

PERSONALNI računari

BIGZ
1831

89

FEBRUAR
1993.

CENA 20.000 D

SLT 500 o Denara 16000 o DEM 10 o USD 6 o ATS 70

SOFTVER

WINDOWS FOR WORKGROUPS

AMI PRO FOR WINDOWS

QUATTRO PRO FOR WINDOWS

PKZIP 204C

UPOREDNI TEST

10 SVGA MONITORA

MIKROPROCESORI

KAKO IZABRATI

PRAVI PROCESOR?

SERVERI

COMPAQ SYSTEMPRO

SPECIJALNI DODATAK NA 32 STRANE

MICROSOFT WORD 2.0

REFERENTNI PODSETNIK

PKZIP 204c

INTERVJU

DR MILAN BOŽIĆ

Virtual Library of Faculty of Mathematics - University of Belgrade
elibrary.mathf.bg.ac.rs



CAD RAČUNARI

CAD RAČUNARI I OPREMA

CAD računari i oprema
386 i 486 računari optimizovani za
CAD i grafičke namene
DIAMOND i HERCULES
procesorske grafičke karte
EIZO kolor monitori visoke
rezolucije
HEWLETT-PACKARD
kolor štampači, skeneri i ploteri
POLAROID film rikorderi

SOFTWARE

AutoCAD r.12
AutoSHADE
ANIMATOR Pro
3D Studio 2

USLUGE

AutoCAD škola
CorelDRAW škola
RENTIRANJE plotera,
laserskih i kolor štampača
kolor skenera,
film rikordera

KONSALTING

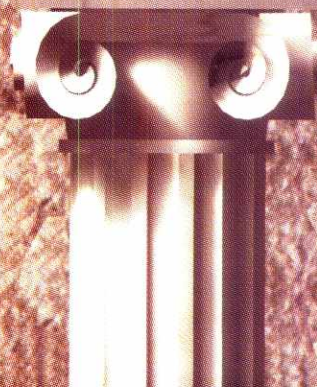
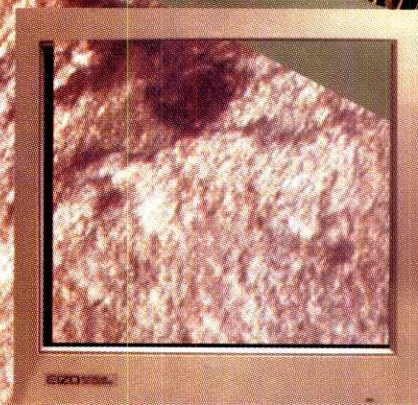
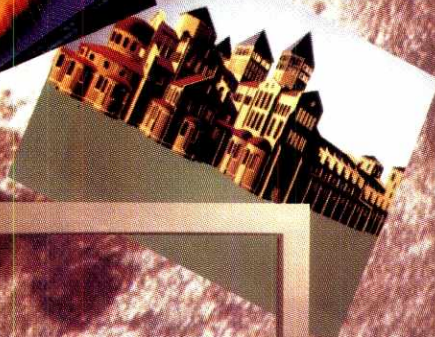
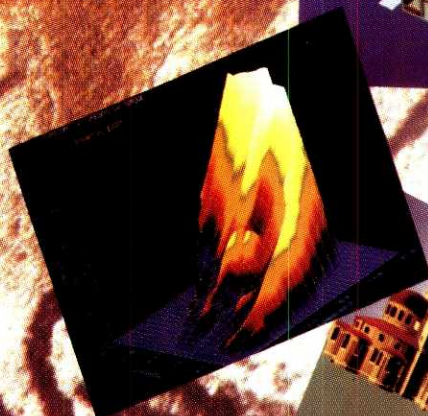
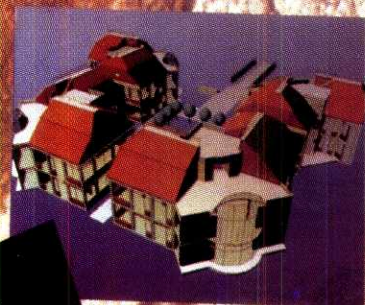
ArhiCAD i Arhi 24
globalna rešenja za
primenu CAD-a u arhitekturi

CAD RAČUNARI STVARAJU UMETNOST

PREDUZEĆE

MICRO ANIMA

Beograd, 29. novembra 71
Tel. 343 940 / 343 294



Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17
Generalni direktor
Bilja Rapajić
Glavni i odgovorni urednik
Jovan Regasek
Komercijalni urednik
Vesna Jeremić
Prevodilac
Ranka Jovanović
Tehnički urednik
Radoje Kavedžić
Ilustratori
Dragan Kovačević
Dejan Medić
Stručna redakcija
Nenad Batočanin (baze podataka),
Ranka Jovanović (aktuelnosti), Dejan
Ristanović (softver), Jovan Skuljan
(programiranje), Vladimir Stamenović
(radne tabele), Dejan Veselinović (har-
dver), Dragutin Vuković (mreže), Zoran
Životić (softver)
Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III
Telefoni
033-748 (redakcija), (prodaja) 651-666
0259
Agencija BIGZ-a
011) 651-793, 653-565
Telefaks
011) 648-140, 647-955
Preplate za zemlju
Za šest meseci: zvatı prodaju
Za jednu godinu: zvatı prodaju
(na žiro-račun: D.P. BIGZ, 60802-603-
0264)
Preplate za inostranstvo
Za jednu godinu 51 USD, 80 DEM, 67
SWS, 27 GBP, 271 FRF
na devizni račun: D.P. BIGZ 60811-
020-16101-820701-999-03377)
Za dug uspostrenog prenošenja uplata
preko banke, molimo pretplatnike da
nam posle svake nove uplate odmah
pošalju foto-kopiju uplatnice.
Kopijisi se ne vraćaju.

SEZAM (011) 648-899 (deset linija)
sistem za modemske komunikacije
Upravitelj sistema
Jovan Regasek
Administratori na sistemu
Zoran Životić i Dejan Ristanović

CENOVNIK OGLASNOG PROSTORA

Cene oglasnog prostora se izražavaju u
"dnevima" čija vrednost odgovara vrednosti ne-
kvalifikovane marke na slobodnom tržištu na dan faktu-
risanja:

- 1/1 poslednja strana korica 1.000 bodova
- 1/1 druga strana korica 900 bodova
- 1/1 treća strana korica 900 bodova
- 1/1 kolor unutrašnja strana 800 bodova
- 1/2 kolor unutrašnja strana 600 bodova
- 1/1 crno-bela strana 650 bodova
- 1/2 crno-bela strana 400 bodova
- 1/4 crno-bela strana 300 bodova
- 1/8 crno-bela strana 200 bodova
- mali oglas 50 bodova

Fakturisanje se vrši na dan ugovaranja
oglasnog prostora i prijema potrebnih materijala
za oglas, sa obavezom da se uplata izvrši pre
ulaska broja u štampu.

Rok za dogovor i rezervaciju oglasnog
prostora je 35 dana pre izlaska broja iz štampe
(25. oktobar za decembari broj).

Molimo vas da se za dogovore i informacije
javite na telefon: 011/653-748, fax 011/648-140,
modem 011/648-899 (mail write redakcija). Kon-
takt: Vesna Jeremić

SADRŽAJ

- 4 Vesti
ŠTA IMA NOVO
- 10 Radne stanice
LEP, SNAŽAN, KOMPAKTAN
- 13 Grafičke kartice / Superstation 3D
SILA SLICI DAJE UBRZANJE
- 15 Polemike / Local bus
OBRAČUN KOD O.K. LOKALA
- 16 Monitori / Uporedni test
EKSPLOZIJA BOJA
- 22 Serveri / Compaq SystemPro
SERVER PO KRVI
- 26 Tehnologija / Mikroprocesori
POVUCITE PRAVI POTEZ
- 32 O čemu misli, šta radi... / Dr Milan Božić
MOJ VIRTUELNI SVET
- 36 Radne tabele / QPW
QAUTTRO... PA OSTALI
- 40 Obrada teksta / AmiPro 3.0
NAJBOLJI PIŠČEV PRIJATELJ
- 46 Operativni sistemi /
Windows for Workgroups
USLUGA ZA USLUGU
- 52 Arhiveri / PKZIP 204c
ZIP JE STIGAO
- 56 Prvi utisci / ACROS 325SE
BELEŽNICA
- 60 Kompajleri / Fortran PlusFort
PRVA LJUBAV, DRUGI PUT
- 62 Alatkke / Stacker
RASTEGLJIVI DISKOVI
- 67 Projekti / Programski jezik Manevro
LEGO JEZIK
- 70 Tehnike programiranja
OBJEKTI U PRAKSI
- 74 Mikroprocesori / i486
ADRESNI MODOVI
- 78 Algoritmi / Pretraživanje
ELEMENTARNE METODE
- 82 Tehnike programiranja / Baze podataka
KNJIGA U KOJOJ SVE PIŠE
- 88 Tehnike programiranja / C++
KONTROLA TASTATURE
- 92 Dejanove pitalice
DELJIVO SA SEDAM
- 93 Zoran Životić
MOJA ŠKOLA C-a
- 94 Dejan Ristanović
BAJTOVI LIČNE PRIRODE
- 96 Bilteni
SEZAM BILTEN
- 98 Polemike
PRESTUP S PREDUMIŠLJAJEM



Na naslovnoj strani: Računari su postali već toli-
ko uobičajena stvar da ih koriste svi, pa čak i
(barem prema zamisli našeg fotografa Ljubiše
Tešića) - anđeli na nebu. Pošto ove neobične ko-
risnike, iz razumljivih razloga, nismo mogli da
snimimo u prirodnom ambijentu, zamolili smo
sekretaricu „Sezama“ Aleksandru Jovanović da
nam pomogne da dočaramo nebesku atmosferu.
Snimak: Studlo Tešić & Nenad Petrović

SADRŽAJ OGLASA

AB SOFT	76
ADACOM	44,47
ADA COMPUTER	72
ASYS COMMERCE	35
BEST	43
CET	12,30
COMTEC	38
CONTRAD	14,4K
ELEGRA	31
ENTER	9
ERC COMMERCE	66
INTEL	7,45
INFO-D	35
INSTITUT „VINČA“	86,87
INTERSOFT	39
JUGODATA	11
MICRO AIR	2K
MICRO KNJIGA	58
MICROSYS	20,21
MIKRONOVA	3K
MIMICO	81
MP BIRO	77
MR SYSTEM	65
OLYMP	73
OSA	5
OTC	57
PERINHARD	54
PREDUZEĆE MZ	58
PROSOFT	55
PS CLUB	51
PS PLUS	50
SAGA	49,25
SOFT PROJEKT	59
ŠUTLIĆ	64
TENNICOM	55

OPERATIVNI SISTEMI

OS/2

Beta-verzija 2.1 odlazi na 5000 adresa

„IBM Personal Software Products“ najavljuje beta-verziju 32-bitnog operativnog sistema – OS/2 2.1. Beta-verzija će otići na više od 5000 adresa, nezavisnim proizvođačima softvera i hardvera. OS/2 2.1 dozvoljava da se OS/2, Windows 3.1 i DOS aplikacije izvršavaju istovremeno na istom displeju i ima novu 32-bitnu grafičku mašinu visokih performansi. Ugrađeni su i softverski drajveri za 256-kolor XGA i SVGA, za brze štampače i popularni CD-ROM, kao i mini-aplikacija za slanje i primanje faksimilnih poruka. To još nije sve – tu je i kod za pen-računare, za sisteme koji primaju glasovne naredbe i za multimedia ekstenzije.

OS/2 2.1 podržava APM (Advanced Power Management – usavršeno upravljanje strujom), što produžava vek trajanja baterije kod notebook računara i hardverskih uređaja dimenzija kreditne kartice, sa novom PCMCIA (PC Memory Card International Association) specifikacijom.

OS/2

OS/2 2.0 na kuru za mršavljenje

IBM će pokušati da osvoji nove korisnike novom verzijom svog operativnog sistema OS/2, koji će biti štedljiviji u korišćenju memorije. Tokom prve plovine ove godine OS/2 bi trebao da se vrati sa kure za mršavljenje, na koju su ga poslali IBM-ovi softverski inženjeri. Oni su putem anketa zaključili da mnogim korisnicima i sistemskim menadžerima smeta baš taj preterani obim u kome se OS/2 širi preko svih sistemskih resursa. Ako neko želi da na svom računaru pod OS/2 koristi mogućnosti višeprogramskog rada i da bez problema uporedo izvršava više aplikacija, bolje je da ni ne počinje bez bar 8 MB radne memorije. A pravi višeprogramski rad je, kako je nedavno pokazala anketa koju je sproveo „Computerworld“, za najveći broj korisnika glavna prednost OS/2 nad Windows-om. Ukoliko neka od aplikacija koje se izvršavaju kraćih, ostatak sistema će neometano raditi dalje.

Halapljivo trošenje RAM memorije i prostora na hard-disku bili su glavni razlog da IBM pošalje OS/2 na „posni tretman“. Novi OS/2, koji će iz toga proizaći, trebalo bi da bude tako vitak i „stesan“ da može da se instalira i na laptop i notebook računare, koji retko mogu da se prošire na više od 4 MB radne memorije, kao i na jeftinije stone računare, a da ostavi i prostora za pokretanje aplikacija.

Ipak, sa operativnim sistemom nije ista situacija kao npr. sa puterom bez masti, kafom bez kofeina, ili duvanom bez nikotina. Kod operativnog sistema se, na žalost, ne može štedeti na isti način. Bez upravljanja memorijom i procesima i bez savremenog sistema datoteka, funkcioniše jedino DOS. Gde onda uštedeti? Alatkama i pomoćnim programima ovaj 32-bitni sistem je, u poređenju sa Windows-om, ionako oskudno opremljen.

Iz IBM-a stižu uveravanja da u Lite verziji neće nedostajati nijedna od značajnijih mogućnosti sistema. Štedeće se isključivo na „nataloženom salu“: veliki delovi koda biće prerađeni i bolje napisani, a suvišne stvari uklonjene.

To ne samo da zvuči dobro, već i obećava da će ovaj operativni sistem raditi brže. Ali, sistem neće biti samo iznutra pokretljiviji. Povećanjem slobodne radne memorije skraćuje se vreme pristupa hard-disku, kao i vreme potrebno za izvršavanje programa. U slučaju da time ne bude umanjena funkcionalnost, korisnici će u Lite verziji dobiti ne jedan oslabljen operativni sistem, već „sportsku“ i bržu verziju starog sistema.

Da li je ovo suviše lepo da bi bilo istinito? Iza svega se, verovatno, krije ono što su u IBM-u imali na umu pri ovim preradama: tiho lansirati sledeću verziju OS/2 sa novim imidžom. Za razliku od prethodnih godina, kada su nove verzije OS/2 kasnile ili bile nedorađene, ovaj put se čini da je „Veliki Plavi“ ozbiljniji.

Glavni konkurent OS/2 sistemu – Windows 3.x – nalazi se na sve više računara, mada nije pravi 32-bitni sistem, nema jednake mogućnosti rada u mreži, niti upravljanje memorijom i multitasking-funkcije kao OS/2. Kraj DOS ere je konačno na vidiku. Nadajmo se da će nam proizvođači softvera ubuduće isporučivati operativne sisteme i aplikacije čiji će zahtevi biti umereni. A.K.

Novell Netware 4.0

Beta-verzija ne obećava puno

Mada je Netware u oblasti lokalnih mreža bez sumnje vodeći softverski proizvod, ima problema pri primenama kod mreža koje se prostiru na velikom području (WAN – Wide Area Networks) i pri upravljanju vrlo velikim lokalnim mrežama. Verzija 4.0 ovog programa trebalo bi da otkloni ove nedostatke. Veliko je zato bilo razočarenje kada je stigla beta-verzija. Zlobnici već tvrde da bi takvom softveru mnogo bolje odgovarala oznaka alfa nego beta. Poznavaoci tržišta, sa druge strane, smatraju da „Novell“ nije želeo suviše rano da otkrije sve svoje karte, pa stoga nije u beta-verziju ugradio sve planirane funkcije.

Opseg funkcija u beta-verziji Netware-a je u svakom slučaju uzak i nedovoljan. Nikakvih problema ne bi trebalo da bude pri povezivanju radnih stanica pod DOS-om, dok je za Apple računare i mašine koje rade pod OS/2 potrebno svakako izvršiti još neke dorade. Tu je i jedan, za sada prilično spor, pomoćni program za upravljanje mrežama pod Windows-om. Još uvek se ne zna koliko dobro će verzija 4.0 rešiti problem heterogenih PC-mreža.

Dobra ideja je implementacija automatske kompresije podataka. Biće podržani i internacionalni sklopovi karaktera. Netware 4.0, nasuprot drugačijim glasinama, nije pretpostavka za korišćenje udvojenih servera. Dugo očekivani dopunski proizvod – SFT Level III, koji to omogućava, može se koristiti i sa prethodnim verzijama Netware-a i trenutno je u fazi testiranja kod velikih mušterija. On će obezbediti optimalnu toleranciju otkaza pri radu, pošto će dva servera biti međusobno povezana optičkim kablom. Ukoliko se jedan server pokvari, drugi će automatski preuzeti njegov posao. SFT Level III ne zahteva da dva servera budu identična. Može se, na primer, kao glavni server izabrati jedan 80486 računar, a kao rezervni neki 80386 računar.

„Novell“ zaostaje za konkurencijom (Banyan) i po globalnim uslugama. Sa sadašnjom verzijom Netware-a može se preko globalne liste pozvati najviše 8 servera. Konačna verzija bi trebalo da omogući komunikaciju sa 50 servera. Posebno je za upravljanje sistemom u velikim kompanijama bitno da imaju pristup do svih raspoloživih servera. U svakom slučaju, korisnici će biti pošteđeni napornim prijavljivanjima na svaki pojedinačni server. Mreža nudi strukturu sličnu stablu direktorijuma na hard-disku. Kod Netware verzija 2 i 3, pristup mreži je bio moguć samo ako je upravnik sistema prijavio korisnika na određenu LAN mrežu, dok to nije neophodno u Netware verziji 4.0. Nova verzija Netware-a trebalo bi da podržava do 1000 korisnika. „Banyan“ je ipak u prednosti: uz pomoć „Banyan“-ovog globalnog servisa Streettalk, na primer, upravnik sistema iz Minhena može bez problema da konfiguriše mrežu u Hamburgu, a korisnici imaju, između ostalog, i mogućnost da se priključe na najrazličitije mreže širom sveta. Ukoliko bi Novell uspeo da ostvari iste takve funkcije, koje će uz to biti i ljubazne prema korisniku, bio bi to još jedan plus, uz prednost koju ima na lokalnom polju, gde Banyan lošije stoji.

Ostaje da se vidi i da li će Novell ponuditi tzv. „Burst Packet Mode“, koji bi naročito trebalo da poboljša performanse mreža sa velikim saobraćajem. A.K.

SOFTVER

Baze podataka

Bolji pristup: Microsoft Access

Kao pećurke posle kiše niču baze podataka za Windows. Do sada su ovim tržištem vladali Omnis, Paradox i Superbase. Sada se pojavio novi takmičar – „Microsoftov“ Access.

Zajedničko svim Windows bazama podataka je to da kroz interaktivno ili objektno programiranje olakšavaju pristup velikim količinama podataka. Očigledno je da „Microsoft“ više ne želi da zaostaje na tom planu. Sa Access-om i najavljenim FoxPro for Windows ubrzo će se na tržištu naći dva „Microsoftova“ proizvoda koja bi trebalo da prokose sadašnjoj i budućoj (dBase) konkurenciji.

Za razliku od FoxPro-a za Windows, Access nema mnogo zajedničkih tačaka sa dBase-om. On nudi isto tako fleksibilne strukture podataka i mogućnosti upita (query) kao i ostali programi koji se danas mogu naći na tržištu. Međutim, u mnogim aspektima ovaj

program ide novim putem. Uz pomoć Access-a trebalo bi da korisnik bude u mogućnosti da bez mnogo programiranja sam postavi na noge složene bazne aplikacije. Ovo se ostvaruje pomoću različitih editora: na raspolaganju su moćni generatori tabela, obrazaca i upita. Uz to, tu su i doterane makro-funkcije i opsežan programski jezik Access-Basic. Korisnik, pre svega, mora definisati neophodne tabele u svojoj bazi. Na raspolaganju su mnogi tipovi podataka (samo za tip „Datum/vreme“ ih ima sedam). Već i kod definisanja tabela moguće je zadavanje različitih parametara, kao što su, na primer, provera ispravnosti unetih podataka ili standardne vrednosti za neko polje.

Za pregledan unos podataka može se upotrebiti editor obrazaca. Access omogućava crtanje kompletnih ekranskih maski. Moguće je i njihovo komformnije kreiranje, pomoću internih „pomoćnika“ za izradu obrazaca.

Svoju jaču stranu Access pokazuje, pre svega, u odnosu na izveštaje. Moguće je generisanje tabela sa samo nekoliko pritiska na taster miša. Format koji su unapred zadati mogu se promeniti. Podjednako elegantan je i rad sa upitima (queries), koji se mogu izraditi najvećim delom pomoću miša. Upiti omogućavaju proizvoljna povezivanja između tabela u bazi podataka. Ni fleksibilni upiti nisu nikakav problem: pomoću automatski generisanog dijaloga, korisnik u svakom trenutku može zadati uslove po kojima se pretražuje.

Da bi se u formulare integrisala „dugmad“ (buttons), još uvek nije potrebno posegnuti za programiranjem. Ona se dodaju u editoru formulara, a zatim im se pomoću dijaloga pridružuju svi neophodni parametri – čak i makroi koji se sa svoje strane generišu u makro-editoru. Makro je prost skup i kombinacija stavki iz padajućih menija. Ova jednostavna metoda omogućava izgradnju kompleksnih aplikacija sa korisnički definisanim radnom okolinom, „dugmadima“ i sopstvenim menijem.

Kome ovo nije dovoljno, taj može da piše sopstvene programske module uz pomoć ugrađenog jezika, koji je u ovoj verziji izuzetno moćan: on nije samo prosta suma uobičajenog jezika i naredbi za rad sa bazama, već su tu i funkcije potrebne za razne proračune.

Program je dobro opremljen i mogućnostima uvoza/izvoza (import/export) podataka i poseduje filtere za sve važne formate. Onima koji ne žele da napuste svoje stare SQL podatke, u pomoć priskabe nova ODBC (Open Database Connectivity) tehnologija. U praksi, ODBC znači povezivanje sa podacima iz nekog SQL servera uz pomoć ODBC drajvera. Ukoliko je takav drajver na raspolaganju, recimo za Oracle, moguće je direktno koristiti Oracle-ove podatke i njima manipulirati. Access dopušta i laicima da se zbliže sa pojmom „baza podataka“. Za profesionalce, Access može da posluži i kao razvojni alat.

Kojim ciljnim grupama je Microsoft namenio svoje dve baze – FoxPro i Access? FoxPro je zamišljen za sistemske programere koji žele da razvijaju složene aplikacije. Access je više korisnički orijentisana baza podataka, za manja i srednja rešenja, pri čemu u „Microsoftu“ priznaju da preklapanja između ove dve ciljne grupe postoje. A.K.

Komunikacije

Microsoft i WordPerfect: Novosti u elektronskoj pošti

„Microsoft“ ubrzano radi na projektu pod šifrom „Calvin & Hobbes“, programu za kreiranje obrazaca za rutiranje elektronskih poruka, koji će se koristiti uz Windows for Workgroups. Elektronska pošta je sve popularnija u lokalnim mežama i radnim grupama, a ovakav program omogućava korisnicima da upućuju e-mail, faks, pa čak i glasovne poruke (voice mail) na druge nodove i mreže tako što jednostavno označe odgovarajuće rubrike u elektronskom obrascu. Program će se oslanjati na „Microsoftov“ Messaging Application Program Interface (MAPI), skup funkcija i servisnih poziva koji omogućavaju kreiranje aplikacija za slanje poruka, kao što je „Calvin & Hobbes“.

MAPI se gradi sa nezavisnom arhitekturom koja omogućava, pomoću samo jedne aplikacije, pristup sistemima za elektronsku poštu koji odgovaraju MAPI standardu, kao i faks-mašinama i uređajima za slanje glasovnih poruka.

I korporacija „WordPerfect“ radi na razvoju sličnog proizvoda, ali će on kombinovati elektronsku pri-

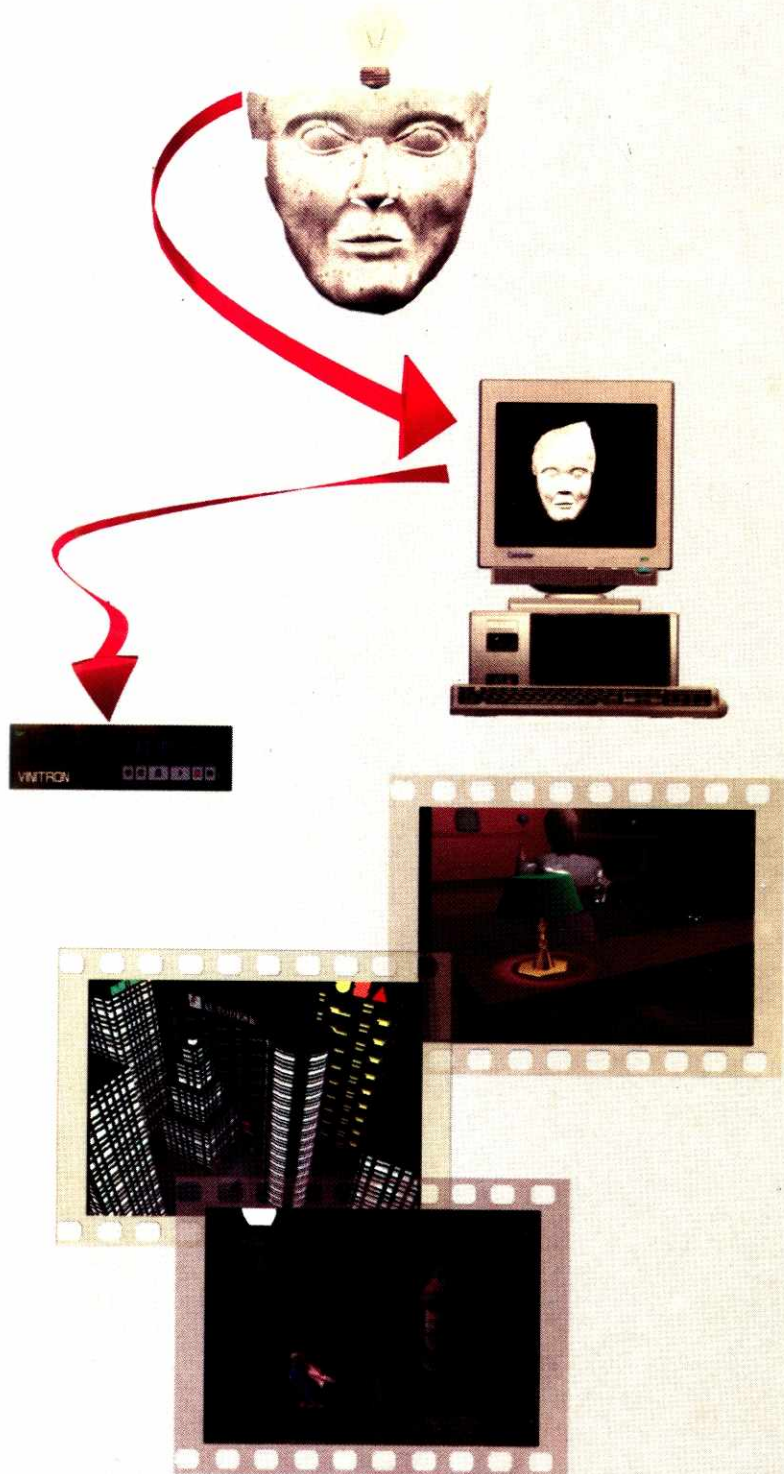


Animacija

- 2D i 3D animacija
- kombinovanje žive slike i računarske grafike
- specijalni efekti
- veliki izbor materijala
- prebacivanje na video:
 - betacam
 - umatic
 - vhs



ČAROLIJA STVARANJA



RAČUNARSKI INŽENJERING

ŠUMADJSKI TRG 6a, BEOGRAD, TEL. 011/ 55 75 92, FAX. 011/ 55 98 45

premu obrazaca i distribuciju pošiljki. InForms 10 će biti čvrsto integrisan sa programom WordPerfect Office koji ima ugrađenu opciju za elektronsku poštu. Program se očekuje početkom 1993, a sadržaća dva posebna paketa: dizajnerski modul će imati alate za kreiranje i održavanje obrazaca i direktan pristup na nekoliko poznatih baza podataka; Filer modul će omogućiti korisniku da popunjava formular, a radiće pod operativnim sistemima DOS, Windows, OS/2, Unix i PenPoint. WP obrasci će imati i opciju „elektronski potpis“, što će sprečiti neovlašćene korisnike da menjaju njihov sadržaj.

Programski jezici

Uskoro Visual Basic 2.0

Visual Basic 2.0, najavljen u novembru, obećava više značajnih novina – više kontrole, više opcija, ključnih reči, događaja, alata za dibaging, pomagala za programiranje i – što je možda najvažnije – bolje performanse.

VB 2.0 će imati dve verzije – Standard Edition i Professional Edition. Profesionalna verzija je zamena za Professional Toolkit 1.0 i Control Development Kit, dok je u standardnu verziju dodato nekoliko opcija iz paketa Toolkit 1 – OLE kontrola za klijenta i usavršen spređiš objekat.

Alatke u obe verzije imaju dve nove opcije. Prva je grafička kontrola, koja omogućava kreiranje raznih oblika, linija i labela kao i ranije, ali uz korišćenje manje Windows resursa, tako da se mogu graditi veće aplikacije. Druga opcija je tzv. „image box“ – kao i raniji „picture box“, prikazuje BMP i ICO grafičke datoteke, ali omogućava i promenu veličine slike. Osim toga, umesto samo 16, prikazuje grafiku u 256 boja.

Sve kontrolne opcije sada su smeštene u prozor koji se može pomerati i skrolovati (za razliku od ranije, statičnog). Globalne i tipске definicije, ranije dozvoljene isključivo u GLOBAL.BAS datoteci, sada se mogu ubaciti u svaki kodni modul (ali ne i u modul osnovasca). Ako se deklaracija počne sa instrukcijom Option Explicit, VB 2.0 zahteva da sve promenljive budu deklarirane pre upotrebe. To znači da programer mora malo više da se pozabavi strukturiranjem, a što zato izvorni kod postaje lakši za razumevanje i održavanje, naročito u projektima za radne grupe.

Nekoliko novina omogućava gradnju većih aplikacija. Visual Basic 2.0 dozvoljava četiri puta veći broj procedura po jednoj aplikaciji, dvostruko veći broj promenljivih u jednom modulu, dvostruko veći broj instrukcija u jednom obrascu, dvostruko veći broj globalnih promenljivih, neograničen prostor za stringove (prema raspoloživoj memoriji i probijanje barijere od 64 KB – ali samo u unapređenom Windows režimu). Sve u svemu, Visual Basic 2.0, za sada samo u beta-verziji, donosi značajne novine i vredi ga pogledati.

Sinteza glasa

Pro Voice: da programi progovore

ProVoice vam omogućava da svojim DOS-aplikacijama dodate sintizovani govor. Paket koristi Speech Engine kompanije „First Byte Software“, koji nizove slova, brojeva i drugih podataka pretvara u govor, koristeći rečnik i gramatička pravila engleskog jezika. U rečnik možete ubaciti nove reči, koje Pro Voice ne prepoznaje ili ih pogrešno izgovara.

Program ima ugrađene alate pomoću kojih se ProVoice Speech Engine vezuje sa izvornim kodom programskih jezika Microsoft C, Turbo C++, Turbo Pascal, Turbo Assembler i Microsoft Assembler. Ove veze omogućavaju da aplikacije direktno pozivaju TSR govornu mašinu, koja tekst pretvara u govor.

Cena: \$595.

Kontakt adresa: First Byte Software, Torrance, CA, (800) 523-2983 or (310) 793-0600; fax (310) 793-0601

Programske alatke

QuickStart 2.0: 32-bitne aplikacije u

16-bitnom Windows-u

Još jedna novost za programere – novi softver kompanije „Phar Lap Software“ omogućava kreiranje 32-bitnih Microsoft Windows aplikacija bez napuštanja

16-bitnog Windowsa 3.1. QuickStart verzija 2.0 koristi Win32s, „Microsoftov“ API (application programming interface), uz koji se aplikacije pisane za dugo očekivani Windows NT savršeno izvršavaju i u Windowsu 3.1. Upotrebom Win32s, programeri koriste samo jedan deo kompletnog Win32 API-ja. Po rečima kompanije „Phar Lap“, to znači da programeri koji rade pod NT-om mogu koristiti sve prednosti drugih razvojnih alata namenjenih već proverenim okruženjima. „Alatke obično zaostaju za operativnim sistemom,“ kaže predstavnik „Phar Lap“-a Maria Vetrano. Alternativa bi bila da se razvoj programa obavlja u Windowsu 3.1, da se zatim ponovo startuje mašina i testiranje obavi u NT-u.

Postavlja se pitanje da li se može desiti da se neka aplikacija drukčije ponaša kada koristi Win32s pod Windowsom 3.1 nego pod NT-om. „Phar Lap“ tvrdi da nisu naišli na takve probleme. Gotove aplikacije se mogu pokrenuti pod NT-om, radi konačnog testiranja i otklanjanja bagova.

Za prvo vreme, kompanije koje se bave razvojem softvera dobiće QuickStart 2.0 besplatno, dok se proizvod ne učvrsti na tržištu i dok se ne stekne utisak o tome „gde se uklapaju sa NT-om“.

Inače, Phar Lap je poznat po softveru za proširenje DOS-a, koji omogućava da programi pod DOS-om rade u 32-bitnom režimu. Njihova 386/DOS Extender tehnologija je ugrađena u najnoviju „Microsoftovu“ bazu podataka FoxPro 2.5 za DOS.

Kontakt adresa: Maria Vetrano, Phar Lap Software, 617-661-1510, fax 617-876-2972.

Programski generatori

Proteus 6.0: Jednostavno kreiranje demo programa

Trideset i pet video i animacijskih efekata, podrška za Music Definition Language, glasovnu datoteku Sound Blastera i proširenu memoriju – to su dodaci kojima je „Genus Microprogramming“ obogatio novu verziju Proteusa, svoje prototipske/demo alatke za PC. Proteus 6.0 nudi jednostavno kreiranje demo programa, raznih programa za obuku (tutorial) i softverskih prototipova koji kombinuju tekst i grafiku sa specijalnim efektima, zvukom i muzikom. Paket omogućava da se sve demo datoteke smeste u jednu EXE datoteku, te da se odredi datum posle kojeg demo više ne može da se startuje.

Cena: \$349.

Kontakt adresa: Genus Microprogramming, Houston, TX, (800) 227-0918 or (713) 870-0737; fax (713) 870-0288.

Zanimljivosti

Microsoftov konkurs za najbolji softver

Na novembarskom COMDEX-u, u Las Vegasu, „Microsoft“ je objavio rezultate konkursa, otvorenog na svetskom nivou, za najbolji Windows NT „shareware“ softver. Prispelo je 50 programa iz celoga sveta (među njima i jedan iz bivšeg Sovjetskog Saveza), a u žiriju su bili nezavisni novinari: Diego Aranda sa „Interneteta“, Steve Gibbons i Brian Livingston, novinari „Infoworlda“, Fred Langa, urednik „Windows Magazine“-a, i Jerry Pournelle, jedan od urednika Byte-a. Sponzori su bili, između ostalih, „Artists Graphics“, „CompuServe“, „Jolt Cola“ i „Microsoft“. Većina aplikacija je sa Windows 3.1 ili Unixa prenesena na NT platformu.

Takmičenje je podeljeno u pet kategorija, a za svaku su predviđene tri nagrade. U poslovnoj kategoriji je pobedio program „While You Were Out“, Windows NT sistem za slanje i prijem poruka kompanije „Caliente Software“. Drugo mesto je zauzeo program „Mortcalc“ za obračun hipotečke amortizacije Roberta Paula, a treće „Time and Money Tracer“ iz „Wintrionix“.

U kategoriji alataki pobedio je konfigurabilni tekst-editor za programere, „Microemacs“. Na drugom mestu je bio Windows komunikacioni program „Kermit“ Wayne Warthena, sa ikonom koja predstavlja popularnog žapca Kermita iz emisije „Muppet Show“, dok

je treći bio simulator strujnog kola, „Nutmeg-32“ Roberta Zeffa.

Prvi u kategoriji uslužnih paketa je bio „Winbatch“, batch jezik za NT Wilsona Windowwarea. Drugi je bio „Ados“, program koji usavršava rad iz Windows NT komandne linije, a treći „Trashman“, program za brisanje „drag-and-drop“ datoteka koji je potpisao „Trigon Software GmbH“.

U „Widget“ kategoriji, najbolji je bio „Icon Manager“, komplet alataki za editovanje, sistematizaciju i instalaciju Windows NT ikona. Drugi je bio program Norberta Unterberga „Winmod“, zvučni plejer datoteka koji pokreće datoteke iz Amiga MOD formata, a treći „Prime32“, softver koji izračunava i šalje na ekran sve primarne 32-bitne brojeve.

U poslednjoj kategoriji su ocenjivane igre. Pobedio je „Klotz“ (još jedna igra na temu blokova koji padaju), druga je bila popularna igra sa uklapanjem pločica, „Mah Jongg“, sa novim oblicima ploča, novim putokazima i srednjevekovnim simbolima. Treća je bila varijanta kartaškog „pasijansa“, „Thieves and Kings“.

Pobednici su dobili Artist Graphics Winsprint 180 video-karticu i Nanao 340i 15-inčni monitor. Druga nagrada je bio NEC Multimedia Upgrade Kit, a treća Artist Graphics Winsprint 100 video-kartica.

„Microsoft“ je zakazao novo takmičenje čiji će pobednici biti objavljeni na prolećnom COMDEX-u, a zainteresovani kandidati detalje konkursa mogu naći na Internetu ili CompuServe Microsoft Developers Forumu.

MULTIMEDIJA

Prepoznavanje govora

IBM-ove mašine umeju da slušaju

Tokom jesenjeg COMDEX-a, IBM je predstavio četiri proizvoda koji odgovaraju na davnašnju želju korisnika – da kompjuteri razumeju ljudski govor. Dva su rađena po licenci pionira u oblasti prepoznavanja govora, kompanije „Dragon Systems“, dok je dva razvio sam IBM.

IBM je otkupio licencu za DragonDistate-7K, sistem za diktiranje za DOS računare. Pomoću ovog softvera, korisnici kontrolišu DOS i mnoge aplikacije govorom a ne kucanjem naredbi. Ima vokabular od 7.000 reči i rečnik od 100.000 reči, a reaguje na svaki glas.

Druga licenca je za Dragon Talk-To Plus, paket za prepoznavanje govora za Microsoft Windows. Talk-To Plus ima biblioteku ugrađenih komandi, a istovremeno mogu biti aktivne 64.

Oba „Dragon“ proizvoda zahtevaju PC sa 386SX ili moćnijim procesorom i zvučnu karticu – pominju se IBM-ov M-Audio Capture i Playback Adapter (M-AC-PA), kao i „Creative Lab“-ov Sound Blaster 16. Talk-To Plus će, počevši od marta, prodati i IBM i Dragon, po ceni od \$149, a DragonDictate-7K po ceni od \$2.295.

IBM je već ranije uzeo licencu za „Dragonov“ VoiceType softver koji služi kao zamena za tastaturu.

IBM je najavio i seriju Speech Server, softvera za RISC System/6000 radne stanice. Speech Server radi i na RS/6000 serveru, dok PC mašine rade pod OS/2 kao klijenti. Po rečima IBM-ovog predstavnika, Speech Server je izuzetno precizan sistem koji prepoznaje samo glas određenih govornika i ima vokabular od 20.000 reči. Može pratiti diktat do 70 reči u minuti, a rečnik se može prilagoditi potrebama korisnika.

Speech Server će imati cenu od \$6.950 za server program, \$695 za OS/2 klijentski softver, i \$1.900 za RS/6000 karticu za ubrzanje govora. Zaseban softver za RS/6000, koji radi sa IBM-ovom AIX varijantom Unixa, će koštati \$2.495 i očekuje se u maju.

Konačno, IBM je lansirao i Continuous Speech Series Developer's Program, koji nezavisnim proizvođačima softvera nudi alate za ugradnju opcije za prepoznavanje govora u pakete pisane za OS/2 i RS/6000. Softver ima aktivni vokabular od 1.000 reči, iz postojeće baze od 20.000. Ove alatke će se prodavati po ceni od \$3.995.

Kontakt adresa: Tara Sexton, IBM, 914-766-3781 or 914-642-4662; Esther Agonis, Dragon Systems, 617-995-5200.

INTEL

INTEL
Vam obezbeđuje
WINDOWS 3.1
486+53
super-grafika
32000 boja

INTEL
486SX-33MHz
4MB RAM
40MB HDD
1,2Mb FDD
SISA i more
75412
VGA i hard
diskovi

INTEL
486SX
- minimalno početno ulaganje
manje nego za 386DX
- za samo 5 minuta možete
nadgraditi svoj računar u
486DX-33 ili 486DX-50
- displej samo za razliku u ceni

INTEL 486 je računar
koji raste zajedno sa Vama

1999deni

INTEL - Institut za Mikrotalasnu Tehniku i Elektroniku
Bulevar Lenjina 165b, 11070 Novi Beograd
Tel. (011) 135-420 . 134-516 Fax: (011) 139-423 . Telex: 12757 INTELJU

Alatke

Norton Commander: Disk alatke za OS/2 2.0

„Symantec“ je predstavio seriju Norton uslužnih alatki za IBM-ov grafički operativni sistem OS/2 2.0. Norton Commander je sličan programu Norton Desktop for Windows i omogućava upravljanje datotekama, pronalaženje datoteka i podršku za mrežne menije.

Program nudi i upravljanje sistemom datoteka visokih performansi (HPFS).

Alatka za pronalaženje datoteka sa globalnim (džoker) znacima (global file find utility) traže drajrove ili datoteke po imenu ili ekstenziji, putem džoker-znakova iz DOS-a, kroz celi drajv hard-diska. Datoteke koje odgovaraju kriterijumu pretraživanja se prikazuju u listi, odakle se mogu startovati ili pregledati.

Korisnici mogu pregledati, editovati, kopirati, premenovati, premeštiti ili izbrisati datoteku ili grupu datoteka uz pritisak samo nekoliko tastera ili klikova mišem. Moguće je i poređenje foldera i premeštanje datoteka iz jednog foldera u drugi. Mogu se koristiti read-only meniji unutar računarske mreže i slati na druge OS/2 kompatibilne mreže.

Da bi se podaci bolje organizovali, jedan meni se može smestiti unutar drugog. DOS i Windows aplikacije koje rade pod operativnim sistemom OS/2 se mogu unapred konfigurisati tako da se startuju sa minimalnim, maksimalnim ili normalnim prozorima.

Maloprodajna cena za Norton Commander je \$49.

Alatke

PKZip za Windows: Arhiviranje iz Windows-a

Korisnike Windows-a koji koriste PKZip za kompresiju i dekompresiju datoteka kompanija „Spiffy Software“ iz Sietla će obradovati uslužnim programom Drag 'N' Drop for PKZip, koji će im omogućiti da arhiviranje obavljaju iz Windowsa, bez prelaska u DOS.

Programu Drag 'N' Drop se pristupa iz Windows File Managera, klikom na datoteku ili grupu datoteka, koja se zatim prevuče na ikonu koju animira korisnik i koja ima naslikan patent-zatvarač (zipper). Datoteka će biti automatski komprimovana ili dekomprimovana. Kod kompresije, patent-zatvarač na ikoni je zatvoren, a kod dekompresije otvoren. Drag 'N' Drop se izvršava u pozadini, a korisnik dobija obavještenje kada je operacija završena. Ako komprimovana datoteka sadrži nekoliko datoteka, korisniku se nudi opcija da odabere samo određene delove za dekompresiju i da odredi gde da se smeste.

Kada je datoteka „zipovana“ (komprimovana ili dekomprimovana), smešta se u privremeni zip-fajl. Korisnik ovaj korak može preskočiti, navodeći ime određene datoteke i direktorija.

„Spiffy Software“ preporučuje da se Drag 'N' Drop smesti u startup, s obzirom da zauzima samo malo memorije, da bi se automatski učitavao.

Da biste koristili Drag 'N' Drop, treba vam nekakva verzija PKZipa, koji se može „skinuti“ sa gotovo svakog BBS-a. Cena za Drag 'N' Drop je \$24.95.

TEHNOLOGIJA

Mikroprocesori

Priča se o Intelovoj „šestici“

Još se nije pojavio famozni „Intelov“ mikroprocesor P5, a već su procurile vesti o njegovom nasledniku – P6. Kažu da će se „šestica“ pojaviti za 18 meseci i da će imati 10 miliona tranzistora – triput više nego petica i osam puta više nego 486DX. Ako se predviđanje ostvari, P5 će biti najkratkotrajnija generacija procesora u istoriji Intela: 286 je došao 44 meseca posle 8086; 386 je usledio za još 44 meseca, 486 za 42 meseca, dok se P5 očekuje oko 45 meseci posle 486-stice.

Memorije

Fleš-memorija: jedna dobra i jedna loša vest iz Intela

Intel je još pre nekoliko meseci najavio da će 8-megabitni fleš-memorijski čipovi kasniti oko godinu dana. Sada se rok opet pomera, zbog velike potražnje, uglavnom od strane firmi koje proizvode PCMCIA (PC Memory Card International Association) kartice za skladištenje za najnoviju generaciju minijaturnih, ručnih kompjutera.

Dok tržište obiluje 1-Mb fleš-memorijским čipovima, „Intel“ je jedini proizvođač koji nudi čipove većeg kapaciteta. Fleš-memorija je veoma privlačna, jer daje povećan kapacitet na manjem prostoru, troši malo struje i čuva sadržaj čak i kada se struja isključi.

Neki analitičari su predviđali da će fleš-memorija zameniti hard-disk drajrove. Međutim, da bi se to desilo, fleš-memorija bi morala biti manja, sa više kapaciteta, i jeftinija.

„Intelov“ predstavnik Majki Saliven je nedavno novinarama saopštio dve vesti – jednu dobru i jednu lošu. Dobra vest je ogroman rast potražnje – za poslednjih pet meseci povećana je za 500%. Loša vest je to što proizvodnja ne može da prati toliku potražnju. Muku muči Intelov japanski snabdevač „NMB Semiconductor“, iako „Intel“ stalno šalje nove ljude u Japan. Saliven je rekao da se planirani datum isporuke, kraj 1992, odlaže za godinu dana. Zašto toliko zakašnjenje? Za sada, 8 Mb čipove proizvodi samo „NMB Semiconductor“, ali se razmišlja o početku proizvodnje i u Intelovom postrojenju „Fab 7“ u Santa Klari, dok se istovremeno vrši veliki pritisak na „Sharp“, s kojim „Intel“ već ima ugovor o proizvodnji 8 Mb čipova.

Kašnjenje jeste neprijatno, ali u „Intelu“ veruju da neće imati finansijske reperkusije, s obzirom da je udeo svih memorija, pa i fleš-memorije, zanemariv u odnosu na profite od mikroprocesora (u Intelovom slučaju, 10%). Ali, predviđanja govore da će tržište fleš-memorije naglo procvetati i sa sadašnjih 130 miliona dolara do 1995. porasti na 1,5 milijardu.

„Advanced Micro Devices“ i „Fujitsu Limited“ najavljuju svoje 8-ične fleš-memorijske pločice (wafers), koje bi se mogle pojaviti 1994, a omogućile bi proizvodnju čipova od pola mikrona, pa i manje.

I IBM je najavio planove za utruženje memorije na istom prostoru, takozvano „kubiranje“, pomoću tehnologije razvijene u kalifornijskoj firmi „Irvine Sensors“.

PERIFERIJE

Tastature

Telefoniranje preko tastature

CompuPhone, PC-kompatibilna tastatura sa 101 tasterom, firme „Integrated Technology“ ima jednoičnu telefonsku liniju i interfejs za slušalice sa mikrofonom, tako da nije potreban poseban telefon za telekomunikacije. Koristi se numerička tastatura umesto telefonskog brojanika, a postoji ugrađen softver za automatsko okretanje broja.

Cena: \$299

Kontakt adresa: Integrated Technology, Inc., 76 South Orange Ave., South Orange, NJ 07079, (201) 907-0200; fax (201) 762-7234.

Štampači

Star: Malo, jeftino, Windows kompatibilno

Firma „Star“ proširila je liniju svojih proizvoda izbacivši na tržište dva nova i jeftina modela – LC24-100 Multifont i LC-100 Colour. Ova dva modela se po svojim mogućnostima nalaze na donjem kraju „Starovog“ asortimana. Uprkos tome, oba nude interesantne mogućnosti. 24-iglični štampač LC24-100 podržava maksimalnu rezoluciju od 360x360 tačaka po inču i nudi čitav niz fontova. Štampač LC-100 je 9-iglični, sa maksimalnom rezolucijom od 240x216 tačaka po inču, ali zato može da štampa u boji.

LC24-100 se odlikuje izuzetnom podrškom Windows-u. Sa ovim modelom se isporučuje disketa na kojoj se pored četiri TrueType fonta i različitih drajvera za standardne aplikacije nalazi i jedan specijalno prilagođen drajver za upotrebu u Windows okruženju.

On ne samo da optimalno upravlja štampačem, već i komprimuje podatke pre prenosa do štampača, što ubrzava štampanje i smanjuje opterećenje sistema prilikom štampanja. LC24-100 može da prenese na papir 5 NLQ fontova u deset varijacija. Kao što je kod matičnih štampača uobičajeno, tu je samo jedan draft-font. Brzina štampe koju navodi proizvođač se kreće oko 64 znaka u sekundi u NLQ režimu, a u običnom draft režimu se penje na 192 znaka u sekundi.

LC-100 štampa nešto sporijim tempom. On štampa 180 znakova draft kvaliteta i 45 znakova NLQ kvaliteta u sekundi. Četiri NLQ fonta u sedam varijanti omogućavaju pristojnu pismenu korespondenciju i sa ovim štampačem. Ono što LC-100 čini interesantnim za korisnike-amatere je mogućnost štampanja u boji. I sa njim se dobijaju drajveri za većinu standardnih aplikacija.

Firma „Star“ nastoji da upotrebu svojih štampača učini što je moguće više jednoobraznom. Podešavanje oba štampača je veoma dobro rešeno. Prekiđač za uključivanje štampača nalazi se kod oba modela levo, na prednjoj strani. Oba štampača imaju po četiri tastera, dovoljna za sva podešavanja. Preko njih se podešava veličina fontova i upravlja se papirom. Devet svetlećih dioda simulira mrske DIP-prekiđače, koji se sad mogu lako podešavati pomoću ona četiri tastera. Ipak, objašnjenja u priručniku nisu dovoljno jasna, tako da se od novog korisnika zahteva izvesno predznanje o štampačima.

Cena modela LC24-100 Multifont je 798 DEM, dok model LC-100 Colour košta 648 DEM. Mogu naručiti od: Star Micronics, Frankfurt. A.K.

ZAŠTITE

Virusi

IBM i Western Digital: PC imun na viruse

IBM planira predstavljanje antivirusnih personalnih računara, zasnovanih na novoj hardverskoj tehnologiji koju je uveo „Western Digital“. Primenjena na standardnu arhitekturu (ISA), sa WD7855 kontrolerom u vidu jednog čipa, ova hardverski zasnovana tehnologija predstavlja preventivni lek protiv virusa.

„Western Digital“ kaže da se ugrađena „imunizatorska“ logika za kontrolu sistema pokreće preko SMI-a (System Management Interrupts) i zatim preko WD7855 kontrolera prati sve zapise na hard-disk, reagujući na svaki sumnjiv postupak.

Ovaj imunizator (Immunizer) zabranjuje upis u sve regije hard-diska u kojima se nalaze izvršne datoteke, jer su upravo te datoteke u koje obično prodiru virusi, preuzimajući kontrolu nad centralnim procesorom i počinju svoj prijavni posao. Immunizer štiti i boot sektor hard-diska, što je posebno važno. Naime, statistički podaci koje je objavio IBM ukazuju da se ogromna većina „divljih“ virusa javlja upravo u boot-sektoru. „Western Digital“ naglašava da WD7855 kontroler može sprečiti virusnu infekciju a ne samo otkriti virus. Inače, program otkriva i poznate i nepoznate viruse, i ne utiče na sistemske operacije i performanse. U kombinaciji sa konvencionalnim softverskim antivirusnim paketima, WD7855 kontroler obezbeđuje zaštitu od sve šire epidemije postojećih, novih i mutiranih virusa.

Ova tehnologija će biti uključena u sve buduće sistemske kontrolere ove kompanije, uključujući i 32-bitne. IBM će ugraditi „Western Digital“ kontroler i u svoju novu desktop liniju, sa IBM-ovim 486SLC2 mikroprocesorom.

Pripremila: Ranka Jovanović

YU SLOVA
EPSON, STAR, PANASONIC
I SVI OSTALI ŠTAMPAČI,
LATINICA, ЋИРИЛИЦА
011/685-779

021 369 907



Virtua Library of Faculty of Mathematics - University of Belgrade
library.matf.bg.ac.rs

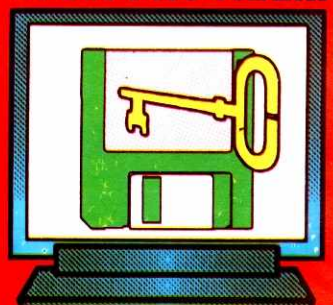
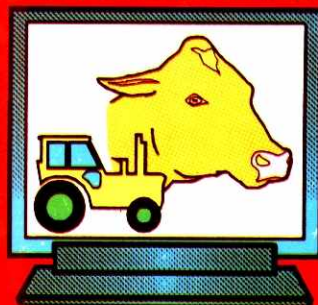
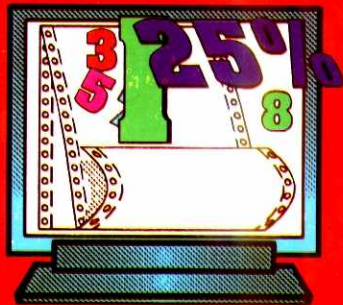
APLIKATIVNI SOFTVER SA HARDVEROM

BANKARSTVO

KNJIGOVODSTVO

POLJOPRIVREDA

OPŠTI PROGRAMI



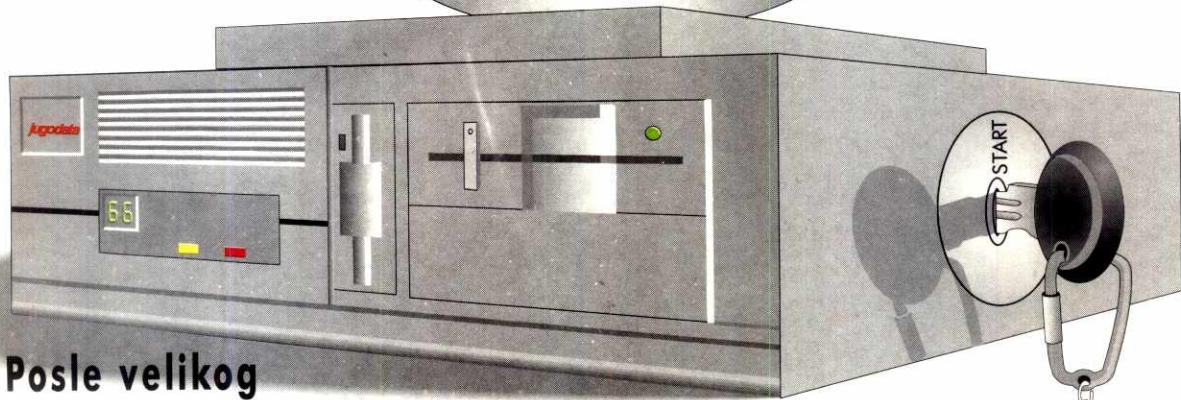
Enter®

kompjuterski inženjering i AOP
Novi Sad, Narodnog fronta 53

Pokrenite svoje računare!



DEMINO



Posle velikog uspeha verzije 1.0 nova, poboljšana verzija PLATE 2.0 stoji Vam na usluzi.

Mogućnost obračuna na bazi tri vrste bodova, fiksnih i varijabilnih iznosa, satnice... ● Uključivanje u obračun do 100 različitih vrsta primanja i obustava ● Neograničen broj isplata u toku meseca i godine ● Praćenja primanja po dva osnova: obračunski i isplatni period ● Dokumentacija za isplatu u SDK: ZID, specifikacija virmani... ● Dokumentacija firme: platni spisak, virmani za uplatu po bankama, isplatni listići radnicima, arhivska dokumentacija... ● Dobijanje četiri različite vrste proseka ● Brz obračun za veliki broj radnika ■ DODATAK: instalacija, obuka, prvi obračun



Pogled u budućnost!

jugodata

S O F T W A R E

B. Revolucije 326, 11050 Beograd, tel/fax: (011) 418-326, 419-768; (081) 11-618

LEP, SNAŽAN, KOMPAKTAN

U prošlom broju predstavili smo vam SGI Iris Indigo – jednu od UNIX radnih stanica firme koja je orijentisana na 3-D grafiku. Međutim, najveći deo UNIX grafičkih radnih stanica na svetskom tržištu prodaje druga firma, kalifornijski „Sun”, čiji je vrtohlavi uspeh zasenio čak i imena kao „Apple” ili „Microsoft”. Praktično, „Sun” je osmislio i materijalizovao pojam grafičke radne stanice, njenog povezivanja i umrežavanja, i zato mu pripada zaslužno mesto u istoriji računarstva.

Od mnogobrojnih modela zasnovanih na SPARC porodici RISC mikroprocesora, nazvanih SPARCstations, u ovom broju ćemo prikazati jedan od najatraktivnijih i najkompaktnijih – SPARCstation IPX.

NA PRVI POGLED

SPARCstation IPX je, najpre, izuzetno lep računar. Atraktivan dizajn je i inače odlika većine UNIX grafičkih radnih stanica, ali Sun modeli se baš posebno ističu. Izvanredni dizajn SPARCstation serije je, kao i NEXT stanice i NEC FG monitori, delo poznate dizajnerske kuće „FrogDesign”. Za razliku od SPARCstation 2, koji je u sada već čuvenom i naveliko kopiranom „pizza-box” kućištu, krem kućište IPX je, kao što na slici vidite, „lunch-box” oblika. Pri tome je usklađena identična rupičasta tekstura prednje strane kao i na SPARCstation 2. interesantna činjenica je da se „Sun”-ovi spojni dodaci (diskovi, trake, CD-ROM, itd.) po vertikali uklapaju sa SPARCstation 2, a po širini sa IPX-om, tako da je lepo uklapanje moguće po vertikali ili po horizontali.

SPARCstation IPX u svom malenom kućištu (9.6"x10.4"x4.6") sadrži osnovnu ploču sa 40 MHz SPARC procesorom i koprocesorom, 64 KB keša, 16 do 64 MB RAM, 1.44 MB flopi i 257 ili 424 MB hard disk. Na osnovnoj ploči se, pored procesorskog dela i memorije, nalazi i ultrabrz GX grafički akcelerator koji postiže 480.000 2-D i 310.000 3-D vektora u sekundi – dovoljno za *real-time* 3-D manipulaciju žičanih modela, što i jeste jedna od namena IPX-a. Te performanse će za PC svet ostati san čak i sa najbržim *local bus* grafičkim karticama sadašnje generacije. Sam procesor, inače, postiže pristojnih 24 SPECmarks (SPECmark je, ukratko rečeno, VAX-MIPS kome se može verovati, sabran iz više realnih aplikacija). Te performanse u ne-grafičkim aplikacijama predstavljaju brzinu oko 30% veću od 486DX2-66 klase u celobrojnom i oko 2 puta u FP delu (grafičke aplikacije postižu znatno veća ubrzanja čak i bez ikakvih dodatnih akceleratora, jednostavno zbog daleko naprednije RISC arhitekture). Sa svojih 136 registara opšte namene, simetričnim setom instrukcija i snažnom podrškom multitaskingu, SPARC generacijski tuče Intelovu 80x86 porodicu.

Memorija je u posebnim 33-bitnim (paritetni bit po reči) SIMM modulima, od kojih svaki ima kapacitet 4 ili 16 MB, zavisno od toga da li koristi 1 Mb ili 4 Mb memorijske čipove. Sun SIMM standardi u najvećem broju slučajeva ne odgovaraju PC ili PS/2 SIMM standardima, tako da ne možete vršiti razmenu memorija između njih. To inače važi i za ostale proizvođače grafičkih radnih stanica.

Na polju interfejsa, IPX je solidno opremljen: ugrađeni su SCSI-2, Ethernet, 2 X.25 kompatibilna RS-232/RS-423 porta sa posebnim konektorima koji se spolja izvode u standardne preko adaptera, audio I/O portovi kao i portovi za FDD, tastaturu i miša. Na ploči su i dva Sbus ekspanziona slota brzine 80 mega-

Nebojša Novaković

bahta u sekundi. Sbus pločice su znatno manjeg formata nego AT, na primer, ali zato postižu znatno bliskije povezivanje sa osnovnom pločom nego što je to slučaj kod PC-ja, izuzev u *local-bus* verzijama. Na raspolaganju je nekoliko stotina Sbus kartica najrazličitijih namena.

Ugrađeni zvučni potencijali nisu ništa posebno u odnosu na IRIS Indigo, ali su u svakom slučaju izvrsni u poređenju s onim što još uvek standardno nudi PC. Audio ulaz i izlaz „telefonskog” kvaliteta je podržan sa 8-bitnim uzorkom i 8 kHz uzorkovanjem. Periferije podržane na audio-portu su dinamički mikrofon, kasetofon, slušalice i spoljno pojačalo.

SA SOFTVERSE STRANE

SPARCstation koristi SunOS, veoma raširenu „Sun”-ovu verziju UNIX-a, koja je odnedavno doživela novu mladost pod imenom Solaris. Novi Solaris 2, koji objedinjuje puni multiprocesorski UNIX V.4 sa dodacima i izvrsno OpenWindows grafičko okruženje, raspoloživ je sada i za Intelove 32-bitne 80x86 procesore.

Ukratko, rečeno, Solaris sa svojih 130 MB koje korisnik dobija na CD-ROM, je najviše „Mac-like” UNIX i, u isto vreme, treći operativni sistem (prvi u UNIX svetu) po rasprostranjenosti na svetu posle DOS-a i MacOS-a.

OpenWindows je „Sun”-ova mrežna prozorska okolina koja podržava puni multitasking i distribuirano procesiranje više poslova, za više korisnika, preko više računara u mreži – aplikacije mogu raditi u realnom vremenu na nekom udaljenom računaru u mreži a biti prikazivane i upravljane lokalno sa mišem. Uz OpenWindows dolazi DeskSet, set priručnih aplikacija nalik onome u Windows 3.1, koji sadrži: File Managerom, TEXTEDIT, Debug, Calendar Manager, sat, Calculator, Print alatku, Mail, Snapshot - hard-copy ekrana, Icon Editor, Performance meračima, kao i kompletan on-line idiot-proof udžbenik. Za razliku od DOS-a, Solaris ima ugrađen ceo mrežni operativni sistem, napredniji i snažniji i od Novell Netware 3.1 – besplatno, bez obzira na broj korisnika! Uostalom, Sun NFS (Network File System) je svetski standard za mreže.

Za razliku od „Silicon Graphics”-a, „Sun” radne stanice nisu isključivo usmerene na grafiku – one su jednako orijentisane i na tržište brzih i kompaktnih database i file servera sa višekorisničkim i mrežnim bazama podataka (ORACLE, Informix, Ingres...), kao i snažnih višekorisničkih računara („supermikro”) sa i više stotina terminala. Slična gradacija takođe postoji i kod IBM (Bull takođe), HP i DEC.

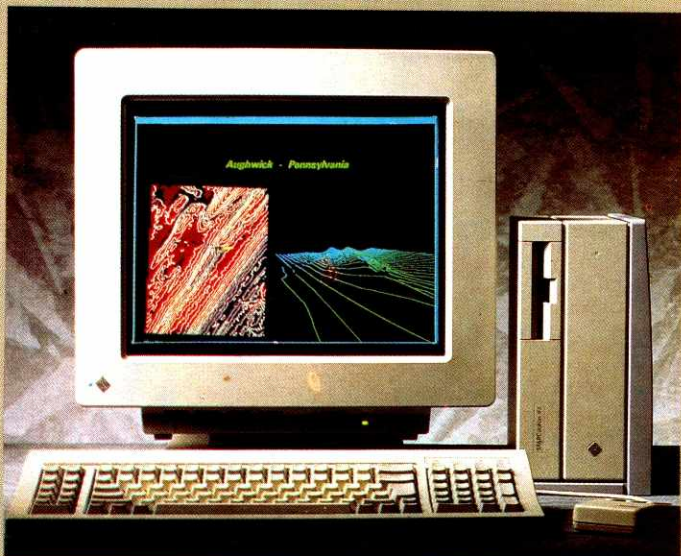
„Sun”-ove SPARC mašine su u svetu trenutno Nr.1 na polju razvoja softvera. Kao karakterističan primer navedćemo da se AutoCAD i ORACLE razvijaju na Sun SPARC stanicama, a posle prebacuju na sve ostale platforme. Praktično sav PC-softver po PC cenama a znatno boljim performansama radi na SPARC platformama, a tu su još i stotine ultraozbiljnih naslova sa naučno-inženjersko-grafičkih područja radnih stanica – gotovo 5,000 proizvoda.

POGLAD UNAPRED

Zahvaljujući novom TI SuperSPARC procesoru, osetno bržem i od još nerođenog 586, nova generacija ultrabrzih multiprocesorskih SPARC radnih stanica i servera već je ugledala svetlo dana. Postižući preko 300 MIPS iz samo 4 paralelna procesora, SPARCstation 10 sabija celih 512 MB RAM i preko 4 GB disk kapaciteta u „pizza-box” kućištu. A tokom dolazeće godine pojavice se i prve prave 64-bitne SPARC stanice...

Korisna adresa:

AEROINŽINJERING
Bulevar Nikole Tesle 42A
Tel: 698 600, fax: 691 882



Preko 300 MIPSa: Četiri paralelna procesora, 512 MB RAM-a i 4 GB u tvrdim diskovima



MI VAM NUDIMO IZLAZ

Oblast računara sve više liči na **lavirint**. Koji hardware, koji software, namenski program, prateća literatura, nameštaj, kako sve to povezati u sistem koji funkcioniše?...

CET COMPUTER SHOP nudi Vam hardware različitih proizvođača. Mi smo ovlašćeni dileri svetskih proizvođača software-a (**Microsoft, Borland, Autodesk, Lotus...**). Uz našu pomoć možete

kupiti bilo koji licencni software. **CET COMPUTER SHOP** nudi veliki izbor domaće i strane literature iz ove oblasti, kao i potrošnog materijala i rezervnih delova. Kod nas možete da izaberete i kupite specifičan nameštaj. U **CET COMPUTER SHOP-u**

imamo još nešto... Mi posedujemo **TIM KVALIFIKOVANIH LJUDI** koji će Vam pomoći da formirate sistem koji funkcioniše.



Computer Equipment & Trade

11000 Beograd • Skadarska 45 • Tel: 343-043

SILA SLICI DAJE UBRZANJE

Generisanje i prikaz slika spada u najteže poslove u grafičkim aplikacijama (od računara se zahteva i manipulisanje ogromnim količinama podataka i intenzivno računanje). Zato ne čudi što proizvođači PC hardvera sve teže prate zahteve u kompjuterskoj grafici i multimediji (*true color, photo realistic, real time visualisation* – prirodan izgled predmeta po boji i obliku, uz trenutnu pojavu slike na ekranu). Grafička kartica *Superstation 3D GB860*, krunski model u čitavoj familiji grafičkih akceleratora firme Hercules, ima ambiciju da na PC platforme donese prirodnu sliku, brzinom – superkompjutera.

Jovan Regasek

SUB konektora. Prebacivanje sa jednog izvora na drugi vrši se preklopnikom sa prednje strane monitora. Preko video genloka kartica se može povezati sa NTSC i PAL video rikorderima, za snimanje grafike i animacija na video traku.

WINDOWS: ŽIVE SLIKE

Kod prikaza u DOS-režimu između ovih kartica, naravno, nema nikakvih razlika. Ali, prelazak u *Windows* je, doslovce, preproudio jedan (ni po čemu izuzetan) kolor monitor od 20 inča – bez obzira na odabranu rezoluciju, boje su bile življe, a likovi slova mnogo oštrij. *Windows 3.1* je jedna od dve ciljne aplikacije za ovu karticu i zato je softverski veoma bogato podržana – na listi drajvera nalazi se petnaest grafičkih formata za rezolucije od 1280x1024 tačkica u 256, do 1024x768 tačkica u 32.768 boja i do 768x576 tačkica u 16,7 miliona boja. Prelazak sa režima na režim, u verziji koja je nama bila na raspolaganju, obavlja se na mukotran *Windows* način, preko *setup/options/display* menija sa obavezanim restartom programa. Nedavno smo saznali da je u pripremi nova verzija drajvera, koja omogućuje izbor režima u letu, na pritisak tastera, bez napuštanja aplikacije.

U softverskom paketu nalazi se i program *Image-Prep* za obradu digitalnih fotografija u boji. Ova alatka (koju toplo preporučujemo grafičarima i dizajnerima) na ubedljiv način pokazuje kolorističke mogućnosti, pa i brzinu *Superstation 3D* kartice. Imali smo utisak da bi se kapima rose na rascvetaloj crvenoj ruži mogla ugasiiti žed – toliko su bile stvarne. Za brzinu kažemo „pa i“, jer je brzina relativna i krajnje varljiva stvar. Na male dobitke u brzini korisnik se brzo navikava, a „zvučni zid“ se, izgleda, ne probija baš tako

lako, čak ni kada vozite mašinu kojom biste to mogli da učinite.

TESTOVI: PRAKSA JE NEŠTO DRUGO

Mašinski i aplikativni testovi *Superstation 3D* kartice daju spektakularne i sasvim očekivane rezultate – procesorsku snagu koja je ugrađena u ovu „superstanicu“ mogu da opravdaju samo drastični dobitci u brzini. Prema testovima *Windows Benchmark 2.0* (PC Magazin), kartica je u upisu blokova podataka (BitBlit) brža 8, u crtanju vertikalne linije 7, a u popunjavanju pravougaonika: bojom 2 puta od VGA adaptera u režimu „minimalna rezolucija / maksimalan broj boja“, odnosno 20, 8 i 7 puta u režimu „maksimalna rezolucija / minimalan broj boja“. Ovi rezultati, koji se odnose na najelegantnije grafičke operacije, vredni su svakog poštovanja, ali ih ne treba uzeti zdravo za gotovo.

Slični rezultati se dobijaju i u programu *WinTach*, koji je razvio Texas Instruments za testiranje svojih grafičkih procesora u stvarnim uslovima. Program

Slika 2 Windows brzinomer

	ET4000 Superstation	Faktor
Obrada teksta	4,91	10,03
CAD / crtanje	7,38	63,38
Radne tabele	4,96	14,02
Slikanje	6,55	12,90
Prosečna brzina	6,06	25,08
		4,13 puta

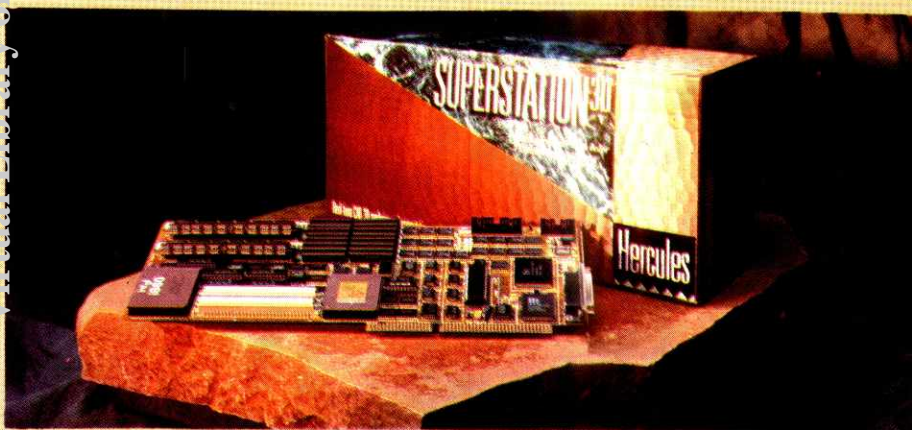
Aplikativni testovi pokazuju veliki faktor ubrzanja, ali se ono ne oseća u stvarnoj primeni, jer *Windows* aplikacije ne mogu da koriste hardverske pogodnosti kartice *Superstation 3D*

obuhvata četiri vrste testa – obradu teksta (sa intenzivnim formatiranjem i promenom pisama), radne tabele (sa intenzivnim skrolovanjem, modifikovanjem i grafičkim prikazom podataka) *CAD/Draw* (sa intenzivnim crtanjem, zumiranjem i pomeranjem crteža) i slikanje (od osnovnih grafičkih elemenata do kompletne slike) – a brzinomer je na našoj test-mašini (486/33 MHz sa procesorskim kešom 256 K i 8 MB RAM-a) pokazivao, u zavisnosti od izabrane rezolucije, uvek preko 25 jedinica. Pod potpuno istim uslovima, ET4000 kartica u susednom slotu je razvijala prosečnu brzinu od samo 6 jedinica. U svim aplikacijama *Superstation 3D* je bio 2 do 3 puta brži, osim na CAD testu, gde je umakao u brzini za čitavih osam puta. Ne znamo koliko je ovaj brzinomer pouzdan, ali su i ovi rezultati, s obzirom na namenu kartice i na ugrađene resurse, sasvim očekivani i – logični.

Nema, međutim, bojazni da će se korisniku, kad startuje neki *Windows* program, od ove brzine zavrteti u glavi, jer *Windows* ne ume da papučicu gasa pritisne do daske. Korisnici programa *CorelDraw*, verovatno najpopularnijeg paketa za grafički dizajn u *Windows* okruženju, biće razočarani kada vide da *Superstation 3D* poznatu demo-sliku „Koliba pod snegom“ crta, umesto očekivanih osam puta brže, nešto (5 sekundi) sporije od najstandardnije VGA kartice (97:88 sekundi). Ni druge popularne aplikacije, kao što su *Ventura* ili *Excel*, koje vapiju za brzim prikazom, neće procvetati sa ovom karticom. U čemu je tajna? U tome što *Windows* aplikacije nisu prilagođene za ovu grafičku mašinu. Umesto da kartici prosledi listu poslova i njoj prepusti sve radnje oko generisanja i osvežavanja slike, *CorelDraw* sam crta sliku, oslanjajući se na procesor u računaru, da bi je poslao samo na prikaz, kao što to čini i sa bilo kojim drugim video adapterom. On, dakle, vozi „porše“ na isti način kao i

Svojim izgledom *Superstation 3D GB860* izaziva, u najmanju ruku, strahopoštovanje. Dva ogromna procesora u paraleli (Intelov 64-bitni i860 RISC procesor na 33 MHz i Texas Instruments 32-bitni grafički procesor TMS34020 na 32 MHz), 16 MB programske, 2 MB brze video memorije i obilje pratećih čipova koji pokrivaju celu površnu šesnaestobitne ISA kartice punog PC formata nagoveštavaju da se radi o grafičkoj mašini za najteže CAD poslove. Izbor i uparivanje procesora (i860 ima pajplajn arhitekturu za 3D grafiku i 3D instrukcije u hardveru i premešta 50 megapiksela u sekundi, dok TMS34020 ima 2D instrukcije i raster u hardveru, a piksele premešta brzinom od 20 megapiksela u sekundi) svedoče o želji firme Hercules da ponudi totalno rešenje na polju grafičkih akceleratora. Kartica obezbeđuje obilje grafičkih formata do 1280x1024 tačka i do 16,7 miliona boja, uz osvežavanje slike brzinom većom od 72 puta u sekundi. Nedavno smo imali zadovoljstvo da sa ovom karticom svakodnevno radimo gotovo dva meseca i tako provećimo kako RISC procesorska tehnologija funkcioniše u praksi – i to na najtežim poslovima.

