cu cablu subțire de 10 MB/s (având adaptoare compatibile NE-1000 ISA și Micro Channel, precum și compatibile NE-2000), cea mai nouă realizare fiind gama completă de adaptoare și soft pentru cablarea rețelei pe standardul BASE-T (fir torsadat) în sistem Ethernet, ceea ce duce la simplificarea extremă a cablării rețelei.

O versiune interesantă este LANtastic Z, care face legătura doar între 2 calculatoare (de exemplu un desktop și un notebook), prin intermediul unui cablu serial sau paralel, oferind toate facilitățile rețelei "normale" LANtastic, la prețul de numai 125\$!

De asemenea, se furnizează o versiune LANtastic (numai soft), care se poate instala și cu adaptoare Ethernet produse de alte firme (există drivere pentru multe din cele mai răspândite tipuri).

O componentă de viitor este adaptorul vocal (ce cuprinde o placă și un receptor tip telefon), precum și softul aferent, ce permit deja, pe lângă mesageria vocală (inclusiv un fel de telefonie de uz intern prin intermediul rețelei) devenită deja clasică la Artisoft, inserarea de mesaje (fragmente) vocale în editoarele de texte, ceea ce reprezintă elemente de multimedia.

Noutăți

Dintre ultimele realizări ale ARTISOFT putem enumera:

- *LANtastic for Windows* - o versiune ce simplifică foarte mult configurarea rețelei, profitând în același timp din plin de grafica și modul de lucru standardizat al acestui mediu.
- *LANtastic for Netware* - se instalează peste sistemul de operare NetWare 2.1 (declarat "mort" de către NOVELL), extinzându-l cu pînă la 300 de utilizatori, la un preț incomparabil mai mic decît un upgrade NOVELL asemănător și cu performante aproape identice.
- *LANtastic for APPLE* - face o punte de transfer între rețelele compatibile NetBIOS și cele LocalTalk, permițînd accesul reciproc la fișiere, precum și accesul partajat la resursele de printare.
- *Communication Center* - este un procesor autonom (conține atît hard, cît și soft) pentru gestionarea comunicației pe canale seriale și modem/fax, precum și cu printare este modulul care permite cuplarea la distanță cu alte rețele.
- *The Network Eye* - este un utilitar ce face supravegherea generală a modului de utilizare a rețelei, precum și alocarea resurselor.

ARTISOFT oferă gama completă de accesorii de conectică pentru instalarea rețelei (mufe, cabluri, HUB-uri, etc.)

Spațiul nu ne permite, dar am putea prezenta o mulțime de extra-se din presa de specialitate care se referă la subiectul LANtastic. Multe firme, de la simple birouri pînă la centrale atomice, chiar firme ce produc ele însele tehnica de calcul (cum ar fi multinaționala NEC), au analizat piața de rețele și au constatat (chiar practic, prin achiziționare) că rețelele de tip LANtastic reprezintă cel mai echilibrat compromis între performanță (de altfel, în permanentă creștere), preț (deo-

sebit de rezonabil în cazul nostru), costurile de instalare și întreținere (aproape inexistente, pentru produsele ARTISOFT) și dificultatea utilizării (în cazul analizat în acest material, orice operator ce știe să dea comenzi în DOS este pregătit să facă același lucru și pentru rețea).

În opinia autorului, rețelele LANtastic constituie exact veriga care lipsește între dorința de evoluție către conectivitate a masei de calculatoare achiziționate în ultimii trei ani în România, și buzunarul destul de sărac al românului, dar fără a afecta funcționalitățile dorite.

Un principiu important în alegerea rețelelor spune: "aveți mare grijă să nu confundați dorințele dvs. relativ la rețele cu necesitățile reale ale aplicațiilor ce trebuie deservite. Păgubit va fi numai buzunarul celui ce plătește". Gîndiți-vă că buzunarul ce plătește e, totuși, al românului.

ing. OCTAVIAN CUDALBU

Un tabel de control pentru succes

Ce trebuie pretins de la un sistem de gestionare a desenelor/documentelor

Odată cu apariția rețelelor, beneficiile utilizării unei forme oarecare de sistem de gestionare a deseneilor/documentelor (DMS) devin mai clare. Utilizatorii de CAD și managerii sînt confrunțați cu o multitudine de factori de influență, incluzînd tehnologiile în schimbare, un număr în creștere de utilizatori, un set tot mai mare de desene și documente conexe, cereri ale clienților și o piață competițională într-un climat economic foarte strîns.

Fiind dați acești factori, cum ar trebui o întreprindere să procedeze în identificarea și implementarea unei soluții DMS?

Prima și a doua generație DMS

Așa cum am constatat în cazul altor aplicații (de ex. spreadsheet-uri și baze de date), sofisticarea și funcționalitatea programelor DMS evoluează. În lumea mediilor de dezvoltare a bazelor de date relaționale se aude tot mai frecvent de a patra generație de limbaje (4GL). Tot astfel, în noul domeniu al DMS-urilor a existat o primă generație, urmată de o a doua în plină dezvoltare

ca rezultat a doi factori critici: trecerea de la stații de lucru independente către configurații de rețea și capacitatea de lucru în rețea a AutoCAD Release 11, care oferă mecanismele de bază pentru protecția fișierelor în rețea.

Programele de gestionare a deseneilor nu sînt o noutate. Ele sînt deja disponibile de cîțiva ani, fiind caracterizate de funcții ca localizarea deseneilor, examinarea lor indiferent de programul CAD din care provin și forme simple de adnotare a lor (ca numele proiectului, descrierea, data ultimei utilizări etc.). De fapt unele dintre aceste sisteme timpurii erau denumite "de vizualizare". Ele erau în mod normal suficiente pentru utilizatori individuali operînd pe o stație de lucru izolată cu un număr ușor manevrabil de desene și nevoi reduse de partajare a acestora și a datelor conexe.

În cadrul activităților CAD în rețea apar probleme care nu se întîlnesc la nivelul PC. Cu siguranță, aplicațiile au fost dezvoltate în vederea rezolvării problemelor evidente de protecție a fișierelor și chiar de autorizare a utilizatorului concurent. Oricum, operați-

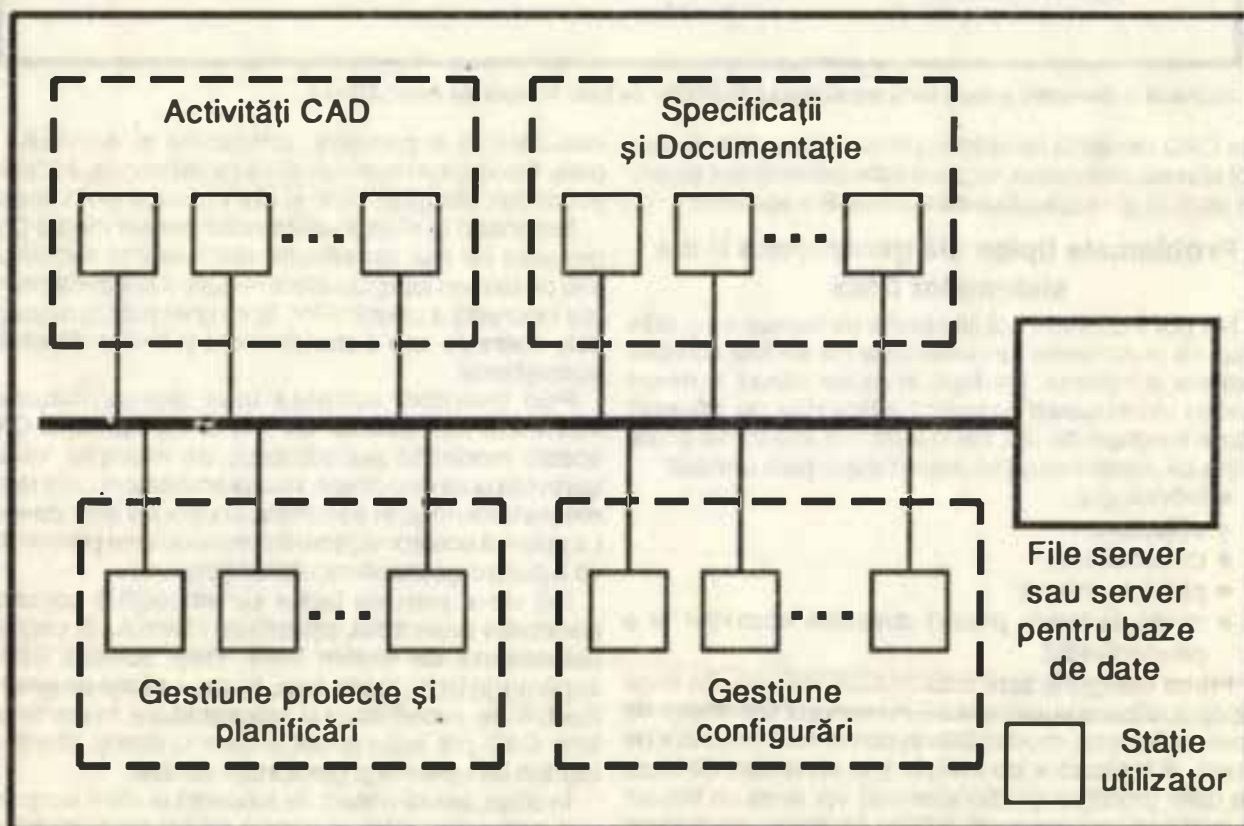


Diagrama 1: Sistemele DMS au oferit de-a lungul anilor funcții ca localizarea deseneilor, examinarea și extragerea acestora indiferent de programul CAD care le-a creat precum și forme simple de adnotare a lor.

