

if

revistă de Informatică
editată de firma Micro ATCI

Director: ing. Dumitru Dunca

La realizarea acestui număr
au colaborat:

ing. Attila Darvas,
ing. Iosif Fettich,
ing. Tibor Kallo
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:

mat. Eugen Rotaru
ing. Sava Stan

Tiparul: tipografia Tîrgu Mureş
Tipografi: Halaşiu Mihai,
Cormoş Mircea

Revista apare lunar

Preţ. 57 lei

Adresa şi telefonul redacţiei:
Micro ATCI,
RO-4300 Tîrgu Mureş,
C.P. 64, O.P. 1,
tel. 954/17024, fax 954/35208.

Manuscrise originale sau listing-uri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu directorul. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate la următoarele tarife:

1 cm ² (alb-negru)	30 lei,
1 cm ² (alb-negru şi o culoare suplimentară)	40 lei,
1 cm ² color	60 lei.

Doritorii sînt rugaţi să ia legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei.

Sys"if"

"Orice început este greu", dar așa de greu ca la acest număr, ca la numărul trecut și ca la celelalte numere care le-au precedat nu ne-a fost la nici un număr. Atunci cînd în urma luptei cu forțele naturii (umane) dezlănțuite mai reușim să "scoatem" un număr, ne crește inima de bucurie; am atins culmea. Bucuria este însă de scurtă durată, aidoma lui Sys"if", nici n-apucăm să răsuflesc ușurați, că ne și trezim din nou la poalele muntelui. Speranța că după cîteva urcușuri vom cunoaște măcar "traseul" și că tocindu-ne tălpile vom tocii cît de puțin și din munte, s-a dovedit a fi zadarnică. Dar nu disperăm (probabil că vreun șurubel joacă), și în loc să ne perfecționăm cunoștințele în tehnica vidului (de putere, din stomac, ș.a.m.d.), încercăm să implementăm o revistă de informatică, în spațiul carpato-danubiano-pontic, pornind din Transilvania și nu din Transvaal. N-am vrea să fim greșit înțeleși, noi nu înjurăm pe nimeni, ne-ajunge ce-i în jur.

În acest număr vom descoperi împreună alte tărîmuri ale mirobolantei lumi a calculatoarelor personale. Sperăm că acest prim contact (în if) cu informatica de proces, grafica pe calculator și compilatoarele, să vă ofere răspunsuri la multe întrebări, chiar dacă aceste răspunsuri vor genera și mai multe întrebări.

Aceste teme, ca și multe din temele anterior abordate, sînt foarte complexe și nu pot fi epuizate la o simplă trecere în revistă. Din acest motiv noi vom reveni periodic asupra unor subiecte pentru a le nuanța și a observa tendințele lor de evoluție.

Începînd din acest număr venim, în sfîrșit, în întîmpinarea dorințelor formulate de unii cititori cu două cursuri, unul de rețele neuronale și unul de programare orientată obiect. Acesta este doar un prim pas. Vom încerca să inițiem, în numerele următoare, și alte cursuri solicitate. Avînd în vedere intervalul de timp destul de mare dintre două apariții succesive ale revistei vom încerca să fragmentăm subiectele în așa fel încît fiecare episod să aibă, pe cît posibil, o existență de sine stătătoare.

Criticile, aprecierile și relațiile d-voastră ne vor ajuta să înțelegem unde am greșit (sigur am greșit !), ce am făcut bine (dacă am făcut ceva bine !) și ne vor permite să ne apropiem mai mult de "ce-i mai mîndru pe la noi". Vă rugăm să nu vă supărați pe noi dacă ecoul spuselor d-voastră nu se va face simțit chiar din primul număr care va urma scrisorii. Datorită ritmurilor levantine în care se antamează mai toate pe acest picior de plai, iubit de zei și bătut de soartă, și "interstațiilor amăgirii", de regulă, lucrăm cu două, trei numere în avans. Așteptăm scrisorile d-voastră !

ing. Romulus Maier



Cuprins

Știri

- Cu RISC și multitasking pag.4
- hobBIT pag.5

Magazin

- Înapoi în viitor pag.6

Fabrica de vise are un star nou: calculatorul. Cele mai bune "efecte speciale" ale ultimelor succese cinematografice de la Hollywood i se datorează lui.

Grafică

- Programe de grafică pag.8
- Artă în 256 de culori pag.11
- Formate grafice pag.13

Software

- Babilonia limbajelor de programare . . . pag.16
- Quick-C cu Quick-Assembler 2.5 . . . pag.18
- Quick Basic 4.5 pag.18
- Quick-Pascal 1.0 pag.18
- Turbo-Pascal 6.0 pag.19
- MS-Basic 7.1 - PDS pag.19
- Assembler MASM 5.1 pag.19
- Turbo-Assembler 2.0 pag.20
- Microsoft-C 6.0 - PDS pag.20
- Turbo C++, Prof. Version 1.0 pag.20
- Zortech C++, Dev. Edition 2.1 pag.21

Cursuri

- Pe drumul spre o tehnică "inteligentă ?" pag.22

Sistemul nervos uman prelucrează mai multe miliarde de informații în modul său propriu. Acest mod de lucru al biologicului și transpunerea sa în sisteme tehnice este astăzi parte integrantă a domeniului Inteligenței Artificiale. Stadiul și posibilele influențe ale acestor

cercetări vrem să vi le prezentăm în acest curs structurat în patru părți.

- Obiectele se încapsulează pag.24

Dacă doriți să programați mai comod și să lucrați mai ușor cu proiecte mari, atunci nu puteți face abstracție de OOP (programarea orientată obiect).

Hardware

- Sisteme de achiziție de date și control cu calculatoare personale compatibile IBM PC/XT/AT pag.29
- Ce este IEEE 488? pag.39
- Digitizoarele -
interfețe spre grafică și artă pag.42

Reușita muncii în studiourile CAD, sau în studiourile de grafică, depinde în mod hotărâtor de caracteristicile ergonomice și de modul de funcționare al digitizoarelor. Ele sînt interfața între mină și calculator.

Practică

- Să nu uităm CP/M-ul... pag.44
- Sfaturi practice pentru
întreținerea harddisk-urilor pag.48

Rețele

- Administratorul unei rețele locale . . . pag.51

Virusologie

- Institutul European contra Virușilor pag.53

Rubrici

- Editorial pag.1
- Poșta redacției pag.54
- Mica publicitate pag.54

Cu RISC și multitasking

Cambridge - Active Book Company a construit un calculator Notepad cu procesor RISC, care umbrește tot ceea ce s-a realizat pînă acum în acest domeniu. Calculatorul de dimensiunea unui format DIN A4 va fi deservit doar cu un creion electronic.

În laboratoarele de cercetare ale firmei Active Book Company din Cambridge se naște viitorul. Vă prezentăm un interviu luat de revista PC Magazin directorului acestei firme, Hermann Hauser, în care vor fi prezentate cele mai noi realizări din domeniul Notepads: este vorba despre "Active Book" cu suprafața utilizator "Open Book".

PC Magazin: Încă de la mijlocul lui 1990 vorbești despre un prim prototip Active Book. Cît au avansat cercetările ?

Hauser: Dispunem acum de un prim beta-model avînd toate facilitățile funcționale. Între timp am mai adus cîteva îmbunătățiri versiunii planificate. Active Book va intra în producție de serie în al doilea trimestru al anului 1991.

PC Magazin: Pentru ce piață ați conceput calculatorul Active Book?

Hauser: Piața spre care țintim este pe de o parte piața verticală a mobililor oameni de afaceri, dar vizăm și alte piețe, ca de ex.: sănătate, poliție, asigurări, personalul vînzător din așa numitele "Fast moving consumer goods" sau domeniul cercetărilor din domeniul pieții.

PC Magazin: Vedeți calculatorul Active Book doar ca pe un Notepad, chiar dacă mai rapid, între celelalte ?

Hauser: În nici un caz. El este mai mult o unealtă productivă pentru manageri din toate domeniile.

Lucrări cum ar fi planificarea costurilor, sau asemănătoare, care necesitau adesea mult timp pe un calculator de birou, vor putea fi rezolvate în doar cîteva minute.

PC Magazin: Unitatea centrală a calculatorului Active Book este realizată cu un procesor RISC. Este același procesor cu care sînt echipate și calculatoarele Archimedes ?

Hauser: În Notepad-ul nostru am implementat un procesor ARM-2as, numit și Hercules, un așa-numit "Full Static Device". Acesta este încă și mai rapid decît ARM-3, deoarece accesează o memorie RAM statică. Procesorul poate atinge 12.5 MIPS în modul de lucru Burst și 5 MIPS în medie la o încărcare de durată. Acestea nu sînt însă limitele pe care le poate atinge. ARM-2as poate atinge o viteză de 40 MIPS, dar din motive economice am subdimensionat tactul la care lucrează cipul. Pentru viitor aceasta înseamnă că vom putea, în cazul unor acumulatoare avantajoase, să lăsăm procesorul să lucreze mai rapid, viteza și puterea fiind în corespondență directă.

PC Magazin: De ce nu v-ați bazat pe tehnicile procesoarelor existente, de ex. Intel ?

Hauser: Decizia de a folosi un procesor RISC ARM n-a fost pusă nici un moment sub semnul întrebării. Ca director și coparticipant la elaborarea acestui CPU la Acorn sînt convins că doar această tehnică ne va permite să aducem pe piață cel mai rapid Notepad. În plus cipul RISC ARM reunește pe aceeași plăcuță de silicu mai multe funcțiuni: unitatea centrală (CPU) propriu-zisă, RAM-ul static, unitatea de management a memoriei, controller-ele de grafică și DMA ca și controller-ele pentru magistralele de 8, 16 și 32 de biți. Toate ace-

te facilități sînt realizabile în tehnicile obișnuite de construcție doar prin utilizarea mai multor module. Avantajul utilizării cipului Hercules este evident: el este singurul procesor din lume care reunește într-un singur cip un întreg calculator. Ne-am planificat realizarea unui cip ARM-4 care va fi încă și mai rapid decît cele mai rapide procesoare ARM de pînă acum.

PC Magazin: Datele culese pe sisteme DOS vor putea fi preluate, prelucrate pe Active Book și transferate înapoi pe un sistem PC-DOS ?

Hauser: Datele DOS vor putea fi importate deoarece Active Book este compatibil DDE și DOS. În plus este posibilă utilizarea fie a unui lector de dischete de 3,5" fie a unui harddisk de același format. Normal că datele vor putea fi preluate în Active Book și prin intermediul unei interfețe seriale. Sîntem compatibili cu formatele de date Word, Wordperfect și Lotus 1-2-3. Cu ajutorul creionului electronic se vor putea, de ex., tăia interactiv celulele dintr-o tabelă electronică, introduce valori noi și altele asemănătoare. Prin utilizarea conceptelor de carte (Book) și creion (Stift), prin digitalizarea caracterelor se va obține același efect ca și în cazul utilizării unei tastaturi. Aceasta este doar o opțiune, dar noi facem încă un pas mai departe. Olivetti deține 8 procente în Active Book Company. Dar ce-are a face acest fapt cu compatibilitatea noastră ? Emulatorul nostru de DOS, o soluție pur soft, este chiar mai compatibil decît, de ex., hardware-ul Olivetti. Acest lucru l-am stabilit cu ajutorul testelor "Validation Suites", pe care Olivetti ni le-a pus la dispoziție avînd în vedere participarea la afaceri. Aceste sînt teste care verifică compatibilitatea. Pe Active Book poate fi rulat chiar și simulatorul de zbor al lui Microsoft.

PC Magazin: Ce aplicații există pentru Active Book ?

Hauser: Împreună cu acest Notepad livrăm o paletă completă de aplicații uzuale, de ex., un program de calcul tabelar, un pachet de grafică, un program de planificare de termene. Aceste soft-uri nu oferă tot atâtea facilități ca și produsele Microsoft, dar prezintă avantajul integrării, deci schimbul dinamic de date între programe atinge o rată mai ridicată. Tocmai a fost abordată funcția DDE, înafară de aceasta sistemul de operare permite multitasking-ul. El este un sistem Unix, are mărimea de 2 MBytes, se numește Helios și este încărcat în memoria ROM; eu cred

că înaintea noastră n-a mai încărcat nimeni atîta software în ROM.

PC Magazin: Ce medii de memorare pot fi utilizate pentru Active Book ?

Hauser: Pe de o parte este opțională utilizarea unui harddisk sau utilizarea unui lector de dischete, pe de altă parte pot fi utilizate diferite memorii de masă ca și cartele ROM sau PC Memory Cards după standardul PCM-CIA. Cu un harddisk și cartele de memorie se poate ajunge ușor la mai mult de 100 MBytes memorie de masă.

PC Magazin: Cîte ore poate lucra independent Active Book ?

Hauser: La o utilizare de durată cca. 8 pînă la 10 ore.

PC Magazin: Avînd în vedere utilizarea unei unități centrale RISC și multitudinea facilităților oferite, Active Book va fi un Notepad scump ?

Hauser: După calculele noastre prețul se va ridica la cca. 4000 DM. Avînd în vedere caracteristicile de putere putem spune că este avantajos.

(PC Magazin 3/91)

Cu o acțiune de urmărire specială poliția din München încercă să rezolve cel mai mare furt de pînă acum din domeniul calculatoarelor. Faptele și împrejurările în care a fost comis furtul duc la concluzia că faptașii sînt hoji profesioniști cu cunoștințe în domeniu. De la întreprinderea de comerț CTT din München au fost furate 1110 discuri Winchester și 25 de adaptoare "Host" în valoare totală de 750.000 DM. Au fost sustrate doar aparate Seagate noi. Spărgătorii nu s-au atins de produsele altor producători și nici de modelele Seagate reparate a căror ambalaj diferea doar cu puțin de cel al modelelor noi. CTT speră să dea de urma făptașilor după numerele de înregistrare de pe discuri și după certificatele de garanție, de îndată ce aceștia vor ieși cu ele pe piață. Spărgătorii, care au folosit cel puțin o camionetă, au încărcat bunurile în lăzi fără ambalaj ceea ce ar putea duce la deteriorări în timpul transportului. CTT subliniază că nu le-a fost diminuată capacitatea de livrare datorită acestui furt.

În sfîrșit: o revistă pentru posesorii de HC-uri!

hobBIT

se numește noua apariție; revista este editată de Clubul Român de Calculatoare (P.O.Box 37-131 București) și conține, pe parcursul a 32 de pagini, informații despre CRC, Beta Basic, Simon's Basic, o pagină pentru începători, jocuri comentate, tips & tricks, o pagină pentru constructorii amatori și altele. Revista costă 35 lei plus 4,80 lei taxă de expediție.

Cu versiunea 11 a programului de proiectare asistată de calculator Auto-CAD, Autodesk a adus nu numai o mulțime de facilități noi ci s-a hotărît și asupra unei direcții strategice: noua versiune va putea fi rulată doar pe PC-uri DOS 386 sau 486 sub SunOS. Alte sisteme de operare, suprafețe grafice sau platforme (Macintosh, Apollo, VAX) nu vor mai fi luate în considerare. Una din cele mai importante noutăți ale versiunii 11 este facilitatea restrînsă de lucru în rețea, ce va fi atinsă prin tehnicile Filelocking (blocajul fișierelor) și Plotspooling (desenare prin spooling). Acum pot fi afișate simultan mai multe vederi în ferestre variabile și acestea pot fi și tipărite împreună. Textele de cotare pot fi afișate în culoarea și la înclinarea dorită în cazul unui desen.

Seminar

În perioada 6-8 februarie și respectiv 12-14 februarie, la București și Sibiu a avut loc seminarul "Micro-Informatica, Sisteme și Management" organizat de firma LOGIC sub egida CNI.

Spicuum din tematica seminarului:

- Micro-Informatica - prezent și viitor
- Sisteme informatice în contextul economiei de piață
- Management și Informatica
- Sisteme de operare pentru microcalculatoare
- Baze de date și limbaje de programare
- BORLAND: prezentarea produselor program
- Rețele locale de calculatoare (LAN)
- Arhitectura și caracteristicile rețelelor de tip NOVELL
- Configurații pentru rețelele de tip NOVELL
- Administrarea rețelelor de tip NOVELL

Înapoi în viitor

Fabrica de vise are un star nou: calculatorul. Cele mai bune "efecte speciale" ale ultimelor succese cinematografice de la Hollywood i se datorează lui.

Apa trăiește: oamenii care își așteaptă ultima salvare într-o stație subacvatică nu-și pot crede ochilor. Din apa liniștită a ecluzei se ridică un sorb care se îndreaptă în mod evident spre atractiva comandantă a submarinului. În momentul în care creatura marină a ajuns la înălțimea chipului femeii, îi reproduce fizionomia - 100 de litri de apă de Pacific îi surd. Cel mai târziu în acest moment, cinefilul se simte cuprins de afecțiune pentru o masă de apă. Aceasta este cea mai bună dovadă pentru funcționarea trucului.

Întîlnirea cu creatura acvatică face parte din filmul pentru copii "Abyss", care în 1989 a produs mare vîlvă. Motivul: acțiunea foarte palpitantă, înscenată de regizorul James Cameron (Aliens), a fost condimentată cu efecte inexistente pînă acum. Acestea provin în bună parte din calculator, căci cu ajutorul efectelor speciale uzuale asemenea trucuri, cum ar fi coloana de apă inteligentă, nu ar fi posibile. Calculatorul își face intrarea în fabrica de vise.



Monstrul marin a fost creat pe calculator

"Grafica pe calculator potențează creativitatea. În momentul în care un regizor trebuie să se întrebe mereu dacă ideile sale vor putea fi traduse în viață, ajunge să se cenzureze singur. Dar dacă orice există în fantezia sa poate fi realizat pe ecran, atunci el se poate concentra asupra acțiunii filmului și poate crea o lume de vis." ne spune Doug Kay directorul departamentului de grafică pe calculator al firmei ILM ("Industrial Light and Magic"), firma cu cei mai buni specialiști în materie de trucuri din lume.

În spatele firmei ILM stă o legendă vie a Hollywood-ului: producătorul George Lucas, care a realizat, împreună cu regizorul Steven Spielberg, filme de mare succes, cum ar fi: "Războiul stelelor", "ET" și "Indiana Jones". Secția de trucuri, înființată în 1981, trebuie să răspundă unor mari pretenții: ILM trebuia să producă, pentru Lucas și Spielberg, iluzii perfecte. Tot ceea ce este prea costisitor de realizat, sau prea periculos pentru cascadori, este făcut de ILM. "Specialitatea noastră este imposibilul", afirmă Doug Kay.

Chiar și monstrul marin din Abyss părea de nefilmat. Problema: apa este transparentă, deformează imaginea din fundal și aruncă pete de lumină pe pereți. Deoarece se dorea ca scena să pară reală, nu se putea filma pe baza trucurilor uzuale, cu modele sau cu păpuși. "Nici specialiștii noștri în butaforie n-au descoperit încă apa care să fie în același timp și solidă și fluidă", ne spune Doug Kay. Din acest motiv specialiștii în trucuri au optat pentru grafica pe

calculator, deoarece calculatorul este în stare să calculeze la fel de bine imaginea unei coloane de apă ca și pe cea a unei bile de crom sau a unei mese de lemn.

Problema era cum să se introducă monstrul, în mod perfect, în mediul real al filmului. "Monstrul trebuia să arate natural, deci pereții submarinului trebuiau să se reflecte în apă." descrie Doug Kay una din problemele principale. Dar cum se pot introduce culisele deja filmate în calculator ?

Soluția este dată de utilizarea unui scanner cu laser, care înregistrează cadrele filmului unul după altul, și le înmagazinează în memoria calculatorului. Dacă imaginile de bază se află în calculator, specialiștii pot modifica imaginile după dorință.

În Abyss, calculatorul a calculat, pe baza formulelor din fizică, în mod exact și corect, reflexiile luminii, așa cum ar fi fost văzute dacă monstrul ar fi existat în realitate. Imaginea completată, în calculator, cu coloana de apă va fi transferată cu ajutorul unei imprimante laser color pe o nouă peliculă de film, și astfel efectul special este gata.

Colectivul a lucrat intens patru luni, pînă cînd detaliile au cores-puns în totalitate și pînă cînd regizorul James Cameron a fost convins. "N-a crezut că așa ceva s-ar putea realiza cu calculatorul. A trebuit să-l convingem că monstrul va părea real și nu va arăta ca o grafică ieftină pe calculator, și am reușit." spune Doug Kay.

Cea mai uluitoare aplicație a manipulării imaginii: pe calculator reușesc metamorfoze la care nici pictorul M. C. Escher n-ar fi putut visa - dintr-o clipă într-alta un om se poate transforma într-un leu, sau o capră se poate transforma într-o

pasăre. Doug Kay declară: "Cu calculatorul putem transforma orice figură în oricare altă figură. Pentru aceasta avem nevoie doar de imaginea inițială și de cea finală - restul îl rezolvă programele noastre."

Cu acest truc a fost realizat și teribilul final din filmul "Indiana Jones și ultima cruciadă". Ticălosul Donovan vrea să bea din izvorul sfânt, cu un pahar fermecat, descoperit de Indiana Jones, pentru a deveni nemuritor, dar alege greșit paharul și se transformă în praf în câteva secunde. Doug Kay ne povestește: "În mod normal pentru această scenă actorul și trei machiori ar fi fost ocupați mai multe zile. Pentru trei secunde de film ar fi trebuit făcute mai mult de 150 de modificări de machiaj. Într-o astfel de situație calculatorul este mai rentabil și menajează și nervii colectivului."

Tehnicienii ILM modifică, cu ajutorul calculatorului, nu numai imagini, ei îl folosesc și pentru mixajul digital de imagini. În filmul "Înapoi în viitor", actorul principal Michael J. Fox stă de patru ori cu el însuși la masă, de fiecare dată în alt rol. Pentru aceasta fiecare poziție a fost filmată separat, iar calculatorul a copiat într-o scenă imagine cu imagine, astfel încât diferitele variante Fox să se suprapună parțial. Calculatorul este în mod automat atent ca perspectivele să corespundă, astfel ca o persoană din prim plan să pară mai mare decât una din fundal.

Pentru mixarea unor imagini truate și a unor imagini reale, se folosește de secole tehnica "blue box". Toate scenele se înregistrează pe fundalul unui cer albastru. Prin metoda "blue box" fundalul poate fi mascat. Tot ceea ce este albastru va dispărea. Prin suprapunere se pot mixa două imagini diferite.

În studiourile ILM, izolate ermetic, se află cel mai mare ecran albastru din lume. El este mai înalt de



Nu machiorii ci calculatorul este cel care-l transformă pe Donovan în praf în filmul Indiana Jones și ultima cruciadă

15 metri și lat de 20 de metri, și este suficient pentru a face o mașină să zboare, în filmul "Înapoi în viitor", sau pentru a face să zboare trei copii pe biciclete, în "ET".

Indiferent de cât de perfecte apar trucurile specialiștilor, o greșeală, cât de mică, tot se strecoară în fiecare film. Foarte rar însă ea este atît de crasă ca în partea a doua a

filmului "Războiul stelelor". Chiar la început apare pe ecran, cu litere imense, titlul filmului: "Imperiul contraatacă". Este o greșeală clasică de scriere de care calculatorul nu este vinovat. Nobody is perfect, nici chiar George Lucas.

(Computer Live 9/90,
Gregor Neumann)

Programe de grafică

Denumirea de programe de grafică ascunde o întreagă lume. Și pentru că această lume este atât de diversă, alegerea unui program anume este dificilă. Cine spune doar că dorește să cumpere un program cu care să deseneze grafice, poate călca în străchini. Să dăm deci străchinile la o parte și să ne lămurim asupra problemei.

Programele de grafică sînt atît de bogate în fațete, încît unele programe nici nu pot fi încadrate corect într-o categorie anume. Dar care sînt categoriile ? Programele de grafică s-ar putea clasifica în: programe de pictură, programe de desenare, programe pentru ilustrare, grafică de afaceri, pachete de prezentare evolute, animație în două dimensiuni (2D), programare asistată de calculator (CAD), în diferitele sale variante (de la programe 2D orientate pe muchii pînă la modelarea solidelor), programe de prelucrare a imaginii, programe de "umbrire" (Rendering Software) și de "iluminare" (Raytracing Software), programe de trasare automată și tot ceea ce poate constitui o combinație a celor de mai sus, sau care se află la limita a două sau mai multe categorii. Cel care a reușit să depeze acest ghem și a înțeles fiecare termen va găsi programul de grafică dorit mult mai ușor.

Marea diferență

Formatele grafice pot fi clasificate în două grupe principale:

- prima cuprinde formatele grafice vectoriale, respectiv pe cele orientate pe obiecte.
- a doua cuprinde formatele grafice orientate pe pixeli, sau bitmap.

În cazul programelor grafice vectoriale, (numite și "programe de desen") liniile, pătratele sau cercu-

rite sînt tratate ca și obiecte, iar toate informațiile despre imagine sînt administrate de programe sub forma unor descrieri matematice ale obiectelor, în acest mod fiind și salvate pe dischete sau harddisk-uri.

Programele grafice bitmap (hartă de biți), denumite deseori și "programe de pictură", înmagazinează și administrează imaginile descriind amănunțit fiecare punct al imaginii. În cazul lor, un grafic este înțeles ca o imagine construită pe rînduri, în mod asemănător cu imaginea de pe ecranul TV, care și ea este construită rînd cu rînd, iar în cadrul rîndului punct cu punct. În acest caz se memorează culoarea fiecărui punct, într-o anumită succesiune, iar în cazul imaginilor alb-negru, se vor memora pentru fiecare punct informațiile "punct aprins" (alb) sau "punct stins" (negru).

Ambele tipuri de programe au atuuri puternice și deci și domenii speciale de utilizare. Programele de grafică vectorială au fost dezvoltate inițial pentru domeniul construcțiilor (proiectare asistată de calculator PAC - CAD în original: Computer Aided Design), dar datorită creșterii continue a performanțelor lor au ajuns să aibă înțietate în tot mai multe domenii. În domeniului design-ului profesional aceste programe sînt utilizate aproape în exclusivitate.

Utilizarea lor este avantajoasă deoarece operînd cu obiecte, desenele pot fi foarte ușor modificate și corectate (mult mai ușor decît la programele orientate pe pixeli). Spre exemplu, orice obiect de pe suprafața de desen este liber, poate fi mișcat liber, este scalabil (poate fi mărit sau micșorat), poate fi rotit, i se poate defini în mod liber culoarea (totul depinde doar de posibilitățile programului lansat). În plus, oricare ar fi imprimanta folosită pentru tipărire, desenul va fi

tipărit de fiecare dată cu rezoluția și finețea maximă oferită de imprimantă.

Punct cu punct: cui folosește ?

La programele grafice bitmap, situația se prezintă altfel. Ele nu se pretează la construcția de planuri sau imagini, deoarece fiecare linie sau cerc care sînt desenate rămîn (aproape) definitive. Majoritatea programelor posedă, de cîtva timp, și funcția UNDO, cu ajutorul căreia desenatorul poate anula ultima operație efectuată. Dar dacă, din greșeală, ați desenat un pătrat negru peste un desen fin și complicat, atunci nu se mai poate repara nimic.

La programele orientate pe obiecte desenul din linii se mai poate vedea și după ce pătratul negru a fost îndepărtat. Nici la tipărirea pe imprimantă imaginile bitmap nu pot fi tipărite atît de flexibil ca și cele vectorizate. Dacă, de ex. desenăm cu un program bitmap un cerc cu un diametru de 100 de puncte, atunci acest cerc va fi tipărit ca atare. Dacă mărim graficul, ceea ce cu un program DTP se poate face foarte ușor, atunci spațiul ocupat de grafic va crește, bineînțeles, dar rezoluția va rămîne aceeași și calitatea imaginii va scădea, iar prin mărire succesivă treptele graficului (care apar datorită punctelor dreptunghiulare, singulare ale imaginii) vor fi din ce în ce mai vizibile.

Dar unde sînt atunci avantajele programelor bitmap ? În primul rînd în domeniul artei pe calculator. Programele bitmap pot reprezenta și prelucra un număr tot mai mare de culori. Standardul actual, în domeniul profesional, atinge cifra maximă de 16.7 milioane de culori. Aceasta este suficient chiar și pentru profesioniștii cei mai pretențioși.

Un alt domeniu, foarte important, de utilizare a programelor grafice orientate pe pixeli este prelucrarea și retușarea imaginilor, care sînt preluate pe calculator prin intermediul unui scanner sau camere video. Acest domeniu este foarte important pentru utilizatorii care doresc să preia imagini, deja existente, în documentele DTP.

Categoriile programelor de grafică

Programe de pictură

Programele de pictură sînt programe cu ajutorul cărora pot fi desenate imagini orientate pe puncte. Dintre acestea amintim: Deluxe Paint II PC, PC Paintbrush, GEM Paint, PC Atelier, Colorix VGA Paint. Tot din această categorie fac parte și programele de retușare a imaginilor scanate. Programele de pictură servesc în primul rînd la pictarea de imagini și în al doilea rînd la prelucrarea lor.

Programe de prelucrare a imaginilor

Aceste programe sînt o formă mai dezvoltată a programelor de pictură. În cazul lor, nu funcțiile de desenare sînt importante, ci funcțiile de prelucrare a imaginilor deja existente. Se pune un accent deosebit pe prelucrarea fotografiilor de rezoluție înaltă scanate. Cele mai importante funcții sînt cele de reglare a contrastului, de montaj a imaginilor și cele care permit obținerea de efecte fotografice. Programele de prelucrare a imaginilor au nevoie, de obicei, de multă memorie și în plus necesită plăci grafice cu trepte de gri de rezoluție maximă. Cel mai cunoscut produs, din această categorie, sub MS-DOS, în momentul de față este "Imagestudio Windows". Deoarece aceste programe necesită anumite condiții speciale, ele se vînd în majoritatea cazurilor împreună cu cartelele grafice corespunzătoare, ca soluții complete.

Grafică de afaceri și de prezentare

Aceste programe produc în primul rînd reprezentări grafice ale unor mulțimi de cifre, pe scurt grafice (charts). Aceste grafice pot fi fie diagrame tort, fie histogramme, grafice plasă, etc. Pachetele soft Microsoft Chart, Harvard Graphics, Mirage sau Perspective, sînt în primul rînd programe de desenare de grafice. Mai nou toate programele de calcul tabelar au integrate și funcții de grafică de afaceri. Unele programe s-au specializat pe grafice tridimensionale, altele s-au orientat spre grafica de prezentare. În unele programe de prezentare au fost preluate funcții de aranjare în pagină, ca și cele de la programele DTP, și au fost preluate chiar și funcții specifice programelor de pictură și de desen. Unele conțin rutine de prezentare (show) și rutine de animație. Diferențele dintre caracteristicile de performanță în cadrul acestei categorii sînt considerabile.

Programe de animație

Cu ajutorul acestor programe se poate produce grafică animată. Cele mai multe dintre aceste programe sînt comandate de limbaje Script, de exemplu Grasp și VCN Concorde. Grasp atinge puțin și categoria grafică de prezentare. Cel mai nou și mai eficient program din acest domeniu este Autodesk Animator, care conține diferite tehnici de animație, cum ar fi: "page-flipping" - alternarea de imagini întregi, ca la desene animate, "cel animation" - mișcarea doar a unor părți ale imaginii, în timp ce partea fixă a imaginii rămîne neschimbată, "tweening" - fazele intermediare între două stări ale unui obiect sînt calculate în mod automat de către program și cuplarea cu imagini umbrite.

Programe de desenare

Programele de desenare sînt programe simple orientate pe obiecte, fără funcții speciale supli-

mentare. Reprezentanții standard ai acestei categorii sînt: GEM Draw Plus, sau Micrograf Draw Plus. Ele nu oferă nimic extraordinar, cum fac programele CAD, sau cele de ilustrare, dar tocmai din acest motiv sînt și mult mai ieftine. De regulă funcțiile acestor programe sînt suficiente pentru a produce ilustrații simple, diagrame de flux, grafice tehnice puțin complicate, și altele asemenea.