Superstation 3D je projektovana da radi kao grafički akcelerator u mašinama 386 i više klase, i ne može se pokrenuti, čak ni pod DOS-om, dok se ne izvrši softverska priprema kartice i mašine u koju se ugrađuje i ne instaliraju TIGA drajveri. To znači da se ugradnjom ove kartice ne možete odrediti usluga standardnog grafičkog adaptera VGA klase. Za VGA adapter je na ovoj „superstanciji“, kao i kod svih specijalizovanih grafičkih kartica, predviđena VGA prevodnica (*pass-through*), koja sistemu obezbeđuje potpunu softversku kompatibilnost, pri čemu obe kartice bez poteškoća dele jedan monitor. *Superstation 3D* se povezuje sa monitorom preko BNC priključaka, sa posebnim vodovima za RGB boje i signale za sinhronizaciju slike, a VGA (u našem slučaju na bazi čipa ET4000) na uobičajeni način, preko standardnog D-



Slika 1

	Standardna VGA	Superstation 3D	Superstation 3D	Faktor ubrzanja	Faktor ubrzanja
	640 x 480	640 x 480	1280 x 1024	640 x 480	1280 x 1024
	16 boja	16,7 mil. boja	256 boja	16,7 mil. boja	256 boja
BitBlit	348.364	3.009.571	6.704.182	8,64 puta	19,24 puta
Vertikalna linija	303.352	2.110.253	2.604.954	6,96 puta	8,59 puta
Popunjavanje pravougaonika	3.328.859	6.365.623	2.174.7699	1,91 puta	6,53 puta

Brzina (piksela u sekundi) elementarnih grafičkih operacija u režimu „prirodne boje“ (16,7 miliona) i „maksimalna rezolucija“ (1280 x 1024); poslednje dve kolone daju faktor ubrzanja

ComTrad

the **ART** of
computer making

ComTrad 386SX/25

Procesor 386SX/25MHz, 2MB RAM

Floppy uređaji 1.2MB i 1.44MB

Tvrđi disk 105MB IDE

Dva serijska i jedan paralelni port

SVGA grafički adapter 1024x768/512kB

SVGA kolor monitor

Desktop kućište, ASCII tastatura, miš

2,095.-

ComTrad YU • Genex apartmani, V. Popovića 6, Beograd
Telefon: (011) 222-41-51, 222-26-51 • Fax: (011) 222-41-39
Radno vreme: 9-17, ponedjeljak-petak



ComTrad
COMPUTERS

SPRINT
Novi Sad
Novosadskog
sajma 35
Tel. (021)
623-717
Fax (021)
623-901

ComTrad Shop
Beograd
Cvijičeva 104
Tel. (011)
752-663
Fax (011)
752-663

INFOTRADE
Pristina
Hekurudha 7
Tel. (038) 25-830
Fax (038) 25-822

COMTRAD
KRAGUJEVAC
Kragujevac
Kragujevačkog
oktobra 102
Tel. (034) 60-336
Fax (034) 67-117

MANIR
Ruma
Grobljanska 2
Tel. (022) 421-265
Fax (022) 421-265

OBRAČUN KOD O.K. LOKALA

Ovaj članak je odgovor na tekst D.V. Veselinovića „Lokal je stigao” u „Računarima” 88. Članak je primer da izreka „Lies, damn lies, benchmarks” ima puno opravdanja. Da bi se uradio test bilo koje komponente računara, potrebno je veliko teoretsko predznanje o tome šta se testira, a naročito kako se testira i šta znače dobijeni rezultati.

U prikazu osnovne ploče *Headland HT340* sa ugrađenim *local-bus* VGA kontrolerom HT216, dominira razočarenje autora, jer je očekivao značajno bolje rezultate u odnosu na druge komponente na testu (a posebno WD 90C30 karticu). Uzroci njegovog razočarenja su u nepoznavanju mogućnosti *local-bus* tehnike, u loše urađenom testiranju i pogrešno protumačenim rezultatima testiranja.

Pre nego što pređemo na analizu rezultata, razmotrimo kakve su teoretske mogućnosti *local-bus* tehnike i gde su ograničenja.

LOKALNI-(auto)BUS ILI FERARI

Na IBM AT osnovnoj ploči postoji nekoliko magistrala; npr. magistrala za lokalne periferije, magistrala za lokalnu memoriju, sistemska ISA magistrala, a ove se dalje mogu deliti na adresne, kontrolne i magistrale podataka. „Local-bus” označava magistralu koja se dobija priključivanjem direktno na spoljne linije procesora, bez kontrolera i arbitera magistrale, tako da vežna logika minimalno usporava saobraćaj na magistrali. Pristup lokalnim resursima je lakši i brži, i radi se o nom brzinom kojom to omogućavaju procesor i resurs.

Koja je teoretska razlika između ISA i *local-bus* magistrale? Ako se radi samo po standardu (što je preporučena metoda), vreme ciklusa na ISA magistrali je minimalno 250 ns (uz upotrebu 0WS signala), a magistrala je širine 16 bita. Za lokalnu magistralu teoretsko minimalno trajanje ciklusa je 30 ns za 80486DX-33 procesor, a širina magistrale je 32 bita. Lako se izračunava da je teoretski maksimalni odnos brzina ove dve magistrale 1 : 16,5.

Poznato je da je minimalno vreme pristupa (po deklaraciji proizvođača) za memorije koje se koriste na VGA karticama obično 70 – 80 ns. Kada se ne bi koristile specijalne tehnike pristupa, minimalno vreme memorijskog ciklusa koje bi se moglo postići bilo bi oko 150 ns. Kad se uzme u obzir i kašnjenje u kolima adresnih dekodera, bafera i slično, taj broj se povećava na minimalno 170 ns. Ako sada taj broj uporedimo sa 250 ns, što je trajanje ciklusa na ISA magistrali, vidimo da se neka spektakularna ubrzanja ne mogu postići.

D.V. Veselinović je radio sa taktom od 13.3 MHz na ISA magistrali (trajanje ciklusa 150 ns). U tom slučaju je odnos takta magistrala „33 / 13.3 = 2.48”. Na osnovu ovog broja, D.V. Veselinović očekuje ubrzanje oko 1.9 puta. Ovako složenom „analizom”, metodom deljenja 3 kruške i 2 jabuke, vrlo retko se mogu dobiti oke tačniji rezultati. O zaključcima na osnovu takvih rezultata ne treba ni govoriti. Moguće ubrzanje treba računati posebno za uobičajene grafičke operacije, a pri tome uzeti u obzir i kako procesor 80486 može

Dragan Dumelić

realizovati ove funkcije, kao i vreme potrebno za osvežavanje ekrana.

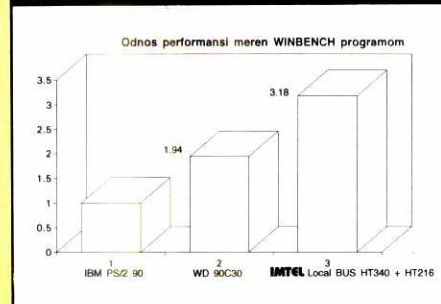
Posebna priča su mogućnosti procesora 80486. Iako je trajanje takta samo 30 ns, treba pokazati da procesor može izgenerisati neki koristan podatak koji bi se tom brzinom i prenosio. Uobičajeno je da se najbržim instrukcijama smatraju STOSW i MOVSW. Kratak pogled u tabelu vremena izvršavanja pokazuje da se ove instrukcije izvršavaju za minimalno 5, odnosno 7 taktova. Ode sanak pusti o letoćoj grafici na lokalnoj magistrali! Iako može da se govori i o problemima optimizacije koda za 80486 procesor, očigledno je da spektakularnog ubrzanja neće biti.

Veći dobitak se može postići time što je lokalna magistrala 32-bitna, ali su za sada vrlo retki VGA kontroleri koji ovo podržavaju. Pojavili su se u poslednjih

Greške u članku „Lokal je stigao”

1. Standardna sabirnica računara ne postoji, već sistemska, koja može biti ISA ili EISA (1.2)
2. AT Bus kartica ne znači ništa (2.6)
3. Linearna veza između procesora i video podsistema ne postoji; možda direktna veza, dok je linearna zavisnost (7.1)
4. Takt od 50 MHz nije problematičan, ploče sa tim taktom se teže prave i relativno su skuplje, ali je ta tehnologija savladana. Te ploče obično imaju takt od 50 MHz, dok ploče sa procesorom od 33 MHz obično imaju takt od 66 MHz. (2.9)
5. To što je „referentni oscilator 60 MHz” ne znači da su u obzir uzeti VESA modovi niti ima bilo kakve veze sa VESA standardom. (5.12)
6. To što je Music čip brz ne piše se sa 80 ns već 80 MHz (5.16)
7. Grešaka ima još, ali se mogu podvesti pod drugi deo.

Slika 1



Odnos performansi meren Winbench programom

nekoliko meseci i očekujemo da ih testiramo za nekoliko nedelja.

Pokažimo samo još neke smernice, koje bi omogućile realnije sagledavanje problema. Razvijeno je dosta tehnika za ubrzanje rada sa video memorijom. Nabrojamo nekoliko osnovnih:

1. Upotreba memorijskih kola koja omogućavaju stranični pristup. Kada se prvi put pristupa podatku koji je u okviru jedne stranice, obično oko 2000 memorijskih lokacija, pristup traje oko 170 ns. Ako se ponavlja pristup lokaciji unutar iste stranice, vreme ciklusa se može značajno smanjiti na približno 80 ns.
2. Upotrebom „cache” sistema za prikazivanje slike, ostvaruje se rasterećenje video memorije. Podaci se očitavaju u blokovima, tako što se koristi stranična tehnika očitavanja memorije. Između čitanja ovih blokova, memorija je potpuno slobodna za korišćenje.
3. Upotreba „cache” sistema za pristupe procesora. Svaki upis se vrši u interne registre VGA kontrolera, a stvarni upis u memoriju kad je to moguće. Ova tehnika funkcioniše samo kad se podaci ne prenose u blokovima.

Kakva ubrzanja u radu sa VGA karticom možemo očekivati u slučaju korišćenja *local-bus* tehnologije na primeru nekih češćih grafičkih operacija?

1. Upis bloka podataka u video-memoriju (npr. za brisanje prozora). Jasno je da je ovde ograničenje brzina procesora. Uz upotrebu stranične tehnike rada, može se očekivati minimalno vreme pristupa oko 80 ns, ali procesor ne može održati korak, pa je očekivano vreme ciklusa barem 120 ns.

2. Crtanje linije. Ovo je mnogo složenija operacija od direktnog upisa u video memoriju. Podrazumeva izvršavanje algoritma za crtanje (obično Bresenham) koji, kad se uđe u petlju crtanja, ima u proseku oko 5 mašinskih instrukcija, dva pristupa memoriji i jedan pristup i/o registru. Kad se uzme u obzir da se u praksi obično nalazi i poziv funkcije, razna testiranja oblasti crtanja („clipping”), onda se pokazuje da je približno jednako vreme koje se utroši na rad procesora i vreme potrebno za pristupanje VGA kartici. Upotrebom *local-bus* tehnike može se smanjiti samo ovo drugo vreme. Odmah se može videti da nisu moguća neka spektakularna ubrzanja za ovu grafičku funkciju (inače veoma traženu u programima za crtanje, kao npr. AutoCAD) i da ubrzanje može ići samo do približno 40 %.

3. Ispis teksta, rasterske operacije. Obično se koriste metode kopiranja dela video ili operativne memoriji, samo što sada, osim pristupa, treba uraditi još i neku logičku operaciju. Očigledno je da je ovde dobitak manji nego kod npr. brisanja memorije.

A UPALIŠ MOTORKU!

Postoji dosta teorija o tome kako treba vršiti testiranje komponenti računara. U konkretnom slučaju, VGA kartice treba testirati specijalizovanim programom, uvek u kombinaciji sa pripadajućim veznikom.

U ovom slučaju su korišćeni gotovi programi *Winbench 2.51* za *Windows* okruženje i *Gperf* za *AutoCAD* (ovi drugi nisu prikazani u tekstu). Zbog sprostosti ISA magistrale, danas su sve popularniji VGA grafički akceleratori. Njihova prednost je brzina, a mana što za svaki program mora postojati veznik, koji koristi specijalne mogućnosti akceleratora. Bez upotrebe veznika, dobici su mali ili nikakvi. Na žalost, rezultati u dijagramu na slici 3, str. 26, Računari 88, su rađeni u *Windows 3.0* okruženju, bez odgovarajućeg veznika i u rezoluciji 640x480 tačaka sa 16 boja. Odmah se prisećamo starog vica o drvosečama, kada Mujo na kraju kaže: A, upališ motorku!? Tako možemo reći i D.V. Veselinoviću: A, pokreneš veznik!?

Kad se uzmu u obzir stari rezultati i dobijeni odnosi pomoću *Winmark* testa u *Winbench 2.51*, može se nacrtati korigovana tabela, koja izgleda znatno drugačije. Kada se izvrši korektno testiranje i upotrebi odgovarajući veznik u nekoj boljoj rezoluciji, vidi se da je

BEOKURIR

GRADSKIE I MEDUGRADSKIE KURIRSKIE USLUGE - OD VRATA DO VRATA

Beograd

011

402 538



EKSPLOZIJA BOJA

Polako ali neizbežno, personalna informatika prelazi iz crno-belog u obojeni svet, a monitor u boji danas predstavlja standardnu opremu svake naprednije PC konfiguracije. Nakon uporednog testa monitora od 17 inča u prošlom, koji predstavljaju najbolji kompromis za korisnike *Windows-a*, u ovom broju dajemo prikaz deset modela od 14 inča, koji su namenjeni najširem mogućem tržištu.

U SREDINI: ADI Microscan 3E+

Kvalitet slike: Uočljiva greška u konvergenciji proizvodi kod ovog monitora (sa ekranskom maskom rastera 0.28 mm) neoštre ivice i to prvenstveno u srednjem delu ekrana. Utisak popravljaju dobra geometrija slike i tačan prikaz boja. Ali, pošto je stabilnost slike ispod proseka i nedostaje dugme za demagnetizaciju, 3E+ je ovde dobio ocenu „zadovoljava“.

Veličina slike: Od celokupne 14-inčne katodne cevi, kod modela 3E+ je vidljivo 13.8 inča, te se on po tome nalazi uz bok monitoru NEC 4FG, koji koristi katodnu cev od 15-inča. Na žalost, korisna dijagonala iznosi samo 11.9 inča, tako da je slika sa površinom od 431 cm² ovičena širokim crnim okvirom. Zato je ocena samo „zadovoljava“.

Ergonomičnost: Sa svojom maksimalnom ekranskom frekvencijom od 48 kHz, *Microscan* sliku rezolucije 800x600 prikazuje 76 puta u sekundi. Pri rezoluciji od 1024x768 tačaka, uspeva da izvuče samo 60 slika u sekundi. Slika u standardnoj VGA rezoluciji prikazuje se bez treperenja, sa teoretskim maksimumom od 95 Hz. Ovaj monitor ispunjava i uslove koje postavlja MPR II norma o zračenju, tako da je zaista ergonomičan. Ocena: „dobar“.

Podešavanje: Devet tastera, koje sakriva mali poklopac ispod okvira ekrana, imaju više estetsku nego korisnu svrhu. Pomoću njih se regulišu vertikalna i horizontalna pozicija slike i njena veličina. Deveti taster (Reset) koristi se za ponovno postavljanje fabrički odabranih vrednosti. Na zadnjoj strani monitora nalazi se deo rezervisan za servisiranje, preko koga serviser može izvršiti dodatne korekture pojedinih parametara monitora, ali sam korisnik ovom delu ne može pristupiti.

Tasteri su dovoljno veliki i razmaknuti jedan od drugog, tako da je isključena mogućnost nehotičnog pritiska na pogrešan taster. Točkici za podešavanje kontrasta i osvetljenosti, kao i prekidač za uključivanje izdvojeni su od ostalih regulatora izvan poklopca i zgodno postavljeni. Priručnik ima 12 strana i dovoljno opširno opisuje monitor. Ocena: „zadovoljavajuće“.

Cena/performance: Distributer ovog monitora je odredio cenu od 1251 DEM, što je prikladno i zaslužuje ocenu „dobar“. U prodaji se ovaj monitor nudi za oko 1100 DEM. I za ovu cenu ostaje ocena „dobar“.

Komforno podešavanje: Eizo T240i-M

Kvalitet slike: Trinitron katodna cev sa rasterom od 0.25 mm briljira jasnim prikazom. I greške u konvergenciji su u prihvatljivim granicama, a javljaju se ili u gornjem ili u donjem delu ekrana, u zavisnosti od podešavanja pomoću regulatora konvergencije. Na celom ekranu nema praktično nikakvih izobličenja. Za

jedno sa monitorom *Diamond Pro 14* firme *Mitsubishi*, ovaj Eizo-monitor se u pogledu geometrije slike nalazi ispred svih ostalih testiranih modela. Nema zamerke ni na mešanje boja kod velikih površina. T240 je jedino „omano“ kod stabilnosti slike – dimenzije slike se izraženo menjaju pri naglim izmenama crnih i belih slika. Nedostaje i *Degauss-taster* (za demagnetizaciju). Zato je ocena smanjena na „dobar“.

Veličina slike: I T240i-M prikazuje manje nego što bi zaista mogao. Od ukupno 13.4 inča, za test-sliku je preostalo samo 12 inča. 433 cm² donela su ocenu „zadovoljava“.

Ergonomičnost: Sa velikodušno odmerenom ekranskom frekvencijom od maksimalno 60 kHz, T240i-M prikazuje sliku u rezoluciji 800x600 do 95 puta u sekundi. Pri rezoluciji od 1024x768, frekvencija ponavljanja slike je oko 75 Hz. Apsolutno bez treperenja je prikaz u rezoluciji 640x400 tačaka, gde bi ovaj monitor mogao prikazivati sliku 135 puta u sekundi, ali današnje grafičke kartice to još ne mogu da podrže. T240i-M ispunjava i MPR II normu, tako da je u ovoj oblasti zaslužuje ocenu „vrlo dobar“.

Podešavanje: Eizo je ovom svom modelu podario sve kontrolne instrumente koje poseduje i njegov veći rođak, 17-inčni F550i-M. Nijedna želja u vezi sa podešavanjem optimalne slike neće ostati neuslišana. Moguće je korigovati i samu konvergenciju, kao i boje. Samo podešavanje vrši se preko jedinstvenog regulatora, dok se funkcije koje se podešavaju prethodno biraju preko tastera. To je verovatno najjednostavnije i najkomfortnije rešenje za ovakav posao. Moguće je memorisati 32 podešena režima. Kao i kod oba veća monitora ove firme, kontrolni instrumenti su suviše podvučeni pod sam ekran. Prateće uputstvo na 22 strana je iscrpno i pregledno. Ocena: „dobar“.

Cena/performance: Ovaj Eizo-monitor uveliko prednjači po pitanju kvaliteta slike, ergonomičnosti i podešavanja. Ali, preporučena cena od 2268 DEM je previsoka čak i za tako dobar 14-inčni monitor. Ocena: „zadovoljavajuće“. Srednja cena u prodavnicama se kreće oko 1700 DEM, što ovom kvalitetnom monitoru donosi najvišu ocenu: „vrlo dobar“.

Tri frekvencije: Escom 2824

Kvalitet slike: Nepravilnosti u konvergenciji gotovo da i nema na ekranu ovog monitora, što rezultuje jasnim i ostrim prikazom. ES 2824 se uzdiže iznad proseka i sa geometrijom slike. Problemi se javljaju kod prikazivanja jednobojnih površina, kada se jasno vide senke. Dolazi i do razlivanja boja usled namagnetisanosti katodne cevi, dok dugme za demagnetizaciju nije ugrađeno.

Prilikom testiranja geometrije slike uočena su blaga odstupanja od idealno kružnog oblika kod obe desne kružnice na ekranu. Pošto nema zamerki na stabilnost slike, ocena za Escom-ov monitor je u ovoj kategoriji „dobar“.

Veličina slike: Ovaj Escom-ov monitor se sa korisnom dijagonalom od 11.8 inča nalazi iza ADI-modela. Vidljivo je 13.4 inča. Za površinu prikaza od 419 cm², ocena je samo „dovoljan“.

Ergonomičnost: Teoretski, ovaj monitor (sa najvišom ekranskom frekvencijom od 48.19 kHz) prikazuje sliku 76 puta u sekundi, pri rezoluciji 800x600 tačaka. Pri rezoluciji od 1024x768 tačaka, ovaj monitor uspeva da prikaže sliku samo 60 puta u sekundi. Iako slabo zrači, zadovoljava samo staru normu MPR I. Ocena: „zadovoljavajuće“.

Podešavanje: ES 2824 je jedini testirani monitor koji ne radi u opsegu frekvencija, već sa tri fiksirane frekvencije od 31.5 kHz, 35.5 kHz i 48.19 kHz. Zbog toga je izbor režima rada monitora moguć samo u uskim granicama. Osim toga, ovaj monitor nije radio sa grafičkom karticom korišćenom na testu (Mega-Eva) i njenim standardnim parametrima. Kartica je morala biti prekonfigurisana pomoću DIP-prekidača.

I manipulisanje prikazom je svedeno na minimum. Pored osvetljenosti i kontrasta, mogu se podešavati još samo horizontalna pozicija i veličina vertikalne slike. Utisak popravljaju to što su točkici za podešavanje laki za okretanje i zajedno sa prekidačem za uključivanje postavljeni sa prednje strane. Priručnik na šest strana pruža samo najneophodnije informacije. Sve ovo zajedno zaslužuje samo ocenu „dovoljan“.

Cena/performance: U ovoj kategoriji je monitor, koga direktno nudi Escom, dobio dve ocene, jer se njegove osobine mogu sagledati sa dva aspekta. Kao prvo, sa cenom od 798 DEM ovaj monitor je za 1200 DEM jeftiniji od NEC-ovog modela 4FG i za 900 DEM jeftiniji



Karakteristike	1	2	3	4	5
Kvalitet slike					
Veličina slike					
Ergonomičnost					
Podešavanje					
Cena/performance : proizvođač					
Cena/performance : trgovina					

U svim testovima se ovaj monitor sa reduciranim zračenjem našao u sredini. Cena mu je sasvim povoljna.



Karakteristike	1	2	3	4	5
Kvalitet slike					
Veličina slike					
Ergonomičnost					
Podešavanje					
Cena/performance : proizvođač					
Cena/performance : trgovina					

Monitor koji osvaja komfornim podešavanjem i dobrim kvalitetom slike. Dobro odmerena prodajna cena čini ga naročito interesantnim.

Escom ES 2824



Karakteristike	1	2	3	4	5
Kvalitet slike					♦
Veličina slike		♦			
Ergonomičnost			♦		
Podešavanje				♦	
Cena/performance : proizvođač					♦
Cena/performance : trgovina					

Koncepcija ovog monitora, i pored dobrog kvaliteta slike, nije ubedljiva. Na raspolaganju ima samo tri fiksne ekranske frekvencije.

Escom ES 2624



Karakteristike	1	2	3	4	5
Kvalitet slike					♦
Veličina slike		♦			
Ergonomičnost			♦		
Podešavanje				♦	
Cena/performance : proizvođač					♦
Cena/performance : trgovina					

Ovaj monitor ne reducira zračenje. Zato je pri ocenjivanju odnosa cena/performance dobio nižu ocenu.

Highscreen AS 57K



Karakteristike	1	2	3	4	5
Kvalitet slike					♦
Veličina slike		♦			
Ergonomičnost			♦		
Podešavanje				♦	
Cena/performance : proizvođač					♦
Cena/performance : trgovina					

Dobio je samo ispodprosečne ocene, a i zrači – zato mu ne pomaže ni niska cena, najniža od svih.

Mitsubishi Diamond Scan 14



Karakteristike	1	2	3	4	5
Kvalitet slike					♦
Veličina slike		♦			
Ergonomičnost			♦		
Podešavanje				♦	
Cena/performance : proizvođač					♦
Cena/performance : trgovina					

Niske frekvencije ponavljanja slike pri visokim rezolucijama gurnule su ovaj Mitsubishi-jev monitor u prikrajak.

niji od modela T240i-M firme Eizo. Pri tom ovaj monitor nudi kvalitetnu sliku, što je neobično u toj klasi cena. Sve to zaslužuje ocenu „vrlo dobar”. Sa druge strane, podešavanje je lošija strana ovog monitora, a naročito je zastareo koncept rada sa tri fiksirane ekranske frekvencije. Uz to ispunjava samo staru MPR I normu. Zato je, za ovaj aspekt, ocena „dobar”.

U crnom kućištu: Escom ES 2624

Kvalitet slike: Nejasne ivice i vidljive greške u konvergenciji (mestimične crvene senke, pre svega u gornjem delu ekrana) umanjuju kvalitet slike. Što se toga tiče, ES 2624 zaostaje za modelom ES 2824, iako je baziran na Trinitron katodnoj cevi sa finim rasterom (0.26 mm). Sa druge strane, ovaj model daje odličan prikaz boja i ima geometriju slike koja je gotovo bez zobljenja. Stabilnost slike je na prosečnom nivou, dok je dugme za demagnetizaciju uzaludno tražiti. Sve zajedno zaslužuje jedno „zadovoljava”.

Veličina slike: Dijagonala slike iznosi 12.1 inč, dok je dijagonala vidljivog dela katodne cevi 13.9 inča. 434 cm² korisne površine dobila su ocenu „zadovoljava”.

Ergonomičnost: Sa ekranskom frekvencijom od 57 kHz, ES 2624 dobro stoji u ovoj kategoriji. Maksimalne učestanosti ponavljanja slike su: 90 Hz pri rezoluciji od 800x600 tačaka i oko 70 Hz pri rezoluciji od 1024x768 tačaka. Šteta je što ovaj monitor nije „imun” od zračenja. Zbog toga, ocena je „dovoljan”.

Podešavanje: Što se tiče podešavanja, ovaj monitor upakovan u crno kućište nudi malo više od svog svetlijeg brata. Mogu se podešavati vertikalna pozicija slike i njena veličina po horizontali. Doduše, ovde je položaj četiri regulatora mnogo lošije izabran, jer su poprilično skriveni sa leve donje strane kućišta monitora. Veliki i lako pokretljivi regulatori za podešavanje osvetljenosti i kontrasta, kao i prekidač za uključivanje monitora postavljeni su sa prednje strane ispod katodne cevi. Priručnik na devet strana nije ništa opširniji

od onog namenjenog modelu ES 2824. Ocena „dovoljan”.

Cena/performance: Cena od 998 DEM za monitor koji nije zaštićen od zračenja i koji ima prosečan kvalitet slike je dovoljna samo za ocenu „dovoljan”.

Osrednji kvalitet slike: Highscreen AS 57K

Kvalitet slike: Inženjeri firme Vobis nisu baš najbolje podesili karakteristike ovom monitoru. Njegov ekran (sa rasterom tačaka od 0.28 mm) pokazuje jasnu grešku u konvergenciji, koja je naročito izražena u desnom delu ekrana. Zbog toga trpi i oštrina slike. Highscreen zaslužuje kritiku i zbog geometrije slike. Četiri test-kružnice vidljivo su se razlikovale od idealnog oblika. Donje su bile veće nego gornje. Uz to, gornje kružnice nisu bile baš sasvim „okrugle”. Što se tiče stabilnosti slike, tu su „sudije” izrekle samo dve reči: „krajnje nestabilno”, ali je zato razlivanje boja besprekorno. Naravno, mogućnost ručne demagnetizacije ni ovde nije na raspolaganju. Za kvalitet slike, ocena je „dovoljan”.

Veličina slike: U ovoj kategoriji se Highscreen monitor nalazi na samom začelju. Za upotrebu je ostavljena dijagonala od celih 11.4 inča, ili površina od 408 cm². Vidljiva dijagonala iznosi 13.4 inča. Ocena: „dovoljan”.

Ergonomičnost: Kao ni ES 2624, ni Highscreen-ov mališa ne ispunjava ni jednu švedsku normu o zračenju. Ekranska frekvencija mu je 60 kHz, što znači da ne koristi ni ergonomske frekvencije koje se teorijski kreću oko 95 Hz (800x600) i oko 75 Hz (1024x768). Ocena: „dovoljan”.

Podešavanje: Nikakvih iznenađenja ni u ovoj kategoriji nema kod Highscreen-a. Mogućnosti podešavanja ograničene su na najnužnije, npr. veličina slike se može menjati samo po vertikali. Srećom, svi regulatori su postavljeni sa prednje strane i označeni simbolima u boji. U priručniku za upotrebu se ističe da se radi o „inteligentnom ekranu” (!?). Ali kada u tom istom priručniku pročitate da se ovaj monitor ne sme koristiti „u vlažnom, prašnjavom, ili suvom (!) prostoru”, pomislite da ste uzeli pogrešnu knjižicu u ruke. Ovaj priručnik je inače zadovoljavajuću ocenu srušio na „dovoljan”.

KAKO IZRAČUNATI FREKVIJCU PONAVLJANJA SLIKE

IZRAČUNAJTE I SAMI

Proizvođači uglavnom ne spominju podatke o frekvencijama ponavljanja slike svojih *multiscan* (višefrekvencijskih) monitora u raznim rezolucijama, već radije daju oblast za sve frekvencije, od „minimalne” do „maksimalne” (npr. od 50 do 90 Hz). Uz pomoć jednostavnog računara, koji međutim nije opšte poznat, možete i sami proveriti koliku frekvenciju ponavljanja slike postiže neki monitor u određenoj rezoluciji. U tehničkim podacima za monitor pronađite njegovu maksimalnu ekransku frekvenciju. Na odgovarajući broj linija na ekranu dodajte još 5%, da biste uzeli u obzir eventualne proizvođačke tolerancije i linije koje elektronski mlaz prelazi pri povratku na vrh ekrana i slično. Frekvenciju ponavljanja slike za tu određenu rezoluciju sada možete izračunati ako ekransku frekvenciju podelite sa brojem ekranskih linija.

Primer: Monitor ima maksimalnu ekransku frekvenciju od 60 kHz. Koja mu je najviša frekvencija

ponavljanja slike pri rezoluciji od 1024x768 tačaka? Račun teče ovako: 60000 podeljeno sa 806 (768 plus 5%) daje kao rezultat 74.44. Dakle, monitor može pri rezoluciji od 1024x768 da prikazuje sliku 74 puta u sekundi, a verovatno i još malo brže.

Ipak, budite oprezni: da li će monitor pri visokim rezolucijama raditi sa ovakvim frekvencijama zavisi i od grafičke kartice. Starije grafičke kartice pri rezolucijama od 800x600 i 1024x768 često ne obezbeđuju rad sa ergonomskim frekvencijama – rade sa manje od 70 Hz. Dakle, ako razmišljate o kupovini monitora, ne zaboravite grafičku karticu. Naime, ukoliko želite da u visokim rezolucijama radite sa ekranom koji ne treperi, verovatno ćete morati da kupite novu karticu, što bi moglo da bude veoma skupo. Grafičke kartice koje podržavaju visoke frekvencije ponavljanja slike (od 80 Hz pa naviše) trenutno bez izuzetka imaju vrlo visoku cenu.

Mitsubishi Diamond Pro 14



Nepraktično podešavanje je ovaj, inače dobar monitor, unazadilo pri oceni odnosa cena/performance.

NEC Multisync 4FG



15-inčni Multisync odlikuje se odličnim kvalitetom slike i brzim podešavanjima. Bio je to jedan od najboljih monitora na testu.

Cena/performance: Ovde je slična situacija kao i kod modela ES 2824: sa jedne strane, cena od 698 DEM jeste mala, ali za te pare dobijate monitor koji se u svim testiranjima našao na začelju. A i zrači... Uprkos najnižoj od svih cena, sve to je samo „zadovoljavajuće“.

Jako treperi: Mitsubishi Diamond Scan 14

Kvalitet slike: Sa ovim monitorom koji koristi raster od 0.28 mm neće vam biti potrebne naočare za čitanje. Grafika i tekst se prikazuju sa oštrim ivicama i uz odličan prikaz boja. Što se toga tiče, ovaj model samo neznatno zaostaje za najboljim kandidatima na ovom testu. Greška u konvergenciji je minimalna. Mitsubishi prednjači i kada su u pitanju neizobličen prikaz slike, razilaženje boja i prikaz jednobojnih oblasti. Osrednje rezultate ovaj model je iskazao jedino kod stabilnosti slike. Nedostaje i dugme za demagnetizaciju. Sve u svemu – ocena „dobar“.

Veličina slike: Ni *Diamond Scan* ne koristi u potpunosti svoj ekran. Test-slika je imala dijagonalu od 11.5 inča, što je za 1.6 inč (ili 4 cm) manje od ukupne vidljive dijagonale. 412 cm² korisne površine dostajalo je samo za ocenu „dovoljan“.

Ergonomičnost: Sa maksimalnom ekranskom frekvencijom od 38.5 kHz, prikaz na *Diamond Scan*-u treperi kada se radi sa rezolucijom od 800x600 tačaka (frekvencija ponavljanja slike je 60 Hz, što je premalo za jedan moderan monitor). Ergonomska frekvencija ponavljanja slike od 76 Hz dobija se samo pri rezoluciji od 640x480 tačaka. Uprkos poštovanju norme MPR II, ovaj monitor je ovde dobio ocenu „manjkav“.

Podešavanje: Kao i nekoliko drugih testiranih monitora, i *Diamond Scan 14* omogućava podešavanje samo osnovnih stvari: osvetljenosti, kontrasta, veličine i pozicije slike. Šest točkića je postavljeno sa prednje strane i omogućavaju vrlo tačno podešavanje. Uz iscrpan priručnik, došlo se do ocene „zadovoljava“.

Cena/performance: Najveća mana monitora *Diamond Scan 14* je niska učestanost ponavljanja slike pri višim rezolucijama (60 Hz pri rezoluciji 800x600). On je, zapravo, upotrebljiv samo za rad u standardnoj VGA rezoluciji. Stoga se cena od 999 DEM može okarakterisati jedino ocenom „dovoljan“. Pošto je ovaj monitor tek stigao u prodavnice, još nema značajnijih smanjenja cene.

Iznad proseka: Mitsubishi Diamond Pro 14

Kvalitet slike: *Diamond Pro 14* je u ovoj kategoriji ocenjen pozitivno kao i monitor *Scan 14*. Čist prikaz na ekranu sa neznatnom greškom u konvergenciji ide ruku pod ruku sa savršenim prikazom boja. Tim više čudi nestabilnost slike pri brzim izmenama crnih i belih slika. Po geometriji slike, ovaj model se nalazi iza modela *Scan 14*. Leve kružnice su jajolike. Nema ni Degauss-tastera (za demagnetizaciju). Zato, jedva ocena „dobar“.

Veličina slike: Sa 437 cm² površine namenjene slici, *Diamond Pro* se nalazi ispred *Scana 14*, a na nivou ADI- i Eizo- monitora. Od 13.2 inča ukupne vidljive dijagonale, na raspolaganju je 12.1 inč. Ocena: „zadovoljava“.

ZRAČENJE MONITORA

RIZIČNO ILI NE?

Na početku jedna napomena: „Zračenje“ nije tehnički ispravan naziv za različite uticaje koji izviru iz monitora. Ipak, ovaj izraz se odomaćio, tako da ćemo ga i ovde, jednostavnosti radi, prihvatiti.

Zračenje monitora čovek niti vidi, niti čuje, niti ga može omlirisati. Neki ljudi ga i ne osećaju, dok se drugi, posle dugotrajnog rada ispred monitora, žale na glavobolju, upalu očiju ili čak na promene na koži. U SAD se još uvek ne poklanja preterana pažnja monitorima sa reduciranim zračenjem, ali su u Nemačkoj i Švedskoj oni već više od dve godine u žiži interesovanja. Jedno je izvesno – ili su tamo kancelarije skućeniije ili su ljudi mnogo predostrožniji.

Godine 1986. Šveđani su definisali granične vrednosti za „elektromagnetne emisije katodnih cevi“. Švedska Organizacija radnika i službenika

(TCO) izdala je tabele sa graničnim vrednostima za zračenja monitora (1987: MPR I, 1990: MPR II, 1991: TCO91 – ove tabele prilažemo).

Ali čak i u Švedskoj, koja je najviše napredovala, sve je ostalo samo na preporukama i ne postoje striktni propisi koje proizvođači moraju poštovati. Ipak, raduje činjenica da sve više proizvođača prihvata preporuke i pravi monitore koji emituju manje štetnog zračenja.

Zračenja možemo podeliti u dve kategorije: u jednoj je rendgensko zračenje (X-zraci). Ova opasnost je u ekranima (računara i televizora) dobrom zaštitom svedena na minimum. Nasuprot tome, vidljiva svetlost kao i električno i magnetno polje nisu zanemarljivi. U njima udela imaju optičke komponente (vidljive i ultraljubičaste) visokofrekventnog polja, niskofrekvencijska magnetna i električna polja, kao i elektrostatičko polje.

MONITORI 14 INČA : SAŽETI PODACI

	ADI Microscan 3E+	Eizo T240i-M	Escm ES 2824	Escm ES 2624
OPŠTI PODACI				
Cena proizvođača	1251 DEM	2268 DEM	798 DEM	998 DEM
Maloprodajna cena	1100 DEM	1700 DEM	****	****
EKRAN				
Veličina (dijagonala)	14 inča	14 inča	14 inča	14 inča
Vidljiva veličina (dijagonala)	13.8 inča	13.4 inča	13.4 inča	13.9 inča
Upotrebljiva veličina (800x600) (dijagonala/površina slike)	11.9 inča/431 cm ²	12.0 inča/433 cm ²	11.8 inča/419 cm ²	12.1 inča/434 cm ²
TEHNIČKI PODACI				
Tip monitora	multi	multi	3-frekvencijski	multi
Maksimalna rezolucija	1024x768	1024x768	1024x768	1024x768
Frekvencija ponavljanja slike	50 – 100 Hz	55 – 90 Hz	47 – 90 Hz	50 – 90 Hz
Ekranska frekvencija	30 – 48 kHz	30 – 60 kHz	31.5/35.5/48.19kHz	30 – 57 kHz
Opseg video signala	75 MHz	80 MHz	64 MHz	nema podataka
Raster tačaka	0.28 mm	0.25 mm	0.28 mm	0.26 mm
Norme zračenja	MPR II	MPR II	MPR I	nijedna
PODEŠAVANJE				
Osvetljenost/kontrast	da	da	da	da
Pozicija slike (vertikalna/horizontalna)	da	da	da (horizontalna)	da
Veličina slike (vertikalna/horizontalna)	da	da	da (vertikalna)	da
Konvergencija	ne	da	ne	ne
Korekcija boja	ne	ne	ne	ne
Demagnetizacija	ne	ne	ne	ne
DIMENZIJE/TEŽINA				
Dimenzije (V×Š×D)	36.2x37x38.6 cm	41x34.5x36.7 cm	34.5x37x38 cm	35x35x40.5 cm
Težina	12.5 kg	14 kg	14 kg	15 kg
GARANCIJA				
Trajanje (meseci)	12	12	6	6

Nokia 44BN-151



Karakteristike	1	2	3	4	5
Kvalitet slike					
Veličina slike					
Ergonomičnost					
Podешavanje					
Cena/performance : proizvođač					
Cena/performance : trgovina					

Prosečan monitor firme Nokia, bez velikih prednosti ili slabosti, sa cenom koja je takođe prosečna.

Sigma Ergoview 14



Karakteristike	1	2	3	4	5
Kvalitet slike					
Veličina slike					
Ergonomičnost					
Podешavanje					
Cena/performance : proizvođač					
Cena/performance : trgovina					

Ergoview 14 je stvoren za pobednika, a i cena mu je pristojna.

Ugroženost od ultraljubičastog zračenja može se isključiti, kao i visokofrekventno zračenje, koje je prema dosadašnjim saznanjima zanemarljivo. Na elektrostatička polja ljudi su navikli od rođenja.

Sasvim je drugačija situacija sa ostalim vrstama zračenja. U Švedskoj je ispitivana moguća veza između jakih ekranskih magnetnih polja i promena na koži. Ova ispitivanja su potvrdila rezultate drugih naučnih studija koje su na svetlost dana iznele povezanost kožnih obolenja i promenljivih polja monitora, ali još nije tačno utvrđen mehanizam njihovog čestog delovanja na ljude.

U kancelarijama se pored monitora najčešće nalaze i drugi izvori zračenja, počevši od kablova i vodova za napajanje, preko neonskih sijalica, do laserskih štampača, pri čemu svi daju svoj doprinos elektrostatičkom „kancelarijskom spektru”. Međutim, ekran monitora ostaju u centru pažnje, jer su oni izvori zračenja kojima je korisnik najbliži. Jačina polja kod kolar-monitora je veća nego kod monohromatskih, pošto se kod prvih koriste tri elektronska mlaza za prikaz na ekranu, a kod drugih samo jedan.

Iako su moguće razmere bioloških oštećenja još neistražene, svi veliki proizvođači monitora po-

kušavaju da kod svojih proizvoda značajno smanje rizik od zračenja. „Zdraviji bio-model” monitora skuplji je za otprilike 10 do 20% nego konvencionalni, što proizvođači pravdaju većim troškovima za materijal i složeniju izradu.

Jačina elektromagnetnog polja zaista se može znatno umanjiti primenom tehničkih sredstava. Međutim, često će direktno na radnom mestu (dakle ispred monitora) zračenje biti smanjeno, ali će zato na drugim mestima (sa zadnje strane monitora) biti povećano.

Naravno, i proizvođači dodataka se trude da popune praznine na tržištu. Tako se nude filteri koji se stavljaju ispred ekrana, ili cela metalna kućišta u koja se stavlja monitor, ali je pitanje koliko su ova sredstva svrsishodna i koliko utiču na optimalno prilagođene komponente monitora.

I još nešto: bez obzira na sve rizike od zračenja, pri izboru monitora ne treba izgubiti iz vida ni druge njegove aspekte. Oštrina slike, veran prikaz boja, prikaz bez treperenja i optimalni korisnički programi su barem isto toliko važni kada je reč o prijatnoj radnoj sredini.

Ergonomičnost: *Diamond Pro* deluje znatno profesionalnije od svog „mlađeg brata”. Maksimalna ekranska frekvencija od 58 kHz dovoljna je da pri rezoluciji od 800x600 tačaka frekvencija ponavljanja slike bude oko 92 Hz. I pri rezoluciji od 1024x768 ova frekvencija je još uvek iznad 70 Hz. Skoro da se već podrazumeva da je švedska MPR II norma o zračenju zadovoljena. Sve zajedno – ocena: „dobar”.

Podешavanje: Sa prednje strane monitora *Diamond Pro 14* nalaze se samo taster za uključivanje/isključenje i regulatori za podешavanje osvetljenosti i kontrasta. Na prvi pogled, sa ostale tri strane kućišta nema nikakvih dodatnih regulatora. Da nešto nije...? Ipak nije, jer su sa leve strane kućišta, iza tajnog poklopca, spretno sakriveni još neki regulatori. Pomoću dva (plus/minus) tastera podешavaju se parametri koji se prethodno izaberu pomoću jednog kliznog regulatora. Mogućnosti podешavanja su zaista mnogostruke. Pored veličine slike i njene pozicije, mogu se korigovati i razna izobličenja prikaza na ivicama ekrana.

Priručnik je opširan, a ocena „zadovoljava”.

Cena/performance: Preporučena cena monitora *Diamond Pro 14* ČMDNMC je 1806 DEM. Dobar kvalitet slike, visoke frekvencije ponavljanja slike i smanjeno zračenje donose mu ovde ocenu „zadovoljava”. Pošto je ovaj monitor sasvim nov na tržištu, cena u prodaji se ne razlikuje mnogo od preporučene.

**Nešto sasvim posebno:
NEC Multisync 4FG**

Kvalitet slike: Ovaj NEC-ov „malša” odlikuje se oštrim prikazom i bogatim kontrastima. Male greške u konvergenciji pojavljuju se samo u levom gornjem delu ekrana. Isto tako, i stabilnost slike i prikaz boja stavljaju monitor 4FG ispred ostalih. Jedino kod geometrije slike ovaj model pokazuje male slabosti – visina slike je desno nešto manja nego levo. Eventualna magnetizacija ekrana može se ukloniti pomoću odgovarajućeg tastera. Sve u svemu, ocena „vrlo dobar”.

Veličina slike: Katodnu cev ovog NEC-ovog monitora uokviruje širok ram. Zato, iako je monitor opremljen 15-inčnom katodnom cev, vidljivo je samo 13.9 inča. Po tome se 4FG nalazi u istoj klasi sa ADI- monitorom. Međutim, ono što ovaj monitor stvarno izdvaja od ostalih jeste optimalno iskorišćenje vidljivog dela ekrana. Za 13.6 inča korisne dijagonale (570 cm²), ocena „vrlo dobar”.

Ergonomičnost: Praktično isti podaci kao za monitor *Diamond Pro* važe i za 4FG. Sa ekranskom frekvencijom od 57 kHz on sliku pri rezoluciji 800x600 prikazuje 90 puta u sekundi. Sa 70 Hz, pri rezoluciji od 1024x768, takođe su zadovoljeni svi zahtevi. Sve to, uz poštovanje MPR II norme, zaslužuje ocenu „dobar”.

Podешavanje: Kao i kod većeg monitora 5FG, i ovde se vrlo brzo pronalaze optimalni parametri pomoću unakrst postavljenih tastera. Na raspolaganju su i mogućnosti podешavanja izobličenja prikaza na ivicama i korekcije boja. Kao i monitor firme Sigma, i 4FG poseduje dugme za demagnetizaciju (koje smo već spomenuli). Priručnik na 21 strani objašnjava sve što treba da znate o instalaciji, podешavanju i uklanjanju tipičnih grešaka. Ocena: „dobar”.

Cena/performance: Ovaj monitor je u svim kategorijama zaslužio najbolje ocene. Zajedno sa Eizo-monitorom, nalazi se ispred svih ostalih testiranih 14-inčnih monitora. Naravno, kvalitet ima i svoju cenu, ali je

Highscreen AS 57K	Mitsubishi Diamond Scan 14	Mitsubishi Diamond Pro 14	NEC Multisync 4FG	Nokia 51	Sigma Ergoview 14
698 ****	999 DEM 999 DEM	1806 DEM 1806 DEM	2622 DEM 2000 DEM	2040 DEM 1700 DEM	1615 DEM ****
14 inča 13.4 inča 11.4 inča/408 cm ²	14 inča 13.1 inča 11.5 inča/412 cm ²	14 inča 13.2 inča 12.1 inča/437 cm ²	15 inča 13.9 inča 13.6 inča/570 cm ²	14 inča 13.2 inča 13 inča/517 cm ²	14 inča 13.5 inča 12.5 inča/481 cm ²
multi 1024x768 52 – 100 Hz 30 – 60 kHz 77 MHz 0.28 mm nijedna	multi 1024x768 40 – 90 Hz 30 – 38.5 kHz 40 MHz 0.28 mm MPR II	multi 1024x768 50 – 90 Hz 30 – 58 kHz 70 MHz 0.28 mm MPR II	multi 1024x768 55 – 90 Hz 27 – 57 kHz 75 MHz 0.28 mm MPR II	multi 1024x768 50 – 100 Hz 30 – 50 kHz 63 MHz 0.28 mm TCO91	multi 1280x1024 50 – 100 Hz 30 – 64 kHz 75 MHz 0.25 mm MPR II
da da	da da	da da	da da	da da	da da
da (vertikalna)	da	da	da	da	da
ne ne ne	ne ne ne	ne da ne	ne da da	ne ne ne	ne ne da
35.8x36.3x37.2 cm 12.5 kg	35.3x35.4x39 cm 13.5 kg	35.2x34.9x38.6 cm 13.5 kg	39.6x37.2x41.4 cm 17.5 kg	33.5x36x36.5 cm 12.5 kg	36.1x36x40.7 cm 14.5 kg
12	12	12	12	12	48

PUNIH 10 GODINA

MINICIR

Virtual Library of Faculty of Mathematics - University of Belgrade
elibrary.matf.bg.ac.rs



PC386sx-33

Procesor AMD386sx 33 MHz
RAM 1 MB, SIMM
Hard disk 42 MB, IDE
Flopi 5.25", 1.2 MB
TVGA9000, 512 KB VRAM
VGA mono monitor 14"
Desktop kućište, 200W
Tastatura ASCII, 101

1399 DEM

PC386DX-40

Procesor AMD386DX 40 MHz
CACHE memorija 64 KB
RAM 2 MB, SIMM
Hard disk 42 MB, IDE
Flopi 5.25", 1.2 MB
TVGA9000, 512 KB VRAM
VGA mono monitor 14"
Desktop kućište, 200W
Tastatura ASCII 101

1668 DEM

BEZ PRAVE KO

WORKSTATION

Procesor 80c286 20 MHz
RAM 1 MB
Flopi 5.25", 1.2 MB
Herkules, TTL monitor 14"

759 DEM

NOVI SAD - Beočinski put b.b
(021) 611-366

U RAČUNARSTVU

MICROSOFT

PC486DX-33

Proc. INTEL 486DX 33 MHz
CACHE memorija 64 KB
RAM 4 MB, SIMM
Hard disk 42 MB, IDE
Flopi 5.25", 1.2 MB
TVGA9000, 512 KB VRAM
VGA mono monitor 14"
Desktop kućište, 200W
Tastatura ASCII 101

2657 DEM

PC486DX-50

Proc. INTEL 486DX 50 MHz
CACHE 8 KB + 256 KB
RAM 4 MB, SIMM
Hard disk 42 MB, IDE
Flopi 5.25", 1.2 MB
TVGA9000, 512 KB VRAM
VGA mono monitor 14"
Desktop kućište, 200W
Tastatura ASCII 101

3365 DEM

486DX2-66

Proc. INTEL 486DX2 66 MHz
CACHE 8 + 256 KB
RAM 4 MB, SIMM
Hard disk 42 MB, IDE
Flopi 5.25", 1.2 MB
TVGA9000, 512 KB VRAM
VGA mono monitor 14"
Desktop kućište, 200W
Tastatura ASCII 101

3744 DEM

INKURENCIJE !

DOPLATE

SVGA kolor monitor 14"
1024x768.....292 DEM
1 MB RAM.....78 DEM
FDD 3.5".....135 DEM
Disk 89 MB.....139 DEM
Disk 105 MB.....207 DEM
Disk 125 MB.....277 DEM
Disk 205 MB.....605 DEM

ŠTAMPACI

HP IIIp - 1 MB, toner
HP IV - 2 MB, toner

EPSON LX-400, YU-set
EPSON FX-1050, YU-set
EPSON LQ-570, YU-set
EPSON LQ-1070, YU-set
EPSON DFX-5000

MREŽE

Novell NetWare 2.20
Novell NetWare 3.11
Novell NetWare Lite
SCO TCP / IP
Ethernet card NE2000 kom.
ArcNet card 8/16 bit
ArcNet Active Hub 4/8
Pribor, Instalacija, Podrška

BEOGRAD - Molerova 70
(011) 432-690, 430-059

SERVER PO KRVI

Svedoci smo sve bržeg porasta značaja i primene računarskih mreža, a prvi korak obično predstavlja neka aktivnost u vezi sa lokalnim računarskim mrežama (LAN). Cilj ovog članka je da predstavi pregled funkcionalnih karakteristika PC kompatibilnih servera i neka aktualna tehnička rešenja. Predstavićemo COMPAQ SYSTEMPRO/LT kao jedno konkretno rešenje i prikazati deo njegovih mogućnosti u ulozi servera mreže, poredeći ga sa „klon” serverom.

Any to any using many

Diskusija o performansama i konkretnim ciframa (msec, KBPS i slično) je prilično nezahvalan i jalov posao, jer je za korisnika ipak najvažnije da sistem efikasno (pouzdan, itd.) izvršava konkretnu klasu poslova. Sa druge strane, postavlja se pitanje izbora test-programa i radnog okruženja, tako da je prava vrednost svakog rezultata testiranja uvek prilično neizvesna. Svih ovih opasnosti smo bili svesni i ovom prilikom, pa smo se odlučili za sledeći pristup. Na dve konkretne mreže – jedna sa COMPAQ-ovim serverom a druga sa klonom (ovaj drugi je, dakle, server samo po funkciji), oba servera sa identičnim nazivnim performansama ključnih elemenata (vrsta i takt procesora, veličina i brzina memorije, kvalitet i kapacitet podsistema diskova) – isprobaćemo nekoliko tipičnih aplikacija iz baza podataka i uporediti rezultate, koristeći *Novell NetWare 3.11*. Ne pretendujemo da iz svega izvlačimo dalekosežne zaključke, ali smatramo da će barem jedna grupa korisnika imati koristi od ovakvih iskustava i da će sagledati razlike među mrežama sa „serverom po krvi” i „serverom funkciji”.

OPŠTA ANATOMIJA

Prvi korak u definisanju i uspostavljanju tržišta servera učinila je firma COMPAQ COMPUTER CORP. predstavljanjem svoje mašine SYSTEMPRO, krajem 1989. godine. Na taj način je i formalno predstavljena nova klasa mašina, tzv. namenskih servera (*dedicated file server*), koji zadovoljavaju potrebe velikih računarskih mreža, kombinujući snažne komponente, obezbeđujući proširivost i pouzdanost (mi, nažalost, jednako prevodimo *dependability* kao i *reliability* – ovde podrazumevam pouzdanost u smislu prvog pojma).

U proteklom periodu veći broj proizvođača računara je predstavio svoje servere. Obično je reč o najjačim mašinama iz proizvodnog programa odgovarajućeg proizvođača. U redim slučajevima reč je o mašinama koje su posebno projektovane za ove namene (COMPAQ, IBM, Tricord, Everex, Dell, ...), a u još redim radi se o tzv. „superserverima”, koji mogu imati do 10 procesora (na primer, NF450FT firme „Netframe Systems”).

Server je nova vrsta računara projektovana da opslužuje korisnike („klijente”) u mrežnom okruženju. Dakle, server je tipičan višekorisnički resurs, koji stoji na raspolaganju korisnicima u mreži. Postoji nekoliko elemenata značajnih za ispravno i efikasno funkcionisanje mreže, ali većina svih zahteva koje generišu korisnici, kao i većina podataka koji kruže kroz mrežu, pre ili kasnije završava u serveru, odnosno na njegovom disku. Najveći teret mreže je na plećima servera i stoga on mora biti kombinovan od najboljih elemenata PC mašina, miniračunara, inženjerskih radnih stanica i velikih računara.

Spiridon Matić



Snimio: Studio Tešić & Nenad Petrović

Jedna moguća klasifikacija okruženja servera bazirana je na broju korisnika: radna grupa, odeljenje, cela kompanija. Nisu usvojene konkretne cifre za razgraničenje ovih klasa, ali

je sadašnja nomenklatura COMPAQ-a sledeća: do 50, do 200, preko 200 korisnika.

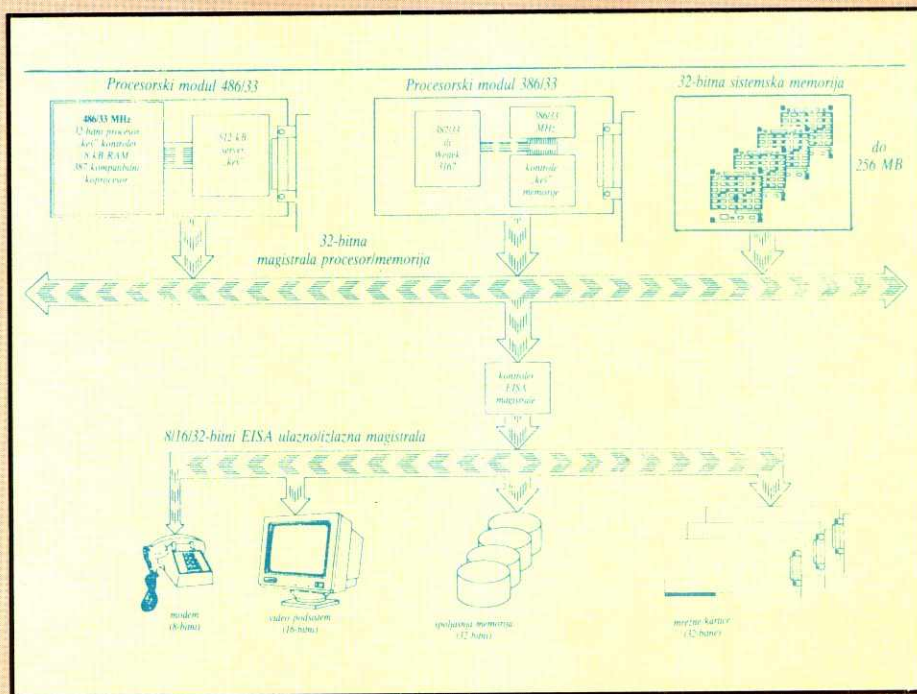
U pogledu klasifikacije poslova, takođe postoji veliko šarenilo. Predstavljamo jednu moguću nomenklaturu:

- **File/Print usluge:** centralizovano čuvanje zajedničkih podataka; štampanje; pristupa veći broj korisnika koji na radnim stanicama izvršavaju prevashodno jednorisničke aplikacije (obrada teksta, tabelarna izračunavanje i slično).
- **Komunikacije:** pristup drugim mrežama, koristeći zajednički *bridge* ili *gateway*, kao i korišćenje elektronske pošte unutar lokalne mreže.
- **Aplikacioni programi:** automatizacija specifičnih funkcija radne grupe koristeći DBM (*Data Base Management*) sistem.
- **Kombinovane usluge:** kombinacija dve ili više gore pomenutih usluga na jednom serveru.

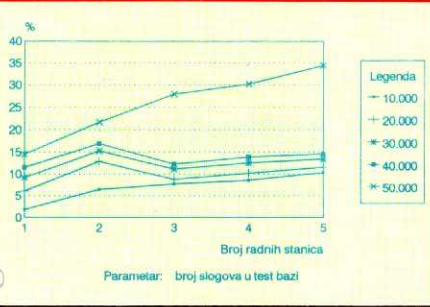
OSNOVNE KARAKTERISTIKE

Postoje različita mišljenja o značaju i prioritetu pojedinih karakteristika servera, sa stanovišta korisnika. Jedna moguća i često citirana klasifikacija bi bila sledeća:

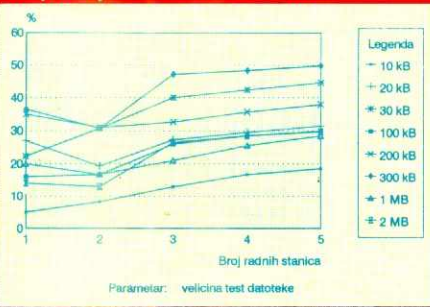
- kompatibilnost,
- pouzdanost,
- mogućnost nadgradnje (*upgradeability*),
- mogućnost proširivanja (*expandability*),
- performanse,
- dobavlјivost.



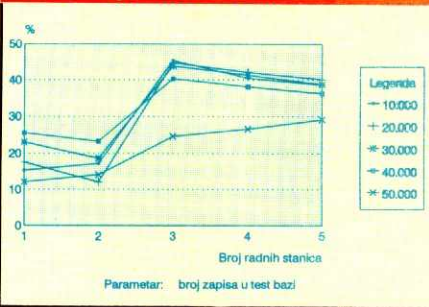
Sortiranje baze podataka



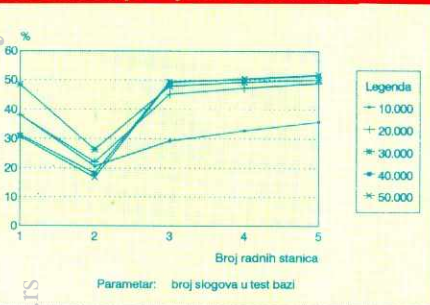
Kopiranje sa diska servera



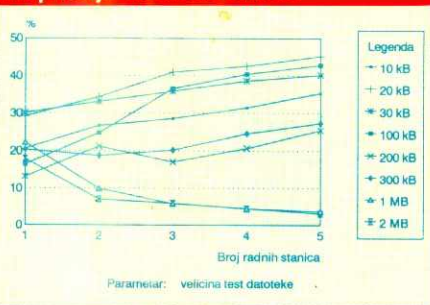
Sekvencijalni read-write



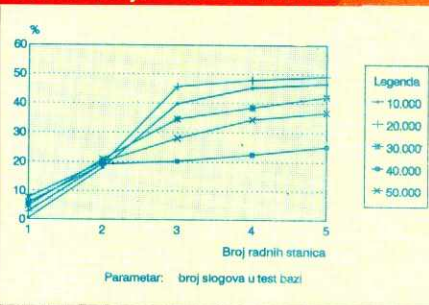
Indeksiranje – jednostruki indeks



Kopiranje na disk servera



Indeksiranje – dvostruki indeks



Kompatibilnost je značajna karakteristika sa aspekta realizacije mreža, a omogućava korišćenje proizvoda (hardverskih i softverskih) različitih proizvođača (*multivendor solutions*), ili pak proizvoda istog proizvođača, ali iz različitih vremenskih perioda. Iskustvo nas uči da po pitanju deklarisanosti kompatibilnosti proizvođača treba biti veoma obazriv, jer je složenost današnjih proizvoda (kako hardverskih tako i softverskih) takva da se neki latentni uzroci mogućih problema otkrivaju posle dužeg ili kraćeg perioda eksploatacije („sve je to kompatibilno, samo je nešto malo manje a nešto malo više kompatibilno“).

Bez obzira što je server samo jedan od elemenata mreže, ima vrlo mnogo argumenata u prilog tezi da njegova **pouzdanost** postaje jedna od najznačajnijih karakteristika. Pобоljšanju pouzdanosti prilazi se na dva načina.

Prvi je poznat pod nazivom *fault avoidance* i predstavlja skup metoda, postupaka i tehnologija kojima se preventivno deluje na pojavu eventualnih grešaka. Najkraće rečeno, reč je o projektovanju i praviljenju kvalitetnog proizvoda.

Drugi način poznat je pod nazivom *fault tolerance* i predstavlja skup metoda, postupaka i tehnologija kojima se obezbeđuje da pojava greške ili otkaza u serveru ne dovede do zastoja u radu celog sistema. Pri tome se, naravno, dozvoljava da se performanse sistema degradiraju do određenog stepena. Od poznatih tehnika koje obezbeđuju otpornost na otkaze, u današnjim serverima se tipično primenjuje zaštita operativne memorije (RAM), što već i nije specifičnost samo servera, kao i zaštita podsistema diskova, primenom različitih nivoa RAID tehnike. Ovde bi se još moglo svrstati i primene upravljivih izvora neprekidnog napajanja (UPS).

Mogućnost nadgradnje predstavlja svojstvo koje obezbeđuje očuvanje prvobitne investicije korisnika i povećanje performansi servera, zamenu nekih ključnih elemenata novijim, kao što je npr. zamena CPU (Central Processing Unit) ploče novijom, bržom. COMPAQ je

uveo ovakvu praksu za svoje mašine bazirane na EISA magistrali, pri čemu korisnik plaća samo razliku u ceni.

Mogućnost proširivanja predstavlja svojstvo koje obezbeđuje očuvanje prvobitne investicije korisnika i povećanje kapaciteta pojedinih resursa servera (RAM, diskovi, povećanje broja procesora u višeprocorskom serveru i slično).

Posmatrano sa tehničkog stanovišta, kriva **performansi** mreže, bez obzira kako ih definišali i merili, bez obzira na tip servera i klasu poslova, pokazuje tendenciju zasićenja, što je uostalom karakteristika svih tehničkih sistema. Na početnom delu krive, „kapacitet“ servera je nepotpuno iskorišćen – dakle, mogu se dodavati novi klijenti, bez uticaja na ukupne performanse. Nakon zasićenja servera, ukupan protok podataka kroz mrežu ostaje približno konstantan, dok svaki klijent ponasob uočava njegov pad (linearan ili još i brži), sa daljim povećanjem ukupnog broja klijenata.

Nivo zasićenja mreže zavisi od više različitih faktora. Ovde ćemo navesti one koji se tiču samog servera: podsistem CPU/memorija, podsistem diskova, podsistem mrežnog interfejsa.

Pod pojmom **dobavljivost** podrazumevamo više različitih parametara, koje svaki potencijalni i ozbiljni korisnik mora uzeti u obzir pri opredeljivanju za konkretni proizvod: da li postoji ovlašćeni zastupnik ili distributer, tj. mogućnost nabavke od „lokalnog“ prodavca; uslovi i procena realnih mogućnosti realizacije ponuđenih uslova garancije, servisiranja i podrške u garantnom i vangarantnom roku; i, konačno, nabavna cena. Naravno, svaki korisnik će, na bazi ovih parametara i eksploatacionih parametara celog budućeg sistema (koliko košta zastoj sistema, itd.) definisati svoju politiku.

TEHNOLOGIJA SERVERA

Današnji serveri tipično koriste jedan ili više procesora (najčešće dva). U ovom drugom slučaju (npr. ALR, COMPAQ, EVEREX, ...),

obično je primenjen princip simetrične arhitekture. Dakle, oba procesora su ravnopravna i izvršavaju poslove koje im odgovarajući dispečer (*scheduler*) dodeljuje. Za razliku od ovoga, postoji i asimetrični slučaj, gde su procesori specijalizovani po funkcijama koje obavljaju. Upotrebljena je brza magistrala za pristup operativnoj memoriji, uz intenzivnu primenu *cache* tehnike. Tipični minimalni kapacitet operativne memorije je 8-16 MB, dok je maksimalno mogući u najvećem broju slučajeva 64 MB (COMPAQ SYSTEMPRO do 256 MB). Za veličinu spoljašnje „keš“ memorije uglavnom se usvajaju vrednosti između 128 kB (najčešće) i 512 kB, dok u izboru arhitekture i algoritma „keširanja“ postoji veliko šarenilo: *two-way set associative, write through, direct-mapped, write back, direct-mapped, write through* – da pomenemo samo neke od češće primenjenih, ali o detaljima, možda, nekom drugom prilikom. Teorijski i simulacioni rezultati u pogledu prednosti i nedostataka pojedinih algoritama su dobro poznati, ali bi bilo zanimljivo ispitati njihov uticaj na performanse realnog sistema za pojedine tipične klase aplikacija.

U pogledu izbora I/O (input/output) magistrale, opredeljenje većine proizvođača je za EISA arhitekturu, dok je znatno manje za MCA. Naravno, još uvek ima i rešenja sa ISA magistralom. Broj slobodnih utičnica na magistrali kreće se u od 6-10, u zavisnosti od proizvođača.

Problem podsistema diskova svodi se na pitanje izbora sprežnih kontrolera, organizacije podsistema i kapaciteta. Naravno, zahtevi korisnika su: što veći kapacitet, što veći protok i što veća pouzdanost. U primeni je nekoliko različitih tipova sprežnih kontrolera: IDE, ESDI, SCSI, SCSI-2, IDA, IDA-2. IDA i IDA-2 su isključivo 32-bitni EISA kontroleri, dok se ostali tipovi mogu naći u 16- i 32-bitnoj verziji. Najčešće se primenjuju inteligentni kontroleri, koji obezbeđuju nekoliko različitih efekata.

Tehnikom „keširanja“ i primenom nizova diskova omogućuje se veći protok. Kapacitet „keš“ memorije disk kontrolera tipično se kreće u granicama 1 – 4 MB, dok npr. *SCSI Driver*

Array firme „FCS Ltd.“ može imati do 64 MB. Nizovi diskova predstavljaju novi pristup u organizaciji podsistema diskova, koji, pored povećanog protoka i većeg kapaciteta, omogućava realizaciju pouzdanih podsistema diskova, primenom RAID tehnike (vidi COMPAQ Bilten u ovom broju). Izbor konkretnog tipa organizacije je stvar kompromisa između traženog raspoloživog kapaciteta (dakle, stvarni, fizički umanjnjen za deo koji se koristi za povećanje pouzdanosti), protoka (koji, opet, zavisi i od klase aplikacija), pouzdanosti i, naravno, cene.

Tipični kapaciteti diskova kreću se u opsegu 1-4 GB.

COMPAQ-ov PRISTUP

Unapređujući opšte usvojenu osnovnu koncepciju konkurentne magistrale, naročito primenom posebnih metoda za „keširanje“, COMPAQ je zaokružio svoju arhitekturu, poznatu pod nazivom Flex. Kasnije je razvijena i proširena verzija, Flex/MP, koja je predviđena za višeprocorsorski rad. Principijelna šema Flex/MP arhitekture prikazana je na slici 1.

EISA specifikacija predstavlja proširenje ranije usvojene ISA (Industrial Standard Architecture) magistrale. Predviđena je za snažnije sisteme, što se vidi iz osnovnih karakteristika:

- podržava 32-bitne *bas mastere* (do 15 mastera),
- podržava 32-bitni DMA rad na magistrali.

Sa druge strane, obezbeđena je potpuna kompatibilnost sa ISA magistralom, što omogućava korišćenje svih postojećih dodatnih modula i kontrolera.

Kada su jedinice diskova u pitanju, COMPAQ podržava većinu sada postojećih standarda (IDE, ESDI, napredni ESDI (sa EISA sprejom), SCSI-2, IDA i IDA-2). Ovom prilikom okrećemo pažnju na IDA i IDA-2 kontrolere, koji podržavaju nizove diskova, obezbeđujući, pored povećane pouzdanosti i bolje performanse.

COMPAQ SYSTEMPRO/LT je prikazan sa više aspekata u dosadašnjim izdanjima COMPAQ Biltenu. Jedno značajno svojstvo familije COMPAQ SYSTEMPRO prikazano je u ovom Biltenu. Zbog svega ovoga, u ovom tekstu smo se ograničili na prikazivanje nekih karakterističnih tehničkih rešenja i na testiranje performansi SYSTEMPRO/LT servera u *Novell NetWare* okruženju.

Testirana standardna konfiguracija spada u kategoriju najslabijih COMPAQ-ovih servera: procesor 386/33, RAM 8MB, 210 MB IDE disk, 16 kB *four way set-associative* keš memorije.

oči u oči

Klon mašina za poređenje je imala sledeću konfiguraciju: procesor 386/33, RAM 8MB, 210 MB IDE disk, 64 kB keš memorije.

Kao radne stanice upotrebljene su 286 mašine, sa procesorom na 16MHz, RAM 2 MB i sa 44 MB IDE diskovima.

Očigledno je da bi, ako bismo ovako upoređivali oba servera, dobijeni rezultati, barem što se tiče performansi, bili približno jednaki, i tu ne bi trebalo ništa posebno ni ispitivati – zbog toga što, u tom slučaju, ni jedan od resursa COMPAQ mašine, koji je kvalitetno razlikuje od klona, nije iskorišćen. Obe mašine su pre podizanja mreže testirane programom BENCH, ver. 6.0 (PC MAGAZINE). Dobijeni su vrlo slični rezultati, sem u pogledu video podsistema (COMPAQ je bio drastično brži, ali ovo

nije od značaja za njegovu funkciju mrežnog servera).

Jedna od značajnih razlika je to što COMPAQ ima 32-bitnu EISA magistralu, tako da je omogućeno korišćenje brzih perifernih kontrolera. Da bismo donekle uspostavili realnije odnose, COMPAQ je testiran sa 16-bitnom mrežnom karticom, dok je klon server imao 8-bitnu karticu istog proizvođača. Tako je svaki server bio za korak iza najboljih mogućih uslova, u pogledu mrežnog interfejsa. Smatramo da rezultati testiranja ovako organizovanog okruženja imaju određeni praktični značaj, jer prikazujemo samo relativne odnose rezultata merenja performansi oba servera.

Sve radne stanice su bile opremljene 8-bitnim mrežnim karticama. Instaliran je mrežni softver *Novell NetWare ver. 3.11*.

Formirana je mreža sa 5 radnih stanica. Na svakoj radnoj stanici se izvršavao identičan program, pri čemu je broj radnih stanica varirao od 1 do 5. *FoxProLAN* je konfigurisan tako da su svi radni katalozi smešteni na disk servera. Dakle, maksimalno je opterećen disk servera i sama mreža.

Testovi merenja brzine rada sa bazom podataka rađeni su na bazama veličine 10.000, 20.000, 30.000, 40.000 i 50.000 zapisa. Test programi su pisani u jeziku *FoxProLAN* verzija 1.02. Korišćeni su sledeći testovi:

- READ: sekvencijalno čitanje jednog polja iz zapisa, modifikacija i ponovni upis;
- SORT: sortiranje baze;
- INDEX1: indeksiranje baze po jednom jedinstvom indeksu;
- INDEX2: indeksiranje baze po jednom dvostrukom indeksu.

Za merenje brzine rada sa datotekama korišćene su datoteke veličine 10 kB, 20 kB, 30 kB, 100 kB, 200 kB, 300 kB, 1MB i 2MB

- COPY1: kopiranje datoteke sa diska na disk servera. Izvodi se 10 puta kopiranje i brisanje datoteke;
- COPY2: kopiranje datoteke sa diska servera na disk radne stanice. Izvodi se 10 puta kopiranje i brisanje datoteke;

REZULTATI TESTIRANJA...

... predstavljenih konfiguracija mogu se sumirati u dve grupe utisaka. Prvu grupu je teže kvantifikovati, a sastavljena je od ocena i iskustava dobijenih, pre svega, u periodu intenzivnog rada sa konfiguracijom. Najkraće rečeno, COMPAQ je pouzdana mašina: sva instaliranja softvera izvode se pravolinijski, bez ikakvih neugodnih iznenađenja, a ponašanje mašine u eksploataciji je stabilno.

Druga grupa rezultata predstavljena je grafički. Na slikama su prikazani rezultati merenja performansi primenom test-programa READ, SORT, INDEX1, INDEX2, COPY1 i COPY2, respektivno. Prikazano je relativno povećanje vremena izvršavanja pojedinih programa na klon serveru u odnosu na COMPAQ server.

COMPAQ SYSTEMPRO/LT se, naravno, može koristiti i bez IDA podsistema diskova i 32-bitne mrežne kartice, ali u tom slučaju se neće osetiti njegova puna snaga i korisniku će biti uskraćena mnoga zadovoljstva. Zadovoljstvo rada sa kvalitetnom, pouzdanom i stabilnom mašinom je uvek vrlo prisutno, čak i kad vozite sjajan automobil koristeći samo prvu i drugu brzinu. Ali, za puni doživljaj treba još nešto više. O tome u jednom od naših sledećih priloga.