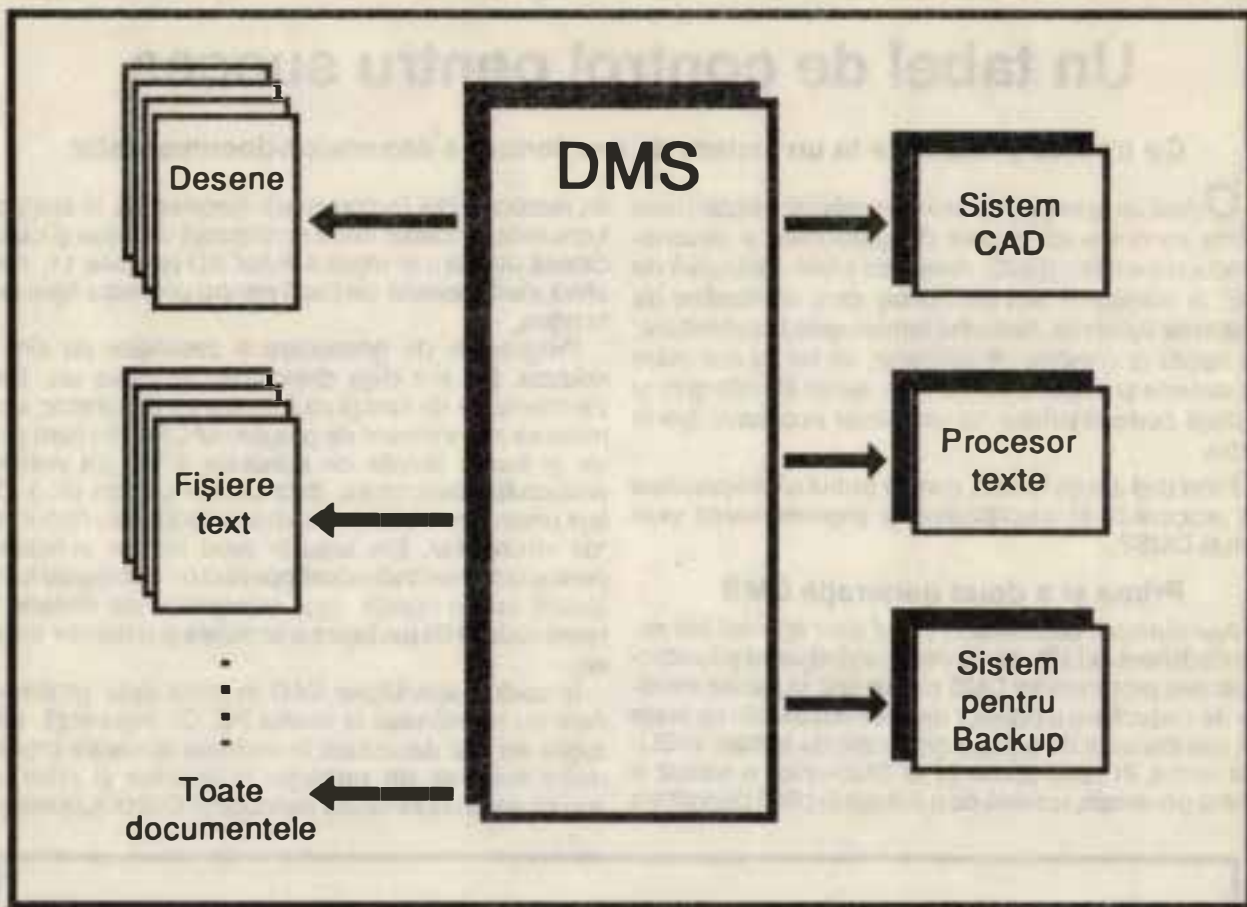


Diagrama 2: Generația a doua DMS are avantajul facilităților de lucru în rețea ale AutoCAD-ului.

nile CAD necesită facilități suplimentare cum ar fi control al versiunii/reviziei, legături între desen și documentul asociat și reactualizarea automată a acestora.

Problemele tipice ale generației a II-a a sistemelor DMS

Nu pot fi obținute soluții optime pe termen lung atâta timp cât problemele fundamentale nu au fost complet definite și înțelese. De fapt, în multe cazuri, a reveni pentru identificarea completă a factorilor de influență poate fi extrem de util. Într-o astfel de analiză se poate arăta că acești factori se împart după cum urmează:

- tehnologie;
- utilizatori;
- documente;
- proiecte, clienți;
- strategia firmei privind dinamica costurilor și a productivității.

Prima categorie este influența tehnologiei. Pe lângă funcționalitatea suplimentară în interiorul sistemelor de operare în rețea, modificările în domeniul memoriilor de masă, al legăturilor între rețele și al serverelor de baze de date (configurații client/server) vor avea un impact semnificativ asupra modalităților de acces, gestionare și control al desenelor/documentelor. Aceste tehnologii vor deschide larg porțile spre eforturi de

îmbunătățire a partajării, schimbului și accesului la date. Conceptul nostru de dată se va extinde, incluzând acum text, imagine, voce și alte mijloace multimedia.

Schimbări la nivelul utilizatorilor într-un mediu DMS pot avea cel mai semnificativ efect asupra succesului său pe termen lung. Un efect negativ îl pot avea instruirea incorectă a utilizatorilor, lipsa unei politici manageriale coerente sau a standardelor și liniilor directoare operaționale.

Prin definiție, editarea unui desen/document înseamnă modificarea iui. Într-o configurație CAD, aceste modificări pot influența, de exemplu, varianta/revizia unui document, relația acestuia cu alte documente/desene și, în particular, conținutul altor desene. La apariția acestor schimbări preocuparea primordială va fi pentru gestionarea ultimei versiuni.

Cel de-al patrulea factor de influență îl constituie parametrii proiectului, așteptările clientului și cerințele utilizatorului de sistem DMS. Deși aceasta este o împărțire în limite foarte largi, ea presupune un grad de flexibilitate, sensibilitate și adaptabilitate. În marile ateliere CAD pot exista proiecte pentru diferiți clienți, cu moduri de operare și gestionare variate.

În sfârșit, există o sferă de influență al cărei scop este reducerea costurilor pe seama măririi productivității, a optimizării și a atingerii unui nivel înalt al calității. Accentul în creștere pus pe rețele și arhitecturi de tip

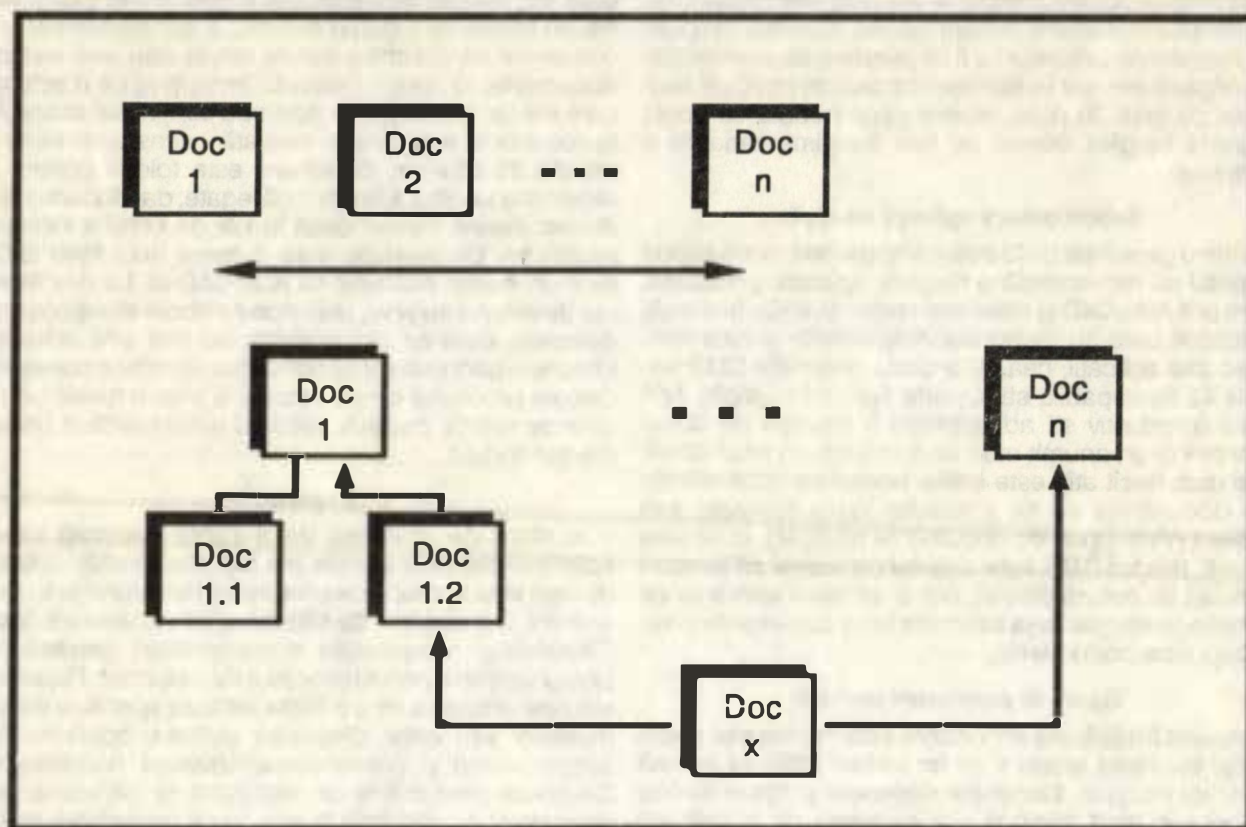


Diagrama 3: Într-un Sistem DMS, pe o relație orizontală, modificări într-un document vor afecta un altul. Pe o relație verticală, modificările documentului nu vor afecta în mod necesar un altul.

client/server nu lasă nici un dubiu că dimensionarea minimală acaparează rapid atenția managerilor. Aceste influențe ale tehnologiei, la care se adaugă mărirea independenței utilizatorilor, necesită o reevaluare a modalităților de atingere a scopurilor unei anumite întreprinderi.

Acestea sînt toate cerințe foarte importante influențînd orice mediu DMS. Oricum, se pot articula soluții răspunzînd nevoilor de astăzi ale unui mediu CAD în rețea și pregătind adaptarea la viitoare schimbări.

Soluțiile oferite de cea de-a doua generație

La succesul operațiilor unui sistem de gestiune a documentelor sînt necesare următoarele facilități:

Suportul oferit de o configurație în rețea.

Deși o aplicație poate opera eficient în rețea, există două zone funcționale importante care trebuie să fie disponibile într-un mediu DMS: posibilitatea de a bloca în prealabil accesul la un fișier și/sau înregistrare pe durata utilizării și, în al doilea rînd, de a permite drepturi suplimentare de acces în rețea. Capacitatea de protejare a fișierelor este primordială. Aceasta impune că, o dată ce un document a fost folosit într-o aplicație, alți utilizatori nu pot reface sau salva acel document pînă cînd utilizatorul care l-a refăcut anterior nu l-a eliberat.

În acest caz, sistemul de administrare a desenelor trebuie să prevadă o condiție de stare, pentru a identifica cine, cînd, ce și unde a accesat documentul. Ideea drepturilor suplimentare de acces în rețea implică posibilitatea ca un mediu DMS să poată fi configurat să creeze (nu să elimine) restricții ulterioare. De exemplu dacă un utilizator din rețea lucrează la proiectul A, atunci el poate accesa doar documentele asociate aceluia proiect, cu toate că drepturile lui de utilizare a rețelei sînt mai largi.

Caracteristicile documentelor, adnotări și informații descriptive

Oricărui document îi sînt asociate informații care descriu conținutul acestuia, modul lui de folosire, legăturile cu proiectul, precum și alte caracteristici. Aceste informații de tip text sînt externe documentului. Caracteristicile descriptive sînt nu numai comode cînd se trece în revistă "profilul" unui document, dar pot avea o valoare semnificativă în momentul în care se încearcă regăsirea documentelor pe baza anumitor condiții. Atributele documentului sau cîmpurile sînt folosite pentru a stoca aceste informații. Un sistem DMS ideal ar trebui să permită o combinație de cîmpuri predefinite și cîmpuri definite de utilizator. Această ultimă trăsătură este indispensabilă atîta timp cît fiecare companie are un sistem propriu de evidență a desenelor/documentelor. Numărul cîmpurilor definite de utilizator. Această ultimă trăsătură este indispensabilă

atîta timp cît fiecare companie are un sistem propriu de evidență a desenelor/documentelor. Numărul cîmpurilor definite de utilizator va fi dependent de necesitățile de organizare, dar în același timp acestea trebuie evaluate cu grijă. În plus, fiecare cîmp trebuie să poată suporta lungimi diferite de text (lungime variabilă a cîmpului).