Programe de ilustrare

De regulă programele de ilustrare sînt versiuni dezvoltate ale programelor de desen, unele sînt însă concepute în mod special pentru prelucrarea unor grafice obținute cu programe de desen. Programele de ilustrare sînt utilizate mai ales pe piața DTP (Desktop Publishing) și ele pot fi folosite, printre altele, și pentru producerea graficelor pentru cataloage. Programele de ilustrare stăpînesc separarea culorilor în cele patru culori de bază ale imprimantelor color, funcțiile lor de desenare sînt specializate pe ilustrarea cu contururi rotunjite (spre deosebire de programele de desenare care de regulă tratează doar obiecte colțuroase). Concurența între produsele acestei categorii este foarte mare, astfel încît la fiecare trei luni apare pe piață cite o nouă versiune, mai performantă, a unuia sau a altuia dintre produse. Cele mai noi în momentul de față sînt Designer 3.01 și Corel Draw 2.0, dar tot din această categorie fac parte și Diagraph Windows și Artline.

Programe CAD

CAD a fost la început prescurtarea pentru "Computer Aided Drafting" (desenare asistată de calculator) și a devenit mai tîrziu prescurtarea pentru "Computer Aided Design" (proiectare asistată de calculator). Aceste programe au fost gîndite în primul rînd pentru scopuri de construcții. Deoarece totul trebuie reprezentat cu o precizie matematică, funcțiile programelor CAD sînt specializate pe

desene tehnice. Deci ele dețin funcții de măsurare, permit introducerea coordonatelor exacte, tipărirea la scară, etc. De obicei ieșirea se face pe plotter, obținându-se în final desenele tehnice de execuție. Majoritatea programelor CAD dispun și de drivere pentru imprimante laser sau matriciale.

Există programe CAD specializate pentru proiectarea cablajelor imprimate, pentru arhitectură, arhitectură internă, construcții de mașini, design auto, și chiar și unele specializate pe decorațiuni interioare. Cel mai cunoscut program CAD este Autocad care a acaparat mai mult de 40% din piață.

Sistemele CAD se mai pot clasifica în sisteme 2D și în sisteme 3D. Sistemele CAD 2D pot fi la rândul lor orientate pe muchii sau pe elemente. Există diferite realizări de sisteme 3D, de la cele simple, tip model din sîrmă, pînă la sistemele orientate pe suprafețe sau pe volume, numite și de modelare a solidelor ("Solid Modeling"). Sistemele orientate pe volume sînt răspîndite mai ales în domeniul construcțiilor mecanice, deoarece cu ajutorul lor se pot calcula modificările de volum, sau greutate, ale unor piese după o prelucrare, de ex. ștanțarea unei găuri. Aici CAD se apropie foarte mult de domeniul fizicii științifice.

Programe de desenare orientate CAD și programe CAD pentru începători

Aceste programe sînt un amestec de programe de desen și programe CAD. Din această categorie fac parte: Autosketch, Gfa-Draft, In-a-Vision, etc. În SUA această categorie este denumită "CADD", primul D venind de la Drafting, deci desenare, iar al doilea D vine de la Design, dar ar trebui să fie doar o jumătate de D, deoarece partea de proiectare nu este tot atît de performantă ca și cea a sistemelor CAD profesionale.

Aceste produse sînt destul de bune pentru lucrările simple de construcții. Unele produse nu sînt

altceva decît versiuni simplificate ale programelor CAD profesionale, care permit utilizatorului să facă primii pași în această lume la un preț acceptabil. Așa s-au născut de exemplu Easycad și Fastcad. Altele, ca In-a-Vision de exemplu, au fost obținute prin dezvoltarea unor programe de desenare (în cazul nostru din Windows Draw). Uneori aceste programe pot fi folosite tot atît de simplu ca și programele de desen, altele sînt la fel de greoaie ca și programele CAD.

Programe de umbrire (Rendering) și de iluminare (Raytracing)

Cele mai multe programe CAD de modelare a solidelor posedă deja și funcții de umbrire. În același scop pot fi folosite și produse specializate, cum ar fi Autoshade. Ele se folosesc, de regulă, pentru a da impresia că obiectele construite sînt reale.

Majoritatea programelor de umbrire sînt oferite ca programe auxiliare unor programe CAD. Deoarece ele sînt folosite de obicei de arhitecți care vor doar să privească rezultatul, rutinele de umbrire trebuie să fie rapide. Din acest motiv se folosesc algoritmi de umbrire "Phong" și "Gouraud", care produc imagini semireale.

Programele de iluminare au nevoie sensibilă de mai mult timp. Ele calculează pentru fiecare punct de pe o suprafață modul de reflectare al luminii și nuanțele de culoare ale umbrei acestui punct. Această metodă oferă imagini mult mai reale decît celelalte metode, dar este gîndită pentru a fi utilizată mai ales "peste noapte".

Programe de trasare automată (Autotracer)

Aceste programe sînt specializate în vectorizarea desenelor orientate pe puncte. Ele sînt capabile astfel să tipărească și imaginile scanate cu o rezoluție foarte mare, de exemplu pentru o instalație tipografică. Majoritatea programelor de ilustrare conțin și funcții autotra-

ce. Un produs nou, Adobe Streamline, s-a specializat pe trasare și poate transforma în grafice orientate pe obiecte, chiar și imagini complexe. Înainte de a folosi rezultatul trasării, aceste imagini mai sînt prelucrate, de regulă, cu un program de ilustrare.

Denumirea vine de la cuvintele "auto" (automat) și "tracere" (urmărire, trasare). La copierea unei imagini cu mîna se urmărește conturul imaginii aflate sub folia pe care se copiază. În mod asemănător lucrează și aceste programe.

Trasarea automată, autotracer, nu trebuie confundată cu găsirea automată a traseului, autoroute. Programele profesionale de proiectare a cablajelor imprimate dețin de obicei o funcție autoroute, care caută în mod automat cea mai bună legătură (cel mai bun traseu), între două puncte de pe placă.

Și restul ...

În această enumerare nu au fost amintite multe utilitare și programe experimentale. Bineînțeles că o parte din ele nici nu pot fi categorisite corect. Unele produse sînt oferite de producător cu eticheta "performant CAD", cu toate că performanțele amintesc mai degrabă de un program de desen simplu.

Multe produse rămîn în umbră, cu toate că sînt bune, deoarece mulți cumpărători încearcă să le încadreze între niște limite clare. Trasarea limitelor se dovedește însă a fi o greșeală - nu vă alegeți softul doar după categorie ci și după caracteristicile de performanță. Astfel, de ex., un program de desenare cu funcții CAD simple poate face uneori servicii mai bune decît un sistem CAD, greu de învățat. Un alt exemplu: combinația dintre un program de grafică economică și unul de pictură, suficientă pentru multe scopuri costă de multe ori doar pe jumătate din cît costă așa-numitele programe de prezentare. Este bine deci să reflectați de două ori înainte de a vă cufunda în "haosul graficii".

(R.M.)

Artă în 256 de culori

Programele de pictură nu au nici o utilitate. Ele nu sînt de folos nici planificării bugetului, nu produc nici scrisori de afaceri și nu ne amintesc nici termene importante. În multe alte privințe, ele depășesc însă aplicațiile "serioase" de pe PC-uri, ele dau mai multă satisfacție. Pe primul loc în această categorie se află "Deluxe Paint II", al lui Electronics Arts, care în versiunea extinsă este denumit "Enhanced". Cei mai importanți concurenți sînt "VGA Paint" al firmei Rix Softworks și "PC Paintbrush IV Plus" al firmei ZSoft. Cele trei programe au un singur lucru în comun: ele folosesc ecranul PC-ului ca pînză și mouse-ul ca pensulă.

Deluxe Paint poate lucra cu o paletă de 256 de culori, dar rezoluția pe care o poate atinge este de doar 320X200 de pixeli. Concurenții săi pot atinge o rezoluție de pînă la 640X400 de pixeli. VGA Paint mai oferă și posibilitatea de a se lucra cu rezoluția, nedocumentată, de 360X480 de pixeli. Acest lucru este util atunci cînd placa VGA nu are o rezoluție de 640X400 de puncte, în acest caz la nevoie se poate lucra și cu rezoluția mai slabă de

320X200 de puncte.

La pornirea programelor, cîteva diferențe sînt frapante: în timp ce Deluxe Paint și PC Paintbrush lucrează cu tehnica ferestrelor și cu meniuri Pull-Down, VGA Paint folosește meniurile Pop-Up. Aceasta înseamnă că pe ecran este vizibilă doar grafica, meniurile apărînd doar la cerere, în timp ce la Deluxe Paint și la PC Paintbrush rîndurile de meniu și "icon"-urile sînt mereu prezente. Acest lucru are atît avantaje cît și dezavantaje: pe de o parte structura meniurilor și deci modul de utilizare este mereu la dispoziție, pe de altă parte ecranul nu poate fi folosit în întregime. Acest neajuns se observă mai ales la Deluxe Paint, suprafața utilizator poate fi văzută în întregime doar pentru afișarea unei imagini, pictarea obiectelor care ar putea umple ecranul este imposibilă, datorită suprafeței mici puse la dispoziție pentru desen. Aceste obiecte sînt practic tăiate la marginea ferestrei.

Mai importante decît aceste lipuri sînt însă funcțiile de desenare pe care le oferă fiecare program în parte. La acest capitol Deluxe Paint se află în frunte. Datorită multitudinii posibilităților de manipulare, PC Paintbrush și VGA Paint rămîn privitori de pe margine. De ex. imaginile pot fi îndoite cilindric, pot fi întoarse și răsucite, paleta de culori poate fi animată (poate fi parcursă ciclic) și fiecare desen (porțiune de desen) poate fi multiplicat simetric cu funcția de caleidoscop. Cel mai important lucru este însă modalitatea de combinare a tuturor funcțiilor. Prin aceasta chiar și cele mai complexe modificări pot fi făcute în modul zoom (efect de lupă), și fiecare funcție poate fi cuplată cu un Anti-Aliasing-Effect cu ajutorul căruia pot fi șterse treptele colțuroase de pixeli. Avantajul es-

te clar: mai mulți pași individuali pot fi reduși la unul singur.

PC Paintbrush aduce, în schimb, alt avantaj și anume posibilitatea de a prelua imagini prin intermediul unui scanner. Această posibilitate lipsește atît la Deluxe Paint cît și la VGA Paint. Ca și număr de funcții PC Paintbrush nu poate ține pasul cu Deluxe Paint. El este mai bun însă pentru desenaarea unor detalii mici dar fine. El are posibilitatea de mișcare limitată a cursorului orizontal și respectiv vertical, astfel încît se pot desena linii drepte orizontale sau verticale chiar și cu mîna liberă. Și funcția de introducere text este interesantă. Textele pot fi prevăzute în mod automat cu umbre și culori, ceea ce în cazul celorlalte două programe nu se poate realiza decît cu mari eforturi.

VGA Paint se deosebește de cei doi concurenți ai săi în două puncte: este singurul care pune la dispoziție întregul ecran pentru desen și chiar și pe cele mai mici plăci grafice VGA pune la dispoziție un mod de lucru care permite folosirea unei palete de 256 de culori la o rezoluție mult mai bună decît învechita 320X200.

În concluzie se poate preciza că utilizatorii unor plăci grafice VGA mici ar putea folosi VGA Paint, în timp ce PC Paintbrush ar trebui folosit atunci cînd trebuie scanate și prelucrate imagini deja existente. În sfîrșit, Deluxe Paint este cel care le depășește pe celelalte două, nu pune limite creativității utilizatorului, multitudinea posibilităților de a obține efecte speciale deschide perspective cu totul noi, iar prețul este inferior celui al concurenței.

*(Computer Persönlich 25/90,
Matthias Fichner)*



Tabel comparativ

Program	Deluxe Paint II Enhanced	Colorix VGA Paint	PC Paintbrush Plus
Date tehnice			
EMS	recomandabil	nu recunoaște	va recunoaște
Placa grafică	HGC, CGA, EGA, (E-)VGA	VGA	HGC, CGA, EGA, (E-)VGA
Rezoluția maximă	1024X768 puncte (16 culori)	1024X768 puncte (16 culori)	1600X1200 puncte (2 culori)
Nr. maxim de culori	256 (la 640X480 puncte)	256 (la 800X600 puncte)	256 (la 1024X768 puncte)
Funcții de desenare			
Mînă liberă	da, 3 variante	da, 2 variante	da, 3 variante
Punct	da	nu	nu
Linie	da, 2 variante	da	da, 3 variante
Curbe	da, 2 variante	da	da, 3 variante
Cerc/Elipsă	da, 6 variante	da	da, 8 variante
Dreptunghi	da, 4 variante	da	da, 4 variante
Polygon	da, 3 variante	nu	da, 2 variante
Umplere	da, 10 variante	da	da, 2 variante
Pulverizare	da, și cu conținutul pensulei	da	da
Funcții de secționare și pensulă			
Secționare dreptunghiulară	da	da	da
Secționare cu mina liberă	da	nu	da
Mărire/micșorare	da	da	da
Basculare	da	da	da
Rotire	da	da	da, 90 grade
Tragere	da	da	da
Îndoire	da	nu	nu
Umplere	da, 2 variante	nu	nu
Efecte			
Anti-Aliasing	da	da	nu
Colorare	da	nu	da
Transparență	da	nu	nu
Perspectivă 3D	da	nu	nu
Ungere	da	da	da
Umbrire	da	da	da
Strălucire	da	da	da
Animație culori	da	nu	nu
Degrade	da	da	da
Contrast	nu	da	da
Luminozitate	nu	da	da
Funcții text			
Seturi de caractere	21 mono, 2 multicolor	35	20 bitmap, 4 outline
Variante	4	3X4	7
Scalabile	nu	da, prin mărirea pixelilor	da, fonturile outline
Degrade	nu	nu	da
Umbrire	nu	nu	da
Fonturi policolore	da, două	nu	nu
Altele			
Lupă	da, de 2 pînă la 16 ori	da, de 4 pînă la 8 ori	da, de 2 pînă la 8 ori
Șabloane de culoare	da	nu	nu
Pagini de rezervă	1	1	niciuna
Paletă liber definibilă	da	da	da
Scanner	nu	nu	da
Pipetă de culoare	da	nu	da
Editor de tranziție	da	da	nu
Descompunere prin raster	opțional	opțional	nu
Caleidoscop	da, 3 variante	nu	nu
Produs			
Dischete	4X5,25" / 2X3,5"	3X5,25" / 1X3,5"	6X5,25"
Manual	230 pagini	260 pagini	210 pagini
Manual de referință	da	nu	nu
Preț (DM)	399	495	499
Producător	Electronic Arts	Rix Software Inc	ZSoft Corporation

Formate grafice

Cine dorește să prelucreze cu ajutorul unui program de grafică o imagine obținută cu un alt program, se lovește de probleme. De obicei programele nu vorbesc aceeași limbă.

Cine scrie texte cu ajutorul calculatorului, acela dorește să le și illustreze. Din acest motiv, de multe ori, utilizatorul stă în fața unei probleme: "cum va arăta imaginea grafică, desenată cu programul x, în cadrul procesorului de texte y, sau în cazul programului de publicistică (Desktop Publishing) z?". Sau: "cum aş putea prelucra cu succes o imagine creată cu un anumit program de grafică, cu propriul program de grafică?".

Motivul: pentru întințitate se cere o sumedenie de programe de grafică, publicistică sau procesare de texte. Pentru a atinge cea mai bună calitate posibilă, în ceea ce privește programele, aproape fiecare producător de soft merge pe drumul său propriu, și lucrează cu formate fișier proprii. Din acest motiv s-a ajuns la o junglă de formate de date în care te poți rătăci foarte ușor. Pentru a găsi totuși drumul cel bun aveți nevoie de câteva cunoștințe de bază și de programele potrivite. În acest articol încercăm să vă dăm câteva informații necesare.

Să începem cu întrebarea cea mai importantă: Ce este de fapt un format fișier? Fiecare dintre programele de aplicație - fie de desenat, de prelucrare de texte sau bază de date - are câte o funcție prin care rezultatul muncii lor (de ex.: scrisori, imagini, listă de adrese) poate fi memorat pentru o durată mai îndelungată. Sub forma unor combinații de biți, deci într-o formă pe care calculatorul o poate percepe, programele înregistrează informațiile pe dischete sau pe harddisk-uri. Rezultatul este un așa-numit fișier. Numele lui este conținut în directorul dischetei sau harddisk-ului. Dacă cerem să ve-

dem conținutul unui disc, cu comanda DOS DIR, putem observa imediat că numele fișierului este format din două părți, și anume: prima din maxim opt caractere (numele sau prefixul), care este stabilită de utilizator, de ex.: "TEST", a doua este terminația (extensia sau sufixul), care se pune după punct și care constă din maxim trei caractere. Extensia este stabilită, de regulă, în mod automat de către program (de ex.: PCX, DRW, etc.).

Conținutul propriu-zis al fișierului, imaginea deci, poate fi dezvăluită dintr-un fișier grafic, doar cu programul cu care a fost creată. De ex. atunci când fișierul este citit de program, interpretat, și imaginea este afișată pe ecran.

De obicei, încercarea de a citi o imagine, cu un alt program decât cel cu care a fost creată, nu reușește, pentru că metoda cu care de ex. programul de grafică A cifrează informația "desenează un cerc cu raza x, având centrul în y, de culoarea z", nu este normată și nu poate fi înțeleasă de alte programe.

În ceea ce privește formatele grafice pe PC-uri, există o multitudine babilonică de limbaje. Din acest motiv este greu să poți avea o privire de ansamblu asupra diferitelor formate. Pentru a vă crea o imagine, vă vom prezenta cele mai importante formate de fișiere utilizate de programele de grafică, pe scurt formatele grafice.

Formatele bitmap

Unele formate sînt legate direct de un program grafic, care scrie și citește exact acel format. Altele au devenit, între timp, un cvasistandard, care este înțeles de mai multe programe diferite. Formatul fișierului poate fi recunoscut după cele trei litere ale extensiei sale.

PCX

Formatul PCX (cu extensia PCX) este unul din formatele cele mai răspândite pentru programele bitmap și în același timp este și cel mai vechi format de grafică în domeniul PC-urilor. El este foarte strîns legat de firma ZSoft care a elaborat programul de pictură foarte popular "PC Paintbrush", ajuns între timp la versiunea a IV-a. Formatul PCX a traversat unele modificări, deoarece Paintbrush a continuat să fie dezvoltat, mai ales a crescut numărul culorilor ce pot fi reprezentate. Dacă la început puteau fi reprezentate maxim 16 culori (ceea ce corespundea performanțelor plăcilor grafice CGA și EGA) astăzi s-a ajuns la un număr maxim de 256 de culori dintr-o paletă de 262144, deci adaptat la standardul VGA de astăzi. Cum pot fi alese cele 256 de culori din mulțimea uriașă de peste 1/4 de milion? Soluția este simplă. La începutul fișierului PCX se definesc o așa numită paletă, în care sînt înscrise definițiile valabile ale imaginii, deci care sînt cele 256 de culori folosite. Mărimea și rezoluția imaginilor nu este limitată de format, ele depind doar de capacitatea programului de pictură, de memoria disponibilă a calculatorului folosit și de cartela grafică înglobată. Datorită mărimii lor, fișierele PCX se memorează mai ales sub formă comprimată.

TIFF

Acest format grafic, (prescurtare de la "Tag Image File Format", extensia fișierelor: TIF), a rezultat ca urmare a înțelegerii survenite în anul 1986 între firmele Aldus, Microsoft și producători de scannere de frunte. TIFF este foarte aproape de a deveni standardul pentru grafica bitmap, mai ales în domeniul profesional și semiprofesional, atît pe PC cît și pe Apple.

Spre deosebire de PCX, pentru

TIFF nu există un program anume care să genereze acest format, ca format de bază, existând în schimb foarte multe programe care pot citi și scrie acest format. Cu versiunea actuală, 5.0, se pot prelucra imagini alb-negru, grafice cu diferite nuanțe de gri și imagini cu pînă la 16.7 milioane de culori (în format 24-bit-TIFF).

Structura internă a fișierelor TIFF este fixă și este cu ceva mai complexă decît cea a fișierelor PCX. Și în acest caz se folosește procedeul de comprimare a datelor. Nu există o limitare în ceea ce privește mărirea și rezoluția imaginilor.

GIF

Formatul GIF provine din SUA. El a fost proiectat de firma CompuServe din Ohio. CompuServe este unul dintre cei mai mari producători de cutii poștale electronice și de sisteme informatice ce pot fi folosite prin intermediul unui modem, din lume.

GIF a fost proiectat, mai ales, pentru a pune la dispoziția sistemelor uzuale PC, Atari ST, Mac, Amiga, un format grafic, care să poată memora datele făcînd economie de spațiu. De ce chiar programatorii de la CompuServe au pus accent pe acest lucru este clar: timpul necesar transferării graficii prin telefon este lung și costă bani. Cu cît fișierul este mai mic, cu atît mai repede se poate efectua transferul.

Și GIF, ca și PCX, poate folosi 256 de culori dintr-o paletă de 262144 de culori. Pînă la această oră nu cunoaștem decît un program de aplicație care știe scrie și citi formatul GIF: este vorba despre programul Autodesk Animator. Există, totuși, cîteva programe freeware, ca de ex. "VPIC.EXE", care poate prezenta și imagini GIF.

LBM / IFF

Acest format grafic este răspîndit mai ales pe calculatoarele Amiga, numele lui vine de la: "Interchange File Format", extensia fișierelor este IFF. Acest format

grafic și-a făcut intrarea în lumea MS-DOS în momentul în care programul de pictură "Deluxe Paint" a fost transcris de pe Amiga pentru PC. Pe Amiga fișierele aveau extensia LBM. Deci imaginile grafice realizate cu Deluxe Paint pe Amiga sînt compatibile cu cele create de versiunea PC a aceluiași program, ele putînd fi încărcate și prelucrate pe PC. Deoarece calculatoarele din familiile Amiga și PC, (cu cartelă grafică EGA sau VGA), au rezoluția ecranului și raportul lățime/înălțime diferite, la transferul imaginilor apar deformări, care trebuie corectate ulterior.

Amiga mai posedă și modul grafic HAM cu 4096 de culori simultane și cîteva programe de pictură Amiga, de ex. "Photon Paint", folosesc această multitudine de culori. Deoarece nici cartelele VGA moderne, (standard PC), nu pot stăpîni acest mod de lucru, la asemenea grafice pot apare deformări crase de culori, care trebuie corectate ulterior de mîină.

Formatul IFF a fost proiectat nu doar pentru memorarea de grafică ci și pentru memorarea unor piese muzicale, zgomote, sau text, în acest format standard. În acest caz nu s-a produs nici o extensie înafara domeniului Amiga. Și IFF/LBM folosește metoda de comprimare a datelor.

MSP

Această prescurtare, care coincide și cu extensia fișierelor, este folosită pentru "Microsoft Paint" și este numele formatului folosit de programul cu același nume. Microsoft Paint a fost livrat împreună cu suprafața utilizator grafică Microsoft Windows pînă la versiunea 2.11. MSP este un format alb-negru curat, neexistînd opțiunea de culoare. Formatul MSP există în două variante 1.0 și 2.0. Versiunea 1.0 a fost folosită doar de Microsoft Paint, livrat împreună cu Windows 1.0. Dacă se dorește folosirea unei astfel de imagini cu versiuni mai noi ale programului, trebuie folosit programul de conversie "CVTPAINT.EXE", care a fost livrat

odată cu Windows pînă la versiunea 2.11. Începînd cu versiunea Windows 3.0, Microsoft Paint a fost înlocuit cu o variantă specială a clasicului Paintbrush (proiectat tot de ZSoft).

Această versiune Paintbrush poate utiliza pe lîngă formatul inițial al casei ZSoft, PCX, și fișiere MSP vechi, pe care le poate citi, dar nu mai poate scrie în format MSP.

BMP

BMP este prescurtarea pentru Bitmap, extensia BMP, și este formatul pe care versiunea Paintbrush din Windows 3.0 îl poate scrie și citi în mod standard. În acest format grafic sînt livrate și imaginile pe care Windows 3.0 le poate afișa pe fundalul ecranului. Se pare că Microsoft are gînduri mari în legătură cu acest format, pentru următoarele versiuni ale lui Paintbrush. Dacă în Paintbrush alegem meniul "Memorare sub ..", atunci după clic apare cîmpul "opțiuni" cu mai multe posibilități de memorare, dintre care se poate alege formatul de memorare: fie format BMP cu 2, cu 16 sau cu 256 de culori, fie format 24-bit, deci cu 16.7 culori. La Windows 3.0 s-au pus deci bazele unei prelucrări cu culori naturale.

IMG

Acest format grafic ar trebui să fie cunoscut posesorilor de calculatoare Schneider de tipul PC1512 și 1640. Odată cu aceste calculatoare, Digital Research a livrat suprafața utilizator grafică "GEM", care conținea programul "GEM Paint". Acesta folosea formatul IMG (Image), care suporta 16 culori. Cel care deține multe imagini în format IMG nu trebuie să renunțe la ele, deoarece există nenumărate posibilități de a le converti în alte formate mai uzuale.

Targa

Formatul Targa provine din lumea profesională a minicalculetoarelor, a supermini-urilor și a stațiilor

de lucru. Pe PC aceste fișiere au extensia TGA. Formatul Targa este un format bitmap care poate reprezenta 16.7 milioane de culori și care este folosit în prelucrarea profesională a imaginilor și în rețușarea lor. Pe PC există și programe de grafică vectorială, cum ar fi Mirage, care pot memora imagini în formatul TGA, deoarece recalcularea unei imagini vectoriale într-un format bitmap nu este mare lucru. Conversia în sens invers nu se poate face, însă, decât cu pierderi mari de calitate.

Astăzi formatele TGA sînt folosite foarte des pentru fixarea imaginilor pe diapozitive deoarece aproape orice studio specializat poate prelucra formate TGA.

SCx

Acesta este formatul de bază al programelor "Colorix EGA Paint" și "Colorix VGA Paint" de pe PC. Primele două caractere ale extensiei sînt întotdeauna caracterele "SC", al treilea caracter diferind în funcție de rezoluție și de numărul culorilor folosite. La rezoluția VGA normală, cu 640X480 puncte și 16 culori, extensia este "SCP". În acest format pot fi memorate imagini în toate rezoluțiile cuprinse între CGA (320X200 puncte cu 4 culori din 16 posibile) și Super-VGA (1024X768 puncte și 256 de culori simultan).

MAC

Un alt format bitmap foarte răspîndit pe Apple Macintosh, alături de TIFF, este MAC, formatul de bază în programul de pictură "Macpaint". El poate fi convertit în alte formate cu ajutorul unor utilitare.

Formate vectoriale

În ciuda multitudinii de formate bitmap, pe PC există totuși două cvasistandarde, și anume PCX și TIF, care pot fi citite de aproape toate programele Desktop Publishing.

În cazul formatelor vectoriale este cu totul altfel - nu neapărat mai rău.

Toate programele de grafică vectorială au formate proprii de date, care posedă caracteristici specifice.

DRW

DRW este o prescurtare pentru "Drawing" și este formatul versiunii 3.01 al lui Micrografx Designer. Pe lîngă acesta coexistă și versiunile 1.0 și 2.0 care pot fi prelucrate de cea mai nouă versiune a programului. Caracteristici: utilizează curbele Bézier și poate folosi pînă la 16.7 milioane de culori.

PIC

PIC este un format destul de rar întîlnit în ultima vreme, el poate fi scris de programul "In-A-Vision" al casei de soft Micrografx. Programul In-A-Vision a apărut în iunie 1985 și a fost una din primele aplicații pentru Windows și unul dintre primele programe de desenare orientate pe obiecte de pe PC.

GRF

GRF este formatul programului Charisma, un program pentru grafică economică, produs de Micrografx. Acest format stochează atît informațiile grafice cît și informații despre datele dintr-un fișier propriu. În rest se aseamănă cu formatul DRW. Charisma poate memora o imagine și în format DRW.

XLC

XLC este prescurtarea pentru "Excel Chart". Acesta este formatul în care programul de calcul tabelar Excel, al lui Microsoft, își salvează graficele.

Lotus PIC

Lotus PIC este formatul folosit de programul de calcul tabelar Lotus 1-2-3 pentru memorarea graficelor și imaginilor produse.

Pict I & II

Acesta este un format vectorial

foarte răspîndit pe Apple Macintosh. Este folosit, de ex., de programul de desen "MacDraw".

GEM

Acest format, cu extensia GEM, este folosit de suprafața utilizator grafică cu același nume pentru memorarea graficelor vectoriale. Sub acest nume există două formate: în primul rînd formatul simplu al lui "GEM Draw", și în al doilea rînd formatul lui "GEM Artline 2.0", care posedă, față de Draw, cîteva caracteristici în plus, de ex.: tranziția culorilor și așa-numitele curbe Bézier.

CDR

CDR este formatul de bază folosit de programul grafic "Corel Draw".

CED

CED este formatul de bază folosit de programul grafic "Arts & Letters".

DXF, SLD, ADI

Formate de export/import pe care programul "Autocad" al firmei Autodesk le folosește pentru schimbul de date cu alte programe.

WMF

WMF este prescurtarea pentru "Windows Metafile", un format vectorial folosit de Windows pentru schimbul de date între programe.

CGM

CGM este prescurtarea pentru "Computer Graphics Metafile" și un format de fișier foarte des folosit pentru transferul de date între diferite programe de grafică și între diferite sisteme de calcul. CGM nu este formatul de bază al unui anumit program grafic, ci apare ca opțiune de export/import a unor programe.

(continuare în pag. 23)

Babilonia limbajelor de programare

Unul știe doar C, altul are încredere doar în dialectele de Basic, un al treilea fără compilatorul Pascal ar fi pierdut, și fiecare crede că doar limbajul "lui" își poate da mințea cea de pe urmă. Această introducere își propune să arate de ce greșesc cu toții.

Cel care consideră că unul sau altul dintre limbajele de programare reprezintă singura și cea mai bună soluție, acela se înșeală. Legea inerției este responsabilă de inexistența unui limbaj universal pentru toată lumea. Căci cine nu este obligat de profesie, sau de specializare să treacă de la un limbaj, învățat cu greu, la un alt limbaj, acela nici n-o va face. Nici un african nu va învăța chineza, doar pentru faptul că mai mult de un miliard de pămînteni o folosesc.

În plus, în ceea ce privește limbajele evoluat, nu se observă diferențe semnificative în ceea ce privește performanțele compilatoarelor. Nu cunoaștem nici o problemă care să poată fi rezolvată doar în C, doar în Basic sau doar în Pascal.

Strategii iscusite ai reclamei și fețiștii ai compilatoarelor reușesc mai întotdeauna să demonstreze, cu ajutorul unor exemple bine alese, că un limbaj anume este cel mai bun. Ei n-ar trebui crezuți pe cuvînt, deoarece un adevărat profesionist ar trebui să stăpînească cel puțin trei limbaje evoluat și două asamblatoare. Un motiv suficient pentru această situație l-ar putea constitui, de ex., posibilitatea ca lucrînd într-o echipă să se ia decizia că o anumită aplicație trebuie scrisă într-un anumit limbaj de programare, de ex. C, sau, de ex., se poate întîmpla ca o aplicație deja scrisă, să zicem în Pascal, să trebuiască dezvoltată folosind același limbaj de programare. Mediul de dezvoltare API, (Application Programm Interface - o colecție de rutine, gata

scrise, pentru utilizarea caracteristicilor sistemului), oferit de sistemul de operare pe care se lucrează, joacă un rol hotărîtor în alegerea limbajului de programare ce urmează a fi folosit.

Sistemul de operare al calculatoarelor Apple Macintosh, de ex., a fost scris în Pascal, toată documentația și toate exemplele puse la dispoziție sînt concepute în Pascal. Deci cel mai eficient limbaj de programare a sistemului este limbajul Pascal. Altfel se prezintă situația în cazul suprafeței utilizator PC Windows 3.0. Ea a fost programată în limbajul C, în C fiind scrise și exemplele, bibliotecile și fișierele antet (header) ale sistemului. Un programator fără cunoștințe de C se descurcă greu în acest caz.

API-urile scrise în C domină net: MS Windows, OS/2, GEM, Atari ST, Amiga, X-Windows și întreaga lume Unix sînt scrise în C. La Apple, Pascal este încă puternic, dar C rămîne limbajul principal de programare a sistemelor. Tot aici își găsesc principalul domeniu de aplicabilitate și asambloarele. Ele sînt folosite atunci cînd trebuie rezolvată o problemă critică în mod optim, sau cînd trebuie făcută o programare apropiată de hardware. De aceea compilatoarele bune permit inserarea unor secvențe de instrucțiuni scrise în limbaj de asamblare direct în textul sursă, iar limbajele de asamblare moderne, cum ar fi MASM 5.1 al lui Microsoft, oferă interfețe confortabile spre C.