Nastavak sa strane 19

preporučena cena od 2622 DEM ipak previsoka, čak i za monitor koji se zasniva na 15-inčnoj katodnoj cevi. Tu je ocena samo „dovoljan“. Srednja prodajna cena kreće se oko 2000 DEM, što već zaslužuje ocenu „dobar“.

Pomalo preskup: Nokia 44BN-151

Kvalitet slike: Zaista oštar prikaz na ekranu ovog monitora kviri greška u konvergenciji u donjem delu ekrana, koja se manifestuje crvenkastim senkama. Zbog toga su linije i ivice u toj zoni blago nejasne. Geometrija slike je prihvatljiva (oba leva kruga su malo jajolika). Prikaz boja i stabilnost slike su bez greške. Nedostaje mogućnost ručne demagnetizacije ekrana. Ocena: „zadovoljiva“.

Veličina slike: Monitor firme Nokia se u ovoj kategoriji našao na drugom mestu, iza NEC-ovog monitora. Od 13.2 inča ukupne vidljive dijagonale iskorišćeno je 13 inča, što daje korisnu površinu od 517 cm² i za dlaku ocenu „vrlo dobar“.

Ergonomičnost: Dok pri rezoluciji od 800x600 tačaka ovaj monitor sa ekranskom frekvencijom od 50 kHz ima frekvenciju ponavljanja slike od dobrih 80 Hz, dotle ona na rezoluciji od 1024x768 pada na nekih 60 Hz. Ali zato ovaj monitor, jedini od svih testiranih, zadovoljava uslove koje postavlja nova, strožija norma o zračenju – TCO91. Ovo poslednje je uticalo da monitor u ovoj kategoriji dobije ocenu „dobar“.

Podešavanje: Svi elementi pomoću kojih se vrše podešavanja, zajedno sa prekidačem za uključnje/isključnje monitora, ugrađeni su u desnu stranu kućišta. Pored osvetljenosti i kontrasta, sa četiri pokretljiva točkica se regulišu i veličina slike i njena pozicija. Priručnik na sedam strana objašnjava samo ono najosnovnije. Sve to zaslužuje samo ocenu „dovoljan“.

Cena/performance: Preporučena cena od 2040 DEM je za ponuđeni kvalitet slike i nepraktično podešavanje previsoka – uprkos poštovanju stroge TCO91 norme. Ocena: „dovoljan“. Ocenu „zadovoljiva“ dobija srednja prodajna cena ovog monitora, koja iznosi oko 1700 DEM.

Nov i kvalitetan: Sigma Ergoview 14

Kvalitet slike: Sa svojom Trinitron katodnom cevi, sa rasterom od samo 0.25 mm, *Ergoview* je imao najbolje predispozicije da u ovom delu testa zauzme neko od prvih mesta i on je to očekivano i ispunio. Prikaz na ekranu je veoma oštar, tako da su, čak i pri višim rezolucijama, i mala slova lako čitljiva. Na gornjoj ivici slike primećuje se mala greška u konvergenciji. Odstupanja kod geometrije slike i stabilnosti takođe su neznatna. Raduje činjenica da je na raspolaganju i dugme za demagnetizaciju, tako da oceni „vrlo dobar“ u ovoj kategoriji ništa nije stajalo na putu.

Veličina slike: Ovde monitor nije dobio najvišu ocenu, ali nije ni loše prošao. Za korisnu površinu ekrana od 481 cm² (12.5 inča od 13.5 inča vidljive dijagonale) *Ergoview 14* je dobio ocenu „dobar“.

Ergonomičnost: Odličnih 64 kHz ekranske frekvencije omogućavaju frekvenciju ponavljanja slike i do 100 Hz pri rezoluciji od 800x600 tačaka, što je tek poneka grafička kartica u stanju da prati. I pri rezoluciji od 1024x768 tačaka ova frekvencija još uvek iznosi celih 80 Hz. Takođe, *Sigma Ergoview* poštuje MPR II normu. Ocena: „dobar“.

Podešavanje: *Sigma Ergoview 14* nudi ručnu demagnetizaciju i regulisanje izobličenja na rubovima ekrana. Oba ova regulatora smeštena su sa zadnje strane monitora. Sa prednje strane pronaći ćete preostale regulatore za podešavanje osvetljenosti, kontrasta, veličine i pozicije slike. Veliki obrtni regulatori omogućavaju brzo i tačno podešavanje.

Prateće uputstvo na 16 strana je iscrpno napisano. Sve u svemu – ocena „zadovoljiva“.

Cena/performance: Za 1615 DEM kolika mu je preporučena cena, *Ergoview 14* nudi najbolji kvalitet slike, visoke frekvencije ponavljanja slike i zaštitu od zračenja. To je za ocenu „dobar“. Ovaj model je sasvim nov, ali se može očekivati da prodajna cena bude oko 10-20% ispod preporučene. Tada će *Ergoview* po odnosu cena/performance daleko odmaći od konkurencije.

Sa nemačkog preveo:
Kragl Aleksandar

A new
Star

star

You can finally reach the

Virtual Library of Faculty of Mathematics - University of Belgrade

elibrary.matf.bg.ac.rs



Star ComputerPrinter LC-20

star
MICRONICS
the ComputerPrinter



AUTHORIZED DEALER "SAGA" D.O.O. BEOGRAD, YUGOSLAVIA

Milentija Popovića 9. "SAVA CENTAR" 11070 NOVI BEOGRAD

Tel: & Fax: 011/222-3579; 147-182; 222-4323 ext. 256 & 259

POVUCITE PRAVI POTEZ

Potencijalnom kupcu DOS-PC računara može se zavrtneti u glavi od šarolike ponude mikroprocesora. Četiri kompanije – Intel, AMD, Cyrix i IBM – nude najmanje 14 varijanti procesora iz klase 386 i 486, sa brzinama od 16 do 66 MHz. Samo Intel nudi preko 100 varijanti mikroprocesora 80x86, dok za 1993. planira 25 verzija 486-tice. Ne zaboravimo ni uskoro očekivani Intelov čip P5, kao ni nove klonove čipa 80x86, koje za prvu polovinu godine obećavaju ostali proizvođači. Trka za najbrži mikroprocesor unela je na tržište konfuziju, ali i donosi niže cene PC sistema.

Izbor se više ne može zasnivati ni na ceni ni na performansama. Potrošnja struje, prisustvo matematickog koprocesora i stepen integriteta u velikoj meri određuju kako će se CPU nositi s vašim dragocennim aplikacijama. Zar je onda čudno što je kupac PC računara zbunjen?

Najzbunjeniji su individualni korisnici i oni u malim firmama. „Da li je 386/40 brži od 486SX/20? Takve odluke su tvrd orah čak i za profesionalne prodavce računara. Poseban predmet zabune je razlika između SX- i DX- klase čipova. Jedan iskusni prodavac kaže da ni ne pokušava da sve to razjasni svojim mušterijama – samo se trudi da im ponudi sistem koji će najbolje odgovoriti njihovim potrebama.

Procvat ponude mikroprocesora dobrim delom je rezultat utrkivanja sve većeg broja proizvođača za mesto na tržištu, ali je ta raznolikost i odgovor na istinske potrebe korisnika. Na primer, novi Intelov procesor 486SL koji troši malo struje je prevashodno namenjen notebook računarima, gde su ranije dugotrajnost baterije i visoke performanse bili pojmovi koji su se uzajamno isključivali. Cyrix-ov procesor DRu2, koji uskoro treba da se pojavi, će se prodavati kao direktna dogradnja (nožica na nožicu) iz klase 486SX za korisnike Intelovih 386DX mašina.

Da bi se napravio pravi izbor sistema, neophodno je poznavanje skrivenih jakih i slabih strana kako postojećih tako i tek najavljenih procesora. Potrebno je znati i odnos cene i performansi, kao i koji CPU nudi najbolje mogućnosti dogradnje.

Srećom, pitanje kompatibilnosti se ne postavlja. Magazin BYTE je intervjuisao brojne izvore i sproveo niz testova sa čipovima Intela, AMD-a i Cyrix-a. Niko nije pomenio problem kompatibilnosti. Predsednik kompanije „Acer America“ Ronald Čvang (Ronald Chwang) je izjavio da je AMD 100% kompatibilan sa Intelom. Po rečima Džona Petersona (John Patterson) iz Tandy-ja, „kompatibilnost je apsolut“. Tandy ima potpuno poverenje u Cyrix-ov 486SLC i koristi ga za svoj 3800 HD notebook PC. Najozbiljniji problem uočen u našim testiranjima je nesposobnost nekih starijih programa da prepoznaju novije mikroprocesore.

POKRETAČKE SILE

Konkurencija je ubrzala razvoj mikroprocesora. Dok je Intel držao celokupno tržište procesora 80x86, mogao je sebi dozvoliti duže razvojne cikluse. Sada AMD i Cyrix proizvode procesore koji konkuriraju postojećim Intelovim jedinicama ili nude bolje performanse, i to često po nižoj ceni.

I IBM proizvodi unapredene verzije procesora 386SX i 486SX, koristeći licencu koju je otkupio od Intela. Po tom ugovoru, IBM ima pravo da ih koristi za svoje sisteme ili da prodaje matične ploče sa tim čipovima.

Takva konkurencija je stvorila kupovno tržište za proizvođače PC-ja i doprinela sniženju cena sistema. Jedan američki proizvođač matičnih ploča kaže da je AMD kao konkurent naterao Intel da iz temelja izmeni cenovnik 386 familije čipova. Slika 1 prikazuje uticaj konkurencije na Intelov cenovnik. Cena 16-MHz 386DX procesora je pala za oko 20% od kraja 1988 do drugog kvartala 1991. godine, kada je AMD izbacio svoj 386DXL. Od tada do danas, Intel je smanjio cenu 386DX za 43 posto, za trostruko kraće vreme.

Projektovanje mikroprocesora koji će biti poslednja reč tehnologije nije ni malo lak posao. Cyrix je potrošio 10 miliona dolara da bi izneo 486SLC i 486DLC na tržište, što je tričava suma u odnosu na 250 miliona koje je uložio Intel u razvoj svog procesora

486. Međutim, veličina mikroprocesorskog tržišta deluje primamljivo. „Dataquest“, firma koja se bavi istraživanjem tržišta, procenjuje da je celokupno CPU tržište (uključujući i čipove koji ne spadaju u klasu 80x86) u periodu od 1990-1991 vredelo gotovo 4 milijarde dolara, od čega je 70% pripadalo Intelu i AMD-u. I novi proizvodi odnose veliki deo kolača, kako u malim tako i u velikim tržišnim segmentima, u što se AMD uverio.

AMD je izbacio prvi klon Intelovog procesora 386 nakon duge sudske bitke sa Intelom. Presuda je odlučila da AMD ima pravo da prodaje svoj CPU na osnovu ugovora koji je potpisao u vreme kada je procesor 286 bio na vrhuncu. Od tada je AMD veoma uspešan na tržištu 386-tice – kompanija je planirala da do kraja 1992. proda preko 2 miliona mikroprocesora iz klase 386, što čini gotovo 50% udela na tržištu.

AMD je predstavio sistem sa 50-MHz 486DX čipom, koji koristi zaštićeni Intelov mikrokod. Predsed-

nik kompanije Džeri Senders (Jerry Sanders) izjavljuje da će početkom 1993. izbaciti ovaj CPU na tržište, bilo tako što će dobiti pravo na korišćenje mikrokoda sudskim putem ili tako što će napisati novi mikrokod.

Diversifikacija PC sistema stvorila je priliku za nove projekte mikroprocesora. Nekadašnja filozofija „jedan čip za sve namene“, zamerjena je procesorima projektovanim prema specifičnim potrebama. 386 i 486 procesori iz klase SL koji troše malo struje koriste se gotovo isključivo za prenosive (portable) sisteme koji rade na bateriji. Jeftini procesori iz klase SX postali su uobičajeni za ulazne desktop sisteme. Klasa 486DX sa visokim performansama opslužuje high-end radne stanice i servere. A u poslednje vreme, visoko integrirani CPU iz klase 8086 i 286 sve su popularniji izbor za novu kategoriju „ručnih“ (hand-held) računara (vidi tekst „Rezervni igrači osvajaju nove segmente tržišta“).

BYTE je razgovarao sa 20 vodećih svetskih proi-

LABORATORIJSKI TEST: DA LI JE VAŽNA MARKA?

Iako je količina i raznolikost procesora koji se nude pomalo zbunjujuća, pravo pitanje koje se postavlja jeste da li kupac uopšte treba da zna koji čip – Intel, AMD ili Cyrix – čini srce njegovog novog računara. Magazin BYTE i NSTL (Nacionalna laboratorija za testiranje softvera) su ispitali dve oblasti moguće diferencijacije: brzinu i kompatibilnost. Obavljeni su i razgovori sa brojnim ljudima koji znaju kako rade popularni mikroprocesori.

Testiranje smo obavili na mašini Tandon PC386, koja ima nekoliko pomoćnih kartica koje koriste četiri osnovna procesora: 33 MHz Intel 486DX, AMD 25 MHz 386SXL i 40 MHz 386DXL, i 25 MHz Cyrix 486SLC. Naši rezultati pokazuju da glavni procesor nije jedini faktor. 25 MHz Cyrix je dao vidno bolje performanse od Am386SXL iste brzine (vidi sliku A), što ne treba da čudi, s obzirom da Cyrix ima procesorski keš na samom čipu. Međutim, keš na procesoru nema mnogo uticaja na operacije koje nisu vezane za CPU, kao što su operacije sa bazama podataka.

U drugom testu smo proverili tvrdnju Cyrix-a da njihov procesor obavlja intidžer množenja četiri puta brže od konkurentskih. Sproveli smo test sa serijom 16-bitnih množenja u petlji. Pokazalo se da je tvrdnja Cyrix-a tačna: protok je poboljšao za oko 400% (vidi sliku B).

Naš test je pokazao i neodrživi tvrdnji o povećanju brzine zasnovanih na testovima sa malim kodom. Pri pisanju koda za testiranje, dobili smo veoma različite rezultate – varirali su i do 50% – zavisno od položaja testiranog koda u našoj .exe datoteci. To je zavisilo isključivo od toga da li se test-kod izvršava u potpunosti unutar procesorskog keša ili ne. Keš se pokazao veoma bitnim i kod porođenja 40 MHz čipa AM386DXL sa 33 MHz čipom 486DX, pošto AM386DXL ima 32 KB eksternog keša na dodatnoj kartici.

U cilju ostvarenja kompatibilnosti, kompanija AMD koristi Intelov mikrokod. Cyrix tvrdi da je sproveo potpunu implementaciju 80x86 procesora, koji je testiran na signalnom novou kako bi se garantovala puna kompatibilnost sa Intelovim zahtevima.

Mnoge kompanije kao što je proizvođač BIOS-a „Award“ kažu da nikada nisu imali potrebu da promene ni jednu liniju koda zbog nekompatibilnosti sa non-Intel čipovima. Sve eventualne teškoće u identifikaciji „Cyrix-ovog“ čipa se rešavaju pomoću posebnog identifikacionog registra koji obezbeđuje sam Cyrix. Džefri Flink (Jeffrey Flink), inženjer

Award-a kaže da raznolikost čipova i potreba za pomoćnim čipovima koji procesor povezuju s ostatkom kompjutera predstavljaju izazov za projektante.

Isto tako, ni nekoliko ICE (in-circuit emulator) proizvođača sa kojima smo razgovarali nisu nailazili ni na kakve posebne probleme u pogledu kompatibilnosti sa AMD i Cyrix procesorima – kompatibilnost je potpuna (nožica na nožicu). Ipak, još uvek niko ne podržava specifične funkcije (npr. upravljanje strujom i interni keš) koje nude Cyrix čipovi.

Ovlastili smo NSTL da obavi testiranje kompatibilnosti četiri Tandon procesorska modula. Umesto aplikacionog softvera, testiranje smo zasnovali na uslužnim programima, kod kojih je veća verovatnoća da će zagaziti u rede korišćene putanje procesora.

Prvi problem se javio kod softvera koji pokušava da otkrije tip procesora na kojem se izvršava. Kod „Cyrix-ovog“ čipa, tri programa su „prepoznala“ različite procesore. „Nortonov“ *Systema* je smatrao da se radi o 28 MHz Intelovom 486DX; *PCToolc SI* je javio da se radi o 25.5 MHz Intelovom 486SX; *Control Room* je smatrao da je to 50 MHz 386. Ovo ukazuje na problem identifikacije novih procesora pomoću starog softvera i trebalo bi da posluži kao upozorenje proizvođačima softvera da izbegavaju kod koji zavisi od brzine procesora.

Još ozbiljniji problem se javio kod dela paketa *PCTools* verzija 7.1 („Central Point Software“), takozvanog *CPBackup*. Naime, pre no što napravi bekap, *CPBackup* obavlja proveru DMA (Dynamic Memory Acces) kontrolera. Ovaj test dovodi do zamrzavanja računara kod minimalno konfigurisanog sistema (tj. kada nisu učitan drayveri uređaja). Po rečima inženjera kompanije Central Point, ovaj problem prouzrokuje vremenska petlja u softverskom kodu. (U novoj verziji paketa problem je rešen tako što je uklonjena vremenska petlja koja zavisi od procesora.) Mada ovo, strogo uzevši, nije problem kompatibilnosti, jer se ne radi o tome da li se ova ili ona naredba pravilno izvršava, on ipak ukazuje na teškoće koje se mogu javiti pri promeni konfiguracije procesora. Pokretanjem bilo kog upravljača memorije, kao što je QEMM, ovaj se problem u potpunosti otklanja.

Poslednji uočeni problem ukazuje na to da pojednaku pažnju treba posvetiti kompletnoj interakciji sistema i svakog pojedinog procesora. Problem se pojavio kada smo pokrenuli *SoftIce*, program fir-

zvođača PC računara. Samo 25% je izjavilo da sarađuje isključivo sa Intelom. Neki su najavili, mada još ne isporučuju, sisteme sa čipovima koji nisu proizvedeni u Intel-u. Drugi opet, kao na primer „Dell“, namećavaju da i dalje posluju isključivo sa Intelom. „Intelovi čipovi nude performanse koje nam trebaju i cene koje smo spremni da platimo,“ kaže jedan od direktora „Della“.

KOLIKO ZNAČI IME?

Proizvođači čipova se i ne trude da korisnicima omoguće direktna poređenja. U stvari, kao da njihovi stručnjaci za marketing namerno stvaraju zbrku. Na primer, Cyrix je svoje čipove nazvao 486SLC i 486DLC. Međutim, ovi čipovi nemaju matematički koprocesor, imaju drukčiji raspored nožica i mnogo manji keš od Intelove 486-tice. Ipak, podržavaju set naredbi 486-tice, dok intidžer množenje obavljaju četiri puta brže od mikroprocesora klase 386.

Proizvođači i prodavci PC-ja znaju u čemu je razlika, ali većina korisnika ne zna. Cyrix-ovi mikroprocesori su „386-tice sa značajno unapređenim performansama“, kaže predstavnik „Acer-a“ Čvang, i nastavlja da 486SLX ima performanse za oko 20-30% sporije od Intelovog čipa 486SX, što potvrđuju i testovi obavljani u BYTE-ovoj laboratoriji.

Bil Berkman (Bill Berkman), direktor marketinga u kompaniji „Mylex“ (proizvođač sistemskih ploča) tvrdi da Cyrix-ove cene nisu konkurentne cenama procesora 386 koji dolaze iz Intel-a ili AMD-a. Naime, 40-MHz Cx486DLC je dvostruko skuplji od 40-MHz procesora AM386DXL (mada je u Cyrix-ovu cenu uključen zaseban matematički koprocesor). Korisnik bi verovatno radije platio još 100\$ i dobio istinsku 486 mašinu.

Ipak, mnoge kompanije – Tandy, Tandon, Com-

puAdd i U.K. Opus – koriste Cyrix procesore. „Za nas je Cyrix nova avantura,“ kaže Peterson iz Tandy-ja, imajući na umu prijem kod potencijalnih kupaca. „Tandy“ se odlučio za Cx486SLC jer je odgovarao njihovim zahtevima, kako svojim performansama tako i cenom.

Predstavnici Intel-a su svesni da se ljudi teško snalaze u moru raznoraznih mikroprocesora. Zato su uveli poseban sistem za merenje performansi, verujući da će tako rasvetliti situaciju: to je iComp (Intel Comparative Microprocessor Performance), test brzine koji jednim indeksom izražava ukupne performanse mikroprocesora. Osnovna CPU platforma, 25 MHz 486SX ima indeks 100. Test je namenjen brzom rangiranju, pre svega, Intelovih procesora, ali se može primeniti i na sve ostale kompatibilne procesore.

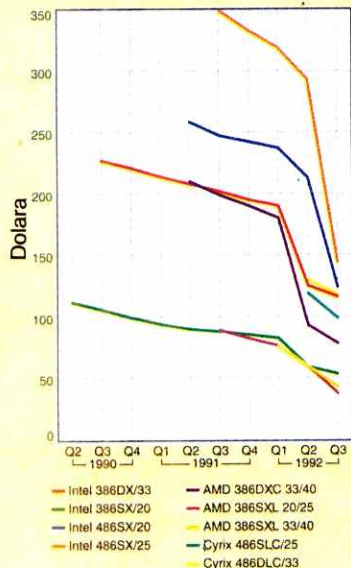
Konvencija po kojoj se imenuju procesori dodatni je izvor zabune. Proizvod čije ime sadrži 486 (ili samo 4), ili nema jasne indikacije da se radi o Cyrix ovom čipu, odvešće kupca na krivi put. Na primer, CompuAdd ima čitav niz Intel i Cyrix mašina, uključujući i 425SLC, 425SX, 433i i 433DLC. Prosečan kupac verovatno neće različitosti da dve od ovih mašina imaju Cyrix-ove procesore.

Intel će i ubuduće imati problema s nazivima, jer Američki zavod za patente ne prihvata kao zaštitni znak ime koje se sastoji od tri broja (npr. 586). Dakle, ništa ne stoji na putu ostalim proizvođačima da svoje čipove takođe nazovu 586, u nameri da stvore konfuziju koja njima ide na ruku.

RAZMIŠLJANJA O DOGRADNJI

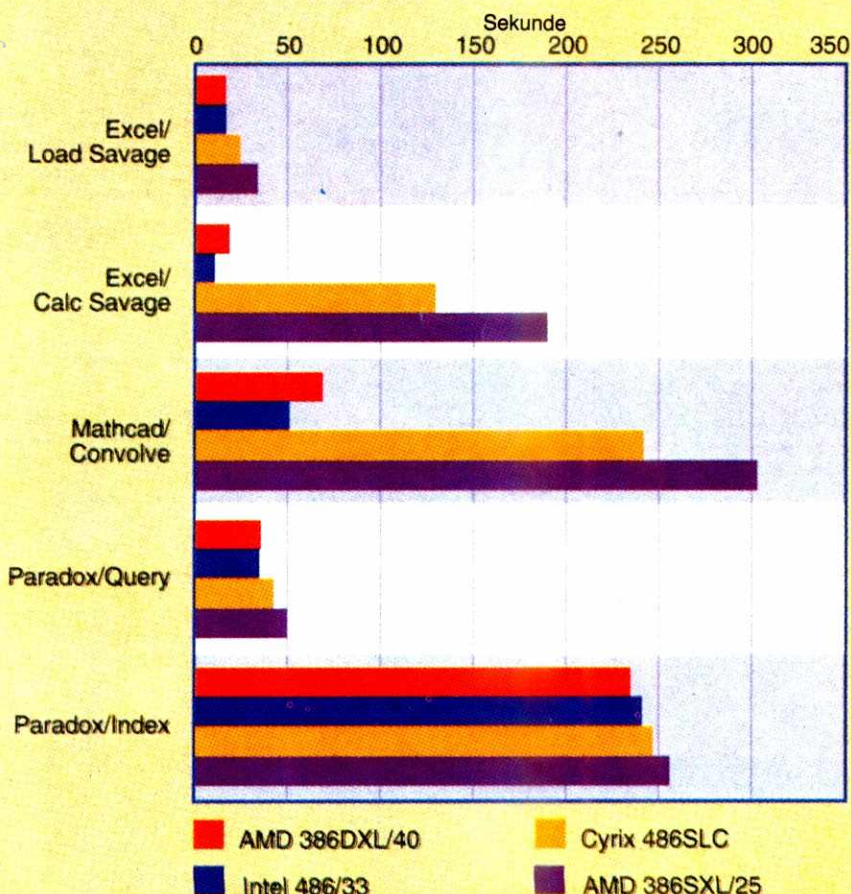
Pri izboru mikroprocesora, treba voditi računa o mogućnosti dogradnje, performansama i ceni. Pronalaženje prave kombinacije ovih elemenata usko je ve-

POREDENJE CENA CPU



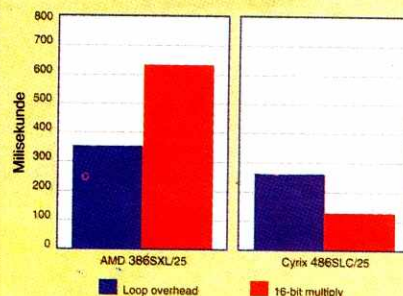
Slika 1: Ovaj grafikon prikazuje kako su proizvođači kompatibilnih čipova, kao što je AMD, uticali na CPU tržište. Otkako je AMD stupio na scenu, u martu 1991, Intel je snizio cenu 33 MHz 386DX čipa sa \$208 na \$107. Sve Cyrix-ove cene uključuju i poseban matematički koprocesor.

CPU PERFORMANSE



Slika A: Kao što smo i očekivali, Intel 486/33 je brži pri većini operacija. Brži radni takt AMD procesora daje mu prednost u drugim oblastima.

POREDENJE BRZINE 16-BITNOG MNOŽENJA



Slika B: Provera Cyrix-ove unapredene jedinice za 16-bitno intidžer množenje. Mada se operacije sa „sirovim“ brojevima obavljaju i više nego četverostruko brže, ubrzanje se jednim delom postiže zahvaljujući izvršavanju unutar keša jednog čipa. Kada to uzmemo u obzir, ubrzanje je nešto manje od četverostrukog koje navodi proizvođač.

me „Nu Mega“ (dibager na mašinskom nivou – low-level) na AM386DXL/40 procesorskom modulu. Prolaskom kroz testirani uzorak koji poziva BIOS funkciju „čekanja događaja“ (Event Wait – INT 15h, funkcija 83h), dolazi do prekoračenja internog steka i pada sistema. Kada je isti program pokrenut, punom brzinom, iz samog *Sottice*-a, problem se nije javio. Ovom prilikom nismo uspeali da izolujemo uzrok. Inženjeri firmi „Nu Mega“ i „Tandon“ bezuspešno su pokušali da isti problem provociraju na drugim sistemima, koristeći Am386DXL/40.

Da zaključimo, smatramo da su testirani procesori kompatibilni sa Intelovim standardom. Brzina i performanse zavise od ukupnog sistemskog okruženja koliko i od samog procesora. Naravno, 40 MHz procesor je brži od 33 MHz, ali se razlika može ublažiti dodavanjem eksterne hardverske podrške. Na kraju, kupovina od proverenog proizvođača ipak ostaje najbolji način da se osigura potpuna kompatibilnost.

zано za oblik sistema kome je namenjen. Mogućnost dogradnje najvažnija je za desktop korisnike – mada se i na portabl sistemima već viđaju OverDrive utičnice, tako da se na Intel 486SX sistem može instalirati 486DX CPU.

Dogradnja omogućava da vaš stari sistem drži korak sa stalnim unapređenjem performansi mikroprocesora, a da pri tom ne morate menjati celo kućište. Za nekoliko stotina dolara 16 MHz 486SX PC računar možete pretvoriti u 50 MHz 486DX (vidi „Razlozi za dogradnju“). Međutim, još uvek je mali broj korisnika koji su se odlučili za dogradnju, jer se poslednji tržišni ciklus još nije završio. Drugim rečima, oni koji su kupili modele koji se mogu dograditi još nisu osetili veliku potrebu za unapređenim performansama.

Intel je napravio pionirski potez kada je uveo dogradnju u obliku jednog čipa, takozvani *OverDrive* procesor, koji udvostručuje unutrašnji radni takt procesora i povećava ukupne performanse za oko 50%. Pre pojave *OverDrive* čipa, dogradnja je ostvarivana zamenom matične ploče ili instalacijom dodatne kartice ili ploče. „Ljudi nisu hteli da daju 2000\$ više za *AST Premium* (koji se mogao dograditi preko dodatne kartice) kada su za 1500\$ mogli kupiti novu matičnu ploču. Međutim, sa čipom koji dvostruko ubrzava rad mikroprocesora, dogradnja je postala jedan od ključnih elemenata pri kupovini.

Intelovi konkurenti tek sada ulaze na tržište dogradnji. Cyrix je nedavno predstavio verziju čipa 486DLC, DRu², koji se koristi kao dogradnja za IBM i Compaq 386DX/20 sisteme (vidi sliku 2). U Cyrix-u kažu da će DRu² raditi i sa ostalim 386DX sistemima, mada iz mehaničkih razloga – dvostruko je viši od čipa Intel 486DX – nekim PC računarima neće odgovarati. Opcije čipa 486 DLC koje zahtevaju hardversku podršku, kao što je keš ili upravljanje strujom, neće raditi optimalno, ali će povećanje performansi ipak biti značajno. Najznačajniji element ubrzanja će biti Cyrix-ova tehnologija udvostručenog radnog takta, koja će omogućiti da čip podešen na 20 MHz ostvari unutrašnju frekvenciju od 40 MHz.

BYTE je testirao prototipsku 33 MHz verziju DRu² na računaru IBM PS/2 Model 80. Sve u svemu, pokazao je 28% bolje performanse u odnosu na 16 MHz Intel 386DX. Video ubrzanje je variralo od 5% kod skrolovanja do čak 59% kod grafike. Ukupno unapređenje video performansi iznosi 35%. Kao što je i očekivano, testiranje diska na mašinskom nivou (low-level) je pokazalo najmanje poboljšanje – samo 10%. Mada su se datoteke učitalave i za 22% brže, opšti test koji je zatevao direktna učitavanja i zapisivanja je pokazao poboljšanje od samo 1%, budući da to zavisi od brzine drajva a ne od procesora.

Za sada, Cyrix namerava da ovaj čip prodaje direktno velikim kupcima i ne planira maloprodaju. Procenjuju da će DRu² koštati između 200 i 300\$.

U kompjuterskoj industriji vlada izvesni skepticizam u pogledu broja korisnika koji će se odlučiti za dogradnju. Džim Čepmen (Jim Chapman), zamjenik direktora marketinga u Cyrix-u kaže da ako se samo 1% od 30 miliona 386DX mašina dogradi, Cyrix će imati dovoljno veliko tržište. I AMD očekuje da tržište za dogradnju bude unosno – kompanija je javno izjavila da će podržati *OverDrive* utičnicu koju već imaju mnoge 486SX mašine.

BRZINE NIKAD DOSTA

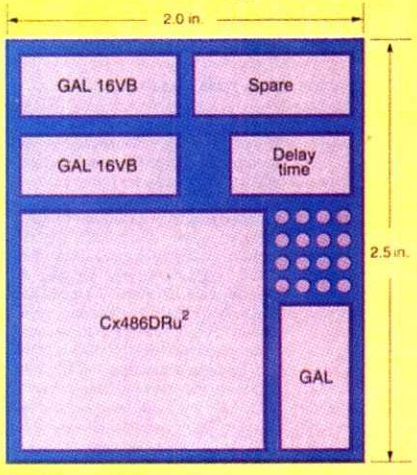
Razlike u performansama između 386 i 486 čipova takođe stvaraju zabunu. AMD se drži jedan korak ispred Intelovih linija 386SX i 386DX, tako što uspeva da izbací brže verzije svojih procesora 386SXL i 386DXL pre Intela.

Cyrix tvrdi da je ostavio ubrzanje performansi pomoću aritmetičke jedinice koja obavlja intidžer množenje. Nanijel Julit (Daniel Oulette), zamjenik predsednika kompanije „Microslate“, koja proizvodi pen-računare, kaže da je testirao brzinu procesora Cyrix 486SLC i da se pokazao za 1,8 do 1,9 puta bržim od procesora Intel 386SX pri istoj frekvenciji. Kod testiranja sa aplikacijom, razlika je bila još uočljivija. Pen-sistemi firme „Microslate“ popularni su među korisnicima geografskih informacionih sistema. Julit kaže da je osvežavanje ekrana bilo 4-5 puta brže.

Do izlaska ovog broja, Cyrix je verovatno već uveliko predstavio novu verziju popularnog čipa Cx486, Cx486S2/50. Za razliku od 486SLC i 486DLC, 50 MHz Cx486S2/50 sa udvostručenim radnim taktom odgovara 486 utičnici. Pošto nema matematičkog koprocera, naziva se ekvivalentom čipa 486SX.

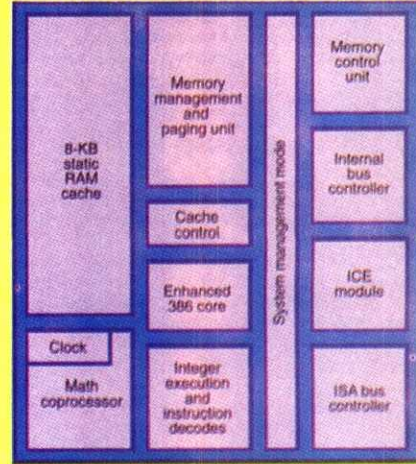
Cyrix tvrdi da će Cx486S2/50 imati unapređene

CYRIX DRu²



Slika 2: Cyrix-ov čip DRu² je direktna zamena (nožica na nožicu) za Intel 386DX/20. BYTE-ovi testovi su pokazali da se performanse mogu unaprediti za čak 28%. Glavni mikroprocesor je najveći kvadrat u donjem levom uglu. Ostale jedinice na pločici dimenzija 2x2.5 inča su pomoćni čipovi.

MIKROPROCESOR INTEL486SL



Slika 3: 486SL je Intelov 3.3-V mikroprocesor namenjen *notebook* sistemima. Osim brzine veće od procesora 386SL, nudi duže trajanje baterije. Pošto radi i na 5 V, bez teškoća se integriše u postojeće projekte 386SL mašina.

NEOBIČAN SLUČAJ FIRME NexGen

„NexGen Microsystems“ (San Hoze, Kalifornija) je po mnogo čemu zagonetna kompanija. Osnovana 1986, NexGen je počeo sa 386-kompatibilnim mikroprocesorom. Šest godina kasnije, NexGen još uvek nema gotov čip. 1990. se činilo da je na tragu nečeg značajnog, kada je najavljen čip-set koji koristi CISC arhitekturu da bi postigao RISC performanse, emulirajući 386/486 instrukcije. Kompjuteri sa ovim čip-setom bi radili dvostruko brže od 486, SPARC ili Mips sistema i približili se mašini IBM RISC System/6000. Nedavno je Nex Gen najavio P5-kompatibilan čip koji će se pojaviti samo mesec dana nakon što Intel izbací svoj procesor nove generacije, početkom godine.

Ovoga puta, posmatrači misle da gde ima dima, možda ima i vatre. Analitičari pretpostavljaju da bi

NexGen mogao 486 jezgru, koje navodno već ima, dodati paralelne intidžer jedinice, proširiti keš, promeniti radni takt i početi s prodajom „P5-kompatibilnog“ procesora već sredinom 1993.

Majkl Selejtjer (Michael Slater) iz *Microprocessor Reporta* kaže da NexGen nema P5 čip, već samo brzu implementaciju seta instrukcija 486, koja se približava performansama čipa P5.

Isto mišljenje ima i Ken Loui (Ken Lowe), analitičar firme za istraživanje tržišta, „Dataquest“: „NexGen pokušava ono što je Cyrix uradio sa svojim 486-kompatibilnim procesorom. Upoređuju se sa P5 tehnologijom, jer žele da postignu P5 performanse. Dobar marketinški potez.“ Bilo kako bilo, ovo bi mogla biti poslednja šansa za NexGen da se dokaže.

NOVI ČIPOVI NA HORIZINTU

Tokom sledećih šest meseci možemo očekivati nove procesore glavnih takmaca na tržištu Intel-kompatibilnih mikroprocesora. Pošto proizvođači mikroprocesora rade u dosluhu sa proizvođačima sistema, te su oni u toku novih zbivanja mnogo pre zvaničnog predstavljanja, verovatno će se uskoro pojaviti i mašine sa novim procesorima.

Intel 486SX/33

486SX/33 je brža verzija postojeće linije procesora. Predstavljen je u septembru.

IBM 486SLC

IBM verzija procesora Intel 486SL. Predstavljen u oktobru.

Cyrix 33 MHz 486SLC

Brža verzija procesora SLC, sa čitavim nizom novih opcija za upravljanje strujom. Zamišljen kao direktna konkurencija Intelovom čipu SL. Predstavljen u oktobru.

Intel 486SL

486SL troši malo struje, a ima performanse procesora SX. Mnogi proizvođači *notebook* računara već su se odlučili za njega. Pojavio se u novembru.

Cyrix DRu²

To je 486DLC koji udvostručuje performanse 20 MHz 386DX sistema. Zamišljen kao dogradnja. Pojavio se u novembru.

Cyrix 486S2/50

486SX kompatibilan procesor sa manjim kešom nego kod Intelovog pandana. Cyrix tvrdi da je smanje-

nje keša nadoknađeno time što je on sada *write-back* tipa. Pojavio se u novembru.

Intel P5

Čip koji će zameniti 486 na vrhu Intelove proizvodne linije. Očekuje se u prvom kvartalu 1993.

AMD – čipovi klase 486

AMD obećava 25-, 33-, i 50 MHz verzije svojih 486-kompatibilnih mikroprocesora, mada analitičari javljaju da će se verovatno pojaviti i čipovi sa udvostručenim unutrašnjim radnim taktom. Očekuju se do sredine 1993, zavisno od ishoda sudskog spora u vezi sa korištenjem Intelovog mikrokoda.

Chips & Technologies PC/Chip sa 386 jezgrom

C&T nije više na margini mikroprocesorskog tržišta. Novi čip će biti prvi rezultat ogromnih napora uloženi u proizvodnju 386-kompatibilnog procesora za „ručne“ i male *notebook* kompjutere. Očekuje se tokom 1993.

Intel P24T

Intel P24T je procesor koji služi kao dogradnja za 486DX2 sisteme. Utičnice za ovaj čip već se ugrađuju na neke matične ploče. Utičnica je slična onoj za *OverDrive*, ali ima jedan red nožica više. P24T će koristiti P5 tehnologiju za unapređenje performansi. Intel nije želeo da precizira šta podrazumeva pod P5 tehnologijom, ali je moguće da će P24T, kao P5, imati poboljšan FPU i paralelne intidžer papjove.

NexGen – klon čipa P5

Radi se o čipu tipa P5 koji NexGen namerava da izbací nakon što se pojavi Intelov P5. Poznajući NexGen, možda ćemo čekati i do 1994.

REZERVNI IGRAČI OSVAJAJU NOVE SEGMENTE TRŽIŠTA

Veliki broj proizvođača mikroprocesora je najavio planove za proizvodnju Intel-kompatibilnih procesora za usko specijalizovane aplikacije, a među njima su i „NEC Technologies”, „Chips & Technologies”, „International Meta Systems”, „Vadem” i „NexGen”.

Najstariji igrač na polju procesora kompatibilnih sa Intelom je bio NEC, sa svojom V-linijom procesora koja se pojavila 1984. Originalni mikroprocesori su projektovani u Intelu, a NEC je dobio licencu. Kasnije je ova linija zamenjena procesorima projektovanim u samom NEC-u. Danas NEC prodaje 30 verzija V procesora uključujući i nisko-voltnu VHL 33 MHz procesore. Oni se nalaze u raznim mašinama, od Olivettijeve *Quadrom SubNotebook* do Compaqovog *IDA* drajv kontrolera. Najnoviji je V55 procesor – kompatibilan sa procesorom 80186, na 12,5MHz. Originalni mikrokontroler korišten za 8086 i 80186 je ugrađen u hardver, tako da se većina instrukcija izvršava u jednom ciklusu.

„International Meta Systems” ima ambicioznije planove. IMS tvrdi da ima novi 100 MHz Risc mikroprocesor, koji će početi da se proizvodi sredinom ove godine. Biće u stanju da emulira Intel 486 ili Motorola 68040, istom brzinom ali po znatno nižoj ceni. IMS je novi čip namenio pen-računarima koji zahtevaju visoke performanse za funkcije kao što su obrada i tumačenje rukopisa, ali će biti pogodan i za „kameleonski kompjuter” koji će izvršavati pro-

grame pisane i za PC i za Macintosh.

U aprilu prethodne godine, mali proizvođač čipsetova „Vadem” je predstavio mali PC zasnovan na jednom jedinom čipu, takozvani „palmtop” (računar koji stane na dlan). Proizvođači kompjutera mogu koristiti *VG-230 Sub-Notebook Engine* i vrlo brzo izbaciti na tržište jeftine „palmtop” računare. „Vademov” čip je projektovan tako da inkorporira 16 MHz NEC V30HL mikroprocesor u uređaj sa jednim jedinim čipom, koji sadrži LCD kontroler, interni skener tastature i ugrađenu jedinicu za upravljanje strujom. PCMCIA 2.0 modul pruža perifernu podršku. Poslednja verzija podržava i digitalno pero, tako da odgovara pen-računarima. Sekundarni prostor displej memorije prima trag elektronskog pera, tako da sistem obrađuje poteze i prepoznaje rukopis, ne utičući pri tom na originalnu sliku na ekranu.

C&T se takođe okreće tržištu „ručnih” kompjutera. U avgustu je kompanija najavila da će se usredsrediti na integrisane procesore i logičke čipove koji će proizvođačima PC računara pružiti rešenje u obliku jednog čipa za „ručne” i *notebook* kompjutere. C&T već ima PC/Chip integrisani procesor koji daje PC osobine 8086 jezgrou. Sada će razviti 386 jezgro za PC/Chip i VGA kontroler koji će zameniti CGA displej drajver niske rezolucije. Gordon Kembel (Campbell), predsednik kompanije C&T, tvrdi da će se veoma brzo pojaviti prvi ozbiljni sistemi sa baterijom koja će omogućavati 50 sati rada.

performanse zahvaljujući *write-back* kešu koji daje veći protok (throughput) sa manje keša. Cx486S2/50 će koristiti 2KB keša, što je ekvivalentno Intelovom *write-through* kešu od 8KB. *Write-through* keš procesor koristi samo kod *read* operacija, dok *write* odlazi direktno u mnogo sporiju glavnu memoriju. Kod *write-back* tehnike procesor zapisuje promene u keš memoriju sve dok je ne napuni. Tada se ceo memorijski blok kopira u glavnu memoriju brzom *block-move* operacijom. Na ovaj način se postiže znatno ubrzanje. Cyrix tvrdi da će performanse prosečne aplikacije biti jednake onim koje daje 50 MHz procesor Intel 486DX2, dok će performanse grafike biti unapređene za oko 10%.

Cyrix-ov čip će moći da pristupa memoriji u jednom ciklusu, a ne u dva. Cena procesora Cx486S2/50 će biti oko 250\$ za proizvođače sistema.

Često prisutno u minikomputerskim i *mainframe* okruženjima, profiliranje instrukcija se javlja i kod mikroprocesora. Profiliranje instrukcija je analiza relativne frekvencije instrukcija koje koriste određene apli-

kacije. Sa ovim informacijama, inženjeri mogu podesiti arhitekturu tako da je za češće korišćene instrukcije potrebno manje ciklusa, čak i po cenu da se koristi više ciklusa za izvršavanje ređih instrukcija, čime se znatno unapređuju performanse.

Intel uveliko koristi profiliranje instrukcija pri projektovanju novih procesora i tako stvara ogromnu bazu podataka koja sadrži putanje instrukcija različitih komercijalnih aplikacija. Rezultat Intelovog profiliranja je i uočena razlika u upotrebi instrukcija kod aplikacija pisanih jezikom C++ i onih pisanih drugim jezicima, što opet ukazuje na blisku vezu između kompajlera i procesora. Ako C++ prevladava na polju komercijalnih alati za razvoj softvera, Intel i ostali će se možda odlučiti da svoju arhitekturu podese prema frekvenciji instrukcija iz ovog jezika.

PERFORMANSE SA BATERIJAMA

Performanse dobijaju sasvim drugu dimenziju u areni prenosivih računara, gde su štednja struje i visok stepen integrisanosti ključni faktori. Intel je planirao predstavljanje porodice 486SL koja troši malo struje. Prva dva mikroprocesora će raditi na 25 odnosno 33 MHz i oba imaju ugrađen matematički koprocessor. Intel kaže da je 486SL fizički manji i ima veći stepen integrisanosti nego 386SL, što će omogućiti proizvođačima da smanje dimenzije *notebook* matičnih ploča za oko 60%. 25 MHz čip već se proizvodi, dok se 33 MHz verzija planira za prvi kvartal 1993.

486SL (vidi sliku 3) predstavlja značajan napredak u odnosu na 386SL. Intel tvrdi da se performanse kreću od 11 MIPS-a za 25 MHz 486SL, do 1.4 MIPS-a za 33 MHz verziju (radi poredjenja, najbrži 386SL (25 MHz) ima brzinu od 5.2 MIPS).

Jednako značajno je i svojstvo „fleksibilne voltaže” koje poseduje 486SL. Iako je 486SL 3,3 voltni statični uređaj, može raditi i na 5 V. Zahvaljujući tome – kao i kompatibilnosti čipa 486SL sa SMM-om (modul za upravljanje sistemom) procesora 386SL i I/O čipom – već krajem godine možemo očekivati *laptop* računare sa ovim čipom. Dok će neki proizvođači jednostavno prilagoditi 486SL postojećim 5-voltnim matičnim pločama, naprednije mašine će raditi na 3,3V. Kada se 486SL prilagodi ostalim 3,3-voltnim delovima, potrošnja struje se smanjuje za 50% u odnosu na 5-voltni 386SL sistem. Intel kaže da bi ovo trebalo da produži trajanje baterije za 1-4 sata.

Kao i 386SL, novi procesor 486SL ima PI (Peripheral Interface) sabirnicu koja je po svojoj funkciji ekvivalentna lokalnoj sabirnici kod desktop kompjutera. PI sabirnica omogućava grafičkom kontroleru ili fleš-kartici da zaobiđu I/O sabirnicu i direktno se vežu za CPU. PI sabirnica radi punom brzinom mikroprocesora, dok I/O sabirnica ima relativno sporu propusnu moć od 8MHz.

FAKTOR CENE

Konkurencija je smanjila cene čipova, kao i cene sistema. 25 MHz 486DX je prvobitno koštao 950\$ i znatno je doprinio ukupnoj ceni sistema; sada košta 328\$. Po rečima Stiva Vorena (Steve Warren), zamjenika predsednika kompanije „Altima Systems” koja proizvodi *laptop* kompjutere, CPU sada mnogo manje utiče na cenu sistema. „Altima” prodaje AMD 386SX sistem i Intel 386SL sistem. AMD sistem je 10% jeftiniji od sistema sa Intelovim mikroprocesorom, ali zato ovaj drugi portabl daje bolje performanse zahvaljujući eksternom kešu.

Kad je u pitanju cena, Intel ima značajnu prednost u odnosu na svoje konkurente: kooperativnu reklamu. Naime, Intel plaća proizvođačima sistema da podrže kampanju pod nazivom „Unutra je Intel” kada reklamiraju svoj proizvod. Ovakav aranžman je posebno privlačan za kompanije koje vrše prodaju putem narudžbenice, jer njihov prihod potpuno zavisi od reklame. Dva britanska proizvođača, „Elonex” i „Viglen” kažu da Intel snosi 5% ukupnih troškova za reklame u novinama i časopisima – jedini uslov je da reklamni oglas sadrži i logo „Unutra je Intel”. Jedan američki proizvođač pominje sličan ugovor, ali dobija od Intela samo 3%. Za mnoge proizvođače ova pogodba uveliko nadoknađuje razliku u ceni između skupljih Intelovih i konkurentskih čipova.

Mnogi korisnici i proizvođači imaju više poverenja u Intel. Mark Vena, direktor proizvodnje kompanije „Epson America” kaže: „Ako bi dva procesora – AMD i Intel – imali istu cenu, ipak bismo uzeli Intelov zbog zajedničkih resursa i dosadašnje saradnje.”

Isto stanovište zastupa i proizvođač softvera „WordPerfect”. Peter Mogan (Peter Maughan) je zadužen za održavanje mreže i nabavku radnih stanica za potrebe svoje kompanije. Testira svaku mašinu koju „WordPerfect” kupi. On kaže: „Najmanje problema su nam stvarale mašine sa Intelovim procesorima.”

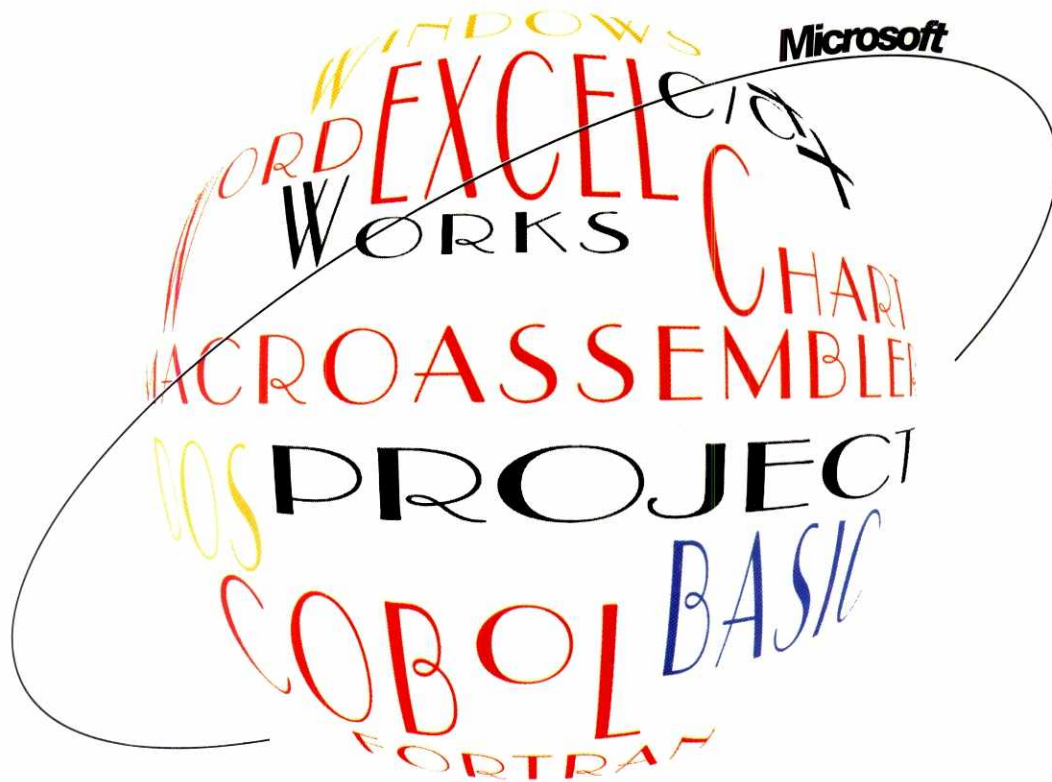
PRILAGOĐENE FUNKCIJE

Sistemi postaju sve raznolikiji (vidi sliku 4). Oblasiti koje su se nekada smatrale marginalnim delovima tržišta sada su u centru pažnje. Na primer, udeo koji na tržištu zauzimaju desktop računari je smanjen otako su *notebook* računari postali dovoljno moćna a time i sasvim prihvatljiva alternativa. Predviđanja koja daje „InfoCrop”, firma za istraživanje tržišta, pokazuju da će prodaja *notebook* računara porasti za 22,7% u toku sledeće dve godine, dok su u 1992. godini zauzeli samo 14,5% ukupnog PC tržišta.

Intel je proširio svoju ponudu mikroprocesora da bi konkurisao ostalim proizvođačima na tržištu specijalnih kompjutera. O tome svedoči 25 verzija procesora 486 koji treba da se pojave tokom 1993. godine. Veliki je broj procesora koje će proizvođači sistema podržati. Procesori će biti sve brži i trošiti sve manje struje. AMD i Cyrix će imati čipove potpuno kompatibilne sa 486-ticom, a čak i Intelovoj velikoj nadi, P5, prete NexGen i ostali konkurenti. Po svemu sudeći, nezbežno će doći do smanjenja Intelovog udela na tržištu, a time i do smanjenja profita.

CPU I RAZNOLIKOST SISTEMA			
„Ručni” PC	Notebook	Desktop LAN stanica	Server radna stanica high-end desktop
			
C&T PC/Chip NEC V series Vadem AHM AT&T Hobbit	Intel 486SL Intel 386SL Intel 386SX AMD 386SLXV IBM 386SXC	Intel 486SX Intel 386DX AMD 386DXL AMD 386SLC Cyrix 486SLC Cyrix 486DLC Cyrix DRU	Intel 486DX Intel 486DX2 Cyrix M6 AMD 486DXL Intel P5 IBM 486SLC2

Slika 4: CPU se sve više projektuje tako da odgovara specifičnim potrebama svake od kategorija sistema. Ručni kompjuteri zahtevaju visok stepen integrisanosti i mali utrošak struje. Portabl sistemi traže i performanse i štednju struje. Desktop PC traži visoke performanse po prihvatljivoj ceni. Konačno, radnim stanicama i serverima treba čista kompjuterska snaga.

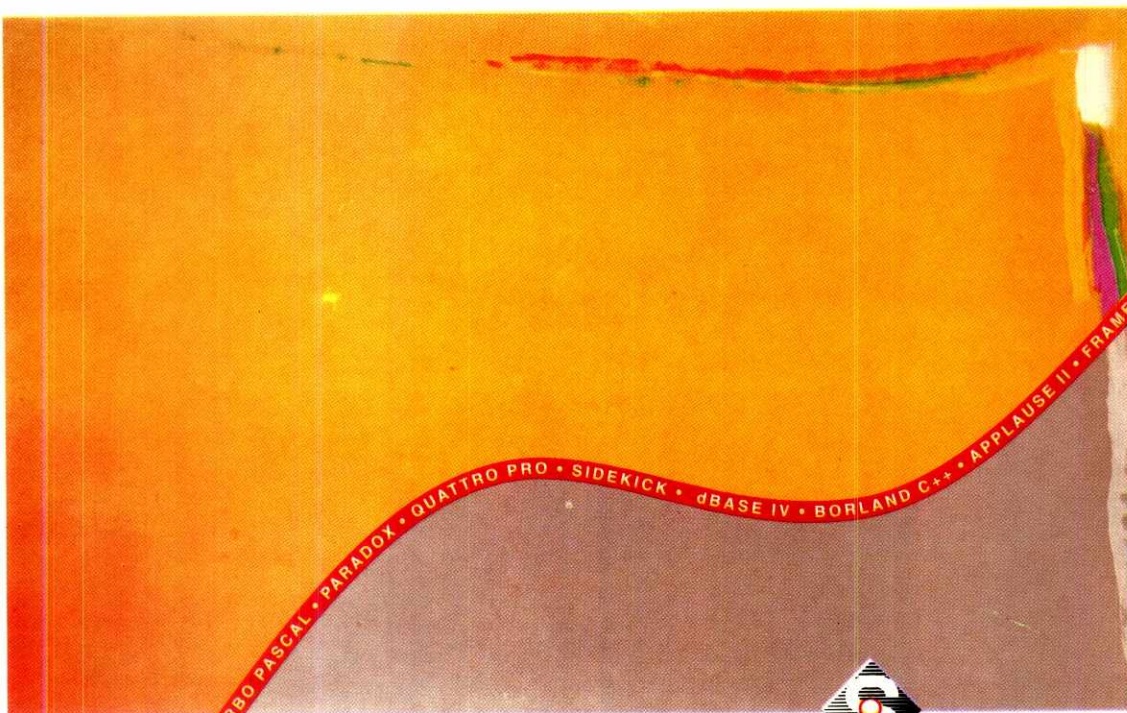


MISOFT OVLAŠĆENI DISTRIBUTER KORPORACIJE MICROSOFT

11000 BEOGRAD, SKADARSKA 45, TEL: 011/343-043

BORLAND
SOFTWARE

HIGH YELLOW, JULES OLITSKI



SOFT LAND

OVLAŠĆENI SUBDISTRIBUTER KORPORACIJE BORLAND

11000 BEOGRAD • SKADARSKA 45 • Tel: 343-043

SiGraf



SiliconGraphics
Distributer

Sistem Integrator

- ❖ *RISC tehnologija od grafičkih radnih stanica do superkompjutera paralelne arhitekture*
- ❖ *UNIX/IRIX, LAN/WAN, TCP/IP, Relacije baze podataka, CASE alati*
- ❖ *Kompletan inženjering poslovnih informacionih sistema*
- ❖ *Aplikativni softver: proizvodnja, finansije, bankarsko poslovanje*
- ❖ *3D – računarska grafika, CAD/CAM, arhitektura, geografski informacioni sistemi, vizuelno procesiranje, dizajn, animacija*

MOJ VIRTUELNI SVET

Da je dr Milan Božić atraktivan sagovornik, čovek koji se ne libi, ume i ima šta da kaže - bilo je jasno već onog trenutka kada je, pod kišobranom cyničnog slogana „baviću se politikom da se ona ne bi bavila sa mnom” - uzburakao domaću scenu i izvan ezoteričnih krugova univerziteta. Matematičar, erudita, ljubitelj filozofije i naučnik u najboljoj tradiciji aristotelovske škole mišljenja - dr Milan Božić s razlogom izaziva pažnju kako poklonika, sledbenika i saradnika, tako i ljutih protivnika. Ravnodušnih nema.

Kada je 1950. legendarni britanski matematičar i logičar Alen Turing zamislio čuveni test u kojem čovek sedi sam u sobi povezan teleprinterom sa drugim čovekom ili mašinom (pitanje je sad), dajući sve od sebe da otkrije sa kim (ili čim) je u vezi - kontroverza mogu li kompjuteri da misle stavljena je na dnevni red.

- Kontroverza povezana sa pitanjem „Mogu li kompjuteri da misle?” koja je, kako Vi kažete, iskrsla u vezi za Turingovim zamišljenim testom iz 1950, mnogo je starija, držim, nego ovaj vek!

Ovo što sam rekao može delovati paradoksalno, ali kada malo bolje razmislimo videćemo da je to pitanje, u stvari, preformulisano pitanje koje glasi: „Mogu li mašine da misle?” ili, još „filozofskije”, „Mogu li nežive stvari da misle?”. Tako gledano, ono je staro koliko i ljudski rod ili bar koliko filozofija sama, ako je shvatimo kao pokušaj samorazumevanja.

Moja filozofska pa prema tome i epistemološka pozicija je takva da na ovakva pitanja uvek odgovorim onako kako zamišljam da bi odgovarao autentični pozitivista. Držim, dakle, da je ovde po sredi problem značenja glagola „misli” ili imenice „misao”, ako je ona starija. Problem je u tome što u svima nama postoji jaka intuicija, a neke starije filozofske škole su je zdušno podržavale i obrazlagale, da je mišljenje osobina ljudskog roda i, eventualno, nekih životinja, živih bića, jednostavnije rečeno. Ako je tako, onda, naravno, kompjuteri ne mogu da misle.

Međutim, prethodni odgovor je prilično cyničan, jer mi veoma dobro znamo da onaj koji mišljenje pripisuje samo živim bićima, prečudno, pretpostavlja da su živa bića nešto „veoma posebno”, da su obdarena duhovnošću, dušom ili nečim sličnim, te da zato i mogu da misle. Ovu pretpostavku, to bar danas dobro znamo, savremena nauka veoma uspešno otklanja. Živa bića se, u osnovi, sastoje iz nukleotida i proteina - analogija sa softverom (nukleotidi) i hardverom (proteini) je prosto frapantna, ali je treba ostaviti u domenu „kompjuterske poetike” da ne bi izazvala nove rasprave - te su prema tome sasvim „obična” ali, naravno, veoma komplikovana „prirodna pojava” - kaogod i kvazari, koji, bar mene znatno više impresioniraju. Pri tome, naravno, otklanjam, gotovo kao vic, tezu da se živa bića razlikuju od kompjutera po tome što su nastala kroz evoluciju koja je trajala više miliona ili milijardi godina, jer to nije ono o čemu mi ovde govorimo.

Ako se naš hipotetični sagovornik, sa kojim, opet hipotetično, polemishem složi sa prethodnim, onda nam preostaje da razjasnimo značenje reči „mišljenje”. Jedino mu još tu ostaje nada da ospori tezu da kompjuteri mogu da misle. E, sada je on u neprilici a ne ja! Jer, on raspolaže samo veoma uopštenim defnicijama tipa „mišljenje je sposobnost snalaženja u novim situacijama”, i slično. Tu sam ja „na svome”, jer uvek mogu tvrditi da i kompjuteri to mogu. Ukoliko, pak, pribegnem „poslednjem sredstvu” i počne da nabraja neke posebne zaključke do kojih se stiže „pomoću mišljenja”, tek je onda aktivirao svoje pravo oružje! Sasvim je izvesno da kompjuteri mogu postići mnogo toga što i mišljenje. Izvesno je takođe da neke stvari ne mogu postići, na primer ja ću u uličnoj gužvi lako prepoznati svog prijatelja ili rođaka, a to kompjuteri ne mogu.

Međutim, i tu je kraj moje priče, nema principijelnih prepreka da to jednom ne bismo mogli. Ukratko, držim da kompjuteri mogu da misle, ukoliko je svet koji nas okružuje uopšte kauzalan, ali, bar za sada to čine prilično rudimentarno!

Sejmor Pejper sa MIT-a misli da se veštačka inteligencija može uporediti sa veštačkim letom; hipotetički skeptik bi po njemu rekao: - Vi matematičari imate posla sa zamišljenim fluidima - stvarna atmosfera je neuporedivo složenija! Medjutim, okreni - obrni, isti

Razgovor vodila: Vesna Ćosić

principi odnose se na stvarne i zamišljene fluide i primenjivi su bez obzira da li fluid struji preko perja ili aluminijumskog krila...

- Poređenje je sasvim dobro, odlično čak! Dupustimo hipotetičnom skeptiku da je u pravu. Međutim, on ovom tvrdnjom ne dokazuje ništa! Naime, avioni ipak lete i to je valjda bio glavni cilj braće Rajt a i svih poslenika avijacije kasnije. Oni nisu pravili pticu, oni su pravili - i uspeli da naprave - letilicu!

Na isti način oni koji prave računare - pri čemu ovaj izraz upotrebljavam u najširem smislu, odnosi se i na hardver i na softver - ne pretenduju da naprave mozak - ni pileći a nekmoli ljudski, bez obzira što su se u vreme mog detinjstva kompjuteri nazivali „elektronskim mozgovima” - već pravuju koja će obavljati izvesne jasno definisane poslove koje može da obavlja i ljudski mozak.

Tako, kao što avion nije ptica već „ume” da radi nešto što ume i ptica - to jest da leti - ali ne ume da

Živa bića se, u osnovi, sastoje iz nukleotida i proteina - analogija sa softverom (nukleotidi) i hardverom (proteini) je prosto frapantna.

nosi jaja, tako kompjuter ume nešto što ume i čovek - recimo da obavlja osnovne aritmetičke operacije što svakako nije sporno - ali ne ume da se, recimo, politički angažuje.

Takođe, kada se o ovakvim stvarima raspravlja, valja stalno imati na umu da pridev „veštački” koji počinje od ne baš najsrećnijeg, ali jedino mogućeg, prevoda engleskog prideva „artificial”, u srpskom jeziku ima negativan vrednosni predznak. Taj vrednosni predznak, mada postoji i u engleskom jeziku, nije u njemu toliko jak. „Artificial” označava više nešto što je „napravljeno” a ne nešto što je „veštačko”. Veštačko mleko, veštačke grudi ili veštački zubi nisu kod nas Srba neka osobito cenjena stvar. Moje insistiranje na ovom detalju može delovati pomalo neobično, ali imam utisak da ne malo nesporazuma i psihološke odojnosti prema „veštačkoj inteligenciji” leži i u nerazumevanju onoga što joj je stavljeno u zadatak, i u zameni teza, prema kojoj joj se pripisuje nešto što ona svakako nije.

Dobro, može li hrpa silicijuma i programa da dosegne mozak, jer ako jednom rast znanja čak i sazna sve zakone i principe koji kontrolišu svest-duh-dušu, može li se ona duplicirati? Uopšte, zaplćemo li se u paučinu filozofske razudjenosti roneći kontroverzama veštačke inteligencije?

- Najiskrenije, i svakako najtačnije, bi bilo da odgovorim: „Ne znam!”

Ipak, to me ne sprečava da malo spekuliram.

Pre svega, čini mi se da je prerano postavljati to pitanje matematičarima, inženjerima kao i ma kome ko se bavi računarstvom. Kažem „prerano”, jer nama još uvek nije formulisana zadatak. Šta mi to treba da uradimo ili napravimo da biste nam vi rekli „E, ova hrpa silicijuma i programa je dosegla mozak”?

Preciznije, to znači da biologija i medicina - o psihologiji ne bih ni govorio jer se ona veoma teško može svrstati u nauku, to je više neka vrsta esejistike koja pretenduje na egzaktnost - još uvek nisu popunile nedostajuće pločice mozaika i mi nemamo tačno definisan i formalno opisan objekat koji se naziva „mozak”. Ne tvrdim da je to uopšte moguće, ali bi se do rezultata najverovatnije moglo doći simbozom, sintezom računarstva sa biologijom i medicinom.

Tu leži odgovor i na drugi deo Vašeg pitanja, jer mi se čini da će „filozofska razudjenost” biti neizbežna posledica radikalne promene pojmovnog aparata koji će se tokom sledećih nekoliko decenija razvijati kroz pomenutu simbiozu.

Mi, recimo, na sasvim legitiman način, danas razumevamo elementarne čestice u fizici kroz aparat kvantne mehanike, ali je taj aparat svakako nedostupan filozofskom mišljenju postavljenom na klasičan način.

Ako je Turing bio u pravu i savršena simulacija razmišljanja jeste razmišljanje - šta reći za mišljenje filozofa Denijela Deneta koji je ovakvo stanovište nazvao „visokom crkvom kompjucionalizma”, u njene prelata strpao ličnosti kao što su Herbert Sajmon i Marvin Minski, a legendarni MIT etiketirao kao svojevrsni Vatikan?

- MIT je, nema sumnje, najugledniji svetski centar za „poslove ove vrste”. U tom smislu on jeste „svojevrsni Vatikan”. Takođe, sasvim je razumljivo što u savremenom svetu koji sve više postaje, ako već i nije postao, Makluanovo „globalno elektronsko selo” reklama, uspeh i publicitet igraju ogromnu ulogu, te se kolegama sa MIT-a ne može zameriti što svoje rezultate i uspehe, kao i svoje poglede na problematiku, glasno i agresivno zastupaju.

Čini mi se, međutim, da je Denetovo mišljenje, što je izgleda „bolest” svih filozofa koji su vaspitavani da misle globalno, znatno pretencioznije. Ono sadrži tezu da je MIT, ili krugovi bliski njemu, formirao nekakvu dogmu koju brani na način kako to čine crkve.

Ova teza naprosto nije tačna.

Prvo, prilično je sporno da uopšte postoji nekakva dogma koja bi se svodila na tezu „savršena simulacija razmišljanja = razmišljanje”, već bih pre rekao da je ono što se MIT-u, a i celoj savremenoj nauci, može pripisati, u stvari, jedan gnostički pristup. Naprosto postoji u čitavoj svetskoj nauci jedno jako ubeđenje da se svet može spoznati. Ovo ubeđenje je epistemološki karaktera i svakako ne spada u dogme. Koначno, u dosadašnjoj istoriji nauke, nije se dogodilo ništa što bi nas razuverilo u ovu predstavu. Naprotiv!

Drugo, aktivnosti koje, eventualno, proizilaze iz ove pretpostavljene dogme, sigurno nisu takve da idu na sprečavanje ili osujećavanje napora druge vrste, mada - neka mi bude dopušteno da budem pomalo sarkastičan - nije mi jasno koji bi to „napori druge vrste” uopšte mogli da postoje?

Zajedjlive knjige, poput one Hjuberta Drajfusa „Šta kompjuteri ne mogu da rade” - izazivaju talase polemika, ali i zanimljivih istraživanja...

- Svakako! Uvek je za jednu nauku važno da ima puno zajedljivih kritičara. Istina, oni mogu zadati grdne muke pojedinim naučnicima, ako se na njih lično okome, ali za nauku kao celinu, oni su od neprocenljivog značaja. Značaj takvih ljudi u savremenom svetu, što je pomalo paradoksalno, raste. Naime, savremenu nauku karakteriše praktično nestajanje naučne kritike. Ova pojava je posledica razudjenosti nauke, pojave esnafskog i ekspertskog žargona i, konačno, ali ne i najmanje važno, dramatične masovnosti savremene nauke. Naprosto, javnosti je i nerazumljivo i nezanimljivo bilo kakvo javno vrednovanje naučnih rezultata. Ono se obavlja unutar profesije, ponekad po ispravnim a ponekad po krajnje sumnjivim kriterijumi-

Držim da kompjuteri mogu da misle, ukoliko je svet koji nas okružuje uopšte kauzalan, ali, bar za sada, to čine prilično rudimentarno!



Dr Milan Božić: "Ovaj naš razgovor obačnom čoveku deluje kao razgovor ludaka. Voleo bih da nam se u 1993. dogodi, ne narod, dosta nam ga je bilo, nego vreme u kome ovaj intervju neće biti razgovor ludaka."

ma, ali ne dopire do javnosti i tako nauka ostaje bez čuvene povratne sprege koja ju je podsticala na razvoj sve do prvih decenija ovog veka.

Ljudi kao što je Drajfus upravo obavljaju ovo značajno posredovanje između javnosti i nauke. Oni to rade na prilično uprošćen, popularizatorski način i ne mogu izbeći kritiku da ovim načinom iskrivljuju stvarnu sliku: ali se sa razlogom možemo zapitati: „A kako bi inače mogli da stignu do šire publike?“

Kompjuterski program je ipak nešto više od talasa elektrona; Ako ga stavite u drugi kompjuter, gde je struktura silicijuma i metala potpuno različita, pa je i način kretanja elektrona različit – program je isti, čini isto. Da li to znači da smo i mi samo procesori neuronskih simbola? Jer i pojedinačni molekuli enzima,

Prived „veštački“ u srpskom jeziku ima negativan vrednosni predznak. Veštačko mleko, veštačke grudi ili veštački zubi nisu kod nas Srba neka osobito cenjena stvar.

lipida i DNA od kojih je sačinjena ćelija prilično su jednostavni, ponašaju se po poznatim zakonima fizike i hemije i ni jedan od njih se ne može izdvojiti i nazvati živim...

– Svakako da se mi ne možemo nazvati „procesorima neuronskih simbola“ (bez obzira što je izraz nejasan, mislim da otprilike shvatam na šta ste mislili) zbog puke okolnosti da isti program, kada promeni „stanište“ tj. kompjuter radi opet isto. Međutim, analogija između nas i kompjutera se ne može izbeći, ona nam se, šta više, sve više nameće kako savremena biologija, molekularna pre svega, napreduje.

Pre svega, slika živih organizama, koju nam daje savremena molekularna biologija, neodoljivo podseća na kompjutere.

Centralna dogma molekularne biologije tvrdi, a sva teorijska i eksperimentalna istraživanja idu u prilog ove teze, da genetsku informaciju nose samo nukleinske kiseline a ne proteini, koji služe kao nekakav hardver. Nukleotidi, osnovne jedinice od kojih su sastavljene nukleinske kiseline, neposredno korelišu sa tokenima nekog programskog jezika.

Osim toga, svi eksperimenti, u koje je poslednjih decenija investirano i mnogo pameti i mnogo novca, pokazuju da se repliciranjem nukleinskih kiselina, bar na nivou jednostavnih organizama, mogu replicirati i ti organizmi, što je krunski dokaz o hemijskoj ili – ako smatrate da je to zasebna disciplina – bio-hemijskoj osnovi života, koja se može opisati bez ostatka.

Uspesi koje je postigla molekularna biologija su u toj meri učvrstili naše ubeđenje da između kompjutera koji izvršava neki program i živih bića nema nikakve

principijelne razlike, da je vrlo teško pronaći naučnika koji bi, danas, tvrdio suprotno.

Što se odgovora na poslednji deo pitanja tiče, on je u svetlosti ovakve metodologije jednostavan – ni procesor, ni memorija, ni sredstva za masovno skladištenje podataka, ni sami programi ni periferni uređaji nisu, uzeti svaki za sebe, radeći kompjuter. Njega čine sve ove komponente zajedno, i to samo ako su funkcionalno povezane.

Nije tajna, digitalizacija je učinila komuniciranje između ljudi i mašina, između mašina, pa i mašina sa ljudima lakim kao što je to ono među ljudima. Da li je dan kada će i ukusi, mirisi i što da ne, osećanja, moći da se pohranjuju, obradjuju i prenose – već tu negde iza ugla? Ili je bard SF-a Isak Asimov bio u pravu tvrdeći da „ništa ne može da zameni iskustvo iz prve ruke“?

– I ovde pošten, iskren i, pre svega, tačan odgovor mora da glasi: „Ne znam!“ Naime, okolnost da život, terijski i principijelno, za nas više nije tajna ne mora da znači da ćemo mi, ikada, moći sa njim da „radimo šta god hoćemo“. Tako, na primer, možda mi nikada nećemo moći da napravimo „veštačkog čoveka“, tj. živo biće, sintetizovano u laboratoriji koje do na molekul, i što je mnogo važnije, do na njihov raspored!, odgovara nekom drugom čoveku.

Jednom rečju, posedovanje informacije o nečemu ne povlači, automatski, posedovanje tehnologije da se to „nešto“ kontrolišu ili replicira.

Tako, na primer, mi danas gotovo savršeno poznajemo zakonitosti nuklearne fuzije i znamo da je ona potencijalni izvor ogromnih energetske resursa, koji bi na duge staze, a možda i zauvek, rešili sve naše energetske probleme. Međutim, niko do sada ni-

Uvek je za jednu nauku važno da ima puno zajedljivih kritičara.

je napravio fuzioni reaktor. Šta više, nisam uopšte siguran da će on ikada biti napravljen. A evo i zašto. Pod fuzionim reaktorom se podrazumeva uređaj koji radi u linearnom, dakle, stabilnom, režimu, a ne u impulsnom – takve „uređaje“ već posedujemo, to su nuklearne bombe, ali ne verujem da će se ma čiji energetski problemi rešiti ako mu spustimo jedan takav „uređaj“ na glavu – i proizvede više energije nego što se u njih ulaže. Sasvim je moguće da će se jednog dana ispostaviti, izračunati i dokazati, da je energija potrebna za održavanje stabilnog polja od stotinak miliona stepeni veća od energije koju nam takvo polje fuzijom čestica koje su u njemu vraća natrag, i naš san o fuzionom reaktoru će biti raspršen.

Slično tome, možemo zamisliti hipotetičnu budućnost u kojoj će postojati mogućnost da se prikupe ap-

Šta mi to treba da uradimo ili napravimo da biste nam vi rekli „E, ova hrpa silicijuma i programa je dosegla mozak“?

solutno sve informacije koje, na primer ja, u sebi nosim, ali se ipak neće moći napraviti uređaj koji će po liku prepoznavati mog komšiju u ma kom realnom vremenu, dok ja to činim, što bi rekli, „u trenutku“.