Suport pentru aplicații multiple

Prima generație DMS este caracterizată de un suport capabil să recunoască o singură aplicație principală, cum ar fi AutoCAD și desenele respective. Cu mai mulți utilizatori care au responsabilități diferite și care folosesc alte aplicații, cea de-a doua generație DMS trebuie să fie capabilă să suporte aplicații multiple. Ar fi contraproductiv să administrezi o colecție de documente într-un anumit mod iar o alta într-un mod diferit. Mai mult decît atît, este foarte posibil ca tipuri diferite de documente să fie implicate într-o activitate sau proiect (Vezi Tipuri de documente multiple). În situația ideală, mediul DMS este capabil nu numai să lanseze aplicații de natură diferită, dar și să ofere abilitatea de a înțelege relațiile deja existente între documente (Vezi Relații între documente).

Tipuri de documente multiple

Această trăsătură importantă este consecința necesității enunțate anterior ca un sistem DMS să admită aplicații multiple. Deoarece desenele și documentele folosite în mod obișnuit sînt generate de o aplicație specifică, sistemul DMS trebuie să înțeleagă această legătură pentru a putea invoca aplicația corectă.

În mod similar, necesitatea de a suporta desene și documente produse în exteriorul calculatorului este critică dacă o organizație menține un număr mare de documente și desene pe hîrtie. Conversia acestora într-un format electronic este de cele mai multe ori nepractică și chiar imposibilă. Cu un sistem DMS capabil să suporte acest tip de document, toate celelalte funcții DMS, cum ar fi conexiunile din document, (vezi Relațiile dintre documente) pot funcționa.

Relațiile dintre documente

Cu un mediu DMS care asigură suportul pentru tipuri multiple de aplicații și documente, abilitatea de a specifica, menține și înțelege relațiile dintre documente devine un instrument puternic și important pentru aplicațiile generale de administrare a documentelor. Fișierul de referință externă din AutoCAD Release 11 permite utilizatorilor accesul la desenele existente fără o scriere fizică a entităților acestuia în structura unui alt desen. Avantajul, desigur, este că aceste fișiere de referință sau desene folosite în alte desene sînt definite o singură dată. Preocuparea evidentă privește posibilitățile DMS de a face cunoscut utilizatorului ce desene vor fi afectate dacă un fișier de referință va fi modificat sau șters.

Exemplul precedent care folosește fișiere de referință evidențiază unul din cele două modele de relații dintre documente ce pot fi aplicate.

Exemplul este privit ca un model "de conexiuni". În acest caz, un document este legat cu altul. Mai mult

decît atît, relațiile dintre ele pot fi directe sau indirecte. Într-un model cu legături directe,, o schimbare într-un document va schimba natura unuia sau mai multor documente. O relație indirectă înseamnă că o schimbare într-un desen poate schimba conținutul altora. Al doilea model este relația derivată - acesta consideră situația în care un document este folosit pentru a determina un altul. Ele sînt încă legate, dar documentul derivat devine invalid dacă în cel de origine intervin modificări. Un exemplu este crearea unui fișier DXF dintr-un desen executat cu AutoCAD-ul. La modificarea desenului original, utilizatorul trebuie atenționat că desenele derivate din aceasta nu mai sînt actuale. Metodele care realizează acest model derivat pot varia. Oricum problema critică este de a oferi o formă oarecare de reacție pozitivă, validînd sau invalidînd documentul derivat.

Raportul

În afară de abilitatea de a capta informații într-o aplicație, cea de a asimila și a raporta cantități diferite de date este aproape cea mai importantă funcție a unei aplicații. Capacitățile de căutare și de recuperare (vezi Căutarea și recuperarea documentelor) servesc ca precursori/filtre pentru procesul de raportare. Raportarea este abilitatea de a colecta atribute specifice documentelor sau valori cîmpurilor definite, operarea cu aceste valori și prezentarea formarea rezultatelor. Deoarece necesitățile de raportare și formatele pot diferi de la o companie la alta, vor fi prezentate patru categorii diferite de metode de raportare. O combinație a acestora poate fi utilizată pentru a asigura majoritatea necesităților de raportare. Acestea sînt:

- Ad-hoc: abilitatea de interogare în limba engleză.

- SQL (Structured Query Language): limbaj structurat de interogare - similar abordării Ad-hoc dar care folosește un standard industrial. Puterea sa constă în posibilitatea abordării Ad-hoc dar care folosește un standard industrial. Puterea sa constă în posibilitatea de a interoga mai multe fișiere de date la fel de bine ca și servere pentru baze de date.

- QBE (Query By Example): se concentrează asupra prezentării vizuale a coloanelor (cîmpurilor) disponibile, permite selecția cîmpurilor pentru a fi afișate și specifică criteriile condiționale dorite. Avantajul acestei abordări este că "se poate vedea ceea ce se va obține" ("see-what-you-will-get").

- Programe de scriere de rapoarte: tind să combine o abordare QBE cu un domeniu larg de opțiuni de formatare. Unele vor include funcționalitate SQL în care funcțiile ad-hoc sînt transparente utilizatorului final.

Facilități de căutare și recuperare a documentelor

Măsura critică a oricărui sistem DMS este abilitatea de a găsi și recupera documente indiferent de locație, nume și orice alte caracteristici ale lor (vezi Caracteristicile documentelor). Abilitatea de căutare trebuie să admită un număr nelimitat de clauze condiționale și operatori folosiți la crearea criteriilor de căutare. Clauzele condiționale sînt condițiile "AND" și "OR" ("ȘI"/"SAU" condițional) al unei operații. Operatorii includ funcțiile Booleene cum ar fi egalitatea, mai mic,

mai mare, incluziunea, apartenența la un domeniu etc. Criteriul ideal de recuperare îl reprezintă abilitatea de căutare pe o condiție de "configurație fonetică asemănătoare". Trebuie găsit de exemplu un document cu numele apropiat de "TWO". Rezultatul poate fi reprezentat prin șirurile de caractere următoare "2", "TO", "TOO" și desigur "TWO".

Controlul evoluției etapelor de prelucrare a unui document

Urmărirea și prezentarea istoricului activităților legate de un document va fi unul din instrumentele manageriale cele mai puternice pentru reducerea costurilor și creșterea productivității și calității. Istoricul unui document ar trebui să includă data, durata, utilizatorul, stația de lucru și activitățile respectiv operațiile executate asupra documentului. Activitățile ce vor fi înregistrate trebuie să conțină accesul, modificările, copierea, mutarea, redenumirea, printarea, plotarea, etc. În plus, poate fi util ca schițele și notele scrise cu mâna să fie

asociate acestor activități la dorința utilizatorului. Cu posibilități de configurare după dorință, pot fi integrate și alte aplicații conexe, ca estimarea timpului de lucru, evidența materialelor și a costurilor, analiza în timp a fluxului tehnologic sau suportul de gestiune a configurațiilor.

Medii configurabile de utilizator

Cu cât mai mulți utilizatori sînt legați la o rețea cu atît va crește gama de discipline, funcții, considerații de acces și responsabilități. Ca o consecință, utilizatorii pot întîlni desene, proiecte, coduri și convenții precum și alte colecții operaționale de date care sînt irelevante și/sau care nu sînt cerute de propriile sarcini. Oferind posibilitatea de configurare a mediului, operatorii au acces la informații specifice sarcinilor și ariei lor de responsabilități. Aceste "sesiuni" pot defini aplicația implicită curentă, locul în care sînt salvate desenele, tipul documentelor încărcate la pornire, condițiile de acces și restricțiile.

TABEL DE CONTROL PENTRU EVALUAREA DMS

Caracteristici/Funcții	Pondere	Produce A		...	Produce n	
		Notă	Scor		Notă	Scor
Exemplu	8 ^{II}	9 ^{II}	72		6 ^{II}	48
Suport pentru rețea						
Caracteristici document						
Suport pentru aplicații multiple						
Tipuri de documente multiple						
Relații între documente						
Raportare						
Căutare și extragere documente						
Control evoluție prelucrare doc.						
Medii configurabile de utilizator						
Convenții denumire documente						
Verificare intrări/ieșiri						
Planificare tehnologică						
Arhivare						
Sisteme de gestiune baze de date						
Utilizare surse de date existente						
Adaptare de către utilizator						
Elemente specifice utilizatorului						
Element 1						
Element 2						
Element 3						
SCOR TOTAL						

II Folosiți valorile 0-10 unde 10 reprezintă valoarea maximă acordată ponderii importanței acordate unei caracteristici DMS respectiv notei de evaluare a produsului.

Tabel 1. Alegerea DMS-ului potrivit.

Evaluînd corect factorii și cerințele prevăzute în acest tabel, managerul de rețea poate lua o decizie, bazată pe condiții exacte, derivate din necesitățile companiei referitor la sistemul de gestiune a documentelor DMS.

Convenții de denumire a desenelor/documentelor

O limită a DOS-ului stabilește numărul de caractere ale numelui unui fișier la opt. Pe de altă parte, multe sisteme de organizare folosesc coduri ce depășesc această limită de opt caractere. În astfel de cazuri un sistem de organizare DMS trebuie să înlăture acest neajuns. Ca o extensie la aceasta, este capacitatea unui mediu DMS de a începe un nou document prezentînd automat utilizatorului nume implicite succesive, într-o secvență controlată. Aceasta ar fi soluția ideală pentru sistemul de codificare al propriei firme,, desene multiple cu număr de pagini secvențial, etc.

Verificarea intrărilor/ieșirilor

În mediile în care modificările desenelor/documentelor nu pot fi făcute în mod arbitrar, trebuie să existe metode de verificare a intrărilor și ieșirilor. Verificarea intrărilor/ieșirilor este în primul rînd procesul de plasare a unei copii a unui document într-un spațiu de lucru în scopul trecerii în revistă și/sau modificării, de posibilă trecere spre alt utilizator pentru verificare/aprobare și apoi replasare a documentului în zona de stocare inițială. Acest proces trebuie să prevină inadvertențele sau schimbările nedorite și să controleze accesul simultan la aceiași documente de către mai mulți utilizatori. Indiscutabil, semnarea și aprobarea unui document este un proces lipsit de rigoare, standarde și convenții. Unele dintre aceste lipsuri apar datorită deficiențelor tehnologice actuale.