Concluzia este că programele pentru un API nu pot fi scrise în Basic sau Pascal. Într-adevăr, fără un efort considerabil, nu pot fi scrise aplicații Windows adevărate în Turbo Pascal. Totuși pot fi scrise programe care să arate asemănător. Cu toate că a durat destul de mult, între timp și Borland sprijină, cu Turbo Pascal 6.0, programarea ferestrelor, a meniurilor

Pull-Down și a căsuțelor de dialog, cît și deservirea cu mouse-ul. Nici Microsoft, inventatorul acestei suprafețe SAA, n-a fost mai iute și abia odată cu versiunea 7.0 a Basic-ului a livrat și modulele necesare.

În concluzie: cine dorește să programeze un sistem cu un API scris în C, în această categorie fiind în primul rînd OS/2 și Windows 3.0, acela nu poate trece pe moment peste C. La Amiga, Atari ST și Macintosh s-au dezvoltat între timp versiuni Basic, C și Modula care sprijină un acces direct la API. În mod cert cauza acestei stări de fapt o constituie faptul că la aceste calculatoare funcțiile componente fac parte din sistemul de operare, în timp ce Windows nu este decît o extensie a sistemului de operare DOS, pe care o poți avea sau nu.

După cum am mai precizat, versiunile Basic moderne pot face același lucru ca și în Pascal, dacă luăm în considerare doar ceea ce se poate face cu aceste sisteme. Cum se poate face aceasta este o altă întrebare. Din punctul de vedere al unui programator Pascal, o instrucțiune de genul "DIM i AS INTEGER" în loc de "i integer", într-o declarație de articol, în care cuvîntul "record" nici nu apare, sună ciudat. Pentru un programator care a crescut cu Basic și cunoaște această sintaxă, care e în urmă cu doi ani nici nu există, totul pare cît se poate de normal. El nu va găsi nici un motiv de a trece la Pascal. Dar cu ce limbaj să începi?

Diferența dintre cele două limbaje constă în faptul că declarația variabilelor în Basic este facultativă, în timp ce în Pascal este obligatorie, o conversie de tip se face automat în Basic, în timp ce în Pascal trebuie făcută explicit.

Altfel spus: în Pascal papa Worth ia programatorul de mîna și îl ține

strada, în Basic are voie să treacă singur, iar în C poate trece chiar și atunci când semaforul este pe roșu. Deci un începător n-ar trebui să înceapă cu C, mai rămân Basic și Pascal.

Un avantaj la Basic era ritmul interpretorului, introducerea instrucțiunilor, urmată de RUN, ducea la apariția instantanee a răspunsului pe ecran, în timp ce la Pascal trebuia trecut prin lanțul editor-compiler-linkeditor-test. Cu ajutorul unor medii de dezvoltare integrate și a unor compilatoare foarte rapide, (un principiu care a adus faimă lui Turbo Pascal), timpul de execuție a ajuns între timp același, dacă facem abstracție de programele foarte mici. În cazul lor Basic se află în mod cert în frunte. Programatorii folosesc acest avantaj atunci când doresc un transfer rapid de fișiere între două calculatoare, cu un program de cinci linii. În acest caz MS-Basic cu al său OPENCOM nu poate fi întrecut de nimeni, și mai are încă un avantaj: specificarea parametrilor de interfață se face și la Amiga și la Macintosh și la PC la fel. În schimb la Pascal, și chiar la C, pentru acest lucru trebuie cunoscute câteva detalii despre interiorul sistemului de operare și/sau despre funcțiile de bibliotecă. Până le-am afla, sau am

căuta în manual, programul Basic ar rula deja.

Exemplul ce urmează va arăta un alt avantaj al Basic-ului: un programator a scris un program destul de complex în MS-Basic pe Macintosh. După câțiva ani îl transferă pe Amiga, programul va pomi instantaneu. Orice program Basic PC, fără instrucțiunile PEEK și POKE, funcționează și pe un Macintosh, dar nici un program Pascal PC, scris în Turbo Pascal, nu va funcționa pe un Mac sub Turbo Pascalul de acolo.

Se poate spune că MS-Basic, (sau GW Basic), a devenit un standard pentru mai multe calculatoare, în timp ce la Turbo Pascal, limbajul cel mai standardizat la origine, diferențele între dialecte sînt sensibile.

Și o ultimă deosebire: datele și programele pot fi structurate foarte bine și în Basic-ul modern, dar acest lucru se face facultativ, în timp ce la Pascal această structură formează conceptul limbajului și apare în mod natural. De aceea programele mari, mai ales cele orientate obiect, se pot formula mult mai elegant în Pascal.

Începătorul are deci două posibilități: fie începe cu Basic, caz în

care la început nu va trebui să facă eforturi prea mari, dar apoi trebuie să treacă singur la învățarea tehnicilor moderne de programare, fie începe cu Pascal, va studia mai mult limbajul decât soluțiile problemelor, dar îi va fi mai ușor mai târziu. În orice caz, dacă primul limbaj învățat este Pascal, al doilea se învață mai ușor.

Decizia de a utiliza unul sau altul dintre limbaje va trebui să depindă de scopul pentru care programatorul dorește să utilizeze compilatorul. Tabelul anexat poate servi pentru o orientare corectă. S-au acordat note între 0 și 3, 0 pentru cea mai rea apreciere și 3 pentru cea mai bună, pentru fiecare criteriu în parte.

Și-au mai rămas doar 10

Oferta de noi și noi medii de programare este de necuprins. Dacă stai însă să alegi făina de țărțe, atunci mai rămân doar câteva. Vă prezentăm în continuare cele mai importante 10 compilatoare.

În căutarea celor mai bune compilatoare te lovești inevitabil de două nume: Microsoft și Borland. Aceste două firme sînt, de multă vreme, cei mai mari, și aproape și singurii, ofertanți de medii de dezvoltare software profesionale. Cu multă pricepere au reușit să-și împartă piața. Ca urmare a acestui cvasi-dublu monopol există o apropiere între structura și puterea compilatoarelor oferite.

Între-timp a devenit neimportant ce compilator este folosit: Basic, C, Pascal sau chiar Assembler. Puterea limbajelor de nivel înalt a devenit atît de mare încît nu mai există probleme care să poată fi rezolvate doar într-un anumit limbaj. Assembler-ul își păstrează și pe mai departe rolul de unealtă a specialiștilor.

Și pentru a face puțină lumină în lumea compilatoarelor, pe următoarele pagini, vă prezentăm cele mai importante 10 compilatoare. De remarcat este faptul că

Criteriu	Asamblor	C	Basic	Pascal
Efort de învățare redus	0	1	3	2
Probleme puține la început	0	1	3	2
Deschizător de drumuri	0	2	1	3
Programe mici	0	1	3	2
Obținerea rapidă a unei soluții	0	1	3	2
Programe mari	0	3	1	2
Proiecte mari (echipă)	0	3	0	2
Structuri complexe de date	0	2	1	3
Structuri de program complexe	0	2	1	3
Apropiat de sistem	3	2	0	1
Apropiat de hardware	3	2	0	1
Soluții cu timp de rezolvare critic	3	2	0	1
Cod compact	3	2	0	1
Acces API	1	3	0	2
Ofertă module/biblioteci	0	3	1	2
Utilitare suplimentare	0	3	1	2
Aplicații OOP	0	2	1	3
Facilități OOP	0	3	1	2
Portabilitate	0	3	2	1
Siguranță în viitor	0	3	2	1

Tabelul arată care limbaj pentru care scop este mai indicat

prezentarea va fi amestecată, deci vor fi puse în relație compilatoare de diferite categorii. Prezentarea nu urmărește să stabilească care compilator de Basic este cel mai bun compilator de Basic, ci care este cea mai bună unealtă de dezvoltare aplicații ?

Quick-C cu Quick-Assembler 2.5

Quick-C cu Quick-Assembler este o combinație dintre un compilator C și un asamblor care funcționează sub aceeași suprafață utilizator. Pachetul este compatibil cu MS-C 6.0, MASM 5.1 și cu Debugger-ul din trusa de scule (Programmer's Workbench) oferită de MS-C 6.0. Diferența de bază față de versiunea profesională este că pot fi create programe care să ruleze doar în Real-Mode și nu și pe OS/2.

Mediul integrat de dezvoltare dispune de meniuri pull-down și de ferestre în standardul SAA. Din interiorul editorului pot fi scrise programe, compilate, linkeditate, lansate în execuție și depanate.

La acestea se adaugă un sistem de help-uri contextuale, de primă mână, care conțin practic toate manualele, un curs de C și o multitudine de exemple. În ceea ce privește timpul turn-around, timpul de așteptare în cazul asamblorului este practic nul. Compilatorul C poate avea nevoie de câteva secunde deoarece scrie chiar la început pe disc. Dacă se dispune de un RAM-disc, 4000 de instrucțiuni sînt compilate în mai puțin de o secundă, ceea ce duce la concluzia că timpii de compilare depind doar de timpul de acces la harddisk. Depanatorul dispune de funcții tracing, backtracing, undo și permite fixarea de puncte de întrerupere. Conținutul variabilelor poate fi examinat și modificat, iar în cazul asamblorului același lucru este valabil și pentru conținutul registrelor.

Pentru Quick-C există biblioteci bogate, în standard ANSI, și există

și biblioteci speciale, de ex. de grafică. Asamblorul este super. În loc de bucla obișnuită editor- asamblor- linkeditor- EXE2BIN- debugger- editor, se rămîne în editor, o apăsare de tastă este suficientă pentru tot ciclul.

În furnitura de livrare, a versiunii 2.5, există trei manuale groase și un caiet numit "Up and Running". Manualele "Programmer's Guide" și "C for Yourself" sînt manuale de prima mână pentru începători.

Quick-Basic 4.5

Și Quick-Basic poate fi descris foarte simplu. Este Quick-C, cu singura deosebire că textul sursă prețins de compilator trebuie să fie în Basic. Acest Basic are câteva caracteristici, cu care, (din păcate), Pascal și C nu se pot lăuda. Astfel editorul dispune de un analizor de sintaxă care evaluează fiecare instrucțiune în momentul introducerii, ceea ce face ca timpul turn-around să fie practic nul. Motivul: fiecare instrucțiune este tradusă, imediat după introducere, într-un pseudocod (Threaded P-Code). Abia după ce se consideră că totul poate rămîne așa cum s-a introdus, se apelează (restul) compilatorului care atinge o viteză de 150.000 de instrucțiuni pe minut.

Datorită diferitelor structuri de control, inclusiv selector CASE, se pot folosi tehnici de programare care sã nu stea cu nimic în urma celor din Pascal. Sînt posibile chiar și definiții de articole și de tipuri de date Pascal. Cu ajutorul instrucțiunii DIM extinse pot fi făcute și declarații de variabile.

FUNCTION corespunde în modul de scriere și în efect cu funcția corespondentă din Pascal, dacă se utilizează și cuvîntul cheie STATIC atunci se cumulează și avantajele lui C. Pe lîngă funcții mai există și proceduri care cu excepția cuvîntelor cheie SUB și END SUB nu pot fi deosebite de cele din Pascal. Parametrii pot fi transmiși prin valoare sau prin referință (adresă).

Ergo:- Quick Basic 4.5 este un compilator Basic modern, foarte puternic.

Quick-Pascal 1.0

Ultimul reprezentant al familiei Quick este Quick-Pascal. El se înscrie pe aceeași linie ca și predecesoarele, dar pretinde text sursă în Pascal. Limbajul în sine este aproape o copie a lui Turbo Pascal 5.0, fără overlay-uri, dar cu o extensie pentru programarea orientată obiect. Există puține cuvinte cheie pentru Pascal orientat obiect: cel mai important este OBJECT care permite definirea de obiecte și clase. Mai sînt și cuvintele cheie OVERRIDE și INHERITED. Aceasta este totul, restul este Turbo Pascal.

Quick Pascal și Turbo Pascal 6.0, în comparație cu sistemele OOP de pe Macintosh și cu MS-Windows, au unele lipsuri, deoarece acolo tehnicile OOP sînt integrate deja în sistemul de operare, respectiv în extensia acestuia. Totuși Microsoft este pe fir, și chiar acest lucru vrea să-l modifice și încă în DOS.

OOP pe baza sistemului poate avea consecințe cu bătaie lungă. Metoda unui obiect s-ar putea numi atunci: dă-ți drumul atunci cînd se întîmplă un anumit eveniment în sistem. Aceste informații despre evenimente ar putea fi declanșate, ca și sub Windows, de mouse, de tastatură sau de ceas.

Combinația control prin evenimente și OOP este tehnica de programare ideală. S-ar defini simplu un obiect, de ex. un icon (image grafică care simbolizează o anumită acțiune), și i s-ar defini o metodă de genul: pornește atunci cînd se dă un clic pe mouse deasupra ta. Următorul obiect ar putea aștepta apăsarea tastei F1, și al treilea (ceasul) ar porni atunci cînd ar primi o anumită informație de la generatorul de tact. Ar fi treaba sistemului de operare să semnaleze obiectelor producerea unor astfel de evenimente. Avantajul: obiectele ar fi pe deplin independente unul

de celălalt, dar ar lucra totuși împreună, atunci când ar fi nevoie. S-ar putea adăuga un nou obiect, și deci noi facilități, într-un program, fără a fi nevoie să se modifice nici măcar o singură literă în programul existent.

Nu ne rămâne decât să sperăm că Microsoft va avansa rapid pe acest drum pe care a pornit.

Turbo Pascal 6.0

Turbo Pascal pînă în versiunea 5.0 este în general cunoscut în cercurile programatorilor. Versiunea 5.5 a adus în plus programarea orientată obiect (OOP), care a fost desăvîrșită în versiunea 6.0, și care permite scrierea unor programe cu totul noi. Pentru instalarea sistemului utilizatorul nu trebuie decât să tasteze: INSTALL. Un harddisk este obligatoriu, un mouse recomandat.

Mediul integrat de dezvoltare oferă meniuri pull-down și ferestre aproape în standard SAA. Din editorul Multi-Windows/Multi-File programele pot fi compilate, linkeditate, lansate în execuție și depanate.

Timpu turn-around este foarte scurt. Dintr-un program de mai bine de 1400 de instrucțiuni plus trei unități, pe un 386, după cîteva zecimi de secundă, se obține un fișier EXE în memorie, care în mai puțin de o secundă este transferat pe disc.

Dacă memoria disponibilă nu este suficientă - cca. 270 KBytes - se poate folosi varianta TPC, sau extensia profesională TPCX, care lucrează în mod protejat și în memoria extinsă.

Helpul contextual permite un help în genul celui de la compilatoarele Quick. Lăudabilă este noua fereastră "Registers" care permite examinarea conținutului registrelor unității centrale. Aceasta se potrivește cu noua instrucțiune ASM care permite inserarea în progra-

mul sursă a unor secvențe scrise direct în limbaj de asamblare.

Cu Turbo Pascal 6.0 se livrează și "Vision", un cadru orientat obiect pentru programe comandate de evenimente. În opt unități sînt oferite obiecte pentru manevrarea de ferestre suprapuse, pentru scalarea ferestrelor, controlul mouse-ului, crearea de meniuri pull-down, boxe de dialog, manipularea evenimentelor (tastatură, mouse) și o paletă bogată de controale.

Turbo Pascal 6.0 se dovedește consecvent cu el însuși în tratarea modalităților OOP introduse odată cu versiunea 5.5. Rezultatul este un mediu de dezvoltare, orientat grafic, reușit cu multe obiecte și metode utile.

Dar despre Turbo Pascal 6.0 și despre Turbo Vision vă vom spune mai multe într-un număr viitor.

MS-Basic 7.1 - PDS

O analiză a pieții din SUA făcută de Microsoft a dus la concluzia că peste 50% dintre cumpărătorii de Quick-Basic sînt programatori profesioniști și că 62% dintre aceștia scriu aplicații comerciale. S-a stabilit de asemenea că cerințele crescînde ale portabilității nu pot fi satisfăcute dacă nu sînt respectate următoarele condiții: Basic trebuie să fie tot la fel de rapid ca și Turbo Pascal, să susțină algoritmi puternici de tratare a bazelor de date, overlay-uri, OS/2 și EMS, și să ofere toate metodele moderne de structurare a programelor și datelor. Pe de altă parte se doreau și obișnuitele utilitare de la Make și bibliotecar (Library Manager), pînă la depanatorul simbolic comandabil prin mouse, și se dorea să se rămînă în mediul obișnuit al lui Quick-Basic.

Instalarea se face de pe 12 dischete și după instalare, în funcție de dorința utilizatorului, pe harddisk va fi ocupat un spațiu de 8 pînă la 14 MBytes; acestea da dimensiuni !

Basic 7 este compilatorul profesional BC 6 extins cu o multitudine de împrumuturi. Astfel funcțiile de calcul financiar și de calcul al datei, sau puternicele formatări numerice, provin de la Excel, depanatorul de la Codeview, și suprafața utilizator provine dintr-o încrucișare între Quick-Basic 4.5 și Workbench C 6.0.

Cu ajutorul lui pot fi scrise programe pînă la o dimensiune de 16 MBytes, cu overlay-uri și recunoaștere automată EMS. Dacă este disponibilă o memorie suficientă, overlay-urile sînt încărcate direct, altfel se face o swapare via harddisk.

Impresionantă este tratarea integrată a bazelor de date ISAM. Ea este activată cu instrucțiunea "OPEN FOR ISAM". Și încă o noutate este demnă de remarcat: acum programatorii își pot dota aplicațiile cu suprafețe grafice, modulele necesare aflîndu-se în furnitura de livrare.

Ar mai rămîne de remarcat doar că în medie se obține un cod cu 68% mai mic decît cel obținut cu Quick-Basic 4.5, și că în 37 de teste matematice s-a dovedit cu 225% mai rapid.

Assembler MASM 5.1

MASM face parte din categoria celor mai cunoscute compilatoare, sînt mulți cei care dispun de o versiune oarecare de MASM, dar sînt puțini cei care cunosc facilitățile noii versiuni 5.1.

În loc să vină cu sfatul lapidar: "luați un editor ASCII ..." MASM 5.1 dispune de un editor propriu: "M". Acesta este liber configurabil, Multi-Window/Multi-File și totodată un mediu integrat. Din cadrul editorului se poate asambla și linkedita, iar la apariția unei erori cursorul este poziționat automat în linia cu pricina. Suprafața utilizator orientată tastatură necesită mai întîi o acomodare, dar apoi se dovedește a fi extrem de puternică. Confortul obișnuit SAA apare din nou abia în

depanator, care provine dintr-o nouă versiune de la Codeview.

Instalarea de pe cele cinci dischete se rezumă la a tasta: "SETUP" și a răspunde la întrebarea: "OS/2 sau DOS?". De fapt sînt livrate două versiuni ale asamblorului. A fost testată doar versiunea DOS căci a programa sub OS/2 în limbaj de asamblare este o artă în sine.

Asamblorul însuși strălucește prin modalitățile puternice de macrouri, inclusiv recursive, printr-o mulțime uriașă de operatori pentru expresii complexe și prin tratarea procesorului 80386 și a coprocesorului 80387. O sintaxă strictă și o verificare strictă de tip ajută la evitarea scamelor. "Typos" se utilizează mai rar deoarece există noi directive de segmentare pentru toate apelurile la DOS și BIOS, așa-zise pentru mixare ("mixed"). Ultima permite, alături de alte directive, legătura cu limbajele de nivel înalt. Rutine asamblor pot fi scrise și ca funcții C sau Pascal. Dar spre deosebire de codul "inline" (inserat direct în sursă), rutinele scrise în limbaje de nivel înalt vor constitui module, sau biblioteci.

De anunțat ar mai fi utilitarele: BIND, IMPLIB, ILINK și EXEHDR, care permit conlucrarea cu OS/2 și Windows.

Turbo-Assembler 2.0

TASM este coresponentul Borland la MASM, versiunile 4.x, și cu aceasta încă un produs al noii linii cu care cei doi giganti soft "se ceartă" pentru piața atît de frumos împărțită înainte. Ca urmare TASM este compatibil cu MASM tot la fel cum și Quick-Pascal este compatibil cu Turbo-Pascal, totuși în cîteva mărunțșuri nu se potrivesc.

TASM solo ne dă, în manual, mărșul mesaj: "luați un editor ASCII și ...", deoarece în furnitura de livrare nu este conținut și un editor. Defaptnici nu este necesar, deoarece cu toate că sistemul este altfel complet, el nu-și poate dove-

di întreaga putere decît împreună cu un mediu integrat de dezvoltare Borland (IDE - Integrated Development Environment).

Asamblorul strălucește, ca și înaintășii săi, printr-un macroprocesor puternic, însă nu este actualizat cu toate facilitășile oferite de MASM 5.1, și din acest motiv nu posedă întreaga gamă de operatori pentru expresii complexe. Dar chiar și facilitășile oferite de MASM 4.0 erau atît de bogate, încît nu erau folosite la întreaga capacitate decît rar.

Și TASM sprijină întreaga gamă de procesoare pînă la 80386 și 80387. El poate lucra și "mixat", manualul descriind pe larg interfețele spre celelalte limbaje de programare din familia Turbo. OS/2 în limbaj de asamblare este o muncă grea și nici programarea Windows nu este așa de simplă.

Microsoft-C 6.0 - PDS

C 6.0 - PDS - PWB, ceea ce vine de la Professional Development System și Programmer's Workbench, este un mediu de dezvoltare C pentru DOS, OS/2 și Windows 3.0. Instalarea se face de pe șapte dischete de 1.2 MBytes și duce la ocuparea a 15 MBytes pe hard-disk.

La prima vedere mediul de lucru se aseamănă cu mediul lui Quick-C, totuși el oferă mai mult. Este oferit un editor multi-fereastră/ multi-fișier, înregistrator de macrouri, fișiere make complexe și toate acestea la o simplă apăsare pe tasta mouse-ului.

Componentul Source-Browser permite examinarea programelor complexe. Sistemul de help-uri, după principiul hipertext, este excelent și conține informațiile din cinci manuale, dar și ocupă 1.9 MBytes pe harddisk.

În ceea ce privește timpul turnaround: la o compilare și linkeditare incrementală, pe un 386, se atinge viteza de 80.000 de linii/mi-

nut. Cu toate switch-urile, pînă la optimizarea funcțiilor și la o optimizare pragmatică, programele mari au nevoie de cîteva minute. Timpul de execuție, al codului, bate ulterior însă atît timpii obținuți cu Quick-C cît și pe cei obținuți cu Turbo-C, cu o evidență dramatică. Inserînd inline cod în limbaj de asamblare tempoul poate fi sporit și mai mult.

Noul Codeview 3.0 poate depana acum și programe care constau din mai multe procese, așa-zisele programe cu DLL, și aceasta cu o editare multi-fereastră. Pe scurt MS-C 6.0 poate fi descris astfel: mediu de lucru optim, cu unele care se pot folosi foarte comod, sistem deschis (configurabil și extensibil), cu tehnologii noi de optimizare a codului, cu noi caracteristici limbaj, suport perfect pentru Windows 3.0 și OS/2 și cu un depanator de generația a treia.

MS-C 6.0 este un sistem pentru profesioniștii care trebuie să elaboreze aplicații complexe pentru DOS, Windows și OS/2.

Turbo C++, Prof. Version 1.0

Dacă Turbo C++ ar permite programarea sistemelor MS-Windows și OS/2, ar putea fi pus în aceeași linie cu MS-C 6.0. Versiunea pentru Windows va apare în curînd pe piață, pentru OS/2, Borland crede că încă nu există piață.

Altfel este respectată compatibilitatea cu standardul ANSI, de care Borland se ține cu fermitate. Cei doi de plus suplimentari urmează directivele AT&T.

Instalarea se face de pe 15 dischete de 360 KBytes, și necesită un spațiu pe harddisk de 5 MBytes.

Mediul de dezvoltare integrat oferă meniuri pull-down și ferestre (aproape) în standardul SAA. Mediul de lucru amintește de cel de la Turbo Pascal (ca să fim corecți situația este inversă deoarece mediul lui Turbo C++ a fost primul).

Tabel comparativ

Nume	Turbo-Asm.									
	Quick-C ou Assembler 2.5	Quick- Basic 4.5	Quick- Pascal 1.0	Turbo- Pascal 6.0	MS-Basic 7.1-PDS	Assembler MASM 5.1	2.0 ou Tools	MS-C 6.0- PC8	Turbo C++ Prof. V.1.0	Zortech C++ - Dev.-Ed. 2.1
Preț (DM)	360	350	350	520	1180	350	400	1380	600	360
Date tehnice										
Necesită harddisk	da	da	da	da	da	nu	nu	da	da	da
Spațiu harddisk necesitat (MBytes)	2.5	2	2	5	8, resp. 14	1.5	1.5	15	5	8.5
Suprafață SAA	da	da	da	apropiată	da	nu	apropiată	da	apropiată	apropiată
Editor multi-fereastră/multi-fisier	2 ferestre	nu	da	da	da	da	da	da	da	da
Suportă mouse	da	da	da	da	da	nu	da	da	da	nu
Utilizează EMS	nu	nu	nu	da	da	nu	da	da	da	da
Help on-line	da	da	da	da	da	da	da	da	da	aproape
Utilitare de dezvoltare										
Debugger	da	da	da	da	da	nu	da	da	da	da
Profiler	nu	nu	nu	nu	nu	nu	da	nu	da	nu
Source-Browser	nu	nu	nu	nu	da	nu	nu	da	nu	nu
Inline-Assembler	da	nu	nu	da	nu	nu	da	da	da	da
Macroassembler	nu	nu	nu	nu	nu	da	da	nu	da	nu
Overlay-Manager transparent	nu	nu	nu	nu	da	nu	nu	nu	da	da
Tehnici de programare										
OOP	nu	nu	da	da	nu	nu	nu	nu	da	da
++	nu	nu	nu	nu	nu	nu	nu	nu	da	da
Sisteme suportate										
DOS	da	da	da	da	da	da	da	da	da	da
OS/2	nu	nu	nu	nu	da	da	nu	da	nu	da
Windows 3.0	nu	nu	nu	nu	nu	da	nu	da	nu	da

În orice caz din mediul IDE pot fi compilate programe, linkeditate, lansate în execuție și depanate.

În ceea ce privește timpul turn-around: compilatorul este cu ceva mai lent decât cel al lui MS-C. Calitatea codului este foarte bună și poate ține pasul cu cea oferită de MS-C 6.0. Depanatorul este foarte bun, avînd întreaga funcționalitate a lui Codeview, dar fiind extins și pentru OOP. Lăudabil este prețul de numai 600 DM, sau chiar 250 DM (fără extensia OOP).

Turbo C++ este un sistem pentru profesioniști care poate da

clasă lui MS-C 6.0 nu numai în ceea ce privește prețul.

**Zortech C++ ,
Dev. Edition 2.1**

Zortech C++ ar putea fi aproape echivalat cu MS-C 6.0. O singură diferență: colecția de unelte este mai puțin puternică, dar în schimb Zortech C++ oferă pe cei doi de plus. Compatibilitatea se păstrează și în ceea ce privește componentele Codeview și MASM.

La prima vedere bancul de lucru pare aidoma celui oferit de Turbo C++, totuși la o privire mai atentă

se poate observa că oferă mult mai puțin decât oferă Borland. În ceea ce privește timpul turn-around: compilatorul în sine este mult mai rapid decât MS-C. Un apel la bancul de lucru și returnul, pentru un mini-program durează o secundă. Pînă cînd MS-C face același lucru pot trece pînă la 40 de secunde.

Calitatea codului este foarte bună, un motiv pentru acest lucru: aproape toate sursele din biblioteca oferită sînt scrise în limbaj de asamblare.

(Computer Persönlich 1/91,
Peter Wollsclaeger)

Operatori

aritmetici:	Basic:	+	-	*	/	\$		
	Pascal:	+	-	*	/	DIV	MOD	
	C:	+	-	*	/	%	++	--
logici:	Basic:	AND	OR	NOT				
	Pascal:	AND	OR	NOT				
	C:	&&		!				
pe biți:	Basic:	AND	OR	NOT	XOR			
	Pascal:	AND	OR	NOT	XOR	SHL	SHR	
	C:	&		~	!	§	<	>
atribuire:	Basic:	=						
	Pascal:	:=						
	C:	=	+=	-=	*=	/=	%=	&=
		§=	â=	<=	>=			

Numărul operatorilor diferiți în C este neatins

Pe drumul spre o tehnică "inteligentă" ?

REȚELE NEURONALE

Partea I

Sistemul nervos uman prelucrează mai multe miliarde de informații în modul său propriu. Acest mod de lucru al biologicului și transpunerea sa în sisteme tehnice este astăzi parte integrantă a domeniului Inteligenței Artificiale. Stadiul și posibilele influențe ale acestor cercetări vrem să vi le prezentăm în acest curs structurat în patru părți.

Cuprinsul cursului:

- **Partea 1:** Privire de ansamblu asupra temei
- **Partea 2:** Teoria propagării înapoi
- **Partea 3:** Sisteme expert și asociativitate
- **Partea 4:** Recunoașterea scrisului

"Science fiction"-ul a fost perceput de cele mai multe ori ca un amestec de ipoteze utopice și irealiste avansate de anumiți scriitori. Dar uneori aceste ipoteze nu sînt altceva decît posibilități științifice reale nerezolvate încă. Că aceste ipoteze pot rămîne încă mult timp inaccesibile este de la sine înțeles. Și totuși, ipoteza calculatoarelor "inteligente", visul sau uneori coșmarul, unor scriitori de "science fiction" este deja prezent în unele încercări. A stoca informațiile deja existente, a le analiza, a reacționa pe baza lor așa cum reacționează un organism uman, toate acestea au fost pînă acum mari probleme nerezolvate ale cercetării în neuroinformatică. În cursul nostru ne vom ocupa de stadiul actual al acestor cercetări și, mai înainte de toate, vă vom sugera cît mai multe aplicații practice ale lor.

Tînărul domeniu de cercetare al Neuroinformaticii se ocupă deci cu transpunerea modului de prelucrare a informației folosit de sistemul nervos uman în rețele neuronale artificiale folosite în tehnică. Ca și

în cercetările generale ale Inteligenței Artificiale, cu ajutorul rețelelor neuronale se încearcă dezvoltarea unor sisteme cu performanțe "inteligente" cum ar fi recunoașterea imaginilor (recunoașterea unor mostre de obiecte, dar și a unor sunete și cuvinte articulate), memoria asociativă a cunoștințelor sau conducerea mișcărilor unor roboți. Domeniile de aplicare ale acestor sisteme sînt foarte variate, de exemplu în medicină (comanda protezelor, sisteme de diagnoză, analiza radiografiilor) sau tehnica circulației (supravegherea spațiului aerian, conducerea unor autovehicule autonome).

Produsele acestor cercetări nu mai sînt programe uriașe rufînd pe supercalculatoare de tip Cray sau Suprenum, ci devin programe scrise pentru hard-uri special concepute pentru a efectua funcții inteligente. Aceste hard-uri speciale vor fi compuse din mai multe mii de procesoare legate între ele ca neuronii, și care se vor constitui într-o rețea neuronală (ca noduri și sinapse).

Una dintre funcțiile inteligente care se dorește a fi realizată este aceea de a se putea descrie orice problemă matematică nu prin program ci prin autoorganizarea rețelelor neuronale. Acest proces de autoorganizare ar trebui să aibă loc pe parcursul unui proces de învățare obținut prin cooperarea unei topologii inițiale, a unor reguli de învățare și a unui număr mare de antrenamente. Acest proces de învățare este, fără îndoială, principalul scop al cercetării în Neuroinformatică.

Alături de aceste rețele autoorganizabile pomenite pînă acum, rețele sprijinite de lucrările unor specialiști ca T. Kahonen, S. Grossberg, Malsburs și alții, a apărut și o altă clasă - așa numitele rețele supervizoare.

Chiar dacă existența lor nu este motivată biologic așa cum sînt ce-

lealte prin neuroni, sinapse, faze de învățare, rețelele supervizoare sînt totuși foarte asemănătoare cu primele. Aceste rețele se bazează în principal pe ideea de a face cîteva încercări care să poată fi generalizate după aceea.