Lično, ne mislim da je takva budućnost stvarna i imam utisak, možda preterano optimistički, da je manje-više sve što se može saznati i ostvarivo.

Naša mašta teži da se vezuje za ograničenja iz prošlosti; vizija sutrašnjice često je obazrivo konzervativna i podseća na istraživanje koje je Mercedes Benc sproveo 1900. godine, u kome se predviđa da svetska potražnja za automobilima nikada neće preći cifru od milion, pre svega zato što neće biti dovoljno šoferi.

– Prema takvim pitanjima imam dvojak odnos. Sa jedne strane, nema nikakve sumnje da je naša stvarnost nadmašila i najludše snove naših predaka. Tako gledano, maštanja a la Žil Vern treba ne samo podržavati nego ih valja i podsticati. Lično sam ubeđen da Žil Vern nije samo „predvideo“ neka od čuda savremene tehnike već je i sam uticao na njihovo stvaranje – neki od dečaka koji je čitao njegove knjige sigurno je kasnije postao naučnik ili inženjer, pa je, vođen se-

Mislim, dakle, da će civilizacija biti najsrećnija i najuspešnija ako Zemljinom kuglom, ravnopravno, budu hodili i veliki sanjari ali i skeptici.

čanjima na literaturu svog detinjstva, bar nešto od toga pokušao da ostvari, i, bogami, ostvario!

Sa druge strane, neodmerena maštanja mogu vući istraživanja pa i čitave kulture na stranputicu. Pri tome ne mislim na budalaštine Denikenovskog tipa, jer to je ipak literatura za razbibrigu, već na promašaja je ozbiljnih veličina kao što je, na primer, bio Nikola Tesla. Setite se samo njegove fantazmagorije o „bežičnom prenosu energije“ i problema koje je izazvala i njemu i drugima.

Mislim, dakle, da će civilizacija biti najsrećnija i najuspešnija ako Zemljinom kuglom, ravnopravno, budu hodili i veliki sanjari ali i skeptici, jer, skepsa je ponekad dobra. Ona, naime, sprečava slobodan razmah gluposti!

Međutim, nije mali broj istraživača koji smatraju da će doći do trajne konvergencije informativnih tehnologija – televizije, telefona, kompjutera i drugih – i stvaranja jedinstvenog komunikacionog sistema. Neki je opisuju kao „telematik“ – kombinovanje kompjutera sa telekomunikacionim tehnologijama, drugi govore o „informatiku“ – računanje i informacije, uključujući i vesti, treći elaboriraju „kompjunikacije“ – računanje i komunikacije...

– Neke od ovih izraza sam čuo, a neke ne. Sve mi se nešto čini da su njih smislili dokoni naučnici, ali i novinari koji moraju popunjavati stranice svojih listova!

Nema nikakve sumnje da će u budućnosti doći do konvergencije informativnih tehnologija. Ono što je gotovo izvesno je njihova digitalizacija, što, via facti, zapravo označava njihovu kompjuterizaciju. Druga izvesnost je pojava, već i sada prisutna u najrazvijenijim zemljama ali trenutno veoma skupa, komunikacionih kanala ogromne propusne moći – recimo, reda veličine terabajta u sekundi.

Ova dva elementa su sasvim dovoljna da u razumnoj budućnosti objavimo da je objedinjavanje informacionih tehnologija obavljeno – „sve ostalo su in-janse“, što rekao Đoka Balašević.

Nijanse, međutim, mogu da budu značajne, jer je unutar zdrave pameti da će se informacione tehnologije razvijati kroz komercijalnu primenu, te će moćne korporacije veoma insistirati na konkurenciji i odsustvu standardizacije. To je verovatno šteta, sa tačke gledišta korisnika, ali je verovatno i korisno, jer će stimulisati konkurenciju, pa samim tim i razvoj.

Revolucija komunikacija, upravljanja, informacija, revolucija informativne tehnologije, kompjuterska, naučna i tehnološka – u postindustrijskom svetu, nije sporno, razvija revolucionarne načine prenošenja informacija. Pored idile „Maklunanovog sela“, nova realnost nudi i manipulatore, diktatore, profesionalne „izmišljače“ orijentisane na uništavanje granice između stvarnosti i mašte. Suočavanje sa tipovima a la Bak Rodžers, novim oblicima ograničavanja slobode nije više samo na polici sa delima SF produkcije...

– Ne bih rekao da se radi o novim oblicima ograničavanja slobode već, ipak, o novim načinima ograničavanja slobode. Razlika, na prvi pogled, ne deluje značajno, ali mislim da je značajna sa stanovišta načina „borbe protiv“, jer možemo koristiti već postojeća iskustva naše civilizacije.

Svaka manipulacija informacijama odnosno onim koga (dez)informišemo je uvek ciljana – ima neku svrhu, zadatka. Najupoštenije rečeno, cilj je stvaranje nekog virtuelnog sveta u kome treba da misli da živi potencijalna žrtva ove manipulacije. Način kako aktuelni režim u Srbiji vodi informisanje preko državne televizije danas ili Crkva, pogotovu hrišćanska, koja je to veoma uspešno izvodila preko hiljadu godina, paradigme su ovog stvaranja virtuelnog sveta.

U svakom od istorijski značajnih slučajeva koje odaberemo, ja sam odabrao ova dva, jer je jedan izuzetno dugo trajao a drugi osećamo na svojoj koži, mi možemo uočiti jednu trivijalnu zakonitost – manipulacija informisanjem uspeva samo ako postoji monopol na informisanje.

Prema tome, naš hipotetični Bak Rodžers ima šansu da uspe samo ako mu mi stvorimo politički milje u kome će on biti jedini posednik sredstava za saopštavanje istine, što nas vraća u ravan prethodnog pitanja i odgovora na njega. Da zaključim, samo „checking and ballance“ – ravnoteža i uzajamni nadzor, kako se u američkom političkom sistemu opisuje nezavisnost sudske, zakonodavne i izvršne vlasti –

Okolnost da život, terijski i principijelno, za nas više nije tajna ne mora da znači da ćemo mi, ikada, moći sa njim da „radimo šta god hoćemo“.

proširen i na oblast informisanja može da nas spase od potpunog ograničavanja slobode. Delimičnih ograničavanja će uvek biti.

Život unutar sopstvenog mentalnog videa, koje podržava novotehnološka stvarnost, kreira nove zidove. Današnji gledaoci sopstvenih neprekidnih filmova i sutrašnji uživaoci u usavršenim kompjuterskim bicima, sposobnim da vode na neobuzdane mentalne puteve, a la „Neuromancer“ Vilijama Gibsona – generacija su sa kojom treba računati. Znači li to da će se ostvariti proročanstvo Marvina Minskog – da će pisci naučne fantastike, stoga što najozbiljnije pokušavaju da shvate posledice i implikacije pojedinih stvari, poput Isaka Asimova i Vilijama Gibsona, kroz nekoliko stotina godina biti smatrani važnim filozofima XX veka, a profesionalni filozofi biti načisto zaboravljeni?

– Ne verujem! Mada nije nemoguće da će savremeni vrhunski pisci naučne fantastike jednog dana biti ocenjivani povoljnije i značajnije od savremenih sociologa i psihologa.

Filozofija, kao pokušaj samorazumevanja je, pak, nešto sasvim drugo i mnogo ozbiljnije.

Kao što i prvi deo Vašeg pitanja sugeriše, oni se ipak bave problemom virtuelnih svetova i istražuju moguću implikaciju. Problem nije u njihovom odsustvu sofisticiranosti, analitičke dubine ili analitičkih pretpostavki, već u pukoj okolnosti da se oni bave istraživanjem hipotetičkih grananja stvarnosti. Mi sada ne znamo, a ne možemo čak ni da slutimo, koja će od ovih grananja postati stvarnost ili deo stvarnosti, koja će ostati u domenu sopstvenog „mentalnog videa“, a koja će postati toliko značajna da na nas utiču ili nas, čak, ugroze. Tako nešto ipak spada u književnost a ne u filozofiju.

Alvin Tofler, pisac „Šoka budućnosti“, „Trećeg talasa“, „Zaokreta moći“ – opisuje „skankvork“ (skunkvork) organizacije, specifičan oblik timskog rada, rasterećen birokratskih zaduženja i hijerarhijskih opterećenja. Primer okupljanja preko elektronske mreže (gde su veze između sektora, ranga i geografske lokacije irrelevantne), vama je blizak, zar ne?

– Da, sticajem profesionalnih okolnosti ali i interesovanja, stalan sam korisnik i, možda, ovisnik takvih sredstava.

U ovom trenutku, one su u povelju i predstavljaju tek jednostavniji i brži način distribuiranja inače standardnih informacija koje bi se mogle zamisliti i na papiru. Osim toga, nama koji smo profesionalci, one omogućuju dostup određenim bazama podataka i kompjuterskim resursima po svetu, što za širu publiku uopšte nije interesantno.

Međutim, mada ne znam kako će to stvarno izgledati, ove mreže otvaraju neslućene mogućnosti promene tipa informisanja, pa samim tim i načina života.

Redovne konferencije koje se vode na njima, recimo, omogućavaju ljudima koji se uopšte uzajamno ne poznaju, niti postoji šansa da bi ikada stupili u kontakt – geografske, klasne i intelektualne razlike među njima su prevelike – da razmenjuju mišljenja i, što je možda važnije, alternativne informacije.

Recimo, tokom ovog rata, komunikacija sa Zagrebom, Ljubljanom i našim i njihovim političkim i ekonomskim emigrantima po svetu, nije ni jednog trenutka bila prekidana. Imam utisak da smo se mi, koji obitavamo po tim mrežama, mnogo bolje razumeli, iako imamo suprotstavljene interese, nego i najvestiji i najpomirljiviji politički pregovarači na obe strane.

Nove metode skladištenja ogromnih količina digitalnih informacija probile su svoj put kroz istraživačke laboratorije, jureći prema komercijalnoj, pa čak i potrošačkoj primeni. Život sabijen u katridž daje novi „prostor za kretanje“. Svetovi privida, simulacije, donose na ekrane novu stvarnost. Moram da vas pitam, iskoracite li i koliko iz ovozemaljskog i praktičnog u uzbudljivo i nemoguće? Igrate li se?

– Još uvek sam skloniji klasičnim vidovima „iskoračenja“ iz stvarnosti kao što su literatura, muzika ili film i televizija. Istina, u domenu vizuelne prezentacije film i televizija su veoma kontaminirani kompjuterizacijom, te mi se tako kompjuteri vraćaju „na mala vrata“ u moj virtuelni svet.

Ipak, verujem da je to tip senzacija koje se ne mogu zaobići, ali će samo mladim generacijama, koje su sa njima rasle, značiti i nerazdvojni deo života.

Što se igre u užem smislu reči tiče, obogaćavam kompjuterske igre, ali, i time ću razočarati mnoge ljubitelje sofisticiranih igara koje traju satima i danima, mnogo sam skloniji igrama a la Tetris, mada priznajem da se lako može braniti teza da one, prvenstveno, služe abregovanju neuroza, da su nešto poput štrikanja.

Virtuelna stvarnost medijum je u povelju. Neće proći još dugo vremena, a tehnike i tehnologije virtuelne stvarnosti predstavljaju standard zabavnog, poslovnog i komunikacionog okruženja. Bliski susret sa „prisustvom na daljinu“ podrazumeva kombinaciju velikog broja „interfejsa“. Koliko domaći ambijent, takav kakav je, ima šanse da se razvija u tom pravcu?

– Pa, kao i u svim dosadašnjim industrijskim, tehnološkim i političkim „revolucijama“ – mi možemo očekivati da budemo tek recipijent nečega što je stvoreno van nas, nezavisno od nas i ne za nas. Mislim da u tome i leži jedan od glavnih uzroka naše ksenofobije, naše nesposobnosti da se uklopimo u svet. Nepoznato nam se čini neprijateljskim. Ovo je, naravno, opšte mesto, jer se može pripisati svakom primitivnom narodu, ali to ne znači da mi treba da budemo srećni i zadovoljni zbog toga što smo primitivni.

Međutim u svakoj nesreći ima i neke pouke, prilike za boljitak. Iz ove nesreće – informatičke revolucije – koja nas upravo „snalazi“, pruža namse prilika da izvučemo korist. Naime, globalnost promena i njihova brzina, učiniće naše kalificirane socijalne moćnike smešnim i trapavim, lakše će se sasušiti i otpasti, a mi ćemo lakše uhvatiti korak sa svetom. Ne zato što to želimo – mi se tome žestoko opremo – već će nas šel na to naterati. Oni više nemaju vremena!

Vaš pogled u 1993...

– Za 1993. je jedino izvesno da će nastupiti.

Sve ostalo je u domenu nagađanja. Bilo bi ili cinično ili neukusno da u ovom užasu koji nas okružuje izjavljujem ma šta.

Da ovaj moj odgovor, ipak, ne deluje previše tužno, evo i jedne optimističnije slike. Ovaj naš razgovor, kada se pogledaju teme koje smo dotakli, našem običnom čoveku, čak i ako je intelektualac a razume ono o čemu smo govorili, deluje kao razgovor ludaka. Voleo bih da nam se u 1993. dogodi, ne narod, dosta nam ga je bilo, nego vreme u kome ovaj intervju neće biti razgovor ludaka.

Nastavak sa strane 29

Raznolikost tehnologije daje šansu Intelovim konkurentima da postignu uspeh sa procesorima koji koriste otvaranje sasvim novih segmenata tržišta kao i rupe u Intelovoj proizvodnoj liniji. Na primer, AMD je uspeo sa svojim 40 MHz 386DX čipom, jer je odgovorio na potrebu za 386-ticom visokih performansi. Intel je istisnut iz pojedinih segmenata tržišta, a tek treba da ostvari ozbiljno uporište na tržištu „ručnih“ kompjutera. Međutim, Intel je udružio snage sa kompanijom „VLSI Technology“, u nameri da proizvede čip-set klase 386SL za „ručne“ mašine.

Intel je spreman da ponudi čak i čipove prilagođene specifičnim aplikacijama. Njihov *RapidCad* koprocesor, lansiran u februaru prošle godine, jeste dogradnja u vidu dva čipa koji odgovaraju 386 i 387 utičnicama. Utičnicu 386 zauzima čip koji predstavlja hibrid procesora 386 i 387, dok utičnicu 387 CPU koji se radi po narudžbi (custom) i rukuje izuzecima (tj. greškama tipa „deljenje sa 0“). Ovaj čip-set radi na bilo kojoj 386 mašini do 33 MHz i daje performanse bolje za oko 20-30% kod izrazito grafičkih aplikacija, kao što je CAD.

SLEDEĆA GENERACIJA

Mada Intel nije nameravao da lansira P5 pre početka 1993, mnogi detji o ovom mikroprocesoru već su obelodanjeni. Imaće superskalarnu arhitekturu, što znači da će imati dve ili više izvršnih jedinica. To će omogućiti procesoru da tokom jednog ciklusa paralelno izvršava više instrukcija.

Intidžer jedinica čini paplajn od pet nivoa čipa 486 još bržim, tako što dva rade paralelno. Paplajn dekodira dve instrukcije istovremeno, pa ako nisu konfliktne, izvršavaju se paralelno. Ovaj dvostruki paplajn može za 50% ubrzati čip ranijeg dizajna. Dva druga ključna dela intidžer jedinice su *branch target buffer* (ciljni bafer za instrukcije grananja – „branch“) i keš sa dvostrukim pristupom. *Branch target buffer* predviđa pojavu ogranka; ako je predviđanje tačno, ogranak se izvršava bez zastoja. Keš sa dvostrukim pristupom prima i podatke i adrese iz papjova i sadrži logiku za rešavanje zavisnosti adresa.

Matematički koprocesor čipa P5 ima tri posvećene aritmetičke jedinice i paplajn od osam nivoa koji je integrisan sa intidžer paplajnom, ali ima još dva izvršna nivoa. Mada je matematički koprocesor podešen za dvostruko precizne *memory-to-register* operacije (najčešće očekivani tip), Intel tvrdi da se jednostrukom precizne i *register-to-register* operacije obavlja ju istom brzinom.

486 koristi jednomikronsku tehnologiju da bi smestio 1.2 miliona tranzistora; P5 koristi 0.8-mikronsku separaciju i ima 3 miliona tranzistora. Intelovi ljudi kažu da će prva verzija čipa P5 raditi na 66 MHz i preko 100 MIPS-a. Izrazito grafičke aplikacije, serveri sa bazama podataka za klijent/server sisteme i multitasking aplikacije samo su neki od korisnika koji će na najbolji način iskoristiti njegovu snagu.

Ali, Intel nije jedini proizvođač koji ide dalje od 486-tice (vidi tekst „Novi čipovi na horizontu“). Subodh Toprani, direktor marketinga i inženjeringa personalnih računara kompanije AMD, kaže da „imaju značajne planove za čipove koji će prevazići 486“. Toprani ističe da je AMD vodeći snabdevač proizvodima RISC arhitekture, što je vrlo značajno, s obzirom da će RISC tehnologija biti ključni faktor u sledećoj generaciji čipova. Toprani kaže da će sledeću generaciju čipova ponuditi tokom 1993.

Cyrix, sa svoje strane, najavljuje da će odmah nakon pojave P5 ponuditi svoj P5-kompatibilni procesor, dok će novu generaciju čipova, takozvanu klasu 686, izbaciti pre Intela. To će uspeti zahvaljujući svojim kraćim proizvodnim ciklusima. Slejter iz *Microprocessor Report-a* kaže: „Gotovo svi su potcenili teškoće koje će se javljati pri integraciji ovih superskalarnih čipova. Dok ne vidimo „Cyrix-ove“ čipove, ovakve izjave moramo primati s rezervom.“

Bilo kako bilo, mnoštvo različitih mikroprocesora stvara neodumice pri izboru sistema. Međutim, ta nevolja je potpuno zanemariva u odnosu na drastično unapređenje performansi i pojeftinjenje sistema koje je donela konkurencija među proizvođačima mikroprocesora. „Upotreba kompjutera postaje sve jednostavnija,“ kaže jedan predstavnik kompanije „Chemical Bank“, „Što budu imali veću moć, to će ih ljudi više koristiti.“

Endi Redfern, Byte

Prevela: Ranka Jovanović

EPSON

ZAŠTITA

- LX 400
- FX 1050
- DFX 5000
- LQ 100
- LQ 570
- LQ 870
- LQ 1070
- LQ 1170
- LQ 2550
- DLQ 2000

SERVIS

DODATNA OPREMA

REZERVNI DELOVI

KOLIČINSKI POPUST

- HW KLJUČEVI /HARDLOCK/
- ANTIVIRUSNI PAKETI
- SREDSTVA ZA KONTROLU PRISTUPA
- KRIPTO ZAŠTITA
- ZAŠTITA PRI PRENOSU PODATAKA
- STRUČNI SAVETI
 - ZA REŠAVANJE SISTEMA ZAŠTITE
 - ZA IZBOR OPREME
 - ZA OPTIMALNO KORIŠĆENJE POSTOJEĆE OPREME

TELEFAX
(021)616-887

TELEFONI
(021)623-928
624-501

Novi Sad
Bulevar 23. oktobra 25
KONSALTING I INŽINJERING



UPS

- 300 VA
- 550 VA
- 1000 VA
- ON LINE UPS
/prema zahtevu/

ADRESA: GRADSKI PARK B.B. (ZGRADA CRVENOG KRSTA), 11080 ZEMUN, TEL./FAX: 38 11 613 822
 ★ PREDSTAVNIŠTVO U BUDIMPEŠTI: ANDRÁSSY UT 122, 1062 BUDAPEST, HUNGARY, TEL./FAX: 36 1 132 4923



ČAO!



PERSONALNI RAČUNARI - ŠTAMPAČI - PRATEĆA OPREMA

QUATTRO... PA SVI OSTALI

Pre nekoliko brojeva prikazali smo, u najkraćim crtama, *Quattro Pro for Windows*. Tada smo imali na raspolaganju samo „beta release“, a sada imamo kompletan program. S obzirom da se radi o zaista „vrućoj stvari“, ponovo pišemo o istom paketu. Prenećemo vam neke nove utiske stečene u radu uz korišćenje kompletne dokumentacije, a ponovićemo i nešto od prvih utisaka koji se nisu promenili.

Program se isporučuje u, za Borlandove proizvode, standardno ogromnoj kutiji sa gomilom dokumentacije i pet disketa od 5.25", 1.2 Mb. Od uputstava za rad tu su:

- *User's Guide*, korisnički vodič, kompletno uputstvo za rad sa tabelom na 450 strana.
- *Getting Started*, kratko uputstvo o instalaciji i mogućnostima programa (90 strana)
- *Database Desktop Guide*, uputstvo za rad sa bazom podataka (150 strana).
- *Building Spreadsheet Applications*, knjiga sa kompletnim pregledom funkcija, makroa i njihovom upotrebom u izgradnji korisničkog interfejsa na 460 strana.
- *Quick Reference Guide*, kratki pregled ikona, komandi, funkcija i makroa s kojima raspolaže QPW.

Odmah treba reći da je QPW, kao i većina drugih *Windows* programa, ima velike zahteve u pogledu hardvera. Nadao sam se da tako neće biti u ovoj finalnoj verziji, ali ipak treba obezbediti računar sa više od 4 Mb RAM-a, dok je na disku neophodno oko 10 Mb slobodnog prostora. U suprotnom, od instalacije nema ništa. Interesantno je da nakon instalacije QPW zauzima, ukupno, manje od 8 Mb prostora na disku. Onaj zahtevani višak od oko 2.5 Mb stavljam na dušu instalacionom programu. On je urađen veoma atraktivno, pokreće se iz *Windowsa*, ali je prilično spor, mada brži od beta-verzije. Da bi raspakovao i prebacio datoteke sa četiri diskete na disk, potrebno mu je više od 35 minuta, a može vam se desiti da na samom kraju instalacioni program prijavi neku grešku, ko zna iz kojih razloga, pa ćete morati sve ispočetka.

Moram da kažem da su mi veliku pomoć pružila gospoda iz firme „Mladost“ Holding – Loznica, Beograd, koji su mi, za ovu priliku, dali na upotrebu računar 386 SX sa taktom od 25 MHz, 8 Mb RAM-a, VGA karticom i monitorom, i diskom od 220 Mb. *Windows* i svi programi za njih su radili kao sat. Odličan računar.

NAJLEPŠI NA SVETU

Nakon završene instalacije možete pokrenuti program. Kao što sam u prethodnom članu napomenuo, a mišljenje nisam promenio, nema lepšeg i atraktivnijeg interfejsa u svetu *spreadsheet* programa. Ponoviću neke svoje osnovne utiske, jer to ovaj program zaslužuje.

Središnji i najveći deo ekrana, razumljivo, zauzima tabela. Ona se sada zove **Notebook**, a svaka ima svoj naziv, kao NOTEBK1.WB1, NOTEBK2.WB1, itd., što ne smeta da tabelu snimite pod nekim svojim imenom. Možda je najzanimljivija novost u programu da je prihvaćen Lotusov koncept trodimenzionalne tabele (iz verzija 3.xx). Naime, u QPW svaki **Notebook** ima do 256 tabela i jednu grafičku stranu. Sve to zajedno se smešta u jednu datoteku! Vrlo je lepo rešenje, u interfejsu, za promenu aktivne tabele. One su poredane jedna iza druge, a svaka ima izrez, kao u rokovniku, obeležen nazivom tabele (od A do IV, a može dati svakoj strani i svoje sopstveno ime).

Vladimir Stamenović



Vidljivo je samo sedam prvih naziva tabela, dok za ostale postoji, u donjem levom uglu, mali skrol-bar pomoću koga se može doći i do ostalih tabela. Brzi prelaz iz aktivne tabele u poslednju i obrnuto se ostvaruje klikom na **SpeedTab** dugme, koje se nalazi u sredini donjeg reda na ekranu, desno od oznaka tabela, a obeleženo je strelicom levo ili desno, zavisno od toga na koju stranu se krećete. Za prelaz iz jedne u drugu tabelu, dovoljno je da mišem kliknete na odgovarajuću oznaku. U levoj polovini donjeg reda ispod tabele i sa leve strane tabele, nalaze se horizontalni i vertikalni skrolbarovi, koji, kao i kod svih *Windows* programa, služe za kretanje po tabeli.

Iznad tabele, u prvom redu, nalaze se komande za rad sa programom. Njihov raspored, aktiviranje, kao i skraćenice za rad su uobičajeni u svakoj vrsti programa. Prva po redu je **File** komanda kojom se barata sa datotekama, štampaju dokumenti i izlazi iz programa. Sledeća je **Edit** komanda, koja omogućuje rad sa **Clipboard**-om (**Cut, Copy, Paste**), a sadrži i komande **Goto, Undo** i **Search**. Za reorganiziranje tabele služi komanda **Block (Move, Copy, Fill, Reformat, Insert** i **Delete** kolona, redova i strana). Rad sa bazom podataka se odvija kroz komandu **Data**, a veoma zanimljive mogućnosti nudi komanda **Tools**. U njen domen spadaju rad sa makroima, kombinovanje i linkovanje datoteka, rad sa matricama, „šta-ako“ analize, **Solver** i mogućnost promene izgleda meni-bara (*UI building*). Grafičku prezentaciju podataka ćete raditi kroz komandu **Graph**. Tu možete kreirati, doterivati i posmatrati grafičke i tzv. *Slide show*, kao i ubacivati gotove slike (*clip art*). Komandom **Property** možete vršiti razna podešavanja za svaku vrstu objekta, dok za reorganiziranje prozora služi komanda **Window**. Na kraju menija je komanda **Help**.

Ispod menija nalazi se „bar“ sa nizom različitih ikona koje bi trebalo da ubrzaju najčešće obavljane poslove u tabeli (pun naziv je **Notebook Window SpeedBar**). Ima ih dvadesetak. Prve tri služe umesto komandi **Cut, Copy** i **Paste**. Podaci iz ćelija ili blokova se vrlo brzo,

pomoću ovih ikona, mogu isecati, kopirati i premeštati bilo gde u tabeli, u sve tri dimenzije. Tu je i ikona za ubacivanje simbola za dugmad kojima će se izvršavati neke makro procedure. Brzo kreiranje grafikona, poravnavanje teksta, na sva tri načina, promena formata teksta (**bold** i *italics*) i brojčanih podataka je samo deo poslova koji su znatno ubrzan preko ikona. Postoji i ikona koja omogućuje brzo sabiranje, promenu visine reda ili, vrlo interesantna, **SpeedFormat** ikona, koja otvara dijalog-boks sa pet unapred definisanih formata za tabelu. To znači da je već određeno kako će tabela izgledati, sa iscrtanim linijama ispod zaglavlja, osenčenim važnijim blokovima, različitim tipovima fontova... Impresivno! Treba reći da se „bar“ sa ikonama menja u zavisnosti od posla koji se radi. Recimo, ako editujete grafikon, tu su potpuno nove ikone koje omogućuju veoma komforan i lagan rad na doterivanju izgleda grafikona.

U trećoj liniji ekrana je smešten „edit bar“ (**Edit Mode SpeedBar**). U njegovom levom delu se vidi adresa aktivne ćelije ili bloka, a ako pritisnete taster **F2**, onda se u nastavku vidi i sadržaj koji se može korigovati. Pri unosu podataka od pomoći su ikone koje ubrzavaju posao oko, na primer, ispisivanja makro komandi (zajedno sa vitičastim zagradama).

Borland-ovi programeri su učinili veliki napor da korisnicima omoguće komforan i brz rad i u tome su u potpunosti uspeali.

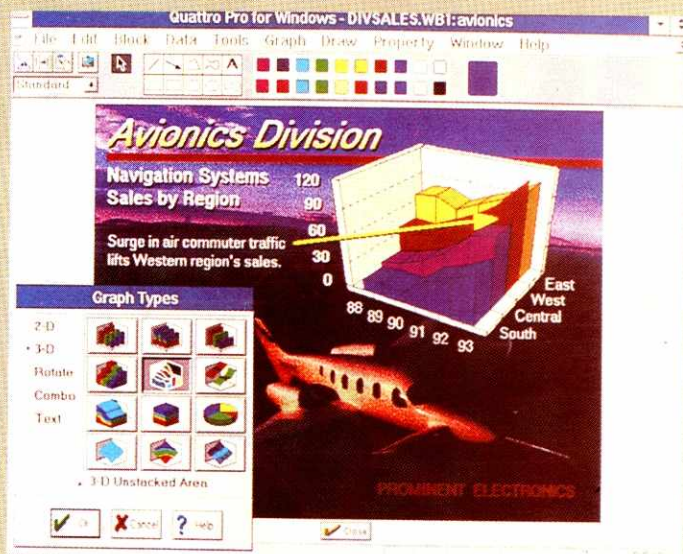
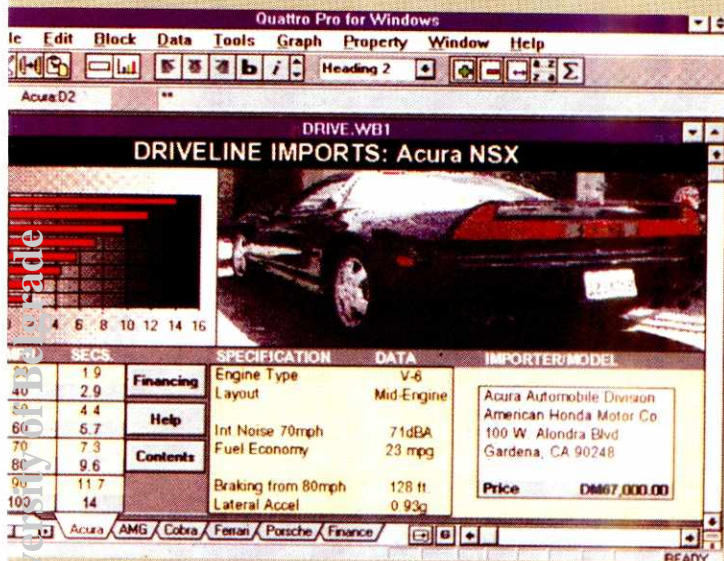
S obzirom da je ova verzija QPW instalirana na dobrom 386 računaru, program je radio veoma brzo. Ni traga od nepoželjnih i sporih osvežavanja ekrana. Ikone su brzo odgovarale na aktiviranje, pa je rad pravo zadovoljstvo.

Dokumentacija je, po običaju, sasvim zadovoljavajuća i pomoću nje se mogu relativno brzo savladati osnove programa. **Help** je takođe na zadovoljavajućem nivou i iz njega se mogu saznati sve ili gotovo sve osnovne stvari o programu.

RADNI PROSTORI

QPW ima pet radnih prostora. Prvi je **Notebook Window**, prostor gde unosimo i prikazujemo podatke, a mi ga nazivamo tabela. Sadrži sve standardne mogućnosti *spreadsheet* programa, a naravno, i mnoga proširenja i dodatke. U njemu se obavlja najveći deo posla. Kreiranje, doterivanje i ispravka grafikona, baziranih na podacima iz tabele se obavlja u **Graph Window**. Ako pravite sopstvenu aplikaciju koja će raditi u QPW, korišćete **Dialog Window**. **Graphs Page** je poslednja strana u svakoj tabeli (**Notebook**). Tu vršite povezivanje grafikona i slika u *slide show*, ili povezujete sopstvene dijalog-boksove koji će činiti neku aplikaciju. Peti radni prostor je **Print Preview Screen**, gde se može pregledati rad pre štampanja.

Da se vratimo tabeli. Dakle, kao što sam rekao, imamo trodimenzionalnu tabelu. Svako ko je ikada radio sa Lotusovim 1-2-3 verzija 3.xx, zna kakva je to prednost u odnosu na „običnu“ tabelu. Primera radi, ako imate nekoli-



ko tabela koje služe za obavljanje sličnih poslova ili su, čak, međusobno linkovane, onda je daleko čistija situacija ako svaku od tih tabela ubacite u poseban list u jednom **Notebook-u**. Tu je povezivanje podataka daleko lakše, a rad znatno brži. Na kraju krajeva, umesto nekakvo datoteka imaćete samo jednu. Poslednja strana u **Notebook-u** je grafička strana koja sadrži ikone koje predstavljaju svaki od napravljenih grafikona, *slide show* ili dijalog boksove koje ste sami kreirali. Karakteristično za QPW je da skoro svi elementi koji se vide na ekranu predstavljaju objekte. Svaki od njih se može posebno podešavati i doterivati. Na primer, u objektu koji se zove *Aktivni blok*, možete vršiti najrazličitija podešavanja na aktivnoj šeliji ili bloku, ali to isto, ali na **Graph** objektu.

Kao u verziji QP 4.0 za DOS, i ovde mi se najviše dopalo da radim u grafičkom prozoru. Pravo je zadovoljstvo kreiranje i doterivanje grafikona, mešanje sa gotovim sličicama (probao sam samo .CGM format) ili njihovo preuređenje. Ovaj deo programa je urađen vrlo kvalitetno, sa dosta mašte i pažnje. Kada editujete grafikon ili sliku, u posebnom prozoru, u **SpeedBar-u** dobijate nov set ikona, a meni ima nekoliko novih komandi. Uglavnom, posao je jednostavan. Klikom obeležite objekt, a zatim preko ikona ili kroz meni vršite odgovarajuće doterivanje. Dijalog-boksovi su impresivni, na višem nivou od onih u *MS Excel-u*. U njima se vidi reakcija na svaku promenu u podešavanjima nekog objekta. Posebno me je zainteresovala mogućnost da prepravljam gotove sličice. Načini na koji se to može uraditi gotovo da nemaju broja: promena boja i nijansi svih elemenata na slici, promena šrafure i punoće boje, debljina linija, dodavanje novih elemenata, itd... Vrlo lako i zabavno.

QPW je poboljšana u više pravaca u odnosu na proizvode za DOS. Recimo, ima dosta novih funkcija, tako da ih je sada ukupno 121. Izdeljene su u nekoliko grupa: matematičke, statističke, *database*, logičke, finansijske, za datume i vreme, i string funkcije. Nema sumnje da će ovoliki broj dobro osmišljenih funkcija omogućiti korisnicima da prave dobre i kompleksne tabele za najrazličitije poslove.

Što se tiče kompatibilnosti sa 1-2-3, izgleda da je sve u redu i da QPW prihvata sve makroe i funkcije iz 1-2-3 verzije 2.xx i 3.xx. To bi trebalo da bude veliko olakšanje za one kori-

snike programa 1-2-3 koji se, eventualno, reše da pređu na rad sa QPW. Prilikom ubacivanja makroa iz Lotusovih tabela ne treba zaboraviti da se mora aktivirati takozvani „1-2-3 menu tree“, a za upotrebu uobičajenih komandi u 1-2-3 i QP za DOS treba aktivirati **Key Rider**. Makroi koji se u 1-2-3 pokreću uz pomoć **Alt** tastera, ovde će se startovati pomoću **Ctrl** tastera.

U ovoj verziji, QPW ima zadivljujućih 150 makro komandi. Izgleda da su Borlandovi programeri mislili na sve, jer, prosto nema onoga posla koji se pomoću njih ne može uraditi. Ovo ocenjujem kao veliki napredak u *spreadsheet* programima. Kod pisanja makro procedura od velike pomoći je ugrađeni **Macro Debugger**, pomoću koga se makroi mogu kontrolisati na nekoliko načina: korak po korak; postavljanjem tačaka prekida, kada makro staje ako je zadovoljio određeni rezultat ili uslov; pokretanjem makroa punom brzinom do tačke prekida, a onda korak po korak. Rad sa *Debugger*om se odvija u posebnom prozoru – **Debug window**.

POTPUNA PRILAGODLJIVOST

Veliki kvalitet programa QPW je to što ga korisnik može sebi maksimalno prilagoditi. Recimo, ne morate koristiti standardne *Windows* meni komande, već to možete raditi iz nekog od alternativnih (1-2-3 ili QP) menija koje ćete pozivati sa „/“. „Meni-bar“ možete zameniti sopstvenim menijem, koji će sadržati vama potrebne komande. Za to treba iskoristiti mogućnosti koje se nude u alatima za izradu sopstvenih aplikacija. U ovom programu aplikacija je mešavina i kombinacija makroa, dijalog-boksova i menija. Za izradu aplikacija postoje alati, takozvani **Application Tools**. Pomoću njih možete doterati *Quattro Pro* tako da ispunjava specijalne zahteve korisnika; možete isključiti nepotrebne komande, čime interfejs postaje jednostavniji (setite se „Balera“ za 1-2-3), a postoji i mogućnost kreiranja interfejsa za nove, vaše, komande.

Taj posao, pravljenje aplikacija, predstavlja pravo zadovoljstvo. Napravićete sopstvene dijalog boksove, **SpeedBar-ove** i menije i veoma ih jednostavno povezati u celinu – aplikaciju. Sve se odvija u dijalog-prozoru do koga se dolazi komandama **Tools Builder**. Kada se nalazite u njemu, imate na raspolaganju poseban **SpeedBar** sa potrebnim alatima. Vaš dijalog-boks može sadržati razne oznake, polja za

unos podataka, takozvane radio tastere koji će aktivirati različita podešavanja, list-boks sa listom opcija ili bitmap tastere (obično **OK** i **Cancel**). Povezivanje dijaloga-boksa sa aplikacijom ide preko makro komande <**DIALOG**> sa odgovarajućim argumentima.

Za poslove koji se često izvode, korisno je napraviti sopstveni **SpeedBar** koji će sadržati skraćene za te poslove i tako ih znatno ubrzati. Razlika između **SpeedBar-a** i *dijalog-boksa* je u tome što se **SpeedBar** pojavljuje na vrhu *Quattro Pro* prozora i korisnik ga ne može ukloniti (osim ako to nije makrokom predviđeno) i čuva se u odvojenoj .BAR datoteci. Da bi ste ga prikazali u aplikaciji, morate prvo ukloniti originalni, a onda, postaviti sopstveni **SpeedBar**.

Što se menija tiče, možete praviti sopstvene menije pomoću čijih komandi ćete inicirati ikone, pokretati makroe, prikazivati dijalog-boksove ili pozivati druge menije. Naravno, sve će to biti u *Windows* stilu.

Uputstvo „Building Spreadsheet Applications“ je prilično solidno, pa mislim da korisnik ne bi trebalo da ima većih problema u pravljenju aplikacije.

Ovo je, po mojoj oceni, najbolji spreadsheet koji radi pod *Windows*-om. Ostavio je na mene bolji utisak od *MS Excel-a*, da i ne govorimo o programu 1-2-3 za *Windows*. QPW ima fantastičan interfejs, manipulacija objektima je izuzetna, kretanje po trodimenzionalnoj tabeli je veoma lako, a ovoliki broj i ovakve funkcije i makro komande, te sistem građenja aplikacija, garantuju da će korisnici imati velike mogućnosti u pravljenju sopstvenih aplikacija. Ako vam ne smetaju poprilični zahtevi u pogledu hardvera (min. 4Mb RAM-a i gotovo obavezni 386 procesor), *Quattro Pro for Windows* će sigurno zadovoljiti vaše potrebe, a ponavljam – mislim da je jedan od najboljih, ako ne i najbolji *Windows spreadsheet* na tržištu. No, videćemo da li će ovaj program, bez velike DOS baze korisnika, uspeti da se nametne kao standard.

Korisna adresa:

SOFTLAND
11000 Beograd, Skadarska 45
Tel: 011 / 343 043



Sve standardne konfiguracije PC računara

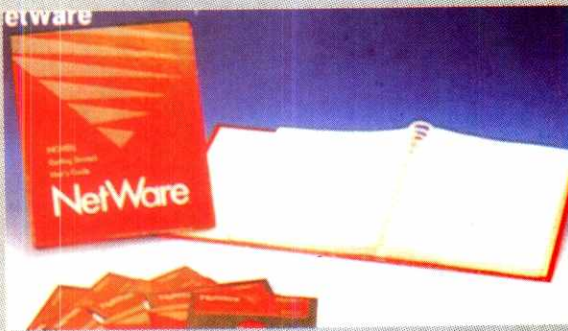
LapTop i notebook računari
Veridata



ALR
Advanced Logic Research, Inc.



Snaga i sigurnost koja Vam je potrebna



Računarske mreže pod operativnom sistemom **NOVELL**



UNIX, XENIX
Open Desktop

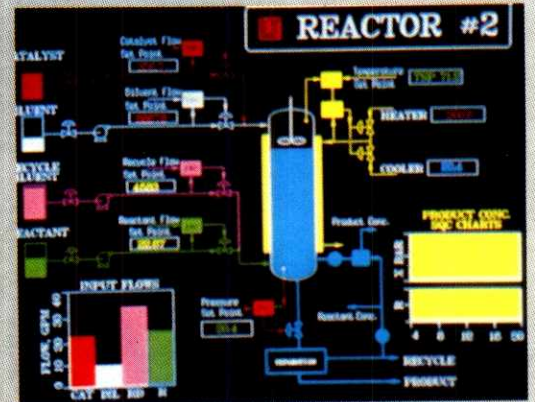
SCO
THE SANTA CRUZ OPERATION

Kompletno rešenje za automatizaciju Industrijskih procesa i laboratorija bazirano na PC računarima

PC-LabCard / Industrijski PC računari / PC-ToolCard



TEXAS MICRO



COMTEC

Industrial & Lab Automation with PCs
ADVANTECH

LABTECH

"COMTEC" d.o.o. Preduzeće za Informatički Inženjering i automatizaciju
Novi Sad, Vojvodanskih brigada 7/IX, tel:021/27-181, tel/fax:021/22-258



POČETAK I KRAJ JEDNE GODINE

INTERSOFT

29. novembra 43/l, tel: 323-539, 325-393, fax: 324-120

NAJBOLJI PIŠČEV PIJATELJ

„Lotus Development Corporation” je velika firma. Njihov spređit *Lotus 1-2-3* je jedan od najuspešnijih programa ikada napisan za PC, *Freelance Graphics* jedan od najmoćnijih paketa za poslovnu grafiku, dok je *CC:Mail* vodeći svetski LAN sistem za elektronsku poštu. Ovom impozantnom skupu odnedavno se pridružio i *Ami Pro*, verzija 3.0, koji pretenduje da postane tekst procesor broj 1 za *Windows*.

Virtual Library of Faculty of Mathematics - University of Belgrade

Sa osobitim zadovoljstvom prilazim pisanju ovog prikaza – razlog bih mogao da potražim, pre svega, u višegodišnjem traganju (za izgubljenim vremenom?) za idealnim tekst procesorom. Ukratko, mislim da sam pronašao Sveti Gral tekst procesora. I koliko god da sam subjektivan, bez namere da povredim bilo koga – „vordovce”, „vordperfektovce”, ili korisnike „Chi Writer”-a i drugih tekst procesora – trudiću se da svaku svoju tvrdnju stavim na, kako kaže Kiš, „probni kamen činjenica”.

Konkurencija na polju tekst procesora je u svetu veoma oštra, rekao bih čak – nemilosrdna. Svako zaostajanje, bilo u ergonomiji, bilo u performansama, skupo se plaća. Na takvom tržištu, među divovima kao što su „WordPerfect” i „Microsoft”, svoje mesto je odlučio da izbori i Lotus – sa pravom. Čitajući strane časopise i razgovarajući sa ljudima (sa univerzitetu Iowa, UCLA, New York), saznao sam da svi prelaze na *Ami Pro*. U ovom prikazu, pokušaću da odgovorim na pitanje: Zašto?

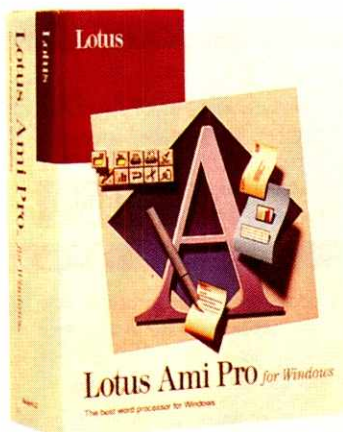
TRIJUMF ERGONOMIJE

Ami Pro 3.0 je urađen savršeno – potpuni trijumf ergonomije, performansi (brzina rada, modularnost, „razumevanje” sa velikim brojem programa drugih proizvođača (ne samo tekst procesora!), rad sa grafikom, rad u mreži...) i cene.

Ami Pro 3.0 traži barem 286 procesor i 2 MB RAM memorije. Moraćete da prežalite cca 15 MB na hard disku (u punoj konfiguraciji). Inače, moguće je instalirati *Ami Pro* i u minimalnoj konfiguraciji, na približno 5 MB. Računajte na dodatnih 3 MB na hard disku za „temporary” fajlove, plus prostor za vaše radne datoteke. Nemojte da vas ovo uplaši – *Ami Pro* je vredan toga! Da odmah budemo i realni – prijatan rad zahteva barem 386 mašinu i 4 MB RAM memorije.

Ami Pro verzija 3.0 se isporučuje u 3.5”/

Ante Ćurlin



GENERALNA ANALIZA

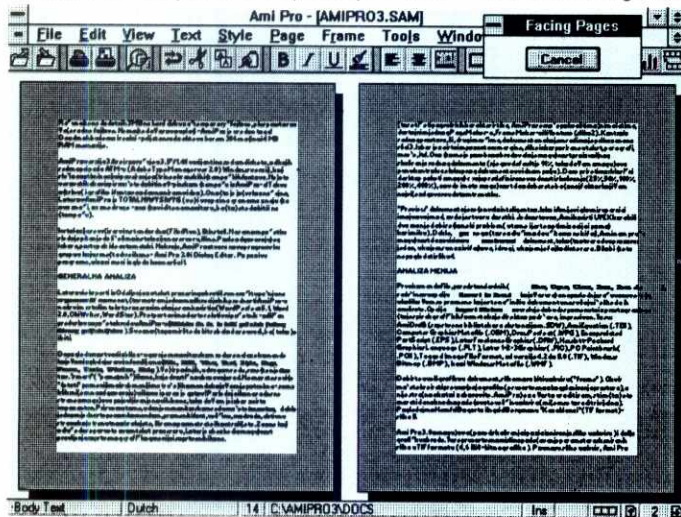
Lotusovi eksperti iz Odeljenja za tekst procesore obratili su naročitu pažnju na ergonomski momenat, što smatram jednom od krucijalnih prednosti *Ami Pro*-a nad svim ostalim tekst procesorima koje sam koristio (*Word Perfect 5.1*, *Word 2.0*, *Chi Writer*, *Word Star*). Pretpostavimo da ste relativni početnik – odličan preduslov za početak rada u *Ami Pro*-u (mislim da će to biti početak jednog divnog prijateljstva). Sve ono što pomislite da bi trebalo da se uradi, baš tako će i biti.

Da pogledamo stvari izbliza – u gornjem monitorском redu se nalaze komande koje deluju sasvim poznato (**File, Edit, View, Text, Style, Page, Frame, Tools, Window, Help**). Već ispod njih, u drugom redu, smešten je deo tzv. „smart” („pametnih”) ikona, koje drastično ubrzavaju rad. Ne morate se više „šetati” po menijima i podmenijima tražeći komandu koja vam je potrebna – dovoljan je samo klik mišem na odgovarajuću ikonu i posao je gotov! Poslednja ikona sa desne strane omogućava poziv novih ikona, tako da je izbor zaista impozantan. Pri svemu tome, u donjem monitorском redu možete trenutno, dakle jednom jednostepenom komandom, promeniti font, veličinu, režim rada, aktivnu stranu koju trenutno obrađujete, ili samo posmatrate i kontrolišete. Za one koji odluče da se posvete ovom tekst procesoru, Lotus je obezbedio mogućnost pravljenja sopstvenog grafičkog menija i sopstvenih ikona.

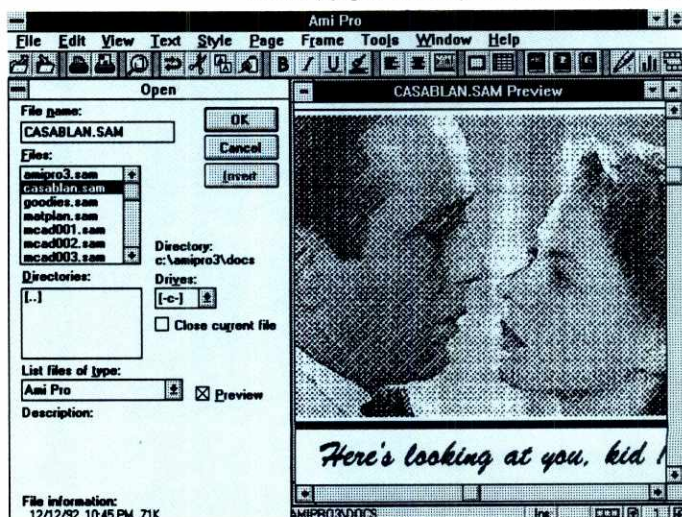
Što se tiče tipografskih karakteristika, *Ami Pro* se može pohvaliti moćnim alatima, dostojnim jednog *PageMaker*-a, *FrameMaker*-a ili *Venture*. Za kontrolu radnog prostora, ili, drugim rečima, dokumenta na kojem radimo stoji nam zaista impozantne mogućnosti: margine, slike inkorporirane u tekst, paragrafi, mreže, itd. Ono što me je posebno obradovalo je mogućnost proizvodnjnog skaliranja radnog dokumenta (njegov *default* je 91%, tako da omogu-

1.44 varijanti na sedam disketa, od kojih sedam pripada ATM (*Adobe Type Manager ver 2.0*) *Windows* verziji, koji služi za optimizaciju i povećanje oštine ekranskih i štampačkih fontova. Nećete verovati kakav ispis možete dobiti na 9-pinskom štampaču iz *Ami Pro*-a! Takvu oštrinu (i grafike i fontova) odavno nisam video. Ono što je vrlo značajno, Lotusov *Ami Pro* je TOTALNI WYSIWYG (već i vrapci na granama znaju šta ovo znači, za one druge – ono što vidite na monitoru, baš to ćete dobiti i na štampaču).

Instalacija se vrši sasvim standardno (**File/Run**), B:install. Na samom početku sleduje pitanje da li želimo instalaciju na serveru ili ne. Posle odgovarajućeg izbora, postupak ide automatski. Na kraju, *Ami Pro* otvara novu programsku grupu u koju smešta dve ikone – *Ami Pro 3.0* i *Dialog Editor*.



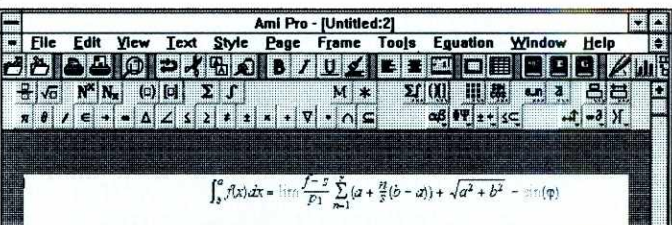
Rad sa tekstem (Facing Pages)



Preview mod kod otvaranja dokumenta



Import grafike u Ami Pro 3.0



Meni matematičkog editora

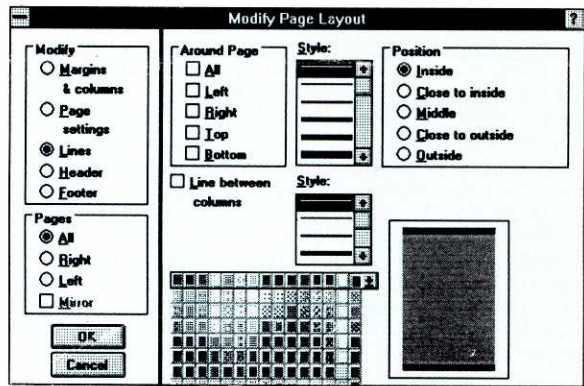
čava punu kontrolu celokupnog dokumenta u vidnom polju). Da se priseto: klasični desktop paketi omogućavaju predefinisane vrednosti skaliranja (25%, 50%, 100%, 200%, 400%), a ovde imate mogućnost da odaberete baš onaj faktor koji vam u datom trenutku najviše odgovara.

Rad sa paragrafima je doveden do savršenstva. Paragraf čini bilo kakav tekst, slovo, ili razmak <Space>, ali između dva pritiska na taster <Enter>. Sa paragrafom u Ami Pro-u možete činiti sledeće stvari: a) menjati font, boje i attribute teksta (podebljan (Bold) i zakošen (Italic)), b) poravnati tekst nalevo, nadesno, centrirati ga, kao i poravnati sa obe strane (justified), c) postavljati tabulatore, d) uvlačiti tekst, e) određivati razmak između redova, f) prelom strana (pre, posle ili u okviru samog paragrafa), g) formatirati tekst kao bilten, h) razdvajati paragrafe linijama čiji se atributi takođe mogu menjati, i) obrađivati aritmetički podatke unutar tabela (4 osnovne operacije +, -, *, /, i %), j) deliti reči na kraju reda, itd. Moguće je raditi i sa dve strane na jednom dokumentu (facing pages). Ukratko, rad sa tekstom je olakšan do krajnjih granica, i u ogromnom broju slučajeva korisnike lišava potrebe da prelaju tekst u nekom desktop paketu (još i njega treba naučiti!). Sve se može uraditi u Ami Pro-u.

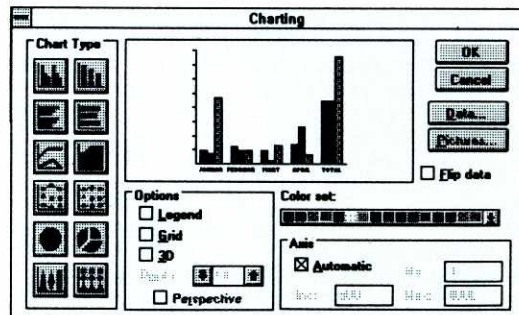
„Preview“ dokumenta je rešen vrlo inteligentno. Iako i Amijevi glavni suparnici imaju ovaj mod, Ami koristi UVEK korak ili dva manje da bi rešio neki problem (u tome i jeste optimizacija i pomoć korisniku). Dakle, pre nego što se odlučimo da učitamo neki fajl, Ami nam pruža mogućnost da zavirimo u neotvoreni dokument, tako što otvara dva prozora: jedan, u kojem su nazivi fajlova, i drugi, u kojem je fajl koji nas zanima.

ŠETNJA PO MENIJIMA

Prva komanda File, pored standardnih (New, Open, Close, Save, Save As), sadrži



Podešavanje izgleda strane



Prozor za izbor tipa grafikona

novu opciju **Revert to Saved** koja vas vraća na poslednju sačuvanu verziju, ukoliko vam se promene koje ste načinili u dokumentu ne sviđaju i želite da ih anulirate. Opcija **Import Picture** zavređuje da bude spomenuta iz prostog razloga što je spisak grafičkih formata koje direktno podržava impresivan. To su: *AmiDraw* (sopstvena biblioteka sa ekstenzijom .SDW), *AmiEquation* (.TEX), *Computer Graphics Metafile* (.CGM), *DrawPerfect* (.WPG), *Encapsulated PostScript* (.EPS), *Lotus Freelance Graphics* (.DRW), *Hewlett-Packard Graphics Language* (.PLT), *Lotus 1-2-3 Graphics* (.PIC), *PC Paintbrush* (.PCX), *Tagged Image File Format*, od verzije 4.2 do 5.0 (.TIF), *Windows Bitmap* (.BMP), kao i *Windows Metafile* (.WMF).

Da biste uveli grafiku u dokument, slika mora biti u okviru (frame). Okvir možete kreirati pre unošenja grafike (prvenstveno zbog planiranja prostora), ali nije strašno ako to i zaboravite. *Ami Pro* će to uraditi sam, s tim što ćete morati da naknadno podešavate veličinu okvira (mišem se to radi kao od šale). Pogledajmo Hemfri Bogarta i Ingrid Bergman u „Kazablanci“ (TA format).

Ami Pro 3.0 omogućava (pored skaliranja i pozicioniranja slike u okviru) i dalju grafičku obradu. Tu se prvenstveno misli na podešavanje parametara skeniranih slika u TIF formatu (4, 6 ili 8-bitna grafika). Po unosu slike u okvir, *Ami Pro* automatski podešava osvetljenje i kontrast da bi dobio najbolje rezultate. Za ručno podešavanje, potrebno je aktivirati opciju **Tools/Image Processing**, i zatim podešavati osvetljenje (*Brightness*), kontrast (*Contrast*), kontrast između same slike i ivica unutar slike (*Edge Enhancement*), kao i „mekoću“ slike (*Smoothing*).

Opcija **Doc info** je standardna za „jače“ tekst procesore i nećemo se na njoj zadržavati. Vredan pomena je trud (ideja, prvenstveno!) Lotusove ekipe da ugradi u ovakav paket nešto što bi se moglo nazvati *PC Tools* u malom. Naime, opcija **File Management**, bar kako je meni „zazvučala“, treba da prekine i poslednju sponu sa dozlazboga dosadnim DOS-om (ili

Windows File Managerom), za ljude koji se bave obradom teksta a ne žele uopšte da „izlaze“ iz svog tekst procesora. Zaista, kad sam problem počeo da posmatram iz ovog ugla, shvatio sam da je korisnicima dovoljan „mini DOS servis“ (**Copy, Move, Rename, Delete, Attributes**), kao i opcija **View**, da bi zadovoljili sve svoje potrebe.

Master Document omogućava pravljenje sadržaja, indeksa, kontrolu numeracije strana, obeležavanja i svih stvari dostojnih najkvalitetnijeg tekst procesora i desktop paketa.

Opcija **Merge** omogućava automatsko povezivanje podataka iz fajla sa, na primer, adresama, imenima i prezimenima ljudi – ovo je specijalno interesantno kod slanja cirkularnih pisama, što *Ami Pro* (jednostavno i elegantno, a kako bi drukčije?) radi automatski. Svega su tri koraka u pitanju: kreirajte ili preuzmite već gotov fajl sa imenima i adresama (vaših poslovnih partnera?), otkucajte cirkularno pismo i izdajte komandu **Merge** i **Print**. To je sve!

Opcija **Print Envelope** omogućava da štampate adrese na kovertama. Ako se u bilo kom momentu „izgubite“ (pretpostavimo da je sad pravi trenutak za to), iskoristite *Ami*-jevu pomoć u kontekstu. Istovremeno pritisnite taster <Shift> i funkcijski taster <F1>. Strelica će se pretvoriti u neobičan znak pitanja. Jednostavno kliknite na odgovarajuću komandu iz menija koja vas interesuje, i dobićete detaljnu pomoć, vezanu baš za tu komandu.

Ami se pobrinuo da olakša i posao oko kreiranja dokumenata. Čega sve tu nema – preko 50 unapred definisanih struktura: od poslovnih pisama, cirkulara, agendi, raznoraznih kalendara (sa danima, nedeljama, mesecima), faks dokumenata sa logotipovima, profesionalnih rešenja za desktop publikacije, raznih labela, plakata, EPP struktura, naučnih disertacija, indeksa, memoranduma, molbi, žalbi, pisama, novinskih stubaca, zaglavilja, fus-nota, raznih izveštaja, itd. Ako želite da otkucate neki dopis, a ne znate formu, odaberite iz *Ami*-jevog menija ono što vam treba – ostatak je najobičnija rutina.

KONTROLA TEKSTA

Ovo je vrlo moćan deo Lotusovog tekst procesora. Sa tekstem se mogu čuda činiti, ali sve to ne zahteva veliki trud da bi se naučilo. Kako se menjaju osnovni atributi teksta? Prvo se obeleži željeni deo dokumenta (najjednostavnije i najbrže mišem – kao i kod svakog *Windows* orijentisanog tekst procesora), zatim se otvara opcija **Text/Font** i – gotovo. Možete obojiti tekst, možete odabrati proizvoljnu veličinu – uz standardne mogućnosti podebljavanja, podvlačenja (čak i dvostrukom linijom!), ili pretvaranje teksta u italik.

Treba naglasiti da *Ami Pro* verzija 3.0 koristi sve *Windows True Type* fontove. Da ponovimo još jednom, kontrola veličine i tipa fonta se mnogo brže može obavljati sa donje monitorske linije *Ami*-ja – u jednom koraku.

Kontrola izgleda cele strane se vrši preko opcije **Page/Modify Page Layout**. Tabela parametara je detaljna i jasna – možete raditi sa tekstem bukvalno sve što hoćete.

Ne treba zaboraviti još jednu dobru ideju realizovanu kroz opciju **Text/Fast Format**. Pretpostavimo da imate jedan dokument na koji ste utrošili silno vreme: podesili ste fontove, veličine, paragafe, boje, stilove, sve što je moglo da se uradi, uradili ste. E, sad nastaju problemi – imate još nekoliko dokumenata (neobrađenih, recimo u čistom ASCII formatu) i želeli biste da izgledaju isto kao i onaj prvi. Ništa lakše – izaberite opciju **Fast Format** i sledite uputstva (iz knjige, ili iz **Helpa**, potpuno je svesjedno), i doći ćete do željenog cilja. Da sam pre nekoliko godina imao ovakvu alatku, ne bih se mučio sa tipskim kucanjem tekstova.

PROFESIONALNI ALATI

Opcija **Tools** krije u sebi glavne alate koji su potrebni specifičnim korisnicima. Uz već dobro poznati **Spelling Checker**, tu je i **Grammar Checker**. On će vam ponuditi statistiku i ukazati na greške koje ste počinili u gramatičkom domenu naravno, engleskog jezika), dok će, sa druge strane, **Spelling Checker** da „odradi“ svoje. Za one koji prevode na engleski jezik – prava stvar.

Rad sa tabelama predstavlja pravo uživanje – za razliku od pojedinih tekst procesora koji ovakav posao pretvaraju u noćnu moru. Jedan klik mišem na **smart** ikonu, i *Ami* nudi **default** tabelu sa 5 redova i 5 kolona – podesite mišem koliko vam treba i – gotovo. Rad sa tabelama uključuje elementarne aritmetičke operacije – možete sabirati sadržaje ćelija, oduzimati ih, množiti i deliti, kao i nalaziti procenat (%). Po otvaranju tabele, u meniju se pojavljuje nova komanda **Table**, koja omogućava veliki broj operacija sa tabelom i promenu svih atributa. To je toliko veliki deo *Ami Pro*-a, da sigurno zaslužuje posebnu analizu.

CRITANJE I GRAFIKONI

U *Ami Pro*-u možete nacrtati sliku (naravno, ako imate miša) koja sadrži i komplikovane operacije (i elemente) od već legendarnog *Paintbrush*-a koji pripada *Windows*-ima. Dakle, linije, lukovi, krugovi, elipse, pravougaonici i kvadrati, sve to „garnirano“ sa puno unapred definisanih tekstura, boja, oblika, itd. Lepa je stvar što se mogu editovati Lotusovi PIC fajlovi (ako su već kreirani grafikoni, zašto ih ne iskoristiti?), *Lotus Freelance* fajlovi, kao i *Draw Perfect* fajlovi – zašto baš njih spominjemo? Jednostavno zato što ih *Ami Pro* automatski konvertuje u svoj .SDW format, tako da ih mo-

žete menjati do mile volje. Po izboru opcije **Tools/Drawing**, pojavljuju se nove **smart** ikone.

Pored standardnih opcija, *Ami* uvodi i neke specifične, koje već pripadaju grafičkim paketima koji su samo za to namenjeni. Na primer, ukoliko crtate ravnu liniju, a želite da to bude pod uglom od 45 stepeni, prvo pritisnite <Shift>, a onda pokrećite miša. Zatim, tu je interesantna mogućnost rotacije teksta za proizvoljan broj stepeni (pri tome se originalni font zadržava!), zatim grupisanje i rotacija definisanih objekata, puna kontrola boja, mreža, itd. Ukratko, dobili ste moćan crtački alat pomoću kojeg možete nacrtati ono što vam treba, preuzeti već gotovu grafiku iz celokupne game grafičkih formata, ili doraditi već kreiranu grafiku. Ako je i od jednog tekst procesora, više je nego dovoljno!

Ono što je pre nekoliko godina bila privilegija Lotusa (paketa 1-2-3 i njemu sličnih), sada je samo ugrađeni deo tekst procesora. Ala se vremena menjaju! Dakle, podržani su svi tipovi grafikona: trakasti, kumulativni, linijski, površinski, kombinovani linijski sa slikom, grafikon u obliku pite, i mnogobrojne kombinacije čije bi prevođenje moglo da izazove samo smeh kod čitalaca – nazovimo ih kombinovani grafikoni. Ima tu svega i svačega – od postavljanja mreža, komentara, fusnota, legendi, do određivanja perspektive. Da se ne bismo previše udaljivali od tekst procesora (jer, ipak, o njemu je reč), i da se ne bih osećao kao da pišem o Lotusu ili Excelu, preći ćemo na drugu temu.

MATEMATIČKI EDITOR

Kao korisnik matematičkih paketa, osećam se pozvanim (i prozvanim) da analiziram malo pobliže *Ami*-jev matematički editor. Da ne bi bilo zabave, proverite prvo da vam slučajno neko nije obrisao *Windows* kompatibilni *Symbol font* (za prikazivanje i štampu jednačina). Ako je to u redu, otvorimo meni **Tools/Equations**. U ovom modu možemo ubacivati „template“ u jednačine, kao i sve poznate matematičke simbole, funkcije, itd. Pošto moje kolege uglavnom rade u *TeX*-u, obradovače ih da *Ami Pro* prihvata i *TeX* komande! Naravno, moguće je sačuvati jednačinu ili neki njen deo u *TeX* formatu, kao i importovati *TeX* fajl. S obzirom da je *TeX* prihvaćen kao univerzitetски standard za matematički slog, evo još jedne važne sponne između (ipak isuviše komplikovanog *TeX*-a) i jednog krajnje „user-friendly“ orijentisanog paketa.

Test koji sam mogao da izvedem za relativno kratko vreme je test zvani „ruska matematička zbirka“. Priznajem da sam izgubio bitku – nisam pronašao matematički izraz ili simbol koji *Ami* ne bi mogao da prikaže (naravno, to ne znači da ih nema). Fasciniran sam brojem opcija, bibliotekom matematičkih simbola (operatora, matrica, integrala, suma, proizvoda, itd.), ali perfekcija ide dotle da možete da birate da li hoćete indeks levo ili desno, gore ili dole, podebljan ili nepodebljan simbol, u ovoj ili onoj boji, i tome slično – da se čovek jednostavno zapita: hoće li mi to ikada zatrebati? S obzirom na ranije patnje zbog skromnih mogućnosti tekst procesora (a bogami, i krajnje mračnog interfejsa *Venture 2.0* za pisanje jednačina), ovo mu dođe kao melem na matematičku ranu.

Mada sam oduševljen *Ami*-jevim matematičkim editorom, moram da priznam da mi se učinio pomalo sporim na mojoj 386/20 mašini (ili je, ipak, *Windows* okruženje u pitanju?). Možda bi 486/66 sa lokal basom malčice pomoglo stvar?

RAZMENA PODATAKA

Zbog velikog ličnog iskustva, ovom aspektu pridajem posebnu pažnju – vrednost paketa procenjujem (naravno, ne bezuslovno!) i kroz komunikaciju sa drugim (standardizovanim) paketima. Iako cenim „Microsoft“, moram da priznam da sam razočaran siromašnim *Word*-ovim filterima za uvoz i izvoz fajlova, kao i njegovom brzinom u odnosu na *Ami Pro*. Da pogledamo listu ulaznih filtera za *Ami* (ili, drugim rečima, koje pakete *Ami* direktno uvozi – zvezdicom (*) ćemo označiti i u kojem formatu je *Ami* sposoban da ih eksportuje): *Lotus 1-2-3* (verzije 1, 1A, 2.0 i 2.01, .WKS i .WK1 ekstenzije), *Lotus 1-2-3* (verzije 3.0, 3.1 i *Windows* verzija, ekstenzije .WK3), *Advance Write**, *Ami Pro** (ranije verzije), ASCII*, dBASEIII, dBASEIII+, dBASEIV, DCA/FFT* (Final Form Text), DCA/RFT* (Revisable Format Text), DIF, *DisplayWrite** 4 i 5, *E-Mail**, *Enable Versions** 1-5 do 2.5, *Executive Memo Maker**, *Lotus Manuscript 2.0* i 2.1, MS EXCEL 3.0 i sve ranije verzije, *MS Word** 4.0, 5.0, 5.1 i 5.5, *MultiMate* verzija* 3.3, *MultiMate Advantage II**, *Navy DIF**, *Paradox* sve verzije do 3.5, *Peach Text** 2.11 i ranije verzije, *Rich Text Format**, *Samna Word**, *SmartWare* verzija 1, *SuperCalc* verzije 3 i 4, *Symphony* verzije 1.0, 1.01 i 1.1 (ekstenzije .WRK i .WK1), *Windows Write**, *Word za Windows**, *WordPerfect** verzije 4.1, 4.2, 5.0 i 5.1, *WordStar 2000** verzije 1 i 3. Neverovatno spisak, koji garantuje sigurnost u radu – više uopšte nije važno ko radi u kojem tekst procesoru (ili aplikativnom paketu). Preuzmete rad od korisnika u njegovom formatu, obradite ga u *Ami Pro*-u, i eksportujete ponovo u njegov format! Kako vam se to sviđa?

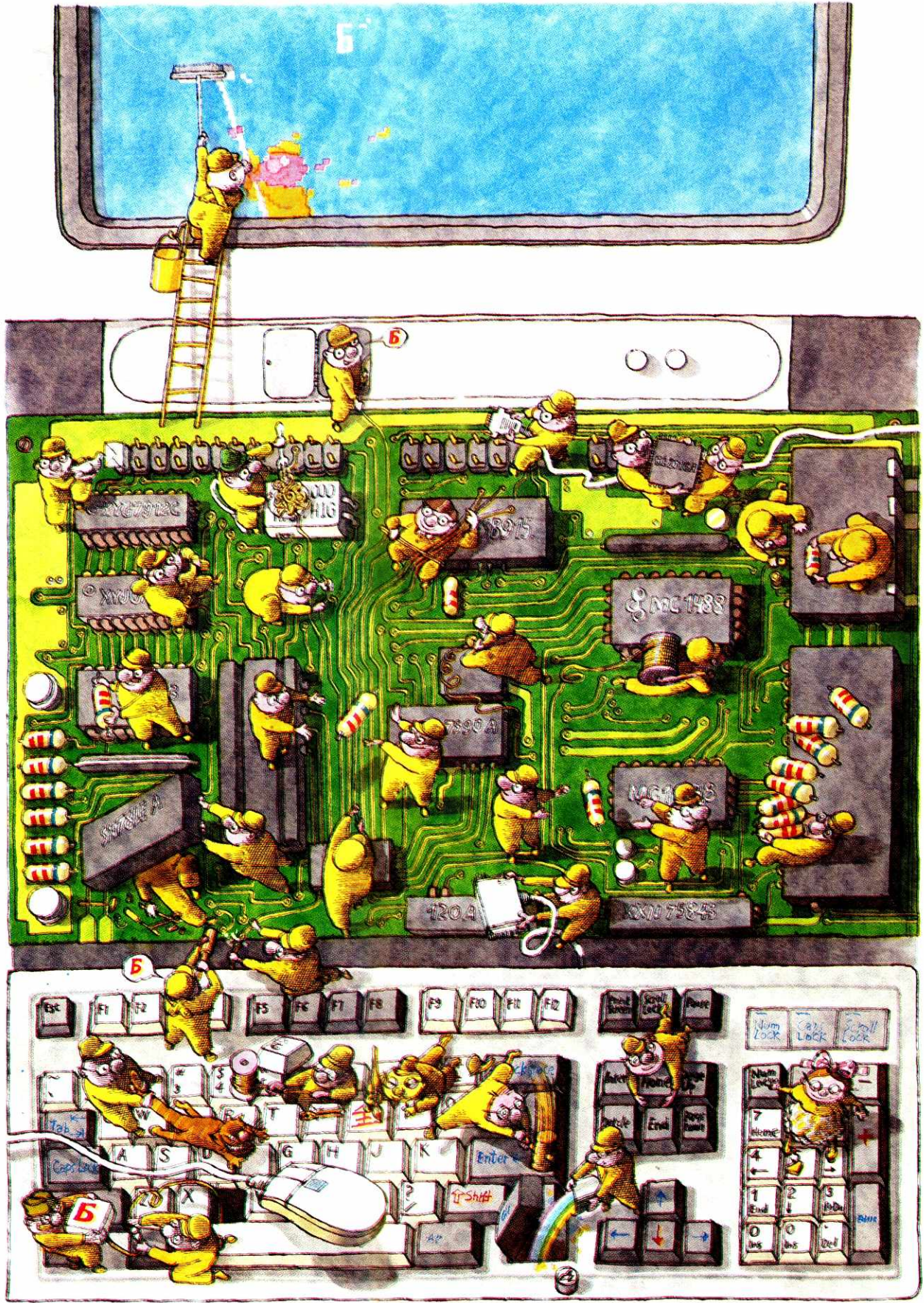
AL JE LEP OVAJ SVET

Makroi su oduvek bili posebna priča – oni dolaze kao kruna vašeg rada, dakle, podrazumeva se određen broj sati (nedelja, meseci?) rada u programu. *Ami*-jevi makroi će i ovog puta biti samo spomenuti, sa nadom da će u nekom budućem prikazu dobiti dostojno mesto. Inače, automatizacija poslova nezadrživo napreduje i samo je pitanje vremena kada će i makroi izboriti zasluženo mesto pod suncem.

Ami Pro verzija 3.0 poseduje ogromne mogućnosti. One, pre svega, leže u *Ami Pro Macro* jeziku koji omogućava, na relativno jednostavan način, zapis makro sekvenci (operacija koje želite da se ponavljaju u strogo utvrđenom redosledu). Mislim da se automatizacija posla ili grupe poslova kod *Ami Pro*-a približava apsolutu.

Spomenućemo još jedan segment *Ami Pro*-a koji će sigurno imati svoj poseban osvrt u nekom od narednih brojeva „Računara“, a to je rad u mreži. Po informacijama koje stižu, korisnici su prezadovoljni, a i naša *Ami Pro* „kasta“ je ubeđena da će *Ami Pro 3.0* u nezadrživom jurišu osvojiti ogromno tržište.