Planificarea tehnologică, programarea activităților și conducerea evenimentelor

Fluxul tehnologic și planificarea sarcinilor de lucru a fost multă vreme o aplicație separată, situată adesea la nivelul aplicațiilor de conducere a proiectelor. Oricum, cu mai mulți utilizatori operînd în interiorul unei rețele, această funcțiune poate fi un avantaj foarte important într-un mediu DMS. Cînd documentele parcurg o serie de stadii este ideal să ai posibilitatea să repartizezi/planifici o activitate pentru un anumit operator sau document. De fapt cele mai multe organizații au deja conturate și stabilite stadiile și operațiile. Aceasta poate include atribuirea de sarcini specifice individuale sau atribuirea globale în scopul analizării. De exemplu, după terminarea unui desen, acesta poate fi înaintat pentru operația de verificare/analizare. Ca și în schimbul de mesaje dintr-o rețea, atribuirea va fi notificată ca o activitate nerezolvată. Această notificare poate apărea ca un mesaj pe ecran sau sub forma unei liste de evidență a activităților ce trebuie făcute. Această trăsură asigură desenelor integritatea; identifică în ce stadii se află desenele, cine le are și cît timp au rămas în anumite stadii permite o mai bună înțelegere a operațiilor CAD și a altor resurse (plottere, dispozitive pentru efectuare backup-ului, etc).

Arhivarea

Odată ce o colecție de desene sau documente pentru un anumit proiect este completă o operație de arhivare poate elibera spațiu pe disc prin înlăturarea informației care nu mai este actuală. În unele cazuri, după arhivare poate apărea necesitatea de a restaura

aceste documente precum și datele conexe pentru o actualizare a informației. Asociate unui desen sau document gestionat într-un mediu DMS vor fi adnotarea informației, istoricul, legăturile posibile cu alte documente. La arhivare trebuie mutat nu numai documentul actual, ci și datele asociate acestuia.

Sisteme de gestiune a bazelor de date DBMS

Funcțiunile de bază ale unui DMS - gestiunea documentelor și datelor asociate acestora, înzestrarea cu opțiuni extinse de interogare și raportare și asigurarea platformei pentru mai mulți utilizatori - presupun utilizarea unui sistem de gestiune a bazelor de date (Database Management System - DBMS). Platforma DBMS este temelia unei soluții DMS. Deși există alte abordări, ele tind să fie soluții bine determinate și sînt proiectate în general pentru o aplicație strict specifică (sisteme dedicate), cu flexibilitate minimă. Pe măsură ce cerințele unei organizații se schimbă și tehnologiile hardware și software se îmbunătățesc, aceste abordări strict specifice vor fi rapid abandonate, atingîndu-și limitele. Deși așa numitele modele de baze de date "orientate pe obiect" asigură o comprehensibilitate superioară, DBMS-ul relațional rămîne în mod cert modelul ce trebuie ales.

Utilizarea surselor de date deja existente

Un sistem DMS va fi rar implementat într-un mediu în care nu există date definite anterior care să se constituie ca suport pentru controlul și gestiunea documentelor. Acestea pot fluctua de la un simplu sistem de numerotare a desenelor pînă la o aplicație complexă cum ar fi un mediu de gestiune complet configurat. În orice caz aceste surse de date existente pot constitui un avantaj ajutînd la menținerea unei investiții primare în organizarea structurilor de date existente și în aplicațiile asociate lor.

Modul ideal de abordare ar fi abilitatea mediului DMS de a citi și scrie în aceste surse de date externe. Aceasta elimină nevoia importului/exportului de date și asigură că datele folosite sînt cele mai recente. Unele sisteme oferă posibilitatea de scriere/citire din programele obișnuite de baze de date și calcul tabelar.

Adaptarea organizării DMS la cerințele utilizatorului

De cele mai multe ori ceea ce doriți să faceți astăzi nu coincide cu ceea ce veți dori ulterior. Cum am spus mai sus, există prea multe influențe pentru ca marea majoritate a aplicațiilor să rămînă nemodificabile. Nevoia de a adapta o anumită organizare va fi determinată de imposibilitatea ca aceasta să rămînă în limitele unui sistem cu arhitectură închisă. Adaptabilitatea înseamnă abilitatea de a modifica meniul, fereastra pentru introducerea datelor și operațiilor precum și raportarea. Managerul rețelei trebuie să se informeze ce funcții adiționale pot fi dorite/cerute mai tîrziu, și dacă mediul DMS poate fi adaptat la modificări ulterioare.

Direcții de orientare: abordare practică

Acum după ce direcțiile de orientare au fost stabilite, vom sugera un model de lucru (tabelul 1) pentru cei care intenționează să înceapă sau sînt implicați în identificarea și alegerea unui sistem DMS pentru propria întreprindere. Cu toate că există multiple elemente ce trebuie luate în considerare, unele pot fi mai puțin importante sau chiar inutile. În plus, pot apare și alte considerații importante ce trebuie reținute. Folosind tabelul 1, atribuiți o pondere corespunzătoare importanței fiecărui element precizat și adăugați alte elemente specifice necesităților proprii. Odată stabilite aceste

ponderi, evaluați versiunile sistemelor DMS candidate, comparînd cu rezultatele din tabel ("checklist"). Puneți o notă fiecărei componente evaluate și înmulțiți-o cu ponderea atribuită. Odată obținut totalul pentru fiecare sistem, aveți deja determinat cel mai bun sistem corespunzător necesităților, după luarea în considerație a costului, specificul activității, posibilităților de instruire și a altor factori.

Traducere și adaptare după articolul "A checklist for success" publicat de Martin M. van der Roest în revista CADENCE realizate de Vincențiu Popescu

Rezumat

Spre deosebire de alte aplicații ce produc diverse forme de rezultate tangibile, un sistem DMS aduce beneficii sub forma creșterii eficienței și productivității. Pe de altă parte dacă sistemul este inadecvat sau impropriu implementat rezultatele pot fi foarte costisitoare.

Acest tabel de control este similar cu stabilirea unui plan de vacanță. Stabiliți mai întii ce doriți să faceți, itinerariul, etc. Rezultatele unui plan reușit se vor materializa într-o plăcută și memorabilă experiență. În același mod, cu o atenție evaluare și implementare, un sistem DMS vă va produce profituri pe termen lung. Astfel veți avea posibilitatea să vă concentrați asupra afacerilor și vă va permite să vă ocupați de ceea ce știți cel mai bine.

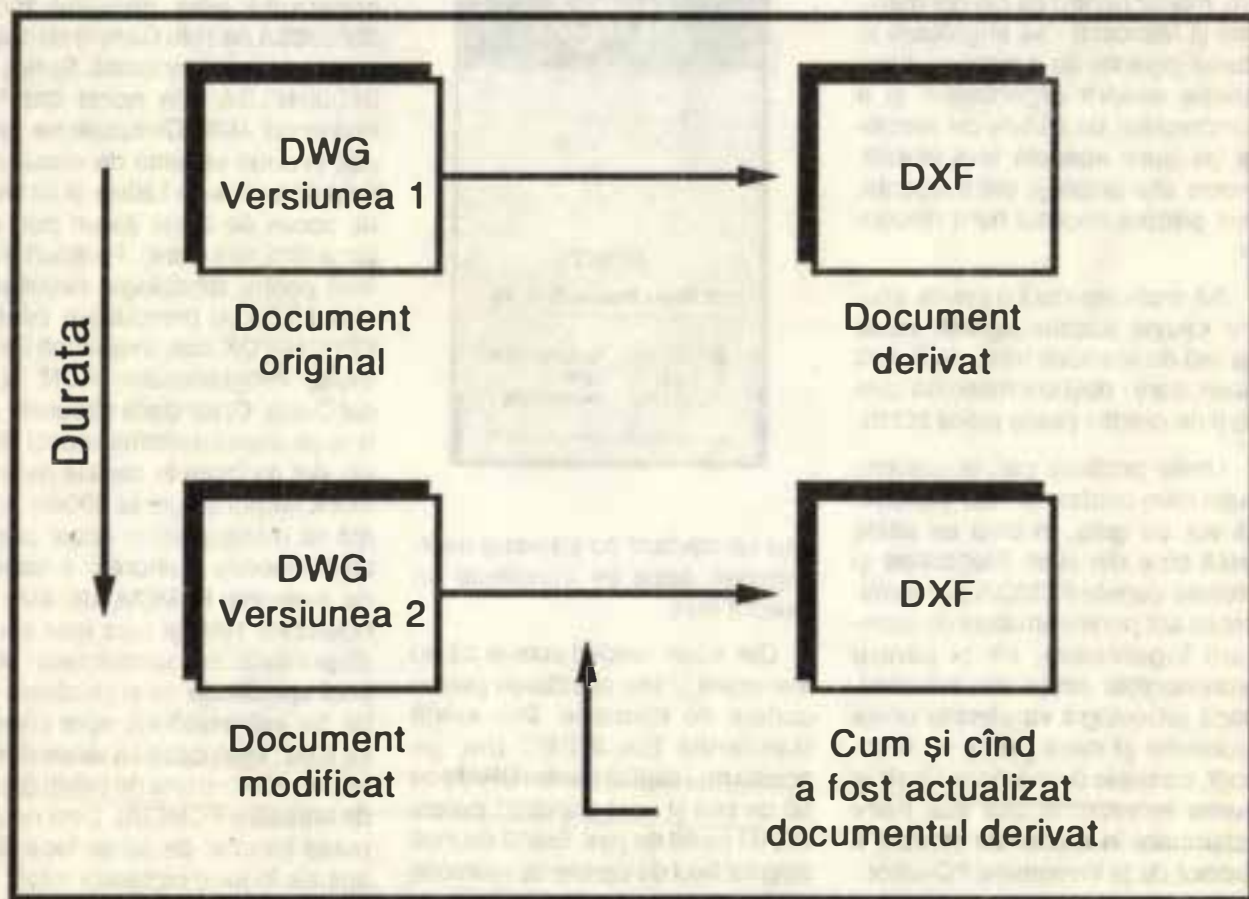


Diagrama 4: Într-o relație derivată, sistemul DMS determină modul în care un document derivă dintr-altul, cum ar fi, de exemplu, un fișier DXF obținut dintr-un desen DWG realizat în AutoCAD. Într-un astfel de model, documentul derivat devine invalid dacă desenul original este modificat.

Flash stîrnește senzație

Cartelele de memorie Flash produse de Intel sînt compacte și rapide. Datorită propriului lor standard internațional și datorită marelui interes manifestat de profesioniști, ele ar putea vesti sfîrșitul hard discurilor.

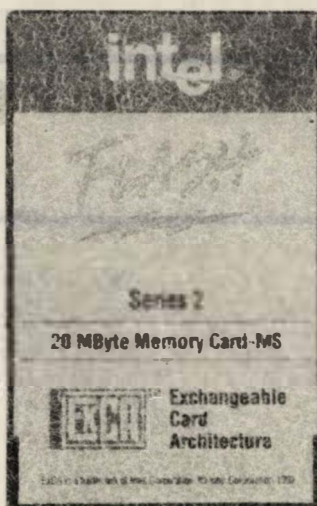
PCMCIA înseamnă Personal Computer Memory Card International Association (Asociația Internațională pentru Cartele de Memorie pentru Calculatoare Personale). Despre această melodioasă abreviere veți mai auzi multe în viitor, dacă nu din alte motive, măcar pentru că cei doi mari - Intel și Microsoft - se angajează în efortul gigantic de a atrage tuturor atenția asupra organizației și a standardelor de cartele de memorie pe care aceasta le-a stabilit. Printre alte pretenții extravagante, Intel prezice decesul hard discurilor.