O probă de antrenament constă dintr-o mostră oarecare (de exemplu o combinație de pixeli ai literei "A") și din scopul dorit al identificării (în exemplul nostru, obținerea codului ASCII corespunzător literei "A"). Generalizarea ar consta în aceea că nu toate combinațiile de pixeli posibile pentru litera "A" trebuie memorate. De fapt va fi nevoie doar de cîteva (bine înțeles foarte diferite și bine alese) probe de antrenament, cu ajutorul cărora rețeaua să învețe o "imagine universală" a literei "A". Pe parcursul expunerii vom mai oferi și alte exemple.

Rădăcinile comune ale celor două modele de rețele se găsesc mult înapoi în timp; în anul 1943 McCulloch și Pitts au dezvoltat un formalism matematic pentru descrierea rețelelor neuronale, și au demonstrat că pentru orice problemă de rezolvat se poate dezvolta o rețea neuronală care să o rezolve. Lucrările lor au constituit bazele pentru teoria automatelor finite, domeniu atît de important astăzi. Din păcate nu s-a dat nici o indicație asupra metodelor de a rezolva problema în mod automat.

Un prim algoritm de învățare (motivată biologic) a fost dat în 1949 de Dr. Hebb: puterea legăturii dintre doi neuroni crește atunci cînd amîndoi sînt activi în același timp. Această formulare nu poate fi înțeleasă izolat - nu precizează ce înseamnă noțiunile de neuron, putere de legătură, activitate și simultaneitate - dar dă totuși un nucleu care s-a păstrat într-o formă ușor modificată pînă în abordările actuale ale teoriei rețelelor neuronale.

În anii '60 Widrow și Hoff au dezvoltat modelul neuronal adaptiv

Adaline (numit mai târziu Madaline) în timp ce F. Rosenblatt dezvolta modelul "Perceptron". Cei doi au demonstrat noile proprietăți ale rețelelor: ele se puteau autoorganiza parțial, erau capabile de autoînvățare, puteau clasifica mostre simple și chiar puteau generaliza experiențe la modul prezentat mai sus. Se puteau deci recunoaște probe pe care rețeaua nu fusese antrenată anterior.

Principalele slăbiciuni ale acestui tip de rețele de supervizare au fost evidențiate în 1969 de către M. Minski și S. Papert. Abia în anii '80 au fost redescoperite aceste rețele și scoase la lumină de către Rumelhardt și Hinton care au organizat rețelele în mai multe straturi și au pus la punct un algoritm de învățare ingenios numit algoritmul de propagare înapoi.

Acest algoritm a adus o schimbare majoră, cu ajutorul lui rețelele reușind să depășească toate obiecțiile lui Minski. Totuși timpii de convergență ai algoritmilor de învățare (deci timpul de învățare) rămân destul de mari.

Cum funcționează acești algoritmi de propagare înapoi vom exemplifica mai târziu prin numeroase exemple.

Această primă parte a cursului a încercat o succintă introducere la dezvoltarea și diversele stadii prin care a trecut cercetarea în acest domeniu. articolul al doilea se va ocupa de algoritmul de propagare înapoi și de topologia de rețea aferentă. Vom realiza aceasta prin exemple și nu printr-o prezentare teoretică aridă.

În partea a treia ne vom referi la o aplicație anume a rețelelor neuronale: sistemele expert. Spre deosebire de abordările clasice ale sistemelor expert (de tip Prolog), de data aceasta ne vom referi la regulile de producție de tipul "dacă A este adevărat atunci rezultă B". Vom vedea că noua abordare este mult mai flexibilă și mult mai tolerantă la greșeli. Această particularitate a rețelelor neuronale, de a asocia la seturi vagi de date de intrare, date de ieșire ferme, o vom folosi la stocarea informațiilor într-o bază de date.

În ultima parte a seriei vom vorbi despre necesitatea combinării

rețelelor neuronale cu alte tehnici mai moderne. De cele mai multe ori problemele practice sînt greu de rezolvat folosind în exclusivitate rețele neuronale. Problemele trebuie descompuse, în general, în probleme mai simple pe care să le prelucrăm separat cu ajutorul rețelelor neuronale sau cu alte tehnici.

Un exemplu tipic pentru aceasta este recunoașterea scrisului, transpunerea unei coli de hîrtie plină de semne în codurile ASCII corespunzătoare pe care le poate înțelege calculatorul. Pentru început contrastul alb-negru de pe hîrtie este digitizat cu ajutorul unui scanner. În imaginea rezultată trebuie diferențiat mai întîi textul de restul graficii, apoi în interiorul textului trebuie delimitată fiecare literă. Abia după aceea se poate începe identificarea propriu-zisă a semnelor cu codurile ASCII corespunzătoare.

Dar despre toate acestea în numerele viitoare ale revistei.

(PC + Technik 2/91, Peter Schmitz)

Formate grafice

(continuare din pagina 15)

EPS, PS

PS, respectiv EPS, reprezintă prescurtarea pentru "Postscript" sau "Encapsulated Postscript". Acesta este un format foarte important și foarte răspîndit în domeniul Desktop Publishing. Formatul EPS nu este nici el formatul de bază al unui anumit program, ci este un format în care pot fi create fișiere din diferite programe de grafică, cum ar fi: "Arts & Letters", "Corel Draw", "Adobe Illustrator" (pe Mac, dar în curînd și pe PC), și de "Aldus Freehand" (pe Mac). Aceste fișiere conțin informații despre grafice sau texte în formatul limbajului de descriere a paginilor "Postscript". Aceste informații sînt folosite pen-

tru imprimarea pe imprimante laser sau pe exponatoare Linotronic.

Deoarece Postscript este un limbaj puternic de descriere, care cunoaște practic toate elementele de grafică și de text și toate modalitățile de prezentare, el poate importa aceste fișiere din toate programele importante DTP. Astfel obiectele grafice importate apar în dimensiunea lor reală și în poziția dorită și pot fi tipărite pe imprimante compatibile Postscript foarte corect și cu rezoluția maximă.

În programele DTP - cu excepția măririlor/micșorărilor - nu se mai pot face alte manipulări (de aici și denumirea "Encapsulated"). Pentru a putea vedea pe ecran, din interiorul unui program DTP, conținutul unui astfel de fișier, fișierele EPS pot conține pe lîngă informațiile referitoare la vectori și informații suplimentare, cum ar fi o imagine rudimentară a unei grafici

bitmap. Programele DTP, cum ar fi Pagemaker sau Ventura Publisher, pot vizualiza pe ecran o astfel de imagine. Dacă așa-zisul header (antet) TIFF lipsește, pe ecran va apare doar o casetă gri.

HPGL

HPGL este prescurtarea pentru "Hewlett-Packard Graphics Language" și este un format asemănător cu Postscript, care conține instrucțiuni de comandă pentru o imprimantă sau un plotter, cu deosebirea că nu se folosește limbajul de descriere Postscript, ci un limbaj de descriere total diferit, limbajul special al imprimantelor și plotter-elor fabricate de firma Hewlett-Packard.

(Computer Live 2/91, Klaus Schrdl)

Obiectele se încapsulează

Decă doriți să programați mai comod și să lucrați mai ușor cu proiecte mari, atunci nu puteți face abstracție de OOP (programarea orientată obiect).

Până acum în programare se făcea o distincție netă între structurile de date și funcțiile care lucrează cu ele. Tehnicile de programare bazate pe acest principiu au fost verificate în practică. Odată cu sporirea complexității problemelor care se cereau rezolvate, s-a vădit însă că aceste metode trebuie modificate, dezvoltate în continuare, sau trebuie găsite metode cu totul noi. Ideea de a combina datele și metodele lor de prelucrare nu este nouă, dar abia odată cu introducerea unui limbaj de programare orientat obiect (C++), pe un microcalculator, a dat o semnificație sporită tehnicii OOP. Filozofia programării orientate obiect se abate de la metodele obișnuite, în plus ea introduce o serie întreagă de termeni noi care ar putea îngreuna înțelegerea.

Merită deci să ne ocupăm mai îndeaproape de acest domeniu. Însușirile obiectelor facilitează o elaborare economică a programelor și ușurează,

Planificarea cursurilor

Cursul 1: Noțiuni și termeni; definiții și însușiri ale obiectelor; încapsulare și moștenire

Cursul 2: Declarații dinamice; polimorfism

Cursul 3: Constructor și Destructor; metode virtuale

Cursul 4: Ierarhii de obiecte; diferite tehnici OOP

Cursul 5: Exemple de OOP.

(mai ales), întreținerea bibliotecilor de programe. Utilizând cu consecvență tehnicile OOP, multe programe devin mai transparente și pot fi cuprinse mai ușor dintr-o privire.

Un tip obiect este un nou tip de date. Definiția sa este dată de utilizator, deci un obiect este un "tip de date definit de utilizator". Odată definit un obiect, cu el se poate opera la fel ca și cu toate celelalte tipuri de date recunoscute de limbajul de programare în care se lucrează. Declararea unui obiect se face ca și declararea unei variabile normale. Compilatorul îi rezervă o zonă de memorie care poate fi ulterior accesată prin intermediul numelui obiectului.

Cum arată însă un obiect concret? În Pascal el este definit și declarat în mod asemănător cu un articol (record). La început trebuie stabilit deci un tip obiect la care să se poată raporta ulterior un obiect.

Listingul OOP_prima_incercare vă arată cum funcționează așa ceva în Turbo-Pascal. În partea de defini-

re a tipului este definit obiectul cu numele Caracter. Deosebirea vizibilă față de o declarație de articol o constituie utilizarea cuvîntului rezervat OBJECT. Toate variabilele sînt cuprinse sub un identificator, ca și în cazul unui articol. Un obiect este însă mai mult: sub același acoperiș sînt unite nu numai datele ci și metodele lor de prelucrare. Tipul de obiect Caracter permite patru metode de prelucrare: "stabilește_pozitie", "citește_pozitie", "scrie" și "citește". Toate aceste metode influențează sau utilizează variabilele interne "linia" și "coloana". La stabilirea unui tip obiect se spune doar ce face o metodă, este numită și i se specifică parametrii. Metodele de prelucrare propriu-zise sînt declarate înafara declarației tip obiect - în listingul OOP_prima_incercare se poate vedea și cum. Acest lucru se întîmplă în același mod ca și în cazul declarațiilor obișnuite de funcții și proceduri. Numele este format din numele tipului obiect și din numele metodei, separate printr-un punct. Puteți lucra deci așa cum sînteți obișnuți de la lucrul cu articole. Puteți utiliza chiar și ins-

Tip obiect: Un tip obiect este un tip de date definit de utilizator. În el sînt definite structurile de date și metodele de prelucrare corespunzătoare.

Obiect: Un obiect este stabilit și manipulat ca și o variabilă. Compilatorul rezervă deci pentru un obiect spațiul de memorie necesar, care este accesibil prin intermediul numelui obiectului.

Metodă: Pentru a putea lucra cu datele stabilite în interiorul unui obiect, în cursul stabilirii lui trebuie declarate și funcțiile și procedurile corespunzătoare. Tocmai acestea însă sînt metodele.

Sarcină: După stabilirea unui obiect, numele unei metode este utilizat pentru a-i transmite o sarcină acestuia.

Protocol: Totalitatea sarcinilor accesibile unui tip obiect este desemnată ca protocol.

Încapsulare: Cuprinderea împreună a structurilor de date și a metodelor de prelucrare se numește încapsulare. Definirea obiectelor în cadrul unui Unit întărește încapsularea.

Moștenire: Tipurile obiect își pot transmite toate însușirile unor noi obiecte. Această modalitate de moștenire elimină necesitatea de a redefini metodele deja definite. Trebuie stabilite doar ceea ce s-a schimbat față de obiectul "mamă" sau care sînt datele și metodele care trebuie adăugate.

Ierarhie de obiecte: Deoarece tipurile obiect își pot moșteni însușirile, se poate construi o ierarhie arborescentă a tuturor urmașilor. Toate tipurile obiect ale acestei ierarhii sînt legate prin "legături de rudenie". De datele și metodele obiectului rădăcină pot dispune toți urmașii.

```

program OOP_prima_incercare;

const Seg_BWS = $B800;
{Adresa segmentului pentru prezenta-
rea culorilor}

{Stabilirea unui obiect}
type
  Caracter = OBJECT
    linia,coloana : byte;
    procedure stabileste_pozitie
      (x,y : byte);
    procedure citeste_pozitie
      (var x,y : byte);
    procedure citeste (var ch : char);
    procedure scrie (ch : char);
  end;

{Stabilirea metodelor pentru obiectul
Caracter}

procedure Caracter.stabileste_pozitie
      (x,y : byte);
begin
  coloana := x;
  linia := y;
end;

procedure Caracter.citeste_pozitie
      (var x,y : byte);
begin
  x := coloana;
  y := linia;
end;

end;

procedure Caracter.citeste
      (var ch : char);
var Ofs_BWS : word;
begin
  if ((linia>25) or (coloana>80)) then
    exit;
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  ch := chr (mem[Seg_BWS:Ofs_BWS]);
end;

procedure Caracter.scrie (ch : char);
var Ofs_BWS : word;
begin
  if ((linia>25) or (coloana>80))
    then exit;
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  mem[Seg_BWS:Ofs_BWS] := ord (ch);
end;

{Declaratia unui obiect}
var
  z : Caracter;

{Programul principal}
begin
  z.stabileste_pozitie (10,10);
  z.scrie ('A');
end.

{LISTING1.PAS sfirsit}

```

<pre> type Sir_de_caractere = OBJECT (Caracter) procedure citeste_text (var st : string; lng : byte); procedure scrie_text (st : string); end; {Listing 2} Prin mostenire obiectul Sir_de_caractere pastreaza toate insu- sirile obiectului Caracter. O redefinire nu este necesara. Listingul 3 are deci aceeasi semnifi- catie ca si listingul 2. </pre>	<pre> type Sir_de_caractere = OBJECT linia, coloana : byte; procedure stabileste_pozitie (x,y : byte); procedure citeste_pozitie (var x,y : byte); procedure scrie (ch : char); procedure citeste (var ch : char); procedure scrie_text (st : string); procedure citeste_text (var st : string; lng : byte); end; {LISTIN23.PAS sfirsit} </pre>
--	--

truțiunea "with" pentru a avea mai puțin de scris.

Surprinzător este faptul că varia-
bilele interne "linia" și "coloana" pot
fi modificate numai prin intermediul
metodelor "stabileste_pozitie" și "ci-

teste_pozitie". Utilizarea acestor va-
riabile înafara metodelor obiectului
nu este posibilă. Ceea ce la prima
vedere pare incomod, căci s-ar fi

putut și mai direct, constituie de fapt o însușire de bază a programării orientate obiect. Înămănuncherea structurilor interne de date cu metodele corespunzătoare de prelucrare se cheamă încapsulare.

Această noțiune sugerează faptul că pentru utilizator structura obiectului tip este neesențială și inaccesibilă. Pentru utilizator obiectul tip există doar sub forma tuturor metodelor de acces la el. Cum funcții-

nează propriu-zis o metodă nu este interesant, fiind suficient să știm ce se poate face cu metodele. În acest mod structura internă a datelor poate fi protejată împotriva unor accese nedorite.

```

program OOP_a_doua_incercare;
    mem[Seg_BWS:Ofs_BWS] := ord (ch);
end;

const Seg_BWS = $B800;
{Adresa segment monitor color}

{Doua tipuri de obiecte}
type
    Caracter = OBJECT
        linia,coloana : byte;
        procedure stabileste_pozitie
            (x,y : byte);
        procedure citeste_pozitie
            (var x,y : byte);
        procedure citeste (var ch : char);
        procedure scrie (ch : char);
    end;

    Sir_de_caractere = OBJECT (Caracter)
        procedure citeste_text
            (var st : string;
             lng : byte);
        procedure scrie_text (st : string);
    end;

{Metodele tipului de obiecte Caracter}
procedure Caracter.stabileste_pozitie
    (x,y : byte);
begin
    coloana := y;
    linia := x;
end;

procedure Caracter.citeste_pozitie
    (var x,y : byte);
begin
    x := coloana;
    y := linia;
end;

procedure Caracter.citeste
    (var ch : char);
var Ofs_BWS : word;
begin
    Ofs_BWS:=2*(coloana-1)+160*(linia-1);
    ch := chr (mem[Seg_BWS:Ofs_BWS]);
end;

procedure Caracter.scrie (ch : char);
var Ofs_BWS : word;
begin
    Ofs_BWS:=2*(coloana-1)+160*(linia-1);
    mem[Seg_BWS:Ofs_BWS] := ord (ch);
end;

{Metodele tipului de obiecte
Sir_de_caractere}
procedure Sir_de_caractere.scrie_text
    (st : string);
var i,j : integer;
begin
    j := coloana;
    for i := 1 to length (st) do begin
        scrie (st[i]);
        inc (coloana);
    end;
    coloana := j;
end;

procedure
    Sir_de_caractere.citeste_text
    (var st : string; lng : byte);
var i : integer;
    ch: char;
begin
    st := '';
    for i := 1 to lng do begin
        citeste (ch);
        st := st + ch;
        inc (coloana);
    end;
    coloana := coloana - lng;
end;

{Declararea obiectelor si
variabilelor necesare}
var z : Caracter;
    t : Sir_de_caractere;
    st: string;

{Programul principal}
begin
    z.stabileste_pozitie (10,10);
    z.scrie ('A');
    t.stabileste_pozitie (12,50);
    t.scrie_text ('Mostenim insusiri');
    t.citeste_text (st,12);
    t.stabileste_pozitie (14,50);
    t.scrie_text (st);
end.

{LISTING4.PAS sfirsit}

```

```

unit t_handle;

Interface

{Adresa segment monitor color}
const Seg_BWS = $B800;

{Doua tipuri de obiecte}
type
  Caracter = OBJECT
    linia,coloana : byte;
    procedure stabileste_pozitie
      (x,y : byte);
    procedure citeste_pozitie
      (var x,y : byte);
    procedure citeste (var ch : char);
    procedure scrie (ch : char);
  end;

  Sir_de_caractere = OBJECT (Caracter)
    procedure citeste_text
      (var st : string;
       lng : byte);
    procedure scrie_text (st : string);
  end;

Implementation

{Metodele tipului de obiecte Caracter}
procedure Caracter.stabileste_pozitie
  (x,y : byte);
begin
  coloana := y;
  linia := x;
end;

procedure Caracter.citeste_pozitie
  (var x,y : byte);
begin
  x := coloana;
  y := linia;
end;

procedure Caracter.citeste
  (var ch : char);
begin
  var ofs_BWS : word;
  ofs_BWS:=2*(coloana-1)+160*(linia-1);
  ch := chr (mem[Seg_BWS:ofs_BWS]);
end;

procedure Caracter.scrie (ch : char);
var ofs_BWS : word;
begin
  ofs_BWS:=2*(coloana-1)+160*(linia-1);
  mem[Seg_BWS:ofs_BWS] := ord (ch);
end;

{Metodele tipului de obiecte
Sir_de_caractere}
procedure Sir_de_caractere.scrie_text
  (st : string);
var i,j : integer;
begin
  j := coloana;
  for i := 1 to length (st) do begin
    scrie (st[i]);
    inc (coloana);
  end;
  coloana := j;
end;

procedure
  Sir_de_caractere.citeste_text
  (var st : string; lng : byte);
var i : integer;
ch: char;
begin
  st := '';
  for i := 1 to lng do begin
    citeste (ch);
    st := st + ch;
    inc (coloana);
  end;
  coloana := coloana - lng;
end;

end.

{LISTING5.PAS Sffirsit}

```

Unui obiect i ne adresăm dîndu-i o sarcină, tot restul lăsînd în seama lui. Listingul nostru OOP_prima_incercare dă lămuriri și în această privință. Obiectului Z i-am dat sarcina "stabileste_pozitie(10,10)", realizarea lăsînd-o în seama lui. Următoarea instrucțiune transmite sarcina "scrie('A')". Numele metodei împreună cu eventualii parametrii nu sînt deci altceva decît sarcini pe care obiectul le poate înțelege. To-

talitatea sarcinilor care pot fi executate de un obiect se numește și protocol.

Dar programarea orientată obiect nu are doar însușirea încapsulării. Ar fi și prea puțin.

Dacă, de ex., doriți să utilizați obiectul Caracter pentru a opera cu un șir, mînuirea lui devine incomodă. Dacă însă nu doriți să renunțați la însușirile obiectului

Caracter, pentru că de ex. funcționează excelent și fără greșală, atunci le puteți lăsa moștenire unui alt obiect, căruia puteți să-i mai adăugați și alte însușiri. Din Listingul 2 se poate vedea cum se moștenește. Noul tip obiect

"Sir_de_caractere" dispune de toate metodele "mamei" sale. De unde se trage obiectul Sir_de_caractere se poate vedea din instrucțiunea de definiție:

```

program OOP_a_treia_incercare;

uses t_handle;

var z : Caracter;
    t : Sir_de_caractere;
    st: string;

begin
  z.stabileste_pozitie (10,10);
  z.scrie ('A');
  t.stabileste_pozitie (12,50);
  t.scrie_text ('Mostenim insusiri');
  t.citeste_text (st,12);
  t.stabileste_pozitie (14,50);
  t.scrie_text (st);
end.
{LISTING6.PAS Sfirsit}

```

Sir_de_caractere =
OBJECT (Caracter)

Totodată se mai definesc două noi însușiri, și anume "citeste_text" și "scrie_text".

Moștenirea are câteva avantaje. O nouă definire a metodelor moștenite nu mai este necesară. Obiectul Sir_de_caractere din Listingul 2 este ca urmare identic cu cel din Listingul 3. Metodele deja moștenite pot fi modificate și dezvoltate în continuare, totuși aceasta este deja o anticipare. În acest loc ne mulțumim să vă prezentăm un exemplu care vă va arăta cum funcționează mecanismul de moștenire.

Să facem o a doua încercare cu Listingul 4. La început, în partea de definire, se definesc două tipuri obiect. Tipul obiect Caracter este deja cunoscut din Listingul 1. Sir_de_caractere provine din tipul Caracter, în mod clar vizibil după "mama" cuprinsă între paranteze. La sfârșit urmează stabilirea tuturor metodelor. Programul principal demonstrează în mod foarte clar faptul că obiectul Sir_de_caractere a moștenit însușirile obiectului Caracter. Cu toate că nu a fost stabilită explicit, metoda "stabilește_pozitie" poate fi transmisă și obiectului Sir_de_caractere.

Pînă în acest moment v-am prezentat regulile jocului în cazul pro-

```

type
  <Nume> = OBJECT
    <Lista datelor>
    <Lista metodelor de prelucrare>
  end;

procedure <Metoda>
  [( <Lista de parametrii > )]

procedure <Tip obiect>.<Metoda>
  (<Lista de parametrii >)

type
  <Nume> = OBJECT (<Nume obiect mama>)
  {OOP112.PAS Sfirsit}

```

gramării orientate obiect, în orice caz partea care ne-am propus-o.

Tipurile obiect sînt stabilite ca și articolele în partea de definire, dar aici sînt unite sub același acoperiș atît datele cît și metodele lor de prelucrare. Această caracteristică se numește încapsulare. (Fig. OOP112.PAS, liniile 1-5).

Lista metodelor de prelucrare descrie doar, ceea ce fac metodele individuale, deci numele și eventualii parametri (Fig. OOP112.PAS, linia 7).

Numele unui obiect este de fapt sarcina cu care poate fi activat obiectul referit. Metodele propriu-zise sînt stabilite înafara tipului obiect, totuși sînt legate de acesta. Numele tipului obiect precede definirea numelui metodei. (Fig. OOP112.PAS, linia 9). Tipurile obiect pot lăsa moștenire anumite însușiri unor obiecte noi. Noului tip obiect trebuie să i se specifice proveniența (Fig. OOP112.PAS, linia 11-12).

În noul obiect tip pot fi stabilite noi date și metode, acesta posedînd astfel atît însușirile vechi cît și pe cele noi.

Să revenim încă odată asupra faptului că obiectele, odată stabilite, pot fi utilizate la fel cu celelalte variabile. Desigur aceste stabiliri pot fi făcute și în interiorul unei unități (Units). Să mai încercăm odată programul cunoscut, folosind de această dată Listingul 5. Nu trebuie

decît să declarăm sursa ca și Unit și să stabilim segmentele Interface și Implementation. Se poate observa că în partea de Interface nu sînt prezente decît definițiile celor două obiecte tip. Stabilirea tuturor metodelor urmează în partea Implementation. Rezultatul unei astfel de construcții este o întărire a încapsulării, deoarece în acest caz și metodele vor fi ascunse. Pe această cale utilizatorul obiectelor află doar ce poate face cu metodele, modul lor de funcționare rămînînd necunoscut.

Programul din Listingul 6 este identic cu cel din Listingul 4. Definițiile obiectelor au fost ascunse într-un Unit în Listingul 5.

Pînă acum am stabilit obiectele doar static. Bineînțeles, în Turbo-Pascal, obiectele pot fi manipulate și dinamic. Sintaxa este aceeași atît în cazul declarațiilor statice cît și a celor dinamice, trebuie doar avută grijă ca să se respecte regulile jocului în programarea dinamică, cum ar fi rezervarea de memorie, eliberarea memoriei, să nu se piardă din vedere simbolurile speciale ale limbajului.

Dar despre acest lucru vom vorbi, printre altele, în episodul următor. Acolo vă vom prezenta modul în care obiectele pot fi tratate dinamic.

(PC+Technik 2/91,
Andreas Schimpf)

Sisteme de achiziție de date și control cu calculatoare personale compatibile IBM PC/XT/AT

Majoritatea dintre noi, am fi de acord cu faptul că PC-urile au avut o mare influență asupra tehnicilor de măsurare, control, analiză și prezentare. Din păcate însă, PC-ul nu poate fi considerat un "RĂSPUNS" la toate întrebările noastre, ci mai degrabă o "SCULĂ". În ideea de a folosi cât mai eficient această sculă (tehnologie), trebuie mai întâi să-i cunoaștem caracteristicile, capacitățile, puterea, dar și LIMITELE sale.

Poate una dintre cele mai importante limitări se referă la TIPUL SEMNALELOR pe care PC-ul le poate accepta și prelucra. Aceasta, deoarece majoritatea semnalelor din lumea reală NU pot fi conectate direct la PC. Echivalenții electrici ai temperaturii, presiunii, vitezei, debitului și a stărilor ÎNCHIS/DESCHIS, nu sînt direct compatibile cu cerințele PC-urilor. Cum intră și ies atunci semnalele în și din PC?

Acest articol prezintă informațiile și recomandările de bază necesare privind achiziția și controlul datelor, cu calculatoare personale compatibile PC/XT/AT.

Achiziții de date și control: o privire de ansamblu

Termenul de ACHIZIȚIE DE DATE ȘI CONTROL - AD&C (data acquisition and control - DA&C), poate fi înțeles diferit de fiecare dintre noi. Supravegherea unui voltmetru și înregistrarea manuală a valorii citite, reprezintă o ACHIZIȚIE DE DATE. Pe de altă parte, dacă aprindem lumina într-o încăpere, înseamnă că am efectuat o operație de CONTROL.

În concluzie, achiziția de date, este procesul de colectare a informațiilor care descriu o anumită situație. Datele culese reflectă ce s-a întimplat în momentul îndeplinirii unei anumite condiții, care poate fi definită cu ajutorul unui EVENIMENT.

Sistemele în timp real se caracterizează prin abilitatea lor de a executa un anumit TASK de AD&C într-o anumită FEREASTRĂ DE TIMP. TIMPUL DE RĂSPUNS al sistemului depinde de viteza și precizia cerută de aplicația respectivă. Pentru orice aplicație de AD&C, oricît de lentă ar fi, există o aplicație suficient de lentă, care pentru acea aplicație reprezintă un sistem în timp real.

CONTROLUL implică generarea unui semnal de ieșire ca răspuns la un semnal de intrare. Controlul poate fi de tip : BUCLĂ - DESCHISĂ (OPEN - LOOP) sau BUCLĂ ÎNCHISĂ (CLOSED LOOP).

Oprirea încălzirii la ora 4:00 p.m. este un exemplu de control în buclă deschisă, în timp ce oprirea

încălzirii din cauză că este prea cald este un control în buclă închisă.

În general, datele se culeg de ingineri, tehnicieni, fizicieni, chimiști sau alți oameni implicați în cercetare, test, dezvoltare, producție, calitate, control, management etc., pentru a produce analize, rapoarte, a lua decizii și a executa operații de control. Datele sînt înregistrate, analizate, utilizate și comunicate. Metodele de lucru, însă, se pot schimba. Acum ne concentrăm asupra faptului de a folosi mașinile pentru a rezolva multe din problemele noastre de achiziție de date și control.

Motivul principal este CREȘTEREA PRODUCTIVITĂȚII. Viteza, precizia, fiabilitatea și costul sînt doar factori colaterali. Astfel, sistemele de astăzi se bazează pe cea mai productivă mașină - calculatorul. Figura 1 ne prezintă componentele unui sistem de AD&C. Calculatorul realizează nu numai analiza și luarea deciziilor ci și controlul funcțiilor active de preprocesare și conversie a semnalelor.

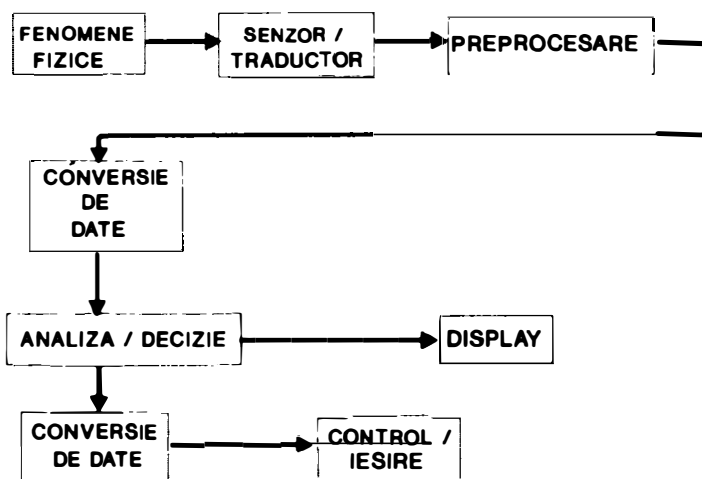


Fig. 1. Schema unui sistem de achiziție de date și control

Calculatoarele moderne oferă viteză ridicată, flexibilitate, adaptabilitate, fiabilitate și cantitate mare de memorie, proprietăți ce permit realizarea unor funcții ca: analiză, stocare, afișare, generare de rapoarte, control și comunicație. Oricum, însă, cele mai multe semnale din lumea înconjurătoare (temperatură, presiune, debit, viteză, intensitate, poziție etc.) nu pot fi citite direct de calculatoarele numerice. Acești parametri sînt reprezentați de semnale analogice care se caracterizează prin nivelul lor CONTINUU.

Calculatorul poate recunoaște numai nivele DIGITALE (închis sau deschis). De aceea este nevoie de dispozitive ce realizează o conversie a acestor semnale.

LEGĂTURA cu lumea reală - Produsele de AD&C convertesc semnalele din lumea reală într-un format acceptat de calculatoarele numerice. Sistemele de AD&C pot regenera semnale analogice precum și alte semnale pe baza instrucțiunilor de program realizînd astfel, puntea de legătură între calculator și lumea reală.

PC-urile, în comparație cu alte tipuri de calculatoare, s-au dezvoltat cel mai rapid spre a deveni ce-

lula de bază a sistemelor de AD&C. Ele au cucerit deja diferite domenii în acest sens cum ar fi: culegerea de date și automatizări în laborator, culegerea de date de la aparate medicale și monitorizarea a echipamentelor (Automatic Test Equipment - ATE, care include inspecția, testul de durabilitate, testul de producție și testarea finală), monitorizarea și controlul proceselor industriale.

Diferite tipuri de sisteme și modurile de conectare a lor la un PC

O parte importantă a oricărui sistem de AD&C este calculatorul GAZDĂ (HOST computer). Există două posibilități de interfațare HARD a sistemelor de AD&C cu calculatorul: conectare directă pe magistrala PC-ului (produse pentru BUS-ul INTERN) și conectarea prin intermediul unui canal de comunicație standard ca de exemplu RS-232, RS-422 sau IEEE-488 (produse pentru BUS-ul EXTERN). Fiecare dintre cele două are avantaje și dezavantaje.