Osećam se kao na početku. *Ami Pro 3.0* ima toliko mogućnosti, da stalno otkrivam nove i nove – granica je vrlo daleko. Namenjen je ljudima velike kreativnosti, do maksimuma optimizovan, oslobođen suvišnih poteza, odličan u performansama. Lepo je videti na knjizi natpis „Lotus Development Corporation, Word Processing Division“. Lotus je dugo čekao da se uključi u trku za deo kolača na tržištu tekst procesora. Ali, izgleda da sad imaju Karla Luisa – čeka se odgovor „Microsoft“, „WordPerfect“ i ostalih. Što se mene tiče, svaki put kad sednem za računar i počnem da radim u *Ami Pro*-u, osetim neki unutrašnji glas koji mi govori: „Play it, again, Sam!“



BEST
computers
011/320-103,
Beograd, Majke Jevrosime 42;

ADACOM

personalni računari, računarski inženjering, hardver,
softver, instalacije, održavanje, proizvodnja i razvoj

386 SX - 25 MHz 1550

Desktop kućište

SVGA mono

2 MB RAM

Hard disk 85 MB

DEM

1990 Hard disk 85 MB

386 DX - 40 MHz

Desktop kućište

SVGA mono

4 MB RAM

Hard disk 85 MB

Virtual Library of Faculty of Mathematics - University of Belgrade

elibrary.matf.bg.ac.rs



ADACOM



ADACOM, Čika Ijubina 12, Beograd tel./fax: 3811/629-233

IMTEL

INSTITUT ZA MIKROTALASNU TEHNIKU I ELEKTRONIKU

CAD STATION

◆
EISA 486-50/66, 256kB CACHE, 8-32MB
SCSI DISK 200MB-1500MB

◆
TIGA VIDEO KARTICA (1400X1200)
TMS 34020 TRUFCOLOR
MULTISYNC MONITOR 17"-20"

◆
EISA ETHERNET KONTROLER, MASTER
EISA SCSI-2 KONTROLER SA 1-16MB CACHE
◆

TEL 135 420 FAX 138 928

USLUGA ZA USLUGU

Windows for Workgroups (WFW) novi je proizvod „Microsoft”-a u oblasti Windows 3.1 sistema. Proizvod obezbeđuje Windows 3.1 integralne sposobnosti za peer-to-peer umrežavanje, elektronsku poštu i funkcije „rasporeda časova” za radne grupe. Namenjen je grupama sa najviše 20 korisnika koji imaju potrebu za deljenjem informacija i resursa. Za grupe veće od 20 korisnika, „Microsoft” preporučuje mnogo formalniji pristup, kao što je Lan Manager, proizvod njihove kuće. Po mišljenju predstavnika „Microsoft”-a, malim grupama nisu potrebne client-server usluge.

Pored omogućavanja peer-to-peer umrežavanja, WFW se izvrsno uklapa u „Microsoft”-ovu globalnu Windows NT strategiju. Klijentske mašine vezane na Windows NT server biće ili LAN Manager klijenti, ili će imati Windows for Workgroups. Ono što „Microsoft” ne kaže eksplicitno jeste da je WFW rešenje jeftinije i jednostavnije, naročito za manje zahtevne korisnike. „Microsoft” namerava da ograniči broj korisnika koji se mogu priključiti na NT server; prebrojni korisnici mogu se sa ostatkom grupe povezati preko Windows for Workgroups.

WFW se pojavio jako brzo na tržištu – oktobarski brojevi evropskih računarskih časopisa tek su najavljivali njegovo predstavljanje na „Microsoft”-ovim prezentacijama. „Računari” su ga dobili na testiranje početkom decembra, zahvaljujući firmi „PC Club”, direktno iz SAD. Sve u svemu, od najave do pojave paketa proteklo je jedva nešto više od mesec dana. U prodaji je kompletan paket, kao i Add-on verzija, koja zahteva da je na računaru ranije već instaliran standardni Windows 3.1. Verzije paketa se razlikuju i po broju korisnika koji podržavaju. Interesantna je verzija WFW Starter Kit, koja sadrži i komplet hardvera za povezivanje dva računara, dve mrežne kartice, kablove i konektore.

PROLAZNE GLAVOBOLJE

U kutiji ćete naći osam 3.5" disketa, priručnike „Getting Started”, „User's Guide”, „Schedule+ User's Guide” i „Mail User's Guide” i uobičajene pamflete za registraciju proizvođača, zamenu 3.5" disketa od 1.44MB za diskete kapaciteta 720kB, kao i papir sa odgovorima na uobičajena pitanja korisnika.

Zahtevi su slični kao i ranije: MS-DOS 3.3 ili noviji (5.0 se preporučuje), za 386Enhanced način rada potreban je 386SX procesor ili viši, sa 3MB ili više memorije, dok se 4MB preporučuje. Deljenje resursa moguće je samo u 386Enhanced modu. Za one kojima deljenje resursa nije potrebno, biće dovoljan i procesor 286 sa 2MB memorije. Slobodnog prostora na disku mora biti bar 9.5MB, iako se preporučuje bar 14.5MB. Ako je na računaru koji ima 386SX ili viši procesor Windows 3.1 već instaliran, treba imati bar 3MB memorije i najmanje 3.5MB slobodnog prostora na disku (preporučuje se 8.5MB). Naravno, da bi računar radio u mreži mora imati i mrežnu karticu, a mrežni kablovi moraju biti postavljeni kako to zahteva izabrani protokol i topologija mreže.

Instalaciju sam obavio na dve mašine: na prvoj je Windows 3.1 već bio instaliran, uz razne dodatke: Norton Desktop for Windows, WinSpeed drajver za SVGA kartu, QEMM i drugi. Drugi računar bio je prazan – samo MS DOS 5.0. U oba računara bile su instalirane Ethernet mrežne kartice, kompatibilne sa poznatom „Novell” NE-2000 kartom. Ni sa jednim računarom nije bilo većih poteškoća pri instalaciji. Prva komercijalna verzija Windows 3.1 nije se baš lepo slagala sa NDW 2.0 paketom: nakon

Zoran Kehler

instalacije ostavljala je Program Manager, a ne NDW kao shell, a bilo je i slučajeva da instalacije uopšte ne prođe. Sa ovom verzijom Windows-a takvih problema nije bilo. Ono što, izgleda, zbunjuje Windows pri instalaciji jeste veliki broj instaliranih fontova: 155 fontova na mojem računaru Setup je zaboravio, uz izjavu da se nalaze u WIN.INI datoteci zapisani pod [w34sfddh4] sekcijom, i da ih moram sam kasnije dodati u [Fonts] sekciju; [w34sfddh4] sekcija sadržavala je, međutim, samo jednu praznu liniju.

Drugi problem javio se u vezi sa drajverom za SVGA kartu. Moja kartica sa ET-4000 čipom pogoni se preko WinSpeed drajvera. To se nije dopalo Windows-u for Workgroups – Setup je ili odbijao da startuje Windows, ili je instalacija prolazila, ali nakon toga Windows nije htelo da se pokrene. Lek za ovo bilo je izabrati standardnu VGA rezoluciju u toku instalacije, koja je kasnije zamenjena rezolucijom 1024x768, uz WinSpeed drajvera.

Iako je „Microsoft” izbacio na tržište Eastern European verzija Windows 3.1 samo nešto malo ranije, Windows for Workgroups ne poznaje opcije vezane za ove delove sveta. Jednostavno će preko već instaliranih drajvera za zemlju Jugoslaviju i srpsku tastaturu instalirati nove drajvere zapadnog sveta. Interesantno da i dalje postoji program Add 852 to DOS, koja treba da usaglasi kodnu stranu izabranu u DOS-u i kodnu stranu koju će Windows koristiti za uprozorene DOS aplikacije. Nejasno je zašto je u WFW zadržan ovaj program, koji je uveden u Windows for Eastern Europe.

Setup program predlaže dva načina instaliranja: Express i Custom. Express opcija sama identifikuje hardver i softver vašeg računara, konfigurise WFW, ostvaruje vezu vašeg računara sa mrežom i ažurira konfiguracione datoteke. Sve što morate učiniti jeste da ukucate svoje ime i odgovorite na par jednostavnih pitanja. Ako slučajno nemate dovoljno slobodnog prostora na disku, Express Setup će predložiti koje opcione komponente možete izostaviti – na primer, module Screen Saver-a, bitne mape za pozadinu radne površine, neke od Accessories programa. Iako Express Setup postavlja vrlo malo pitanja, to ne znači da ga svaki ko-

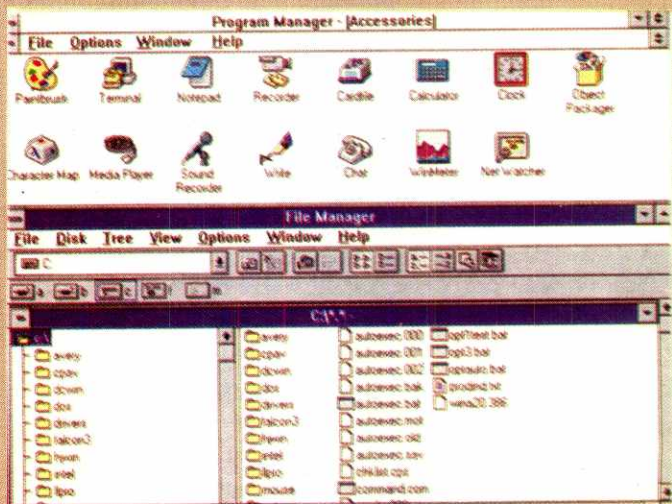
risnik može koristiti bez ikakvog prethodnog znanja o računaru i mreži koju instalira: treba znati ime grupe, portove na koje se priključuju štampači, tip mrežne kartice i njene parametre i eventualne sekundarne mreže na koje će računar biti vezivan.

Zbog lošeg iskustva sa Express opcijom, moram priznati da je nikada ne koristim na svom računaru; naročito je ne bih koristio kada se testirani softver mora po prvi put instalirati. Custom Setup je, dakle, opcija koju sam izabrao i za WFW. Ona je bezbednija i ako se radi o unapređenju već postojeće instalacije Windows 3.1 – dešavalo mi se da Express Setup jednostavno zaboravi sve što je ranije instalirano, uključujući tu i sve programske grupe i drajvere za hardver. Custom Setup omogućava da se preciznije, a svakako po važio sopstvenoj želji, podeše parametri instalacije.

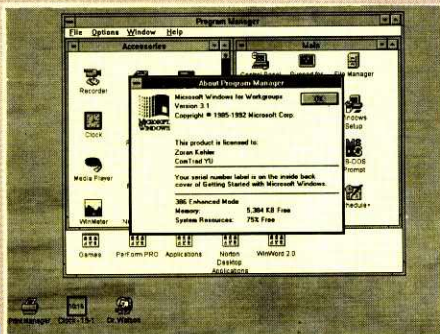
U toku instalacije primetan je novi deo, koji se odnosi na uvođenje mrežnog rada. To je podešavanje opcija mrežne kartice i tipa mreže na koju će računar biti vezan. Pored vezivanja u radnu grupu Windows računara, WFW omogućava i vezivanje na sekundarne mreže: „Novell” NetWare, „Microsoft” LAN Manager i kompatibilne mreže („3Com” 3+Open, „DEC” Patchworks i „IBM” LanServer).

ŠTA PADA U OČI

Kako kažu u „Getting Started” priručniku, „Windows for Workgroups se lako startuje”. Posle uzimanja u obzir ove kratke rečenice kroz koju do nas dopire veliko iskustvo američkog pisca uputstva, otkucao sam win i pritisnuo Enter. Posle uobičajenog ekrana sa logoom, pojavljuje se prvi novitet: dijalog za unošenje korisničkog imena i lozinke, nakon čega



File Manager je dobio nove funkcije, posebno za rad u mreži



Program Manager

se vaš računar prijavljuje mreži. Korisničko ime može se poklopiti sa nazivom koji ste vašem računaru dali da nosi u mreži, ili može biti različito – ako mašinu koristi više korisnika. Pretpostavljeno stanje je da WFW zahteva lozinku pri ulasku u sistem, ali se to može i eliminisati.

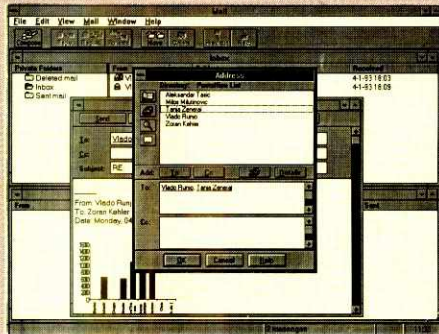
Novе stvari:

- poboljšani File Manager
- poboljšani Print Manager
- ClipBook Viewer kao proširenje funkcije Clipboard-a
- dodatne opcije za Control Panel
- Chat, za časkanje na mreži
- Net Watcher, za pregled stanja mreže
- WinMeter, za praćenje performansi Windows-a pod mrežom
- Mail
- Schedule+

Novi izgled nekih svojih delova Windows for Workgroups je nasledio od Windows-a NT: to je novi look za File Manager i Print Manager, kao i dodatne opcije za Control Panel. Sa strane korisničkog interfejsa, File Manager i Print Manager imaju sada Button Bar ispod menija, liniju sa dugmadima za aktiviranje najčešće korišćenih operacija.

File Manager dobio je nove funkcije, uglavnom vezane za rad u mreži. Ako želite da ostali članovi grupe imaju pristup nekom od direktorijuma na vašem disku, proglašićete taj direktorijum za deljeni. Deljene može podrazumevati read-only ili pun pristup, sa ili bez lozinke koju drugi korisnik mora znati. Deljenje možete postaviti kao stalno, pa će se ono uspostavljati automatski svaki put kada uđete u Windows. Izbor Connect opcije iz Disk menija daje vam mogućnost da se priključite na deljene direktorijume drugih korisnika. Interesantno je da se na istom mrežnom kablju može naći više grupa, pa prvo birate grupu, zatim člana grupe i na kraju sam direktorijum kojem želite da pristupite. Mogućnost da slobodno rovarite po disku kolege koji sedi preko puta vas sigurno će vam biti apsolutni novitet, naročito ako ste naviknuti na usluge i mogućnosti mreža tipa „Novell“ NetWare i sličnih. File Manager sada liči na istu aplikaciju u Windows NT sistemu, ali sa manjim mogućnostima kontrole pritupa i prava korisnika.

Print Manager nije mnogo promenjen, osim što je prerusen u NT-look. Štampači se još uvek instaliraju i podešavaju preko Control Panel-a, a ne kao u Windows NT, iz Print Manager-a. Međutim, noviteta ima, u vezi sa radom na mreži. Proglašavanje vašeg lokalnog printera za dostupnog svima vrlo je jednostavno; i ovde se može postaviti uslov da pridošica na vaš računar mora znati lozinku, ako želi da koristi vaš štampač. I u Print Manager uveden je Button Bar, sa najvažnijim komandama. Iz-



Mail: Slanje poruka i datoteka drugim korisnicima mreže

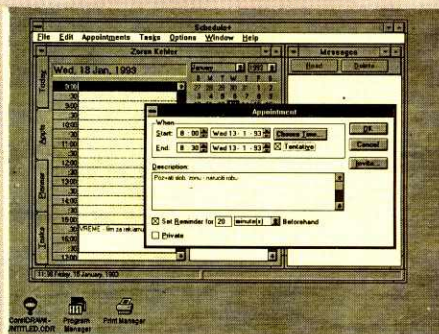
gled Button Bar-a i komande koje su na njemu može menjati iz Options menija.

Koncept beležnice – Clipboard-a – u WFW je podignut na viši nivo. Clipboard je postao samo deo ClipBook-a; svaki ClipBook može sadržavati više objekata koji se na njega prenose pomoću Clipboard-a. Svaki objekat koji se prenese na ClipBook naziva se stranica. Sadržaj ClipBook-a možete videti kao spisak sa imenima objekata, ili ćete izabrati thumbnail prikaz, kada svakom objektu odgovara mala sličica sa prikazom sadržaja. Svoj lokalni ClipBook možete deliti sa ostalim članovima grupe, jer se ClipBook-u može pristupiti kao i deljenim direktorijumima. ClipBook može sadržavati svaki valjani Windows objekat. Ovo otvara mogućnost da se OLE (Object Linking and Embedding) koristi između dva odvojena računara. Na taj način možete objekat sa drugog računara preneti u neki od dokumenata na vašem računaru. Ako objekte povežete, sve promene koje korisnik udaljenog računara izvrši na izvornom dokumentu automatski se prenose i na sve određene dokumente.

Novost u Control Panel-u je Network deo, iz koga se podešavaju parametri mrežne kartice, biraju dodatni tipovi mreža koje će se paralelno koristiti, parametri prijavljivanja na sistem i lozinke. U Network opciji možete odrediti ime koje će računar nositi u grupi, ime grupe, odnos vremena koje će Windows odvojiti za mrežne poslove i vremena za lokalne poslove, kao i da li će resursi računara biti deljeni ili ne. Adapters opcija služi za menjanje parametara mrežne kartice. Ako promenite parametre Windows će vam ponuditi da ponovo startujete program i time preuzmete nove parametre, ili nastavite sesiju sa istim parametrima. U Logon delu zadajete ime pod kojim ćete biti prijavljivani na mrežu. Odavde možete dati komandu da se odjavite sa mreže. U tom slučaju korisnici sa drugih računara i dalje mogu koristiti vaše resurse, ali vi više ne moete pristupiti resursima drugih računara. Izbor Network opcije vodi vas do dijaloga u kome možete izabrati jedan ili više tipova mreža koje će biti aktivne istovremeno kada i WFW peer-to-peer mreža. Na spisku se nalaze NetWare i LAN Manager, a može se dodati i neka od mreža kompatibilnih sa LAN Manager-om.

ŽIVOT NA MREŽI

Chat je nova aplikacija za časkanje na mreži. Ne znam da li spada u uusušne ili rekreacione programe – to znaju oni koji su imali prilike da rade sa sličnim programima, kao što je Phone pod operativnim sistemom VAX/VMS. U vremenima pre rata nije bila retkost da se započne razgovor između tri ili četiri korisnika iz cele Jugoslavije, preko JUPAK i DECNET mreža. Chat omogućava vezu izme-



Schedule+: Elektronski rokovnik za planiranje aktivnosti

đu samo dva korisnika. Program je ukrašen uobičajenim Windows folklorom: ako instalirate drajver za zvučnik (nije u paketu sa WFW!) ili imate zvučnu karticu, imaćete zadovoljstvo da kada pozovete nekoga čujete zvuk koji odgovara onome što se u slušalici čuje kada dobijete broj, dok će onaj koga zovete čuti zvonjavu telefona, dosta bučnu.

Net Watcher je novi uslužni programčić (applet, kako kaže „Microsoft“) kojim možete posmatrati koje resurse – štampače, datoteke i direktorijume – koriste korisnici u vašoj grupi. Naravno, ovo se dešava samo u Enhanced modu, jer je deljenje resursa jedino tada moguće. U prozoru se može videti lista računara koji su „zakačeni“ na vaše resurse, kao i lista resursa koji su stavljeni na raspolaganje članovima grupe.

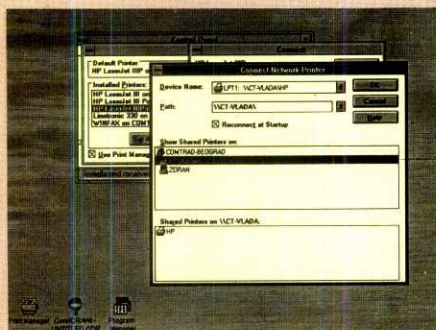
WinMeter grafički prikazuje procenat vremena procesora koje se dodeljuje procesima koji pristupaju vašim resursima, kada Windows radi u Enhanced modu. Ako neko iz vaše grupe pošalje datoteku na štampanje na vašem štampaču, WinMeter će pokazati koji se deo procesorskog vremena angažuje dok drugi korisnik pristupa datoteci, koristi je a zatim štampa preko vašeg štampača.

POŠTA I DNEVNI RASPORED

Mail je aplikacija za slanje poruka i datoteka drugim korisnicima mreže. Pre nego što počnete da koristite Mail, vi ili neko drugi iz vaše grupe morate kreirati Workgroup Postoffice, zajednički poštanski ured za korisnike u grupi. Onaj ko se lati tog posla (i preuzme u odgovornost) postaje administrator Mail-a. Za svakog korisnika kreira se datoteka sa porukama, koja postaje njegov privatni poštanski sandučić.

Kada po prvi put pokrenete Mail, morate se povezati sa postojećim poštanskim uredom, a zatim kreirati korisnika, ako administrator to već nije učinio. Svaki sledeći put u Mail ulazite tako što unosite ime i lozinku koje ste izabrali za vaše poštansko sanduče. Ako želite, ceo postupak se može i automatizovati: za Mail program u File..Properties meniju u Command Line polje na kraj linije unesite ime i lozinku, pa ćete biti pošteđeni ukucavanja svaki put pri ulasku u program. Naravno, možete izostaviti lozinku, ako dosta držite do bezbednosti vaših podataka.

Svakoj poruci može se priključiti jedna ili više datoteka. Pošto se radi o vrlo simpatičnom načinu za širenje virusa, uputstvo upozorava da te datoteke treba testirati na prisustvo virusa. U poruke se mogu ubaciti Embedded objekti, što omogućava da u poštu ubacite dokumente iz drugih aplikacija, a da oni ne izgube formatiranje.



Veživanje na mrežni printer

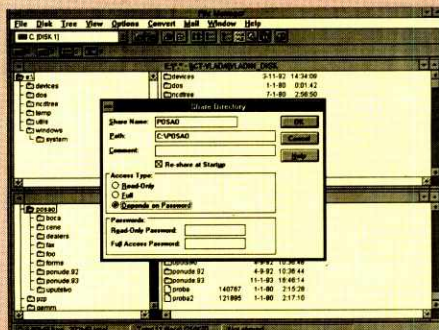
Prema dokumentaciji, radi se o aplikaciji nastaloj od samostalnog „Microsoft” *Mil* paketa, koji je sada u verziji 3.0. Iako perfektno funkcioniše u *WFW* okruženju, da biste se povezali na spoljne *mail* servise drugih proizvođača ili na *mail* na globalnim računarskim mrežama, paket treba unaprediti do verzije 3.0 i dodati softverski *gateway* za spoljne poštanske servise. *Mail* ne dozvoljava da jedan korisnik bude priključen na više poštanskih uređa, što u nekim situacijama može ograničiti upotrebljivost programa.

Schedule+ je elektronski rokovnik, ili raspored časova – možete ga zamisliti na oba načina, zavisno od toga za šta ga koristite. To je alat koji vam pomaže da vodite evidenciju o važnim sastancima i radnim zadacima, da izdvojite iz vaše satnice termine za sastanke, i da sami sebi ostavljate poruke o vašim dnevnim aktivnostima. Pošto *WFW* omogućava deljenje resursa, paradigma satnice može se u **Schedule+** proširiti i na resurse: vi koristite štampač od 9 do 10 ujutru, vaš kolega od 10 do podneva, itd. Pošto je prilagođen za rad u mreži, program omogućava da uvidom u satnice ostalih korisnika planirate zajedničke aktivnosti. **Schedule+** zahteva da ste ranije postali korisnik **Mail WPGO**; vaše ima za poštu i lozinku morate ukucati kada se pojavi prvi dijalog prozor.

Ekran **Schedule+** sadrži prozor satnice, sa poljima za satnicu sastanaka (**Appointments**), planer (**Planner**) i listu radnih zadataka (**Tasks**). Satnica je podeljena na intervale (**slots**), od kojih svaki može da sadrži sigurni ili mogući sastanak, ili njihovu kombinaciju. Ako se neki sastanak ili obaveza ponavlja svaki dan, **Schedule+** će to automatski uneti u raspored za sledeće dane. Ako ne želite da ostali korisnici vide na šta se odnosi svaka od vaših obaveza, možete ih označiti kao privatne – tada se i dalje pojavljuju u pregledu koji ostali vide, ali bez naznake aktivnosti. U raspored se unose i radni zadaci; možete im dodeliti prioritet, planirani rok završetka i grupisati ih u projekte. Kada želite da zakažete sastanak sa ostalim korisnicima, to činite tako što im šaljete zahtev za sastanak; zahtev se dostavlja pomoću **Mail** programa. Pored *online* rada na mreži, **Schedule+** omogućava i *offline* rad. To ćete koristiti kada radite kod kuće, ili na vašem prenosnom računaru. Sve obaveze unesene u satnicu na ovaj način kasnije možete jednostavno priključiti satnici koju vodite na računaru u kancelariji.

RASPODELA RADNE SNAGE

Prema „Microsoft”-ovoj postavci umrežavanja, svaki računar u *WFW* grupi može istovremeno biti i server i radna stanica. Ovakav na-



Deljenje prostora na disku

čin rada jako se razlikuje od rada klasičnog servera „Novell” tipa, koji je obično gurnut u čošak sobe, i na kojem niko ne radi. S druge strane, to znači da radnu snagu vašeg računara stalno delite sa korisnicima koji koriste vaše resurse. Ovo dobrom delu korisnika može predstavljati problem: još uvek se nije postiglo da standardna radna mašina bude bar PC sa 386DX procesorom na 40MHz, brzim diskom i video kartom, bar ne na našim prostorima.

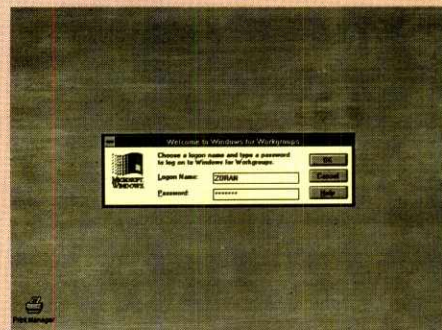
Kako se *WFW* ponašao u ovom pogledu na testovima? Dobro, može se reći. Namerno sam kao računare u grupi pomešao mašine sa 386DX/40MHz i 386SX/25MHz procesorima, kao situaciju koja je unekoliko bliska realnim uslovima. Jedan računar bio je izabran za **WGPO** – **Workgoup Postoffice** (centralni računar za **Mail**), a na njega je bio vezan i zajednički laserski štampač. Na drugi računar u mreži bio je vezan matični štampač, koji je imao dosta posla. Preostala dva računara nisu imala mnogo posla sa zahtevima ostalih stanica. U ovakvoj organizaciji videlo sa da zahtevi ostalih članova grupe ipak mogu značajano usporiti rad računara koji ih obrađuje. To se pre svega odnosi na **Mail** program, koji zbog svoje organizacije jako opterećuje procesor mašine određene da bude zajedničko poštansko sanduče. Zahtevi za štampu na laserskom štampaču nisu predstavljali veliki problem.

Treba reći da *WFW* nije za 286 računare: oni mogu samo koristiti usluge drugih, a ne mogu ih sami pružati, jer je deljenje resursa moguće jedino ako procesor radi u 386 *Enhanced* modu. 386SX radi prihvatljivo, ali imajte u vidu da je za *WFW* neophodno bar 3MB slobodne memorije, ako želite da procesor prirodno startuje u *Enhanced* modu, bez navođenja /3 opcije pri pokretanju. Uopšte, i sa 386DX računarom ne treba počinjati posao bez ugrađenih 4MB, a bolje je bar 8MB.

Nemojte misliti da se svi resursi mogu deliti: nisam mogao da odolim da vidim šta će se desiti ako pošaljem jedan posao za štampanje na fax/modem karticu, ugrađenu u jedan od računara. Karticu je podržavao program **WinFax**, koji stvari postavlja tako da se fax vidi kao štampač priključen na jedan od serijskih portova. Na računaru sa kojeg sam poslao posao nije bio instaliran **WinFax**. Izabrao sam štampanje preko mreže na fax karticu, poslao posao, i... ništa. Posao za štampu je jednostavno negde usput ispario. Moraću još da ispitam da li se fax karta može deliti, jer bi to predstavljalo vrlo zanimljivu stvar.

SEDI I VOZI

„Microsoft” je sa *WFW* prokrijumčario i *peer-to-peer* umrežavanje pod DOS-om. Protokol drajveri, drajveri za kartice i *Workgroup*



Ulazak u program preko lozinke

servise nisu vezani samo za *Windows*, već se mogu koristiti i iz DOS-a. **Setup** u **AUTOEXEC.BAT** datoteku upiše liniju **net start**, čime se zapravo iz DOS-a startuje mreža. Naredbom **net** se kasnije mogu dobiti usluge mreže: vezivanje na deljene direktorijume i štampače. Sve ne funkcioniše tako lepo kao pod *Windows*-om, ali je ipak tu. Doduše, ne može se oprostiti potpuno izostavljanje podataka o DOS strani umrežavanja iz kompletne dokumentacije. Jedino što možete dobiti je škrti help iz same naredbe.

Ni pojedini delovi *WFW*, naročito oni koji se odnose na rad pod mrežom, nisu dovoljno ili uopšte objašnjeni. Na više mesta u priručnicima nalaze se lakonske napomene tipa „... a o ovome možete više saznati iz **on-line help-a**.” Isto tako, **Setup** će u vaše **CONFIG.SYS** i **AUTOEXEC.BAT** datoteke upisati desetak linija čiji sadržaj nije objašnjen, osim što se iz naziva tih komandi može otkriti neka namena čemu služe. S ove strane, izgleda kao da je *Windows for Workgroups* na tržište izbačen u velikoj žurbi. Sa funkcionalne strane nisam imao nikakvih problema za sve vreme testiranja novog softvera.

Nakon rada sa novim *Windows*-om u trajanju od par dana utisci su jako povoljni. „Microsoft” je ovim proizvodom napravio pun pogodak. *WFW* zadovoljava korisnika u savremenom poslovnom okruženju, čije potrebe veće mreže prevazilaze – *NetWare*, *LAN Manager* i ostale. To se, pre svega, odnosi na situacije kada je potrebno povezati desetak PC-ja, jer se želi deljenje informacija i nekih resursa, ali ne postoji potreba za postojanjem svih mrežnih servisa i kompletne mrežne administracije. Mislim da znate na šta mislim: za pokretanje *NetWare* mreže potrebno je izdvojiti poseban računar kao server, na njemu instalirati mrežni softver, pa zatim podesiti sve parametre mreže, napraviti stablo direktorijuma na serveru, podesiti prava pristupa korisnika, definisati ređove za štampu, i tako unedogled. *WFW*, s druge strane, dopušta izvestan stepen prijatne anarhije: ubacite u računar mrežnu kartu, povežite računare međusobno kablovima, označite vaš štampač ili direktorijum na disku kao deljeni resurs i – počnete da radite. Sve je gotovo intuitivno, i ako imate radno iskustvo sa *Windows*-om – koristi se na isti način kao i sve ostalo pod *Windows*-om. To je, naravno, i bila ideja iza celog *Windows* okruženja.

Moja preporuka: požurite i umrežite svoje računare pomoću *Windows for Workgroups*!

Korisna adresa

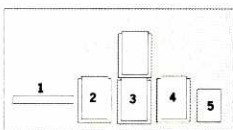
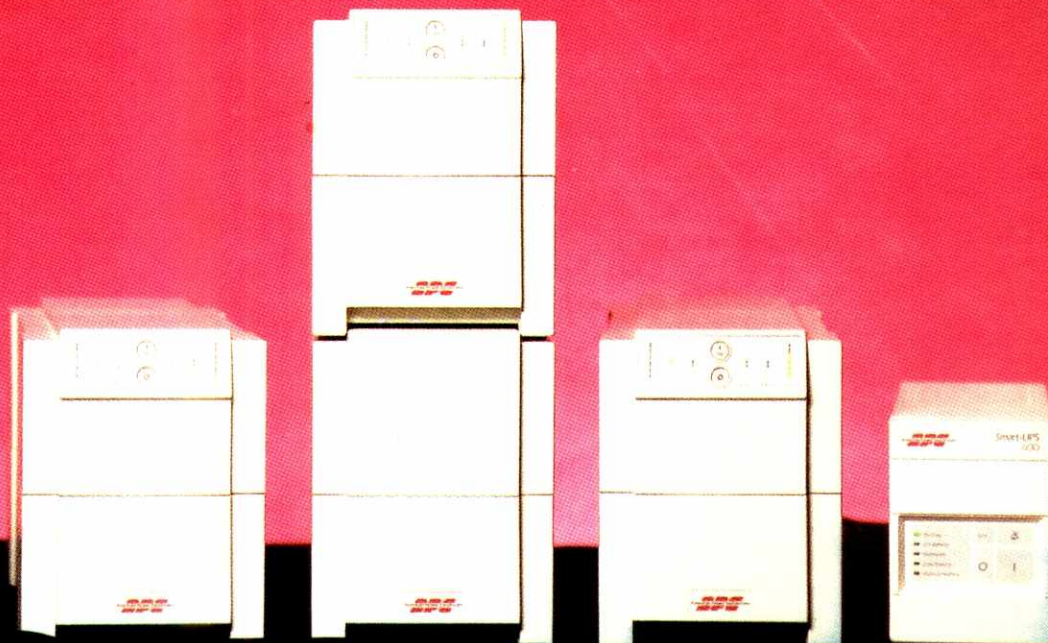
PC Club d.o.o.
11000 Beograd, XXI divizije 41
Tel: 011/444-30-79, Fax: 422-199

The Smart-UPS™ Series

High-performance protection for LANs, Unix systems, engineering workstations, and minicomputers

Virtual Library of Faculty of Mathematics - University of Belgrade

elibrary.matf.bg.ac.rs



1. Smart-UPS 400

for 286, 386 desktop systems and servers

2. Smart-UPS 900

for CAD/CAM, engineering workstations and servers

3. Smart-UPS 2000

for minicomputers, multiple servers

4. Smart-UPS 1250

for multiple servers, telecom equipment

5. Smart-UPS 600

for tower-type servers and engineering workstations

Featuring

Line Interactive design

An innovative design in which the UPS inverter is connected to the output, providing a cleaner response to utility problems and superior filtering when compared to standby UPS technology.

Battery replacement warning

A Smart-UPS automatically tests the health of its batteries and alerts you to potential problems before batteries wear out.

SmartBoost™

A Smart-UPS provides true brownout correction, allowing you to work through problems that shut other UPS systems down.

Sine wave output

For complete compatibility with all applications.

UPS - Link™ Control language

With a Smart-UPS, your site's power quality can actually be monitored and events can be logged automatically. You'll have hard-copy evidence of your return on investment.

Lightning, surge and noise protection

The Smart-UPS outperform other UPSs when subjected to ANSI/IEEE class "A" and "B" surge tests. Full-time EMI/RFI filters prevent line noise from corrupting data files.

Site and unit diagnostics

Automatically spot poor ground and reversed polarity, two common miswirings which usually require an electrician to diagnose.

Full approvals and warranty

UL, CSA, TUV and Novell approvals mean a safe, reliable solution to your power problems.



AUTHORIZED DEALER **"SAGA" D.O.O. BEOGRAD, YUGOSLAVIA**

Milentija Popovića 9. "SAVA CENTAR" 11070 NOVI BEOGRAD

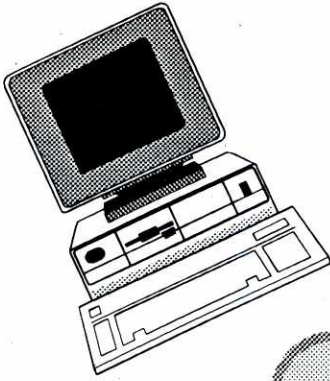
Tel: & Fax: 011/222-3579; 147-182; 222-4323 ext. 256 & 259

Plus po plus...

Računarski sistemi,
računarske komponente,
licencni software,
štampači,
najbolje cene !

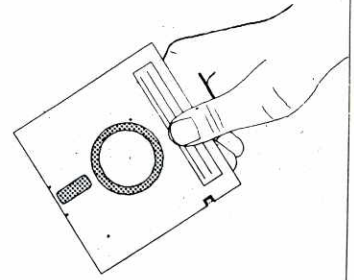


Plus u Vašem džepu



PONUĐA MESECA

Interni streamer
IOMEGA - USA
250Mb
cena 650 DM



A CENE?

1. PRELISTAJTE RAČUNARE,
2. PRONADJITE NAJBOLJU PONUDU
3. JAVITE NAM SE!

MONITORI
HERCULES
Monohromatski VGA
Color SVGA

OSNOVNE PLOČE
486/50, 256Kb cache
486/33, 256Kb cache
386/40, 64Kb cache
386/33, 64Kb cache
386SX/25
286/16

KARTICE...
HERCULES
VGA 256Kb
VGA 512Kb, 1Mb
VGA 1Mb, 32 hiljade boja
WINDOWS akcelerator S3
Sound Blaster
I/O+AT BUS kontroler
ETHERNET

FLOPPY, HARD, CD-ROM...
1,2Mb 5,25"
1,44Mb 3,5"
diskovi 40Mb do 1,2Gb
CD-ROM drive

MEMORIJSKI MODULI
SIMM 4Mb/70ns
SIMM 1Mb/70ns
SIMM 256Kb/70ns

DISKETE I STRIMER-TRAKE
Diskete 1,2Mb MAXELL
Diskete 1,44Mb BASF
Diskete 1,2Mb NO NAME
Diskete 1,44Mb NO NAME
Strimer traka DC-2000, SONY, 3M
Strimer traka DC-2120, DYSAN, CARLISLE

TASTATURE
YU standard
ASCII standard

RAČUNARI

286, 386, 486
sastavite konfiguraciju
koja Vam odgovara!

PC CLUB d.o.o.
Beograd, XXI divizije 44
tel. 444-30-79
fax. 422-199

Interni
fax/modem:
2400 bauda modem
9600 bauda
send/receive fax

MS DOS 5.0
170DM

PROGRAMSKI PAKETI

DOS, XENIX, NOVELL
Programski jezici
Baze podataka
Obrada teksta

WINDOWS i aplikacije
Programi na CD ROM-u



ZIP JE STIGAO

Ne pamtimo da je neki program toliko „očekivan“ kao verzija 2.0 popularnog arhiviera PKZIP, najavljena je još pre dve godine! Prošle godine je stigla alfa verzija, a onda smo čekali, čekali... Radovali su se jedino autori virusa koji su s vremena na vreme izbacivali „lažni“ PKZIP 2.0 koji se brzo širio američkim BBS-ovima i prvio veće ili manje štete. Pred sam kraj 1992. godine firma PKWare Inc nas je ipak obradovala autentičnim PKZIP-om 2.04c. Jedva par dana posle svetske premijere ovaj shareware program je stavljen na raspolaganje i korisnicima Sezama, a verujemo da će i čitaoci „Računara“ biti zainteresovani za njegove karakteristike...

Istorija PKZIP-a je duga i prilično zanimljiva – par godina posle promocije PC računara pojavili su se prvi programi za arhiviranje podataka, a jedan od zapaženijih produkata tog tipa bio je PKARC firme PKWare Inc (9025 N. Deerwood Drive, Brov'n Deer, WI 53223, USA) – slova PK su zapravo inicijali autora programa Fila Katsa (*Phil Katz*), a ARC je trebalo da asocira na arhiviranje. Na žalost gospođina Katsa, u to vreme je već postojao arhiver zvani ARC, pa su njegovi autori tužili PKWare za povrecu autorskih prava. Spor se otego do 1987. godine. PKWare ga je konačno izgubio, ne toliko zbog sličnih formata koliko zbog same ekstenzije ARC. Vansudskim poravanjem obe strane su se složile da gospodin Kats prestane da koristi ovu ekstenziju i ovaj format tako da poslednja verzija programa PKARC (koja je kod nas i dalje u priličnoj upotrebi) nosi datum 27. april 1987. Početkom 1989. godine PKWare izbacuje sasvim novi arhiver, po svim karak. eristikama znatno bolji od PKARC-a, ali (zbog pomenute presude) vertikalno nekompatibilan sa njim – PKZIP. Pojavile se nekoliko verzija tog programa da bi ona konačna, 1.10, opstala na tržištu skoro tri godine – izašla je 15. aprila 1990. Za to vreme PKZIP je ušao u veoma široku upotrebu, pre svega po razm. BBS-ovima koji su ga proglasili za zvanični arhiver; rado ga, naravno, koriste i svi oni koji nemaju mod. em. već „samo“ skladište veće količine podataka na diskete.

SLIČNOSTI...

Naravno, ni konkurencija za sve to vreme nije spavala – pojavio se ARJ Roberta Junga koji je pakovao brže i više od PKZIP-a i imao broj. e opcije koje su povećavale komfor ali i zahtevale pamćenje raznih komandi i parametara. PKZIP je, ipak, „preživio“ – dobro je poznato da je u softverskoj industriji jako važno biti prvi, tako da su mnogi zbog inercije nastavili da koriste PKZIP kojim su bili zacovoljni. I nisu požalili, naročito kada su se pokazale neke nepouzdanosti u radu ARJ-a koje je najzad priznao i sam njegov autor. Bilo kako bilo, ljubitelji PKZIP-a su dve godine čekali novu verziju – u međuvremenu ih je *Phil Katz* obradovao samo jednom beta verzi-

Dejan Ristanović

jom novog arhiviera koje je nosila oznaku 1.93A. PKZIP 2.0 je najavljivan za kraj 1991, pa za početak 1992, sredinu, septembar... povremeno smo na Sezamu prenosili diskusije sa PKWare-ovog BBS-a iz kojih se moglo videti da su autori naišli na probleme sa kompatibilnošću, neke bagove mikroprocesora 80486 i svašta drugo. Strpljenje je na kraju, ipak, nagrađeno – poslednji dani 1992. doneli su nam i novi PKZIP koji je, verovatno da bi se razlikovao od „trojanaca“ koje su razni zlonamerni programeri poslednjih godina izbacivali, dobio oznaku 2.04c.

PKZIP 2.04c u osnovi liči na PKZIP 1.10 pa i na stari PKARC – komandom PKZIP ime_arhive ime_datoteke kreiramo arhivu (standardna ekstenzija ZIP) u koju se smešta datoteka ili datoteke koje smo specifikovali – PKZIP TEST *.* će, na primer, u arhivu TEST.ZIP upisati sve programe iz tekućeg kataloga. Raspakivanje arhive obavljamo sa PKUNZIP ime_arhive posle čega se u tekućem katalogu pojavljuju sve zapakovane datoteke. Najzad, za ZIP2EXE možemo da konvertujemo bilo koji .ZIP u izvršni program koji se, po startovanju, sam raspakuje – ovo je jako zgodno kada arhivu treba poslati nekome ko možda nema PKUNZIP ili ne ume da ga koristi.

Iako je ovaj osnovni oblik mnogima dovoljan, PKZIP obezbeđuje razne opcije koje se navode iza znaka minus, recimo PKZIP -pr ARHIVA *.* Samo se po sebi razume da je PKZIP 2.0 vertikalno kompatibilan sa prethodnom verzijom tako da sve poznate opcije i dalje rade. Uvedene su, naravno, i neke nove...

... I NOVITETI

Glavni novitet je opcija -& koja omogućava upis arhive na veći broj disketa – često se, naime, događa da je arhivirana datoteka veća od 1.2 ili 1.4 megabajta (diskete od 360 K su već davno „zaboravljene“), što je ranije zahtevalo arhiviranje na hard disku i daljenje arhive na diskete, bilo pomoću nekog specijalnog tome namenjenog programa kao što je LCOPY bilo pomoću standardnog (ili nekog komercij-

jalnog) BACKUP-a. Ovakvo distribuiranje arhiva nije naročito pogodno pošto pratilac mora da poseduje odgovarajući RESTORE (razne verzije DOS-a znaju da naprave ozbiljan problem) i, što je ponekad još teže podrazumevati, znanje potrebno da se operacija obavi. Zato novi PKZIP to radi sam – „kažete“ mu PKZIP -& A:ARHIVA *.* i sve datoteke u tekućem direktorijumu biće arhivirane na diskete – računar će tražiti nove i nove diskete dok sve što je potrebno ne bude upisano na njih.

Prirodno pitanje je šta se događa sa onim što je već bilo upisano na te diskete. Ukoliko koristite samo opciju -&, sadržaj disketa ostaje nedirnut – arhiva se nadovezuje na ono što već stoji na disketi, naravno u granicama slobodnog prostora. To najčešće nije ono što korisnik želi – diskete koje će primiti arhivu bi se, dakle, prethodno morale obrisati što je vremenski prilično zahtevna operacija. Osim toga, početnik će se začuditi ako mu PKZIP zatraži mnogo više disketa nego što je predvideo – ako su diskete bile skoro pune, na svaku od njih je, umesto punog kapaciteta, došlo svega po par kilobajta! Zato će se u praksi uglavnom koristiti opcija -&w koja briše sadržaj svake diskete (kako osnovni direktorijum tako i eventualne poddirektorijume) – komanda PKZIP -pr&w A:DISK C:\.*.* će, na primer, napraviti arhivirani backup diska C:; na potrebnom broju disketa. Ukoliko su neke od tih disketa u startu bile neformatirane, koristite -&f umesto -&w.

Što se rada sa više disketa tiče, ovo je otprilike sve što će vam u praksi i biti potrebno. Usvajanjem rešenju, ipak, nedostaje fleksibilnost koju nudi ARJ – tamo možete da kreirate arhive proizvoljnih dimenzija kako na disketama tako i na hard disku. Ovo je jako zgodno kada, na primer, želite da pošaljete neku arhivu na BBS a „patite“ od ograničenog vremena na njemu – arhivu ćete lako podeliti na delove čije su dimenzije usklađene sa vašim vremenskim limitom i onda pri svakom pozivu slati po jedan od njih. Ovakvo nešto sa PKZIP-om ne ide. Postoje i neki drugi, manje važni, dodaci koje ARJ na ovom planu nudi, ali treba znati i da su ti dodaci ponešto iskomplikovali upotrebu ARJ-a. Ukratko, u pitanju je veći dilema softverske industrije – autori ARJ-a su odlučili da ponude veću fleksibilnost uz rizik da upotreba tih opcija postane (pre)komplikovana. Autori PKZIP-a su se odlučili za sasvim jednostavnu upotrebu, ali su izgubili na fleksibilnosti i upotrebljivosti u nekim specifičnim situacijama.

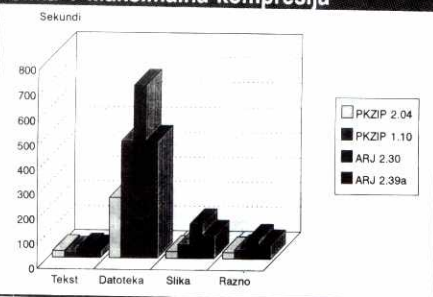
Druga zanimljiva novost je PKSFXJR – program koji omogućava kreiranje samoraspakujućih .EXE datoteka koje će po dužini biti vrlo bliske originalnom .ZIP-u – pri automatskom raspakivanju tada nisu na raspolaganju sve luksuzne opcije i help-ovi, ali samoraspakujući .EXE je najčešće namenjen nekome ko se i tako mnogo u te opcije ne razume. Tu je i program PKUNZIP Jr koji zahteva znatno manje memorije od osnovnog PKUNZIP-a, opet uz žrtvovanje nekih opcija. To je naročito pogodno u situaciji kada se PKUNZIP pokreće iz nekog shell-a pa mu nije na raspolaganju mnogo memorije.

Pomoću datoteke PKZIP.CFG arhiver se može prilagoditi ukusu i potrebama korisnika – ukoliko često koristite neke opcije, permanentno ih aktivirajte kreiranjem ove datoteke i ubuduće štedite na kućanju! Vlasnici registrovanog PKZIP-a ovo konfigurisa-

	TEKST		DATOTEKA		SLIKA		RAZNO	
	ASCII tekst dužine 729237		DBF datoteka dužine 9679447		TIF slika duž. 1080186		87 fajlova duž. 1251105	
ZIP max 2.04 st. min raspak	0:27:0 296805 0:19:0 299351 0:09:0 344618 0:05:0		4:04:0 1081071 1:45:5 1141269 0:55:5 1515069 0:39:0		0:28:0 39482 0:11:0 41857 0:05:5 54277 0:05:0		0:25:5 517021 0:26:0 520088 0:15:5 563999 0:19:5	
ZIP max 1.10 st. min raspak	0:21:0 323068 0:21:0 323068 0:07:5 361320 0:05:0		7:55:5 1368093 7:55:5 1368093 1:02:5 1773233 0:39:0		0:55:5 54162 0:55:5 54162 0:06:0 44588 0:05:0		0:35:5 554119 0:35:5 554119 0:14:5 638125 0:18:0	
ARJ max 2.30 st. min raspak	0:34:0 301757 0:25:0 306223 0:13:0 364176 0:08:0		11:37:0 1130529 3:37:0 1224808 1:37:5 1646040 0:45:5		2:18:5 40223 0:24:0 45263 0:09:5 59480 0:05:0		1:18:0 517901 0:44:5 522605 0:25:5 587530 0:24:0	
ARJ max 2.39a st. min raspak	0:27:0 301595 0:24:5 301707 0:11:0 356750 0:07:0		7:54:5 1130048 3:22:0 1181478 1:38:0 1646040 0:47:0		1:32:0 40240 0:25:5 44545 0:08:5 58520 0:05:0		0:56:5 517828 0:38:0 518977 0:22:0 581964 0:22:0	

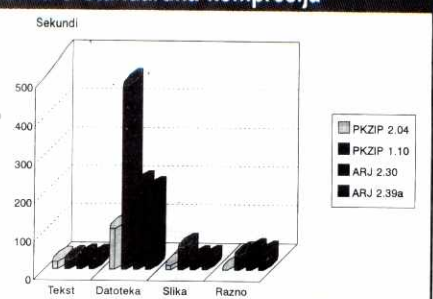
Novi šampion: na svim testovima novi PKZIP pokazuje najbolje rezultate.

Slika 1 Maksimalna kompresija



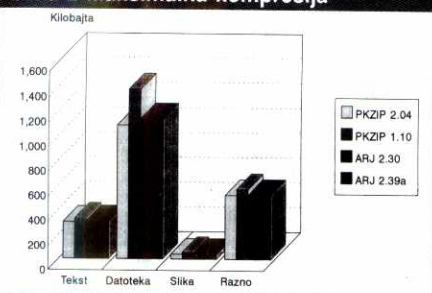
Koliko vremena treba kom arhiveru

Slika 3 Standardna kompresija



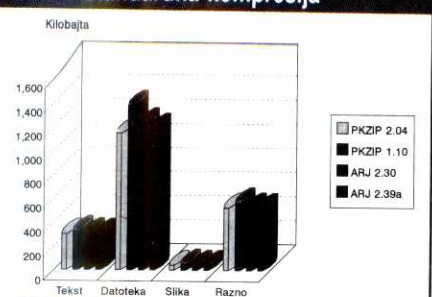
Koliko vremena treba kom arhiveru

Slika 2 Maksimalna kompresija



Koliko se datoteka može koprimovati

Slika 4 Standardna kompresija



Koliko se datoteka može koprimovati

ne mogu poveriti interaktivnom programu PKCFG; ipak, iz njega sve može da se obavi u običnom tekst editoru, ukoliko se pažljivo pročita dokumentacija.

Dosta je učinjeno i na polju sigurnosti – primenom opcije *-slozinka* datoteke možete arhivirati tako da ih raspakuje samo onaj kome je *lozinka* poznata. To je postojalo i u ranijoj verziji ali je, prema implicitnom priznanju autora, kvalitet šifre bio umeren. Sada se tvrdi da je za pogađanje lozinke duge samo 10 slova koja bi se sastojala od osnovnih ASCII karakteriča potrebnom preko 200 miliona godina. Naravno, kriptoanaliza nije samo probanje svih mogućih varijanti tako da u ovakve tvrdnje ne treba baš uvek poverovati. Ipak, šifra je veoma sigurna, sasvim dovoljna za amaterske primene.

Drugi, potencijalno možda interesantniji, vid zaštite pristupačan je samo registrovanim korisnicima PKZIP-a. Često se, naime, dešava da sa nekog manjeg BBS-a preuzmete arhivu u čije poreklo niste baš 100% sigurni – kako da znate da je u pitanju original a ne nekakav trojanac? Ako je autor te arhive registrovao svoj PKZIP, stvari bi trebale da budu znatno jednostavnije – pri registraciji navodite ime svoje firme i PKWare mu dodeljuje identifikacioni kod i kontrolnu sumu kojom registrujete vaš PKZIP.EXE. Ubuduće će sve arhive koje kreirate sa PKZIP-om sadržati ovaj vaš „potpis“ i on će se videti pri raspakivanju – ukoliko se na kraju PKUNZIP-a pojavi ime autora ili njegove firme, sa priličnom verovatnoćom možete da smatrate da je arhiva originalna i da niko nije „čakao“ po njoj. Kažemo „sa priličnom verovatnoćom“ jer je slična opcija postojala i kod starog PKZIP-a 1.10, ali je brzo „provaljena“ tako da su mnogi hakeri ilegalno registrovali PKZIP na razna imena što je ubrzalo širenje raznih virusa i trojanaca. Verovatno zbog toga novi PKUNZIP potpuno ignoriše „potpise“ ugrađene starim PKZIP-om. Sada bi, bar po rečima autora, sve to trebalo da bude dosta pouzdanije, a da li će tako stvarno biti... ostaje da sačekamo i da vidimo!

Kada govorimo o novim opcijama, pomenimo i nove parametre opcije *-e*. Pri radu sa arhivom treba, naime, uspostaviti ravnotežu između vremena izvršavanja i veličine finalne arhive – jasno je da se uz više računanja može više komprimovati i obratno. PKZIP tu nudi više mogućnosti: *-ex* je najkvalitetnija (i najsporija) kompresija, *-en* standardna, *-ef* brza a *-es* superbrza kompresija koja daje najmanji efekat.

Tu je, naravno, i mogućnost da se sa *-e0* datoteke prosto smeste u arhivu bez ikakvog sažimanja. Ukoliko imate dovoljno brz računar, koristite isključivo standardnu ili, eventualno, maksimalnu kompresiju; vlasnici XT-a će se možda opredeliti za neki brži i slabiji algoritam. PKUNZIP automatski prepoznaje izvršenu kompresiju i raspakuje datoteke bez potrebe da se navode posebne opcije.

KRALJ BRZINE

Ovakvo poređani, noviteti PKZIP-a i ne izgledaju kao nešto posebno važno. Pa ipak, glavna stvar nije ni u tim korisnim dodacima, ni u sigurnosti ni o ostaloj „šminki“ koju smo (ili nismo) pominjali. Glavni posao arhivera je da što pre i što više sažme datoteke. A tu novi PKZIP dolazi na svoje.

Kao što se vidi iz tabele 1, novi PKZIP je „ubio u pojam“ kako svoju prethodnu verziju tako i obe „tekuće“ verzije ARJ-a. Što se brzine tiče, jednostavno mu nema premca. Kada govorimo o dimenzijama rezultujućih datoteka, novi metod zvani *Deflating* nije dao neko značajno skraćivanje u odnosu na ARJ, ali je određena prednost uvek (osim u jednom slučaju) bila na strani PKZIP-a. Kratko rečeno, impresivno.

Arhivere smo testirali na više tipova datoteka. U prvom kolonu su podaci za ASCII tekst dužine oko 730 kilobajta (rukopis jedne knjige plus niz poruka sa Sezama), u drugoj za .DBF fajl dužine 9.7 megabajta (podaci o svim firmama u Vojvodini), u trećoj za sliku dužine 1.1 megabajta (tehnički crtež) i u poslednjoj rezultat kompresije sadržaja jednog „privremenog“ direktorijuma našeg diska u kome je bilo 87 fajlova, od toga dosta teksta, nešto programa, nešto arhiva (zip i arj) itd, ukupne dužine 1.2 megabajta. Sve arhivere smo testirali pri maksimalnoj, podrazumevanoj i najbržoj kompresiji, koristeći „standardan“ 386/33/64K cache računar sa 4 megabajta RAM-a i brzim Conner diskom – na 80486 se dobijaju relativno slične proporcije, ali pretpostavljamo da većina čitalaca i dalje radi na 80386 mašinama pa smo prezentiramo ove rezultate.

Jedinu situaciju u kojoj ARJ pouzdano više komprimuje od PKZIP-a primetio je naš kolega sa Sezama *nkbog* – arhive koje se sastoje od puno „malih“ datoteka. Ta ušteda, međutim, ide na uštrb sigurnosti: PKZIP ima centralni katalog koji se nalazi na kraju arhive. On mu obezbeđuje lako oporavljanje arhive ukoliko je njen kraj sačuvan. Ukoliko je pak centralni katalog oštećen, PKZIP će postupiti slično

ARJ-u i oporavljati datoteke redom. U slučaju Marfijskih oštećenja *jedne* od datoteka, sve koje su ARJ-ovane iza nje biće nečitke; PKZIP isti podatak ima još na jednom mestu, pa je i sigurnost veća.

Toliko o kompresiji, a sada da pokušavamo da odgovorimo na pitanje kako PKZIP-u 2.04c uspeva da bude *ovoliko* brži od svih drugih programa. Stvar možda i nije tako složena kao što izgleda – autori PKZIP-a odlučili su da koriste 32-bitne 80386/80486 instrukcije kao i svu memoriju (EMS, XMS, UMB itd) koju „nadu“ u sistemu. Pored toga, otkrivanje DPML-a (*DOS Protected Mode Interface*) koji, recimo, podržava *Windows* garantuje prelazak u pravi 32-bitni mod uz dalje ubrzanje; otkrivanje rada pod *Novell*-om može stvar dalje poboljšati. Priča se da su baš ova odluka izazvale veliko kašnjenje PKZIP-a pošto, kako se pokazuje, nije lako raditi 32-bitno pod DOS-om, a isplivali su i neki *bug*-ovi samih mikroprocesora. Bilo kako bilo, ne verujemo da će neki budući ARJ moći da se takmiči sa PKZIP-om samim poboljšanjem algoritama – moraće da „zagriže“ 32-bitni „kolač“.

NE MOŽE BEZ PROBLEMA

Sve ovo lepo deluje na papiru ali u praksi... nije baš savršeno! Prvi i najozbiljniji problem je DMP1 – pokazuje se da na mnogim sistemima, u prisustvu raznih QEMM-ova, keš programa itd. pokretanje PKZIP-a iz DOS prozora *Windows 3.1* izaziva raznorazne poruke o greškama, „zaglupljuje“ računar ili čak proizvodi .ZIP datoteke koje se docnije ne mogu raspakovati. Ukoliko se ovo događa i na vašem sistemu, ostaje vam samo da u svojoj AUTOEXEC.BAT ugradite *SET PKUNZIP=-* i tako naložite programu da zanemaruje prisustvo DPML-ja.

Dva korisnika Sezama i, reklo bi se, veći broj korisnika raznih američkih mreža i BBS-ova prijavili su i druge probleme koji se svode na kreiranje arhiva koje se docnije ne mogu raspakovati – nije još jasno u kakvim slučajevima se to dešava i kako se može sprečiti, ali izgleda da se problem najčešće javlja prilikom podele arhive na više disketa. Mi smo dosta koristili PKZIP i nismo imali sličnih iskustava, ali nam je poverenje u njega poljuljano ovakvim tvrdnjama – najzad, nepouzdanost i jeste bila glavna mana ARJ-a 2.30! Ostaje da se svaka .ZIP datoteka obavezno testira sa *PKUNZIP -t* – gubite malo vremena, ali ste bar sigurni da nema grešaka!

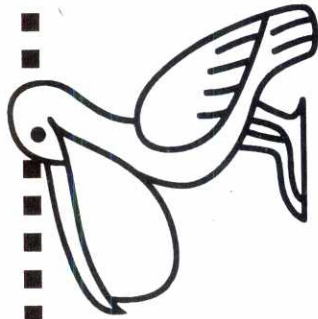
Što se „sitnih“ zamerki tiče, pronađena je jedna manje važna opcija (*-f*) koja ne radi, kritikovan tretman greške *Drive not ready*, uočeno da opcija *-&w* ne vodi računa o lošim sektorima na disketi što može da donese velike probleme... Primitili smo, najzad, da *Norton*-ov antivirus program prijaluje da je PKZIP zaražen *Maltese Amoeba* virusom – radi se, srećom, o lažnoj uzbuni koja je mnoge itekako zabrinula!

PKZIP 2.04c je, kao i prethodne verzije, *shareware* program koji možete slobodno nekomercijalno koristiti i distribuirati prijateljima. Komercijalna upotreba podrazumeva registraciju koja staj 47 dolara i trenutno obuhvata znatno detaljniju dokumentaciju, par uslužnih programa i novih opcija među kojima zapaženo mesto zauzima pomenuto *-l*.

Na kraju bismo rekli da novi PKZIP nesumnjivo predstavlja veliki korak unapred – radi se o programu koji je po većini karakteristika znatno superioran svim postojećim PC arhivierima! Videćemo kada će i kako na ovaj izazov odgovoriti gospodin *Robert Jung*, ali nema sumnje da ovo takmičenje korisnicima donosi veliku korist. Ukratko, pružite šansu PKZIP-u 2.04c... barem do novog ARJ-a!

PREUZMITE SA SEZAMA

Arhiver pkzip 2.04c nalazi se na Sezamu pod imenom pkz204c.exe u direktorijumu r:\ibm-pc\archiver*, odakle ga mogu preuzeti svi pretplatnici našeg sistema za modemske komunikacije.



ПЕРИ®
ХАРД
ИНЖЕЊЕРИНГ

Ивана Милутиновића 24, Београд
Тел: 011/436 019, 432 319, 432 383
Факс: 011/435 513



ЗАШТИТИМО ЧОВЕКОВУ ОКОЛИНУ

ПОЈЕФТИНИТЕ

РАД НА

ЛАСЕРСКОМ

ШТАМПАЧУ



РЕЦИКЛИРАЈТЕ

ПРАЗНЕ

ТОНЕР КАСЕТЕ

ИЛИ НАМ ИХ

ПРОДАЈТЕ

**ИСПОРУКА СА СКЛАДИШТА
ТОНЕРА ЗА НР-ЛАСЕРСКЕ
ШТАМПАЧЕ НР-ІІР; НР-ІІІР;
НР-ІІ; НР-ІІІ; НР-ІІІД**

НОВО! НОВО! НОВО!



**од 1. марта рециклирамо
касете за НР-ЛЈ4**

РЕЦИКЛИРАМО

РИБОНЕ

ЗА ВЕЋИНУ МАТРИЧНИХ

И ЛИНИЈСКИХ ШТАМПАЧА

**НАЗОВИТЕ ДАНАС
ЗА ИНФОРМАЦИЈЕ!**



MI V A S

NE Ć E M O

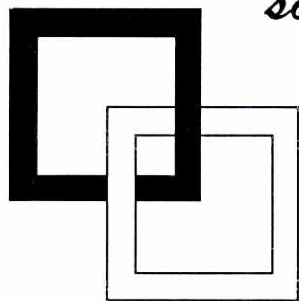
O S T A V I T I

N A C E D I L U

PANASONIC telefoni fax
EPSON printeri
HEWLWT PACKARD laserJet
BASF FUJI diskete
r a c u n a r i
STAEDTLER pera za ploter
ALGOR SUPER SAP
NOVELL mreze
DATATRONICS fax modemi
usluge projektovanja
softver za knjigovodstvo

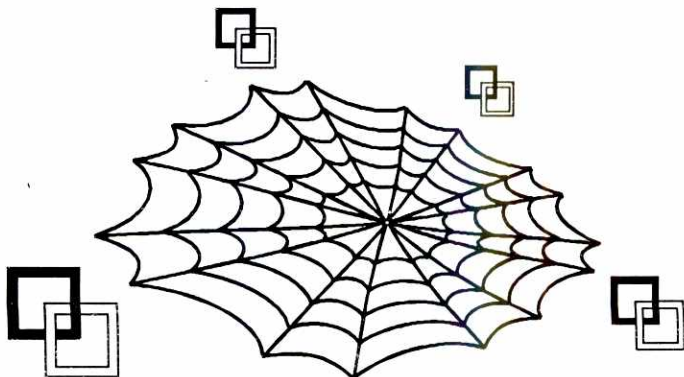
Vojvode Mišića 37
11 000 Beograd
tel (fax) 235 1 108
tel 650 522/602

software company



ProSoft

*Kragujevac * JNA 63 * tel/fax: 034/212-906*



"Povezivanjem Vaših računarskih ostrva možete dobiti jaku mrežu".

Direktor
Bratislav Uraković, dipl. ing.

BELEŽNICA ZA BIZNISMENE

Ako spadate u kategoriju za koju je računar alatka kao i svaka druga, a ne zbirka ovih ili onih tehničkih podataka, onda je notebook ACROS 325SE možda prava stvar za vas. U ovu elektronsku beležnicu smešteno je sve što je neophodno za ozbiljno korišćenje računara u poslu, svakodnevnom životu i – za najmoderniju komunikaciju sa svetom.

Ranka Jovanović

Uprkos malim dimenzijama (58×290×215 mm) i izuzetno laganom telu (6,3 funti – 2.8 kg), u ACROS je nabijena računarska snaga ozbiljnih stonih modela. Računar je izgrađen oko Am386SXL mikroprocesora na 25 MHz, a 4 MB radne memorije (proširivo do 8), brzi tvrdi disk od 2,5 inča kapaciteta 80 MB (postoje i modeli sa 120 MB), crno-beli VGA LCD displej 640×480 tačaka sa 16 sivih nijansi, pa čak i podnožje za matematički koprocesor predstavljaju solidne resurse čak i za najzahtevnije operativni sisteme (uključujući, naravno, nezaobilazni Windows, kao i sve popularniji OS/2) i aplikacije. Minijaturni prostor u koji su smeštene ove komponente nimalo ne utiče na procesnu snagu računara. Na uobičajenim brznim i aplikativnim testovima ACROS 325SE je pokazao, uključujući i video podsistem, rezultate u rang (procenat manje ili više) stonih mašina 386SX klase.



NEŠTO VIŠE

NICAD baterije obezbeđuju dovoljno energije za dva i po sata rada bez priključivanja na električnu mrežu, a za punjenje „do vrha“ im je potrebno pola sata manje. U štednji struje se vodi računa o svakom detalju – ako ključni pod sistemi nisu aktivni duže od 1/16 sekunde, brzina procesora se spušta na 1,64 MHz, tvrdi disk zamrzava aktivnosti a monitor se gasi... rečju, mašina pada u duboki san, iz koga se budi pritiskom na dugme.

ACROS je opremljen svim uobičajenim interfejsima za povezivanje sa periferijskim uređajima. Pored serijskog i paralelnog priključka, postoji i posebna PS/2 utičnica za miša, koji se takođe isporučuje u kompletu sa računarom, a spada, po lepoti, u izuzetnije primerke svoje vrste. I ne samo to. Obavezan deo opreme je i interni fax/modem 9600/2400 boda, čime je mašina usmerena na poslovno tržište. Ma gde se nalazili, možete, ako vam je pri ruci telefonski priključak, poslati pismenu poruku poslovnom partneru, ili se uključiti u svoju omiljenu mrežu za modemske komunikacije. Priključak za štampač omogućuje da na licu mesta odštampate ugovor ili protokol o poslovnom dogovoru. Uz ovakvu spravu, spremi ste za poslovni kontakt i poslovni odnos svuda i na svakom mestu.

U računare, pogotovo PC klonove, obično se ugrađuje samo ogoljeni DOS – tek toliko da se vidi da mašina radi. ACROS je znatno bogatije opremljen softverom – uz DOS, na računaru se nalaze i Windows 3.1 (koji radi veoma brzo i jako lepo izgleda na LCD displeju), komunikacioni alati za fax/modem i Prodigy softver za pristup istoimenom personalnom servisu, koji je namenjen potrebama porodice i u Americi je veoma popularan. Mi od toga ne možemo imati puno koristi, ali beležimo i taj podatak kao zanimljivu ideju. Zašto se ubuđuje ACROS, ili neka druga mašina, ne bi prodavao opremljen svim neophodnim alatcima i potpuno podešen za komunikaciju sa Sezamom ili – Berzom Studija B?

IZ MOG UGLA

Snaga mašine, naravno, nije jedina bitna komponenta za udoban rad. Tastatura sa 82 tastera, na kojoj nema mesta za svih deset prstiju, i LCD monitor su previše mali za dugotrajan rad. Pošto se mi nismo zadržali na prvim utiscima već smo na ACROS-u preveli umetak o Bilu Gejtsu, osetili smo u praksi ove nedostatke. Zaista, za onog ko inače intenzivno piše na PC računaru, problem je prelazak sa standardne na notebook tastaturu, kako zbog dimenzija tako i

zbog promenjenog rasporeda, pa i upotrebe nekih tastera (pre svega funkcijskih). U našem slučaju je bilo dosta lutanja i nehotičnih grešaka u kucanju, što je usporilo rad i zahtevalo posebno pažljivu redakturu teksta.

Prikaz je za potrebe pisanja teksta sasvim zadovoljavajući. Naravno, s obzirom na dimenzije i kvalitet notebook ekrana, teže ide ispravljanje i redigovanje teksta, što je u našem slučaju jedan od osnovnih poslova. Kontrast je slabiji, kursor je manje primetan, obeleženi delovi (highlighted) su manje uočljivi u odnosu na ostatak teksta, što onemogućava tzv. „redlining“, odnosno unošenje korekcija „crvenom olovkom“. Zato smo pribegli posebnoj raspodeli poslova – kod kuće pisali prevod, čuvajući se brzopletog lupanja po tasterima (što nam, na žalost, nije u potpunosti uspelo), a posle na stonom računaru obavljali redakturu.

DVA LICA

Istini za volju, ovakvi poslovi su stavili ACROS na preterano teško iskušenje. Jer, notebook mašine i nisu namenjene za težak fizički rad na računaru, već prvenstveno vlasnicima stonih kompjutera koji ni u jednoj prilici, pa ni onda kada su u pokretu, ne žele da menjaju način rada, radne navike i poslovne običaje. Zahvaljujući jednostavnim dodacima, zamenom najvitalnijih delova, ACROS se lako može preobraziti u stonu mašinu. Računar ima priključak za spoljašnji VGA monitor i spoljašnju PS/2 (a preko posebnog adaptera i AT) tastaturu. Ovako komponovan, ACROS se ni po čemu ne razlikuje od stonog računara, uz jednu značajnu prednost – uvek ga možete otkačiti, staviti u tašnu, čak i za dame, i poneti sa sobom na putovanje ili – poslovni sastanak.

Lepi utisci pri prvom susretu s ACROS-om nisu nimalo pokvareni teškoćama na koje smo naišli u prilogodavanju. Ta spravica koja „sve svoje sa sobom nosi“ pruža korisniku veliko zadovoljstvo u radu i nudi pouzdanu pratnju bukvalno svuda i na svakom mestu.

Računar se prodaje po ceni od 3500 DEM.

Korisna adresa

Berza Studija B
11000 Beograd, Masarikova 5
Tel: 333-333, fax: 685-866.

HT 340 osnovna ploča sa HT216 VGA kontrolerom 61% brža od WD 90C30 u Windows okruženju.

BABA + ŽABA = Winmark?

Test Winbench daje veliku tabelu sa preko 120 rezultata pojedinačnih testova. Za obične smrtnike neupotrebljivo, za stručnjake tvrd orah za analizu i međusobno poređenje. Testovi su vrlo raznorodni i gotovo je nemoguće sabrati ove rezultate u jedan broj. Ipak, u novoj verziji programa Winbench, brzina kartice se označava sa Winmark i izražava u pikselima u sekundi. Da bi se istakle najbitnije grafičke operacije, vrši se ponderisanje rezultata. Pre toga je obavljena obimna analiza najčešće korišćenih programa da bi se odredili koeficijenti. Mada ovaj pristup nije savršen i može mu se uputiti dosta primedbi, za sada je jedini upotrebljiv. Naročito se može mnogo polemisati o koeficijentima, ali teško je predložiti neke druge brojeve.

Nasuprot ovom pristupu D.V. Veselinović svoj rezultat dobija kao aritmetičku sredinu relativnih odnosa rezultata komponente koja se testira i rezultata referentnog sistema. Na taj način, svi testovi dobijaju jednaku težinu u konačnom rezultatu, ma kolika bila verovatnoća njihovog pojavljivanja u programima. Vrednost ovakvog rezultata ne treba komentarisati.

ZAKLJUČAK

Stvar je dobrog vaspitanja da IMTEL ne hvali svoje proizvode, tako da ni ja to neću učiniti ovom prilikom. Naša želja nikad nije ni bila da ponudimo kupcima apsolutno najbrži ili najskuplji model, već onaj koji će im za njihov novac doneti najviše performansi. Na taj način treba posmatrati i ovu osnovnu ploču sa local-bus tehnikom. To nije najbrža ploča, ali svakako nije ni najskuplja. Postoje primene za koje je dobra, a to su prvenstveno sve GUI aplikacije.

Oni koji žele da pročitaju opširniju i bolje dokumentovanu verziju ovog teksta, mogu je potražiti na SEZAM-u ili IMTEL BBS-u.

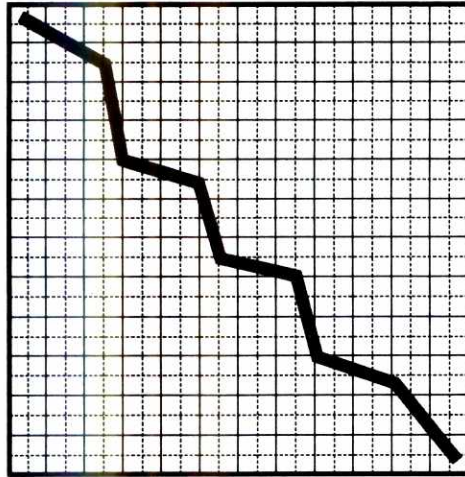
Literatura

1. NEC: Memory Products Data Book.
2. Intel: i80486 Microprocessor.
3. E. Solari: AT bus design IEEE P996 compatible.
4. R. Ferraro: Programmer's Guide to the EGA and VGA Cards.

Jovana Ristića 6
11000 Beograd
tel/fax: 011 64 71 90

besplatni programi za PC računare
otkup polovne PC računarske opreme
prodaja PC računara staro za novo
prodaja repariranih PC računara
servis PC računara i opreme
diskete, riboni, papir,.....





Promene su moguće...
Dovoljno je samo da okrenete list!

Investirajte u **BOLJE SUTRA!**
Pozovite nas. Rado ćemo vam pomoći.



Zaronite u svet knjiga
i mikroracunara

onoliko duboko koliko želite

ABC Serija

ABC DOS-a 5, A. Neibauer, 336 str.
ABC Lotus-a 1-2-3, C. Gilbert, 320 str.
ABC programa Quattro Pro 2,
A. Simpson, 290 str.
ABC programa Word za Windows 1.1,
A. Neibauer, 300 str.
ABC programa Word Perfect 5.1,
A. Neibauer, 336 str.
ABC programa Windows 3.0,
Kris Jamsa, 280 str.

CorelDRAW 2, S. Rimmer, 400 str.
IBM PC Uvod u rad, DOS, BASIC,
S. Milinković, 400 str.

Priručnik dBASE III PLUS,
D. Tanaskoski, 380 str.

CLIPPER 5 vodič za programere,
R. Spence, 540 str.

Programiranje na Clipper-u,
S. Straley, 780 str.

Programiranje i programski jezici

Programiranje na jeziku C,
A. Hansen, 300 str.

Programski jezik C++,
B. Stroustrup, 300 str.

Turbo Pascal 6, S. O'Brien, 440 str.
Pascal priručnik, K. Jensen, 260 str.

Programiranje na jeziku Modula-2,
N. Wirth, 190 str.

TEX za nestrpljive,
P. Abrahams, 350 str.

100 najkorisnijih FORTRAN-skih
potprograma, I. Mendaš, 380 str.

Ostala izdanja

Llinearna integrisana kola,
B. Raković, 500 str.

Elektronski hobi uređaji,
D. Milačić, 110 str.

Elektronske konstrukcije sa 555,
D. Milačić, 120 str.

Arterije mozga, S. Marinković,
Commodore za sva vremena,
D. Tanaskoski, 330 str.

Spektrum priručnik,
V. Janković, 260 str.