Să aruncăm deci o privire asupra cauzei acestei agitații: noua cartelă de memorie Intel, denumită Flash, care - deși are mărimea unei cărți de credit - poate stoca 20Mb.

Unele produse par, la naștere, puțin cam ciudate și dau impresia că vor da greș, în timp ce altele arată bine din start. Elegantele și zveltele cartele PCMCIA par formidabile atît pentru amatorii de accesorii ingenioase, cît și pentru profesioniștii cinici din industrie. Dacă tehnologia va elimina unele probleme și dacă prețul va fi coroborît, cartelele de memorie Flash ar putea reprezenta cea mai mare schimbare în modul de stocare a datelor de la inventarea PC-urilor.

Deși PCMCIA este o interfață standard, standardele se modifică. PCMCIA este o organizație a unor fabricanți cu aceleași intenții care

s-au întîlnit în S.U.A. în august 1989 pentru a se pune de acord cu privire la un standard pentru cartele de memorie. În același timp, un grup din Japonia (fiți pregătiți pentru alte inițiale) numit JEIDA (Japanese Electronics Industry Development Association) are propriile sale idei. Deși este uimitor pentru industria computerelor, cele două organizații au căzut de acord asupra unui standard comun în septembrie 1991, și anume PCMCIA 2.0/JEIDA 4.0. Deci avem



deja un standard cu adevărat internațional, ceea ce constituie un început bun.

Dar să nu credeți cumva că nu mai există și alte standarde pentru cartele de memorie. Măi există standardul EIA/JEDEC (nu, pe acesta nu-l explic) pentru DRAM cu 60 de pini și noul standard pentru DRAM cu 88 de pini. Există de mult timp tot felul de cartele de memorie și, în 1990, s-au vîndut circa un milion. Entuziaștii PCMCIA (poate optimiștii) estîmnează că vînzările să explodeze la 100 de milioane de unități în 1995.

Cartelele normale PCMCIA de Tip 1 au 3,3 mm x 85,64 mm x 54 mm. Cartelele de tip 2 mai puțin răspîndite, au grosimea de 5 mm și celelalte dimensiuni identice. Pe una din laturile de 54 mm se găsește un conector mamă cu 68 de pini (ambele cartele au grosimea de 3,3 mm pe această latură). De obicei, cartela se inserează într-un dispozitiv și se împinge la locul ei. Pentru a o extrage, este nevoie de o sculă specială.

Noua cartelă de memorie Flash se livrează în trei variante: 4,10 și 20Mb. "Cărămidă" cu care este construită este circuitul Intel 28F008SA de 1Mb. Cartela cu chipul inclus este denumită Series 2 IMC004FLSA - în acest caz "4" înseamnă 4Mb. Circuitele se produc în două variante de viteză: cu timp de acces de 120ms și cu timp de acces de 85ms (la un preț cu circa 30% mai mare). Prescurtarea Intel pentru tehnologia circuitelor este ETOX, o prescurtare pentru EPROM-TOX, care înseamnă Electrically Programmable ROM Tunnel Oxide. Chiar dacă circuitele au timp de acces de 85ms, atunci cînd ele sînt montate în cartela de memorie timpul crește la 200ms. Merită să menționăm în acest punct că Mitsubishi a anunțat o cartelă de memorie PCMCIA de 4Mb în noiembrie 1991 și care este acum disponibilă cu aproximativ aceleași specificații ca și produsul Intel. Se estimează că, spre sfîrșitul lui 1992, Mitsubishi va vinde o cartelă de 20Mb și una de 64Mb (limita de adresare PCMCIA). Deci ne-am putea întreba: de ce se face atîta agitație în jurul cartelelor Intel? Se pare că Intel știe cum să producă senzație cu Flash.

Care este diferența între această memorie EPROM "Flash"

și DRAM sau SRAM cu care sînt obișnuiți utilizatorii de PC? Afiți RAM-ul dinamic (DRAM) cît și RAM-ul static (SRAM) au nevoie de energie pentru a-și păstra conținutul și de aceea ele sînt denumite "volatile". De fapt, DRAM-ul are nevoie de operații constante de reîmprospătare ca și un condensator cu pierderi.

În cazul cartelelor non - EPROM această energie este furnizată de o mică baterie. EPROM-ul nu are nevoie de energie pentru a păstra datele, motiv pentru care este denumit "nevolatil", și este - în mod evident - util ca mediu de stocare. După cum ne aducem aminte, nici discurile nu au nevoie de energie pentru a păstra datele - dar ea le este necesară pentru citire sau scriere. Termenul "Flash" înseamnă că datele din circuitul EPROM pot fi modificate pe loc, spre deosebire de EPROM-ul uzual care trebuie să fie scos din calculator și introdus într-un dispozitiv special pentru a fi reprogramat. Pe lîngă proprietatea de nevolatilitate, noile cartele de memorie - mai ales cele de 20Mb - sînt mult mai compacte decît uzualul DRAM SIMM care se găsește în cele mai multe calculatoare.

În pofida numelui său, interfața PCMCIA propriu-zisă nu acceptă numai cartele de memorie. Mulți producători au intuit posibilitățile acestui port care și-a găsit locul în multe mașini și va exista, se pare, în mult mai multe. Deja există pe piață cartele moderne PCMCIA, unul dintre producători fiind chiar Intel. Au mai fost oferite cartele pentru rețea și este în curs de apariție chiar și o interfață de imprimantă PCMCIA. Și de ce nu? Acest tip de interfață a fost caracterizată ca "magistrala de calcul portabilă a viitorului". Nu mai este nevoie să spunem că Intel se află în frunte cu al său ExCa Exchange Card Architecture, un standard pentru drivere de dispozitiv pentru conectorul PCMCIA folosit, de exemplu, de modemuri. Este posibil ca prima

întrebare pusă relativ la o nouă tehnologie să devină: "Se potrivește cu o cartelă PCMCIA?".

Prezența lui Microsoft subliniază potențialul și viitorul progres ale noilor cartele Flash. Menținîndu-se întotdeauna în zona profitului, gigantul a intrat în scena PCMCIA cu Flash File Systems versiunea 2.0 proiectat special pentru EPROM-ul Flash.

Deci, se pare că dispune de aceste minunate dispozitive de mărimea unei cartele de credit care are mari spații de stocare, care nu au nevoie de energie pentru a păstra datele, care sînt mici și care pot fi scăpate pe jos fără riscuri. Ele pot face accesul la date de mii de ori mai rapid decît discurile, au un standard care să elimine redundanțele, și beneficiază de implicarea lui Microsoft. Cu siguranță că discurile hard au murit.

Nu este chiar așa. Prima problemă o constituie prețul. Intel cere 369£ pentru o cartelă de 20Mb, ceea ce înseamnă 18,45£/Mb. Să menționăm că trebuie să se cumpere mii de bucăți pentru a le obține la acest preț. Ultimul disc Fujitsu de 3,5" în de 1082Mb cu timp de acces de 10ms și rată de transfer de 10Mb/sec poate fi cumpărat cu circa 2000£, ceea ce înseamnă 1,85£/Mb - o zecime din preț. Un disc cu tehnologie mai slabă dar mai ieftin, discul Seagate de 1,4Gb, revine la 1,30£/Mb. Deci, în locul discurilor, Flash este foarte scump. Este tot de interesant să comparăm densitatea datelor cartelei PCMCIA - care pare un avantaj major - cu densitatea datelor pentru discuri. Cartela Intel de 20Mb oferă 1,3Mb/cm³ în timp ce discul Fujitsu, un dispozitiv cu înălțime standard, oferă 1,8Mb/cm³. Tehnologia veche cîștigă din nou.

Desigur, discurile consumă mai multă energie în funcționare decît cartelele Flash, iar în cazul PC-urilor alimentate de la baterii consumul este un aspect crucial. Citirea de la cartele cere puțină energie,

dar cînd se începe scrierea într-o cartelă EPROM Flash încep să apară problemele și nu numai în domeniul consumului. Noile cartele Intel folosesc două tensiuni: 5V pentru citire și 12V pentru scriere/ștergere. EPROM-ul trebuie șters înainte a fi înscris și acest proces este atît consumator de energie (pînă la 90mA), cît și foarte lent. Iar utilizarea înrăutățește performanțele. Într-un caz defavorabil al unei cartele vechi (intens folosite) se poate aștepta circa trei minute pentru ștergerea și scrierea a 4Mb; însăși operațiunea de ștergere și citire produce uzura cartelei de memorie Flash. Ștergerea blocurilor de date din circuite poate să aibă loc cu scrierea în altă zonă dar situația se înrăutățește ori de cîte ori se ciclează blocul.

Aici intervine Microsoft cu al său Flash File System. Problema "uzurii" a fost recunoscută de fabricanți și o mare cantitate de inteligență a fost utilizată pentru a ține urma blocurilor folosite și a numărului de ciclări. Această logică utilizează "algoritmi de fixare a nivelului de uzură" și ar putea diferi de la un fabricant de cartele la altul.

Standardul PCMCIA presupune că la începutul cartelei se află un antet care conține informații specifice cartelei și care se numește CIS (Card Information Structure). CIS conține cele mai importante detalii referitoare la tipul cartelei cum ar fi tipul circuitului (Flash, SRAM, etc), structura internă, modul de adresare (cum ar fi la octet și acces aleator) și alte informații pentru interfață. De asemenea, CIS mai poate conține adresa unei zone opționale de memorie cu "atribute" care pune la dispoziție informații de fabricant și mai amănunțite despre organizarea cartelei. Aceste informații sînt concepute pentru a permite o utilizare cît mai bună a diverselor abordări folosite de diverși producători care folosesc aceeași tehnologie de bază.

Dacă scumpele cartele Flash se uzează prin utilizare, cum se poate ști când sînt complet uzate? În momentul de față, tehnologia cartelei măsoară durata ciclului de ștergere/scriere și decide dacă este prea lungă. Dacă nu s-a întîmplat nimic în intervalul de timp dat, procesul este oprit și repetat. Dacă după "n" repetări nu se întîmple nimic, cartela este "moartă". Se poate presupune că, cu mult înainte ca aceasta să se întîmple, cartela raportează către Microsoft File System care, la rînsul său, îl informează pe utilizator despre epuizarea și necesitatea înlocuirii cartelei din conectorul "A".

Deși cartelele de memorie Flash sînt considerate "periferice", ele au posibilitatea de a fi folosite ca ROM suplimentar (sînt prea lente pentru a fi folosite ca RAM suplimentar). În cartele pot fi încărcate aplicații și se folosește o specificație cunoscută sub numele de XIP (eXecute In Place), conform căreia un program este rulat direct de pe cartelă - fără a-l încărcă mai întîi în memoria PC-ului. Aceasta prezintă avantajul evident de a elibera RAM prețios dar mai înseamnă că și conectorul PCMCIA este ocupat la o singură aplicație.

Deci vestesc aceste noi cartele Intel Flash PCMCIA sîrșitul erei discurilor hard așa cum le cunoaștem? Astăzi răspunsul este nu. Dar împotriva neajunsurilor temporare, aceste cartele au proprietăți remarcabile și - pentru anumite aplicații - de utilitate imediată. Și, ceea ce este mai important, combinația între succesul de piață al cuplului Intel/Microsoft și forța industrială a lui Mitsubishi promite un viitor mare acestui mic dispozitiv electronic.