PRODUSE PENTRU BUS EXTERN - Există numeroase avantaje asociate acestor produse cum ar fi:

VIRTUAL, putem configura orice sistem, oricît de mare dorim; sistemul de AD&C poate fi plasat la distanță de calculatorul gazdă (și astfel cît mai aproape de traductorii); sistemul de AD&C poate degreva calculatorul gazdă de unele TASK-uri de culegere-date; sistemul de AD&C poate fi interfațat virtual cu orice tip de calculator.

Figura 2 ne arată o diagramă bloc simplificată a unui sistem de AD&C bazat pe BUS extern. Comunicația prin RS-232, RS-422 sau IEEE-488, necesită ca sistemul de AD&C să aibă microprocesorul lui propriu. Acest microcalculator local, facilitează de asemenea și operațiile de comunicație la distanță, reducînd astfel sarcinile PC-ului gazdă.

Sistemele de acest tip sînt concepute ca sisteme independente, avînd incintele lor proprii în care sînt montate, și care conțin nu numai microsistemul ci și sursa de alimentare proprie precum și hard-ul necesar pentru intrările și ieșirile analogice și numerice. De obicei, funcțiile de intrare/ieșire (Input/Output I/O) sînt grupate pe plăci separate, ceea ce permite o configurare mai rapidă a sistemelor. Sistemele de AD&C cu BUS extern, permit construcția sisteme-

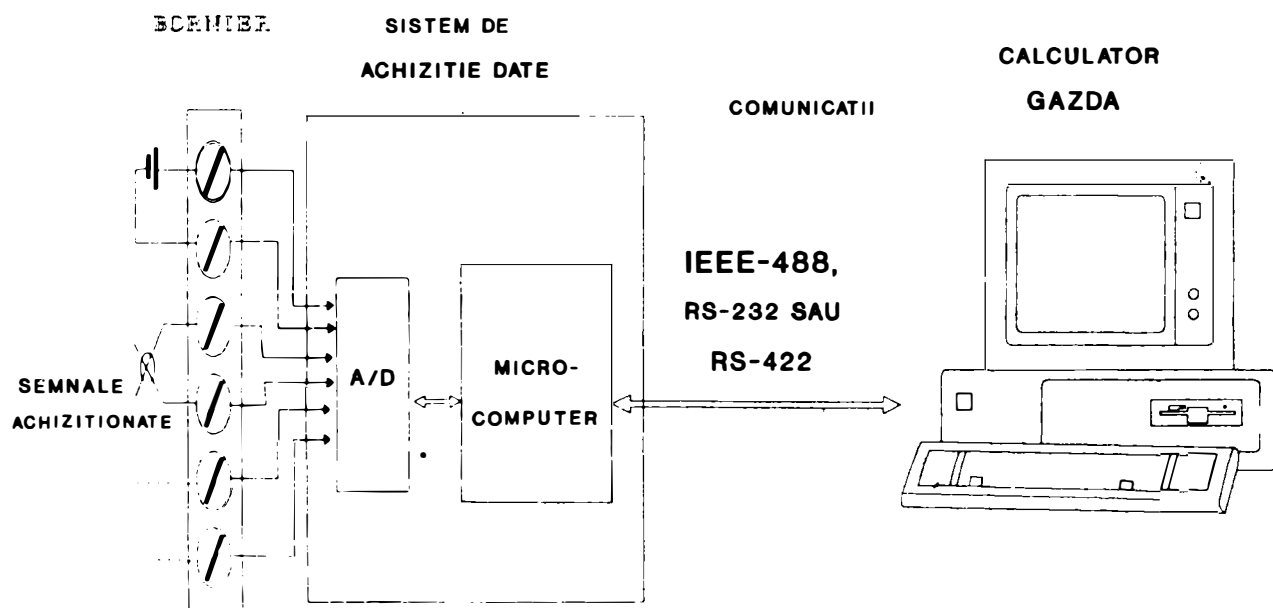


Fig. 2. Diagramă bloc a unui sistem AD&C cu bus extern

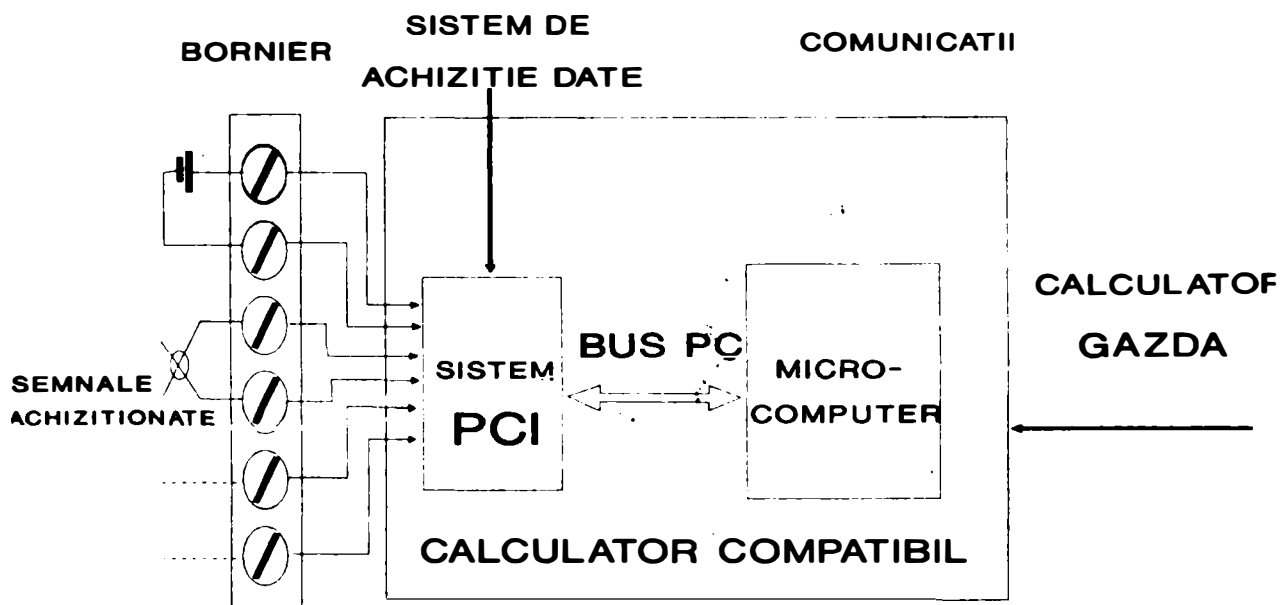


Fig. 3. Sistem AD&C cu bus intern

lor distribuite care pot realiza monitorizarea și controlul unui mare număr de parametri aflați la distanță mare unul de celălalt și la mare distanță de PC-ul gazdă.

PRODUSE PENTRU BUS INTERN - Principalele avantaje ale conectării direct pe BUS-ul intern al PC-ului includ: viteză mare, cost scăzut, dimensiuni mai mici. Aceste tipuri de sisteme de AD&C reduc costul pentru că nu mai au nevoie de incinte și surse de alimentare separate, alimentarea făcându-se din sursa internă a PC-ului. Vitezele mari se obțin datorită eliminării protocolului de comunicație, relativ lent, folosit. De exemplu, rata de achiziție a unui sistem de AD&C, folosind interfața de comunicație RS-232 la viteza de 9600 Baud, este limitată la aproximativ 20 de citiri analogice/secundă. Pe de altă parte, unele produse de AD&C pentru BUS intern, pot achiziționa și la rate de 1 milion de eșantioane/secundă. Figura 3 ne prezintă o diagramă bloc simplificată a unui sistem de AD&C bazat pe BUS intern.

Toate sistemele de AD&C pentru BUS intern sînt compuse din plăci de interfață ce se conectează direct pe SLOT-urile din PC. Unele sînt proiectate cu un număr FIX de

intrări/ieșiri, analogice/numerice, ceea ce prezintă dezavantajul că ele nu permit adăugarea ulterioară a unor funcții noi sau a unor canale noi de monitorizat.

În contrast, sistemele MODULARE permit utilizatorului să selecteze numărul de funcții de I/E dorite. Această posibilitate este concepută ca o familie de module de funcții care prezintă o flexibilitate mai mare, dînd posibilitatea definirii oricărei aplicații specifice.

PC-ul și sistemele de achiziții de date și control

Din punct de vedere istoric, industrial și științific, aplicațiile de tip AD&C au fost implementate mai întîi, cu ajutorul calculatoarelor mari, (MAIN-FRAME), sau cu ajutorul minicalcutoarelor. Acestea erau mașini de 16 biți care dispuneau de sisteme de operare multitasking și multiuser. Complexitatea și costul

acestora dicta ca aceste sisteme să fie configurate ca sisteme centrale, cu mai mulți utilizatori. Lucrările mici nu justificau însă această investiție mare.

Apariția calculatoarelor personale - PC, a făcut teoretic posibil, ca oricine să beneficieze de flexibilitate, putere de calcul și eficiența sistemelor de AD&C. PC-urile oferă performanțe ridicate și cost scăzut pe lîngă o ușurință în utilizare fără precedent. Datorită unui nivel ridicat de standardizare între constructorii de PC-uri și cei de sisteme de AD&C există astăzi posibilitatea ca un om de știință sau inginer să-și poată implementa sistemul său propriu de AD&C într-un timp foarte scurt. Figura 4 ne arată relația dintre sistemul de AD&C și calculatorul gazdă.

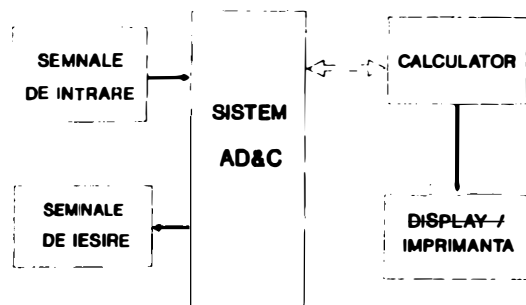


Fig. 4. Un sistem AD&C conectat la un calculator

Principii de conversie a datelor

După cum am văzut, majoritatea semnalelor din lumea reală (de exemplu amplitudine, nivel) nu au formatul care poate fi direct acceptat de calculator. Funcția de conversie este îndeplinită de sistemul de AD&C prin intermediul unei varietăți de componente ce includ: convertoare analog-numerice, multiplexoare, circuite de eșantionare/memorare, amplificatoare, circuite de numărare și temporizare precum și alte funcții mult mai specializate. Caracteristica cea mai importantă, poate, a sistemelor de AD&C este tocmai abilitatea de a integra aceste funcții sofisticate într-un sistem unitar. La alegerea (configurarea) unui sistem de AD&C este util să avem cunoștințele de bază despre principiile de achiziție de date.

Sisteme cu intrări analogice

Funcția fundamentală a unui sistem cu intrări analogice este de a converti semnalele analogice într-un format numeric corespunzător. Această funcție este realizată de convertorul ANALOG-NUMERIC (Analog to Digital - A/D), care transformă informația analogică în cod binar.

Pe lângă convertorul A/D, pentru obținerea unor performanțe optime este necesară utilizarea unor componente ca de exemplu: un amplificator, un circuit de eșantionare/memorare și elemente de preprocesare a semnalului.

CONVERTOARE A/D - Azi, există un număr semnificativ de tipuri de convertoare A/D. Dintre acestea unele se diferențiază prin faptul că sînt cele mai utilizate, ca de exemplu: aproximări succesive (Successive Approximation Regulator - SAR), integrare (dual slope) și convertoare paralele (FLASH). Deoarece convertoarele paralele sînt cele mai rapide, ele sînt și cele

mai scumpe. Majoritatea aplicațiilor de AD&C, necesită o rezoluție de cel puțin 12 biți. Convertoarele de rezoluție mai mare sînt nu numai mai scumpe dar și mai lente, fapt previzibil de altfel. De aceea, este foarte important să definim cît mai precis necesitățile unei aplicații înainte de a ne decide asupra rezoluției folosite.

Un punct de pornire bun este **TRĂDUCĂTORUL**. Unele traductoare au o dinamică foarte mare. **DINAMICA** reprezintă diferența dintre nivelul maxim al semnalului de intrare și cel mai mic nivel detectabil. Nu există o corelare precisă între precizia și dinamica unui traductor. De exemplu un traductor cu precizia de 0,5% poate avea o dinamică de 80 decibeli (dB). Acest traductor are nevoie de un sistem cu o rezoluție de 12 biți. Pentru a menține dinamica maximă, unele aplicații necesită o rezoluție de 14 sau chiar 16 biți. Amplificarea unui semnal de nivel mic, de 10 sau 100 de ori produce o creștere a rezoluției cu 3 respectiv 6 biți.

Un sistem de 12 biți împarte domeniul maxim al semnalului de intrare, în 4096 de părți egale. Astfel, o parte reprezintă $1/4096 = 0,025\%$ din întreg domeniul de lucru. O parte a unui sistem de 16 biți, reprezintă $1/65536 = 0,0015\%$ din întreg domeniul de lucru. De aceea, rezoluția nu determină numai dinamica, dar limitează și precizia întregului sistem. Pe de altă parte, mărind rezoluția unui sistem, nu putem beneficia de acest avantaj decît dacă celelalte componente din sistem, ca de exemplu amplificatoarele sau circuitele de eșantionare/memorare nu acționează ca factori limitatori în sistem.

Cînd variația unui semnal de intrare este mai mică decît rezoluția **MINIMĂ** a sistemului, atunci acest eveniment va trece neobservat. De exemplu, la folosirea unui sistem de AD&C de 12 biți (fără nici o preamplificare prealabilă), orice variație de semnal care nu

depășește 2,44 mV în domeniul de lucru maxim de 10 V, nu va putea fi detectată de sistemul de achiziție. Pe de altă parte, dacă semnalul este mai întîi amplificat de 1000 de ori înainte de conversie, rezoluția poate fi crescută la 2,44 mV (în absența zgomotului).

La viteze mai mari de 100 de eșantioane/secundă, convertorul bazat pe metoda **APROXIMĂRIILOR SUCCESIVE** este cel mai indicat. De fapt, se pot obține viteze de peste 100.000 eșantioane/secundă. Semnalul de intrare se compară cu semnale generate prin "GHICIRI" ponderate binare, pînă cînd cele două semnale devin identice. Este esențial ca semnalul de intrare să rămînă **NESCHIMBAT** pe timpul comparațiilor succesive, altfel, pot rezulta erori semnificative. Pentru a menține semnalul de intrare **CONSTANT**, se folosește un circuit de eșantionare/memorare.

Cînd nu se lucrează la viteze mari, un convertor A/D bazat pe principiul **INTEGRĂRII** poate oferi rezoluții de 12, 14 sau chiar 16 biți la un preț redus. Viteza de eșantionare nominală este cuprinsă între 3 și 50 de conversii/secundă.

După cum îi spune și numele, acest convertor **MEDIAZĂ** orice variație a semnalului de intrare pe timpul ciclului de conversie. Această proprietate filtrează în mod automat zgomotul de la intrare. Liniaritatea și precizia pe întreg domeniul de lucru este în general mai bună decît la celelalte tipuri de convertoare A/D.

PRECIZIA este o mărime importantă a unui sistem de intrări analogice. Ea se definește ca eroarea totală la o anumită intrare. De exemplu, un sistem de AD&C care are o precizie de 0,05% pe întreg domeniul de 10 V, va genera o eroare, în cazul cel mai defavorabil, de 5 mV ($10 \text{ V} * 0,0005$). Dacă precizia sistemului este de 0,1% pe un domeniu de +/-10 mV (de exemplu convertorul A/D pe domeniul de +/-10 V și un amplificator cu

amplificare de 1.000) sistemul va genera în cazul cel mai defavorabil o eroare de 20 microvolți ($20 \text{ mV} * 0,001$).

În concluzie, aprecierea preciziei unui sistem de AD&C cere o analiză foarte atentă. Asigurați-vă că precizia dată se referă la întreg domeniul de lucru.

AMPLIFICATOARE - Amplitudinea semnalelor de intrare analogice poate varia în domenii foarte largi. Convertorul A/D necesită un semnal de amplitudine ridicată ca să funcționeze optim. În majoritatea sistemelor se utilizează un amplificator pentru a ridica nivelul semnalelor mici la valorile cerute.

Ideal, amplificatorul de intrare ar trebui să aibă mai multe trepte de câștig controlate prin soft. Acest dispozitiv este cunoscut sub numele de amplificator cu câștig reglabil (Programmable Gain Amplifier - PGA). În realitate însă, compromisul între cost și performanțe ne poate dicta folosirea unui PGA sau a unui amplificator cu câștig modificabil MANUAL. Aici, "manual" înseamnă selecția unui rezistor sau a unui ștrap pentru a fixa câștigul pe placă.

Un etaj de intrare analogic se prezintă în figura 5. Să reținem că amplificatorul de intrare nu este necesar la orice aplicație. După cum se vede acest circuit crează posibilitatea citirii doar a unui singur canal analogic.

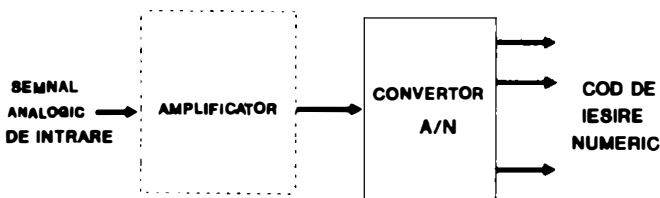


Fig. 5. Etaj de intrare analogic

O metodă care permite măsurarea mai multor canale ar fi folosirea unui convertor A/D și a unui amplificator pentru fiecare canal de intrare. Oricum, există o metodă mult mai ieftină, descrisă mai jos.

MULTIPLEXOARE - Multiplexorul (MUX) arătat în figura 6 este o arhitectură de comutatoare electronice care permite ca un număr mare de canale să poată fi prelucrate folosind un singur amplificator și un singur convertor A/D. Softul poate controla aceste comutatoare analogice pentru a selecta oricare din canalele dorite în vederea procesării la un anumit moment dat de timp. Această metodă oferă o reducere considerabilă a costului, față de utilizarea amplificatoarelor și convertoarelor A/D, separat pentru fiecare canal în parte.

Deoarece amplificatorul și convertorul A/D sînt folosite în comun, viteza de achiziție se va reduce. Ca primă aproximație, viteza amplificatorului și a convertorului A/D va fi divizată la numărul canalelor de intrare gestionate.

RANDAMENTUL EȘANTIONĂRIILOR se definește prin viteza unui canal înmulțită cu numărul de canale.

EȘANTIONARE ȘI MEMORARE (SAMPLE & HOLD - S/H) - În general, semnalul analogic de intrare poate varia în timp. După cum am văzut,

convertoarele A/D bazate pe aproximații succesive sînt condiționate de cerința ca semnalul de intrare să NU se modifice pe perioada ciclului de conversie. Funcția circuitului de eșantionare și memorare (S/H) este de a eșantiona valoarea prezentă la intrare înainte de începerea conversiei. Acest nivel de semnal este ținut constant, în ciuda modificării semnalului de la intrare, pînă cînd ciclul de conversie s-a terminat. Această proprietate asigură conversia precisă a semnalelor de înaltă frecvență.

MULTIPLEXAREA ÎN TIMP - În sistemul descris în figura 6, canalele folosesc în comun amplificatorul, circuitul S/H și convertorul A/D. Utilizatorul selectează rata de eșantionare dorită în funcție de aplicația dată. Dacă fiecare canal trebuie citit de R ori pe secundă, atunci MUX-ul va trebui să eșantioneze cu o rată egală cu $n * R$ (unde n este numărul canalelor citite). Deci, circuitul S/H și convertorul A/D trebuie să fie atât de rapide, încît să execute un ciclu de conversie într-un timp mai mic decît $1/n * R$ secunde. Trebuie să avem grijă, deci, să nu fim induși în eroare de specificațiile de viteză a diferitelor componente din sistem.

TIMPUL DE CONVERSIE - definește numai viteza convertorului A/D, care reprezintă numai o parte

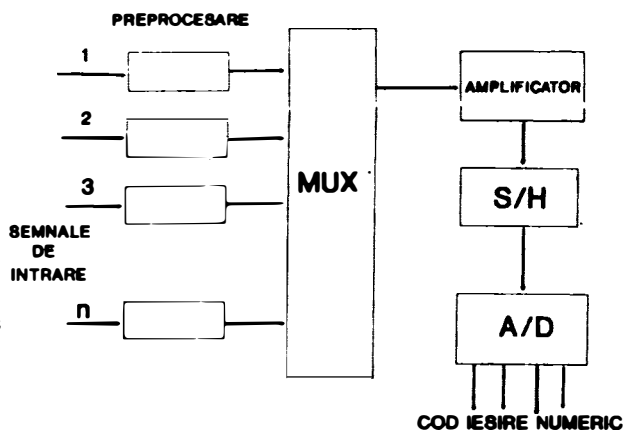


Fig. 6. Subsistem analogic de intrare

din timpul total necesar pentru a măsura un canal dat. Deci, pentru a afla viteza reală a întregului sistem trebuie să știm atât rata de eșantionare per canal sau randamentul eșantionării cât și condițiile în care au fost specificate (de exemplu randamentul eșantionării este funcție de câștigul amplificatorului). Ideal, toate canalele de intrare vor fi citite în același moment de timp, adică la fiecare $1/R$ secunde. Oricum, multiplexarea în timp generează o "ASIMETRIE" sau o diferență de timp între citirile canalelor. Dacă MUX-ul, S/H-ul și convertorul A/D sînt suficient de rapide, atunci putem avea impresia că intrările sînt citite simultan. Unele aplicații sînt foarte sensibile la asimetria de timp, ca de exemplu măsurarea puterii electrice instantanee ($I^2 \cdot U$) sau măsurarea poziției relative a componentelor mecanice. Chiar dacă folosim cele mai rapide convertoare A/D, există încă multe aplicații care nu pot "tolera" diferența de timp între citiri rezultată în urma eșantionărilor secvențiale. În aplicații critice, tehnica folosirii circuitelor de eșantionare/memorare simultane, poate reduce asimetria de timp de 100 pînă la 1000 de ori. Această arhitectură este ideală pentru aplicații în care relația dintre fază și timp a multiplexorului de intrare este critică în tehnica investigației folosite. De exemplu, dacă sistemul din figura 6 ar eșantiona 4 intrări analogice cu un randament de 89.000 eșantioane/secundă, timpul mort între două conversii ar fi de 11,25 microsecunde. Am avea nevoie de 45 microsecunde pentru a digitiza toate cele 4 canale. Aceasta reprezintă o diferență de fază de 162 de grade, între primul și al 4-lea canal, la o frecvență a semnalului de intrare de 10 kHz (45 microsecunde * 360 grade). Pe de altă parte, sistemul ce utilizează tehnica circuitelor S/H simultane din figura 7, poate captura toate cele 4 canale la o diferență de 10 nsec, una față de alta, ceea ce reprezintă o diferență de fază mai mică de 0,04 grade la 10 kHz.

Această tehnică este recomandată mai ales acolo unde trebuie calculate funcții de corelație. De exemplu, analiza vorbirii, măsurarea puterii în sistemele trifazate, analiza semnalelor geofizice și testarea automată a echipamentelor (ATE).

PREPROCESAREA SEMNALELOR - Chiar dacă se utilizează componente de înaltă calitate, este de dorit o preprocesare a semnalelor de la intrare. Preprocesarea, poate fi de două tipuri: **ACTIVĂ** - care include o amplificare și o izolare electrică, sau **PASIVĂ** - care include o divizare de tensiune, conversie curent-tensiune, filtrare și suprimarea brumului de rețea. Amplitudinea maximă a semnalului care poate fi aplicată la intrarea unui amplificator sau convertor A/D este de obicei +/-10V. Divizori rezistivi se folosesc pentru a transforma orice nivel de tensiune în acest domeniu acceptat de valori.

Astfel, se poate realiza monitorizarea unui semnal de 48 V sau 480V. Este important a se lua în considerare că multiplexorul și celelalte componente electronice se pot avaria, dacă se conectează la intrarea acestora semnale ce depășesc 15 V. Prin adăugarea la circuitele de preprocesare a unor dispozitive de limitare, cum ar fi diode Zenner sau MOV-uri (Metal Oxid Varistor), se poate asigura o

protecție față de posibilele vîrfuri de tensiune de la intrare.

Cîteodată este de dorit să se preamplifice semnalele de nivel mic (1mV la 1V) în afara cutiei în care se găsește convertorul A/D pentru a obține un raport semnal-zgomot maxim. Una din aceste metode este "EMIȚĂTORUL PE DOUĂ FIRE". Emițătoarele realizează nu numai o amplificare a semnalelor de intrare ci pot face și izolarea electrică, liniarizarea și conversia în curent la o valoare relativ mare (tipic 4 la 20 mA) Transmiterea semnalelor în curent se poate face la distanțe de pînă la 1500m fără diminuarea semnificativă a preciziei de lucru, în timp ce semnalele de tensiune sînt atenuate foarte repede de rezistența cablului de interconectare. Semnalele de curent se pot comuta simplu în tensiune prin intermediul unui rezistor. Valorile cuprinse între 250 și 500 de ohmi sînt cele mai uzuale, producînd căderi de tensiune de 5 pînă la 10V corespunzător valorilor de curenți între 4 și 20 mA)

Dintre toate categoriile de preprocesare, filtrarea este cea mai dorită, cea mai utilizată și cea mai prost înțeleasă. Practic, filtrarea este folosită pentru a separa semnalele dorite de cele nedorite. Semnalele nedorite includ: zgomot, brum de rețea, interferență electromagnetică captată de la

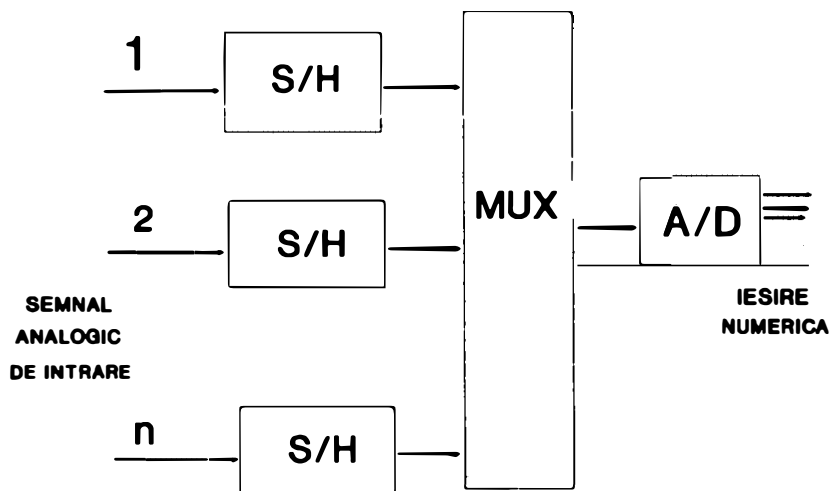


Fig. 7. Sistem cu circuite S/H simultane

stațiile de radio - TV și semnalele cu frecvența peste 1/2 din frecvența de eșantionare. În general, se folosește un filtru trece-jos (LOW PASS FILTER) pentru a atenua aceste surse nedorite de eroare. Când la intrarea convertorului A/D, apar semnale cu frecvența peste 1/2 din frecvența de eșantionare folosită, ia naștere un fenomen numit FALSIFICARE (ALIASING). Aceasta se manifestă prin generarea de semnale "FALSE" în domeniul de frecvență interesat, care nu pot fi distinse de informația reală. Astfel, se pot produce erori foarte mari în interpretarea datelor.

Semnale în mod comun (SINGLE - ENDED) și semnale diferențiale (DIFFERENTIAL)

Semnalele analogice pot fi configurate ca intrări de mod comun sau diferențiale. Intrările de mod comun se definesc toate în raport cu o singură MASĂ comună. Aceste semnale se pot utiliza doar în cazul în care căderea de tensiune pe traseul de masă este mică. Problemele încep să apară când această cădere de tensiune crește. Aceasta generează bucle de masă (scurgeri de curenți prin conductorul de masă) ceea ce poate genera tensiuni de eroare la intrările convertorului A/D. Principalul avantaj al acestei metode este costul scăzut per canal. Este necesar doar un singur multiplexor pentru a trata fiecare canal în parte.

Intrările diferențiale folosesc două fire: intrarea inversoare (-) și intrarea neinversoare (+) a amplificatorului, pentru a se conecta la sursa de semnal. Astfel, orice semnal indus printr-o buclă de masă, va apare ca un semnal de mod comun și va fi rejectat de proprietățile diferențiale ale amplificatorului.

Conexiunile diferențiale pot reduce considerabil efectele buclilor de masă, dar ele necesită două

multiplexoare per canal. Astfel, un sistem de 32 de canale în mod comun, va avea 16 intrări diferențiale.

În unele aplicații se utilizează și așa numitele conexiuni PSEUDO - DIFERENȚIALE. Aceasta este de fapt, o conexiune în mod comun, în care una din intrări este conectată la masa comună a semnalelor de intrare. Astfel, acest canal, măsoară tensiunea indusă în bucla de masă, care poate fi corectată prin soft. Această tehnică este folosită acolo unde toate semnalele de intrare folosesc ca referință același potențial de masă.

Amplificatoare de Instrumentație - INSTRUMENTATION AMPLIFIERS (IA) - Un amplificator de instrumentație este un bloc cu intrări diferențiale cu impedanță de intrare foarte mare la ambele intrări, + și -. Proprietățile de rejecție de mod comun, atenuează efectele buclilor de masă, brumul de rețea precum și semnalele de eroare induse prin zgomot. Astfel IA, este folosit în special pentru a mări semnalele de nivel mic. Dacă IA are câștigul programabil prin soft, el se numește PGIA - PROGRAMMABLE GAIN INSTRUMENTATION AMPLIFIER (amplificator de instrumentație cu câștig programabil).

Ideal, impedanța de intrare, rejecția de mod comun și lărgimea de bandă a amplificatoarelor ar trebui să fie infinit de mare. Aceasta ar însemna ca valorile curenților de intrare și tensiunii de offset să fie NULE. Cu alte cuvinte, acest lucru ar însemna ca sursa de semnal să nu fie influențată de circuitul de măsurare.

În realitate, însă, amplificatoarele au impedanțe de intrare, curenți de intrare și tensiuni de offset de valori finite.

TENSIUNEA DE OFFSET - (Vos) a unui amplificator se referă la tensiunea de ieșire a amplificatorului când la intrare se aplică o

tensiune NULĂ (intrările sînt scurt-circuitate). Practic, Vos este tensiunea ce ar trebui amplificată la intrările +/- ale amplificatorului pentru ca ieșirea lui să fie NULĂ. Existența Vos, se datorează micilor diferențe existente între componentele ce alcătuiesc circuitul de intrare al unui amplificator.

Cu toate că, majoritatea amplificatoarelor sînt prevăzute cu reglaje pentru eliminarea offsetului, aceasta se face însă sacrificînd alți parametri. De exemplu eliminarea offsetului prin reglaj, generează deseori creșterea driftului de offset (Vos se modifică cu temperatura) și alte efecte nedorite. Vos poate fi compensată și prin soft. În cele mai multe cazuri, CURENȚII DE INTRARE dau mai multă bătaie de cap. Pentru denumirea curenților de intrare se folosesc doi termeni: CURENTUL DE BIAS (BIAS CURRENT - Ib) și CURENTUL DE OFFSET (OFFSET CURRENT - Ios). Ib se referă la curentul care intră sau ieșe în sau din terminalul + sau - al amplificatorului. Curentul Ios este diferența dintre curenții de bias la intrărilor + și -. În principiu, diferența dintre cei doi termeni este importantă, pentru că Ios poate fi mult mai mic decît Ib. Acești curenți interacționează cu impedanța sursei de semnal, producînd o tensiune de offset adițională. Dacă impedanța sursei este balansată (BALANCED), adică ea este identică la ambele intrări, + și -, se generează eroare numai datorită lui Ios.

Este esențială existența unui conductor extern între intrările amplificatorului și masa sursei de alimentare, a cărei rezistență trebuie să fie atît de mică încît tensiunea de offset rezultată, $I_b \cdot R_s$ să nu înrăutățească performanțele amplificatorului. În situația extremă, când intrările amplificatorului sînt lăsate LIBER, adică nu există conductor de masă, amplificatorul ajunge într-o stare în care NU poate fi utilizat.

REJECTIA DE MOD COMUN -

Abilitatea amplificatorului cu intrare diferențială de a discrimina între semnalul de intrare diferențial și semnalul de mod comun (semnalul nedorit) pentru amplificarea diferențiată se definește prin raportul rejecției de mod comun (Common Mode Rejection Ratio - CMRR) în decibeli. Pentru un amplificator dat, raportul dintre amplificarea sa în mod comun (G_{cm}) și amplificarea sa diferențială (G_{diff}) este:

$CMRR = 20 \log_{10}(G_{cm}/G_{diff})$
 exprimat în decibeli.