Finalna prerada drveta, D. Skakić,

Mikro knjiga

Petra Martinovića 6
11030 Beograd
Tel.: 542 516 i 542 619

COMPUTERS

Preduzeće „MZ“, 11000 Beograd,
Dubljanska br. 70. Tel: 011/434-812.
Fax: 011 450 471

ŠTAMPAČI EPSON

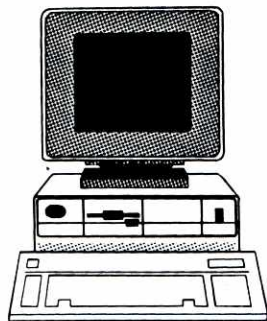
LX-400, 9-pin. A4 format, 180 z/s, ćir/lat
LX-100, 9-pin. A4 format, 240 z/s, CSF, YU-set
LX-850, 9-pin. A4 format, 200 z/s, YU-set
FX-870, 9-pin. A4 format, 380 z/s, YU-set
LX-1050, 9-pin. A3 format, 200 z/s, YU-set
FX-1050, 9-pin. A3 format, 300 z/s, ćir/lat
LQ-100, 24-pin. A4 format, 200 z/s, CSF ćir/lat, ESC P-2
LQ-570, 24-pin. A4 format, 240 z/s, YU-set, ESC P-2
LQ-870, 24-pin. A4 format, 330 z/s, YU-set, ESC P-2
LQ-860, 24-pin. A4 format, 300 z/s, YU-set, kolor
LQ-1070, 24-pin. A3 format, 240 z/s, YU-set, ESC P-2
LQ-1170, 24-pin. A3 format, 330 z/s, YU-set, ESC P-2
LQ-1060, 24-pin. A3 format, 300 z/s, YU-set, kolor
DLQ-2000, 24-pin. A3 format, 270 z/s, YU-set, kolor
DFX-5000, 9-pin. A3 format, 533 z/s, YU-set
DFX-8000, 18-pin. A3 format, 1000 z/s, YU-set
SQ-870, 48-mlaz. A4 format, 660 z/s, YU-set, INK JET
SQ-1170, 48-mlaz. A3 format, 660 z/s YU-set, INK JET
EPL-4000, LASER, 300 DPI, 6 s/m, 1 MB, H-YU-set, (opcija)
EPL-1300, LASER, 300 DPI, 6 s/m, 1 MB, RITech, H-YU-set,
(opcija)
EPL-8100, LASER, 300 DPI, 10 s/m, 2 MB, RIT, POL 5, H-
YU-set, (opcija)
EPL-7500, POSTSCRIPT LASER MEGA2, 300 DPI, 6 s/m, 2 MB,
GT-8000, COLOR IMAGE SCANNER, A4 format, 800 DPI

Roland
DIGITAL GROUP

DXY-1100, A3 format, 8 pera, 42 cm/s, mag. držanje papira
DXY-1200, A3 format, 8 pera, 42 cm/s, ele. statičko papira
DXY-1300, 1MB, A3 format, 8 pera, 42 cm/s, ele. statičko pa-
pira
SKETCH MATE, A3 format, 8 pera
SKETCH MATE, A4 format, 8 pera
DPX-2500, A2, ploter-tabla, 8 pera, 62 cm/s, 1 MB
DPX-3500, A1, ploter-tabla, 8 pera, 62 cm/s, 1 MB
GSX-3000, A1, „ROLL“-ploter, 8 pera, 113 cm/s, 1 MB, aut.
p.v.p.
GRX-300 AG, A1, „ROLL“-ploter, 8 pera, 66 cm/s,
DPX-4600, A0, ploter-tabla, 8 pera, 62 cm/s, 1 MB
GSX-4000, A0, „ROLL“-ploter, 8 pera, 113 cm/s, 1 MB, aut.
p.v.p.
GRX-400 AG, A0, „ROLL“-ploter, 8 pera, 66 cm/s,
LTX-100, A3 format
LTX-120, A3 format
LTX-321, A1 format
LTX-420, A0 format
Data buffer SYA-550
CAMM-1 PNC-1100 PLOTER-REZAČ, format od 50 do
640 mm (do 1600 mm)
CAMM-1 PNC-1800 PLOTER-REZAČ, A0 format (do
3600 mm)
STIKA-Rezač sa ugrađenim skenerom, sa RS232C i progra-
mom

EIZO[®]

EIZO 9080i 16" kolor monitor, 1024x768
EIZO 6500 21" monohromatski monitor, 1664x1200 (60Hz)
1280x1024
EIZO F550i 17" kolor mon, ravan ekran, max. 1280x1024
EIZO F750i 21" kolor mon. ravan ekran, max. 1280x1024
EIZO T560i 17" kolor (TRINITRON CRT), 1280x1024, CRT
Trio p. 0,31
EIZO T660i 20" kolor (TRINITRON CRT), 1280x1024,
EIZO GRAFIČKE KARTICE — VA30, VA41, AA51, MD B09,
MD B10, MD B12



PREDUZEĆE ZA INFORMATIČKI INŽINJERING I TRGOVINU

SOFT PROJEKT

VRANESI 36215 PODUNAVCI
Tel. 036 61-847 Fax. 036 67-271

SDK VRNJAČKA BANJA
Ž.R. 61820-601-9581

1. MODEL SP286/16 MHz	1199	EPSON LX400/800	520
2. MODEL SP386SX/25 MHz	1349	EPSON LX810	570
3. MODEL SP386/33 MHz	1799	EPSON FX850	1199
4. MODEL SP386/40 MHz	1899	EPSON FX1050	1299
5. MODEL SP486/33 MHz	2799	EPSON LQ570	1199
6. MODEL SP486/50 MHz	3099	EPSON LQ1050	1799
		EPSON DFX5000	4999
		Centroniks kabl	25

Osnovne konfiguracije sadrže:

40 MB HDD, 1.2 FDD, tastaturu, hercules 14' crno beli monitor, model 286 i 386 1MB RAM-a, 386 4MB i 64KB cache, 486 4MB RAM-a i 256 cache

DOPLATE:

HARD DISK 80MB	170
HARD DISK 105MB	220
HARD DISK 120	280
HARD DISK 200MB	850
VGA MONO MONITOR	60
VGA COLOR MONITOR	470
VGA card 512KB	80
VGA card 1MB	160
Drugi FDD	110

Kompjuterski papir:

240×12" 1+0	380×12" 1+0
240×12" 1+1	380×12" 1+1
240×12" 1+2	

PANASONIC ŠTAMPAČI LASERSKI ŠTAMPAČ HPIII LASERSKI ŠTAMPAČ HPIII P Desk Jet HP500c

TONER ZA HPIII	230
MOUSE GENIUS	60
MOUSE SEPOM	50
FILTER 14" MREŽASTI	40
FILTER 14" STAKLENI	65
RIBONI ZA EPSON A4	15
RIBONI ZA EPSON A3	25
DISKETE 5.25" DD	20
DISKETE 5.25" HD	30
DISKETE 3.5" DD	25
DISKETE 3.5" HD	35

UZ SVAKI RAČUNAR JEDAN BESPLATAN PROGRAM ZA KNJIGOVODSTVO PREDUZEĆA ILI RADNJE SA UPUTSTVOM ZA KORIŠĆENJE RAČUNARA.

III

CENE SU U DEM. A UPLATE SE VRŠE U DINARSKOJ PROTIVVREDNOSTI PO DNEVNOM KURSU NA DEVIZNOM TRŽIŠTU, ISPORUKA JE ODMAH PO UPLATI, GARANCIJA 12 MESECI, SERVIS OBEZBEĐEN.

PRVA LJUBAV, DRUGI PUT

Iako ništa ne može zameniti dobar kompajler, postoji još mnogo „uslužnih delatnosti” čiji se nedostatak kod pisanja fortranških programa i te kako osećao. Rekompilacija starih kodova ili renesansa na *plusFort* način.

Saša Vlajić

Fortran, veteran na sceni viših programskih jezika, sa svim svojim vrlinama i manama, ostavio je dubok trag kod mnogih generacija naučnika i inženjera. Veliki broj njih će reći da je on njihova prva programerska ljubav. Sadržaji hiljada i hiljada fortranških programa i biblioteka, razvijenih na nekadašnjim hipopotamskim *mainframe* računarima, njihovi algoritmi i podaci, predstavljaju pravo zakopano blago. Stalna potreba za više verzija jednog te istog programa, kako bi se omogućilo izvršavanje na različitim sistemima, PC ili *mainframe*, stvara još uvek prilične komplikacije.

Sagledavajući sve ovo, kao i mnoge druge probleme u vezi s fortranom, programeri sa Salford univerziteta, Velika Britanija, inače specijalisti za ovaj jezik, zajedno sa firmom „Polyhedron Software”, odlučili su da mu obezbede dostojno mesto i na početku devedesetih, novim paketom *plusFORT*. U redovima koji slede, biće dat kratak prikaz svih pet programa iz ovog interesantnog paketa.

SPAG – „RAZMRSIVAČ ŠPAGETI KODA”

Zadatak postavljen pred *plusFORT* je, u osnovi, priprema terena za neki od kompajlera koji se tiču standarda ANSI FORTRAN 77. Najteži dio posla poveren je programu SPAG, inače glavnom delu paketa. Radi se, pre svega, o „podmlađivanju” starih programa pisanih u FORTRAN-u 66. Imajući u vidu da stari izvorni kodovi (*source codes*) često deluju prilično odbojno, zapetljani su i gotovo nerazumljivi, kreatori su SPAG nazvali „razmrsivač špageta” (*spaghetti unscrambler*).

Pod „podmlađivanjem” se podrazumeva analiziranje mnoštva isprepletenih GOTO, aritmetičkih IF i ostalih naredbi karakterističnih za stariji tip fortrana (slika 1a), te formiranje novog izvornog koda pomoću konstrukcija modernog strukturnog programiranja. Logički srodni delovi se grupišu, a broj skokova minimizira. Takav kod je logički potpuno ekvivalentan ulaznom „špageti kodu”. Znači, algoritam, sadržaj, ostaje nepromenjen, pa nekad čak i redosled izvršavanja naredbi. Ono što trpi izmenu je samo forma, zapis, tako da je razumljivost i jednostavnost održavanja sada znatno unapređena.

Stari programi često sadrže „mrtve linije” (*dead code*), koje se nikad ne izvršavaju. SPAG je u stanju da ih raspozna i, po želji, ukloni. Kad smo već kod programskih linija, treba reći da ovaj program ima tzv. „ulepšivač koda” (*code beautifier*) sa mnoštvom opcija za kontrolu labela, uvlačenja retka, razmaka, malog/velikog slova, naredbe CONTINUE, i slično. Ovo omogućuje nosiocima projekata definisanje i podržavanje internih stilova pisanja programa.

Ponekad preuređivanje originalnog izvornog koda, bez ikakvog dodavanja, nije dovoljno efikasno. U FORTRAN-u 66 obično su programi ispadali složeniji u težnji da se višestruko iskoristi nekoliko bajtova koda, npr. da se izbegne stavljanje „N=N+1” više od jednog puta. SPAG ovo ispravlja, ubacivanjem malih kodnih fragmenata na mestima gde će unapređenje restrukuiranog koda biti osetno (slika 1b).

Vrlo važna osobina SPAG-a je što dozvoljava upotrebu mnogih programskih ekstenzija karakterističnih za VAX i FORTRAN 90 (DO WHILE, DO ENDO, SELECT CASE, itd.), te, ukoliko je potrebno, konverziju u obrnutom smeru – u standardan ANSI FORTRAN 77.

Ono što vredi naglasiti je izvanredna jednostavnost upotrebe. Ulazni izvorni kod ne zahteva nikakvu posebnu pripremu. Izvršavanje SPAG-a se, tamo gde je to neophodno, može preusmeriti pomoću komandne linije ili editovanjem ASCII konfiguracijskog fajla. Što se tiče brzine, ovaj program je ispred mnogih kompajlera, a očuvana je pri tom visoka pouzdanost. Desetine miliona dosad obrađenih programskih redo-

va, „špageti koda” to najbolje dokazuju. SPAG se nudi u kompletu sa ostalim programima paketa *plusFORT* ili zasebno. Postoje verzije za različite platforme: IBM PC, DEC VAX, HP, Prime, IBM MVS ili VM, Sun, i druge. Svaka od njih oblikovana je tako da prihvata najčešće sve fortranške dijalekte raspoložive za upotrebu na datom hardveru. Tako, na primer, IBM PC verzija razume sve PC fortrane.

QMERGE – SELEKTOR VERZIJE

Na početku je istaknuto da je pisanje više verzija jednog programa, kako bi se obezbedila podrška raz-

```

13 IF (TODO.EQ.0) GO TO 12
   ACT=MOD (TODO,10)
   TODO=TODO/10
   GO TO
   (62,42,43,62,404,45,62,62,62),ACT
   GO TO 13
42 CALL COPY
   GO TO 127
43 CALL MOVE
   GO TO 144
404      N=-N
44 CALL DELETE
   GO TO 127
45 CALL PRINT
   GO TO 144
62 CALL BACTACT (ACT)
   GO TO 12
127      LENGTH=LENGTH+N
144      DONE=DONE+1
      CALL RESYNC
   GO TO 13
12 RETURN
   END
  
```

Slika 1a: Originalni kod u FORTRANU 66 (izdvajanje decimalnih cifara iz celog broja i grananje u zavisnosti od njihove vrednosti)

```

100 IF ( TODO.NE.0 ) THEN
   ACT = MOD (TODO,10)
   TODO = TODO/10
   IF ( ACT.EQ.1 .OR. ACT.EQ.4 .OR.
     & ACT.EQ.7 .OR. ACT.EQ.8 .OR.
     & ACT.EQ.9 ) THEN
     CALL BACTACT (ACT)
     GO TO 200
   ELSEIF ( ACT.EQ.2 ) THEN
     CALL COPY
     LENGTH = LENGTH + N
   ELSEIF ( ACT.EQ.3 ) THEN
     CALL MOVE
   ELSEIF ( ACT.EQ.5 ) THEN
     N = -N
     CALL DELETE
     LENGTH = LENGTH + N
   ELSEIF ( ACT.EQ.6 ) THEN
     CALL PRINT
   ELSE
     GO TO 100
   ENDIF
   DONE = DONE + 1
   CALL RESYNC
   GO TO 100
200 RETURN
   END
  
```

Slika 1b: Verzija koda u FORTRAN-u 77 (SPAG je umetnuo LENGTH = LENGTH N)

ličnih platformi, do sada bilo nužno zlo. U zavisnosti sebnost. Postoje verzije za različite platforme: IBM PC, od stepena složenosti programa, kontrola unošenja izmena je iziskivala više ili manje truda: različita pravila za dužinu reči, pristupanje sistemu, davanje imena fajlovima... čak i određivanje vremena i datuma nije jedinstveno za sve verzije. No, ono što je najviše nerviralo programere bilo je upravo saznanje da gotovo ni najjednostavnija aplikacija nije univerzalna, u smislu korištenja na različitim sistemima, pa je za n sistemsku potrebno kontrolisati n varijanti programa.

Rešenje ovog problema *plusFORT* nudi u vidu *QMERGE*-a.

Kod javljanja aplikacije, korisnik nema potrebu da ni na koji način odstupa od standardnog ANSI FORTRAN-a 77, ili da vrši pretprocesiranje koda pre kompajliranja. U okviru izvornog koda mogu se držati sve kodne varijante, a određeni njegovi delovi biće, po startanju *QMERGE*-a, stavljeni pod znak komentara, već prema tome kako se postavje logički flegovi.

Ilustriramo ovo primerom. Na slikama 2a i 2b vidi se koje se promene dešavaju na jednom kodnom segmentu kod prebacivanja između VAX i UNIX verzija.

Sve što je potrebno kod prenosa sa jednog na drugi tip jeste startovanje *QMERGE*-a i davanje adekvatne komande u komandnoj liniji. Tako,

```
" SELECT=UNIX,- VAX "
```

znači da se želi UNIX verzija izvornog koda. U slučaju da ovo ne navedemo, program će sam pitati

```
" Is UNIX true? (Y or N) "
```

Nepotrebne delove koda je, umesto stavljanja pod komentar, moguće i potpuno odstraniti, ali u tom slučaju nema više povratka na tu verziju programa.

CRAM – AUTOMATSKO DOKUMENTOVANJE

Bez statističkog prikaza određenih podataka danas se ne može zamisliti nijedan iole ozbiljniji softver. Tu značajnu dimenziju paketa *plusFORT* obezbeđuje *CRAM*, program koji, nakon analize našeg izvornog fortranškog koda, generiše niz izveštaja o njegovoj strukturi (slika3). Oni daju pregled odozgo nadole (npr. odakle se sve poziva određeni potprogram tipa SUBROUTINE, koje CALL naredbe izdaje svaki potprogram...), programsko stablo (opšta struktura u obliku dijagrama) i sl. Naznačavaju se rekurzivni pozivi (npr. A poziva, B poziva A) i navode se rutine u rekurzivnoj petlji. Identifikuju se nerešene sekvence, a i pobliže ukazuju na određene greške. Navedimo, tako, primer gde CALL ima različit broj argumenata od odgovarajuće naredbe SUBROUTINE – tada *CRAM* eksplicitno navodi koji argumenti su izostavljeni.

Vrlo atraktivna je primena *CRAM*-a i kao dopuna linkeru *PlinkB6*, istog proizvođača. Uz njegovu asistenciju, odnosno precizan opis koraka za rad linkera (slika 4), sa *mainframe* sistema skinuti su neki od najvećih fortranških programa ikada primenjenih na računarima zasnovanim na mikroprocesoru 8088. Program sa 110.000 linija izvornog koda, koji kompajliran i linkovan čini .EXE fajl veličine 4 MB, uspešno se izvršava sa samo 512K.

AUTOMATSKO REKOMPILIRANJE

Da li vam se ikada desilo da posle niza izmena zaboravite koje ste fajlove ažurirali, a koje ne? Da se takve zgode ne bi ponavljale, *plusFORT* ima efikasan lek – *CMAKE* program. Jednostavnim unošenjem naredbe „C”, kontrolise se čitav direktorij sa izvornim kodovima, od kojih su neki modifikovani, a neki ne. *CMAKE* tad automatski rekompajlira minimalni set izvornih kodova i ažurira sve fajlove koji su s njima u vezi (kontrolišu se i INCLUDE naredbe). INCLUDE

```

DO WHILE ( todo.NE.0 \
act = mod(todo,10)
todo = todo/10
SELECT CASE (act)
CASE (1,4,7,8,9)
CALL badact(act)
EXIT
CASE (2)
CALL copy
length = lengt + n
CASE (3)
CALL move
CASE (5)
n = -n
CALL delete
length = length + n
CASE (6)
CALL print
CASE DEFAULT
CYCLE
END SELECT
done = done + 1
CALL resync
END
RETURN
END

```

Slika 1c: Verzija koda u FORTRAN-u 90

```

*-IF VAX
PRINT *, 'Digital VAX version 1.1'
file = '[users.john]progl.for'
*-ELSEIF UNIX,XENIX
*-PRINT *, 'UNIX version 1.1'
file = '/usr/john/source/progl.f'
*-ENDIF

```

Slika 2a: QMERGE getnam.for SELECT=VAX

```

*-IF VAX
*-PRINT *, 'Digital VAX version 1.1'
file = '[users.john]progl.for'
*-ELSEIF UNIX,XENIX
PRINT *, 'UNIX version 1.1'
file = '/usr/john/source/progl.f'
*-ENDIF

```

Slika 2b: QMERGE getnam.for SELECT=UNIX,-VAX

fajlovi ne moraju biti u jednom direktoriju. Ime *default* kompajlera i eventualni unos opcija putem komandne linije su sve što se traži od korisnika. Te informacije se smeštaju u kontrolni *batch* fajl pri instaliranju *CMAKE*-a. Ovaj program, inače, ima vlastitu bazu podataka o uzajamno povezanim fajlovima, što mu obezbeđuje jednostavnost i pouzdanost.

GXCHK – JOŠ MALO KONTROLE

S ciljem otkrivanja još nekih grešaka u programima, za čim se pokazala potreba u prvoj fazi ispitivanja *plusFORT*-a, dodat mu je naknadno i *GXCHK* - *global cross checker*. On od *SPAG*-a uzima tabele simbola, kako bi formirao globalni pregled korištenja podataka unutar programa. Možda je to najbolje objasniti pomoću analogije sa kompajliranjem i linkovanjem. Tabele simbola su tada analogne objektnim fajlovima, a *GXCHK* linkeru koji ih kombinuje u jedinstven globalni pregled.

GXCHK otkriva su, između ostalog, i sledeće greške:

- ispušteni ili više puta definisani potprogrami
- nekorišteni potprogrami, *INCLUDE* fajlovi, *COMMON* promenljive i *COMMON* blokovi
- *COMMON* promenljive bez dodeljene vrednosti
- *COMMON* ime promenljive u više nego jednom *COMMON* bloku
- nedoslednosti kod definisanja *COMMON* bloka ili liste argumenata u potprogramu
- *COMMON* blok ili *PARAMETER* definisan u više nego jednom *INCLUDE* fajlu

```

PWMOVE is unresolved
called from SCGOTO (line 81 of
SCGOTO.FOR)
called from SWFRAG (line 23 of

```

Slika 3: Deo izveštaja o strukturi programa

```

BEGIN
SECTION FILE CSDATA,SMAIN
END
BEGIN
SECTION FILE SCGOTO,SFLABL
SECTION FILE SGOTOA
SECTION FILE SINDNT
SECTION FILE SRELAB
SECTION FILE SRELOC
SECTION FILE SRFIG
SECTION FILE SRPROG,SLINK
SECTION FILE STIDY
SECTION FILE SWPROG
END
BEGIN
SECTION FILE SCDEST
SECTION FILE SCINT
SECTION FILE SWSTUN
BEGIN
SECTION FILE SWINIT
SECTION FILE SWSTAT
BEGIN
SECTION FILE SFBRAC
SECTION FILE SWFRAG,SFNUM
SECTION FILE SFCHAR
SECTION FILE SWLABL
END
END
BEGIN
SECTION FILE SWTOKN
END

```

Slika 4: CRAM daje uputstva za rad linkera PLINK86

Ovaj program daje i tabelu koja pokazuje koji su sve simboli upotrebljeni, gde i kako. Ona je od koristi kod utvrđivanja potprograma koji utiču na promenu određene *COMMON* promenljive.

UMESTO ZAKLJUČKA

Navedimo, na koncu, zanimljiv rezultat istraživanja jednog američkog instituta. Po njemu, cena razvoja i održavanja softvera, u toku njegovog veka, iznosi od 50 do 100\$ po programskoj liniji. Uzmu li se u kalkulaciju milioni linija koje su u upotrebi, jasno je zašto se neki stručnjaci žale da samo održavanje softvera proguta više od 2% bruto nacionalnog dohotka u SAD. *plusFORT*, hvale se njegovi kreatori, štiti vašu investiciju. Na stranu to što više neće biti mesta personalizaciji sa Sizifom ili Tantalom, što kod rada u fortranu dosad nije bio redak slučaj.

Lična karta

Programski paket: *plusFORT* – programs for programmers

Sadržaj paketa: kopija softvera na jednoj 3.5" DD disketi, dokumentacija (jedna knjiga)

Cena: za XT/AT PC – 0 890 (godišnje održavanje – 0 100); za 386 PC – 0 1390 (godišnje održavanje – 0 160)

Programi iz paketa se mogu nabaviti i pojedinačno, a dobija se i popust u slučaju da se kupuje više primeraka. Popust se, takođe, odobrava za nekomercijalnu i obrazovnu upotrebu.

Prostor na disku: 1Mb

Adresa proizvođača: Polyhedron Software Limited, Linden House, 93 High Street, Standlake, Witney, Oxon OX 7RH, U.K., tel: (+44) 0865 300 579

KOMPRESIJA PODATAKA

KAKO SMRVITI SLIKU

Nova tehnika kompresije podataka, nazvana *Fractal Transform*, sažima grafiku u boji u odnosu 50:1, ali nije sve ni u uštedi prostora.

Ukoliko nemate previše vremena i prostora na hard disku, korišćenje slika sa velikom rezolucijom i punom paletom boja nije nimalo jednostavna stvar. Datoteke su prevelike za rad, a tradicionalne tehnike za kompresiju podataka koje rade sa tekstom ne snalaze se sa kompleksnošću kolor grafike baš najbolje. *Images Incorporated, Version 2.0*, (čija je cena \$299) je prvi program za kompresiju koji problem rešava pomoću metoda razlamanja.

Ovaj program, baziran na *Windows*-u, koristi šemu za kompresovanje nazvanu *Fractal Transform* da bi zdrobio grafičke slike u relaciji 50:1 i da bi ih povećao na potpuno nov način. Razlomna kompresija postiže ove impresivne osobine tako što najpre identifikuje sadržaj, zatim sadržaj čuva u obliku matematičkih jednačina u *FIF* datoteci (*Fractal Image Format* koji je definisan u *Iterated Systems*), a onda rekonstruiše sliku pomoću iterativnog matematičkog modela. Program koristi softversku kompresionu mašinu nazvanu *POEM*, koja „gnječi” pratećih 192 MB divnih 24-bitnih kolor fotografija i ilustracija u samo 3 MB prostora na disku!

Što se hardverskih zahteva tiče, treba imati 386 ili viši PC, 4MB RAM-a, 3,5MB prostora na hard disku, 4MB permanentne *swap* datoteke, *Microsoft Windows 3.0* ili noviji (preporučuje se 8MB RAM-a, 10MB HD prostora i *Windows* kompatibilni SVGA adapter).

Pošto razlomna kompresija generiše veoma male, rezolucijski nezavisne datoteke, predstavlja relativno novu šemu u poređenju sa *JPEG* (*Joint Photographic Experts Group*) standardom, koji koristi piksel-bazirani *DCT* (*Discrete Cosine Transform*) metod da bi se oslobodio suvišnih bitova. Dodavanjem *JPEG* metoda kompresovanja u verziju 2.0, firma *Iterated Systems* je ovom programu otvorila put za najširu upotrebu.

Veliki proboj predstavlja *Fractal Transform Resolution Enhancement*, tehnika skaliranja koja povećava rezoluciju slike. Iz *Expand* opcije birate faktor uvećanja (2, 4, 8 ili 16) i metod (brza ili kvalitetna ekspanzija razlomljenih delova ili replikacija piksela – na žalost, ova dva pojma su veoma isključiva u ovom slučaju).

U toku ekspanzije, program dodaje detalje interpolacijom vrednosti piksela, a ne jednostavnom duplikacijom. Rezultujuća slika je oštija od slike dobijene replikacijom piksela. Možete, zatim sačuvati povećanu sliku u *Clipboard*-u ili je konvertovati u neki od podržanih formata (*BMP*, *PCX*, *raster*, *TGA* ili *TIFF* u nekoliko dubina boja) za dalje doterivanje u nekom programu za obradu slike, za korišćenje u stonom izdavaštvu ili prezentacionoj aplikaciji.

Razlomna kompresija je asimetrični metod, što znači da brže dekompresuje slike nego što ih kompresuje. *JPEG* je simetrični metod koji za otprilike isto vreme izvrše kompresiju i dekompresiju. Tako za aplikacije kod kojih je brzina kritična, kao što su multimedijске prezentacije, razlomna kompresija predstavlja, svakako, najbolje rešenje.

Prilikom testiranja ova dva metoda kompresovanja, na računaru *Compaq Deskpro 386/25* sa 8 MB RAM-a, dobijeni su sledeći rezultati:

Bilo je potrebno gotovo 4 minuta da se 1.3 MB *TIFF* datoteka konvertuje u 228K *JPG* datoteku, koristeći *JPEG* kompresiju, i dvostruko više vremena da se ista datoteka konvertuje u 148K *FIF* datoteku. Dakle, *JPEG* je brži, ali *Fractal Transform* pravi manje datoteke (50:1 u poređenju sa 20:1 kod *JPEG* metoda). *Fractal* metod je brži kod dekompresije – potrebno mu je 25 sekundi da prikaže *FIF* datoteku, dok je za prikaz *JPG* datoteke potrebno čitavih 2 minuta.

Glavna mana uvećavanja kod razlomnog metoda je trošenje vremena. Potrebno je 10 minuta da bi se duplirala 270K *PCX* slika, 51 minut da bi se učestvostručila, i gotovo 15 sati da bi se uvećala 16 puta. Ovo je delimično rešeno novim, brzim metodom i *batch* tehnikom za uvećavanje i konvertovanje slika „preko noći”. Takođe, s obzirom da se kod viših rezolucija žrtvuje oština, uvećavanje razlomnim metodom nije idealno za visoko ambiciozan rad sa grafikom. Obradeni predmeti nisu tačkasti, ali oštre ivice postaju zamagljene.

Prednosti razlomne kompresije, tako, nisu baš ubedljive.

M. Jolić

RASTEGLJIVI DISKOVI

Verovatno nema korisnika računara koji ne želi da ima barem dvostruko veći disk. Možda postoji (dovoljno jeftino) rešenje i za te želje?

Stacker je programski paket koji se veoma mnogo koristi baš u našim krajevima. Pošto do sada nismo imali nijedan tekst o ovako eksploitantom programu, potrudili smo se da napravimo ne samo prikaz paketa, nego i kratko uputstvo za korišćenje...

Šta je *Stacker*? To je *device driver*, program koji kontroliše komunikaciju između računara i diska, komprimujući i dekomprimujući podatke u toku rada („u trku“). Rezultat njegovog rada je povećanje kapaciteta diskova (uglavnom hard-diska). Koliko će kapacitet biti povećan, zavisi uglavnom od podataka koji se na disku nalaze, ali se povećanje obično kreće od 0 do 100 posto. Samim *Stackerom* ćemo se pozabaviti tek pošto se podsetimo kakva su sve rešenja za povećanje kapaciteta diskova do sada nuda.

MALO ISTORIJE...

Hard-diskovi su do pre nekoliko godina koristili samo dve tehnologije smeštanja podataka na magnetnu ploču, pa smo ih mogli podeliti na **mfm** i na **rll** diskove. Nećemo se previše baviti samim načinima kodiranja, ali nam je interesantna razlika između ova dva standarda: **mfm** diskovi sadrže manje podataka na istoj površini od **rll** diskova, a iz toga slede sve ostale razlike.

Prva značajnija (i primetna od strane korisnika) razlika je u brzini – **rll** diskovi (čije su informacije gušće pakovane) su po pravilu brži od **mfm** diskova. Druga razlika je u kapacitetu – očigledno da jedna ploča diska formatirana po **rll** standardu ima veći kapacitet od **mfm** ploče, pa stoga i ceo **rll** disk ima veći kapacitet od identičnog **mfm** formatiranog diska.

Pominjemo razne **mfm** ili **rll** formatirane diskove, pa vas može zanimati po čemu se jedan disk svrstava u jednu ili drugu grupu. Generalno, odluka da li će se jedan hard-disk formatirati gušće ili ređe zavisi samo od njegovog kvaliteta – ne vredi nam da gusto „isparcelišemo“ disk, ako su podaci na njemu kasnije nestabilni, jednom dostupni a drugi put ne... Zato se uz disk koga je proizvođač deklarirao kao **mfm** po pravilu kupuje i takav disk-kontrolu – jer, da je proizvođač smatrao da on može da izdrži gušće formate zapisa podataka, deklarirao bi ga kao **rll** i prodavao ga po većoj ceni. Za razliku od toga, nije riskantno kupiti kvalitetniji disk i **mfm** kontroler, ali nije ni isplativo.

Upravo je postupak koga smo spomenuli koristio mnogima da povećaju kapacitete svojih hard-diskova: kupili bi neki **mfm** disk i formatirali ga kao **rll**, a onda bi se uzdali u „stepen sigurnosti“ koga je proizvođač sigurno ugradio u taj disk. Naime, **mfm** disk sigurno može da izdrži i veću gustinu zapisa od **mfm** kapaciteta (i analogno tome, **rll** od **rll**-a), pa je takvo formatiranje u velikom broju slučajeva uspevalo. Dobijao se povećan kapacitet hard-diska ali i daleko manja sigurnost podataka na disku. Kasnije bi počeli da se javljaju loši sektori, podaci bi se gubili, a muke koje su imali korisnici takvog diska bi se širile kao horor priče među ostalim korisnicima računara.

Kasnije, kada su veće firme uvidele koliko korisnici PC-računara žude za većim kapacite-

Dušan Popović

tima hard-diskova, u prodaji su se pojavili razni „specijalni“ kontroleri, koji su na „magičan“ način povećavali mogućnosti diskova. Pri tome je u reklamama stajalo da je ovaj sistem „apsolutno siguran“ i da sa njim nema nikakvih problema. Ukoliko ste pratili priču o **mfm/rll** standardima, onda vam je jasno o čemu se ovde radi: opet se neko dosetio da uz pomoć specijalnih kontrolera gušće pakuje podatke po disku, ne poštujući specifikacije proizvođača. Šta se ovim kontrolerima dobijalo, u odnosu na prethodno opisan „**mfm** disk i **rll** kontroler“ postupak? Pa ništa, osim što biste ovaj „specijalni“ kontroler skupo platili...

PRIHVATLJIVO REŠENJE

Očigledno je zašto su svi pobrojani sistemi bili pogrešni – svako povećanje kapaciteta diskova putem hardverskih zahvata (a i povećanje gustine zapisa je hardverski zahvat) nosi sa sobom nepouzdanost, koja će pre ili kasnije rezultirati gubitkom podataka na disku. Takođe je jasno da, kada više ne možemo da se pozdamo u hardverska rešenja, moramo da potražimo softverska.

Stacker je program koji nam omogućava da povećamo kapacitet diska čisto softverski, bez nepotrebnog naprezanja hardvera. Čim se pojavio, postao je hit, pa sada ima veliki broj korisnika. Verovatno vas zanima po čemu se *Stacker* izdvaja od drugih rešenja i šta mu obezbeđuje pouzdanost? Da bismo odgovorili na to pitanje, pogledaćemo kako *Stacker* radi.

Po instalaciji, *Stacker* će se startovati iz **config.sys** datoteke, dakle po svakom uključivanju računara. On tada preuzima sve interapte za rad sa diskom, tako da sva komunikacija između diska i računara ide preko njega. Sada na scenu stupa glavni deo *Stackera*, deo za kompresiju i dekompresiju.

Verovatno ste se već sretali sa programima za kompresovanje fajlova – kao na primer PKZIP-om i ARJ-om. Oni kompresuju fajlove na nekoliko procenata prethodne veličine, a zatim su u stanju da ih nepogrešivo vrate u originalni oblik. *Stacker* takođe vrši kompresiju i dekompresiju, ali u toku rada, tako da se nama čini da su fajlovi na disku u originalnom obliku. Time se dobija mogućnost rada sa (naizgled) većim diskom, ali bez potrebe da podatke koji nam trebaju stalno arhiviramo i dearhiviramo.

Posao koji radi sam program, možemo podeliti na dva slučaja: kada snimamo podatke na disk, program će ih kompresovati i snimiti ih u arhiviranom obliku, tako da zauzimaju manje mesta. Kada učitavamo podatke sa diska, program će dearhivirati zapis na disku, i nama dostavljati raspakovane podatke, tako da mi i ne moramo znati da su bili arhivirani... Ovaj posao *Stacker* vrši do krajnjih detalja, tako da se pri snimanju fajlova ažuriraju i FAT tablice, direktorijumi i svi parametri fajlova. Zato on uspeva da prevari i programe koji direktno (preko BIOS poziva) pišu po disku, kao na primer COMPRESS iz *PC Tools* programa ili *Norton Disk Doktor*...

Svi podaci se snimaju u jedan fajl na dupliranom drajvu (**stacvol.000**) koji je stalne dužine, a sve upise u taj fajl vrši sam *Stacker*, bričući se o alokaciji fajla i svim ostalim „režimskim“ poslovima. Sam **stacvol** fajl se sasvim normalno upisuje u sektore diska, dakle običan DOS zapis.

Sada vam je verovatno jasnije kako *Stacker* radi i zašto ga smatramo pouzdanijim od prethodnih rešenja – na disk se smešta više podataka tako što se oni kompresuju, a ne tako što se fizički drugačije smeštaju.

KO OVO PREŽIVI...

Za razliku od mnogih drugih programa, kod *Stackera* je instalacija najkomplicovanija operacija i zahteva najviše pažnje – kada jednom uspešno instalirate *Stacker*, nikakav (dobro, gotovo nikakav) dalji rad nije potreban. Dakle, kada jednom instalirate *Stacker*, on se ponaša sasvim providno, a vama se čini da radite sa istim (samo bitno većim) drajvom. Često je instalacija i strah od mogućih problema dovoljan razlog za odustajanje od ovog korisnog programa, pa ćemo instalaciju malo detaljnije objasniti. U objašnjenjima koja slede ćemo pominjati pojam duplikacije diska, pri čemu mislimo na korišćenje *Stackera* na tom disku – ne radi se baš uvek o dupliranju (već smo rekli da procenat povećanja diska varira), ali je pojam već odomaćen, pa ga i mi koristimo.

Pre instalacije samog *Stackera*, moramo se pozabaviti organizacijom diska. Verovatno znate kako se vrši particioniranje diskova (podela jednog fizičkog diska na više logičkih – C i D na primer). Dakle, morate se odlučiti kako ćete podeliti diskove, i koje ćete delove duplirati a koje ne. Naš je savet da veoma mali deo date DOS-u i *boot* partitiji, a ostatak *Stackeru*. Iako je moguće čak i *boot* partitiju duplirati, ima puno razloga da se to ne uradi. Kao prvo, na toj partitiji (C disku) je dobro smestiti DOS, *Stacker* i još nekoliko programa za oporavak diskova – na primer, *Norton* paket. Oni kasnije mogu dobro doći u slučaju neke greške na *Stacker* drajvu. Za C disk je dovoljno odvojiti 3 do 4 megabajta, jer se tu mogu smestiti svi pobrojani programi. Ostatak prostora se može duplirati bez problema – znači, ceo disk D, ako ste diskove podelili na C i D kako smo vam savetovali. Diskovi se particioniraju pomoću FDISK programa koji je deo DOS-a. Postupak samog particioniranja premašuje ciljeve ovog prikaza – kako se FDISK koristi, možete videti u nekoj od knjiga koje se bave DOS programima. Posle FDISK-a je potrebno formatirati logičke diskove – dakle **format c:** i **format d:**. Sada je još potrebno instalirati DOS i pomoćne programe, a tek zatim *Stacker*...

Instalacija samog *Stackera* počinje startovanjem **INSTALL** programa. Nadalje sve teče preko menija, i gotovo da nema načina da pogrešnim izborom nešto pokvarite.

Prvo pitanje je da li da u toku instalacije izbacite sve rezidentne programe ili ne. Razumljiva je bojazan autora programa da bi neki rezidentni program koji previše „petlja“ sa diskom mogao da zasmeta instalaciju. Probleme bi, pre svega, mogli da prave razni programi

za keširanje diska. Zato, ukoliko imate aktivan neki od njih – uklonite ga.

Stacker ima mogućnost da napravi veći disk od nekog praznog drajva u računaru ili da poveća neki već zauzet disk. Dupliranje već zauzetog diska vam ne bismo preporučivali – tu opciju ćete iskoristiti samo ukoliko nemate dovoljno disketa za *backup* ili ukoliko imate stvarno jakih razloga. Operacija dupliranja praznog drajva je krajnje jednostavna i preporučljiva.

Ukoliko ipak želite da duplirate drajv na kome se nalaze fajlovi, potrudite se da na disku imate i *norton speed disk*, program koji služi za defragmentaciju diskova. Ukoliko ga nemate, vaši podaci će biti izgubljeni, a da vam treba saznate tek po završetku kompresije. O čemu se radi? *Stacker* prvo kompresuje podatke na disku, i smešta ih u STACVOL.000 fajl na istom disku. Onda obriše deo već komprimovanih podataka, a zatim nastavlja sa radom dok se ceo disk ne nađe u stacvol fajlu. Takav rad garantuje veoma fragmentiran stacvol fajl, pa *Stacker* pokuša da ga kompresuje, upotrebom nortonovog programa za kompresiju. Ukoliko nemate taj program svi podaci će već biti kompresovani, ali ih *Stacker* neće poznati, jer je fajl previše fragmentiran. Ukratko, sve miriše na probleme, pa ovaj način dupliranja diskova izbegavajte po svaku cenu. Drugi razlog za izbegavanje ovog načina je sporost – mnogo je brže ARJ-om ceo disk skinuti na diskete, pa sadržaj po dupliranju vratiti, nego dozvoliti *Stackeru* da ih kopiruje...

Mnogo jednostavniji način je dupliranje praznog drajva – operacija traje izuzetno kratko, čak i ako uzmemo u obzir vreme potrebno za vraćanje sadržaja diska iz arhive (sa disketa).

Pošto odaberete način kompresije, morate odabrati disk na kome će se kompresija izvršiti. Pri tome nije dozvoljeno duplirati već dupliran drajv, što je i logično. Primitičete da je dozvoljeno duplirati i diskete, pa možemo imati flopi od 2-3 Mb!

Zatim vas računar pita koji deo izabranog diska da bude dupliran – odgovor „ceo disk” je logičan, iako vas program ne sprečava da samo deo diska duplirate, a ostatak koristite kao i pre instalacije.

Pre samog kreiranja *stacker* diska možete pozvati **advanced options**, gde se mogu podesiti još neki parametri. Samo ime nas opominje da su ta podešavanja samo za one koji znaju šta rade, i da nema preteranog razloga za njihovo menjanje. Ipak, u naprednim opcijama ima zanimljivih mogućnosti, na primer menjanje veličine klastera na disku ili menjanje podrazumevanog stepena kompresije. Već smo rekli da *Stacker* radi tako što kompresuje podatke. Verovatno znate da se različiti podaci mogu kompresovati sa različitim uspehom. Na primer, podaci koji sadrže veći broj istih delova (tekst fajlovi, na primer, imaju veći broj razmaka „ ”) mogu se bolje komprimovati od *exe* programa. Podrazumevani (ili pretpostavljeni) stepen kompresije je kod *Stackera* postavljen na 2:1, što obično ne treba menjati (zapravo, na našem disku je odnos 1.7:1, ali ni odnos neznačajno veći od pravog stanja ne smeta). Ukoliko na duplirani disk želite da smeštate fajlove za koje ste sigurni da se mogu bolje kompresovati (bitmap fontovi, na primer), možete povećati ovaj odnos do 8:1, ali ne treba preterivati. Ovaj odnos ne menja stvarne mogućnosti *Stackera* – on samo daje informaciju raznim *dir* i *directory maintenance* programima koliko mesta ima na disku. Ukratko, možete ovaj parametar postaviti na 4:1, ali kada disk ne bude mogao da prima više fajlova, DOS će prijaviti DISK FULL...

Pošto ste postavili sve parametre, pozovite opciju **create stacker drive**, i ostaje vam samo da sačekate kraj instalacije. Koliko će instalacija potrajati zavisi uglavnom od toga da li instalirate preko praznog ili popunjenog diska, o čemu smo već pričali.

U STACKER REŽIJI

Po završenoj instalaciji je potrebno resetovati računar da bi promene koje je *Stacker* uneo u **config.sys** počele da deluju. Prvo što svaki korisnik ovog programa uradi je start **dir** komande, a zatim sledi divljenje zbog dobijenog prostora na disku. Već smo rekli da veličinu diska koju **dir** prijavljuje treba uzeti sa rezervom, ali za početak sjajno zvuči 80 slobodnih MB na disku od 40 megabajta...

Kada se naviknete na mogućnosti novog drajva, počete da ga koristite. Rad sa dupliranim diskom se ni po čemu ne razlikuje od rada sa „normalnim” diskom, i ni po čemu ne možete primititi da je nešto menjano. Moguće su varijacije u brzini diska – što je brži računar a sporiji disk (idealna kombinacija 386sx i 28ms Seagate disk) primitičete ubrzanja u radu. To se može objasniti mogućnostima procesora da brzo (ili sporo) kompresuje podatke u odnosu na stvarnu brzinu diska. Odnosno, korišćenjem *Stackera* se rasterećuje disk, a opterećuje procesor u radu sa podacima – odnos njihovih brzina određuje konačni utisak. Ipak promene koje se primete nisu dramatične – rad može postati malo brži ili malo sporiji, ali ne drastično, pa se lako možete navići.

Pri radu još nismo našli na program koji ne radi sa *Stackerom* – bilo da radi pod DOS ili *Windows* operativnim sistemom. Koliko je dobro napisan *Stacker* vidi se i po tome što on bez problema radi sa *Windows*-om. *Windows* je čitav operativni sistem, pa ako uspeva njega da prevari, prevariće i bilo koji drugi program.

Pošto se duplirani disk ponaša kao običan disk, normalno je da ga tako treba održavati. Pre svega imamo problem fragmentacije diskova. On se otklanja nekim od programa za defragmentaciju (na primer COMPRESS-om iz *PC Tools* paketa), ili specijalnim SDFRAG programom koji se dobija uz *Stacker*. Glavna razlika između *sdefrag*-a i bilo kog od poznatijih defragmentatora diskova je u brzini. Naime, za normalnu kompresiju *sdefrag*-u treba čitav sat pa i više, pa ćete ga zato verovatno koristiti samo ako ste na to prisiljeni (ukoliko nemate drugi program).

U toku rada ne smete ništa raditi sa već pomenutim **stacvol.000** fajlom – on je, doduše, sa *system* i *read-only* atributima, ali bolje da vas i mi još jednom opomenemo. Naime, bilo kakvo menjanje (ili, ne daj Bože, brisanje) ovog fajla može rezultirati gubitkom podataka sa diska.

O radu sa dupliranim diskovima nećemo više ništa reći – jednostavno, ponašaju se potpuno kao normalni diskovi.

SRAČUNATI RIZIK

Jasno je da ovakav način organizacije smeštanja podataka na disk povlači za sobom i neke rizike. Pre svega, postoje veće mogućnosti grešaka, samim tim što su podaci „jače” povezani. Na primer, na običnom disku može biti loše alociranih sektora, dok na dupliranom disku *Stacker* odmah diže uzbunu i ne dozvoljava dalji rad.

Postoji više načina da dođe do greške na dupliranom disku. Dovoljno je da program ne-

što snima na disk, a nastane struje. Samim tim je kompresija stigla do neke tačke, podaci su „ni snimljeni ni nesnimljeni”, a FAT je netačan. Pri sledećem uključanju računara, *Stacker* će primiti nelogičnosti u FAT tabeli i pokušati da ih popravi, ali nije sigurno i da će uspeti. Ukoliko FAT nije popravljen, ceo duplirani disk će biti proglašen „samo za čitanje” (*read-only*). Kada se disk proglaši kao *read-only*, to znači da su podaci sačuvani, ali da nije dozvoljeno pisanje po njemu, zbog njihove sigurnosti. To je veoma kritična tačka u radu sa *Stackerom*, i od nje postoje dva puta: ili ćete uspeti da disk vratite u normalno stanje ili ćete biti prisiljeni da radite *backup* diska, ponovnu instalaciju *Stackera*, pa vraćanje podataka, što može biti višestruko veći rad.

Kada je već proglašen *read-only* diskom, postoje načini za ispravku grašaka. Prvi put kada disk bude proglašen kao *read-only* je ubedljivo najpovoljniji trenutak za pokušaj ispravljanja grešaka – svaki sledeći put vam daje manje šanse za spas podataka.

Postoje tri alatke koje preporučujemo za popravke dupliranih diskova. Prvi program je SCHECK, deo *Stacker* paketa. Drugi je CHKDSK, deo DOS-a. Treći je NDD (*Norton disk doktor II*, deo paketa *Norton utilities 5.0*). Prvo startujete SCHECK /f. On će vam dati objašnjenje i eventualni savet kako da popravite greške. Na svako pitanje slobodno odgovarajte sa „yes” (kucajući celu reč a ne samo prvo slovo). Zatim (što obično savetuje i SCHECK) startujete CHKDSK /f. On će takođe pokušati da popravi delove diska. Zatim konsultujte NDD, jer on daje detaljna objašnjenja grešaka na disku. Na kraju resetujte računar, da bi *Stacker* pri inicijalizaciji proverio sa kojim podacima trenutno raspolaže.

Ovaj postupak se ponavlja sve dok sva tri programa ne budu potpuno „zadovoljna”, odnosno sve dok bilo koji program prijavljuje greške. Ukoliko ni posle više pokušaja ne uspete da popravite greške, nema rešenja osim već pomenutog dugog posla – *backup, install*, pa vraćanje na disk...

Da biste izbegli probleme sa diskom, potrudite se da ne gasite računar i da ga ne resetujete ukoliko se ne nalazite u DOS-u. Čak i ako ste u *Windows*-ima, izađite u DOS, pa tek tada isključite računar. Naime, *Windows* uvek ima interno otvorene fajlove, koji pri naglom gašenju mogu da poremete FAT tabelu *Stacker* drajva.

ZA ILI PROTIV?

Autor ovog teksta koristi *Stacker* već nekoliko meseci, a tekst je direktan proizvod iskustva u radu sa programom. U toku korišćenja je dolazilo do raznih problema, koji su uz manje ili više truda otklonjeni. Mislimo da je dovoljna preporuka to što, i pored sitnijih stavki „protiv”, ima dovoljno razloga za dalje korišćenje programa.

Ovaj program se ne koristi samo kod nas (što bi se zbog našeg standarda i cene hard diskova moglo pomisliti) nego je veoma popularan u celom svetu. Pošto ima ogroman broj korisnika, može se smatrati dobro istestiranim i očišćenim od bagova, što je bitno kod programa koji kontroliše sve naše podatke na disku.

Lično ga smatramo dovoljno pouzdanim da (uz pogled čežnjivi pogled na cene novih diskova) bude nezamenljiv. Ako ste raspoloženi da i sami isprobate ovaj program, možete ga preuzeti sa Sezama (direktorijum).

PC386SX, 386, 486 ISA, EISA, LOCALBUS

SC3033SX	SC3040DX	SC4033DX	SC4050DX	SC4066DX2	SC L.BUS
386SX-33MHz RAM 2Mb FD 1.2 Mb HD 105 Mb 13 ms 16bit IDE CONT. SVGA CARD 1Mb 14" MONO MONIT. SVGA 1024 x 768 BABY CASE TASTATURA 101	386DX-40MHz 4Mb/128Kb cache FD 1.2 Mb HD 105 Mb 13 ms 16bit IDE CONT. SVGA CARD 1Mb 14" MONO MONIT. SVGA 1024 x 768 BABY CASE TASTATURA 101	486DX-33MHz 4Mb/256Kb cache FD 1.2 Mb HD 105 Mb 13 ms 16bit IDE CONT. SVGA CARD 1Mb 14" MONO MONIT. SVGA 1024 x 768 MIDI TOWER TASTATURA 101	486DX-50MHz 4Mb/256Kb cache FD 1.2 Mb HD 105 Mb 13 ms 16bit IDE CONT. SVGA CARD 1Mb 14" MONO MONIT. SVGA 1024 x 768 MIDI TOWER TASTATURA 101	486DX2-66MHz 4Mb/256Kb cache FD 1.2 Mb HD 105 Mb 13 ms 16bit IDE CONT. SVGA CARD 1Mb 14" MONO MONIT. SVGA 1024 x 768 MIDI TOWER TASTATURA 101	486DX-50MHz 4Mb/256Kb cache FD 1.2Mb+1.44Mb HD 540 Mb 12 ms 32bit 4Mb CACHE 32bit SVGA 1Mb 15" COLOR MONIT. SVGA 1280 x 1024 MIDI TOWER TASTATURA 101
1.650 DEM	1.990 DEM	2.690 DEM	2.990 DEM	3.350 DEM	5.950 DEM
CENE DELOVA				1-10 KOM.	11-30 KOM.
DISKOVI	HARD DISK 105Mb-TEAC, ALPS 13 ms HARD DISK 170Mb-CONNER 17 ms (serija JAGUAR) HARD DISK 210Mb-CONNER 12 ms HARD DISK 540Mb-CONNER 12 ms IDE III SCSI HARD DISK 1.3Gb-WREN 12 ms SCSI FLOPPY 1.2Mb / FLOPPY 1.44Mb			550 DEM 750 DEM 1000 DEM 2200 DEM 4200 DEM 150 / 130 DEM	500 DEM 680 DEM 900 DEM 2000 DEM 3800 DEM 130 / 120 DEM
KARTICE	16 bit 2HD, 2FD IDE CONTROLER+ 2RS232 + 1 PAR. 32 bit 2HD, 2FD IDE CONTROLER 4Mb CACHE SCSI 2HD, 2FD CONTROLER 4Mb CACHE SCSI EISA 2HD, 2FD CONTROLER 4Mb CACHE ETHERNET 16 bit NE 2000 / 16 bit WD ELITE FAX - MODEM KARTICA MNP5			50 DEM 350 DEM 1200 DEM 1400 DEM 280 / 380 DEM 300 DEM	40 DEM 320 DEM 1000 DEM 1200 DEM 250 / 350 DEM 250 DEM
VIDEO	SVGA MONO MONITOR 14" 1024x768 SVGA COLOR MONITOR 14" 1024x768 SVGA COLOR MONITOR 15" 1280x1024 RAVAN EKTRAN SVGA COLOR MONITOR 17" 1280x1024 RAVAN EKTRAN SVGA CARD TRIDENT 1Mb, 1024x768 S3 SVGA CARD 1Mb, 1280x1024 (6x BRŽA OD TRIDENTA)			300 DEM 650 DEM 1000 DEM 2000 DEM 160 DEM 350 DEM	250 DEM 550 DEM 900 DEM 1800 DEM 140 DEM 320 DEM
PLOČE	BOARD 386 SX-33MHz BOARD 386 DX-40MHz-128Kb CACHE BOARD 486 DX-33MHz-256Kb CACHE BOARD 486 DX-50MHz-256Kb CACHE LOCAL BUS 486 DX-50MHz-256Kb CACHE, 3x32bit SLOT BOARD 486 DX2-66MHz-256Kb CACHE			280 DEM 500 DEM 1200 DEM 1600 DEM 1900 DEM 1900 DEM	240 DEM 420 DEM 1000 DEM 1400 DEM 1700 DEM 1700 DEM
PRINTERI	HP IIIP LASER PRINTER 300x300 HP IV LASER PRINTER 600x600 EPSON PRINTER LX 400 A4, 9 PIN EPSON PRINTER LQ 570 A4, 24 PIN EPSON PRINTER LQ 1070 A3, 24 PIN EPSON PRINTER FX 1050 A3, 24 PIN			2900 DEM 3900 DEM 500 DEM 900 DEM 1300 DEM 1250 DEM	2600 DEM 3600 DEM 450 DEM 850 DEM 1250 DEM 1150 DEM
RAZNO	MEMORIJE SIMM 1Mb / SIMM 4Mb MAT. KOPROCESORI IIT 387SX-33MHz / IIT 387DX-40MHz BABY CASE + 200W / MIDI TOWER + 200W PROFESIONALNA TASTATURA 101 MIŠ - MICROSOFT KOMPATIBILAN STREAMER 120Mb / 250Mb			75 / 300 DEM 180 / 200 DEM 160 / 220 DEM 70 DEM 50 DEM 600 / 800 DEM	65 / 250 DEM 160 / 180 DEM 140 / 200 DEM 60 DEM 40 DEM 550 / 750 DEM

ISPORUKA ODMAH
GARANCIJA 12 MESECI
ŽIRO RAČUN: 60815-601-64787
RADNO VREME 9h-17h
TEL: 011/332-607
FAX: 011/345-126

PREDUZEĆE
ŠUTLIĆ
&
MIKRO DIZAJN

11000 BEOGRAD
Kosovska 32, I sprat
FEBRUAR 1993.

MR Systems & ISC

**SISTEM INŽENJERING INFORMACIONIH SISTEMA
SERVERI • CAD STANICE • RADNE STANICE • TERMINAL SERVERI
TERMINALI • SCO UNIX • TCP/IP • NOVELL LITE • UPRAVLJANJE
POSLOVANJEM • PROJEKTOVANJE PROIZVODA I TEHNOLOGIJA
UPRAVLJANJE TEHNOLOŠKIH PROCESA**

386SX 386DX 486DX ISA - LOCAL BUS - EISA



IZVOD IZ NAŠEG CENOVNIKA RAČUNARSKIH DELOVA

PRINTER HYUNDAI A3 9P	440 DEM
PRINTER HYUNDAI A4 9P	330 DEM
LAN KUĆIŠTE 60W	200 DEM
SLIM LINE KUĆIŠTE 200W	200 DEM
DESKTOP KUĆIŠTE 200W	145 DEM
MINI TOWER KUĆIŠTE 200W	155 DEM
TOWER KUĆIŠTE 230W	275 DEM
BIG TOWER KUĆIŠTE 300W	880 DEM
SVGA CARD TRIDENT 512K	120 DEM
SVGA CARD TSENG 1Mb	190 DEM
MONO VGA MONITOR	275 DEM
SVGA MONITOR 1024X768	590 DEM
INTERNI MODEM MNP5	150 DEM
INTERNI FAX MODEM MNP5	255 DEM
EXTERNI MODEM MNP5	215 DEM
EXTERNI FAX MODEM MNP5	330 DEM
POCKET FAX MODEM MNP5	330 DEM

POPUST ZA DILERE DO 10%

KOMPLET SVGA Monitor, Mini Tower Kućište, YU Tastatura **750 DEM**

Lomina 36 tel. 659-506 505-785 fax. 687-285



MEŠOVITO PREDUZEĆE EXPORT-IMPORT

KOLARČEVA 4/IV

BEOGRAD

TEL. 011 632 532

011 626 792

FAX. 011 633 059

ŽIRO RAČUN:

60803-601-118032

RAČUNARI

— Kompjuter 286/20 MHz	1.290 DEM
1MB RAM, Hard disk 40 MB, Flopi disk 5,25" 1.2 MB, VGA Mono	
— Kompjuter 386SX/33 MHz	1.800 DEM
— Kompjuter 386DX/40 MHz sa 128 Kb Cache	2.000 DEM
— Kompjuter 486DX/33 MHz sa 256 Kb Cache	2.890 DEM
— Kompjuter 486DX/50 MHz sa 256 Kb Cache	3.390 DEM
— Kompjuter 486DX2/66 MHz sa 256 Kb Cache	3.640 DEM
4MB RAM, Hard disk 40MB, Flopi disk 5,25" 1.2MB, VGA Mono	
DOPLATA za SVGA Color monitor 14" PHILIPS BRILLIANCE	
i grafičku karticu TRIDENT 512 Kb rezolucije 1024*768	490 DEM

KUĆIŠTA

BABY Flip Top	150 DEM
MINI Tower	160 DEM
SLIM Line	200 DEM
LAN Stanica	210 DEM
BIG Tower	330 DEM
SERVER BIG Tower	450 DEM

MONITORI

VGA Mono DTK	280 DEM
VGA Mono PHILIPS	400 DEM
SVGA Color ROYAL 14"	650 DEM
SVGA Color PHILIPS 14"	800 DEM
SVGA Color PHILIPS 14"	
LOW Emission	1.080 DEM

MEMORIJA

SIMM 1MB 70ns	80 DEM
SIMM 256*4 70ns	95 DEM
SIMM 4MB 70ns	280 DEM
DRAM 256*8 70ns	85 DEM

KONTROLERI

IDE BUS 2s/1p/1g	55 DEM
IDE BUS ISA 1MB Cache	850 DEM
SCSI-2 ADAPTEC 1542B	590 DEM
SCSI EISA 1MB Cache	1.300 DEM

MAINBOARD

80286/20 MHz	180 DEM
80386SX/33 MHz	350 DEM
80386DX/40 MHz 128 Kb	550 DEM
80486DX/33 MHz 256 Kb	1.350 DEM
80486DX/33 MHz EISA	2.200 DEM
80486DX/50MHz 256 Kb	2.050 DEM

HARD DISKOVI

42MB CONNER	390 DEM
85MB WD	540 DEM
125MB WD	790 DEM
212MB WD	1.200 DEM
520MB FUJITSU	2.600 DEM
340MB SCSI MAXTOR	2.500 DEM

PRINTERI

STAR LC-20, A4	550 DEM
PANASONIC 1123, A4	800 DEM
EPSON LQ 1070, A3	1.450 DEM

VIDEO KARTA

TRIDENT VGA 256Kb	95 DEM
TRIDENT SVGA 512Kb	130 DEM
TRIDENT SVGA 1MB	210 DEM

OSTALA OPREMA

LASER HP IIIp	2.900 DEM	Flopi disk 5,25" ili 3,5"	150 DEM
TASTATURA CHICONY	88 DEM	EPROM PROGRAMATOR	390 DEM
I/O KARTA 2s/1p/1g	40 DEM	CENTRONIKS KABL	20 DEM
NOVELL KARTA NE2000	455 DEM	MOUSE SA PADOM	80 DEM
NOVELL KARTA NE1000	425 DEM	DISKETE FUJI 5,25" 2HD	2.3 DEM
DISKET BOX 70 kom 5,25"	65 DEM	NAPAJANJE 200 W	80 DEM
DISKET BOX 150 kom 3,5"	65 DEM	NOSAC ZA HARD DISK	6 DEM

ERC POPUŠTA U SVEMU — OSIM U KVALITETU

LEGO JEZIK

U svetu postoji nekoliko stotina programskih jezika, svaki za svoju namenu, ali Srđan Mijanović, koga se stariji čitaoci svakako sećaju po „multiprocesorskom spektrumu”, smatra da još uvek ima dovoljno prostora i za inovacije u ovoj oblasti. Donosimo prikaz njegovog jezika *Manevro*, čija je osnovna osobina mogućnost nadogradnje do onog oblika koji korisniku najviše odgovara. Jezik će se uskoro naći i na Sezamu, i tada će svi koje ova tema zanima moći čitavu stvar da ispitaju i uživo.

Srđan Mijanović

Konkretno, *Manevro* mora koristiti RPN: zbog mogućnosti menjanja same sintakse jezika (kao što ćemo videti kasnije), bilo kakav proces preprocesiranja (uključujući pretvaranje iz infiksne u postfiksnu notaciju) postaje nemoguć.

Pojam koji je nerazdvojan od postfiksne notacije jeste stek. Stek je struktura podataka, gde protok podataka ide po sistemu zadnji_unutra_prvi_napolje (LIFO – Last In First Out). Sam DOS (ili bolje reći BIOS) vašeg računara podržava tzv. sistemski stek: na njemu se drže neki privremeni podaci i povratne adrese prilikom poziva potprograma. U principu, ovaj stek ne bi trebalo dirati: bilo kakvi zahvati na njemu mogu dovesti do kraha sistema. Primitimo da su naše operacije + i / iz primera (3) kao stvorene za rad sa stekom: zamislimo da je operacija + izvedena tako da skine dva broja sa steka, sabere ih i rezultat stavi opet na stek. Vidimo da je rezultat sekvence isti kao i u primeru (1).

Postoji algoritam koji će svaki izraz iz infiksne notacije prevesti u postfiksnu (i obrnuto); u stvari, prilikom izračunavanja infiksne notacije, računari se prvo pretvori u postfiksnu, koja je bliža njegovom 'shvatanju izraza'.

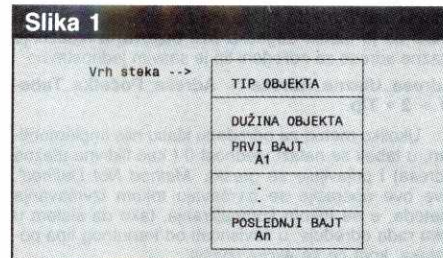
Manevro sistem koristi za interne potrebe dva steka koji se nazivaju povratni stek i parametar stek (*Return Stack & Parameter Stack*). Povratni stek ima istu funkciju kao i sistemski stek: na njemu se nalaze povratne adrese za izlaz iz potprograma kao i brojač **do..loop** petlje. U principu, operacije sa ovim stekom nisu dozvoljene (ili bar nisu preporučljive).

Parametar stek je ostavljen programeru: prenosi svih parametara između operacija se odvijaju preko ovog steka. Tu je i nekoliko osnovnih operacija za rad sa njim. Inače, *Manevro* na ovom steku može držati bilo koju strukturu podataka koja je dugačka do 255 bajtova; normalno, takvu strukturu je potrebno i opisati, što ćemo videti kako se radi malo kasnije. Za sada recimo da se podacima na steku može menjati redosled i da se mogu duplirati. Tim operacijama moguće je na steku napraviti bilo koji željeni razmeštaj podataka.

STEK

Manevro na svom parametar-steku, kao što smo već napomenuli, može držati bilo koju strukturu veličine 1 – 255 bajtova (objekti veličine 0 bajtova nisu dozvoljeni). Njihova reprezentacija na steku je kao na slici 1.

Slika 1



Parametarski stek

Tip objekta je informacija za sistem koja obezbeđuje filozofiju klasno baziranog sistema. Jedina klasa sa kojom *Manevro* 'zna' da radi u početku je tzv. klasa *integer* (celi brojevi u rasponu od -32767 do +32768). Međutim, ovaj sistem je otvoren ka proširenjima, pa

je moguće definisati, recimo, stringove (koje kao strukturu podataka poznaje gotovo svaki jezik), tako da ih kompajler prepoznaje kao da su oduvek bili tu!

Programer nizovima akcija dodeljuje imena koja zovemo reči. Postoji pet osnovnih reči za manipulisanje sa vrhom parametar steka u *Manevro* programskom jeziku (*dup, drop, swap, rot i over*). Preko ovih pet reči je moguće vršiti manipulacije sa prva tri objekta na steku. Pošto većina reči radi sa dva ili jednim objektom (to su ekvivalenti unarnih i binarnih operacija), očigledno je da su ove operacije sasvim dovoljne. Međutim, iako je stek preporučeno mesto za držanje objekata, sigurno nije i jedino (jer šta bi se desilo da na steku imamo više od tri objekta?). Zbog toga su uvedene dve vrste memorijskih referenci: konstante i varijable. Logično, konstante i varijable mogu biti bilo kog tipa, tj. one mogu predstavljati sadržaj bilo koje klase objekata.

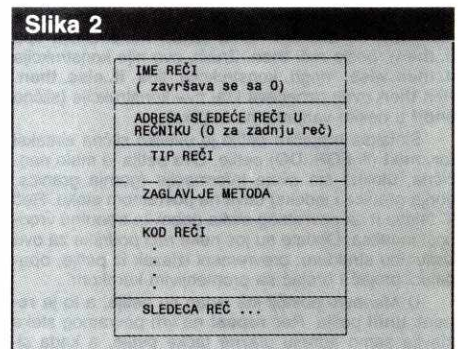
Ovo su, ukratko, bile osnovne operacije sa stekom. Pošto je *Manevro* potpuno otvoren sistem za nadogradnju, nije teško definisati još neke reči za manipulaciju stekom, ukoliko se pojavi potreba za njima.

REČNIK

U *Manevro* sistemu sve deklaracije izvršnih reči (procedure, metodi – koji će biti objašnjeni kasnije – varijable i konstante) se nalaze u posebnoj, jednostruko ulančanoj listi, nazvanoj rečnik. Tri glavne operacije sa rečnikom su dodavanje novih reči (biće prezentirano u sledećem poglavlju), brisanje reči iz rečnika, kao i pregled do tog trenutka definisanih reči.

Sama struktura jedne reči u rečniku se sastoji od imena reči, zaglavlja sa pointerom na sledeću reč, i tipa reči, opcione tabele ulaznih adresa metoda i samog koda te reči. U memoriji, to izgleda kao na slici 2.

Slika 2



Struktura reči u rečniku

Promenljiva *here* je usko povezana sa pojmom rečnika. Na toj adresi se nalazi WORD koji pokazuje na adresu prve neiskorištene memorijske lokacije u sistemu, tj. na prvu slobodnu lokaciju posle rečnika. Od te adrese *Manevro* počinje da kompajlira nove reči.

NOVE REČI

Definicija nove reči počinje deklarativnom rečju ':' (dvotačka). Posle nje se navodi prvo ime nove reči, zatim telo procedure (sastavljeno od poziva ranije definisanih reči), a kraj se označava rečju ':' (tačka-zarez). Reči koje definiše korisnik su potpuno ravnopravne sa rečima koje *Manevro* zna u osnovnoj verziji.

Prilikom redefinisiranja neke reči, prvo slovo stare reči se zamenjuje znakom ' ' (tilda), tako da poslednja definisana verzija ostaje validna.

Kao što se može primetiti, *Manevro* radi u dva modaliteta: kao interpreter i kao kompajler. U interpreter modu (interaktivnom modu), sve što korisnik otkuca se odmah izvršava. Kada interpreter naiđe na deklarativnu reč tipa ':', **const**, **var**, **method**, **class** ili **interrupt**, ulazi u mod za kompajliranje. Kompajliranje traje sve do znaka ';' ili do znaka **iret** (za *interrupt procedure*), ili se završava odmah nakon uzimanja imena nove reči (slučaj sa deklaracijama **const** ili **var**). Normalno, kada je u kompajler modu, *Manevro* ne dozvoljava još jedan ulazak u isti mod: drugim rečima, nije moguće direktno iz procedure definisati konstantu ili varijablu. Napomenuto je da to nije direktno moguće, ali su predviđene tri reči, **_interpret**, **_compile** i **make_proc** koje obezbeđuju nesmetano prebacivanje sistema iz jednog moda u drugi.

INTERAPT PROCEDURE

Razlika u ovom tipu procedura je to što se pri pozivu svi registri procesora čuvaju na povratnom steku. Početak kompilacije ovakve reči označava se pomoću reči **interrupt** (umesto ':'), a završava se rečju **iret**, umesto ':'. Ove procedure služe za pravljenje tzv. TSR (*Terminate and Stay Resident*) programa za IBM PC. Naravno, osim ovih, uvedene su još neke reči za tu svrhu: **init_m** inicijalizuje parametar stek, a **keep** vrši izlaz iz *Manevro* sistema, s tim da ceo paket ostaje u memoriji. Takođe, obezbeđene su i osnovne operacije sa flegovima (*Carry* i *Zero*) kao što su **stc** i **clc** (*Set & Reset Carry Flag*), **if_c** (*If Carry*) koja radi isto kao i naredba **if**, samo što ne očekuje logičku vrednost na vrhu steka i **if_z** (*If Zero*).

KONTROLNE STRUKTURE

Ovaj sistem je podržan standardnim setom kontrolnih struktura. Kontrolna struktura se prevodi isključivo u kompajler-modu, tako da se i ne može napisati da radi u interaktivnom režimu.

Osnovna stvar za bilo koju kontrolnu strukturu jesu konstante **true** i **false**. U *Manevro* sistemu, **true** ima integer vrednost -1, a **false** vrednost 0. Podržane su istoimene reči (**true** i **false**) za dovođenje ovih konstanti na vrh steka.

Osnovna struktura je svakako **if..then..else**, koja se, shodno RPN notaciji, ovde piše u malo izmenjenom redosledu. Reč **if** sa vrha steka skida *integer*, upoređuje ga sa 0, i, ako je on različit, nastavlja sa sledećom instrukcijom. Ukoliko on jeste jednak 0, nastavlja se sa izvršavanjem prve reči navedene posle reči **else**, ili, ako nje nema (znači konstrukcija tipa **if..then**), posle reči **then**. Znači, ovo nije konstrukcija **if..then..else**, nego konstrukcija tipa **if..else..then**. Reč **then** uvek označava kraj ove konstrukcije (slično **endif** u nekim varijantama bejzika).

Sintaksa brojačke petlje je uveliko slična sintaksi **for..next** (ili **FOR..DO**) petlje. Ova petlja je malo neobična, utoliko što svoje informacije (gornja granica, donja granica i indeks) čuva na povratnom steku. Reč 'i' (slovo i) sa povratnog steka dobavlja trenutnu vrednost indeksa. Dodate su još neke reči podrške za ovu kontrolnu strukturu: preвременi izlazak iz petlje, opadajući brojač i brojač sa promenljivim korakom.

U *Manevru* postoji još jedan tip petlje, a to je **repeat..until** petlja. Reč **repeat** na vrhu povratnog steka stavlja samo adresu ulazne tačke petlje, a kada sistem naiđe na **until**, skida *integer* sa steka i upoređuje ga sa 0. Ukoliko jeste 0, vrši se ponovni ulazak u petlju, a ukoliko nije, skida se zaostala ulazna adresa sa povratnog steka i izvršavanje se nastavlja normalno. Vidi se da je uslov izlaska iz ove petlje čisto logička vrednost (**true** ili **false**) koja se nalazi na vrhu steka pre nailaska na kraj petlje.

METODI I KLASI

Razmotrimo sledeći primer: treba implementirati formulu

$$(A+B)*(C+D)$$

tako da ona ravnopravno radi i sa *integer* brojevima, i sa *real* brojevima a i sa matricama. U poznatim proceduralnim jezicima, za svaki tip mora se pisati nova procedura. *Manevro* sistem prati sledeću logiku: za

Listing 1

```

0 var h_addr
0 var h_base
0 var h_flag
0 var h_len
0 var h_res
35 var hex_limiter
0 var h_res

: 160n \ ( :11 -- 16:11 ) stepenovanje broja 16
dup 0 = if drop 1 else 16 swap 1 - 160n * then
;

class hex
hex method get_c \ jedini metod koji nam treba!!!

dup get_byte hex_limiter @ = \ da li je prvo slovo '$'?
if
1 = h_addr | \ ako jeste, pocetak konverzije
4096 h_base |
0 h_len |
0 h_res |
false h_flag |
repeat
h_addr @ dup 1 + h_addr |
get_byte
dup dup 32 = swap 13 + or
if
drop
true h_flag |
else
dup 90 > if 213 and then
48 - dup
0 < if error Hex_What?? then
dup 10 <
if
h_base @ * h_res @ + h_res |
else
7 - dup 16 <
if
h_base @ * h_res @ + h_res |
else
error Hex_What??
then then
h_len @ 1 + h_len |
h_base @ 16 div dup 0 =
if drop 1 h_base | else h_base | then
then
h_flag @
until
h_len @ dup 0 >
if
h_res @
swap 4 swap - 160n
div
h_addr @ ilast
else
error Hex_What??
then
else
clc \ ukoliko prvo slovo nije '$', obavesti kompajler
then
;

```

Primer upotrebe programskog jezika *Manevro*

različite tipove podataka imena operacija ostaju ista, samo se menja njihovo dejstvo. Ukoliko imamo instaliran tip *real* i *matrix*, možemo pisati:

```

$ : formula a @ b @ + c @ d @ + * ;
ok
$

```

i procedura formula će raditi za svaki od navedenih tipova.

Ovo je moguće ostvariti posebnim vrstama procedura, nazvanim *metodi*. Metodi, za razliku od običnih reči, na svom početku imaju tabelu ulaznih adresa. Šta se dešava prilikom poziva metoda: sistem prvo pročita tip objekta sa vrha steka, a zatim nađe ulaznu adresu procedure (sadržanu u tabeli ulaznih adresa) koja obrađuje taj tip objekta. Time se omogućava da iste procedure rade nad svim tipovima, ukoliko su odgovarajuće operacije definisane.

Pošto ima ukupno 255 tipova, tabela ulaznih adresa sadrži mesto za adresu za svaki od tipova. To znači da je tabela dugačka 512 bajtova, a nalaženje ulazne adrese za određeni tip je sasvim jednostavno:

Adresa_Ulazne_Adrese := Adresa_Pocetka_Tabele + 2 * Tip

Ukoliko metod za određenu klasu nije implementiran, u tabeli se nalazi vrednost 0 (kao fiktivna ulazna adresa) i prijavljuje se greska *'Method Not Defined'*. Sve ove operacije se izvršavaju tokom izvršavanja metoda, a ne tokom kompajliranja, tako da sistem u toku rada određuje, u zavisnosti od trenutnog tipa podataka, koja će se akcija izvršiti.