O privire asupra procesării

Over Drive

Circuitele pentru dublarea frecvenței de ceas pot ridica viteza globală a PC-urilor bazate pe 486 cu aproape 100%. Vă prezentăm modul lor de funcționare și aspectele care trebuie avute în vedere.

Dacă sînteți puțin nedumeriți în legătură cu opțiunile de îmbunătățire a procesoarelor, nu sînteți singurul. În timp ce Intel cheltuiește milioane de dolari pentru a evidenția - în reclame - un soclu gol, iar cei care vînd sisteme laudă tot felul de mașini modulare și care pot fi dezvoltate, nu sînt puțini utilizatorii de PC-uri care își sparg capul încercînd să-și dea seama ce să caute.

Căutînd să crească puterea, Intel a încercat să reducă dimensiunile chip-ului 486 și să mărească frecvența de ceas la peste 50MHz. Dar chiar dacă frecvența circuitului se poate apropia de 100 MHz, restul sistemului - chiar și pini care conectează circuitul la placă - se confruntă cu tot felul de probleme la 50MHz și mai mult. De asemenea, eliminarea interferențelor cu frecvențele radio devine mult mai dificil de realizat la frecvențe ridicate ale ceasului.

Cache - dacă se poate

Din fericire, memoria cache inclusă în chip-ul 486 oferă o soluție care permite să se lucreze cu sistemul la frecvențe de ceas mai mari. În cea mai mare parte a timpului, procesul plasează instrucțiunile și datele de care are nevoie în memoria cache din chip. Deoarece memoria cache lucrează independent de restul sistemului, ea permite procesorului să lucreze la o frecvență de ceas mai mare decît a restului sistemului.

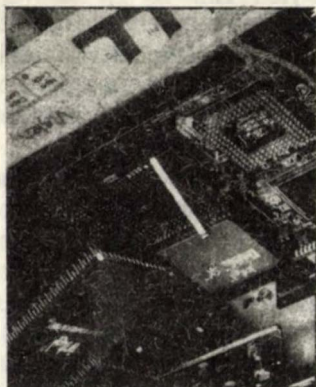
Pentru a exploata această posibilitate, Intel a creat o versiune de 486 cu un circuit de dublare a frecvenței de 25MHz, îi permite să lucreze la 50MHz ori de cîte ori nu este necesar accesul la lumea exterioară. Intel va oferi aceste circuite pentru sistemele cu 486SX și 486DX lucrînd la frecvențe între 16MHz și 33MHz.

Circuitul de dublare a frecvenței ceasului promite o creștere a vitezei cu pînă la 100 la sută dacă aplicația este strîns legată de unitatea centrală și dacă datele încap în memoria cache. Pe de altă parte, cu cît aplicația folosește mai mult discul, scrie mai mult la ecran sau utilizează mai des structuri de date care nu încap în memoria cache, cu atît va fi mai coborît cîștigul de viteză. Intel pretinde că utilizatorii aplicațiilor populare de birou vor vedea îmbunătățiri cu 62% pînă la 77%.

Circuitul de dublare a ceasului trebuie privit ca un 486 turbo. Circuitul se montează în același soclu ca și 486 standard, iar semnalele de pe pini sînt aceleași. Diferența esențială o constituie voicuiunea pe care o dă sistemului: el lucrează în mod turbo - la o viteză dublă atunci cînd datele și instrucțiunile necesare se găsesc în memoria cache.

DX2 și OverDrive

Circuitele cu dublare a frecvenței de ceas vin numai în versiuni 486DX - cu alte cuvinte, cu coprocesor matematic încorporat. Sînt disponibile două tipuri de circuite. Cel pentru fabricanții de sisteme se numește 486DX2, iar cel pentru îmbunătățiri instalate de către utilizatori se numește OverDrive. Cele două diferă numai prin configurația pinilor.



Timp de luni de zile Intel i-a tracasat pe utilizatori "scăpînd" cîte o informație în legătură cu circuitul de îmbunătățire a

performanțelor care se poate monta în soclu pentru coprocesor din sistemele cu 486SX. Firma se bazează pe acest produs pentru a-i lămuri pe utilizatori să renunțe la 386 și să treacă spre 486.

Soclu pentru coprocesor din sistemele cu 486SX este într-adevăr un soclu pentru înlocuirea procesorului. Dacă se instalează un circuit în acest soclu, el va prelua rolul de procesor al sistemului, dezactivînd circuitul 486SX de pe placa de bază. Chip-ul 487SX, singurul circuit care poate fi instalat în acest soclu, este identic cu 486DX, cu excepția unor mici diferențe în asignarea pinilor.

Avantajele circuitului OverDrive

Dar un OverDrive este o alternativă mult mai tentantă decît un coprocesor. În timp ce 487SX îmbunătățește viteza programelor - destul de puține - care folosesc operații aritmetice în virgulă mobilă, circuitul OverDrive va mări viteza tuturor programelor.

Intel a introdus soclul de dezvoltare alături de 486SX pentru a permite îmbunătățirea acestor sisteme. Dar și sistemele cu 486DX includ un soclu similar. Nu ar avea sens să se instaleze un 487SX, deoarece

s-ar înlocui circuitul 486DX de pe placa de bază cu un circuit identic.

Pe de altă parte, circuitul de dublare a frecvenței de ceas constituie o îmbunătățire atît pentru sistemele cu 486DX cît și pentru cele cu 486SX. Pînă în acest moment, puține sisteme cu 486DX conțin un soclu pentru OverDrive, dar majoritatea plăcilor de bază mai noi par să-l includă.

Deci ce opțiuni le sînt oferite celor care dețin sisteme cu 486DX fără socluri de dezvoltare?

Cea de-a doua versiune a circuitului 486 - turbo are aceeași configurație a pinilor ca un 486DX standard. În consecință orice sistem cu 486DX poate fi îmbunătățit dacă se scoate circuitul 486DX și se înlocuiește cu un circuit de dublare a frecvenței de ceas.

Necazurile cu garanția

Această abordare ridică o problemă: nu este chiar atît de ușor să se scoată procesorul și să se planteze un nou procesor pe placa de bază deoarece circuitul și placa de bază se pot deteriora destul de ușor. Dacă se schimbă procesorul, producătorul sistemului va considera garanția anulată. Oricum, mulți utilizatori care posedă cunoștințe tehnice își vor asuma riscul. Intel va produce "pachete" pentru îmbunătățirea sistemelor cu 486DX care nu au soclu liber, dar va recomanda ca înlocuirea să fie făcută de fabricant, de cel care a vîndut sistemul sau de inginerul de service.

Se pot cumpăra circuite OverDrive de 25MHz (699\$) și de 16MHz sau 20MHz (549\$) pentru PC-uri cu 486SX. Circuitele pentru utilizatorii de 486DX vor apare către sfîrșitul anului. Cei care cumpără sisteme cu 486DX2 dotate cu un soclu special de 238 de pini pentru OverDrive își vor putea îmbunătăți mașinile utilizînd viitoarea generație de procesoare Intel care sînt codificate P5 - cîndva în 1993.

Michael Slater

Utilizarea mouse-ului în modul grafic 320x240x256

În acest articol vom face referire la alte două articole care au apărut în revista "If": în nr. 4/1991 (despre rutine scrise în Turbo Pascal pentru utilizarea mouse-ului), precum și în nr. 4/1992 (despre programarea modului grafic 320x240x256).

În legătură cu utilizarea mouse-ului în acest mod grafic se impune o precizare: deoarece driver-ele de mouse nu recunosc acest mod (el nu este standard, singurul mod standard în 256 culori fiind 320x200), nu va fi posibilă afișarea cursorului grafic, și deci nu vom putea utiliza nici o funcție care se referă la desenarea cursorului grafic (definire cursor, afișare etc). Unit-ul *XMouse* vă pune la dispoziție o colecție de proceduri care să suplinească aceste lipsuri. Desigur, ele pot fi adaptate și completate în conformitate cu necesitățile proprii.

Să explicăm acum fiecare procedură folosită în acest program. Înainte de a scrie un program propriu care să utilizeze aceste proceduri, citiți cu atenție indicațiile de programare date în nr. 4/1991.

XResetMouse. Apelează funcția 0 a driver-ului pentru reset hardware și software. În plus se apelează funcția 12 pentru a comunica faptul că dorim să controlăm orice eveniment de mouse (deplasare mouse, apăsare buton, eliberare buton). Procedura de tip Interrupt care realizează acest lucru este *ProcMouse*. Se precizează domeniul de valori pentru coordonatele X și Y (pentru început am fixat valorile 0 - 319 pentru X, și valorile 0 - 219 pentru Y; aceasta este rezoluția pentru acest mod grafic). Cursorul de mouse este poziționat în centrul domeniului ($X = 160$ și $Y = 120$).

XModCursor. Precizează modul de desenare a cursorului. Modelul de cursor folosit în cadrul acestui program este memorat într-o matrice de 8x8 octeți, iar punctul indicat de cursor ("hot spot") în două variabile de tip întreg. Înainte de desenarea cursorului într-o anumită poziție pe ecran, porțiunea respectivă de imagine este salvată într-o zonă de manevră. Cursorul se desenează peste imaginea afișată pe ecran după următoarele reguli: dacă un octet din cadrul matricei are valoarea 0 se va păstra pixelul din imaginea veche (fundalul), altfel el va fi pus în poziția respectivă. După această regulă putem să ne construim orice model de cursor folosind culori între 1 și 255. De remarcat că această regulă de desenare a cursorului este mai simplă decât cea folosită de driver în modurile grafice standard.

ProcMouse și *ProcAMouse*. Sînt procedurile care controlează orice eveniment de mouse. Driver-ului de mouse i se comunică (prin funcția 12) faptul că procedura *ProcAMouse* asigură controlul. Această proce-

dură asigură o funcționare corectă pentru procedura *ProcMouse* (de tip Interrupt). Fără acest artificiu funcționarea programului este posibilă numai dacă în sistem este instalat un driver A4Mouse (care lucrează corect cu două tipuri de periferice: A4 și Genius). Acest driver permite să se conecteze la driver o procedură utilizator care se poate termina fie cu RETF, fie cu IRET. Alte driver-e (ex: Genius sau IBM) cer ca procedura să se termine obligatoriu cu RETF; evident, în această situație, conectarea unei proceduri de tip Interrupt conduce la blocarea sistemului în momentul în care ea este apelată. De altfel, documentația referitoare la programarea funcțiilor unui driver de mouse cere ca procedura să se termine cu RETF (observăm că driver-ul A4Mouse face în acest caz o concesie).