Amplificarea în mod comun se determină prin măsurarea variației tensiunii de ieșire care rezultă datorită unei schimbări la intrarea de mod comun.

$$G_{cm} = DV_{out}/DV_{cm}$$

Înlocuind această valoare în ecuația CMRR, obținem:

$$DV_{out} = DV_{cm} * G_{diff} * 10^{(-CMRR/20)}$$

Ca o regulă generală, CMRR-ul unui amplificator se îmbunătățește odată cu creșterea amplificării sale în buclă închisă. Într-un sistem complet de AD&C semnalul de ieșire din amplificator (incluzând erorile datorate valorii finite a CMRR) ajunge la intrarea convertorului A/D. Acesta nu poate discerne între părțile utile și eronate ale semnalului de la intrare.

Semnificativă este numai relația care există între amplitudinea semnalului de eroare și sensibilitatea convertorului A/D (de exemplu 1 LSB - bitul cel mai puțin semnificativ). Dacă eroarea depășește 1 LSB, convertorul A/D, răspunde. De aceea este mult mai de dorit ca să exprimăm CMRR în unități LSB. Acest lucru se face prin divizarea tensiunii de eroare de mod comun (DV_{out}) la SENSIBILITATEA convertorului.

Sensibilitatea convertorului A/D, se calculează prin divizarea întregului domeniu de lucru (FSR -

FULL SCALE RANGE). la rezoluție

- Eroarea de mod comun (în unități LSB) = $(DV_{cm} * G_{diff} * 10^{(CMRR/20)})/FSR/Rezoluție$

Tabela ce urmează, ne arată relația dintre eroarea CMRR, exprimată în decibeli și unități LSB, pentru un sistem ipotetic. Pentru această comparație s-a considerat un sistem cu o rezoluție de 12 biți pe un domeniu de 10V (0 la 10V sau +/-5V). Un semnal de mod comun de 10V este aplicat la intrările scurtcircuitate ale sistemului. Împărțind eroarea de mod comun (exprimată în unități LSB) la tensiu-

Ciștig de semnal	CMRR	Eroare CM absolută	Eroare CM/VCM
1	80dB	0.4LSB	0.04LSB/Volt
10	90dB	1.3LSB	0.13LSB/Volt
100	100dB	4.1LSB	0.41LSB/Volt
1000	110dB	13LSB	1.3LSB/Volt

nea de mod comun, rezultă o informație utilă cu privire la performanțele întregului sistem. Observați că, o creștere a CMRR la un câștig dat, realizează o creștere a performanțelor sistemului. Oricum, o creștere relativ mică a CMRR, obținută ca urmare a unei creșteri a câștigului, produce o diminuare a preciziei sistemului.

IEȘIRI ANALOGICE - În majoritatea sistemelor, ieșirile analogice sînt necesare. Aceste semnale se folosesc ca semnale de reacție la controlul în buclă închisă. Domeniul de variație a acestor semnale poate fi: +/-5V, +/-10V, 0-10V și 4-20mA. Ieșirile analogice, se realizează cu convertoare numeric/analogice - D/A, care pot lucra în sarcină, cu curenți de pînă la 5 sau chiar 10 mA. Aceasta nu reprezintă o limitare, pentru că majoritatea aplicațiilor, necesită comanda unor sarcini ce au impedanțe ridicate. În cazul în care se comandă sarcini cu impedanțe de valori mici, ca de exemplu: electrovalve, lămpi indicatoare, motoare electrice etc., se folosesc amplificatoare de putere sau generatoare de curent constant de valori mari. Acestea se

vor atașa la sistem, deoarece majoritatea sistemelor de AD&C NU includ etaje de putere.

INTRĂRI ȘI IEȘIRI NUMERICE

- Cele mai multe sisteme de AD&C pot accepta și genera semnale TTL (0 la 5V). În cazul aplicațiilor ce folosesc alte nivele discrete de tensiune, sînt necesare interfețe speciale. Pentru a comanda dispozitive ca: bobine, motoare și rele sînt necesare nivele mai mari de tensiune și curenți.

Pentru a facilita conectarea cablurilor de la traductori la sistemul de AD&C există o varietate de tipuri de BORNIERE (TERMINATION PA-

NELS). Aceste borniere, conțin șuruburi adiționale de cuplare la proces pentru modulele de preprocesare a semnalelor, indicatoare luminoase - LED pentru afișarea stărilor canalelor urmărite, divizoare de tensiune și circuite de izolare electrică. Astfel, se pot monitoriza atît nivele înalte de curent continuu, cît și circuite de curent alternativ.

INTRĂRI ȘI IEȘIRI DE

NUMĂRARE - Există o varietate largă de aplicații de numărare. Alte aplicații cer ca anumite dispozitive, să fie pornite sau oprite la momente de timp foarte precise. Toate aceste funcții, pot fi rezolvate cu circuite de tip numărare/temporizare (COUNTER/TIMER - C/T), care sînt optimizate pentru aplicații ce includ măsurarea de frecvențe și generarea bazelor de timp. Numărătoarele se caracterizează prin numărul evenimentelor de intrare care se pot urmări și de frecvența lor maximă de la intrare. Majoritatea sistemelor folosesc numărătoare de 16 biți care pot înregistra impulsuri la frecvența maximă de 8 MHz. Capacitatea maximă a numărătoarelor de 16 biți

este de 65536 de evenimente. Aceste numărătoare sînt toate independente unele față de celelalte și pot fi utilizate pentru numărarea de evenimente, măsurarea sau divizarea de frecvențe. Generatoarele de impulsuri, permit programarea prin soft a duratei impulsurilor cît și a frecvenței acestora. Un asemenea generator de impulsuri, se folosește, de obicei pentru a genera o bază de timp precisă, folosită pentru achiziția datelor la momente precise de timp. Numărătoarele sistemelor de AD&C, acceptă nivele TTL și pot număra pe un număr de canale predefinit prin soft. Numărarea poate fi startată de la o valoare inițială definită și, de asemenea, numărătorul poate fi configurat pentru a se reseta la această valoare după ce valoarea sa a fost citită.

Intern, numărătorul își decrementează valoarea la fiecare impuls sosit la intrare. Dacă un numărător de 16 biți depășește valoarea de 65535, are loc depășirea capacității sale, generîndu-se un semnal de depășire. Acest semnal poate fi folosit la activarea unor evenimente externe.

Măsurarea frecvențelor folosind numărătoare, poate fi realizată pe mai multe căi, dependent de tipul aplicației. Dacă frecvența necunoscută este un semnal TTL, el se poate aplica direct la numărător. Semnalele analogice cu o amplitudine de cel puțin 100 mV, pot fi convertite în semnale TTL, cu ajutorul modulelor de TRIGGERARE (ALARM/TRIGGER). Divizoarele de tensiune folosind rezistențe și/SAU diode Zenner sau opto - izolatori, pot fi utilizate la divizarea cu aceste trepte, a semnalelor de intrare de nivele mari. În cazul în care se utilizează circuite de pre-procesare, este necesară o analiză în ceea ce privește limitarea de viteză.

Există două opțiuni distincte pentru măsurarea frecvențelor joase și înalte. Prima metodă deter-

mină perioada semnalului necunoscut, comparîndu-l cu perioada cunoscută a semnalului unui generator de ceas. Avantajul acestei metode este că permite o măsurare a semnalelor de joasă frecvență cu o rezoluție mare și minimizează timpul necesar folosit pentru măsurare. De obicei, metoda se folosește sub frecvențe de 10 Hz.

A doua metodă, numără ciclul semnalului necunoscut într-un interval de timp FIXAT. Avantajul acestei tehnici este că permite măsurarea semnalelor peste 8 MHz. Este relativ ușor să se implementeze un algoritm de auto - calibrare ce optimizează rezoluția peste un domeniu foarte larg de frecvențe.

Tehnici software

Softul este cel care face viabil un sistem de AD&C. Un hard ales are o valoare nesemnificativă fără soft. Achiziția de date pe PC-uri a fost concepută în așa fel, încît utilizatorii au posibilitatea de a scrie programe speciale pentru achiziție de date, numărare, afișare și control, în limbaje de nivel înalt. Cînd un soft este livrat cu un sistem de AD&C, acest lucru se face în așa fel încît anumite taskuri să fie cît mai simplu de realizat pentru utilizator.

În general, există, pe PC-uri 3 clase de soft pentru sistemele de AD&C:

- 1) programe de învățare și dezvoltare
- 2) librării de subrutine, ce conțin funcții
- 3) pachete de aplicații cu meniuri, livrate la "CHEIE"

Programele de la punctele 1 și 2 sînt proiectate pentru a permite utilizatorilor posibilitatea de a-și concepe și de a scrie singuri o aplicație soft, UNICĂ. Acestea includ, de obicei "DRIVERE" care reprezintă interfața cu hardul de I/E. Aceste pachete, fac ca scrierea programelor în limbaje de nivel înalt, ca de exemplu BASIC, C, TURBO PAS-

CAL sau ASSYST să fie foarte ușoară. Acest tip de programare este foarte flexibil și util pentru scopuri generale.

Programele de la punctul 3 sînt făcute pentru a pune în funcțiune IMEDIAT întregul sistem de AD&C, de obicei fără necesitatea de a mai scrie alte module program. Oricum, unele dintre aceste pachete oferă utilizatorilor posibilitatea de a modifica softul la cerințele lor. În mod normal, acest tip de produs este oferit pentru o aplicație specifică. De aceea, aceste pachete sînt mai puțin structurate și mai puțin flexibile ca cele din alte clase. Tot aici, putem aminti numeroase colecții de SOFT GENERIC, care sînt programe scrise de firme independente pentru produsele hard ale altor firme. Unele dintre cele mai cunoscute includ: LABTECH NOTEBOOK, HYPERSIGNAL-WORKSTATION și SNAP. Acestea, precum și multe altele, pot realiza pe lîngă culegerea, analiza, plotarea datelor și funcții de control.

Multe, dacă nu chiar toate, dintre aplicațiile de AD&C depind de execuția în timp a operațiilor de intrare/ieșire. Cînd viteza și/SAU timpul este critic, este necesară considerarea a 3 tehnici de lucru soft: POLLING (baleierea), prin ÎNTRE-RUPERI și DMA.

POLLING-ul este cea mai simplă metodă de a detecta îndeplinirea unei condiții și luarea unei decizii. Aceasta implică o buclă de program care conține toți algoritmi de măsurare, analiză și luare a deciziilor. Programul de AD&C testează periodic ceasul sistemului sau o intrare conectată la un element extern pentru a analiza dacă nu s-a produs o schimbare. Cînd apare o schimbare, programul eșantionează toate intrările și memorează valoarea acestora într-un TAMPON. Un tampon este o zonă de memorie care conține valorile ce reprezintă intrările la un anumit moment dat de timp. Tamponul poate fi generat în RAM, pe disc sau pe alte tipuri de memorii. De

fiecare dată când programul "SIM-TE" o tranziție a ceasului de sistem - un TICK - intrările sînt eșantionate și convertite și o nouă valoare se adaugă în tampon.

În acest mod, un PC/AT 286 poate achiziționa date cu o rată de aproximativ 18kHz, depinzînd de ceasul procesului. Pe de altă parte, PC-ul este proiectat în așa fel încît trebuie avută în vedere o toleranță de timp (astfel, o toleranță de timp de 12 microsecunde la un PC este un lucru obișnuit. De asemenea, PC-ul este încontinuu ocupat cînd execută bucla de polling și astfel nu mai poate executa alte taskuri. Cînd un program nu poate "tolera" aceste restricții, va trebui să aplicăm tehnica întreruperilor.

ÎNTRERUPERILE permit, pe lîngă controlul precis al timpului într-un sistem de AD&C, și executarea altor taskuri concomitent cu supravegherea procesului. Sistemele multitasking sînt cunoscute și sub numele de sisteme "FAȚĂ/SPATE" - FOREGROUND/BACKGROUND. Una din metodele de a pune taskul de achiziție de date în "spate" este de a-l lega de o rutină de întrerupere. Semnalul de ceas sau un semnal extern, în loc de a fi urmărit continuu prin polling, este folosit pentru a genera o întrerupere în PC. La apariția unei întreruperi, calculatorul își suspendă activitatea

curentă și execută o rutină de întrerupere de serviciu (ISR - INTERRUPT SERVICE ROUTINE). ISR-ul, în acest caz poate fi un program scurt care eșantionează toate intrările și le memorează în tampon. Calculatorul poate executa și alte operații în "față" în timp ce culege datele în "spate". La apariția unui TICK sau a unei întreruperi externe, calculatorul va reveni în punctul din program de unde a plecat.

Timpul de răspuns al sistemului de întreruperi este mult mai lung decît timpul unei bucle de polling bine concepute. Acest lucru se datorează faptului că, mecanismul de întreruperi, în majoritatea calculatoarelor, implică o cantitate însemnată de soft. Viteza unui PC, în modul de lucru cu întreruperi, este de circa 4 kHz. De asemenea, pot apărea complicații la scrierea unui soft de întreruperi. În mai toate cazurile, programatorul, trebuie să fie pregătit să scrie programe în limbaj de asamblare. În contrast, majoritatea sistemelor ce utilizează metoda polling pot fi scrise în limbaje de nivel înalt. Întreruperile sînt avantajoase în situații în care rata de achiziție de date este lentă, precizia în timp nu e prioritară și aplicația "din spate" este cea importantă. Cînd timpul necesar pentru a executa o rutină ISR este mic în comparație cu rata de apariție a întreruperilor în sistem, atunci această tehnică

poate da rezultate excelente. Aceste considerente trebuie bine cîntărite înainte de a alege între un sistem bazat pe polling și altul bazat pe întreruperi.

Metoda accesului direct la memorie (DIRECT MEMORY ACCESS - DMA) este o tehnică hard care asigură cea mai mare viteză de transfer a datelor, din sau în RAM. Tehnicile DMA utilizează echipament hard mai scump și pot realiza citiri și scrieri de date la momente precise de timp, fără impunerea unor restricții semnificative în ceea ce privește microprocesorul în execuția taskurilor.

De exemplu, PC-ul lucrînd în mod DMA poate citi sau scrie orice combinații de date (analogice, numerice sau de numărare) din sau în RAM la o rată maximă de 360 kocteți/secundă. Această performanță este obținută prin "FURT DE CICLU - CYCLE STEALING", adică prin "oprirea" pentru perioade scurte de timp a UC, pentru a permite executarea în acest timp a unor transferuri directe în sau din memorie - fără participarea unității centrale. Timpul necesar achitării unei cereri DMA este mult mai mic decît cel necesar executării unui ISR, ceea ce permite realizarea comutării rapide între aplicații.

(ing. Kalló Tibor)

Cîteva adrese

La solicitarea unor corespondenți ai revistei noastre, vă prezentăm adresele unor sedii principale ale cîtorva dintre cele mai importante firme din domeniul instrumentației de laborator.

Burr Brown - Burr Brown Overseas Marketing Office, 1 Milfield House, Woodshots Meadow, Watford, Hertfordshire, England, WD1 8YX

Keithley - Keithley Instruments GmbH, Heiglhofstrasse 5, 8000 München 70

Metrabyte - 440 Myles Standish Boulevard, Taunton, MA 02780

Phillips - Philips Export B.V., I&E Export, Building HW3, P.O.Box 218, 5600 MD Eindhoven, Netherland;

● reprezentanță în România: Institutul Politehnic Philips Service, P.O.Box 1/285, str. Polizu 1, București, dl. Stoian

Siemens - Siemens AG, Bereich Automatisierungstechnik, Geschäftsgebiet Messtechnik, Postfach 4848, D-8500 Nürnberg 1

Rohde & Schwarz - Rohde & Schwarz GmbH, Sonnleithnergasse 20, A 1100 Wien

Ce este IEEE 488?

Folosirea calculatoarelor personale în tehnica măsurătorilor PCI (Personal Computer Instrumentation)

Tehnica măsurătorilor de laborator se află în fața unei schimbări esențiale. În timp ce înainte consta în achiziționarea și afișarea unor date pur analogice, lucru care se făcea cu aparate de măsură dedicate pentru efectuarea măsurării unor valori, azi predomină tehnicile digitale. Datorită existenței unor componente electronice ieftine și compacte aparatele de măsură devin din ce în ce mai "inteligente" și permit achiziționarea și interpretarea datelor rezultate din măsurători prin metode noi. Hotărîtor pentru noua dezvoltare este integrarea crescîndă a calculatoarelor personale în activitatea de laborator, PC-urile permițînd controlarea unui ansamblu de aparate de măsură conectate la calculator. Astăzi sistemele de achiziție a datelor din măsurători se împart astfel:

- - sisteme independente
 - aparate de măsură simple utilizate individual (stand-alone)
 - aparate de măsură care permit comunicația cu sisteme de măsură, dar se folosesc în principal în regim independent
- -PC-uri dedicate pentru măsurători avînd plachete specializate sau care permit racordarea unor module de extensie pentru achiziționarea datelor de măsură
- -sisteme bus pentru laborator construite după proceduri byte-seriale sau bit-seriale și care permit interconectarea de aparate de măsură cu calculatoare avînd o interfață adecvată

Avantajele utilizării calculatoarelor personale la efectuarea unor măsurători în laboratoare (în

proiectare, controlul calității, medicină, cercetare, învățămînt) sînt:

- -utilizarea ușoară și care permite o achiziție de date ușor controlabilă și interpretabilă
- -programarea ușoară a procedurii de măsură
- -repetabilitatea măsurătorilor, care permite un volum mare de măsurători în condiții identice
- -existența unor programe standard
- -o interfață utilizator unitară (display, tastatură, mouse) pentru toate aparatele de măsură din configurație
- -preluarea ușoară a datelor obținute din măsurători

Datorită avantajelor pe care le prezintă, se estimează pînă în '93 o creștere anuală de peste 20% pe plan mondial, în domeniul aparaturii ce permit efectuarea măsurătorilor cu PC-uri.

Interfața IEC 625 (IEEE-488.1 respectiv 2, HPIB, GPIB)

(denumirile distincte desemnează în principiu același concept; există însă diferențe de detaliu)

Caracteristici:

- nr. de aparate ce se pot interconecta: 15
- mod de conectare: bus pasiv, funcțiile de interfață fiind realizate în cadrul fiecărui aparat în parte
- cablul de racord: 16 fire active și 8 de masă, ecranate în comun, avînd o lungime maximă de 20m, respectiv 2m pentru fiecare aparat
- mufa de racord: 25 pini

- metoda de transmisie a datelor: byte-serială, bit paralelă, asincronă
- viteza: maxim 1 Mbyte/sec la lungime limitată de bus; la lungime maximă 200...500 kbyte/sec
- lungime maximă mesaj: 10...20 caractere
- nivel semnal: TTL
- posibilități de adresare: 1 byte de adresă ce permite adresarea a maxim 31 de emițători ("vorbitori" - talker) și 31 de receptori ("ascultători" - listener)

Interfața specificată în IEC 625 este de uz general și permite racordarea aparaturii de măsură automate (programabile) la calculator, într-un sistem bus. Comunicația între aparate au loc sub formă de mesaje. Distingem:

- -mesaje de interfață - care sînt legate de condițiile de funcționare a sistemului
- -mesaje dependente de aparat - legate de funcționarea unui anumit aparat
- Din punctul de vedere al funcțiilor interfeței aparatura se împarte în
 - emițător (talker). Aparatul care poate emite este adresabil în mod selectiv cu un mesaj de interfață și, în starea activă (adresat) poate trimite un mesaj dependent de aparat pe bus. În cadrul sistemului doar un singur aparat poate fi emițător activ la un moment dat și mesajul aflat pe bus poate proveni doar de la acesta.
 - receptor (listener). Acest tip de aparat poate fi adresat selectiv printr-un mesaj de interfață și în stare activă (adresat) poate recepționa de pe bus un mesaj de aparat. Pot fi active simultan mai multe aparate receptoare.

■ -de comandă (controller). Acest tip de aparat (de obicei calculatorul) este capabil să adreseze selectiv celelalte aparate, să asigneze funcționarea de receptor și emițător și să transmită și alte mesaje de interfață referitoare la toate sau doar anumite aparate din sistem. Aparatele având doar funcție de controller nu recepționează și emit mesaje dependente de aparat.

Funcția îndeplinită de aparate în sistemul de interfață poate fi dedicat sau modificabil. Un aparat poate avea funcții diferite în momente diferite.

Fiecare aparat are o adresă dedicată, care poate fi de emisie, de recepție sau amîndouă. Două aparate pot avea aceeași adresă de recepție, dacă trebuie să recepționeze aceleași date, dar nu pot fi două aparate cu aceeași adresă de emisie.

Configurația minimă este de un emițător și un receptor, fără aparat de comandă. De exemplu un voltmetru digital racordat la o imprimantă care tipărește rezultatele măsurătorilor semiautomat.

Structura de bus

Avantajul structurii este posibilitatea de a extinde, reduce sau reconfigura ușor sistemul, dar există dezavantajul necesității de adresa-re și comandă mai complicate, înlăturate de programele existente.

În figura 1 se prezintă structura bus-ului IEC

Pe cele 8 linii ale busului de date se transmit mesajele de interfață. Mesajele pot fi de adresă, dată de program, rezultate de măsură sau diferite instrucțiuni și pot proveni de la orice aparat cu funcție de control sau emițător. Denumirea liniilor este Data Input/Output, DIO1...DIO8.

Rolul celor 3 linii de comandă a transferului de date este de a comanda transferurile pe liniile DIO. Vitezele de transfer ale diferitelor aparate diferă semnificativ și în cazul transferului de mesaje pe mai mulți bytes este necesară o asigurare că emisia are loc atunci cînd se poate recepționa și să nu fie mai rapidă decît cel mai lent aparat de recepție. Aceste trei linii asigură

dialogul (handshake - "stringerea de mînă") asincron. Notația liniilor este:

DAV - data valid - pe nivel L înseamnă că datele sînt valide și se pot recepționa

NFRD - not ready for data - pe H indică prezența datelor pregătite pentru recepție

NDAC - no data accepted - pe H indică prîluarea datelor de către receptor

Aceste semnale sînt date de către interfețele aparatelor ce iau parte la comunicație și nu de către aparatul cu funcții de control.

Cele 5 linii de comenzi generale au fiecare un rol specific în comanda transferului de date:

IFC - interface clear - inițializează interfețele. În sistemele cu mai multe aparate cu rol de control poate fi dat doar de cel cu prioritatea cea mai mare

ATN - attention - determină modul în care se interpretează informația de pe bus-ul de date. Pe nivel

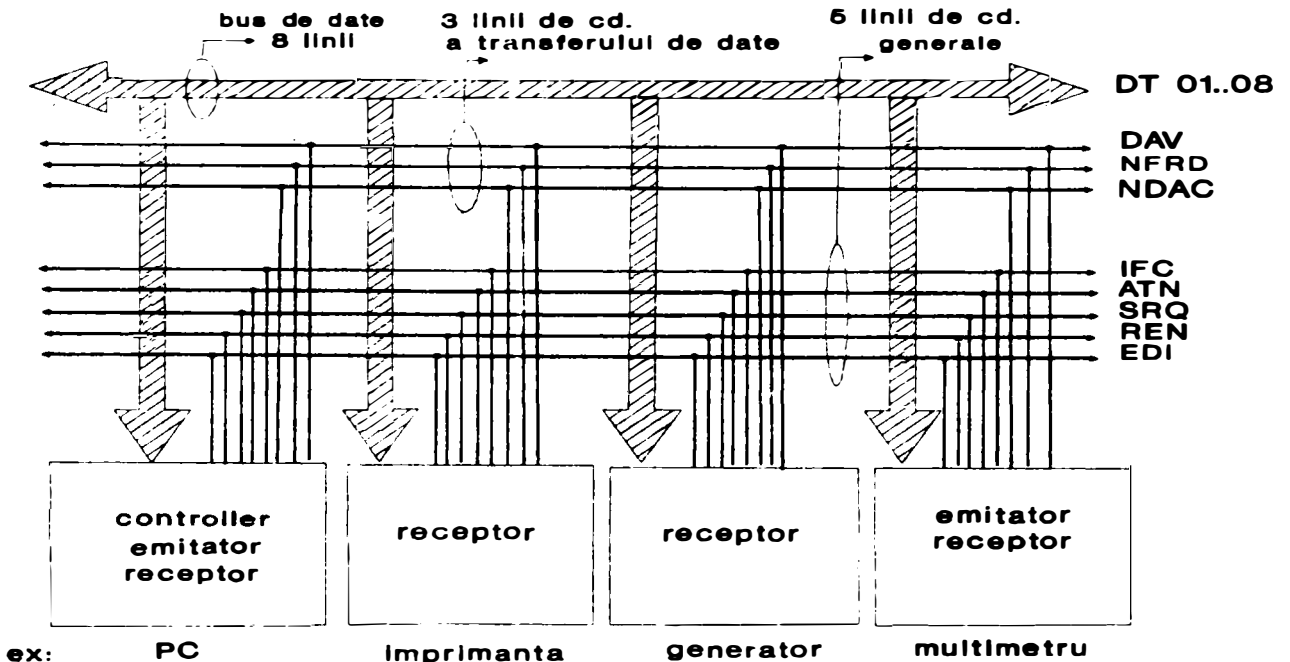


Fig. 1. Structura bus-ului IEC

La avem un mesaj de interfață iar pe H un mesaj de aparat.

SRQ - service request - pe această linie emițătorii pot cere de la aparatul de comandă (controller) întreruperea activității în curs.

REN - remote enable - această linie este folosită de aparatul de comandă (controller) împreună cu alte linii, pentru comutarea unui aparat din sistem în regim de telecomandă

EOI - end or identify - această linie este folosită pentru desemnarea ultimului element dintr-un mesaj din mai mulți bytes și de asemenea pentru interogarea (POLL) liniilor de către controller.

Mufele folosite sînt de două tipuri:

- în **standard IEC** - mufe IEC 48b (secretariat) 80A trapezoidală, cu 25 pini cilindrici (Fig. 2).
- în **standard IEEE** - mufă Amphe-nol sau Cinch series 57, MICRO-RIBBON de 24 pini (Fig. 3).

Cablurile au cîte două mufe la fiecare capăt, una tată și una mamă, ceea ce permite poziționarea lor încălecată (piggy-back) multiplă.

Diagrama de semnale de dialog (Fig. 4).

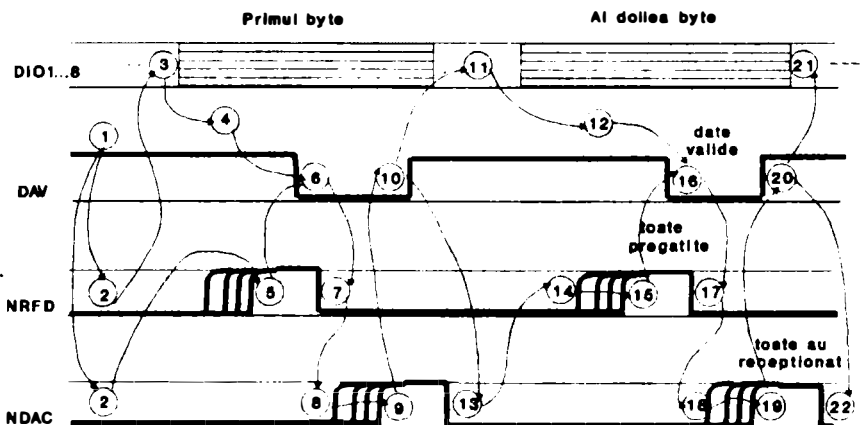


Fig. 4. Diagrama de semnale de dialog

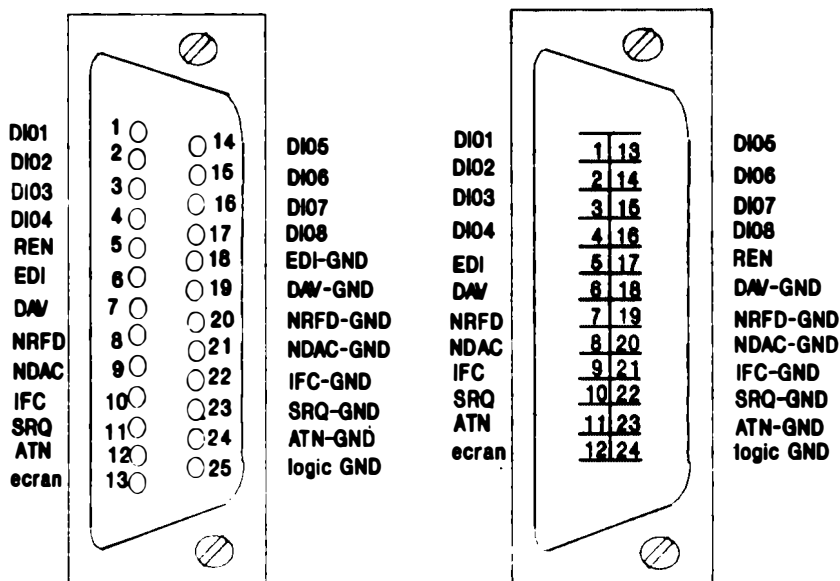


Fig. 2. Mufă standard IEC

Fig. 3. Mufă standard IEEE

La pornirea transferului de date emițătorul activ așteaptă pînă ce receptorii activi sînt gata pentru recepția unui byte de date. Datele sînt emise pe bus, După semnala-re existenței unor date valide pe bus se așteaptă ca ele să fie acceptate de toți receptorii. După ce datele au fost acceptate emițătorul emite următorul byte pe bus și așteaptă din nou ca receptorii să devină pregătiți pentru primirea datelor. Cînd acest lucru are loc emițătorul indică faptul că datele sînt valide și recepția se repetă pînă la terminarea transferului.

La punerea sub tensiune controllerul de sistem (un PC) este totdeauna controllerul activ și controlează toate tranzacțiile pe bus. Controllerul de sistem poate trece controlul unui alt aparat din sistem care devine controller activ, dar menține totdeauna controlul ultim prin liniile IFC și REN prin care poate prelua controlul oricînd do-rește.

Există o multitudine de firme care furnizează interfețe pentru bus IEEE-488. Astfel firma Burr Brown oferă PCI-801K, PCI-803W pentru bus de PC-XT/AT și PCI-802K, PCI-804W pentru MCA, Metrabyte MBC-488, IE-488 pentru bus PC-XT/AT și respectiv IBM-GPIB adapter și CEC-PC488 pentru MCA, ines -ieee 488 pentru bus NI, AT, PS2 etc.

Aceste firme oferă și softul necesar pentru o activitate de laborator complexă; Amintim aici ASYST /ASYSTANT GPIB, LOTUS MEASURE, GURU. Lab Windows, DaDisp488 etc., toate avînd, de regulă, interfețe pentru limbajele de programare cele mai răspîndite (Basic, Pascal, C).

(ing. Darvas Attila)

Digitizoarele - interfețe spre grafică și artă

Reușita muncii în studiourile CAD, sau în studiourile de grafică, depinde în mod hotărâtor de caracteristicile ergonomice și de modul de funcționare al digitizoarelor. Ele sînt interfața între mină și calculator.

Performanțele de putere ale procesoarelor din generațiile 80386 și 80486, au permis pătrunderea PC-urilor în domenii care pînă acum erau rezervate altor tipuri de calculatoare. Un rol important l-ar putea juca rapidele și puternicele 80386 în cazul locurilor de muncă individuale din domeniile DTP și CAD. Achiziția unui PC din această categorie nu depinde numai de puterea de calcul a CPU ci și de puterea mediului soft și de echipamentele de intrare/ieșire de calitate, compatibile 100% IBM. Instrumentul la care nici un utilizator DTP, și nici un grafician, nu poate renunța este digitizorul. El mijlocește cea mai ergonomică posibilă colectare și introducere a ilustrațiilor complexe și a construcțiilor. Cei care nu sînt utilizatori DTP sînt de părere adesea că digitizoarele de departe lucrează după același principiu de funcționare ca un mouse. Valabilitatea acestei presupunerii se limitează doar la faptul că un digitizor poate emula cele mai răspîndite tipuri de mouse. Deosebirea fundamentală între modul de lucru relativ al unui mouse și modul de lucru absolut al unui digitizor este altul. În modul de lucru relativ, denumit și mod mouse, poziția "lupei" sau a "creionului" pe digitizor nu corespunde cu punctul corespunzător al cursorului de pe ecran. Indiferent din ce poziție de pe digitizor se reia desenul, cursorul este pus în mișcare de la ultima sa poziție. În modul de lucru absolut, coordonatele elementului de deservire (lupă sau creion) de pe digitizor corespund cu poziția actuală a cursorului de pe monitor. Dacă lupa sau creionul sînt ridicate de pe digitizor și puse într-un alt

loc, cursorul de pe monitor face același salt.