NOVE KLASI

Manevro obezbeđuje deklarativnu reč **class** koja služi za upoznavanje sistema sa novom klasom. To je, u stvari, deklaracija konstante, čiji će sadržaj biti

identifikator (tip) te klase. Tip nove klase se definiše kao poslednji iskorišteni tip, uvećan za 1. Inače, u originalnoj verziji, u *Manevro* je instalirana samo jedna klasa (klasa **integer**) i njen tip je 0.

Definicija novog metoda je vrlo slična definiciji procedure. Umesto reči ':', koristi se reč **method**, koja na vrhu steka očekuje identifikator klase (tip) za koju se metod definiše. Ukoliko ime metoda ne postoji u rečniku, formira se nova reč zajedno sa tabelom ulaznih adresa, a kod novog metoda se smešta odmah posle tabele.

Ukoliko metod već postoji u rečniku (nađeno je da postoji reč sa istim imenom), ne formira se nova reč u rečniku, nego se popunjava tabela ulaznih adresa ranije definisanog metoda.

Sve do sada rečeno se može ilustrovati sledećim primerom:

```

$ class Moja_Nova_Klasa
ok
$ Moja_Nova_Klasa method +
$ <kod_metoda>
$ ;
ok

```

što će implementirati operaciju + u klasi *Moja_Nova_Klasa*.

KREIRANJE OBJEKTA

Kao što smo videli, objekat na steku se sastoji od podataka, dužine i identifikacionog broja klase. Najjednostavniji metod da se na stek 'ostavi' neki objekat jeste da se on kreira bajt po bajt, no, to je dosta zamoran posao. Istina, kreiranje objekata se ne može izbeći (postoji nekoliko specijalno dizajniranih reči za kreiranje objekata na steku) ali se može automatizovati, tako da kompajler, kada dođe do određenog formata zapisa, može to pretvoriti u konstantni objekat određene klase. To se postiže definisanjem posebnog metoda nazvanog **get_c** (*get constant*).

Pomoću metoda **get_c** kompajler i interpreter služi za prihvatanje konstantnih objekata neke klase. Evo kako teče kompilacija (ili interpretacija) u *Manevro* sistemu: prvo se pretraži rečnik i ustanovi se da li ta reč već postoji. Ukoliko postoji, ona se izvršava, ili se kompajlira poziv te reči na prigodnom mestu. Ukoliko ta reč ne postoji, *Manevro* ulazi u petlju poziva **get_c** metoda. Poziva se prvo metod čiji redni broj klase je 0 (to je **integer**), zatim 1, itd. Metod **get_c** ima za zadatak da prepozna nepoznatu reč, ukoliko se radi o nekoj konstanti. Recimo, isti metod za klasu **integer** proverava da li je neki zapis broj, i, ako jeste, vraća sistemu vrednost tog broja. Ukoliko nije, obavestava sistem o tome, pa on nastavlja sa prozivkom metoda.

Omogućeno je da korisnik sam definiše metod **get_c** koji će prihvatiti konstantne objekte. Ulazni parametar ovog metoda je adresa teksta koji *Manevro* sistem nije prepoznao kao ranije definisanu reč i koja se nalazi na vrhu parametar steka. Ukoliko metod prepozna konstantni objekat, reč **ilast** obavestava sistem gde je kraj nepoznatog teksta, a na vrhu se nalazi konvertovani objekat. Ukoliko tekst nije prepoznat, adresa početka se ostavlja na vrhu steka, a sistem se obavestava o tome instrukcijom **clc**. Na listingu 1 je dat primer kako *Manevro* možemo obogatiti tako da na ulazu shvata i heksadekadne brojeve kao brojeve koji počinju znakom '\$':

(Programeri koji su bliski sa *Forth* jezikom će većinom u potpunosti razumeti ovaj primer; oni koji nisu, uskoro će sa SEZAM-a moći da preuzmu kompletan program i uputstvo za rad sa *Manevro-m*.)

Poziv metoda **get_c** iz vašeg programa uslovljava da objekat koji se nalazi na adresi koju dostavite metodu bude pretvoren u konstantni objekat neke klase. Ukoliko tekst koji želite da pretvorite ne predstavlja objekat ni jedne klase, prijavljuje se greška **'Method Not Defined'**.

Objekti svih klasa mogu počinjati bilo kojim slovom ili znakom, osim sledećih: '\', '[', ']'. Takođe, mora se voditi računa o tome da ovaj metod, u slučaju prepoznavanja objekta, obavezno ostavi na vrhu steka njegovu vrednost sa tačnom dužinom i tačnim identifikatorom klase.

NASLEĐIVANJE

Iako je *Manevro* po definiciji klasno baziran sistem, to ne znači da ne postoji mogućnost nasleđivanja među klasama. Pretpostavimo da imamo opštu

klasu A i njenu podklasu B. Teorija kaže da bismo u klasno baziranom jeziku sve metode za klasu B morali ponovo da definišemo. Međutim, postoji jedan mali trik kojim se programer može poslužiti: ukoliko su reprezentacije objekata klasa A i B na steku slične, definicija metoda za klasu B može biti kao na listingu 2.

Listing 2

```
Listing 2
B method *      recimo da implementiramo metod *
<pretvori B u A> objekat klase B pretvori u objekat klase A
*              ovo sada aktivira metod za klasu A
<pretvori A u B> rezultat je formata klase A
;
```

Primer definicije metode za klasu

Procedure <pretvori X u Y> i ne moraju postojati, ako su reprezentacije objekata među klasama identične (recimo, klasa matrice i klasa kvadratne matrice).

KOMPILACIJA I INTERPRETACIJA

Na ovom sistemu izgrađene su tri veoma interesantne reči koje omogućuju korisniku da prebacuje *Manevro* iz interpreterskog u kompilacioni mod i obrnuto, iz svog programa. To, praktično, znači da je sa svim legalno napisati reč iz koje se definiše nova reč. Primer koji to najbolje ilustruje jeste definicija višedimenzionalnih nizova, gde poziv reči koja predstavlja identifikator višedimenzionog niza prvo odradi računarske adrese određenog elementa.

Prva reč, **_interpret**, nalaže *Manevro* sistemu da pređe u interpreter - režim. Na vrhu steka se očekuje adresa teksta koji će se izvršiti odmah, a koji je terminiran znakom ;s (tačka-zarez s). Iz ovog moda se mogu i kompilirati reči, tj. moguće je još jednom ući u kompajler-režim.

Sledeća reč je **make_proc**. Služi za formiranje zaglavlja reči negde u memoriji (ne mora u rečniku), sve do prve instrukcije. Na vrhu parametar- steka se očekuje adresa teksta po kojoj će reč biti nazvana i adresa od koje počinje reč. Uglavnom se koristi u paru sa rečju **compile**. Ova reč izvodi kompilaciju zadatog teksta od proizvoljnog mesta u memoriji. Pre poziva, na steku se nalazi adresa teksta, koji se završava znakom ;s (tačka-zarez s) i adresa od koje počinje kompilacija. Na kraju, reč **compile** vraća prvu slobodnu adresu. Da bi se izvršila kompilacija u rečnik, kao početno mesto za kompilaciju treba uzeti sadržaj promenljive **here**, a potom **here** korigovati na novu vrednost. Važno je reći da **make_proc** novu reč odmah ulančava sa zadnjom definisanom, tako da nema bojazni da će rečnik izgubiti svoju funkciju.

GREŠKE U PROGRAMU

Vrlo često se desi da reč koju ste napisali ne radi, iz potpuno mističnih i nerazumljivih razloga. U većini slučajeva kriv je sam programer i vrlo je zgodno obezbediti alat za traženje grešaka. *Manevro* ima tu mogućnost da u program koji se testira uključi informacije o izvršavanju. To se postiže rečju **/debug**. Posle navođenja te reči, sve sledeće reči (koje se kompajliraju) će na početku imati kompajliranu instrukciju **INT 63h**. Napravljen je posebni mali debugger program koji vam omogućuje **step** i **trace** kretanje kroz program, kao i praćenje sadržaja parametar-steka i proizvoljnih varijabli. Inače, osnovne reči *Manevro*-a imaju poziv tog interakta na početku, ali u slučaju da je **/debug** postavljen na OFF, instrukcija **INT 63h** se preskače pri kompilaciji.

Prijavlivanje grešaka u *Manevro* sistemu je izvedeno tako da se posle pojave greške program zastavlja, navodi se uzročnik između navodnika, ispisuje se tekst greške, zatvaraju se eventualno otvoreni fajlovi (ovo važi za sistemski otvorene fajlove; ne za fajlove koji predstavljaju datoteke sa podacima koje je otvorio sam programer), briše se parametar i povratni stek i prelazi se u interaktivni režim, bez obzira u kom režimu se sistem do tada nalazio. Naravno, greške je moguće prijaviti i iz vaše reči, upotrebom reči **error** <Tekst_Greške>. Ograničenje je to da se ne može

navesti šta je objekat greške (reč u navodnicima) i da tekst greške ne može sadržati blanko znake.

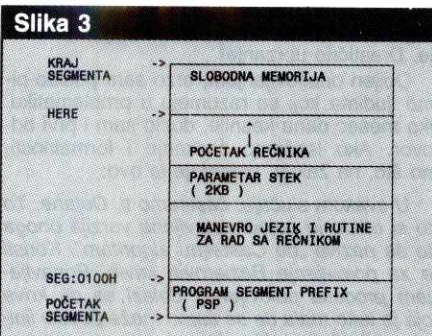
Podržana je i opcija da korisnik sam utvrdi akciju pri nastanku greške. To se obezbeđuje pomoću reči **on_err_goto**, koja očekuje adresu izvršne rutine koja će preuzeti posao **error handler**-a. Prvi deo ostaje isti, tj. ispisuje se tekst i briše se parametar-stek, a potom se kontrola predaje korisnikovoj rutini. Staro stanje stvari, tj. vraćanje na *Manevro* sistemsku rutinu za obradu greške, se ostvaruje posle poziva reči **restore_err**.

MAŠINSKA OKOLINA

Manevro u osnovnoj verziji nema nikakve specijalne instrukcije za rad sa I/O. Jedinne dve reči koje se oslanjaju na DOS vašeg računara su '.' (tačka, koja ispisuje objekat sa vrha steka (za početak **integer**) i uništava ga) i **load**. Reč **load** <FileName> radi redirekciju ulaza (tastature) ka fajlu **FileName** i tim putem se sistem puni programima. Redirekcija traje sve do reči ;s (tačka-zarez s). Isto tako, i u fajlu se može navesti **load**, sa neograničenom dubinom poziva. Sama reč **load** se izvršava isključivo u interpreterskom režimu rada (nije je moguće kompajlirati).

Osim unapred definisanih, postoji dosta reči koje služe za povezivanje *Manevro*-a sa MS DOS okolinom, putem DOS interapt funkcija. Pre svega, tu su reči **_reg** (gde reg. može biti ax, bx, cx, dx, si, di, es, ds) koja postavlja sadržaj odgovarajućeg registra, i reč **reg_**, koja na vrh steka dovodi sadržaj registra (ovi sadržaji se koriste samo pri instrukciji **_int**). Reč **_int** izaziva softverski interrupt, obično 33, koji je zadužen za DOS podršku. Isto tako, tu su i već pomenute reči za rad sa flegovima, koje omogućavaju detektovanje grešaka koje prijavljuje sam DOS.

Druga grupa reči odnosi se na manipulacije sa memorijom. Sama struktura *Manevro* jezika u memoriji je kao na slici 3.



Struktura *Manevro* jezika u memoriji

Kao što se vidi, ceo program sa podacima zauzima jedan segment. Unutar tog segmenta, bajtovi se mogu čitati, smeštati i premeštati. Naravno, tu su i reči za čitanje i pisanje po memoriji kada je zadata apsolutna adresa, koja može biti bilo gde u memoriji vašeg računara. U suštini, programer u osnovnom segmentu mora držati samo stek i rečnik. Prostor za podatke se može dinamički alocirati i nije ograničen na 64KB.

Posebnu grupu procedura čine reči **isp** i **@sp**, koje menjaju sadržaj pokazivača vrha steka. Drugim rečima, moguće je ostvariti dva ili više parametar-steka ili potpuno relocirati postojeći.

ASEMBLER

Kao što je napomenuto, u ovaj programski paket je uvedena mogućnost da programer, iz nekih razlo-

Slika 4

```
$ : vlist
$ asm{
$ code segment
$   mov ax,#vlist
$   add ax,3
$   call ax
$ code ends
$ end
$ } ;

Turbo Assembler Version 1.01 Copyright (c) 1988, 1989 Borland International

Assembling file: TEMP.ASM
Error messages: None
Warning messages: None
Remaining memory: 342k

Turbo Link Version 2.0 Copyright (c) 1987, 1989 Borland International
Warning: no stack
ok
```

Povezivanje sa asemblerom

ga, uključi asemblerski fajl u svoju reč. To se postiže naredbom **asm{**, koja se ne može izvršavati u interaktivnom režimu (što je i logično). Prilikom kompilacije, posle te reči, *Manevro* sistem kreira fajl na tekućem disku pod imenom **temp.asm** i na početak tog fajla piše tekst

ORG <adresa>

gde je <adresa> adresa od koje počinje kompilacija (trenutni sadržaj pokazivača adrese na kojoj se vrši kompilacija reči). Posle toga se prepisuje sadržaj koji je programer napisao sve do znaka '}'. Izuzetak od tog pravila čine reči koje počinju znakom '#' (razmak/hash). Tada se čita reč koja dolazi posle tog znaka, tražeju se rečnik, i ukoliko se takva reč nalazi u rečniku, njen naziv se zamenjuje adresom zadnjeg slova njenog imena (ista adresa koju daje i reč **find_w**). Ukoliko nije nađena odgovarajuća reč u rečniku, cela reč posle znaka '#' (zajedno sa njim) se prepisuje u fajl, kao da se ništa nije dogodilo. Time je obezbeđena minimalna, ali efikasna saradnja asemblerskih fajlova sa *Manevro* rečnikom.

Posle kreiranja fajla **temp.asm**, poziva se eksterini assembler i linker. Komandna linija asemblera se sastoji od imena **temp.asm** (proverite da li vaš assembler podržava parametre u komandnoj liniji), a istoimena linija za poziv linkera (isto eksternog) se sastoji od imena **temp.obj**. Posle pozivanja asemblera i linkera, poziva se DOS program **exe2bin** koji .exe fajl pretvara u formu pogodnu za uključivanje. Fajl **temp.bin** (nastao posle **exe2bin**) se uključuje u definiciju reči, briše se svi fajlovi osim fajla **temp.asm** (koji ostaje radi provere eventualne greške) i nastavlja se normalno sa kompilacijom reči. Sve ovo ilustruje primer na slici 4.

U slučaju greške na bilo kom mestu (greška u asemblerskom fajlu, nemogućnost konverzije u .BIN fajl i sl.), prijavljuje se greška **'Error In Assembler File'**, bez oznake gde je ta greška nastala. Greška se detektuje pregledom fajla **temp.asm**.

Od procesorskih registara, *Manevro* koristi samo BP, koji pokazuje na trenutni vrh parametar-steka. Ovaj registar se može menjati u okviru asemblerske rutine, ali pri izlasku iz reči mora imati odgovarajuću vrednost. Inače, na adresi **byte ptr [bp]** nalazi se tip klase, a na adresi **byte ptr [bp-1]** dužina objekta. Pošto je sam *Manevro.com* fajl, registri CS, DS i ES imaju istu vrednost (koju daje reč **myseg**).

Ceo sistem sa asemblerom je zamišljen, pre svega, kao ispomoc pri pravljenju nekih reči koje je jednostavnije uraditi u asembleru nego u bilo kom drugom jeziku. Isto tako, na ovaj način je omogućeno da korisnik proizvoljno implementira reči sa efektima koji se ne mogu izvesti iz samog *Manevro* sistema, kao što je rad sa memorijskim blokovima, rad sa proširenom (produženom) memorijom, specifičnim I/O uređajima i slično.

Rad na ovom projektu svakako nije okončan, što korisnicima obezbeđuje još dosta novih modula, klasa i reči za razne primene. Posebna pažnja se usmerava ka nalaženju i ispravljanju bagova programa kojih na žalost, sigurno ima (bilo bi ludo nadati se da ih nema), ali koji do sada nisu primećeni. Zato se autor unapred zahvaljuje za svaku prijavljenu grešku u radu kompajlera i za svaki savet koji bi mogao pomoći da ovaj sistem pruži upravo ono što očekujete od jednog programskog jezika.

OBJEKTI U PRAKSI

Čini se da su sve dobre ideje u istoriji bile otkrivane više puta. Žalosno je da čak i danas, uz sve informacione sisteme kojima raspoložemo, devedeset posto svega novoga ispadne „otkrivanje rupe u saksiji”. Setite se toga sledeći put kada budete smišljali neki novi algoritam.

S obzirom na trenutno stanje jugoslovenske privrede, mislim da neću pogrešiti ako kažem da većina programera zarađuje za život na bazama podataka. Kako taj posao ne razvija moždane vijuge baš previše, mnogi ljudi pronalaze još neku oblast kojom se bave kao nekom vrstom hobija. Kriptografija, algoritmi za sortiranje i pronalaženje slučajnih brojeva predstavljaju najzastupljenije razbibrige, mada se i moja oblast, ubrzavanje grafičkih algoritama, nalazi pri vrhu top-liste.

RUPA U SAKSIJI

Među grafičkim primitivama, Bezierova kriva, zbog svojih karakteristika, drži istaknutu poziciju. Za one koji to ne znaju, Bezierovu krivu je ranih sedamdesetih godina otkrio francuski matematičar Pjer Bezier, koji je u to doba radio za „Reno”, pa mu je trebala kriva kojom bi opisivao karoserije automobila u tri dimenzije. „Računari” su već dosta pisali na ovu temu, tako da vas time neću daviti. Ukratko, ukoliko imamo četiri tačke (u prostoru ili u ravni – broj dimenzija i nije bitan), Bezierovu krivu ćemo dobiti po formuli

$$\vec{x}(t) = t^3x_1 + 3t^2(1-t)x_2 + 3t(1-t)^2x_3 + (1-t)^3x_4$$

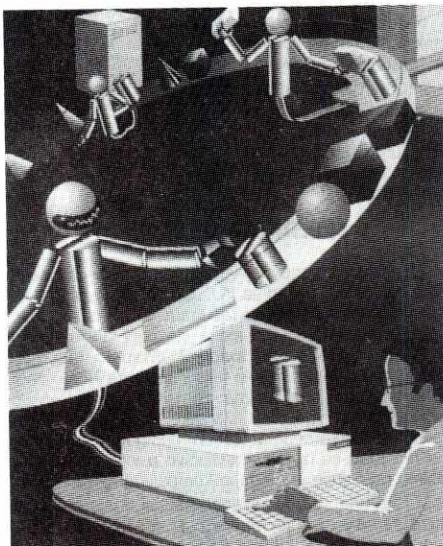
gde se t kreće od 0 do 1. Zamenjivanjem x sa y u ovoj formuli dobićemo, naravno, i y koordinate za našu krivu, te je onda možemo prikazati na ekranu. (Formula koju sam dao jeste samo specifičan slučaj Bezierove krive u kubičnoj formi. Postoje i Bezierove krive koje se zasnivaju na više od četiri tačke, ali se one mnogo manje koriste. Ova je prava stvar.)

Ono što Bezierovu krivu čini zanimljivom je to da kreće iz tačke (x_1, y_1) , završava se u (x_4, y_4) , a (x_2, y_2) i (x_3, y_3) su kontrolne tačke – kriva je u početnoj tački tangentna u odnosu na (x_2, y_2) , a u krajnjoj u odnosu na (x_3, y_3) . To je zgodno, jer sada lako možemo da spojimo dve Bezierove krive: namestimo početnu tačku druge krive na krajnju tačku prve, i spojili smo krive. Sada još namestimo okolne kontrolne tačke, tako da zajedno sa tačkom u kojoj su krive spojene budu kolinearne, i naš spoj će biti „gladak”, tj. na njemu neće biti nikakvog špica. Ovo svojstvo je toliko korisno, da su gotovo sve firme koje se bave izradom vektorskih fontova počele da ih zasnivaju isključivo na pravim linijama i Bezierovim krivim.

Elem, da se vratim na temu, Bezierove krive se zbog svog onog stepenovanja i množenja crtaju veoma sporo. Postoje, naravno, postupci kojima se sve to svodi na jednostavnije operacije i uklanja se potreba za korišćenjem realnih brojeva (t se kreće od nula do jedan), ali je sve to i dalje **sporo**. Kako se Bezierove krive koriste u manje-više svim danas raspoloživim fontovima, onoga ko bi pronašao brži algoritam za njihovo crtanje očekivala bi gomila zelenih šušlavaca. Mlad i zelen, ja sam mislio da imam rešenje. Budala.

Moja „revolucionarna” ideja bila je da prvo na osnovu četiri tačke zamislim tri linije: z_1 do z_2 , z_2 do z_3 , i z_3 do z_4 . Onda bih, na osnovu te tri linije, iskonstruisao još dve – od sredine z_1 - z_2 do sredine z_2 - z_3 , i od sredine z_2 - z_3 do sredine z_3 - z_4 . Kada odbacimo višak (linije vezane za

Dejan Jelović



z_2 i z_3 sada postaju nepotrebne), onda opet ponovimo postupak: za svake dve susedne linije spajamo njihove sredine i onda odbacujemo onu tačku koja ih je spajala, kao i njene dve „parazitske” polovine linija. Ponavljajući ovaj postupak pet-šest puta, dobili bismo sasvim dovoljno tačaka da nacrtamo krivu. Na moje iznenađenje, ovaj algoritam je zaista crtao Bezierove krive, i to praktično dvaput brže nego original. Sve one matematičke operacije su zamenjene samo nalaganjem tačke tačno između druge dve, što se svodi na sabiranje i šiftovanje. Drastično ubrzanje!

Opijen oduševljenjem, brzo sam poslao pisnu ljudima koji se razumeju u problematiku. Oko mesec dana kasnije, dobio sam i prvi odgovor. Ako isećemo ćaskanje i formalnosti, ono što, na žalost, preostaje je ovo:

U svakom slučaju, čestitam ti, Dejane. To što si otkrio je pojednostavljena verzija onoga što se naziva „de Castelju algoritam”. Koristi se za polovljenje Bezierovih krivih. Ponavljanjem procesa polovljenja, dolazi se do krive koja je tako mala da se samo može povući linija između četiri tačke, što mi upravo i radimo.

Tvoja verzija algoritma ima jedan nedostatak, a to je da se krajnje tačke krive zanemaruju, tj. imaju oko sebe mnogo manje određenih tačaka nego sredina krive. Dok je ovo prihvatljivo kod samostalne krive, kada treba da se poveže nekoliko krivih, dolazi do lošeg uklapanja. Žao nam je...

Ode moj milion dolara. Šmrnc.

KO UBIJA OBJEKTE?

Gornja priča (s tužnim krajem?) nosi lepu poruku: pre nego što počnete da se ozbiljno bavite bilo čime, dobro bi bilo da pročitate nekoliko knjiga na tu temu. Niko ne sumnja da sve možete da shvatite i sami, ali – bolje je učiti na njihovim greškama nego na vašim. Koju god oblast da izaberete, neko je na nju već

potrošio desetak godina. Ukoliko taj neko nije baš potpuni idiot, imaće bar nekoliko dobrih saveta koji će vam uštedeti dosta vremena i time isplatiti vreme utrošeno na čitanje njegove knjige. Najčešće to zapravo i nisu pogrbljeni matori profesori matematike koji pišu programe na papiru, već programeri kao i vi, koji su to pisali na osnovu praktičnih iskustava.

U tom pogledu, nema razlike ni u oblasti objektnog programiranja. Ljudi se time bave već dvadesetak godina i prvo treba da vidite šta su drugi po tom pitanju uradili. Ukoliko koristite Turbo Pascal ili C++, nije dovoljno da pročitate knjige koje se bave isključivo ovim jezicima, već treba da potražite i one koji pokušavaju da objasne kako se pišu objektni programi. Evo o čemu govorim:

Automatsko čišćenje memorije predstavlja jedan od sedam „zlatnih” uslova koje svaki OOP jezik treba da ispunjava. Uprkos tome, gotovo nijedan hibridni jezik ga ne podržava, jer bi to usporavalo program. Da bi se ostvarilo čišćenje memorije, svaki program bi morao da ima interapt rutinu, koja bi se javljala svakih nekoliko sekundi, i koja bi:

- Prošla kroz ceo **heap** i uzela podatke o svim alociranim blokovima.
- Prošla kroz stek i ispitala pointere na njemu da li pokazuju na neki od ovih blokova. Ukoliko pokazuje, taj blok bi bio markiran. (Zapravo bi rutina ispitala sve podatke na steku, jer nema načina da utvrdi koji je pointer, a koji, recimo, int.)
- Prošla kroz markirane blokove i za sve pointere u njima opet markirala blokove na koje pokazuju.
- Obrisala sve nemarkirane blokove.

Sve ove operacije, naravno, uzimaju procesorske cikluse i veoma se teško „kaleme” na klasičan jezik kakav je C ili Paskal, pa su zato tvorci hibridnih jezika odlučili da ih ne implementiraju. Nema veze, kažete vi, nije meni teško da za svaki objekat koji sam alocirao na **heap**-u uradim **delete**. E, u tome je stvar! Nije ni njima bilo teško. Problem je, naime, na sasvim drugoj strani:

Recimo da imate klasu **List**, koja predstavlja uvezanu listu u koju stavljamo stringove tipa **String**. Neke od tih stringova stavljamo da bi smo ih odložili, i više ne razmišljamo o njima, a na neke, i pored toga što su u listi, i dalje imamo uperene pointere i koristimo ih u drugim delovima programa. Problem će nastati kada poželimo da uništimo listu, bilo zato što je taj objekat izašao iz funkcije u kojoj je deklarisan, bilo zato što mi to radimo komandom **delete**. Ukoliko lista nije prazna, šta treba uraditi da bismo je uništili? Da li da pretpostavimo da za svaki objekat u njoj programer ima referencu, te da će se sam pobrinuti o tome da ih izbriše, ili da ih sve od reda obrišemo, rizikujući da neki drugi deo programa o tome nema pojma, pa da preko pointera referencira objekat koga više nema? U sistemima sa automatskim čišćenjem memorije, mi o tome ne bismo brinuli. Listu bismo jednostavno uništili, a mehanizam za čišćenje bi sam „ubijao” objekte kada na njih više ne postoje reference.

Naravno, iako je meni najlakše da objasnim

Listing 1

```

{
    Menu menu;
    Choice file ("File", filemenu); // na steku

    menu + file
        + *new Choice ("Edit", editmenu); // alocirano na heap-u
    ...
} // Ovde nastaju problemi!

```

Listing 3

```

class vector3D {
    float v1, v2, v3;
    friend ostream& operator << (ostream &ostr, const vector3D& v)
    { return ostr << '(' << v.v1 << ', ' << v.v2 << ', ' << v.v3 << ')'; }
    ...
};

```

ovaj problem baš na nekoj kontejnerskoj klasi (**list**, **array**, **bag** – objekti koji su zaduženi da sadrže druge klase), ovaj problem se može javiti svuda. Ja sam u svom programu imao klasu **Menu** i njen objekat **menu**, na koji je trebalo da „dodajem” pojedinačne stavke u menijima. Međutim, trebalo je da neke stavke dodatno kontoliram, pa sam imao i dodatne referencije na njih. To je izgledalo, otprilike, kao na listingu 1.

Kada bih na kraju funkcije uništavao objekat **menu**, ukoliko bi **menu** automatski uništavao sve stavke (klase **Choice**), pokušao bi da sa **delete** uništi i stavku **file**, koja je na steku. Sa druge strane, ako ih ne bih uništavao, onda bih, da bih mogao ručno da je uništim, morao za stavku koja vodi u **editmenu** da držim pointer, iako mi on ne treba. Trulo.

Ako hoćete neko „normalno” rešenje, ja ga nemam, a mislim da i ne postoji. Ono što sam uz pomoć Zorana Životića implementirao kao neku vrstu zakrpe, jeste funkcija koja određuje da li je neki objekat na *heap-u* ili na steku (listing 2).

Uz to sam sve kontejnerske destruktore izmenio, tako da uništavaju isključivo objekte na *heap-u*, i uveo konvenciju da sve objekte koji treba da imaju još neku referencu alociram na steku. Ovo radi savršeno, uz dva nedostatka: prvo, užasno je neprenosivo, i, drugo, neđo bog da neko ko ne zna za ovu konvenciju uzme da koristi ove klase. Mirisao bi ljubičice odozdo. Bio bi mrtav. Dead. Kaput.

ŠTA JE VIRTUAL INLINE?

Svaki objekat ima uz sebe i nešto što se zove „pokazivač na tablicu virtuelnih funkcija”. U toj tablici se nalaze, redom, adrese svih virtuelnih funkcija koje figurišu za tu klasu. Svaki put kada treba da se pozove virtuelna funkcija za neki objekat, njena adresa u memoriji se potraži u ovoj tablici. (Tablica je odvojena od objekta, jer je, zapravo, potreba samo jedna tablica za svaku klasu. Bilo bi to zaista traćenje memorije kada bi hiljade objekata iste klase nosili svaki svoju tablicu.)

U takvim uslovima, *virtual inline* funkcije zaista zvuče kao loša šala. *Inline* funkcije ne stoje nigde već se po potrebi ubacuju direktno u kod, a virtuelne funkcije moraju negde da budu smeštene da bi tablica virtuelnih funkcija mogla da pokazuje na njih. Ipak, to nije apsurd. Virtuelne *inline* funkcije se zaista tretiraju kao prave virtuelne funkcije, tj. biva im dodeljeno mesto u memoriji. Što se tiče „*inline*” dela, on je u akciji samo ako implicitno odredite klasu iz koje vam funkcija treba. Znači, ako kažete, recimo

funkcija (5);

bíće pozvana virtuelna funkcija. No, ako kažete **klasa::funkcija (5);**

ona će biti upisana *inline*. Pri tom, naravno, ne treba zaboraviti da u vreme izvršenja konstruktora nijedna funkcija ne biva izvršena kao virtuelna, jer tablica virtuelnih funkcija u tom trenutku još nije napravljena, te je to mesto gde će ovakve funkcije definitivno biti pozvane kao *inline*.

Ako se malo zamislite nad ovim mehanizmom virtuelnih *inline* funkcija, nameće se pitanje: a gde, onda, bivaju u programu zapisane te funkcije, ako negde moraju da budu zapisane da bi bile virtuelne? Za razliku od „normalnih” funkcija, koje se definišu samo u jednom modulu, *inline* funkcije se najčešće definišu u heder-datotekama koje su na raspolaganju svim programskim modulima. Da li onda kompajler ubacuje definiciju *virtual inline* funkcija u svaki modul? Odgovor je, naravno, ne. Ljudi koji su se bavili teorijom objektnog programiranja su (i) ovo predvideli, te je još odavno odlučeno da definicije *virtual inline* funkcija budu smeštene uz proizvoljnu (kompajler sam bira) funkciju koja nije *inline*. Ipak, ni kompajler nije svemoćan. Ukoliko imate klasu u kojoj su sve funkcije *inline*, onda će, ne znajući gde da ih smesti, *virtual inline* funkcije ubacivati najvećevratnije u svaki modul, zajedno sa tablicama virtuelnih funkcija. Zato, ukoliko imate neku klasu sa isključivo *inline* funkcijama, bilo bi dobro da ubacite i neki „*ne-inline*” član i time uštedite dva-tri kilobajta.

NIKO NE SME DA IH DIRA

Ukoliko niste primetili, u prošlom broju „Računara” se pojavio jedan napad na moj život i delo. Ne bih vas sada uvodio u detalje te vatikansko-kominternovske zavere (ko hoće može da pročita moj odgovor negde u ovom broju), ali bih vam preporučio da pročitate to pismo zbog nekoliko zanimljivih rečenica koje otvaraju mali prozor u izuzetno zanimljivu disciplinu dizajna objektnih programa. Mene je, opet, to pismo podsetilo da u „Računarim” još uvek nije pala dobra diskusija na temu **friend** klasa i funkcija. Hajde da to odradimo.

U prošlom broju smo se bavili skrivanjem podataka unutar klase, te neću time opet da davim. Ukratko, za svaku klasu treba jasno definisati ono što je privatno (to samo ona sme da dira), ono što je zaštićeno (to smeju dirati i naslednici), i ono što je javno, što svi smeju da diraju (pričamo o podacima – devojke koje ste nekada znali nemaju sa tim veze). **Friend** klasa i funkcije predstavljaju način da se ovaj me-

Listing 2

```

int isOnHeap (void *object)
{
    #ifndef __MSDOS__
        #error This routine works only on MS-DOS machines
    #endif
    #ifdef __TINY__
        #error Tiny model not allowed for this function.
    #endif
    #ifdef __SMALL__ || __MEDIUM__
        return object < &object;
    #else
        return (((unsigned long)FP_SEG(object)<<4) + FP_OFF(object) >
            ((unsigned long)SS << 4) + _stklen);
    #endif
}

```

hanizam zaobide, tj. time što neku klasu ili funkciju proglasimo za **friend**-a, mi joj dopuštamo da dira čak i privatne delove neke druge klase. Ništa loše u tome.

Ono što jeste problem je to što mnogi ljudi siluju **friend** mehanizam. Umesto da lepo isplaniraju kako mogu koristiti i proširivati klasu, pa da na osnovu toga odrede koji članovi treba da budu javni, koji zaštićeni, a koji privatni, oni to urade „po osećaju”, i zeznu stvar. Kada im mesec dana kasnije zatreba neka „*private*” promenljiva, da se ne bi maltretirali oni samo pozovu zaglavlje za tu klasu i ubace **friend** deklaraciju. Bez veze. Malo planiranja daje odlične rezultate i (koliko to puta ponavljam) zapravo štedi vreme.

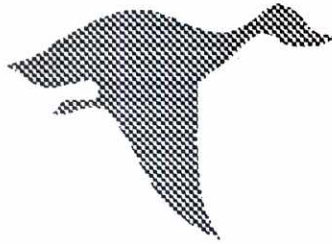
A to što se matematičarima i drugim teoretičarima **friend** klase i funkcije baš ne sviđaju, mnogo me ne uzbuđuje. Nije teško zamisliti situaciju u kojoj je **friend** klasa izuzetno korisna. Na primer, imate objekat čijim pojedinim članovima treba da ima pristupa konačan broj klasa (objektni sistemi nemaju vrha – teoretski mogu da sadrže beskonačno mnogo klasa), a da ostale klase ne treba da imaju nikakvo pravo pristupa. Bilo bi opasno te članove proglasiti javnim; s druge strane, ako bi oni bili privatni, klase koje bi trebalo da imaju pristup ne bi im mogle pristupiti. Rešenje je se, naravno, moglo naći u eksplicitnoj proveru tipa podataka, ali to bi bilo silovanje po svaku cenu – **friend** klase su najnormalnija stvar u takvim situacijama.

(Uzred, ukoliko treba da obezbedite I/O vaših objekata, ne morate da za svaku novu klasu koju napravite proširujete klase **istream** i **ostream**. **Friend** funkcije će vas u tom slučaju izvući mnogo bolje (listing 3).

Pri tom, zapamtite: **friend** funkcije su konceptijski mnogo manje vredne (jer ne mogu biti virtuelne) od funkcija članica. Upotrebljavajte ih samo tamo gde morate.)

TELEGRAMI PODRŠKE

Cela ova stvar sa kritikom mog teksta me je podsetila na staru priču o tome da li treba objavljivati adrese autora u „Računarima” radi kontakta sa čitaocima. Od toga se manje-više odustalo iz nekih tehničkih razloga, ali sada mi se opet čini aktuelnim, pa ako neko hoće da kontaktira sa mnom, može to da obavi na adresu: Partizanska 70, 11090 Beograd, ili me može naći pod BUEF78::JELOV105090D na Decnetu i DJELOVIC na Sezamu. Sve kritike, pretnje i ucene i dalje šalјite na adresu redakcije; ja primam samo krupne apoene stranih valuta, ključeve od skupih kola, telegrame podrške, i pisma vezana za C++ i objektno programiranje.



ADA computers

BEOGRAD

Tadeuša Koščušskog 72

tel/fax: 011/186-267; tel:011/186-355

NOVI SAD

Siriška 42

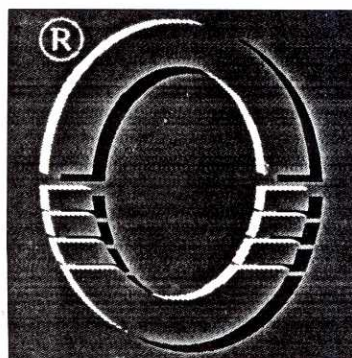
tel/fax: 021/416-189

MODEL 286/20 MHz	1.190	MGP (Herkules) KARTICA	50
MODEL 386SX/33 MHz	1.390	VGA 16 BITNA 512K KARTICA	120
MODEL 386/40 MHz/64K CACHE	1.790	VGA 16 BITNA 1Mb KARTICA	250
MODEL 486/33 MHz/256K CACHE	2.990	AT IDE KONTROLER	50
MODEL 486/50 MHz/256K CACHE	3.490	KOMBI KONTROLER	100
<i>Osnovne konfiguracije sadrže:</i>		1Mb RAM MODUL	100
<i>40 Mb HDD, flopi disk 1,2 Mb, tastaturu, desk top kućište,</i>		256K RAM MODUL	25
<i>monitor mono herkules 14" crno beli, 2S/1P port</i>		TASTATURA KLIK 101	100
<i>model 286 1 Mb RAM, 386SX 2 Mb RAM, 386 i 486 4 Mb RAM</i>		FLOPI DISK 1,2 Mb 5,25"	150
		FLOPI DISK 1,44 Mb 3,5"	120
DOPLATE			
HARD DISK 80 Mb	130	HARD DISK 40 Mb	390
HARD DISK 105 Mb	190	HARD DISK 80 Mb	520
HARD DISK 120 Mb	240	HARD DISK 105 Mb	580
HARD DISK 210 Mb	490	HARD DISK 120 Mb	630
VGA MONO MONITOR 14" SA VGA KARTICOM	130	HARD DISK 210 Mb	880
KOLOR MONITOR SVGA 1024 X 768 14" 512 kb	430	MONO MONITOR (Herkules)	270
KOLOR MONITOR SVGA 1024 X 768 14" 1 Mb	530	VGA MONO MONITOR	300
KUĆIŠTE MINI TOWER	20	KOLOR MONITOR SVGA 1024x768 14"	600
OSTALA OPREMA			
EPSON LX-400/800	480	MIŠ SA PODNOŽJEM	60
EPSON FX-1170	1490	ETHERNET KARTICA 16 BIT	290
EPSON LQ-1170	1790	STRIMER COLORADO JUMBO INTERNI 120 Mb	590
LASERSKI ŠTAMPAČ HP III P	3400	STRIMER COLORADO JUMBO INTERNI 250 Mb	690
LASERSKI ŠTAMPAČ HP IV	4300	STRIMER ARCHIVE INTERNI 150 Mb	1800
KABL ZA ŠTAMPAČ	30	387-33 KOPROCESOR	250
YU SET ZA LX MODELE	50	MODEM INTERNI 2400	250
KUĆIŠTE DESK TOP	170	FAX/MODEM INTERNI 2400/9600	300
KUĆIŠTE MINI TOWER	190	UPS 500VA	980
KUĆIŠTE MIDI TOWER	250	TONER ZA HP III	250
KUĆIŠTE TOWER	310	TRAKE ZA STRIMER	100
MATIČNA PLOČA 286-20	180	DISKETE 5,25" 1,2 Mb	25
MATIČNA PLOČA 386SX-33	310	YU SET ZA HERKULES	30
MATIČNA PLOČA 386-40 CACHE 64K	500	FILTER ZA MONITOR MREŽNI	40
MATIČNA PLOČA 486-33 CACHE 256K	1450		
MATIČNA PLOČA 486-50 CACHE 256K	1950		
I/O KARTICA 1PAR/2SER	50		

**RADNO VREME OD 9 DO 17 ČASOVA, ISPORUKA ODMAH PO UPLATI
CENE SU U DEM, DINARSKA UPLATA PO DNEVNOM KURSU NA TRŽIŠTU U BEOGRADU
GARANCIJA 12 MESECI, SERVIS OBEZBEĐEN**

Dosegnite OLYMP računarstva

Virtual Library of Mathematics - University of Belgrade
e library: ma.f.bg.ac.rs



OLYMP
electronic

tel. 011/400-477
fax 011/410-240

11000 BEOGRAD, Jovana Đaje 10



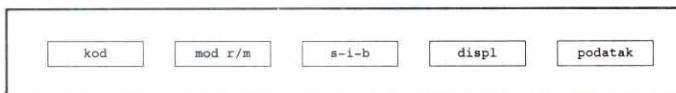
ADRESNI MODOVI

Pošto smo upoznali arhitekturu mikroprocesora 80486, vreme je da polako pređemo na njegov 32-bitni set instrukcija. Počecemo, kao i obično, od modova adresiranja...

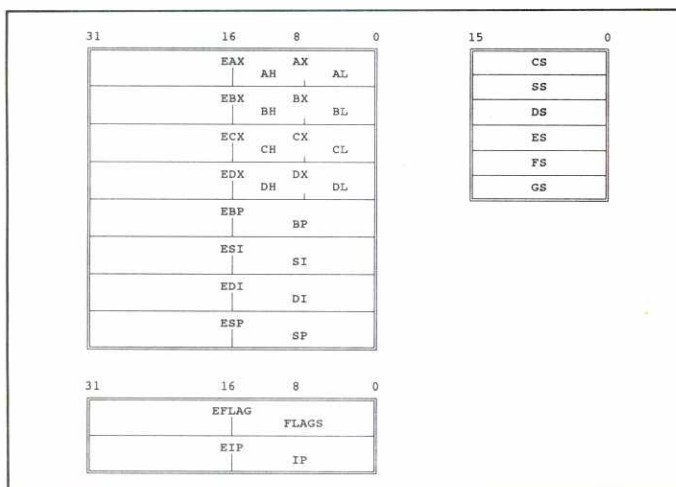
Dejan Ristanović

U prošlim „Računarima“ detaljno smo upoznali načine na koje mikroprocesor komunicira sa memorijom. Njihov zajednički sadržalac svakako je linearna 32-bitna adresa, koja adresnom magistralom putuje od mikroprocesora do memorije. Što se hardvera tiče, ta linearna adresa je zbilja jedino što je potrebno da bi se pribavio (ili upisao) željeni podatak. Sa softverske strane, međutim, stvari su mnogo složenije – za programiranje i, naročito, prevođenje programa pisanih na višim jezicima potrebni su najrazličitiji adresni modovi, od sasvim jednostavnog navođenja adrese do veoma složenih višestrukih indeksiranja. Iako nema sumnje da 80486 predstavlja veoma savremen mikroprocesor, njegova arhitektura je klasično fon-nojmanovska – instrukcije i podaci dele istu memoriju a programi se izvršavaju linijski, uz izvesno preklapanje izvršenja naredbe sa pripremom narednih, što se obavlja potpuno transparentno za korisnika. Zato su adresni modovi prilično klasični i verujemo da će biti razumljivi svakome ko je ikada programirao na bilo kom „Intel“-ovom ili „Zilog“-ovom mikroprocesoru.

Na slici 1 prikazan je globalni način kodiranja instrukcija mikroprocesora 80486. Prvi bajt (ili prva dva bajta) su kod instrukcije, *r/m* i *s-i-b* (po bajt) opisuju modove adresiranja, *displ* (*displacement*) je deo adrese i može se prostirati na 1, 2 ili 4 bajta, dok je *data* podatak iste dužine. Jedino je kod instrukcije obavezan – zavisan od tipa, neke instrukcije neće uopšte imati adresu ili podatke. Instrukcija XCHG EAX, EBX, na primer, ima samo kod i *r/m* polja, dok su kod instrukcije ADD [EBP+2][ESI*4], 123 zastupljena sva polja sa slike 1. Kao dodatak, ispred nekih instrukcija pojavljuje se prefiks ili čak više prefiksa – neki veoma složeni oblici instrukcija mogu imati i četiri prefiksa!



Slika 1: Kodiranje instrukcija mikroprocesora 80486



Slika 2: Registri opšte namene

U okviru ove serije tekstova nećemo se preterano baviti načinom na koji se instrukcije upisuju u memoriju, pošto je taj način malo kome bitan – većina programera će pisati svoje programe na assembleru i onda ih, primenom MASM-a ili nekog konkurentskog programa, automatski prevesti u izvršni kod. Srećna okolnost je što su razni komercijalno raspoloživi asemblerski programi za „Intel“-ove mikroprocesore relativno dobro standardizovani, tako da se sve bitne stvari unose na isti način što, recimo, nije bio slučaj kod mikroprocesora Z-80. Ako razlike i postoje, one se svode na neke dodatne asemblerske direktive koje se i tako relativno retko koriste.

Implicitno adresiranje

Najjednostavniji adresni mod, prisutan kod svih mikroprocesora, je implicitno (podrazumevano) adresiranje, što znači da instrukcija nema adresu niti joj je ona potrebna. Postoji čitava klasa instrukcija koje obavljaju precizno definisan posao koji uopšte ne zahteva komunikaciju sa memorijom – dobar primer bi bila instrukcija STI, koja omogućava interapte ili CLD, koja briše *direction* flag. Implicitno adresiranje se ne odnosi uvek na flegove, nego i na sadržaje registara – instrukcija AAA, na primer, prilagođava sadržaj akumulatora posle ASCII sabiranja.

Registarsko adresiranje

Instrukcija AAA operiše sa sadržajem registra, ali je to tačno preciziran registar. Većina instrukcija, naravno, može da pristupa raznim registrima, što znači da se u okviru same naredbe navodi na koje registre treba da deluje.

To je registarsko adresiranje zastupljeno kod naredbi tipa MOV AX,BX, INC ESI, CALL EDI (poziv potprograma čija je adresa u registru EDI), ADD EDX, EBX, itd. Na slici 2 prikazana je „mapa“

registara opšte namene koje registarsko adresiranje „prepoznaje“. Vidimo da su neki registri, iz razloga kompatibilnosti, podeljeni na polovine i četvrtine, kojima se može slobodno pristupati. MOV EAX, EBX, na primer, prenosi 32 bita, MOV AX, BX 16 bita a MOV AL,BL samo osam bita podataka iz registra EBX u EA

Neposredno adresiranje

Kod neposrednog (*immediate*) adresiranja, podatak se nalazi u okviru same instrukcije. Naredba MOV EAX, 10, na primer, prenosi konstantu 10 u registar EAX, dok JC 3C0h preskače sledećih 960 bajtova ako je *carry* flag setovan. U instrukcijama, naravno, ne figurišu samo ovako „velike“ adrese – neposrednim adresiranjem smatra se i instrukcija nalik na BT EAX, 3, koja kopira treći bit registra EAX u *carry*.

Kod primene neposrednog adresiranja, konstante je pogodno izražavati u heksadekadnom obliku, ali se povremeno ukazuje potreba i za dekadnim i binarnim brojevima. Samom mikroprocesoru je to, naravno, potpuno svejedno, jer sav posao obavi assembler – uobičajeno je da dekadne konstante prati sufiks D (npr. MOV AX, 123D), heksadekadne H, a binarne B. Ukoliko se sufiks ne navede, podrazumeva se D. Neki assembleri ugrađeni u više programske jezike mogu koristiti drugu konvenciju – recimo, da heksadekadnom broju prethodi prefiks \$ ili &.

Specijalni oblik neposrednog adresiranja je adresiranje periferije, *I/O operands* u „Intel“-ovoj terminologiji. Mikroprocesor 80486 ima 65536 I/O portova, pri čemu je komunikacija sa njima 8-bitna, 16-bitna ili 32-bitna. 16-bitne i 32-bitne adrese moraju biti „uravnate“ na početak reči (deljive sa 2) odnosno duge reči (deljive sa 4). Na primer, OUT 08h,AL prenosi podatak iz AL na port 8h, dok IN EAX, DX prenosi u EAX 32-bita sa porta čiji je broj upisan u DX. Treba još znati da oznake porta, ukoliko ih specificirate u okviru instrukcije, ne smeju preći 255 – portovima 256-65535 morate da pristupate indirektno, kao u prethodnom primeru.

Dejstvo pojedinih IN i OUT instrukcija ne zavisi od samog mikroprocesora nego od konfiguracije sistema, tj. periferije koja je (ako je)

```
brojač DD 10 ; 32 bita za BROJAČ, inicijalna vrednost 10
flag DW ? ; 16 bita za FLAG
ime DB 20 DUP (?) ; 20 bajta za IME
DEC DEC brojač ; umanjuje BROJAČ za 1
MOV MOV AH, ime ; upisuje IME[0] (prvi bajt niza) u AH
MOV MOV AL, ime[1] ; upisuje IME[1] (drugi bajt niza) u AL
OR OR flag, 2000h ; setuje bit u deljivi FLAG
```

Slika 3: Direktno adresiranje

```
koord STRUC ; definiše oblik sloga
x DD ?
y DD ?
z DD ?
koord ENDS
tačka koord<> ; rezerviše memoriju za slog
LEA ESI, tačka ; adresa strukture u ESI
MOV MOV [ESI].x ; x komponenta u EAX
DEC DEC [ESI].y ; umanjuje y komponentu za 1
MOV MOV [ESI].x, [ESI].z ; kopira z u X
```

Slika 4: Adresiranje baza plus indeks - pristup strukturama podataka

```

int V[MAX_ELEM];
register int i;
suma = 0;
for (i=0; i<MAX_ELEM; i++) suma+=v[i]*v[i];

prog: XOR   ECX, ECX      ; brojač i (ECX) dobija vrednost 0
      MOV   suma, ECX  ; suma=0
petlja: CMP   ECX, MAX_ELEM ; da li je brojač veći od MAX_ELEM?
      JGE   kraj      ; ako jeste, kraj petlje
      MOV   EAX, ECX   ; brojač u EAX
      SHL   EAX, 2     ; množi brojač sa 4
      MOV   EAX, V[EAX] ; element niza u EAX
      IMUL  EAX        ; kvadrira sadržaj EAX
      ADD   suma, EAX  ; računa sumu
      INC   ECX        ; uvećava brojač
      JMP   petlja    ; nastavak rada u petlji

kraj:  ...

prog: XOR   ECX, ECX      ; brojač i (ECX) dobija vrednost 0
      MOV   suma, ECX  ; suma=0
petlja: CMP   ECX, MAX_ELEM ; da li je brojač veći od MAX_ELEM?
      JGE   kraj      ; ako jeste, kraj petlje
      MOV   EAX, V[ECX*4] ; element niza direktno u EAX
      IMUL  EAX        ; kvadrira sadržaj EAX
      ADD   suma, EAX  ; računa sumu
      INC   ECX        ; uvećava brojač
      JMP   petlja    ; nastavak rada u petlji

kraj:  ...
    
```

Slika 5: C program koji računa zbir kvadrata i dva moguća prevoda

povezana sa tim portom. Neoprezno slanje podataka na portove može da izazove nepredvidljive posledice, koje obuhvataju „rušenje“ sistema ili čak kvar na opremi. Zato većina modernih operativnih sistema (npr. OS/2) omogućava samo privilegovanim programima da direktno opšte sa hardverom na ovaj način.

Direktno adresiranje

Sva prethodna adresiranja i nisu adresiranja u užem smislu te reči – kakvo je to adresiranje ako se ne radi sa memorijom? Direktno (ili apsolutno, mada taj termin baš ne odgovara „Intel“-voj segmentiranoj arhitekturi) adresiranje obezbeđuje komunikaciju između memorije i registara na najjednostavniji mogući način: instrukcijom MOV AL, [12345Ah], na primer, prenosimo podatak iz ćelije čija je adresa 12345Ah u registar AL.

Ukoliko naredbu napišemo bez pominjanja segmenta, ona se implicitno odnosi na segment sa podacima (*data segment*) na koji ukazuje registar DS sa slike 2. Često treba pristupiti nekom drugom segmentu – u tom slučaju se piše ime njegovog registra i dvotačka, na primer MOV AL, ES:[12345Ah]. S obzirom da se ova instrukcija u memoriji kodira u obliku prefiks+kod, neki stari asembleri su zahtevali njeno pisanje u obliku:

```

ES:
MOV AL,[1235Ah]
    
```

pa je taj oblik prisutan i kod novijih programa tog tipa, zbog potrebe za kompatibilnošću. U praksi se, međutim, uvek piše MOV AL, ES:[12345Ah].

Veličinom registra određena je zona memorije na koju instrukcija deluje – da smo koristili MOV EAX, [12345Ah], bila bi prenesena 32 bita a ne samo 8 kao u našem primeru. Kod nekih instrukcija koje operišu direktno sa memorijom, ova koncepcija ne daje rezultate – kako bi, na primer, mikroprocesor znao koliko bajtova treba da šiftuje kada bi naišao na instrukciju SHL [2000h], 3? Zato se u ovakvim slučajevima koristi BYTE PTR, WORD PTR ili DWORD PTR: instrukcija SHL BYTE PTR [2000h], 3, na primer, šiftuje bajt memorije za tri bita ulevo, dok bi INC DWORD PTR [2000h] delovalo ne samo na lokaciju 2000h nego i na tri sledeće, posmatrane kao *longint* odnosno *double word*.

U praksi se apsolutne adrese veoma retko koriste, pošto bi učinile program mašinski zavinsnim i teškim za održavanje. Uspešno ih zamenjuju labele. Primer sa slike 3 najpre reserviše 32 bita za polje BROJAČ, zatim 16 bita za FLAG i najzad 20 puta po bajt za polje IME – prvom polju je dodeljena konkretna vrednost

a ostala nisu inicijalizovana, što je opisano upitnikom (vrednost nebitna). Zatim se koriste instrukcije nalik na DEC BROJAČ ili MOV AL, IME[5] – daleko su preglednije od velikih nizova heksadekadnih cifara!

Bazno adresiranje

Kod baznog adresiranja adresa podatka se nalazi u registru – zato se kod nekih drugih mikroprocesora ono naziva *indirektno* adresiranje (kao neka vrsta suprotnosti opisanom direktnom adresiranju). Instrukcija MOV AL, [ECX], na primer, dovodi u AL 8 bita upisanih u memorijsku ćeliju *data segment*-a, čija je adresa upisana u ECX. Ukoliko je, na primer, u ECX prethodno upisan broj 1234h, MOV AL, [ECX] biće ekvivalentno sa MOV AL, [1234h]. Na ovaj način se takođe može direktno pristupiti memoriji – INC WORD PTR [EAX] će, na primer, uvećati za jedan reč koja je upisana u ćelije na koje ukazuje EAX.

Pri korišćenju ove instrukcije treba obrati pažnju na važan izuzetak: ukoliko koristite registre ESP i EBP, instrukcija **neće** delovati na podatak koji se nalazi u *data segment*-u nego na odgovarajuću adresu *stack segment*-a. Što se tiče ESP, stvar je sasvim prirodna, pošto se on bavi stekom. EBP je ovako „usmeren“ zato što se pri prevodenju programa pisanih na višim jezicima koristi za adresiranje argumenta potprograma, a ti argumenti se obično prenose baš preko steka.

Adresiranje baza plus ofset

Bazno adresiranje je veoma logično jer „goda“ ćeliju na koju ukazuje registar. Ponekad je, međutim, pogodno da registar ukazuje na početak neke fiksne strukture, dok se u okviru same instrukcije navodi rastojanje od početka te strukture do željenog podatka. Pogledajmo, na primer, sliku 4: definisana je struktura KOORD koja ima tri komponente – X, Y i Z koordinatu neke tačke. Zatim je za tu strukturu rezervisan prostor u memoriji (KOORD<>), efektivna adresa tog prostora upisana u ESI (LEA = *Load Effective Address*) i, najzad, njenim komponentama je pristupano sa MOV EAX, [ESI].X odnosno [ESI].Y ili [ESI].Z.

Neka vas ne zavara to što se pominju nekakve „promenljive“ X, Y i Z – radi se naprosto o simboličkim imenima pojedinih komponenti strukture, pri čemu asembler, već u trenutku prevodenja, „zna“ da je X=0, Y=4 i Z=8, pa se ovi fiksni podaci ugrađuju u polje *displ* (u „Intel“-ovoj terminologiji se ofset zove *displacement*) odgovarajućih instrukcija. Ofset je, dakle, kod ovog adresiranja **fiksna**.

Instrukcija	Opis
ADD [ECX][EBP], 12	EBP je bazni registar, koristi se SS
MOV AX, NIZ[EBP]	EBP je bazni registar, koristi se SS
MOV EAX, [ECX][EBP*4]	ECX je bazni registar, koristi se DS
INC WORD PTR [ECX*8][EBP].X	EBP je bazni registar, koristi se SS

Slika 6: Korišćenje registra EBP kod adresiranja 'indeks plus ofset'

Instrukcija	Opis
MOV EAX, [EBP+8][ESI]	Matrica na steku počinje na EBP+8
INC WORD PTR [EBX+EAX*2]	Vektor 16-bitnih podataka na EBX
MOV ECX, KOORD[EAX*8][ESI].Y	Niz struktura KOORD

Slika 7: Adresiranje 'baza plus indeks plus ofset'

Adresiranje indeks plus ofset

Situacije u kojima je ofset fiksna su u praksi relativno retke – mnogo je češći slučaj da je početak nekog niza na fiksnom mestu i da onda treba da pristupimo nekom njegovom elementu čija se adresa menja u petlji. Brojač te petlje će, naravno, biti neki od registara, u praksi se često koristi ESI. Pogledajmo, na primer, instrukciju MOV AH, 2A50h[ESI] – po naitasku na nju, mikroprocesor uzima sadržaj registra ESI, uvećava ga za 2A50h (ovo sabiranje se, prirodno, obavlja u nekom internom registru 80486 – sadržaj ESI će po izvršenju instrukcije biti očuvan) i tako dobija adresu memorijske ćelije u DS segmentu čiji će sadržaj preneti u AH. Svi registri opšte namene **osim** ESP mogu da se koriste kao indeksni.

U praksi se, naravno, vrlo retko navode konkretne adrese – umesto toga, rezervisacemo prostor za niz VEKTOR sa VEKTOR: DD 100 DUP(?), a onda mu pristupati sa MOV EAX, VEKTOR[ESI] ili SUB VEKTOR[EAX], 8.

Prema onome što smo do sada napisali, reklo bi se da se adresiranje „indeks plus ofset“ može svesti na „baza plus ofset“ – sadržaj registra se i u jednom i u drugom slučaju sabira sa nekom konstantom. Pa ipak, postoji jedna suptilna razlika koju ćemo upoznati studirajući C program sa slike 5 i dva njegova (moguća) prevoda na mašinski jezik.

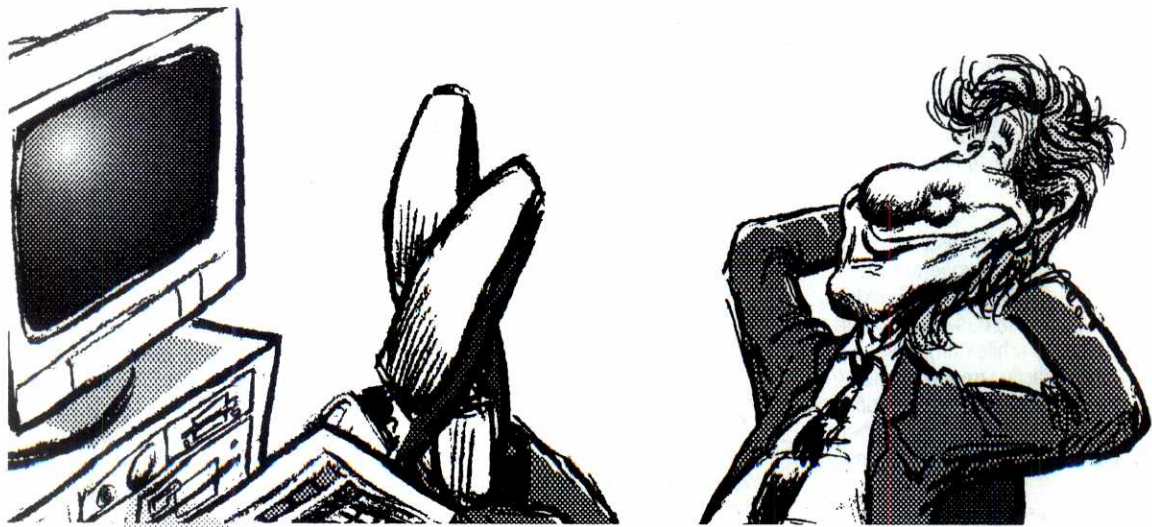
Program sa slike 5 računa zbir kvadrata niza (recimo 32-bitnih) brojeva. U radnoj petlji se, zapravo, menjaju dve vrednosti – indeks *i* i ofset elementa *v[i]* – kada je, na primer, *i*=5, ofset bi trebalo da bude 5*4=20, pošto svaki element zauzima po četiri bajta. Na nekom primitivnijem mikroprocesoru to bi se moralo svesti na množenje, kao u prvom prevodu sa slike 5. Drugi prevod, međutim, koristi jednu specifičnost mikroprocesora 80386/80486 – MOV EAX, V[ECX*4] „množi“ sadržaj ECX sa 4 i direktno pristupa potrebnom bajtu memorije, a opet se ta instrukcija izvrši za **isto** vreme kao MOV EAX, V[ECX], što znači da drugi prevod obezbeđuje daleko racionalniji rad. Ovo „množenje“ se u „Intel“-ovoj terminologiji zove skaliranje (*scaling*) i ograničeno je na vrednosti 1, 2, 4 i 8, pri čemu, jasno, množenje sa 1 nema mnogo smisla.

Videli smo da se za indeksirano adresiranje ne može koristiti ESP, dok EBP i ovde predstavlja izvestan izuzetak – dok je kod baznog adresiranja delovao isključivo na stek segment, sada se pod odgovarajućim uslovima odnosi i na DS (slika 6). Ako je u instrukciji naveden bazni i indeks-registar i ako je jedan od njih EBP, smatra se da je EBP bazni registar ukoliko nije naveden i skalirajući faktor.

УНЕСИТЕ

ПРОМЕНЕ

У СВОЈ НАЧИН ПОСЛОВАЊА



93. gazdaru

AB soft radi za Vas

KNJIGOVODSTVENI SOFTVERSKI PAKETI: FIPO (FINANSIJSKO KNJIGOVODSTVO)

LIDO (LIČNI DOHOCI) • ROK (ROBNO KNJIGOVODSTVO) • MAT (MATERIJALNO KNJIGOVODSTVO)

MAGIC (MAGACINSKO POSLOVANJE) • OSA (EVIDENCIJA OSNOVNIH SREDSTAVA) • VIR (VIRMANI)

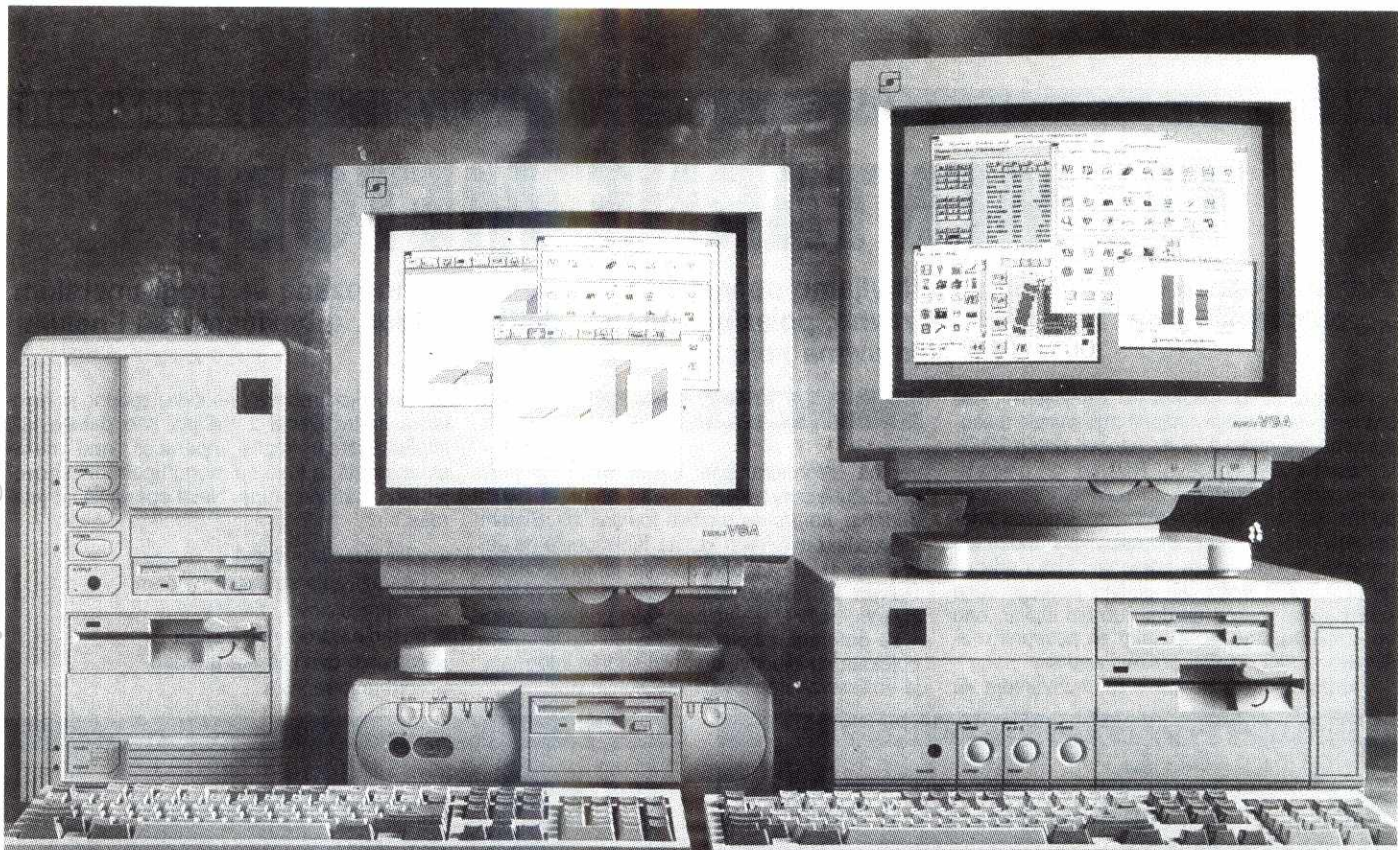
BLAD (BLAGAJNA) • GRACE (EVIDENCIJA ČEKOVA) • KOM (POSLOVI KOMERCIJALE) • OKAM (OBRAČUN KAMATA)

CAR (PRODAJA VOZILA) • TEP (TEHNIČKI PREGLED VOZILA) • EVA (EVIDENCIJA AMBALAŽE)

MAP (MALOPRODAJA) • TAG (POSLOVANJE TURISTIČKIH AGENCIJA)

RAČUNARSKA OPREMA, ŠTAMPAČI, MREŽE...

AB SOFT, BEOGRAD, Kneza Miloša 82, tel/fax 656-857, tel 644-255 lok. 113



PC-AT 386/SX 33, HDD 105MB, FDD 3.5", 2MB RAM, SVGA COLOR 1950 DEM.

PC-AT 386/DX 40, 128 CASH, HDD 120MB, FDD 3.5", 5.25", SVGA COLOR 2550 DEM.

PC-AT 486/DX 33, 256 CASH, HDD 215MB, FDD 3.5", 5.25", SVGA COLOR 3500 DEM.

SVE VRSTE KOMPJUTERSKIH PERIFERIJA!

POSEBNI POPUSTI ZA DALJU PRODAJU!

1950 DEM.

2550 DEM.

3500 DEM.

BIBO
COMPUTER SYSTEMS
BEOGRADSKA 41
TEL. 011-341-392, 496-351



286 / 20 MHz	1 MB	1,450
386 sx / 25 MHz	2 MB	1,550
386 / 40 MHz	4 MB	1,990
486 / 33 MHz	4 MB	2,900
486 / 50 MHz	4 MB	3,400

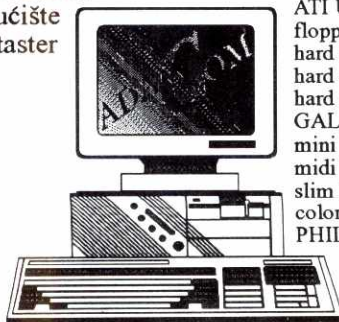
fax/modem interni DISCOVERY	400,-
fax/modem externi DISCOVERY	600,-
modem interni DISCOVERY	350,-
modem externi DISCOVERY	400,-
Grafička kartica ATI ULTRA	900,-
Grafička kartica S3 Windows	550,-
Joystick MS-500	50,-
koprocesor 287-20 MHz	100,-
koprocesor 387-20 MHz INTEL	250,-
koprocesor 387-40 MHz IIT	250,-
Miš GM-6	50,-
Miš GM-6000	80,-
memorija za IIP/IIIP	350,-
Mobile rack za 3 1/2" hard disk	160,-
Monitor filter stakleni	65,-
Mouse pad	10,-
traka za LX-400	15,-
traka za LQ-550	15,-
traka za FX-1050	15,-
toner za IIP/IIIP	250,-
štampanje HP IIIP bez tonera	2900,-
štampanje EPSON LX-400	550,-
štampanje EPSON FX-1050	1350,-
UPS 550 VA	600,-

STANDARD ZA KONFIGURACIJU

- hard disk CONNER 80 MB 17 ms
- floppy disk PANASONIC 5 1/4"
- combi controller (2S, 2P, 1G)
- monohromatski VGA monitor
- TRIDENT 9000 512k
- AT desktop kućište
- tastatura 101 taster

DOPLATA ZA

- 1 MB memory SIMM/SIPP 75,-
- TRIDENT SVGA 8900 1 MB 50,-
- OAK 087 1 MB 75,-
- TSENG ET4000 VESA 1 MB 100,-
- TSENG ET4000 enhanced 32k boja 180,-
- S3 windows akcelerator 400,-
- ATI ULTRA sa koprocesorom i mišem 750,-
- floppy disk 3 1/2" 1.44 MB 130,-
- hard disk CONNER 120 MB 19 ms 130,-
- hard disk CONNER 170 MB 16 ms 260,-
- hard disk CONNER 210 MB 13 ms 550,-
- GALAXY IDE cache controller 550,-
- mini TOWER 200 W 30,-
- midi TOWER 200 W 70,-
- slim line 70,-
- color monitor TOPFLY 1024x768 450,-
- PHILIPS BRILLIANCE 7CM3209 700,-



Tel: 629-233

Fax: 629-672

BEOGRAD, Čika Ljubina 12



ELEMENTARNE METODE

Seriju tekstova o algoritmima nastavljamo operacijom koja je najuže povezana sa programerskim problemima – pretraživanjem. Naravno, polazimo od najprostijih metoda, a bavićemo se i nekim pojmovima koje smo već pominjali.

Normalno, pretraživanje zamišljamo kao proces pronalazjenja tražene informacije u skupu datih informacija, a na osnovu datog ključa iz nekog skupa ključeva. Recimo, pri pretraživanju rečnika, ključevi su reči, a informacije podaci koji se dobijaju (sinonimi, izgovor, značenje). U telefonskom imeniku, ključ može biti prezime, a informacija telefonski broj i adresa (ili obrnuto?). Pošto se obično očekuje veća količina informacija srodnih datom ključu, one se grupišu u zapise, a zatim se posebnim rešenjem iz njih dobijaju traženi podaci.

Aplikacije koje se bave pretraživanjem su

Bojan Petrović

veoma rasprostranjene: banka mora da prati sve račune svojih klijenata i, stalno pretražujući podatke, kontroliše nastale transakcije. Sistem za prodaju avionskih karata funkcioniše na sličan način, uz mnoge specifične mogućnosti. Ustvari, za savladavanje raznih metoda pretraživanja, najbolji je već pomenuti primer rečnika – za program, tabela simbola je rečnik: imena objekata su ključevi, a zapisi sadrže informacije vezane za imenovani objekat.

Kao i kod redova prioriteta, zgodno je algoritme pretraživanja zamišljati kao pakete koji implementiraju razne operacije nad datim strukturama, a koje se mogu upotrebiti i odvojeno od matične grupe. Neke operacije od interesa su:

- inicijalizacija rečnika
- pretraživanje sa datim ključem
- umetanje novog zapisa
- brisanje određenog zapisa
- spajanje dva rečnika u jedan
- sortiranje rečnika

Listing 1

```
struct node
{ int kljuc; int info; };

struct node a[MAXN+1];
int N;

void seq_init(void)
{ N=0; };

int seq_search ( int v )
{
    int x=N+1;
    a[0].kljuc=v;
    a[0].info=-1;
    while (v!=a[--x].kljuc);
    return a[x].info;
}

void seq_insert ( int v, int info )
{
    a[++N].kljuc=v;
    a[N].info=info;
}
```

Pretraživanje pomoću niza

Listing 2

```
struct node
{ int kljuc, info; struct node *dalje; };

struct node *glava, *z;

void list_init(void)
{
    glava=(struct node *) malloc(sizeof *glava);
    z=(struct node *) malloc(sizeof *z);
    glava->dalje=z;
    z->dalje=z;
    z->info=-1;
}

int list_search ( int v )
{
    struct node *t = glava;
    z->kljuc=v;
    while (v>t->kljuc) t=t->dalje;
    if (v!=t->kljuc) return z->info;
    return t->info;
}

void list_insert ( int v, int info )
{
    struct node *x, *t = glava;
    z->kljuc=v;
    while (v>t->dalje->kljuc) t=t->dalje;
    x=(struct node *) malloc(sizeof *x);
    x->dalje=t->dalje;
    t->dalje=x;
    x->kljuc=v;
    x->info=info;
}
```

Pretraživanje pomoću povezanih listi

Listing 3

```
int bin_search( int v )
{
    int l=1; int r=N; int x;
    while (r>=l)
    {
        x=(l+r)/2;
        if (v<a[x].kljuc)
            r=x-1;
        else
            l=x+1;
        if (v==a[x].kljuc)
            return a[x].info;
    }
    return -1;
}
```

Binarno pretraživanje

Listing 4

```
struct node
{ int kljuc, info; struct node *l, *r; };

struct node *t, *glava, *z;

void stablo_init(void)
{
    z=(struct node *) malloc(sizeof *z);
    z->l=z;
    z->r=z;
    z->info=-1;
    glava=(struct node *) malloc(sizeof *glava);
    glava->r=z;
    glava->kljuc=0;
}

void stablo_insert( int v, int info )
{
    struct node *p, *x;
    p=glava;
    x=glava->r;
    while (x!=z)
    { p=x; x=(v<x->kljuc) ? x->l : x->r; }
    x=(struct node *) malloc(sizeof *x);
    x->kljuc=v;
    x->info=info;
    x->l=z;
    x->r=z;
    if (v<p->kljuc) p->l=x; else p->r=x;
}

int stablo_search ( int v )
{
    struct node *x = glava->r;
    z->kljuc=v;
    while (v!=x->kljuc)
        x=(v<x->kljuc) ? x->l : x->r;
    return x->info;
}
```

Pretraživanje pomoću binarnih stabala

Listing 5

```

void stablo_delete ( int v )
{
    struct node *c, *p, *x;
    z->kljuc=v;
    p=glava;
    x=glava->r;
    while (v!=x->kljuc)
    { p=x; x=(v<x->kljuc) ? x->l : x->r; };
    t=x;
    if (t->r==z) x=x->l;
    else
    if (t->r->l==z)
    { x=x->r; x->l=t->l; }
    else
    {
        c=x->r;
        while (c->l->l != z). c=c->l;
        x=c->l;
        c->l=x->r;
        x->l=t->l;
        x->r=t->r;
    }
    free(t);
    if (v<p->kljuc) p->l=x; else p->r=x;
}

```

Brisanje zapisa u binarnom stablu

U pojedinim slučajevima, pokušaćemo da neke operacije spojimo u jednu, pošto tako može biti zgodnije: recimo, fuzija umetanja i sortiranja može biti od velike pomoći kod pojave duplikata ključeva.