Acum este momentul să explicăm procedurile *XWritePixel* și *XReadPixel*. În momentul apelului se verifică dacă coordonatele x și y sînt în domeniul de valori permise (acțiune necesară deoarece este posibil ca uneori cursorul să nu fie vizibil în întregime). În plus secvențele care operează cu regiștrii controlorului VGA sînt încadrate de PUSF/CLI și POPF. Această protejare este necesară deoarece procedurile respective nu sînt reentrante (iar punctul critic îl constituie tocmai accesarea acestor regiștri). Prin această acțiune se permite ca rutina să fie întreruptă, iar procedura de tratare a întreruperii să apeleze tocmai rutina care fusese întreruptă, fără ca prin aceasta să fie afectată funcțiunea anterioară (care a fost executată doar parțial).

XShowCursor afișează cursorul în poziția indicată de coordonatele memorate în variabila *ZonSave*, iar *XHideCursor* ascunde cursorul.

XPosCursor poziționează cursorul de mouse într-un punct de coordonate precizate.

XTerMouse se apelează înainte de închiderea modului grafic, pentru ca driver-ul să deconecteze procedura utilizator *ProcMouse*.

Programul a fost testat pe mai multe calculatoare compatibile IBM-PC, cu mouse de tip Genius și A4 (3 butoane), precum și pe mai multe calculatoare IBM de tip PS/1 și PS/2, cu mouse de tip IBM (2 butoane).

Precizări suplimentare

Programul prezentat lucrează cu o singură pagină de memorie ecran. În cazul în care se lucrează cu mai multe pagini, de regulă se dorește ca afișarea cursorului de mouse să se facă în pagina de afișare (indiferent care este pagina de lucru). Pentru aceasta trebuie ca procedurile de setare a paginilor de afișare (respectiv de lucru) să fie plasate în unit-ul *XMouse* și puse în

legătură cu celelalte proceduri. Schimbarea paginii de afișare se va face în același timp cu redesenarea cursorului (dacă este cazul). Procedura *XShowCursor* va seta corespunzător pagina de lucru înaintea afișării cursorului (deci în această situație pagina de lucru trebuie să coincidă cu pagina de afișare), iar după afișarea acestuia să restaureze pagina de lucru anterioară.

mat. Ioun Cozac

```
unit xmouse;
interface uses xgrafica,dos;
const dimcursor = 7;
type patterntype = record xcursor,ycursor:integer;
  buffer:array[0..dimcursor,0..dimcursor] of byte; end;
var leftb,rightb,centerb:boolean; zonsave:patterntype;
```

```
procedure xresetmouse;
procedure xmodcursor(pattern:patterntype);
procedure xshowcursor;
procedure xhidecursor;
procedure xposcursor(x,y:integer);
procedure xtermouse;
```

```
implementation
var model:patterntype; cstare,cstat:byte;
  regs:registers; activ:boolean;
```

```
procedure procamouse; far; external; {$I xmouse}
```

```
procedure procmouse
  (rll,rcs,rip,rax,rbx,rcx,rdx,rsi,rdi,rds,res,rbp:word);
interrupt;
begin
  if not activ then
    begin
      activ = true; cstat = cstare;
      xhidecursor;
      zonsave.xcursor = rcx;
      zonsave.ycursor = rdx;
      if (rax and 2) < > 0 then leftb = true;
      if (rax and 4) < > 0 then leftb = false;
      if (rax and 8) < > 0 then rightb = true;
      if (rax and 16) < > 0 then rightb = false;
      if (rax and 32) < > 0 then centerb = true;
      if (rax and 64) < > 0 then centerb = false;
      if cstat0 then xshowcursor;
      activ = false;
    end;
end;
```

```
procedure xresetmouse;
begin
  regs.ax = 0; intr($33,regs);
  regs.ax = 12; regs.cx = 127;
```

```
regs.es = scg(procamouse);
regs.dx = ofs(procamouse);
intr($33,regs);
regs.ax = 7; regs.cx = 0; regs.dx = 319;
intr($33,regs);
regs.ax = 8; regs.cx = 0; regs.dx = 239;
intr($33,regs);
fillchar(zonsave,sizeof(zonsave),#0);
with zonsave do
  begin
    xcursor = 160; ycursor = 120;
  end;
regs.ax = 4; regs.cx = 160; regs.dx = 120;
intr($33,regs);
cstare = 0; activ = false;
leftb = false; rightb = false; centerb = false;
end;
```

```
procedure xmodcursor(pattern:patterntype);
begin
  cstat = cstare; xhidecursor; model = pattern;
  if cstat0 then xshowcursor;
end;
```

```
procedure xshowcursor;
var i,j,xo,yo:integer; p:byte;
begin
  if cstare = 0 then
    begin
      xo = zonsave.xcursor-model.xcursor;
      yo = zonsave.ycursor-model.ycursor;
      for i = 0 to dimcursor do
        for j = 0 to dimcursor do
          zonsave.buffer[i,j] = xreadpixel
            (word(xo + i),word(yo + j));
        for i = 0 to dimcursor do
          for j = 0 to dimcursor do
            begin
              p = model.buffer[i,j];
              if p0 then
                xwritepixel(word(xo + i),word(yo + j),p);
            end;
          cstare = 1;
        end;
    end;
end;
```

```
procedure xhidecursor;
var i,j,xo,yo:integer;
begin
  if cstat0 then
    begin
      xo = zonsave.xcursor-model.xcursor;
      yo = zonsave.ycursor-model.ycursor;
      for i = 0 to dimcursor do
        for j = 0 to dimcursor do
          xwritepixel(word(xo + i),word(yo + j),
```

Laborator

```
        zonsave.buffer[i,j]);
    cstare:=0;
end;
end;

procedure xposcursor(x,y:integer);
begin
    if (x=0) and (x<320) and
       (y=0) and (y<240) then
        begin
            cstat:=cstare; xhidecursor;
            regs.ax:=4; regs.cx:=x; regs.dx:=y;
            intr($33,regs);
            zonsave.xcursor:=x;
            zonsave.ycursor:=y;
            if cstat<>0 then xshowcursor;
        end;
    end;
end;

procedure xtermouse;
begin
    regs.ax:=12; regs.cx:=0; intr($33,regs);
end;
end.
```

```
.....
; XMOUSE . ASM
code segment byte public
    assume cs:code
    extrn procmouse:far
    public procamouse
procamouse proc far
    pushf
    call procmouse
    ret
procamouse endp
code ends
end
```

```
.....
program xutmouse;
uses xgrafica,xmouse,dos,rt;
const pattern:patterntype=(xcursor:4; ycursor:4;
    buffer((0,0,6,6,6,6,0,0),
           (0,0,6,5,5,6,0,0),
           (6,6,6,5,5,6,6,6),
           (6,5,5,5,5,5,6,6),
           (6,5,5,5,5,5,6,6),
           (6,6,6,5,5,6,6,6),
           (0,0,6,5,5,6,0,0),
           (0,0,6,6,6,6,0,0)));
var i,j:integer;

begin
    xsetgraph;
```

```
    xsetcolor(1,0,0,63);
    xsetcolor(2,0,63,0);
    xsetcolor(4,63,0,0);
    xsetcolor(5,63,0,63);
    xsetcolor(6,63,63,0);

    for i:=0 to 319 do for j:=0 to 239 do
        xwritepixel(i,j,1);
    for i:=0 to 319 do
        begin
            for j:=0 to 1 do xwritepixel(i,j,2);
            for j:=119 to 120 do xwritepixel(i,j,2);
            for j:=238 to 239 do xwritepixel(i,j,2);
        end;
    for j:=0 to 239 do
        begin
            for i:=0 to 1 do xwritepixel(i,j,2);
            for i:=159 to 160 do xwritepixel(i,j,2);
            for i:=318 to 319 do xwritepixel(i,j,2);
        end;
    for i:=2 to 118 do
        for j:=0 to 2 do
            begin
                xwritepixel(i+j,i,4);
                xwritepixel(319-i-j,i,5);
            end;
        for i:=237 downto 121 do
            for j:=0 to 2 do
                begin
                    xwritepixel(239-i+j,i,5);
                    xwritepixel(i+j+78,i,4);
                end;
            xresetmouse;
            xmodcursor(pattern);
            xshowcursor;
            repeat
                if leftb then
                    begin
                        xhidecursor;
                        repeat until not leftb;
                        xposcursor(160,120);
                        xshowcursor;
                    end;
                if centerb then
                    begin
                        sound(440);
                        repeat until not centerb;
                        xposcursor(160,120);
                        no sound;
                    end;
            until rightb;
            xtermouse; xterminate;
        end.
```

Copierea rapoartelor în PARADOX

O problemă în Paradox o reprezintă copierea unei specificații de raport de la un fișier la altul atunci când cele două fișiere nu au aceeași structură. Acest fapt devine foarte deranjant în momentul în care fișierele au structuri foarte asemănătoare, aproape identice, și copierea raportului nu este permisă.

Pentru înlăturarea acestui neajuns am scris următoarea procedură care are ca parametri de intrare numele fișierului sursă, numele raportului care se doarește a fi copiat, numele fișierului destinație și numele sub care va fi copiat raportul.

ing. Cristian Cosma

```
Proc Blinking(var,line)
if len(var) > 75 then
    message "Length of the string too big !!"
    sleep 2000
    quit
endif
cursor off
@ line,76-len(var) ?? " " + var
paintcanvas attribute 79 line,76-len(var),line,76
@ line,77 ?? " ..."
paintcanvas attribute 207 line,77,line,79
endproc

Proc Copy.report(source,rs,target,rt)
private frs,nfs,nft,s1,s2,t1,t2,i,mm,com1,com2,j,k,jj,ss
result clear
Blinking("Analysing source table and report
specification existance",21)
sleep 800
if substr(type(rs),1,1) <> "A" then rs = strval(rs) endif
if substr(type(rt),1,1) <> "A" then rt = strval(rt) endif
if not istable(source) then clear beep
quit "Source table does not exist !!" endif
if not istable(target) then clear beep
quit "Target table does not exist !!" endif
if rs = "r" then
    frs = source + ".r"
else
    frs = source + ".r" + rs
endif
if not isfile(frs) then
clear
quit "Source report specification doesn't happen
to be here !!"
endif
if rt = "r" then
```

```
    frs = target + ".r"
else
    frs = target + ".r" + rt
endif
if isfile(frs) then
clear style attribute 79
@ 4,12 ?? "Target report specification exists."
+ "Overwrite it ?"

showmenu
    "No" : "Leave the application",
    "Yes" : "Overwrite the target report"
+ "specification and go on"
+ " with the application"

to i
clear
if i < > "Yes"
then Blinking("Leaving",21) sleep 800 quit
endif
endif
clear Blinking("Creating dummy table",21)
nfs = nfields(source) nft = nfields(target)
array s1[nfs] array s2[nfs] array t1[nft] array t2[nft]
mm = min(nfs,nft)
array com1[nfs] array com2[nfs]
create "X__h_q" like source
copyreport source rs "X__h_q" "1"
clear Blinking("Comparing tables",21)
clearall menu {Tools} {Info} {Structure}
select "X__h_q"
i = 1
scan
    s1[i] = [Field Name] s2[i] = [Field Type]
    i = i + 1
endscan
clearall
menu {Tools} {Info} {Structure} select target
i = 1
scan
    t1[i] = [Field Name] t2[i] = [Field Type]
    i = i + 1
endscan
k = 1
for i from 1 to nfs
for j from 1 to nft
if s1[i] = t1[j] and s2[i] = t2[j] then
    com1[k] = s1[i] com2[k] = s2[i] k = k + 1
quitloop
endif
endif
endifor
clearall
if k = 1 then
quit "The source and destination tables don't have"
+ "any common fields !!"
else
    mm = k - 1
endif
```