Prin intermediul driverelor adaptate la diferite programe de grafică sau DTP, lucrul devine mult mai exact decît cu un mouse. Digitizoarele sînt deci o categorie specială de instrumente de intrare. Între tehnologia și ergonomia fiecărei tablete în parte există deosebiri evidente.

În exterior ele se deosebesc doar prin design, dimensiuni și greutate. Sistemul folosit de ele decide în care domeniu poate fi folosit eficient fiecare digitizor în parte. Pentru exemplificare am comparat modul de utilizare în DTP și CAD pentru două tipuri de digitizoare: Summasketch II, produs de Summagraphics, și SD-332, produs de Wacom.

Cumpărătorul potențial trebuie să se gîndească bine dacă prețul scăzut este un criteriu în alegerea sa. Digitizorul Summasketch este mai ieftin, dar este în mod evident și mai puțin performant tehnic decît digitizorul Wacom. Datorită tehnicii sale fără cablu și așa-zisei tehnologii "Give and Receive Resonance", Wacom deține un avans inovativ. Tocmai în domeniile în care creativitatea utilizatorului trebuie să se desfășoare nestingherită, Wacom permite un lucru optim. În definitiv

nu degeaba sînt folosite digitizoarele Wacom în studiourile de animație ale industriei americane de film.

În domeniul CAD se utilizează mai ales digitizoare Summasketch. Ele se caracterizează printr-o flexibilitate mare și o deservire simplă. Cele mai importante drivere oferite sînt cele pentru MS-Windows, ADI pentru Autocad și Autosketch, și un emulator de mouse. Ca accesoriu de serie se livrează un creion și o lupă cu patru taste (la cerere poate fi livrată și o lupă cu 16 taste). Digitizorul Summasketch se conectează prin cablu la interfața serială.

Manualul de prezentare oferă, într-o formă concentrată, toate informațiile necesare unei instalări cu succes. Tehnologia folosită de Summasketch funcționează după principiul electromagnetic, numit și de inducție. Undele transmise de elementele de introducere cu o anumită frecvență sînt recepționate, controlate și retransmise calculatorului de către digitizor.

Digitizorul Summagraphics lucrează, din acest punct de vedere, cu totul altfel decît digitizorul Wacom. Lupa sau creionul au doar rol de emițător, tabela fiind cea care recepționează, comandă și retransmite informațiile. Rezoluția și

	SummaSketch II	Wacom SD-322
Dimensiuni	406X432 mm	528X526 mm
Suprafața activă	297X297 mm	381X381 mm
Rezoluția	0.025 mm	0.02 mm
Precizia	+/- 0.375 mm	+/- 0.25 mm
Driver	ADI pt. Autocad MS-Windows 2.11, MS-Mouse	ADI pt. Autocad MS-Windows 3.0, MS-Mouse
Moduri de lucru	Point, Remote Request Stream, Switch Stream	Point, Switch Stream, Stream, Suppressed, Increment
Sensibilitate	13 mm distanță	8 mm distanță
Rata de explorare	116 perechi de coordonate pe secundă	1 pînă la 200 perechi de coordonate pe secundă
Formate de ieșire	MM/SummaSketch, UIOF	ASCII/Binar
Accesorii de serie	creion cu două taste lupă cu patru taste	creion sau lupă cu patru taste
Preț	cca. 1500 DM	cca. 4100 DM

pasul pot fi comandate prin calculator cu o secvență de comenzi. Și alegerea modurilor de lucru (Point, Remote, Request, Stream sau Switch Stream) se poate face cu ajutorul unui utilitar aflat pe discheta din furnitura de livrare. Suprafața activă de lucru este de 297 mm X 297 mm (deci format DIN A4). Rezoluția este de 10, 20 sau 40 de rînduri/mm și poate fi aleasă în funcție de cerințele fiecărui monitor în parte. Precizia senzorilor din cîmpul activ de lucru este de +/- 0,375 mm.

Digitizorul Summasketch este cu ceva mai subțire și mai ușor decît digitizorul Wacom. Ergonomia este bună. În interiorul domeniului de lucru activ cablul lupei nu are efecte de îngîrdire. La folosirea creionului cablul poate deranja însă.

Digitizorul Wacom se prezintă diferit deja prin forma sa exterioară și prin greutate. Wacom A4+ este cu 4 kg mai greu și mult mai robust decît concurentul său Summasketch II.

Digitizorul Wacom folosește tehnologia electromagnetică Give-and-Receive-Resonance, care elimină necesitatea cablului. Utilizatorul poate lucra în domeniul de lucru fără restricții și cu precizie mare. Pentru a obține acest spațiu liber

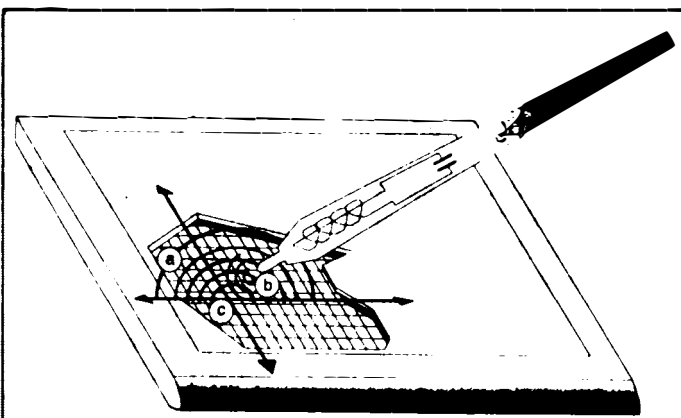
de mișcare, Wacom folosește o tehnologie care schimbă la fiecare 20 de microsecunde distribuția rolurilor dintre digitizor și creion, (sau lupă), ca emițător/receptor.

În rolul de emițător digitizorul emite unde electromagnetice de o anumită frecvență. Ele sînt recepționate de un circuit de rezonanță paralel aflat în creion. Tensiunea indusă în bobina circuitului este înmagazinată într-un condensator legat în paralel cu bobina. Energia înmagazinată aici este folosită de creion pentru a retransmite undele pe o altă frecvență digitizorului. Digitizorul

poate localiza deci, în modul de lucru receptor, undele transmise de creion și-i poate stabili poziția. Procedeele acesta de comutare permanentă emițător/receptor este o particularitate a digitizorului Wacom care permite lucrul fără cablu. O altă caracteristică care derivă din acest procedeu este posibilitatea de a produce linii mai groase sau mai subțiri în funcție de cît de tare este apăsătorul pe suprafața digitizorului. Presiunea exercitată asupra creionului modifică frecvența de rezonanță a circuitului paralel din creion și digitizorul poate sesiza dacă s-a apăsătorul mai tare sau mai

sînt: GSL, ADI pentru Autocad, un driver pentru Windows 3.0 și un driver de mouse Microsoft. Configurarea digitizorului poate fi făcută prin intermediul blocului de microîntrerupătoare 24DIP de pe spatele tabletei, care permite setul de comenzi Bid-Pad-Two proprii Wacom sau setul de comenzi mouse Microsoft. Dischetele din furnitura de livrare conțin pe lângă drivere și cîteva utilitare importante. Cu ele pot fi testate, de ex., conexiunile.

Cu ajutorul softului se poate alege modul de lucru relativ sau absolut și în plus poate fi aleasă unitatea



Prezentare schematică a modului de funcționare al digitizorului Wacom:

- în mod emisie, digitizorul emite unde electromagnetice
- creionul emite pe o altă frecvență spre digitizor
- rețeaua fină de conductori a digitizorului înregistrează și localizează poziția și tăria cu care este apăsătorul creionul

încet. În acest mod poate fi simulată și folosirea unei pensule (la unele programe de pictură), digitizorul Wacom putînd fi folosit și în scopuri pur artistice.

Digitizorul este conectat printr-un conector SUB-D cu nouă pini la interfața serială. Domeniul de lucru activ în formatul DIN A4 este de 30 cm X 30 cm. În domeniul activ de lucru al modelului SD322 a fost integrată o placă detașabilă din sticlă acrilică. Sub această placă pot fi așezate folii de meniuri, suporturi sau planuri de comutație, ceea ce simplifică simțitor transferul desenelor și prelucrarea lor cu diferite programe. Driverele oferite

de măsură, rata de transfer și rezoluția. Digitizorul Wacom atinge o rezoluție de 0,02 mm și recepționează semnale emise de la 8 mm de lupă și de la 4 mm de creion. Precizia este de maxim +/- 0,25 mm. Digitizorul SD-322 poate folosi unul din modulele de operare: Point, Stream sau Switch Stream.

În domeniul CAD ambele tipuri de digitizoare au performanțe comparabile. Din punct de vedere al prețului digitizorul Summasketch este mai avantajos. Digitizorul Wacom are ca avantaje prelucrarea precisă și robustea. În cazul în care

digitizorul este folosit cu programe DTP sau de pictură, tehnica mai avansată a digitizorului Wacom nu poate decît să convingă. Pe suprafața de lucru utilizatorul poate lucra în mod asemănător pictării sau desenării libere pe un carton. Elementele de introducere fără cablu prezintă caracteristici ergonomice perfecte. Caracteristicile de bază ale digitizorului Wacom sînt calitatea înaltă de prelucrare și tehnica optimizată.

(PC+Technik 1/91,
Reiner König)

Să nu uităm CP/M-ul...

Programul DX

Doream să aduc de pe dischete CP/M în harddisk-ul PC-ului niște fișiere. Nu reușeam să fac acest lucru cu ajutorul legăturii seriale din mai multe motive. Întrucât dischetele erau de 5 1/4 inch (80 piste, dublă față, 720 kbyte) provenind de la CUB-Z, mi-am propus și am realizat un program pe PC care să lucreze cu astfel de dischete (citire, scriere, rezumat, ștergere).

Acest program s-a dovedit ulterior mult mai comod decât transmisia serială (cu Kermit, Move-it etc.), devenind o componentă deosebit de utilă în condițiile prelucrărilor distribuite, pe CUB-Z și PC.

Programul are și unele deficiențe, ușor de ocolit însă:

- PC-ul trebuie să "vadă" dischete de 720 kbyte. Precizez că orice AT care "vede" 1,2 Mbyte poate fi făcut "să vadă" 720 kbyte instalând un program adecvat.
- primele două piste ale dischetei CP/M trebuie să fie recunoscute de MS-DOS la funcția DIR. Acest lucru se realizează punând sistem (cu SYSGEN, APPGEN, 5GEN) de pe o dischetă goală formatată pe PC (80 piste, 9 sectoare pe pistă).

ing. Sava Stan, Buzău

```
#include < dos.h >
#include < conio.h >
#include < stdio.h >
#include < stdlib.h >
#include < stdarg.h >
#include < io.h >
#include <fcntl.h >
```

```
#define CITIRE 1
#define SCRIERE 2
```

```
void beep (void);
void bloc_dx ();
void rezumat_disket (void);
void afisrez (void);
```

```
short numar_disket;
short abscisa, ordonata;
short i,j,k,l,numar_fisiere, inceput;
short nf, nrf;
unsigned char c, user = 0;
unsigned char buf [2048];
unsigned char blocuri [360];
unsigned char fisiere [128][12];
unsigned char sort [128];
unsigned char il,i2;
unsigned char art [80];
unsigned char numefis[30], numf [12];
```

```
unsigned short numar_sectoare [128];
unsigned short block [351];
unsigned short nb, nrbl, nrsect;
char nnext;
long lung, kk;
```

```
void main ()
```

```
{
    unsigned char functia;
    clrscr ();
    printf("\n=====");
    printf("\n|          Program DX          |");
    printf("\n| Conversie fișiere disketa CP/M |");
    printf("\n| 5,25 inch, 80 piste, 2 fete    |");
    printf("\n|      in/din fișiere MS-DOS     |");
    printf("\n| Autor: ing. Stan Sava 02-sep-90 |");
    printf("\n| 'Sava Soft', Buzău, tel. 974/12760 |");
    printf("\n=====");
    numar_disket = 0;
    i = system ("DIR A:>NULL");
    rezumat_disket ();
    functia = ' ';
    while (functia == ' ')
    {
        clrscr ();
        printf("\n R = Rezumat disketa");
        printf("\n S = Scriere in disketa");
        printf("\n C = Citire din disketa");
        printf("\n D = Stergere fisier in disketa");
        printf("\n T = Terminare program \n\n");
        abscisa = wherex ();
        ordonata = wherey ();
        c = ' ';
        while (c == ' ')
        {
            gotoxy (abscisa, ordonata);
            printf ("Functie (R,S,C,D,T): ");
            gotoxy (abscisa + 19, ordonata);
            if ((c = getch ()) == '\0')
            {
                getch ();
                c = ' ';
                continue;
            }
            putchar (c);
            switch (c)
            {
                case 'R':
                case 'r':
                    afisrez ();
                    if (getch () == '\0') getch();
                    c = '1';
                    functia = ' ';
                    break;
                case 'C':
                case 'c':
                    afisrez ();
                    if (numar_fisiere == 0)

```

```

{
    printf ("\Disc gol");
    beep ();
    exit (0);
}
printf ("\nNumar fisier:");
abscisa = wherex ();
ordonata = wherey ();
nrf = 0;
while (nrf == 0)
{
    gotoxy (abscisa, ordonata);
    printf (" ");
    gotoxy (abscisa, ordonata);
    buf [0] = 4;
    if (numar_fisiere < 100)
        buf [0] = 3;
    if (numar_fisiere < 10)
        buf [0] = 2;
    cgets (buf);
    if (buf [1] == 0)
    {
        nrf = -1;
        c = '1';
        break;
    }
    nrf = atoi (buf + 2);
    if (nrf < 1 || nrf > numar_fisiere)
    {
        beep ();
        nrf = 0;
    }
}
if (nrf == -1)
{
    functia = '';
    break;
}
i1 = sort [nrf-1];
for (i=0; i<8;
    numefis [i] = fisiere [i1] [i + +]);
numefis [8] = '.';
for (i=8; i<11;
    numefis[i+1] = fisiere [i1] [i + +]);
numefis [12] = '\0';
if ((nf = open (numefis,
    O_BINARY|O_RDWR)) < 5)
    nf = _creat (numefis, FA_ARCH);
k = numar_sectoare [i1] + 1;
for (i=0; i<360; i+ +)
{
    if (blocuri [i] != (i1 + 1))
        continue;
    bloc_dx (numar_disket, i, buf, CITIRE);
    for (j=0; j<16; j+ +)
    {
        k--;
        if (k == 0) break;
        l = writc (nf, &buf [j*128], 128);
    }
}
close (nf);
functia = '';
c = '1';
break;

case 'S':
case 's':
    clrscr ();
    printf ("\nNume fisier:");
    abscisa = wherex ();
    ordonata = wherey ();
    nrf = 0;
    while (nrf == 0)
    {
        functia = '1';
        gotoxy (abscisa, ordonata);
        printf (" ");
        gotoxy (abscisa, ordonata);
        buf [0] = 13;
        cgets (buf);
        if (buf [1] == '\0')
        {
            functia = '';
            c = '1';
            break;
        }
        strcpy (numefis, buf + 2);
        for (i=0; i<11; numf [i + +] = ' ');
       strupr (numefis);
        j = 0;
        for (i=0; i<12; i+ +)
        {
            if (numefis [i] == '\0') break;
            if (numefis [i] == '.')
            {
                j = 8;
                continue;
            }
            numf [j + +] = numefis [i];
        }
        for (i=0; i<numar_fisiere; i+ +);
        {
            for (j=0; j<11; j+ +)
                if (fisiere [i] [j] != numf [j]) break;
            if (j == 11)
            {
                printf ("\nFisier deja prezent");
                beep ();
                functia = '';
                c = '1';
                break;
            }
        }
    }
    if (functia == '') break;
    if ((nf = open (numefis,
        O_BINARY|O_RDONLY)) < 5)
    {
        printf ("\nFisier %s absent", numefis);
        beep ();
        functia = '';
        c = '1';
        break;
    }
    if (functia != '') nrf = 1;
}
if (functia == '') break;
lung = filelength (nf);
nrsect = (lung + 127) / 128;
nrbl = (nrsect + 15) / 16;
j = 0;

```

```

for (i = 2; i < 351; i + +)
if (blocuri [i] == '\0') j + +;
if (j < nrbl)
{
printf ("\nFisier prea mare");
close (nf);
functia = '';
c = '1';
break;
}
nb = 0;
for (i = 2; i < 351; i + +)
{
if (blocuri [i] != '\0') continue;
block [nb + +] = i;
if (nb == nrbl) break;
}
nnext = -1;
nb = 0;
for (j = 0; j < 2; j + +)
{
if (nb >= nrbl) break;
bloc_dx (numar_disket, j, buf, CITIRE);
for (i = 0; i < 64; i + +)
{
if (nb >= nrbl) break;
inceput = i * 32;
if (buf [inceput] == user) continue;
buf [inceput] = user;
nnext + +;
for (l = 0; l < 11; l + +)
buf [inceput + l + 1] = numf [l];
buf [inceput + 12] = nnext;
buf [inceput + 13] =
buf [inceput + 14] = '\0';
k = nrsect - 128 * nnext;
if (k > 128) k = 128;
buf [inceput + 15] = k;
for (l = 0; l < 8; l + +)
{
if (nb >= nrbl)
buf [inceput + 16 + l + 1] =
buf [inceput + 17 + l + 1] = '\0';
else
{
buf [inceput + 16 + l + 1] =
block [nb] % 256;
buf [inceput + 17 + l + 1] =
block [nb + +] / 256;
}
}
}
}
bloc_dx (numar_disket, j, buf, SCRIERE);
}
for (i = 0; i < nrbl; i + +)
{
kk = 2048;
if (kk > lung) kk = lung;
lung = kk;
read (nf, buf, kk);
while (kk < 2048) buf [kk + +] = 0x1a;
bloc_dx (numar_disket, block [i], buf, SCRIERE);
}
close (nf);
rezumat_disket ();

```

```

functia = '';
c = '1';
break;
case 'D':
case 'd':
afisrez ();
printf ("\nNumar fisier:");
abscisa = wherex ();
ordonata = wherey ();
nrf = 0;
while (nrf == 0)
{
gotoxy (abscisa, ordonata);
printf (" ");
gotoxy (abscisa, ordonata);
buf [0] = 4;
if (numar_fisiere < 100) buf [0] = 3;
if (numar_fisiere < 100) buf [0] = 2;
cgets (buf);
if (buf [1] == 0)
{
nrf = -1;
break;
}
nrf = atoi (buf + 2);
if (nrf < 1 || nrf > numar_fisiere)
{
beep ();
nrf = 0;
}
}
if (nrf == -1)
{
functia = '';
c = '1';
break;
}
j = sort [nrf - 1];
for (i = 0; i < 11; numf [i] = fisiere [j] [i + +]);
k = 0;
for (j = 0; j < 2; j + +)
{
bloc_dx (numar_disket, j, buf, CITIRE);
for (i = 0; i < 64; i + +)
{
inceput = i * 32;
if (buf [inceput] != user) continue;
buf [inceput + 9] &= 0x7f;
for (l = 0; l < 11; l + +)
{
if (numf [l] != buf [inceput + l + 1])
break;
}
}
if (l == 11)
{
k = 1;
buf [inceput] = 0xe5;
}
}
if (k == 1)
bloc_dx (numar_disket, j, buf, SCRIERE);
}
rezumat_disket ();
functia = '';

```

```

    c = '1';
    break;
case 'T':
case 't':
    exit (0);
}
}
}

void afisrez ()
{
    clrscr ();
    k = (numar_fisiere + 2) / 3;
    for (i=0; i < k; i++)
    {
        for (i2=0; i2 < 3; i2++)
        {
            l = k * i2 + i;
            if (l >= numar_fisiere) continue;
            i1 = sort [l];
            for (j=0; j < 8; numf[j] = fisiere [i1][j + +]);
            for (j=8; j < 11; numf[j + 1] = fisiere [i1][j + +]);
            numf [8] = ':';
            numf [12] = '\0';
            if (i2 == 0) printf ("\n");
            printf ("%02d.%s %4d", l + 1, numf,
                numar_sectoare [i1]);
            if (i2 != 2) printf (" ");
        }
        if (i == 24)
            if (getch () == 0) getch ();
    }
}

void rezumat_disket (void)
{
    unsigned short cc;

    for (i=0; i < 360; blocuri[i + +] = '\0');
    numar_fisiere = 0;
    for (j=0; j < 2; j++)
    {
        bloc_dx (numar_disket, j, buf, CITIRE);
        for (i=0; i < 64; i++)
        {
            inceput = i * 32;
            if (buf [inceput] != user) continue;
            buf [inceput + 9] &= 0x7f;
            buf [inceput + 10] &= 0x7f;
            buf [inceput + 11] &= 0x7f;
            for (k=0; k < numar_fisiere; k++)
            {
                for (l=0; l < 11; l++)
                {
                    if (buf [inceput + l + 1] != fisiere [k][l])
                        break;
                    if (l == 11) break;
                }
                if (k == numar_fisiere)
                {
                    numar_fisiere++;
                    for (l=0; l < 11; l++)
                        fisiere [k][l] = buf [inceput + l + 1];
                    numar_sectoare [k] = 0;
                }
            }
        }
    }
}

    numar_sectoare [k] += buf [inceput + 15];
    for (l=16; l < 32; l += 2)
    {
        cc = buf [inceput + l] +
            buf [inceput + l + 1] * 256;
        if (cc == 0) break;
        blocuri [cc] = k + 1;
    }
}

for (i=0; i < numar_fisiere; sort[i] = i + +);
for (i=0; i < numar_fisiere; i++)
for (j=i + 1; j < numar_fisiere; j++)
{
    i1 = sort [i];
    i2 = sort [j];
    for (l=0; l < i1; l++)
    {
        if (fisiere [i1][l] < fisiere [i2][l]) break;
        if (fisiere [i1][l] == fisiere [i2][l]) continue;
        sort [i] = i2;
        sort [j] = i1;
        break;
    }
}

void beep (void)
{
    printf ("%c", 7);
}

void
bloc_dx (numar_disc, numar_bloc, buffer, operatia)
short numar_bloc, numar_disc,
operatia;
unsigned char buffer [];
{
    unsigned char blocuri [36] =
        { 0, 2, 4, 6, 8, 1, 3, 5, 7,
          9, 11, 13, 15, 17, 10, 12, 14, 16, 18,
          20, 22, 24, 26, 19, 21, 23, 25, 27, 29,
          31, 33, 35, 28, 30, 32, 34 };
    short j1, j2, k, adresa_sector;

    j1 = numar_bloc / 9 + 1;
    j1 *= 36;
    j2 = numar_bloc % 9;
    for (k=0; k < 4; k++)
    {
        adresa_sector = blocuri [j2 * 4 + k] + j1;
        if (operatia == CITIRE)
        {
            disable ();
            absread (numar_disc, 1,
                adresa_sector, buffer + k * 512);
            enable ();
        }
        if (operatia == SCRIERE)
            abswrite (numar_disc, 1,
                adresa_sector, buffer + k * 512);
    }
}

```

Sfaturi practice pentru întreținerea harddisk-urilor

Discurile hard sint foarte utile atît timp cît funcționează. Acest articol vă dă cîteva sfaturi pentru depanarea lor.

Odată și odată în viața oricărui utilizator vine un moment în care dorește să încheie lucrarea, care l-a costat multă osteneală în ultimele luni. O intenție laudabilă, numai că se pare că în chiar acel moment PC-ul a intrat în grevă. Apar mesaje de eroare, sub forma unor cifre ne-definite, sau mesaje de genul:

"Drive not ready" sau

"Boot Disk failure" care produc panică în rîndul utilizatorilor. Cel mai tîrziu în acel moment va apare părerea de rău, că ultima salvare a fost făcută cu multă vreme în urmă. Dacă nici "mîngîierea" calculatorului și nici alte metode promițătoare nu ajută la nimic, atunci doar cineva care știe ce este defect mai poate ajuta. Trebuie știut însă un lucru: în 90% din cazuri nimic nu este defect.

Prima diagnoză

Calculatorul d-voastră nu mai recunoaște dintr-o dată harddisk-ul, sau nu mai vrea să extragă datele memorate? Ar trebui să verificați că într-adevăr harddisk-ul este stricat, și deci că pana este hard.

- Puteți stabili acest lucru fiind atenți la zgomotul pe care calculatorul d-voastră îl face atunci cînd este pus în funcțiune. Harddisk-urile produc zgomote specifice la pornire. Dacă zumzetul obișnuit al discului nu se face auzit, atunci cel mai probabil este că sînteți ghinionist și că discul este într-adevăr defect.

- Observați cu atenție ecranul în timpul autotestului de la pornirea PC-ului. Dacă eroarea este recunoscută de către sistem atunci ea va fi anunțată printr-un țuit

sau prin apariția unui cod de eroare pe ecran, pe care îl puteți descifra folosind un tabel de coduri de eroare.

- Dacă autotestul dovedește în mod clar existența unui defect pe controller-ul harddisk-ului (unitatea de comandă a harddisk-ului), atunci acesta va trebui deparat într-o unitate specializată. Celelalte probleme pot fi rezolvate de obicei prin programe speciale.

Salvarea datelor pe dischete

Un harddisk defect își anunță din timp iminența prăbușirii. Dacă observați în timpul lucrului erori tot mai dese la citirea sau scrierea pe disc, este bine să salvați cît mai repede datele pe dischete, metodă denumită "backup pe dischete".

Cel mai bine este să efectuați un backup complet pe dischete. Pentru aceasta aveți la dispoziție comenzile DOS "Backup" și "Restore" cît și o serie de programe specializate cum ar fi "Corefast" al firmei Core International sau "Fastback Plus" al firmei Fifth Generation Systems.

Dacă harddisk-ul și-a făcut datarea fără să cricnească și fără zgomote suspecte pînă în "ziua Z", atunci problema va putea fi rezolvată cu programe ajutătoare. Pentru aceasta aveți nevoie de o dischetă sistem sau de una de diagnosticare. Nu este util să țineți programele ajutătoare pe harddisk, căci cum le-ați putea folosi dacă nu puteți porni discul?

Primul ajutor

Introduceți discheta sistem în unitatea A: și porniți din nou calculatorul. Probabil va urma același mesaj de eroare și veți fi solicitat să apăsați o tastă. Urmați această so-

licitare și așteptați pînă cînd calculatorul se va anunța cu ecranul obișnuit de start (așa-numitul prompter sistem). Încercați să accesați acum harddisk-ul. Introduceți următoarea comandă "C: <Enter>". Dacă acest lucru funcționează fără probleme puteți să vă destindeți pentru moment, datele încă mai pot fi salvate.

- Afișați apoi conținutul harddisk-ului cu comanda "DIR"

- Verificați că afișarea fișierelor se efectuează ca și pînă acum și fiți atenți în mod special dacă interpretorul de comenzi al sistemului de operare MS-DOS "COMMAND.COM" apare în această listă. Dacă acesta lipsește schimbați din nou unitatea pe A: și transferați sistemul pe harddisk cu comanda:

"SYS C: <Enter>".

- Apoi copiați fișierul "COMMAND.COM" de pe discheta sistem pe harddisk

- Scoateți discheta din unitate și porniți din nou calculatorul.

Ultima salvare

Dacă încercarea de a schimba unitatea pe C: eșuează cu mesajul de eroare: "Invalid drive specification", atunci executați programul "FDISK" de pe discheta sistem. FDISK este un program standard MS-DOS, destinat administrării harddisk-ului. Mare atenție însă, cu acest program un necunosător poate distruge foarte ușor datele!

- La apariția meniului introductiv al lui FDISK alegeți punctul 4 al meniului, referitor la afișarea partițiilor de pe disc (partiție: domeniu definit pe harddisk pentru sistemul de operare și pentru date). Acest punct al programului folosește la verificarea ocupării ac-

tuale a harddisk-ului. Dacă obțineți un mesaj de genul "partiție nedefinită", atunci nu mai puteți face nimic cu FDISK. Terminați lucrul cu acest program și citiți în continuare.

- Când vă este afișat tabelul partițiilor, verificați dacă partiția "C:" este activă. Acest lucru se întâmplă atunci când în coloana **Status** este trecut un "A". Dacă nu se întâmplă așa, alegeți din meniul principal punctul 2 (activare partiții). Activați partiția tastând un "1". Terminați programul FDISK și pomiți din nou calculatorul. Înainte de acest lucru nu uitați să introduceți în unitatea A: o dischetă sistem. Dacă partiția C: este activă, puteți accesa harddisk-ul. Salvați-vă acum datele și apoi reformatați harddisk-ul cu un format de nivel înalt ("High-Level"). Discul va fi apoi din nou gol și pregătit de lucru.

Doar pentru posesorii de AT-uri: suspendarea SETUP-ului

Posesorii de AT-uri mai au încă o șansă, în comparație cu XT-ul el posedă un spațiu de memorie alimentat printr-o baterie, în care informațiile rămân memorate și după decuplarea calculatorului. Acest domeniu se numește "NVR" (Non Volatile RAM).

În acest spațiu de memorie se memorează datele despre configurația calculatorului. Dintre informațiile stocate aici enumerăm: tipul discului instalat în sistem, numărul de capete de scriere/citire și numărul de unități din care este format harddisk-ul. Se poate întâmpla ca informațiile memorate în NVR să se piardă, fie datorită unei încărcări statice, fie datorită descărcării bateriei de alimentare.

- Presupunând că informațiile despre configurația calculatorului au fost deteriorate de o încărcare statică se poate întâmpla, de ex. ca sistemul să creadă că dispune de un harddisk de 40 MBytes și în realitate să dispună doar de

unul de 20 MBytes. Un harddisk de 40 MBytes posedă mai multe capete de scriere/citire decât un disc mai mic și încercarea de inițializare a harddisk-ului va da greș cel mai târziu în momentul în care se va încerca comandarea capetelor de scriere/citire care de fapt nu există. Urmarea: calculatorul intră în grevă.

- Și mai rău este atunci când datorită unei erori în NVR va fi luat în considerare un harddisk de o capacitate mai mică decât cea existentă în realitate. Calculatorul pornește normal și se oprește brusc fără motiv. Singurul remediu în acest caz este modificarea informațiilor din NVR. Aceasta se poate face cu ajutorul programului SETUP. Acesta poate fi lansat fie de pe o dischetă fie de pe harddisk. El poate fi activat în timpul procesului de inițializare prin apăsarea unei anumite combinații de taste, de obicei: < Alt T > , sau < Del > sau < Ctrl PrintScreen > .

- Dacă AT-ul sesizează combinația de taste introdusă, este activată rutina SETUP și calculatorul va putea fi configurat din nou. Acest lucru se poate face de obicei sub conducerea unui meniu. După ce ați configurat calculatorul va trebui să opriți programul. Calculatorul va efectua automat un nou start cu valorile actualizate. Dacă în acest moment calculatorul d-voastră se comportă normal puteți fi mândru, ați reușit să faceți o reparație excelentă.

Controlul hardware-ului

Dacă n-ați ajuns la nici un rezultat cu programele SYS, FDISK sau cu SETUP, trebuie să vă obișnuți cu ideea că va trebui să desfaceți calculatorul, pentru a controla conectorii. Dar nu uitați că dacă faceți acest lucru în perioada de garanție, pierdeți dreptul la garanție!

Efectuați următoarele operații:

- scoateți ștecherul din priză
- deschideți calculatorul, de obicei

trebuie desfăcute câteva șuruburi cu cap în cruce.

- localizați harddisk-ul. Este ușor de depistat fiind singurul component la care ajung trei cabluri. Două dintre ele sînt tip bandă lată și sînt conectate unul lîngă celălalt. Cel de-al treilea este cablul de alimentare al harddisk-ului și are patru fire colorate: unul roșu, unul galben și două fire negre.
- verificați mai întii dacă firele cablului de alimentare sînt corect introduse în conector
- testați ștecherile celor două cabluri tip bandă lată să fie bine fixate la ambele capete (atît la harddisk cît și la controller)
- mai mult nu puteți face aici. Asamblați calculatorul din nou și mai încercați odată pașii enumerați la "primul ajutor".

Cînd nu mai ajuta nimic: formatare

Dacă aparțineți acelei categorii de utilizatori, căreia nici una dintre metodele descrise anterior nu i-a ajutat, atunci vă rămîne ca ultimă șansă doar reformatarea discului, toate datele deținute fiind pierdute în acest caz. În mod normal un harddisk se formatează o singură dată, în momentul instalării. Această operație, mare consumatoare de timp, face harddisk-ul funcțional. Ea servește la asigurarea schimbului de date dintre harddisk și controller pe de o parte, și pe de altă parte dintre controller și sistemul de operare.

Pentru ca schimbul de date dintre cele trei unități să fie posibil, sînt necesare două formatare ale harddisk-ului: o formatare "Low Level" care asigură comunicația dintre controller și disc și o formatare "High Level" care asigură accesul sistemului de operare pe harddisk.

Pentru formatarea High Level harddisk-ul trebuie să fie deja formatat Low Level și la pornire nu are voie să dea mesaje de eroare refe-

ritoare la harddisk. Cum acest lucru nu este valabil, în cazul exemplului nostru, trebuie să începem mai întâi cu o formatare "Low Level". În această privință între XT și AT există o diferență esențială. De aceea vom descrie separat rutinele de formatare pentru cele două echipamente.

Formatarea Low Level la XT-uri

Controller-ul de harddisk la XT-uri este altfel construit decât la AT-uri. Atunci când pe piață au apărut primele PC-uri, nimeni nu s-a gândit că într-o bună zi prețurile harddisk-urilor și a altor memorii va scădea atât de tare încât fiecare calculator va putea fi dotat cu un harddisk. Din acest motiv PC/XT-ul nu are o comandă proprie, "on Board", pentru harddisk (deci cuprinsă în BIOS-ul propriu). Harddisk-ul va fi comandat prin controller, de aceea toată munca de formatare trebuie executată de acesta.

Ce este de făcut:

- porniți calculatorul având în unitatea A: discheta sistem DOS și așteptați apariția prompter-ului.
- porniți programul sistem "Debug". Atunci când "Debug" este încărcat și așteaptă datele d-voastră, la începutul rîndului va apare semnul "-". Introduceți următorul rînd (atenție în acest moment se pierd toate informațiile de pe harddisk !): "G=c800:5". Folosiți această comandă doar atunci când doriți în mod cert o formatare nouă.
- în continuare apăsați pe tasta "R". Veți fi rugați să introduceți diferite valori dependente de controller. Dacă aveți posibilitatea de a alege o opțiune implicită, faceți-o, ea ușurează munca unui începător.
- înainte sau după formatare vi se va cere să precizați sectoarele defecte (bad tracks). Acestea sînt zone defecte pe disc, apărute încă din fabricație. O listă a sectoarelor defecte este lipită de obicei pe harddisk. Introduceți neapărat aceste sectoare defec-

te, altfel vor putea apărea erori în date.

- în continuare va trebui să introduceți factorul "interleave". Explicarea acestei noțiuni nu intră în tema pe care și-o propune acest articol (pentru informații suplimentare consultați "if" 1/90), dar atenție: un factor interleave greșit introdus poate reduce foarte mult viteza de lucru a calculatorului. La un XT ar trebui ales un factor interleave cuprins între 3 și 5, în funcție de viteza calculatorului. Cu cît calculatorul este mai lent cu atît factorul interleave trebuie să fie mai mare. Dacă formatarea Low Level a decurs normal, atunci v-ați atins deja scopul pe jumătate.

Formatarea Low Level la AT-uri

În cazul AT-urilor, rutinele necesare pentru harddisk sînt gata înglobate, de aceea controller-ul harddisk-ului unui AT nu are "inteligentă" proprie, cum este cazul la XT-uri. În principiu pentru formatarea Low Level a harddisk-ului unui AT este necesar un software special. Unii producători de harddisk-uri oferă acest soft special pentru formatare odată cu produsul. De ex. discurile firmei Seagate sînt livrate împreună cu programul "Disk Manager" al firmei Ontrack. Unele discuri ale producătorului Maxtor sînt livrate împreună cu programul "Speedstore".

Crearea unei partiții cu FDISK

Dacă formatarea de bază s-a derulat cu succes, porniți calculatorul cu o dischetă sistem în unitate și încărcăți programul FDISK

- alegeți din meniu punctul 1 (crearea unei partiții DOS sau a unei unități logice)
- în continuare alegeți din nou punctul 1 (crearea unei partiții DOS primare)

- răspundeți afirmativ la întrebarea dacă doriți să puneți la dispoziția partiției DOS primare întregul spațiu de memorie

- FDISK vă va informa că sistemul va trebui pornit din nou. Acționați o tastă și așteptați pînă cînd calculatorul se va prezenta ca de obicei

Formatarea High Level

Pînă în acest moment a fost efectuată formatarea Low Level pentru ca harddisk-ul și controller-ul să poată comunica între ele. A urmat crearea unei partiții, deci a acelui domeniu de disc pe care vor fi memorate datele. Această partiție trebuie formatată din nou pentru a putea fi folosită de sistemul de operare DOS. Această formatare High Level se face cu ajutorul programului DOS FORMAT.

Pentru aceasta:

- porniți programul FORMAT de pe discheta d-voastră sistem, introducînd comanda: "FORMAT C: /S". Parametrul '/S' comunică programului FORMAT faptul că pe harddisk trebuie de pus un sistem de operare, pentru ca în viitor calculatorul să poată porni și fără dischetă sistem. În acest caz FORMAT face harddisk-ul disc sistem.
- dacă formatarea s-a terminat cu succes, puteți folosi harddisk-ul pentru înmagazinarea datelor și programelor d-voastră. Ar fi de folos dacă ați avea o copie de siguranță a tuturor datelor d-voastră dinainte de a se defecta harddisk-ul, sau de a-l formata. În acest caz ați putea transfera această copie pe harddisk. După cum vedeți există mai multe "trucuri" pentru salvarea informațiilor de pe un harddisk defect. Nu disperați deci atunci cînd apare (din cînd în cînd) cite o eroare la citire.

(Computer Live 1/91,
Joerg Engelsing)

Schlță de post:

Administratorul unei rețele locale

Administratorul rețelei poartă răspunderea pentru rețeaua instalată. Fiind vorba de o meserie nouă, încă nu există o descriere exactă a acestui loc de muncă. În S.U.A. însă, această descriere deja s-a făcut.

Deși "administrator de rețea" este un titlu potrivit pentru cineva care este răspunzător de o rețea locală, acesta este un post "nedefinit". În multe întreprinderi însă, un post de pe stațul de funcțiuni poate fi ocupat abia după ce există o descriere exactă a postului. De-seori, prima activitate a noului administrator constă în formularea unei descrieri a propriului loc de muncă.

Descrierea locului de muncă ar trebui să cuprindă sarcinile tehnice și administrative precum și conținutul pregătirii și devenirea profesională. Prezentăm în continuare un model de astfel de descriere de loc de muncă, așa cum a fost ea concepută în S.U.A. pentru o întreprindere care tocmai și-a instalat prima rețea. Descrierea pornește de la premisa că administratorul de rețea răspunde de selecția, instalarea și funcționarea rețelei locale. Și personalul de întreținere (suport) îi este subordonat.

Cerințe generale

Munca independentă este premisa de bază pentru această poziție. Deoarece profesiunea este încă tânără, administratorul de rețea nu se poate sprijini pe un "tezaur" de experiență existentă. El trebuie să arate siguranță de sine și să aibă inițiativă, pentru a realiza obiectivele cotidiene și cele pe termen lung. El administrează bugetul rețelei; o secție de întreținere (suport) îi este subordonată. Răspunde de planificarea, achiziționarea, instalarea și operarea

unui LAN nou, de modificări la rețele deja existente și de racordarea de LAN-uri la rețeaua globală a întreprinderii prin "gateway"-uri și alte interfețe.

Administratorul de rețea acționează ca interfață între conducere, utilizatori și producători atât în ce privește problemele tehnice cât și cele administrative. Pentru a rămâne "la zi", administratorul de rețea trebuie să se instruiască și să se perfecționeze continuu.

Răspundere globală

Administratorul de rețea este un manager tehnic, care răspunde și de formarea unei secții de întreținere (suport). Până la încadrarea personalului de suport, administratorul preia el serviciile descrise mai jos. După încadrare, răspunderea principală a administratorului constă în planificarea strategică și supravegherea funcționării rețelei.

Activități administrative

Administratorul de rețea cunoaște hard-ul instalat în rețea, sistemul de operare, cablarea și aplicațiile care se folosesc. Menține legătura cu producătorii, "VAR"-i (value added reseller-s - revânzatori a unor produse cărora le-au sporit valoarea adăugând ceva de la ei) și integratori (cei care vînd produse rezultate din integrarea unor subproduse) din zona produselor și serviciilor LAN.

El concepe specificații funcționale pentru hard-ul de rețea, sistemul de operare, cablări, legături spre un host (calculator "gazdă", de obicei de putere mai mare) și aplicații. Valorifică ofertele comercianților și prezintă conducerii concluziile.

Administratorul de rețea stabilește bugetul necesar și justifică

selecția echipamentelor necesare. El alege producătorii și se informează asupra competenței tehnice a acestora, stabilității lor financiare și a calificării de suport-întreținere. Administratorul de rețea comandă hardul și softul selectat. El planifică și coordonează instalarea cablului, a hard-ului de rețea și a sistemului de operare a rețelei.

Activități tehnice

Administratorul de rețea planifică și coordonează întreaga desfășurare a lucrărilor, din faza testărilor și pînă la punerea definitivă în funcțiune. El supraveghează activitățile terților producători și furnizorilor.

Administratorul instalează, supraveghează și modifică organizarea datelor pe server. Modul de organizare a datelor se elaborează în colaborare cu utilizatorii și conducerea întreprinderii. Folosind "unelele" specifice pe care le oferă sistemul de operare al rețelei, administratorul instalează caracteristicile utilizatorilor, parolele, directoarele utilizatorilor și drepturile de acces.

Dacă se dovedește necesară stabilirea unor responsabili de rețea proprii pentru diferite grupe de muncă, atunci administratorul de rețea îi alege pe aceștia și îi pregătește. În afară de asta, este disponibil pentru asistență tehnică a acestor responsabili de echipă. Administratorul de rețea elaborează proceduri pentru integrarea sau eliminarea unor echipe, proiecte sau utilizatori individuali. Funcție de necesități, administratorul poate delega diferite activități înspre responsabilii de echipă și supraveghează executarea acestora.

Administratorul de rețea planifică transferul de date din alte sis-

teme în rețeaua locală. El coordonează instalarea interfețelor de rețea și eventualele legături spre un host.

Administratorul de rețea este răspunzător de instalarea hard-ului. El contractează eventual servicii de întreținere și supraveghează executarea lucrărilor de întreținere. Dacă este necesar, înlocuiește cartele de interfață pentru rețea și conectori. Capacitățile libere în sistem și implementarea unor facilități care să conducă la creșterea performanțelor sînt supravegheate de el.

Administratorul este răspunzător de înmagazinarea și depozitarea echipamentelor și materialelor auxiliare necesare (hîrtie, toner, mine pentru plotter etc.)

Administratorul de rețea este răspunzător de funcționarea cotidiană a rețelei. El elaborează metode pentru recunoașterea și corectarea din timp a unor malfuncții. Poate delega aceste răspunderi asupra personalului de suport. Elaborează o bază de date de configurare.

Administratorul de rețea supraveghează prestațiile serverului cu ajutorul jurnalelor de bord și a unor analize de tendințe. Supraveghează toate modificările la server, la cablare și la alte resurse ale rețelei. El controlează resursele libere și asigură existența unor capacități suficiente pentru aplicații și fișiere noi.

Supravegherea funcționării rețelei

Administratorul de rețea planifică și instalează aplicații noi pe server și îi instruiește pe responsabilii de echipă și utilizatorii finali în utilizarea acestor programe. El testează programe noi, pentru a fi sigur că acestea nu încarcă prea mult rețeaua sau că nu intră în conflict cu alte programe. El elaborează utilitare soft pentru administrarea rețelei, cum ar fi utilitare de imprimare sau fonturi precum și fișiere de comenzi.

Strategia întreprinderii, administratorul stabilește procedurile de asigurare a securității datelor. El planifică și supraveghează salvarea regulată a datelor (backup). Păstrează protocoale ale acestor activități și verifică corectitudinea benzilor pe care se fac salvările. Date deja șterse vor fi restaurate la solicitare de pe copiile de siguranță de către administrator.

Pregătire și suport

Administratorul de rețea elaborează metode și proceduri pentru administrarea rețelei, ținînd seama de strategia întreprinderii. Planifică și elaborează proceduri pentru instruire relativ la rețea. Utilizatorii finali sînt instruiți în permanență. În acest scop, administratorul elaborează, înnoiește și distribuie documente relativ la modificările în rețea.

Administratorul de rețea angajează colaboratori pentru administrarea rețelei și îi instruiește pe aceștia. El elaborează un sistem de asistență tehnică, care este la dispoziția utilizatorilor pentru întrebări.

Dacă mai mulți utilizatori se confruntă cu aceeași problemă, administratorul stabilește dacă este vorba de o funcționare greșită a rețelei sau dacă utilizatorii nu știu să folosească corect aplicația. Corectează problema în rețea sau respectiv îi instruiește pe utilizatori relativ la folosirea softului.

La modificări ale rețelei, administratorul de rețea se îngrijește ca acestea să se facă cu deranjul minim al utilizatorilor. El elaborează un sistem de gestionare a problemelor, pentru a recunoaște timpuriu eventuale malfuncții și pentru a le putea evita. Pentru a avea la dispoziție datele necesare unei recunoașteri timpurii a erorilor, administratorul elaborează un sistem de control a configurației.

Experiența profesională și pregătirea

Pentru această activitate, o premisă este absolvirea unui institut superior informatic. Administratorul trebuie să posede forme de exprimare foarte bune, atît în scris cît și oral.

Se presupune că administratorul de rețea are o experiență de cel puțin doi ani și cunoaște o mare varietate de calculatoare personale. În activitatea anterioară, ar trebui să fi avut deja responsabilități în ceea ce privește hard-ul și soft-ul, suportul unor aplicații precum și instruirea utilizatorilor. Administratorul trebuie să posede cunoștințe profunde despre sistemele de operare folosite în întreprindere. Sînt necesare deasemenea cunoștințe relative la comunicații cu un host și protocoale de comunicație.

Administratorul de rețea ar trebui să aibă 1 pînă la 3 ani de experiență în ce privește responsabilități relative la personal și buget.

Administratorul de rețea este un manager.

Grosul activității sale se află în domeniul administrativ. De aceea, cunoștințe tehnice cuprinzătoare sînt o premisă. Hotărîtoare însă este capacitatea organizatorică și motivațională a administratorului.

De aceea, cînd se elaborează o astfel de descriere de loc de muncă, este important să se sublinieze în mod deosebit activitățile administrative. Căci activitățile tehnice ale administratorului nu sînt totdeauna pe deplin inteligibile pentru conducerea întreprinderii (management). Cheia succesului administratorului rezidă deci în a putea fi considerat de către conducerea întreprinderii ca "unul de-ai noștri", și nu ca un tehnician "pur".

(LAN Magazin 2/90,
Ira S. Hertzoff)

Ira Hertzoff este președinta firmei Tholian Holdings Inc., întreprindere de management și consultanță pentru rețele cu sediul în Columbus, Ohio, S.U.A.

A luat flință

Institutul European contra Virușilor

Experți în viruși ai calculatoarelor din întreaga Europă s-au întâlnit pentru prima dată la congresul Percomp în Hamburg, pentru a se sfătui relativ la măsurile ce ar trebui luate pentru combaterea temușilor viruși. Nume atât de renumite printre inițiați ca cel al prof. dr. Klaus Brunstein, Universitatea Hamburg sau al specialistului islandez Fridrik Skulason, Universitatea Reykjavik s-au numărat printre participanți alături de expertul particular în securitatea și regăsirea datelor dr. Alan Solomon, Marea Britanie. Și producători de soft de renume ca Digital Research și Borland au fost prezenți. Reprezentanți ai polițiilor germane și olandeze au remarcat că problematica virușilor a rămas (încă) ascunsă. Utilizatorii afectați sînt în mod evident mai interesați de limitarea daunelor decît de aflarea vinovaților.

Ca probleme centrale s-au discutat nomenclatura la apariții mul-

tiple a unor viruși identici și rezultînd din aceasta, identificarea "pasagerilor clandestini". Cu mare interes a fost primită intervenția lui Skulason, care a apreciat că numărul diverșilor viruși de calculatoare și al variantelor modificate va ajunge în viitorul apropiat la 1000. Calculul său relativ la perioada de înjumătățire în care devin cunoscuți noi viruși este de circa 10 luni. În decembrie 1990 erau cunoscuți circa 250 de viruși. Pentru producătorii de software antivirus acest lucru înseamnă că schimbul de viruși detectați, experiența în analiza acestora precum și frecvența actualizărilor sînt susepse unor noi dimensiuni.

O altă dificultate o ridică apariția unor viruși de tip nou, care se autocopiază. Analiza unui astfel de cod poate necesita un atare efort de muncă încît nu mai rămîne timp pentru combaterea unor viruși noi. Ca o altă circumstanță agravantă se adaugă faptul că între timp pro-

gramarea de viruși pare să fi devenit un fel de sport de masă. După cum a relatat Veselin Boncev, de la Academia de Științe Sofia, în Bulgaria există o cutie poștală care servește exclusiv schimbului de viruși de calculatoare. Nenoroocirea cu această cutie poștală (mailbox) este că numai acela are voie să-și copieze viruși pe calculatorul propriu care a transferat în cutia poștală un virus necunoscut pînă în momentul respectiv. Acest lucru nu poate fi înțeles decît ca o provocare de a progna singur viruși noi.

Necesitatea unei diviziuni a muncii pe plan internațional a fost general acceptată. Urmînd propunerilor făcute de dr. Solomon, a fost înființat EICVR: European Institute for Computer Virus Research. Această întreprindere operînd pe baze private va permite pe viitor cooperarea forurilor private, oficiale și universitare în combaterea virușilor. În afară de înființarea unei bănci de viruși (care va fi însă numai la dispoziția unor experți antivirus recunoscuți) va fi reglementată și denumirea virușilor noi apăruiți. Ținînd seama de internaționalitatea prelucrării electronice a datelor, este avută în vedere înființarea unui cartel antivirus mondial.

În cercurile de specialiști, gena EICVR e privită ca o senzație. Darea mai departe a unui cod de virus, în cunoștință de cauză, fie chiar și numai unui cerc restrîns de primitori avizați, era considerată pînă acum ca fiind un delict. Crearea unui for care să supravegheze deschis un astfel de schimb ar putea avea ca efect o încredere mult mai mare în softul de protecție folosit în fiecare caz - oricare ar fi producătorul lui. Din punctul de vedere al utilizatorilor, înființarea EICVR reprezintă o inițiativă salutară.

(FC Magazin, 3/91)

lată, din nou, cîteva prețuri orientative (ianuarie 1991):

Sisteme	Preț de stradă (no-name) (mărci)	Aparate de marcă, ca de ex.
AT 80286 -CPU, 1 Mbyte RAM, harddisk de 40 Mbyte, o unitate floppy	1498	IBM-PS/2-50 cca 5800-7200
386SX 80386SX-CPU, 1 Mbyte RAM, harddisk de 40 Mbyte, o unitate floppy, VGA-color	2066	Compaq386S cca. 7700-8700
386/33 80386-CPU, 2 Mbyte RAM, harddisk 150 Mbyte, o unitate floppy, VGA color	4734	Compaq 386/33 cca. 17.000-22.000
486/25 80486-CPU, 4 Mbyte RAM, harddisk de 320 Mbyte, o unitate floppy, VGA color	9834	HP Vectra 486 cca 31.000-34.000

Poșta redacției

"Deși nu am acces (decît sporadic) la un PC (profesional), revista Dvs. mă interesează pentru că vine în întîmpinarea intențiilor mele de viitor: este actuală, bine scrisă și ... unică în ceea ce privește cantitatea informațiilor de calitate!

Mica publicitate

Vind AT 286 tact 12 MHz, driver 5,25" de 1,2 Mbyte, hard-disk 40 Mbyte, 28 ms, 1 Mbyte RAM, Hercules, monitor monocrom, mouse - totul la prețul de 2500 DM. Tel. 961/48631 - Timișoara.

Cumpăr imprimantă Robotron K6313 sau K6414 și unitate de disc pentru Commodore 64. tel. 807059 sau C.P. 69-130, București.

G&Condor oferă: calculatoare personale, periferice, programe aplicații, afișaje reclamă dinamică, lucrări de instalare, asistență tehnică, service, elaborare proiecte, standarde, documentație tehnică pentru echipamente electronice (editare color) Tel. 976/21968

Specialist software: gestiune de date, interfețe utilizator, grafică. Lucrez TurboPascal (fișiere în acces direct, fișiere index, gestiune heap), limbaj de asamblare. Mari disponibilități de continuă perfecționare (26 ani). Ofer serviciile unei firme străine interesate. Tel. 940/41177

Firma "Hello" vă oferă proiectare asistată de calculator, contabilitatea întreprinderilor, instruire diferite sisteme de calcul. Tel. 90/77.05.86 (București)

Execut programe C, Pascal, dBase. Aștept oferte serioase. Tel. 90/654186

Doresc să colaborez cu posesori de **Commodore 64**. Pa-raschiv V. Teodor, Cezar 63, 6200 Galați

De ce am considerat necesar să vă expediez aceste cîteva rînduri?! Din cauza ... "profesiunii de credință" din nr. 1, care îi condamnă pe posesorii de HC-uri la așteptare (...) majoritatea acestora nu au nici cea mai mică idee despre modul cum l-ar putea folosi (eficient).

Sînt convins că nu ați avea decît de cîștigat (enorm!...) acordîndu-le (totuși ...) un minim de spațiu (= atenție); pentru că, mă tem, nimeni nu se va încumeta să "scoată" o revistă numai pentru HC-uri, iar ca să vă concureze - tratînd și PC-uri - ar fi (cred) sinucidere curată!

Ce vor HC-iștii?! Ce le trebuie lor?! (Ce este HC-iștii și ce vrea ei!)

TOTUL!

Adică: noțiuni elementare (teoria informației, algoritmi, limbaje; sisteme de operare - mai puțin); domenii de utilizare (cu multe sugestii practice), tendințe etc.

Să nu uităm că posesorii de HC-uri sînt viitorii (siguri) utilizatori de PC-uri (deci - potențiali clienți ai firmei Dvs, fie și numai prin revista "it!").

Pentru că majoritatea utilizatorilor de PC-uri din România le pot folosi numai la serviciu (...), n-ar fi lipsit de interes (pentru ei) să discute în revistă și probleme specifice anumitor domenii - cu care ei se întîlnesc în cadrul sarcinilor de serviciu; chiar dacă prin aceasta ați face concurență revistelor "emanate" de "emanațiile" I.C.I. Concurența aceasta nu v-ar dezavantaja - cel puțin judecînd după conținutul de pînă acum al revistei Dvs.!

Precizez, pentru clarificarea lucrurilor, că nu posed HC, nici PC și nici nu am acces la PC-uri (decît accidental); iar această situație va mai dura pînă ce-mi voi putea permite să investesc într-un PC, acțiune în care voi fi ajutat (enorm) de tabelele publicate de Dvs.(...)

Cu îngăduința Dvs., mi-aș permite să sugerez cîteva idei care vă vor spori, cred, audiența:

1. continuarea orientării spre "compatibile IBM", avînd în vedere că această firmă pătrunde masiv în întreprinderi (MS-DOS)

2. atenție specială Bucureștiului - o piață imensă (even if ...)

3. un ciclu despre arhitectura sistemelor de operare (util mai ales pentru elevii de liceu din orașele "provinciale")

4. un ciclu despre limbaje de asamblare (bazat pe cel mai răspîndit)

Ambele cicluri (în serial) una, două pagini/număr; cursul de "macro" - împănate cu exemple și probleme propuse.

Succes!

Cu stimă,"

Coveanu Ion, Slatina

"Ceea ce îmi place cel mai mult la revista editată de dumneavoastră este rubrica cititorului, pe care n-am mai întîlnit-o pînă acum la nici o altă revistă."(...)

Manea Alexandru, Dorohol

"Ideea voastră de a specializa revistele mi s-a părut excelentă. Să vedem cine va lansa o revistă pentru home-computere, care se pare că au o mare răspîndire la noi.

Vă propun inițierea unor concursuri pentru utilizatori. Pentru cîteva premii simbolice, veți putea avea surprize plăcute și o colaborare mai strînsă cu cititorii.(...)"

Cristian Popovici, București

Nota redacției: dorim lansarea unor concursuri pentru cititori, dar încă nu am găsit o modalitate care să ni se pară tentantă. Desigur, materialele care le publicăm sînt plăcite (e drept că mai degrabă simbolic). Pînă ce punem ceva pe roate, lată totuși o idee pentru cititorii noștri: la rubrica "Practică", similară ca idee cu cea din revista noastră (sau mai degrabă invers), revista PCplusTechnik oferă un premiu de cel puțin 50 DM pentru fiecare "mecherie" publicată. Adresa completă este Markt & Technik Verlag AG, Redaktion PC Technik, Stichwort: Tips & Tools, Hans Pinselstrasse 2, 8013 Haar, Germania. Desigur, textele pot fi trimise numai în limba germană sau engleză. Noi putem promite doar că îi vom ajuta prin traducere și reexpediere (acesta să fie "premiul" nostru) pe cel care ne trimite materiale în acest sens, materiale care să ni se pară că într-adevăr aduc ceva interesant și pe care le vom publica. Pentru materialele nepublicate nu ne putem asuma nici un fel de răspundere.

Stimați cititori,

Ca urmare a noilor etape de "liberalizare" a prețurilor, simțem nevoiți să scumpim prețul revistei. Astfel începând cu numărul 3/91, un exemplar va costa 57 lei, inclusiv cheltuielile de expedite.

Începând cu numărul 4/91 simțem nevoiți să renunțăm la expedierea revistelor cu plata prin ramburs, deoarece este sensibil mai scumpă, mai complicată și mai nesigură (retururile fiind foarte scumpe). Începând cu numărul 4/91 revista poate fi procurată numai prin abonament sau de la difuzorii noștri.

Puteți încheia un abonament la revistă completând talonul alăturat și expediind în contul nostru, prin mandat poștal, o sumă calculabilă cu formula:

57 lei * nr. exemplare * nr. de apariții pentru care doriți abonament.

Vă rugăm să vă completați cîteș adresa și să dați, la rubrica "Loc pentru corespondență" de pe mandatul poștal, o scurtă explicație a sumei expediate (nr. exemplare, nr. apariții).

Abonamentele intră în vigoare de la data intrării banilor în cont.

Mărturisim că am fi putut încerca să compensăm "liberalizarea" printr-o reducere a numărului de pagini. Volumul informațiilor de comunicat fiind însă atât de mare încît chiar și numărul de pagini actual este insuficient am considerat măsura mării scumpirii printr-o reducere a numărului de pagini ca fiind de neacceptat. (Am fi putut păstra ~~prețul~~ de 39 lei la cca. 20 pagini !). Ținînd cont și de lipsa acurată de documentații, în numărul următor 4/91, vom adăuga încă 4 pagini conținînd un memento DR-DOS 5.0. În funcție și de modul în care vor fi primite aceste îmbunătățiri de către D-voastră, acesta ar putea fi începutul unei noi serii.

Sperăm să rămîneți în continuare un cititor al revistei "if".

Redacția

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

Micro ATCI

C.P. 64, O.P. 1

RO - 4300 Tîrgu Mureș



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

Micro ATCI

C.P. 64, O.P. 1

RO - 4300 Tîrgu Mureș



LOGIC

Societate comercială româno-americană

SIBIU, Calea Dumbrăvii 21, tel. (024) 46652, 45475 Fax: (024) 46704
BUCUREȘTI, Bulevardul Tineretului 8-10, tel. (01) 754900 Fax: (01) 757135
Cont valută BRCE Buc.: 47.38.112.300-2
Cont lei BRCE Buc.: 30.08.112-9
Cont lei BC Sibiu: 30.80.4.03.01

LOGIC vă oferă produsele *BORLAND* în România

- * PARADOX 3.5
- * PARADOX SQL LINK
- * QUATTRO PRO 2.0
- * TURBO PASCAL PROFESSIONAL 6.XX
- * TURBO PASCAL RUNTIME LIBRARY 6.XX
- * TURBO C++ PROFESSIONAL
- * TURBO C++ RUNTIME LIBRARY
- * SIDEKICK 2.0
- * TURBO DEBUGGER & TOOLS 2.XX
- * PARADOX RUNTIME 3.5
- * PARADOX ENGINE
- * QUATTRO PRO SHOW
- * TURBO PASCAL 6.XX
- * TURBO C++
- * TURBO C 2.XX
- * SPRINT 1.XX
- * SUPERKEY 1.XX

Firma **LOGIC** asigură înregistrarea licențelor de utilizare și acordă reduceri de preț pentru versiunile noi.

Important !

Revista "if" își propune editarea unui catalog al întreprinderilor (de stat sau particulare) a căror sferă de preocupări are tangență cu informatica (producători hard, producători soft, depanatori, distribuitori, comercianți, etc.). Catalogul va conține adresele întreprinderilor, o scurtă descriere a obiectului de activitate al acestora și o prezentare succintă a produselor și serviciilor oferite. Catalogul va fi distribuit prin rețeaua de difuzare a revistei "if", putând ajunge astfel direct la persoanele interesate. Întreprinderile interesate de publicarea acestui catalog sînt rugate să ia legătura cu redacția.

Important !

Căutăm difuzori/distribuitori autorizați pentru difuzarea revistei "if".
Condiții avantajoase ! Rugăm persoanele interesate să ia legătura cu redacția.

LEUTRON VISION AG, ZÜRICH, ELVEȚIA & MICRO ATCI, TÎRGU MUREȘ,
... partenerii Dvs. în procesări de imagini - DIP (Digital Image Processing)



După 10 ani de experiență, spectrul de produse al firmei LEUTRON VISION AG cuprinde :

— **DIP PENTRU APLICAȚII MEDICALE**

Memorii video (Video-Framestore) și sisteme de procesare de imagini complete, în special pentru sisteme röntgen C-ARM: MICROSTORE și VIP100XB pentru norma video HDTV normală precum și pentru cea de mare rezoluție, rezolvă total problema suportului röntgen prin captarea instantanee a imaginilor pe monitor, cu sau fără posibilități de îmbunătățire/manipulare a acestora precum și arhivare pe harddisk.

— **DIP PENTRU APLICAȚII INDUSTRIALE**

Memorii video (Framestore) / interfețe de procesare de imagini pentru sistemele IBM PC/AT sau compatibile; SOFTWARE standard sau la cererea Dvs. ca și sisteme de procesare de imagini industriale pentru controlul calității sau sisteme de vizlune pentru roboți. Interfețe de procesare ultra-rapide prevăzute cu TRANS-PUTER-ule din seria INMOS-IMS T80.

· **DIP PENTRU SISTEMELE VME BUS BAZATE PE MICROPROCESORUL MOTOROLA 68000**

Memorii video/interfețe pentru VME BUS, SOFTWARE la cererea Dvs. precum și sisteme complete (turn-key systems).

Creată în 1979, LEUTRON VISION AG dezvoltă, produce și vinde HARDWARE și SOFTWARE atât ca produse independente cât și ca SISTEME LA CHEIE. KNOW-HOW-ul complet, începînd cu analiza sistemului și pînă la nivelul proiectării HARD și SOFT este realizat de producător, fapt care ne permite să vă satisfacem cerințele Dvs. specifice, inclusiv pentru produse în serie mică.

Produsele LEUTRON VISION AG funcționează cu succes în lumea întreagă.

DISTRIBUTOR OFICIAL PENTRU ROMÂNIA

MICRO ATCI

Str. Bartok Bela Nr. 3

CP64, OP 1

RO - 4300 TÎRGU MUREȘ

Tel. : 954 / 20521

dipl. ec. IUDEAN VIOREL

dipl. ing. KALLO TIBOR

CONTACTAȚI-NE ACUM ! NOI REZOLVĂM PROBLEMELE DVS. !