Tips & Tricks

```
clear Blinking("Restructuring dummy table",21)
menu {Modify} {Restructure} select "X__h__q"
if istable("Problems") = True
  or istable("Keyviol") = True then {OK}
endif
ss = nrecords("Struct")
for ij from 1 to ss
  k = true
  for i from 1 to mm
    if [FieldName] = com1[i] and [FieldType] = com2[i]
      then k = false quitloop endif
  endif
  home
for ij from 1 to nft
  j = recno()
  if [FieldName] < > t1[j] or [FieldType] < > t2[j]
    then ins movcto [FieldName]
      ctrlbackspace tycin t1[j] right
      ctrlbackspace typein t1[j] right
    endif
  down
endifor
do it!
while menuchoice() = "Delete"
  {Delete}
endwhile
clearall
clear Blinking("Copying desired report",21)
copyreport "X__h__q" "1" target rt
delete "X__h__q"
reset clear beep
quit "It's been done succesfully !!"
endproc
```

Reprezentarea grafică a unei funcții

"surface.pas" realizează reprezentarea unor funcții de forma $z=f(x,y)$. Versiunea actuală reprezintă grafic funcția

$f(x,y) = \cos(x^2+y^2)$
în limitele intervalului
- $S*L \leq x, y \leq S*L$

cu un pas de $S=0,15$ și un număr de pași $L=28$.

Aici se renunță la algoritmi Hidden-Line de altfel uzuali; în locul acestora folosiți sub Turbo Pascal procedura Fillpoly. Dacă în linia

Setfillstyle (1,...)

înlocuiți 1 cu 0, atunci desenați în locul suprafeței un model în rețea.

Reprezentări "frumoase" vor da, de exemplu, funcțiile:

$f(x,y) = \exp(-x^2-y^2)$
 $f(x,y) = \sin(x) + \cos(y)$
 $f(x,y) = 1,5xy \exp(\sin(x) + \cos(y))/(1+x^2+y^2)$
 $f(x,y) = 1,5 \cos(3(\text{abs}(x)-\text{abs}(y)))$

Eventual trebuie să modificați atunci parametrii de interval S și L, ca poza să încapă complet în ecran. Plasarea punctului privitorului și pregătirea, o manipulați prin parametrii de privire R, D, T, și P. Deci nu se pun bariere fanteziei.

Programul îl puteți compila cel puțin începând cu versiunea 5.0. Aveți grijă aici, să puneți corect calea spre driver-ele BGI în procedura *Initgraph*. Programul rulează cu adaptoarele grafice uzuale Hercules, EGA și VGA. Prin opțiunea de compilare de la începutul programului, utilizați și un coprocesor.

Dacă doriți să introduceți interactiv funcția, este de folos următorul truc:

1. Împărțiți programul în 3 părți:

- partea 1: "inceput.pas" lucrează pînă la

"function f(...):real;"

- partea a 2-a: "mijloc.pas" folosește o funcție (vezi punctul 2)

- partea a 3-a: "sfîrșit.pas" prelucrează restul, începînd cu "end;" inclusiv.

2. Citiți funcția (de exemplu prin program Pascal sau prin comandă DOS "copy con mijloc.pas") ca șir de caractere $f: = \cos(\text{sqr}(x) + \text{sqr}(y))$

de exemplu într-un fișier "mijloc.pas"

3. Puneți cele trei părți din nou împrejur prin

copy inceput pas+mijloc.pas+sfirșit.pas

4. traduceți "f.pas" prin "tpc f" pentru programul executabil.

"tpc.exe" este în cele ce urmează versiunea cu linii de comandă a compilatorului Pascal. Instrucțiunile din linia 7 puteți să le preluați și în compilatorul cu linii de comandă, la care "tpc.exe" îl comandați cu următoarea sintaxă:

tpc f/\$r-/\$s-

Linia 8 vă prezintă procedeul compilării condiționate. Aici definiția "IFDEF" cu compilatorul "CPU87" se referă la computere care utilizează un coprocesor.

5. Apelați "f.exe"

Aici pașii 2 pînă la 5 trebuie să permită rezolvarea unui fișier batch, la care eventualele gerșeli sînt interceptate prin variabila "Errorlevel". Pentru a crește viteza, toate aceste operații ar trebui să lucreze pe un RAM - disc. În continuare parametri de interval și de privire/perspectivă pot fi, de asemenea, fixați interactiv; la fel puteți desena, printr-o simplă interogare în program, îl locul unei suprafețe, modelul în rețea.

(Matthias Fisches/et)

EGA cu 25/43 linii

Vă prouem în continuare două scurte programe care permit comutarea ecranului (conectat la o interfață EGA) în mod 43 linii x 80 coloane și înapoi în modul 25 linii x 80 coloane.

Programele pot fi lesne testate cu ajutorul utilitarului DEBUG. Listingurile sînt sub forma unor "fișiere de comenzi". Lansați DEBUG cu linia:

```
debug < line43.deb
```

C.N.

```
{Program: surface.pas}
{$S-,R}
{$IFDEF CPU87} {$N + } {$ELSE} {$N +,E + }
{$ENDIF}
uses Crt,Graph;
const r = 20;
      d = 900;
      t = 0.5;
      p = 1.1;
      s = 0.15;
      l = 28;
type real = single;
var g,h,i,j,k,xc,yc: Integer;
    x,y,st,sp,ct,cp: real;
    v: array[1..4] of pointtype;

function f(x,y: real): real;
begin
  f := Cos(Sqr(x) + Sqr(y))
end;
procedure q(var p: pointtype;x,y,z: real);
var a,b: real;
begin
  a := -(x*ct + y*st);
  b := (a*sp - z*cp + r)/d;
  p.x := Round(xc + (y*ct - x*st)/b);
  p.y := Round(yc - (a*cp + z*sp)/b)
end;
begin
  g := 0;
  InitGraph(g,h,"");
  if GraphResult < > 0 then Halt;
  xc := GetMaxX SHR 1;
  yc := GetMaxY SHR 1-30;
  SetFillStyle(9,1);
  Bar(0,0,GetMaxX,GetMaxY);
  SetFillStyle(1,4*Byte(g7));
  st := Sin(t); sp := Sin(p);
  ct := Cos(t); cp := Cos(p);
  for I := -l to l-1 do
  for j := -l to l-1 do
  begin
    for k := 1 to 4 do
    begin
      x := (I + k AND 2 SHR 1)*s;
      y := (j + (k-1)SHR 1)*s;
      q(v[k],x,y,f(x,y))
    end;
    FillPoly(4,v)
  end;
  Rectangle(0,0,GetMaxX,GetMaxY);
  repeat
  until KeyPressed;
  CloseGraph
end
```

```
; LINE43.COM
```

```
a
mov ax,1112
int 10
mov ax,0100
mov cx,0406
int 10
mov ax,1200
mov bl,20
int 10
mov ax,4c00
int 21
```

```
rcx
19
nline43.com
w
q
```

```
; LINE25.COM
```

```
a
mov ax,0003
int 10
mov ax,0500
int 10
mov ax,1111
mov bl,00
int 10
mov ah,01
mov cx,0809
int 10
mov ax,4c00
int 21
rcx
1d
nline43.com
w
q
```

Cod	Nume pachet	Nr. dischete	Format	Capacitate	Limba
39	BALPRO - balanță contabilă	1	5,25"	360k	R
40	UNO - completare formulare	1	5,25"	360k	R
45	Hamor Soft: hMISS - secretariat, hCONT-contabilitate, hGEN&UTIL - dezvolt., aplicații	1	5,25"	360k	R
47	Comprex S.R.L.: Spell check pentru limba română	1	5,25"	360k	R
46	AvantGarde Soft: Personal Database - secretariat & contabilitate	1	5,25"	1,2M	R
48	Spero Menus 1.0 - bibliotecă de meniuri orientată obiect	1	5,25"	1,2M	R

SHARE - if

În lista de mai sus sînt trecute programe demonstrative pentru mai multe produse software. Aceste programe "demo" pot fi obținute prin comandă. Taxa pentru copierea unei dischete este de 50 lei.

Pentru toate programele de mai sus vă putem indica numele și adresa autorilor.

PROPUNERE

Sînt convins că în România există mulți informaticieni "solitari" și multe firme cu un număr redus de angajați care, prin eforturi remarcabile, au elaborat programe ingenioase și valoroase. Unele din aceste programe, nu foarte mari ca dimensiune, pot fi extrem de interesante pentru mulți dintre noi. Și, mai ales, pot constitui soluția ideală pentru surmontarea unor dificultăți întâlnite de mulți alți colegi de breaslă. Dar cum ar putea fiecare să afle de existența celui alt?

Pentru a facilita aceste contacte și pentru a contribui la activarea pieței de soft de pe la noi (totuși, cred că "piață de soft" nu înseamnă numai AutoCAD sau FoxPro sau Corel Draw sau OS/2) ne-am gîndit să înființăm un fel de serviciu de distribuire a programelor.

Avînd în vedere costurile unei campanii publicitare, este imposibil pentru un student sau pentru un grup de 2-3 "softiști" să-și promoveze programul de protecție prin parolă a discului hard sau pe cel de balanță contabilă sau pe cel de editare simultană a două fișiere text.

De aceea vă facem următoarea propunere: folosiți-vă de revista "if" pentru a promova și a vinde programele pe care dumneavoastră le considerați interesante. Revista ar putea asigura un spațiu de prezentare a programelor (putem să-i spunem și reclamă) și expedierea dischetelor cu programe și documentații către solicitanți. Pentru aceste servicii, revista și-ar rezerva un comision de 20-25% din prețul de vânzare propus de autor.

Evident, fiind vorba de o propunere, cifrele și metodele de lucru sînt "negociabile".

Dacă ideea vă stîrnește entuziasmul (sau măcar interesul) nu ezitați să ne contactați !

Redacția



O REVOLUȚIE

în lumea rețelelor de calculatoare

LANtastic, produs ARTISOFT - SUA

La un preț de câteva ori mai mic
decît al rețelelelor clasice, obțineți
aceleași funcții esențiale și o

CALITATE GARANTATĂ 5 ANI

Pentru a primi relații și
materiale documentare
căutați-ne chiar acum la

TEL / FAX 99 / 436-433

ART COMM GROUP

Distribuitori / intermediari

zonali sînt bineveniți

Rețelele LANtastic vă oferă ocazia realizării visului oricărui posesor de calculatoare, cu menținerea tuturor facilităților oferite de alte firme, cu mai multă flexibilitate și la un preț incomparabil mai mic.

**NU EZITAȚI. ACESTEA SÎNT REȚELELE PE CARE LE-AȚI
AȘTEPTAT PÎNĂ ACUM!**



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: 0 04-954-31660

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. !**