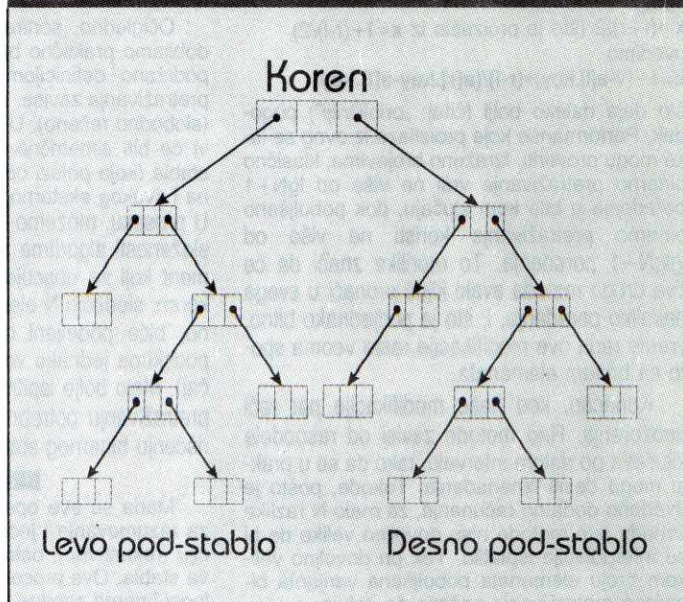
Ovu pojavu možemo obraditi na više načina. Pre svega, možemo jednostavno insistirati da naš rečnik sadrži samo zapise sa različitim ključevima, a da ti zapisi sadrže i pokazivače na povezanu listu zapisa sa odgovarajućim ključem. Ovde, dakle, jednim pretraživanjem indirektno dobijamo sve zapise sa datim ključem. Drugi način je da sve duplikatne zapise ostavimo u rečniku i vratimo prvi na koji naletimo. Sada moramo obezbediti dodatni algoritam za pronalaženje preostalih, što je sve zajedno dobro za aplikacije koje rade sa pojedinačnim zapisima, ali loše sa programerskog stanovišta. Treća mogućnost je uvođenje posebnih identifikatora za svaki zapis, na osnovu koga ćemo u skupu zapisa sa istim ključem naći odgovarajući. I tako dalje. Mi ćemo nadalje podrazumevati mogućnost pojave zapisa sa istim ključevima i nećemo tome eksplicitno posvetiti pažnju.

SEKVENCIJALNO

Najjednostavniji metod je smeštanje zapisa u niz. Novi zapisi se dodaju na kraj niza, a pretraživanje vršimo sekvencijalno (redom). Implementacija na listingu jedan podrazumeva ključevu („ključ“) i informacije („info“) tipa celobrojnih veličina. Mada će stvarne potrebe sa svim sigurno zahtevati komplikovanije zapise i ključevu, to neće imati nekog uticaja na suštinu stvari (promena `int ključ` u `char *ključ`, vodi do promene u `while` izrazu: `while (strcmp(v,a[-x].ključ))`, čime smo uveli stringove).

U datoj implementaciji koristili smo jednostavan način da obezbedimo da će se pretraživanje uvek završiti, a da pri tom koristimo samo jedno ispitivanje. Umesto da u jednom trenutku proveravamo da li je `x==0`, a u drugom da li je `a[-x]` traženi ključ, spojili smo ih u jedno: u `a[0]` smo postavili traženi ključ. Ovo je analogija sa našim ranijim slučajevima kada smo kao ograničenje koristili najmanji ili najveći ključ. Istovremeno, eliminišući jedno ispitivanje, smanjili smo unutrašnju petlju, što, po ranijim razmatranjima, drastično može ubrzati rad algoritma.

Slika 1



Binarno stablo

Sekvencijalno pretraživanje korišćenjem niza „troši“ $N+1$ poređenje za neuspešno, i, u proseku, $N/2$ poređenja za uspešno pretraživanje. U prvom slučaju, zaključak sledi direktno iz koda: svaki zapis mora biti obrađen da bi se ustanovilo nije li naš ključ prisutan (ovo nije čudno objašnjenje – šta ako je niz sortiran? da li tada moramo svaki zapis proveriti?). U drugom slučaju podrazumevamo da je podjednako verovatno da će se zapis naći na bilo kom mestu u nizu, pa sledi:

$$1 \quad N+1 \\ - (1+2+3+\dots+n) = \frac{N}{2}$$

što je u skladu sa gornjim.

Prirodno unapređenje sekvencijalnog pretraživanja pomoću niza bi bilo uvođenje povezanih listi. Naravno, ne zato što je sa njima lakše operisati, već zato što je pomoću njih lako održati niz sortiranim. Jedna moguća implementacija data je listingom dva. Pošto je lista sortirana, osnovni efekat je, kao što je već nagovešteno, mogućnost ranog prekida pretraživanja. Onog trenutka kada se ispostavi da naš ključ nije manji od ispitivanog, pretraživanje obustavljamo. Sličan efekat smo mogli u prethodnoj implementaciji postići sortiranjem niza posle svakog umetanja, što u nekim aplikacijama može biti korisnije. Opet, pretraživanje ograničavamo uobičajenim postupkom za povezane liste – korišćenjem krajnjeg noda (obično se inicijalizuje na NIL, mada smo mi koristili nešto čudniji metod).

Sekvencijalno pretraživanje korišćenjem povezanih listi „troši“ oko $N/2$ poređenja i u neuspešnom i uspešnom pretraživanju. U prvom slučaju se zaključak izvodi kao i prethodni put, dok je u drugom nešto drugačije: polazi se od pretpostavke da je verovatnoća završetka rada jednaka na krajnjem čvoru i bilo kom od prethodnih.

Naravno, postoje i drugi metodi razmeštanja zapisa u listi. Verovatno najefikasniji je da se raspored vrši prema relativnoj frekvenciji pristupa pojedinim zapisima. Optimalno je, dakle, smestiti na početak one zapise koji se najčešće traže. Ova tehnika, međutim, postiže stvarne efekte tek ako je skup tih najčešće traženih zapisa relativno mali. Ako informacija o frekvenciji pristupa nije dostupna, možemo se poslužiti metodom „samoorganizacije“ niza. Kad god pristupimo nekom zapisu (dakle kada

smo ga „pronašli“), pomerimo ga na početak liste. Ova metoda se mnogo lakše, iz očiglednih razloga, implementira sa povezanim listama.

BINARNO

Jedna od metoda koju smo ranije koristili je bila „podeli-i-pobedi“ („divide-and-conquer“): podeli skup zapisa u dva dela, odredi kom skupu pripada ključ i nastavi sa njim na isti način. Binarno pretraživanje je jedna manifestacija ovakvog načina razmišljanja, a primenjuje se kod većih količina podataka.

Osmotrimo listing tri. Da bismo našli ključ „v“, uporedimo ga sa onim na srednjoj poziciji u nizu. Ako je manji, učinimo isto sa levom polovinom niza, ako je veći sa desnom, u protivnom smo ga našli. Dakle, iz ovog slede dva (očigledna) zaključka: niz mora biti sortiran a metoda je prirodno rekurzivna. Kao i kod QUICK i RADIX EXCHANGE sortova, koristimo pokazivače „r“ i „l“ da ograničimo deo niza koji obrađujemo. Kada taj podniz postane prazan, zaključujemo da je pretraživanje bilo neuspešno. U suprotnom, ili će levi pokazivač postati `x+1` (gde je `x` sredina podniza), ili će desni postati `x-1`, ili će `a[x]` biti baš traženi ključ.

Održavanje niza srtranim može biti bolan posao. Recimo, ubacivanje novog najmanjeg elementa u niz povlači pomeranje svih ostalih za jedno mesto. U proseku, ova metoda troši $N/2$ pomeraja u proseku, što je skupo. To znači da je ova metoda generalno bolja za one slučajeve kada nam je dat već sortirani niz, kada binarno pretraživanje stvarno sija.

Ako se dogodi da postoji više zapisa sa istim ključem, pretraživanje će se završiti negde u bloku sačinjenom od tih zapisa (naravno, ne baš obavezno u sredini). Ako su nam, ko zna zbog čega, potrebni svi, moraćemo da dodatno skeniramo sekvencijalno niz levo i desno od onog pronađenog. Vrlo slično se onda rešava i problem nalaženja zapisa čiji ključevi upadaju u neki interval.

Jedno moguće poboljšanje binarnog pretraživanja je unapređivanje načina pogađanja intervala u kom se traženi ključ (valjda) nalazi. Ideja dolazi od prirodnog načina razmišljanja: kada u telefonskom imeniku tražimo ime koje počinje sa B, tražimo pri početku a ne sredini. Ovaj metod interpolacije zahteva jednu relativ-

no trivijalnu modifikaciju algoritma: umesto ranijeg

$$x = (l+r)/2$$

(što je proizašlo iz $x = l + (r-l)/2$)
 uvodimo

$$x = l + (v-a[l].key) * (r-l) / (a[r].key - a[l].key)$$

što daje daleko bolji (čitaj: „približniji”) pogodak. Performanse koje proizilaze iz ovog se lako mogu proveriti. Izraženo brojevima, klasično binarno pretraživanje vrši ne više od $\lg N + 1$ poređenja u bilo kom slučaju, dok poboljšano binarno pretraživanje koristi ne više od $\lg \lg N + 1$ poređenja. To otprilike znači da će ova druga metoda svaki ključ pronaći u svega nekoliko poređenja, i, što je podjednako bitno, vreme rada ove modifikacije raste veoma sporo sa brojem elemenata.

Konačno, kod naše modifikacije par reči upozorenja. Rad metode zavisi od raspodele ključeva po datom intervalu, tako da se u praksi mogu desiti iznenađenja. Takođe, pošto je uvedeno dodatno računanje, za malo N razlike između dve metode nisu dovoljno velike da bi se interpolacija isplatila. Tek pri dovoljno velikom broju elemenata poboljšana varijanta binarnog pretraživanja počinje da deluje.

BINARNIM STABLOM

O binarnim stablima je već bilo reči u ovoj seriji tekstova, tako da se nećemo zadržavati na njihovom opisivanju. Prisetimo se samo da je osnovna odlika stabala činjenica da svaki čvor stabla ima dva ili manje sinova (leva i desna veza) i jednog i samo jednog oca, sem korena stabla koje ima samo sinove. Binarna stabla se odlikuju dodatnom osobinom da su svi zapisi levo od datog manji od njega, a oni desno veći. Ovakva struktura savršeno odgovara binarnom pretraživanju.

Da bismo našli zapis sa datim ključem, poćemo ga sa onim u korenu stabla. Ako je manji idemo levim podstablom, ako je veći desnim, inače stanimo. Proces se zatim ponavlja rekurzivno na dva podstabla (ili manje). U svakom trenutku, zahvaljujući uslovu koji se poštuje pri formiranju stabla (element manji od datog ide levo, veći desno), garantovano je da će se traženi ključ nalaziti u trenutnom podstablu (ako ga u stablu uopšte ima), i nigde drugde. Procedura se zaustavlja u trenutku kada smo ključ našli, odnosno kada više nema podstabla koje nismo obradili. Upozorimo da je razlika između binarnog pretraživanja i pretraživanja binarnim stablom velika, te da se mora dobro razumeti.

Pri formiranju binarnog stabla, zgodno je da uvedemo dodatni čvor koji će imati samo jednog sina, i to baš koren stabla. U ovaj čvor ćemo staviti neku zgodnu veličinu, recimo najmanju od skupa vrednosti koje očekujemo (recimo nula?). Kao i kod sekvencijalnog pretraživanja, u završni čvor postavljamo traženi ključ, iz potpuno istih razloga. Na taj način nijedno stablo neće stvarno postati prazno, a svako pretraživanje će biti „uspešno”: završni čvor će nam ipak vratiti umesto stvarne informacije neki indikator (minus jedan?).

Prazno stablo će, na osnovu gornjeg, sadržati čvor „glava” koji će pokazivati na završni. Implementacija rada sa ovakvim stablom data je listingom četiri. Završni čvor, kao i ranije, pokazuje na sebe. Da bismo u binarno stablo umetnuli zapis, vršimo neuspešnu pretagu za njim, i na mesto završnog čvora od kog smo stigli postavljamo novi zapis (ključ). U implementaciji na datom listingu, umetanje čvora sa ključem koji već postoji u stablu se vrši desno od njega. To istovremeno znači da potraga za svim zapisima sa istim ključem podrazumeva nalaženje samo onog prvog, dok su ostali po-

ređani jedan za drugim (jedna od mogućih varijanti).

Očigledno, sortiranje sa ovom strukturom dobijamo praktično besplatno, jer je kao takvo podržano definicijom strukture. Performanse pretraživanja zavise, zamislite, od oblika stabla (slobodno rečeno). U najboljem slučaju, čvorovi će biti simetrično raspoređeni oko osovine stabla (koja polazi od korena), a između korena i svakog eksternog čvora biće $\lg N$ čvorova. U proseku, možemo očekivati logaritamski red složenosti algoritma za pretraživanje. Prvi element koji se ubacuje u stablo postaje njegov koren; sledećih N elemenata, generalno gledano, biće podeljeni ovim elementom na dva podskupa jednake veličine (što je idealan slučaj). Malo bolje ispitivanje će nam reći da je u pretraživanju potrebno $2 \ln N$ poređenja pri korišćenju binarnog stabla.

BRISANJE

Mada su sve operacije obrađene gore lake za razumevanje i jednostavne za implementaciju (rekurzivno), ostalo nam je brisanje čvorova stabla. Ova procedura može biti krupan zalogaj i pored zgodne strukture stabala i mogućnosti rekurzije, te smo zbog toga njoj posvetili poseban odeljak.

Brisanje čvora je, naravno, lako ako on nema dece – prosto odgovarajuću vezu njegovog oca prepravimo. Čak i ako postoji samo jedno dete, posao nije težak – umesto datog čvora postavimo to jedno dete. Problem nastaje kod one preostale grupe čvorova, gde svaki ima po dva sina.

Jedna od metoda je zamena čvora koji briše sledećim po veličini. Očigledno je da će takav čvor imati najviše jednog sina, pa ga je lako ukloniti (prema ranije rečenom). Procedura data listingom pet briše iz binarnog stabla prvi čvor sa datim ključem. Varijabla „p” čuva podatak o ocu čvora „x”. Program prvo postavlja vrednost „p” (tražeći mesto „t”) da bi po brisanju čvora „t” mogli da vršimo povezivanje. Zatim se proveravaju mogući slučajevi: ako „t” nema samo desno dete, onda će posle brisanja dete čvora „p” postati levo dete čvora „t”; ako čvor „t” nema samo levo dete, po brisanju će desno dete postati dete čvora „p” uz kopiranje leve veze čvora „t”; ako postoje oba sina, „x” postaje najmanji čvor u desnom podstablu čvora „t”, a desna veza tog čvora se kopira na levo mesto svog oca.

Postoje i razne druge metode za brisanje čvorova u binarnom stablu, mada je gornja uobičajena. Konačno, za pretraživanje je uobičajeno da se najviše vremena potroši na dobro brisanje. U nekim slučajevima, može se primeniti i „veštačko” brisanje, kada čvor nekim markerom samo označimo kao obrisan, ostavljajući ga u strukturi. Veliki broj tako „brisanih” čvorova može voditi do preteranih gubitaka u prostoru ili vremenu, što se u realnim aplikacijama često ne događa. Naravno, stablo se može periodično čistiti od „brisanih” čvorova i time neki kompromis.

I DALJE

Mada su binarna stabla opšte prihvaćena i dobro obrađena, imaju svoje loše osobine. Pojava „najgoreg slučaja” je ono čega se programeri najviše plaše. Setimo se, kod sortiranja, koliko problema se moglo stvoriti sortiranjem već uređenog niza, ili, još bolje, niza koji je sortirano u suprotnom smeru. Tada je, čak i kod tako dobrih metoda kao što je QUICK sort, vreme rada raslo daleko preko nekih mnogo lošijih algoritama (u srednjem slučaju). Kod binarnih stabala se ovo manifestuje kod uslova balansirnosti stabla, o čemu će biti reči u sledećem nastavku. Dakle, pred nama su balansirna stabla.

Nastavak sa strane 75

Ukoliko je skalirajući faktor naveden, smatra se da je EBP indeks-registar pa se referiše DS.

Mikroprocesori 8088 i 8086 su vrlo sporo obrađivali instrukcije koje koriste ovaj adresni mod – ponekad je bilo potrebno čak 17 ciklusa za samo izračunavanje adrese. Zato su programeri stekli naviku da ovo adresiranje, ako je to ikako moguće, izbegavaju. Više nema razloga za to – 80386 i 80486, zahvaljujući znatno bolje rešenoj fazi pripreme naredbi, troše isti broj ciklusa, bez obzira na tip adresiranja. Uz jedan mali ali značajan izuzetak – 80386 troši jedan (ali samo jedan) dodatni ciklus pri obradi instrukcija u kojima figuriše bazni i indeks-registar. 80486 donosi dalja poboljšanja, tako da će ovaj ciklus nekada biti potreban a nekada ne – što je procesor složeniji to je teže precizno proračunati broj mašinskih ciklusa potrebnih za izvršenje nekog programa! 80486 će, međutim, dodati jedan ciklus (koga na 80386 nije bilo) ako je vrednost upisana u bazni registar neposredno pre korišćenja ovog adresiranja. Ukoliko pišete neki kompajler, obratite pažnju da ove dve instrukcije odvojite nekom radnjom koja će vam kasnije biti korisna.

Baza plus indeks plus offset

Poslednje i najkomplikovanije adresiranje koristi se za pristup složenim strukturama smeštenim na stek ili matricama čija se bazna adresa nalazi u nekom registru. Na slici 7 prikazano je nekoliko primera ovog adresiranja koje ćete u praksi jako retko koristiti – reklo bi se da sadrži „jednu dimenziju više” od našeg normalnog razmišljanja. Pa ipak, neki kompajleri će uz njegovu pomoć rado racionalizovati rad sa parametrima.

Poslednji primer sa slike 7 je na prvi pogled vrlo neobičan – reklo bi se da su navedena dva ofseta! Radi se, međutim, samo o pogodnom načinu pisanja instrukcije koju će assembler ispravno prevesti – u polje *displ* odgovarajuće instrukcije biće upisan *zbir* $PT+Y$.

Adresiranje steka

Mikroprocesor 80486 koristi stek na sasvim uobičajen način: instrukcijama PUSH i POP na njega se odlažu i sa njega skidaju sadržaji pojedinih registara (ili čak podaci iz memorije), svaki poziv potprograma (CALL) ili interapt smeštaju na stek odgovarajuće informacije neophodne za nastavak programa, dok registar ESP ukazuje na poslednji podatak na steku – kada stigne novi, ESP se umanjuje i podatak upisuje na dobijenu adresu. Ukoliko se podatak ne može smestiti na stek (vrednost ESP manja od dužine podatka), generiše se *stack fault* (INT 12).

Kao i ostale instrukcije, PUSH i POP na osnovu izabranog registra „znaju” koliko se bajtova smešta na stek. Preporučili bismo vam, međutim, da uvek na stek šaljete čitav 32-bitni podatak, bez obzira što vam možda treba samo njegovih 16 ili 8 bita. Stvar je u tome što je 80486 optimizovan prema pretpostavci da se svaki podatak nalazi na adresi deljivoj sa četiri. Ukoliko na stek stavite neki podatak dug dva ili (ne daj Bože) jedan bajt, ova sinhronizacija se gubi i sve buduće naredbe koje rade sa stekom izvršavaju se znatno sporije. S obzirom da većina programa intenzivno koristi stek, jedan neoprezni PUSH će žestoko uticati na performanse čitavog programa, uporevajući ga za tridesetak procenata! Dakle, čak i ako ne koristite sve bitove, obavezno ih pre slanja na stek popunite vodećim nulama (ili jedinicama, ako je broj negativan). Za to se često koriste instrukcije tipa MOVSB ili MOVZX.

"Malo je toga na svetu što neko ne može napraviti malo gore i naplatiti malo manje; oni koji razmatraju samo cenu, nužno su žrtve takvih ljudi."

John Ruskin (1810-1900)



MIMICO

Mi smo oni drugi. Bolji i skuplji.

MIMICO Grupa za informatički inženjering
Maršala Birjuzova 3, YU-11000 Beograd
Tel. (011) 182-848, Fax. (011) 624-846

KNJIGA U KOJOJ SVE PIŠE

Bazu podataka čini skup nekih podataka: te podatke koriste korisnici baze, a ponekad i program koji upravlja bazom za smeštanje svojih internih informacija. Tu spadaju svi podaci koje nije potrebno imati stalno na raspolaganju: razne tabele, parametri i slično. U bazu se mogu smestiti i informacije o samoj bazi podataka: polja u datotekama, indeksi, relacije... Stvar odavno postoji i naziva se *rečnik podataka* – to je skup podataka o ostalim podacima u sistemu.

Prelazak sa nekog standardnog programskog jezika na *Clipper* obično je veoma uzbudljiv. *Clipper* ima u sebi mnoge stvari za koje je potrebno satima pisati procedure na standardnim programskim jezicima. Međutim, on nije samo standardni jezik sa „ugrađenom“ dodatnom bibliotekom funkcija: stvar je u koncepciji. Jednostavno, prisustvo moćnijih funkcija dozvoljava programeru razmišljanje u „krupnijim“ koracima. A takvim čizmama od sedam milja mnogo brže se stiže do postavljenog cilja.

Ovakvo „putovanje“ se uvek može učiniti mnogo udobnijim nego što jeste. Prvi korak ka tome je izdvajanje složenijih funkcija koje se često javljaju u programu u posebne procedure. Ponekad je dobro napraviti sistem procedura koji obavlja određenu funkciju – time se neke standardne operacije daleko pojednostavljaju.

PROBLEMI SA INDEKSIMA

Relacionu bazu podataka u principu čini skup tabela. U *Clipper*-u se te tabele memorišu u **DBF** datotekama. Za svaku **DBF** datoteku (tabelu) vezuje se jedna ili više indeksnih datoteka u kojima se čuvaju odgovarajući indeksi. Program mora sam da vodi računa o pridruživanju indeksnih datoteka odgovarajućim **DBF** datotekama – na žalost, *Clipper* tu ne podržava nikakav automatizam. Iz ovoga proizilaze razni problemi, koji drastično povećavaju vreme potrebno za svaku intervenciju na postojećim indeksima.

Kada je u programu potrebno otvoriti jednu tabelu (datoteku), mora se otvoriti **DBF** datoteka i sve pridružene **NTX** indeksne datoteke. To se radi naredbom **USE**. Na primer:

```
USE Student INDEX Stu_Name,Stu_ID NEW
```

Ovim se otvara datoteka studenata i dve indeksne datoteke. U prvoj se čuvaju indeksi po imenima, a u drugoj po šiframa studenata. Elementarna pravila dobrog programiranja nalažu da se podaci o ovoj datoteci – njena struktura, opis svakog polja, indeksi, izrazi po kojima se vrši indeksiranje i slično – zapišu u nekom dokumentu koji prati program. Ovo je neophodno zbog kasnijeg lakšeg snalaženja u programu. U suprotnom, rizikujete da za mesec ili dva potpuno zaboravite značenje pojedinih polja u datoteci, a to vam svakako neće pomoći u razumevanju programa!

Svaka izmena **DBF** datoteke odražava se i na indekse koji su otvoreni. *Clipper* to automatski radi i o tome nije potrebno voditi računa. Zbog raznih neregularnih situacija (nestanak struje, resetovanje računara...) može se desiti da indeksi ne budu dobro ažurirani i da ne odgovaraju podacima u osnovnoj datoteci. Tada je potrebno izvršiti komandu **REINDEX** koja iznova kreira sve otvorene indekse. To znači da bi na početku programa morala da postoji procedura koja ispituje indekse, i, ako se ustanovi da nisu korektni, kreiraju se novi komandom **REINDEX**.

Na žalost, tu nije kraj komplikacijama: svaka indeksna datoteka na početku ima zaglavlje u kome se čuva izraz po kome treba izvršiti indeksiranje. Ko-

Neoad Batočanin

manda **REINDEX** na osnovu ovog izraza kreira nove indekse. Ali, šta ako je oštećeno upravo to zaglavlje? Jedno je sigurno – u tom slučaju naredba **REINDEX** neće pomoći. Tada je neophodno kreirati novu indeksnu datoteku naredbom **INDEX ON**. Međutim, ova naredba zahteva kao argument i izraz po kome se vrši indeksiranje, tako da se mora pisati nešto poput ovog:

```
USE Student NEW
INDEX ON Upper(s_name) TO Stu_Name
INDEX ON s_id TO Stu_ID
```

Datoteku studenata ćete verovatno koristiti u više procedura. Ako se u tim procedurama menja datoteka **STUDENTI**, menjaće se i indeksi, pa je potrebno uvek držati otvorene sve indekse.

I tu bi priči (možda) bio kraj, da nije jednog važnog perioda u životu svakog programa: izmene. U trenutku pisanja programa lako se može desiti da pogrešite u izboru indeksa, ili vam kasnije zatreba još jedan indeks. Tada počinje posao koji možete preporučiti samo najgorim neprijateljima: po celom programu treba pronaći sve naredbe **USE** i **INDEX**

ON koje se odnose na tu datoteku i izvršiti odgovarajuću izmenu. Naravno, tu izmenu treba odmah uvrstiti i u već pomenutu dokumentaciju programa. Na žalost, ovakve izmene mogu veoma lako da poremete i ostatak strukture programa. Na primer, ako je novi indeks ubačen na prvo mesto, onda naredba **SET ORDER TO 1** više neće aktivirati indeks sa imenima, nego novi indeks.

Prilikom instalacije programa, potrebno je inicijalizovati kompletnu bazu podataka. To se uglavnom svodi na brisanje podataka iz svih datoteka ili na ponovno kreiranje svih potrebnih tabela. Za ovakvu operaciju neophodno je imati podatke o strukturi svih tabela koje ulaze u bazu podataka.

Znači, struktura **DBF** datoteka (uključujući i indekse) veoma utiče na program i dokumentaciju. Zato se pri izmeni strukture baze podataka (odnosno pojedinih **DBF** datoteka) dosta vremena troši na sve ove izmene: treba izmeniti dokumentaciju, modul za kreiranje indeksa, modul za periodično brisanje i sve procedure koje koriste dotičnu datoteku.

U stvari, glavni problem je u zavisnosti programa i baze podataka: izmena u programu povlači neku izmenu u strukturi baze podataka i obrnuto. Ovaj začarani krug je davno uočen i razrađene su tehnike koje ga donekle poništavaju. Osnovna ideja se svodi na **nezavisnost** programa i podataka. Nezavisnost se dobija uvođenjem nekoliko nivoa, od kojih svaki komunicira sa narednim samo strogo kontrolisanim kanalima i minimalno je zavisan od njega. Naravno, potpuna nezavisnost je samo teorijski pojam, ali se mnogo može učiniti reorganizacijama programa koji upravlja bazom podataka.

Pošto je program toliko zavisan od baze podataka, treba sve učiniti da se ta zavisnost smanji: tako će i broj intervencija u slučaju izmene biti manji. Međutim, program **mora** biti donekle zavisan od strukture baze podataka. Na primer, da bi se ubrzalo pretraživanje, program mora imati informaciju da li datoteka ima indeks. Zato se ova zavisnost može smanjiti, ali se ne može potpuno izbeći.

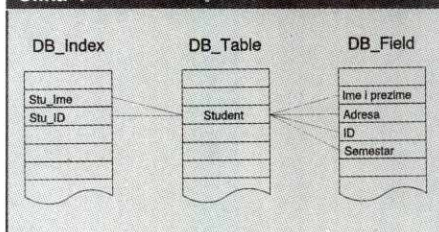
REŠENJE: REČNIK PODATAKA

Pošto se svaka izmena indeksa mora raditi na nekoliko mesta, ima smisla za otvaranje svake datoteke praviti posebnu proceduru i tako promene indeksa vršiti na samo jednom mestu. Međutim, to nas ne spašava od izmene nekih procedura, kao što je ona za kreiranje novih indeksa.

Ovde se može primeniti jedno veoma jednostavno i elegantno rešenje: podaci o strukturi baze podataka takođe se čuvaju u bazi podataka, baš kao i ostali podaci. Taj deo baze podataka obično se naziva **rečnik podataka**. Ova u suštini jednostavna ideja može veoma mnogo pomoći pri unutrašnjoj organizaciji programa.

Kod relacionih baza podataka dosta je jednostavno memorisati kompletnu strukturu baze podataka – u slučaju *Clipper*-a, svih **DBF** datoteka, odgovarajućih polja, indeksa i relacija. Podaci o datotekama, poljima i indeksima mogu se čuvati u tri datoteke: **DB_Table**, **DB_Field** i **DB_Index**. Na slici 1. je prikazan međusobni odnos između ovih datoteka. U svakom slogu datoteke **DB_Table** nalaze se podaci

Slika 1



Struktura rečnika podataka

Slika 2

Slika 2: Struktura tabela za čuvanje rečnika podataka

```

.....
DB_TABLE - Spisak svih tabela
1. DB_TabN : Upper(t_name)
2. DB_TabA : Upper(t_alias)

T_NAME - Naziv table C12
T_REM - Opis table C30
T_ALIAS - Alias kod otvaranja C12
T_DCOND - Uslov po kome se brise C50

.....
DB_FIELD - Definicije polja za table
1. DB_Field : Upper(f_tname+f_name)

F_TNAME - Naziv table C12
F_NAME - Naziv polja C10
F_CHK - Kontrola C10
F_TYPE - Tip polja C1
F_DEC - Decimalni deo N2.0
F_LEN - Duzina polja N2.0
F_REM - Opis polja C30

.....
DB_INDEX - Definicije indeksa
1. DB_Index : Upper(i_tname) + Str(i_ord,2)

I_NAME - Naziv table C12
I_NAME - Naziv indeksa C10
I_REM - Opis indeksa C30
I_EXP - Izraz za indeksiranje C70
I_ORD - Broj indeksa (ORDER) N2.0
.....

```

Struktura tabela za čuvanje rečnika podataka

Slika 3. Open

```

PROC Open (cFile, cAlias)
  LOCAL cName, aIndex, i, nLen

  // Spisak tabela
  USE DB_Table INDEX DB_TabN, DB_TabA NEW
  nLen := Len(t_name)

  // Trazi tabelu
  SEEK PadR(Upper(cFile), nLen)

  IF !Found()
    Alert ("DATOTEKA " + Trim(cFile) + " NE POSTOJI; U SPISKU!")
    QUIT
  ELSE
    cName := Upper(t_name)
    REPLACE t_alias WITH IIF (cAlias == NIL, cFile, cAlias)
  END IF

  // Otvori spisak indeksa u istoj zoni
  USE DB_Index INDEX DB_Index
  DBSeek (cName + " 0", .T.)

  // Nadji sve indekse
  aIndex := {}
  WHILE Upper(i_name) == cName
    AAdd (aIndex, Trim(i_name))
    SKIP
  END DO

  // Otvori tabelu i indekse
  DBUseArea (.F., , cName, cAlias)
  AEval (aIndex, { |i| DBSetIndex (i) })

RETURN

```

Funkcija za otvaranje datoteke i svih indeksa

Slika 4 Order

```

PROC Order (cIndex)
  LOCAL cName, aIndex, i, sSelect, cAlias

  // Zapamti oblast
  sSelect := Select()
  cAlias := Alias()

  // Spisak tabela
  USE DB_Table INDEX DB_TabA NEW
  SEEK PadR(Upper(cAlias), Len(t_alias))

  IF !Found()
    Alert ("NE POSTOJI EVIDENCIJA O TEKUCOJ RADNOJ OBLASTI!")
    QUIT
  ELSE
    cName := Upper(t_name)
  END IF

  // Otvori spisak indeksa u istoj zoni
  USE DB_Index INDEX DB_Index
  DBSeek (cName + " 0", .T.)

  WHILE Upper(i_name) == cName
    IF Upper(cIndex) == Upper(Trim(i_name))
      (sSelect->DBSetOrder(DB_Index->i_ord))
    EXIT
  END IF
  SKIP
  END DO

  // Zatvori tabelu
  CLOSE
  SELECT (sSelect)

RETURN

```

Procedura za promenu aktivnog indeksa

o jednoj tabeli u bazi. Podaci u ovoj tabeli mogu se izabrati po potrebi, ali sigurno treba uključiti naziv tabele (datoteke), opis sadržaja, parametre potrebne pri otvaranju i slično. Za svaki slog datoteke **DB_Table** vezano je više slogova iz **DB_Field** koji sadrže odgovarajuća polja. Na sličan način se za jednu tabelu vezuju i indeksne datoteke, čiji nazivi se čuvaju u datoteci **DB_Index**.

Rečnik podataka ne mora da sadrži samo osnovne podatke: tu je zgodno ubaciti i mnoge druge informacije koje kasnije mogu dobro doći prilikom održavanja sistema. Na slici 2. je data moguća struktura ove tri datoteke. Tu su, pored osnovnih informacija, i opisi elemenata, izrazi za indeksiranje i sl. Mnogi od ovih podataka se neće koristiti direktno, ali će biti dragoceni pri izradi dokumentacije. Izradom ovakvog rečnika centralizovali smo sve informacije o strukturi baze podataka, sa sasvim malim izuzecima. Znači, ideja je da ubuduće svaku izmenu baze radimo samo na jednom mestu: u rečniku podataka.

RAD SA REČNIKOM

Najkritičnije mesto kod rada sa indeksima je naredba **USE**, odnosno otvaranje **DBF** datoteke i svih datoteka sa indeksima. Na slici 3. je prikazana funkcija **Open**, koja otvara zadatu datoteku i sve odgovarajuće indekse, kao i naredba **USE**. Prvo se na osnovu zadatog naziva datoteke traži naziv u tabeli. Ako ne postoji tabela u evidenciji, izdaje se odgovarajuće obaveštenje i prekida se rad programa (mada ima smisla i pokušati sa nastavkom, uz određeno upozorenje). Ako se datoteka nalazi u evidenciji, otvara se datoteka u novoj radnoj zoni i upisuje se odgovarajući alias u rečnik. Korišćenjem ovog podatka, može se kasnije lako otkriti pokušaj otvaranja već otvorene datoteke, što može biti veoma korisno kod velikih aplikacija sa dosta nezavisnih modula. Zatim se iz datoteke svih indeksa izdvajaju slogovi koji se odnose na zadatu datoteku i upisuju u niz. Na osnovu ovog niza se otvaraju svi indeksi koji pripadaju datoteci, korišćenjem funkcije **DBSetIndex**. Naredbu **USE** u programu sada treba zameniti funkcijom **Open**. Na primer:

```
Open ("Studenti")
```

Sada je problem dodavanja ili oduzimanja indeksnih datoteka rešen: izmene treba raditi samo u rečniku podataka, tj. u datoteci **DB_Index**, a funkcija **Open** će uvek otvarati sve indekse koje nade na spisku. Sa ovako otvorenim datotekom radi se sasvim uobičajeno, potpuno isto kao da je za otvaranje upotrebljena naredba **USE.INDEX**.

Ponekad je potrebno datoteku otvoriti bez indeksa ili samo sa nekim određenim indeksom – najčešće zbog uštede na broju otvorenih datoteka. Tada se umesto funkcije **Open** može koristiti obična naredba **USE** – pošto se indeksi ne otvaraju, kasnije izmene u broju ili redosledu indeksa neće se odražavati na ovu datoteku.

Ovakva centralizacija otvara zaista lepe mogućnosti za neke inače dosta komplikovane operacije. Na primer, vrlo je jednostavno napraviti kontrolu pristupa podacima u datoteci: za svakog korisnika se može voditi evidencija o tome kojim tabelama može pristupiti (na primer, opet u nekom delu rečnika podataka) i na početku funkcije **Open** se ispita dozvola za rad sa tabelom. Naravno, pri tom kasnije treba ispitati mogućnost nastavka tekuće procedure, jer se može desiti da rad bez te datoteke bude nemoguć. Takođe se olakšavaju neke dodatne operacije koje je potrebno uraditi pre otvaranja datoteke (provera pristupa kod rada u mreži i sl.).

Ovakvo rešenje krije u sebi i neke nedostatke. Naime, redosled otvorenih indeksa je u programu nevidljiv. Ako datoteku otvorite sa

```
USE Test INDEX Test1, Test2 NEW
```

naredba **SET ORDER TO 2** imaće sasvim vidljivo dejstvo (ili se barem tako čini!), dok se kod otvaranja sa **Open** („Test“) to ne vidi u tekstu programa. Naravno, u definiciji (u datoteci **DB_Index**) uvek se može videti koji je indeks definisan kao drugi, ali programer to ne vidi u tekstu programa. Tako se može javiti sledeća greška: korisnik u rečniku doda još jedan indeks i postavi ga na mesto broj 2, tako da stari indeks broj 2 postaje treći. Time naredba **SET ORDER TO 2** u programu postaje pogrešna: sada treba da glasi **SET ORDER TO 3**.

Problem se elegantno može rešiti uvođenjem funkcije sa slike 4. Ona menja aktivni indeks, ali ne na osnovu rednog broja već na osnovu njegovog na-

ziva. Na primer, umesto **SET ORDER TO 2**, koristi se funkcija **Order** („Test2“), što će uvek postavljati indeks **Test2** za aktivan, bez obzira na izmene u broju i redosledu indeksa.

Vrlo zgodna osobina opisanog sistema je to što ne zahteva velike izmene u već napisanim programima: posle kreiranja rečnika podataka, svi stari programi će savršeno korektno raditi, bez obzira što ste tamo koristili naredbu **USE**, a ne funkciju **Open**. Novi sistem tako možete sasvim postepeno uključivati u svoje programe.

ODRŽAVANJE BAZE

Rečnik podataka je dragocen za kontrolu podataka u bazi. Najjednostavniji primer je kreiranje novih indeksa. U prethodnom delu je bilo govora o potrebi kreiranja novih indeksa, najčešće prilikom raznih neregularnih situacija. Ako podatke o indeksima svih datoteka imate na raspolaganju, problem je gotovo trivijalan: jednostavno treba redom otvarati datoteke sa spiska i kreirati odgovarajuće indekse. Procedura **CreateIndex** koja to radi prikazana je na slici 5. Ovim se zaobilazi naredba **REINDEX** i problem sa oštećenjem indeksnih datoteka: sada se sve kreira iz početka.

Podaci se u bazu podataka neprestano upisuju i, mada se izvesni podaci brišu, rezultat je uvek pozitivan: baza podataka neprestano raste. Zbog ograničenog prostora na disku i brže obrade podataka, potrebno je povremeno brisati stare podatke iz baze. Kod raznih poslovnih primena, podaci se obično brišu posle periodičnih obračuna ili na kraju godine. Ovdje se javlja problem sličan kreiranju indeksa: pošto se struktura baze podataka menja, uvek treba predviđati i brisanje novih tabela. Stalna izmena procedure za povremeno brisanje baze je zaista mučan posao. Međutim, to se sada lako može izbeći korišćenjem opisa u rečniku: za svaku tabelu koju je potrebno periodično brisati navodi se uslov koji treba da bude zadovoljen da bi se slog datoteke proglasio starijim i izbrisao. Na primer, uslov može biti **"fdat <= Date()"** – ovim se iz datoteke brišu svi slogovi stariji od današnjeg datuma. Sada je dovoljno redom otvarati datoteke za koje je zadat uslov (polje **t_dcond** u **DB_Table**) i izvršiti naredbu **DELETE ALL FOR &uslov** – tako jedan zamoran posao po-

staje potpuno automatizovan i krajnje jednostavan. Naravno, sada nije potrebna nikakva intervencija ni kod izmene indeksa: pošto se svi podaci čitaju iz rečnika podataka, dovoljno je izmenu izvršiti samo tamo.

Još čitav niz poslova postaje veoma pojednostavljen: na primer, kreiranje potpuno nove baze podataka prilikom instalacije je trivijalan posao, kada postoje informacije u rečniku podataka. Naravno, kada pravite instalacionu verziju programa, osim **EXE** programa morate isporučiti i tri datoteke koje čine rečnik podataka.

Ako je povećana verovatnoća oštećenja baze podataka, može se lako napraviti program koji proverava strukturu svake datoteke posebno. Ovde je problem što se i rečnik nalazi u **DBF** datotekama, pa šta ako se on ošteti? Onda se (bar za korisnika) gubi mogućnost pristupa svim podacima u bazi. Rešenje ovog problema može biti pomalo neobično: na osnovu podataka iz rečnika, generiše se standardni *Clipper* program koji proverava strukturu, generiše nove indekse, itd. Automatsko generisanje (znači, ne piše ga programer već neki pomoćni program) ovakvog programa je jednostavno. Prednost nad ručnim pisanjem ovakvog modula je velika, jer se uloga programera sada svodi samo na održavanje rečnika podataka, dok se svi ostali poslovi obavljaju automatski. Znači, ako iz bilo kog razloga smatrate da je **EXE** datoteka sigurnija od **DBF** datoteka rečnika, predvidite mogućnost da u finalnoj verziji programa generišete module za proveru strukture, kreiranje indeksa i sl. U veoma osetljivim slučajevima mogu se predvideti obe varijante: baza se može kreirati na osnovu **DBF** datoteka i na osnovu modula za kreiranje, pri čemu se modul upotrebljava kada je rečnik oštećen. Provera da li su podaci u rečniku „dobri“ se može obaviti uvođenjem kontrolne sume za sve njegove podatke.

IZMENA REČNIKA

Pošto su sada svi problemi svedeni na manipulaciju rečnikom, potrebno je razraditi tehniku za upis i ažuriranje rečnika. Rečnik se može menjati iz bilo kog programa koji može da menja strukturu i sadržaj **DBF** datoteka, ali je mnogo bolje napraviti pomoćni program koji to radi. Upis i brisanje datoteka iz rečnika ne mora se uvek raditi „van“ programa: kod nekih praktičnih problema često može zatrebati da se kreira nova ili izbrise postojeća datoteka za vreme izvršavanja programa. Naravno, sam posao kreiranja veoma je jednostavan, ali problem je što onda rečnik podataka neće imati potrebne informacije o novim datotekama. Zato treba prilikom kreiranja (ko-rišćenjem funkcije **DBCcreate** ili na neki drugi način) podatke o novim datotekama upisivati i u rečnik podataka. Na slici 6. prikazane su funkcije za upis i brisanje datoteke iz rečnika podataka.

Funkcija **CreateFile**, na osnovu naziva datoteke i strukture, prvo kreira datoteku, a zatim podatke o njoj upisuje u rečnik podataka. Treći parametar ove funkcije je niz parova (**Izraz, Datoteka**) koji definiše sve indekse za datu datoteku. Na osnovu ovog niza, funkcija upisuje podatke u rečnik i kreira nove indekse. Funkcija za brisanje **DeleteFile** nešto je jednostavnija: na osnovu naziva datoteke brišu se sve datoteke i opis u rečniku.

Time je zavisnost programa od datoteka smanjena, bar što se tiče indeksa: sada je dovoljno menjati indekse samo u rečniku, jer se sve informacije odatle automatski koriste u programu. Naravno, zavisnost još uvek postoji: ako izbrisete neki indeks, naredba **SEEK** u programu nema smisla. Ipak, rezultat je i te kako pozitivan.

Ostala je još jedna veoma važna karika: program najviše zavisi od polja u bazi. Program mora „znati“ koja su polja, kog tipa i veličine u datotekama. U nekim slučajevima se stvar može automatizovati. Ako se sa datotekom rade samo standardne

operacije, nije problem napraviti univerzalne procedure za upis, brisanje, unos i pretraživanje podataka. Međutim, obično se realni problemi ne uklapaju u ovaj šablon, pa takve univerzalne procedure nisu baš sasvim komfornе za krajnjeg korisnika. U suštini, *Clipper* ne razlikuje naročito memorijsku promenljivu i podatak iz sloga **DBF** datoteke – i jedno i drugo je nešto što on ravnopravno može koristiti u daljoj obradi. Jedina razlika je u načinu dobijanja tog podatka. Bilo bi moguće izdvojiti sloj programa koji automatski brine o otvaranju datoteka, uspostavljanju relacija itd., dok bi ostatak programa jednostavno uzimao dobijene podatke. Rečnik podataka može predstavljati sasvim solidnu osnovu za takav sistem. Tu tek treba očekivati primenu objektno orijentisanog programiranja – izgleda da će novi *Clipper* stići u pravo vreme!

Slika 6

```
PROC CreateFile (cFile, aStruc, aIndex)
LOCAL sSav, i

sSav := Select()

// Kreiranje datoteka
DBCcreate (cFile, aStruc)

// Indekspiranje
IF aIndex <> NIL
USE (cFile) NEW
FOR i := 1 TO Len(aIndex)
INDEX ON &(aIndex[i,1]) TO {aIndex[i,2]}
NEXT
CLOSE

// Upis indeksa
Open ("DB_Index")
FOR i := 1 TO Len(aIndex)
APPEND BLANK
REPLACE i_exp WITH aIndex[i,1], ;
         i_name WITH aIndex[i,2], ;
         i_ord WITH i, ;
         i_tname WITH cFile
NEXT
CLOSE

// Upisi u rečnik
Open ("DB_Table")
APPEND BLANK
REPLACE t_name WITH cFile
CLOSE

// Upis polja
Open ("DB_Field")
FOR i := 1 TO Len(aStruc)
APPEND BLANK
REPLACE f_tname WITH cFile, ;
         f_name WITH aStruc[i,1], ;
         f_chk WITH aStruc[i,1], ;
         f_type WITH aStruc[i,2], ;
         f_len WITH aStruc[i,3], ;
         f_dec WITH aStruc[i,4]
NEXT
CLOSE

Select (sSav)
RETURN

PROC DeleteFile (cFile)
LOCAL sSel, cTpp, cOldName

sSel := Select ()

// Fizičko brisanje
ERASE (cFile + ".DBF")
ERASE (cFile + ".NTX")
ERASE (cFile + ".DBT")

// Brisanje iz rečnika
Open ("DB_Table")
PACK
SEEK PadR(Upper(cFile), Len(t_name))

IF !Found()
Alert ("DATOTEKA NE POSTOJI U SPISKU!")
QUIT
END IF

cOldName := t_name
Select DB_Table
DELETE
PACK
CLOSE

Open ("DB_Index")
DELETE ALL FOR Upper(cOldName) == Upper(i_tname)
PACK
CLOSE

Open ("DB_Field")
DELETE ALL FOR Upper(cOldName) == Upper(f_tname)
PACK
CLOSE

Select (sSel)

RETURN
```

Kreiranje nove datoteke i njen upis u rečnik podataka

Zavisnost programa od polja može se donekle ublažiti: neke podatke koji su vezani isključivo za **DBF** datoteke treba izdvojiti u rečnik. Tako bi se za svako polje u rečniku mogli upisivati podaci o unosu i izmeni (format, kontrole, podrazumevane vrednosti, prava pristupa pojedinih korisnika itd.).

Bilo bi veoma dobro spojiti i poslednju kariku u lancu: izmena podataka o strukturi u rečniku i struktura **DBF** datoteka. Ovakve izmene obično se rade uz pomoć programa *DBU*, koji omogućava kreiranje i izmenu strukture. Međutim, *DBU* veoma loše rešava problem: pri nekim složenijim izmenama, program ne radi dobro, i, što je najgore, ponekad uništava podatke u **DBF** datoteci. Tipičan primer je promena naziva ili tipa polja. Zato je dobro koristiti proceduru sa slike 7, koja menja strukturu datoteke u skladu sa rečnikom podataka. Znači, kada je potrebno da napravite izmenu, napravićete je u rečniku, a procedura **ChangeStruc** tu izmenu upisuje u datoteku. Možda deluje nelogično, ali upotreba naredbi **COPY TO** i **APPEND FROM** za izmenu strukture definitivno otpada: pri prvoj izmeni tipa polja, ove naredbe ne rade kako se očekuje. Zato se izmena strukture mora obaviti posebnom procedurom. Procedura **ChangeStruc** prvo kreira nizove **aStr1** i **aStr2** koji sadrže strukturu datoteke u rečniku i u datoteci. Sada je potrebno izmeniti strukturu datoteke, tako da opis u rečniku bude važeći, ali pri tom treba sačuvati što je moguće više podataka. Data procedura upoređuje svaka dva elementa nizova struktura. Ako se ustanovi da su elementi isti, izvršiće se jednostavno kopiranje polja. Ako je u rečnik upisano polje koje ne postoji u datoteci, polje treba dodati u datoteku.

Ako je promenjen samo naziv starog polja, kopira se sadržaj polja sa starim nazivom. Ali, kako razlikovati polje promenjenog naziva i novo polje? Jer, može se izmeniti naziv starog polja **IME** u **NAZIV**, ali potpuno ista situacija je kada se izbrise polje **IME** a zatim doda polje **NAZIV**! Zato je neophodno na neki način voditi evidenciju o „poreklu“ polja, kako bi se razlikovala ova dva slučaja. *DBU* problem rešava tako što strukturu datoteke privremeno učitava u poseban niz, pa se kasnije taj niz poredi sa izmenjenim. U rečniku je primenjeno nešto drugačije rešenje: svaka promena naziva polja prenosi se i na polje **F_CHK**, u kome se takođe čuva naziv polja. Pri upisu nove datoteke, ovo polje se ažurira automatski. Kada treba izmeniti strukturu, menjaju se samo polja **F_NAME** i **F_TYPE**, dok **F_CHK** ostaje neizmenjeno. Na osnovu njega se određuje stari naziv polja (ako je izmenjen).

Izmjena dimenzija polja takođe se rešava jednostavnim kopiranjem: na primer, polje **NAZIV** se iz **CHARACTER 30** menja u **CHARACTER 25**. Najveći problem je izmena tipa polja. Mada deluje čudno, do ove akcije veoma često dolazi: nekad zbog greške, a nekad i namerno. Funkcija **ChangeStruc** pokušava da konvertuje jedan tip podatka u neki drugi koristeći standardne funkcije. Naravno, „inteligenciju“ ove konverzije treba vrlo oprezno prihvatiti.

Rezultat analize je niz trojki (**P_OLD, P_NEW, ACTION**). **P_OLD** i **P_NEW** su nazivi novih i starih polja, a **ACTION** je izraz kojim treba zameniti staro polje u novom polju. Pre upisa nove strukture svakako treba dozvoliti izmenu ovih akcija: moguće je da korisnik želi da uradi neku sasvim prostu operaciju koja nije predviđena. Zatim se na osnovu ovih akcija kreira nova datoteka i u nju kopira sadržaj stare.

Rečnik podataka ne mora sadržati samo podatke o poljima i indeksima – tu se sasvim lepo mogu uklopiti i podaci o relacijama, pa i podaci o nekim memorijskim promenljivim koje se dobijaju iz podataka u bazi. Moglo bi se reći da je sve ovo samo početak jednog uopštavanja problema – sledeći korak bi mogao biti pravljenje rečnika funkcija – u njemu bi se nalazile funkcije umesto promenljivih, njihovih opisi, prava pristupa podacima i slično.

Ali, to je već sledeća priča!

Slika 5 CreateIndex

```

PROC CreateIndex
LOCAL sSelect

sSelect := Select()

// Spisak tabela
USE DB_Table NEW

INDEX ON Upper(t_name) TO DB_TabN
INDEX ON Upper(t_alias) TO DB_TabA

// Indeks1
USE DB_Index NEW
INDEX ON Upper(i_name) + Str(i_ord,2) TO DB_Index

SELECT DB_Table

WHILE ! Eof()
  cName := Upper(t_name)

  IF ! (Trim(cName) == 'DB_TABLE' .OR. Trim(cName) == 'DB_INDEX')

    // Nadji spisak indeksa
    SELECT DB_Index
    DBSeek (cName + " 0", .T.)

    // Otvori tabelu
    DBUseArea (.T., , cName, 'DBF')

    // Kreiranje indeksa
    WHILE Upper(DB_Index->i_name) == cName
      INDEX ON &(DB_Index->i_exp) TO (DB_Index->i_name)
      SKIP ALIAS DB_Index
    END DO

    CLOSE
  END IF

  SELECT DB_Table
  SKIP
END DO

CLOSE DB_Index
CLOSE DB_Table

Select (sSelect)
RETURN

```

Procedura za kreiranje novih indeksa

Slika 7 ChangeStruc

```

PROC ChangeStruc (cFile)
LOCAL aStr1 := {}, aStr2 := {}, i, nPos, a

// Uzmi strukturu iz datoteke (stara struktura)
IF File (cFile + ".DBF")
  USE (cFile)
  aStr1 := DBStruct()
  CLOSE
ELSE
  aStr1 := {}
END IF

// Uzmi strukturu iz rečnika (nova struktura)
USE DB_Field
GO TOP
WHILE ! Eof()
  IF Trim(f_name) == Trim(Upper(cFile))
    AAdd (aStr2, {f_name, Upper(f_type), f_len, f_dec, f_chk})
  END IF
  SKIP
END DO

// Privremena tabela akcija
DBCcreate ("__STMP", { ;
  {"DICT", "C", 17, 0}, ;
  {"FILE", "C", 17, 0}, ;
  {"ACTION", "C", 40, 0}, ;
})

USE __STMP NEW

// Poređenje 2 sa strukturom 1, kreiranje parova akcija
FOR i := 1 TO Len(aStr2)

  // Nadji odgovarajuću poziciju u strukturi 1
  nPos := Ascian (aStr1, { |x| Upper(Trim(x[1])) == Upper(Trim(aStr2[1,5])) })

  // Polje nije novo
  IF nPos <> 0

    // Ako je isti atribut, samo se kopira
    IF Upper(aStr1[nPos,2]) == Upper(aStr2[1,2])

      // Stavi 1 P1 i P2
      APPEND BLANK
      REPLACE File WITH SS(aStr1[nPos]), ;
        Dict WITH SS(aStr2[1]), ;
        Action WITH aStr1[nPos,1]

    // Atribut je promenjen!
    ELSE

      // Svakako stavi 1 P1 i P2
      APPEND BLANK
      REPLACE File WITH SS(aStr1[nPos]), ;
        Dict WITH SS(aStr2[1])

    DO CASE

      // N -> C

```

Procedura za inteligentnu izmenu strukture DBF datoteke

```

CASE aStr1[nPos,2] == "N" .AND. aStr2[1,2] == "C"
  REPLACE Action WITH "Str(" + aStr1[nPos,1] + ")"

// C -> N
CASE aStr1[nPos,2] == "C" .AND. aStr2[1,2] == "N"
  REPLACE Action WITH "Val(" + aStr1[nPos,1] + ")"

// C -> D
CASE aStr1[nPos,2] == "C" .AND. aStr2[1,2] == "D"
  REPLACE Action WITH "CtoD(" + aStr1[nPos,1] + ")"

// C -> M
CASE aStr1[nPos,2] == "C" .AND. aStr2[1,2] == "M"
  REPLACE Action WITH aStr1[nPos,1]

// D -> C
CASE aStr1[nPos,2] == "D" .AND. aStr2[1,2] == "C"
  REPLACE Action WITH "DtoC(" + aStr1[nPos,1] + ")"

// M -> C
CASE aStr1[nPos,2] == "M" .AND. aStr2[1,2] == "C"
  REPLACE Action WITH aStr1[nPos,1]

// Ostalo
OTHERWISE
  REPLACE Action WITH "?"

END CASE

END IF

// Izbrisi polazno polje
aStr1[nPos,1] := ""

// Polje je novo
ELSE

  // Stavi P2
  APPEND BLANK
  REPLACE Dict WITH SS(aStr2[1])

END IF

NEXT

// Ubaci sad stara polja
FOR i := 1 TO Len(aStr1)
  IF aStr1[i,1] <> ""
    APPEND BLANK
    REPLACE File WITH SS(aStr1[i])
  END IF
NEXT

// Pregled struktura i odgovarajuće akcije
GO TOP
CLS
Browse ()

// Uzmi akcije u niz
GO TOP
a := {}
WHILE ! Eof()
  IF ! Empty(Action)
    AAdd (a, {Left(Dict,10), Action})
  END IF
  SKIP
END DO
CLOSE
FErase ("__STMP.DBF")

// Kreiranje dat sa novom strukturom
DBCcreate ("__TMP", aStr2)

// Kopiranje, ako se datoteka ne kreira prvi put
IF File(cFile + ".DBF")
  Open (cFile)
  USE __Tmp ALIAS NewTable NEW

  WHILE ! ((cFile)->(Eof()))
    APPEND BLANK
    FOR i := 1 TO Len (a)
      REPLACE NewTable->&(a[i,1]) WITH (cFile)->&(a[i,2])
    NEXT
    SKIP ALIAS (cFile)
  END DO
  COMMIT

  CLOSE (cFile)
  CLOSE NewTable
END IF

COPY FILE __TMP.DBF TO (cFile + ".DBF")
IF File("__TMP.DBT")
  COPY FILE __TMP.DBT TO (cFile + ".DBT")
END IF
FErase ("__TMP.DBF")
FErase ("__TMP.DBT")

// Iskopiraj nova kontrolna polja
SELECT DB_Field
REPLACE f_chk WITH f_name ALL

CLOSE ALL

RETURN

STATIC FUNC SS (aStr)
LOCAL cRet

cRet := Upper(PadR(aStr[1],10)) + " " + Upper(aStr[2])

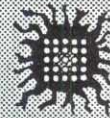
IF Upper(aStr[2]) == "C"
  cRet += AllTrim(Str(aStr[3],2,0))
ELSEIF Upper(aStr[2]) == "N"
  cRet += AllTrim(Str(aStr[3],2,0)) + " " + AllTrim(Str(aStr[4],2,0))
END IF

RETURN cRet

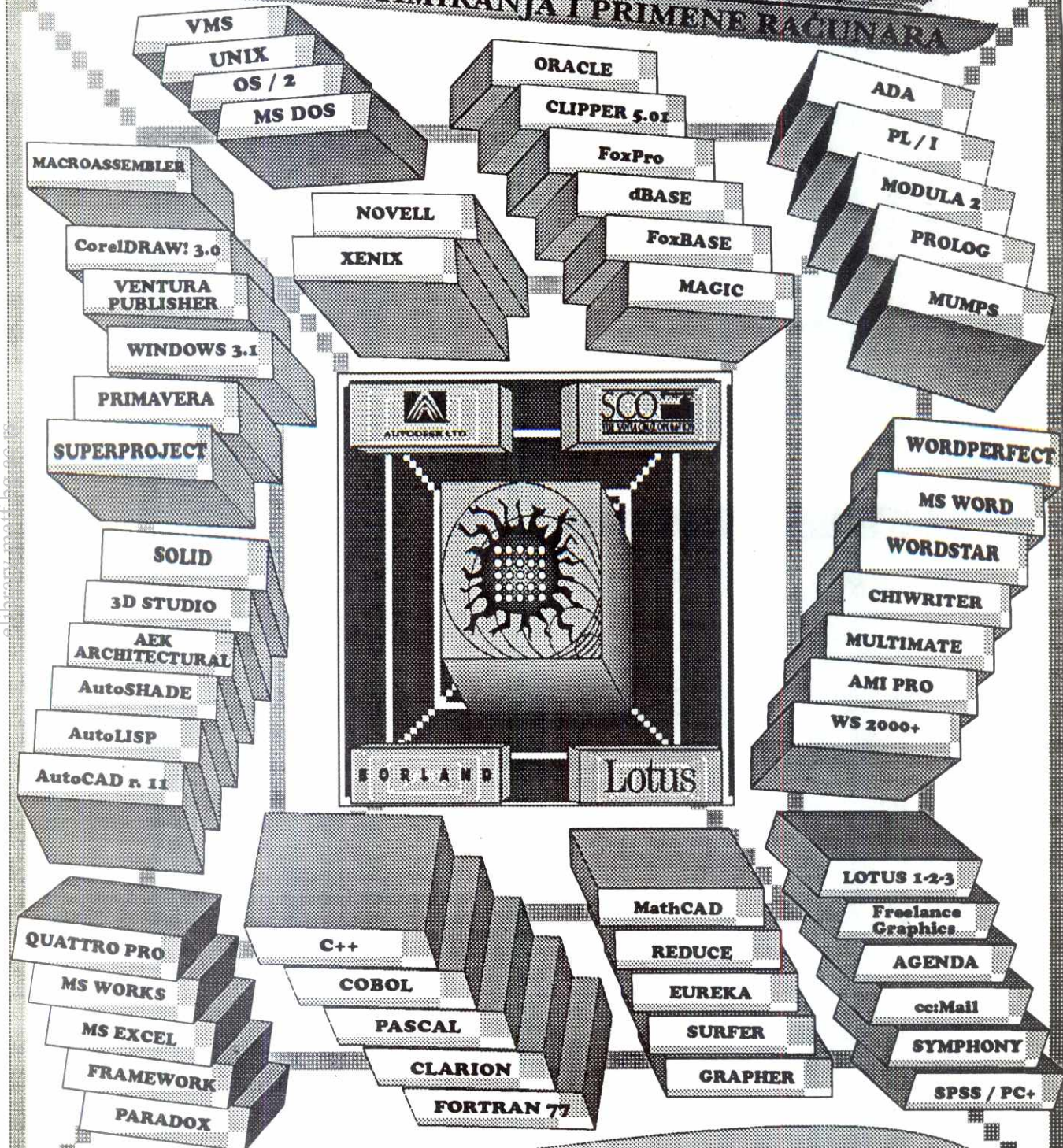
```



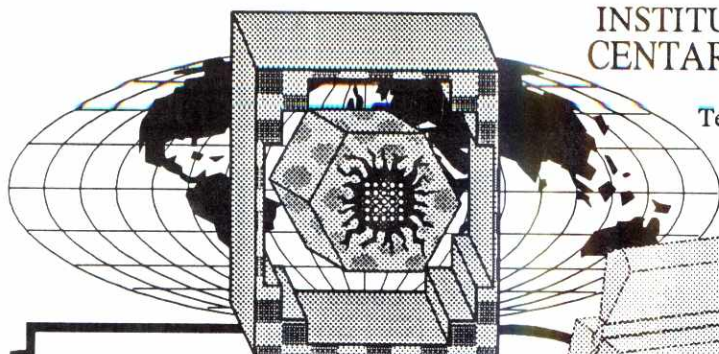
INSTITUT ZA NUKLEARNE NAUKE "VINČA"
CENTAR ZA PERMANENTNO OBRAZOVANJE
11000 Beograd, Nemanjina 4/X
Telefoni: 011 / 683-390, 682-486, 641-155 / 107, 181
Telefax: 011 / 682-486



KURSEVI PROGRAMSKIH JEZIKA, PROGRAMIRANJA I PRIMENE RAČUNARA



- » Jedini autorizovani obrazovni centar u Srbiji softverskih firmi: "AUTODESK LTD", "THE SANTA CRUZ OPERATION", "BORLAND" i "LOTUS".
- » Nastava se održava u Beogradu, Nemanjina 4/X. Centar poseduje preko 70 računara.
 - » Broj polaznika je ograničen (dvadeset u grupi, do dva slušaoca za jednim računarom).
 - » Posebne pogodnosti za preduzeća (kursevi kod korisnika, kursevi van programa, popust za više kandidata)
 - » Slušaoci dobijaju literaturu i, po uspešno položenom završnom ispitu, uverenje o završenom kursu.
 - » Mogućnost plaćanja kotizacije za kurs u ratama.



BIBLIOTEKA INFORMATIKA

- 1. AutoCAD** (verzija 11)
konstruisanje i projektovanje pomoću personalnih računara
Autori: Boris Damjanović i Petar Damjanović
Sedmo izdanje, izlazi iz štampe u aprilu '93.
- 2. Uvod u C jezik**
Autor: Vladan Vujičić
Četvrto izdanje, 1991. - latinica, 317 strana, format B-5
- 3. Primena programa SYMPHONY** na personalnim računarima
Autor: Dragan Pantić
Treće izdanje, 1990. - latinica, 226 strana, format B-5
- 4. OS/2** - vodič za korisnike
Autor: Zorica Jelić
Drugo izdanje, izlazi iz štampe u aprilu '93.
- 5. VENTURA** - računarsko izdavaštvo
Autor: Predrag Davidović
Treće izdanje, izlazi iz štampe u aprilu '93.
- 6. FORTRAN 77**
standard sa dopunama za personalne računare
Autori: Vlakjo Kocić i Zoran Konstantinović
Drugo izdanje, 1990. - latinica, 422 strane, format B-5
- 7. UNIX** - vodič za korisnike
Autor: Zorica Jelić
Treće izdanje, 1991. - latinica, 422 strane, format B-5
- 8. Primena programa FRAMEWORK III** na personalnim računarima
Autor: Dragan Pantić
Drugo izdanje, 1991. - latinica, 326 strana, format B-5
- 9. Programski alati u matematici**
MathCAD, GRAPHER, EUREKA
Autor: Ante Čurlin
Prvo izdanje, 1990. - latinica, 402 strane, format B-5
- 10. QUATTRO PRO 4.0**
Autor: Dragan Pantić
Drugo izdanje, izlazi iz štampe u martu '93.
- 11. DOS UKRATKO** (verzije 3.3 i 5.0)
Autor: Dragan Pantić
Drugo izdanje, 1992. - latinica, 120 strana, format B-5
- 12. Vodič za VAX / VMS**
Autori: Tamaš Kerepeš, Zvonko Oršolić i Saša Matijević
Prvo izdanje, 1990. - latinica, 512 strana, format B-5
- 13. Primena programa EXCEL** na personalnim računarima
Autor: Dragan Pantić
Prvo izdanje, 1991. - latinica, 272 strane, format B-5
- 14. UNIX** - vodič za programere
Autor: Zorica Jelić
Prvo izdanje, 1991. - latinica, 326 strana, format B-5
- 15. WINDOWS 3.0**
Autor: Dragan Pantić
Prvo izdanje, 1991. - latinica, 273 strane, format B-5
- 16. PRIMAVERA**
upravljanje projektima uz pomoć računara
Autori: Jaroslav Urošević i Jelica Draškić - Ostojić
Prvo izdanje, 1991. - latinica, 365 strana, format B-5
- 17. dBASE III+** priručnik
Autor: Milorad Filipović
Prvo izdanje, 1991. - latinica, 249 strana, format B-5
- 18. Osnovi informaciologije i informacione tehnologije**
Autor: Ljubomir Dulović
Prvo izdanje, 1991. - latinica, 538 strana, format B-5
- 19. LOTUS 1-2-3** (verzije 3.0 i 3.1)
Autor: Dragan Pantić
Prvo izdanje, 1991. - latinica, 411 strana, format B-5
- 20. dBASE IV** priručnik
Autor: Ljubomir Lazić
Prvo izdanje, 1991. - latinica, 306 strana, format B-5
- 21. WORDPERFECT** (verzija 5.1)
Autori: Dragan Pantić i Nada Pantić
Prvo izdanje, 1991. - latinica, 300 strana, format B-5
- 22. Programiranje u CLIPPER-u 5.01**
Autor: Alempije Veljović
Prvo izdanje, 1992. - latinica, 465 strana, format B-5
- 23. FoxPro**
Autor: Dušan Čašić
Prvo izdanje, 1992. - latinica, 390 strana, format B-5
- 24. Uvod u strukture podataka**
Autor: Miroslav Jocković
Prvo izdanje, 1992. - latinica, 357 strana, format B-5
- 25. ORACLE** (verzija 5) - arhitektura i administracija
Autor: Vladimir Milojković
Prvo izdanje, 1992. - latinica, 151 strana, format B-5
- 26. WORD za WINDOWS**
Autori: Dragan Pantić i Nada Pantić
Prvo izdanje, 1992. - latinica, 317 strana, format B-5
- 27. MS WORKS** - upravljanje poslovnim procesima
Autori: Mirjana Nikolić i Miroslav Nikolić
Prvo izdanje, 1992. - 273 strane, format B-5
- 28. PASCAL**
standard sa dopunama za personalne računare
Autori: Zoran Konstantinović i Slobodan Simić
Prvo izdanje, 1992. - latinica, 388 strana, format B-5
- 29. LOTUS 1-2-3** (verzija 2.3) i modeli za poslovno odlučivanje
Autori: Mirjana Nikolić i Miroslav Nikolić
Prvo izdanje, izlazi iz štampe u martu '93.
- 30. DOS 6.0**
Autori: Branislav Todorović i Mario Ratančić
Prvo izdanje, izlazi iz štampe u martu '93.

PORUŽBENICA: na liniji uz redni broj knjige upisati broj poručenih primeraka

IME I PREZIME
(Naziv preduzeća)

ULICA I BROJ

BROJ POŠTE I MESTO

TELEFON

20 21 22 23 24 25 26 27 28 29 30
"računari", februar 1993.

Poružbenicu poslati sa petim primerkom uplatnice (informacije o cenama svakog radnog dana od 8 do 20 sati).

Uplata se vrši na žiro račun:

INSTITUT "VINČA", LABORATORIJA 060, broj: 60803-609-698.

Troškove isporuke knjiga snosi poručilac i namiruje ih unapred zajedno sa uplatom za knjige.

KONTROLA TASTATURE

Pri pisanju aplikacije koje zahtevaju komunikaciju sa korisnikom često se javlja problem oko praćenja ili čak zabrane prekida rada programa preko tastature (Ctrl-Break) u toku korišćenja standardnih rutina za čitanje/pisanje (I/O). Rešavajući ovaj problem, shvatio sam da primenjena tehnika krije čitav niz lepih mogućnosti.

Pre četiri godine, u *Računarima* 44/76, o makroima je pisao Zoran Živočić. U tekstu „Mikro program – makro rezultati“ video sam konkretno rešenje. Nisam mogao a da ne probam. Ovo je ujedno bila i jedina pisana (na papiru) literatura kojom sam se služio. Pored ovog teksta, preporučujem da pročitate i taj.

ŠTA JE MAKRO?

U suštini, makroom se naziva niz aktivnosti pripisanih nekom tasteru. To se može postići na više načina, ali osnovni je pamćenje tog niza tokom rada. Jednostavno, kad primetite da neki niz aktivnosti ponavlja, vi to dodelite nekom tasteru i kasnije je dovoljan samo jedan pritisak da bi se ceo posao obavio vrlo brzo.

Određivanje funkcije nekog tastera može da se vrši na više načina. Najjednostavniji način je da se u toku rada, pritiskom na neku kombinaciju tastera, programu stavi do znanja da sledeću sekvencu treba da upamti. Drugi, složeniji način je da se napiše lista aktivnosti, kao i svaki drugi program, i da se ona dodeli nekom tasteru.

Drugi način pravljenja makroa izlazi iz okvira ovog teksta. No, kad sam već pomenuo tu pogodnost, evo nekoliko reči i o tome. Obično se ovakve mogućnosti vezuju za editore (*Qedit*, *WordPerfect*, *Brief*...). Međutim, postoje i u drugim programima (*QuattroPro*, *Lotus 1-2-3*...). *QEDIT* (R75/42) ima najjednostavniji tip makroa. Tu se iza imena tastera jednostavno navede lista aktivnosti iz velikog skupa koji je na raspolaganju. S duge strane, jedan od najsloženijih tipova makroa jeste onaj koji je ugrađen u poznati editor *Brief* (R68/28 i R75/40). Često se naziva makro jezikom, ali ja *Brief* volim da zovem makro-interpreterom.

Osnovna ideja nastala je pre izvesnog vremena i tada nisam imao u planu ništa više osim jednostavnog ignorisanja zahteva za prekidom. Zaista može biti neprijatno ako korisnik prekine program u presudnom trenutku. Zamislite, da podemo od najtrivijalnijeg primera, neki prelepo dizajniran ekran, a posle pritiska na Ctrl-Break dva znaka "C i prompt preko sredine. Nakon uspešno rešenog problema, shvatio sam da primenjena tehnika krije još čitav niz lepih mogućnosti.

POLAKO, PROCESORE

Kontrolu protoka informacija između periferije i procesora obavlja veznik. Veznik služi da prihvati podatke i da, u određenim slučajevima, od procesora zahteva njihovu obradu.

Potreba za prekidom se javlja ispunjenjem nekog uslova, a procesoru odgovarajući zahtev daje poseban veznik. Ovaj veznik je u stvari deo računara i naziva se kontroler prekida (*Interrupt Controller*), a uslov može biti pritisak na taster, pristigla informacija na komunikacioni port (COM), neki vremenski interval i drigo.

Kako se tokom rada računara generiše više vrsta zahteva za prekidom, da bi se tačno znalo koji će biti obrađen postoje različiti prioriteti u obradi prekida. Najviši prioritet ima prekid generisan svakih 55ms (IRQ0, PIT 8253/8254). Taj prekid zahteva pokretanje INT 8. Sledeći po prioritetu je IRQ1, a zahtev se javlja pritiskom na neki taster. Obrada ovog prekida počinje pozivom INT 9...

PIC 8259

Nakon svakog pritiska na taster, „programabilni kontroler prekida“ 8259 zahteva prekid za obradu tastature. Ovu komunikaciju neću detaljnije opisivati, ali moram da priznam da mi je bilo potrebno dosta vremena da nađem nešto više o njoj. Uglavnom, to je najniži nivo, direktna komunikacija i upravljanje pomoćnom „elektronikom“. Ipak, izdvajam najvažnije: dok traje prekid (obrada rutine za čitanje kodova tastature), ni jedan drugi prekid istog ili nižeg prioriteta neće biti obrađen. Kako ovaj zahtev ima visok nivo i veoma važnu funkciju, ako se ne stavi do znanja kontroleru da je obrada prekida završena računara će ostati blokiran.

Svaki kontroler ima jednu ili više adresa (port) preko kojih mu se procesor obraća. To se obavlja preko in i out komandi. Ovdje treba pomenuti da je adresa 8259 kontrole-

Milijan Jovanović

ra 20h i 21h (A0h i A1h), a da se završena obrada prekida signalizira slanjem 20h na adresu (port) 20h:

mov al, 20h;
out 20h, al;

Ovo je u primeru definisano pod imenom `reset_8259()` radi čitljivosti. Posle ove sekvence, sve se nastavlja tamo gde je stalo. Naravno, prekid mora da traje što kraće, inače bi se računaru vidno usporio.

INT 9

Od svih rutina za obradu prekida, najpoznatija je verovatno ova, za obradu tastature. Poziva se automatski, uvek kad je neki taster pritisnut. Posle dekodovanja primljenih podataka, beleži ih u nekom drugom obliku na posebnom mestu u memoriji (*keyboard buffer*). To mesto se nalazi u polju memorije gde BIOS čuva svoje promenljive (prvi segment) i ima mesta za 16 znakova. Kasnije se iz programa može proveriti da li je neki taster u međuvremenu prihvaćen ili ne, i adekvatno reagovati.

Tastatura je uređaj koji šalje računaru sve potrebne podatke na osnovu kojih on tumači da li je neki taster pritisnut ili ne. Svaka dirka ima jedinstven kod za pritisnut/otpušten taster, i on se čita na adresi (port) 60h. „Autorepeat“ je stvar tastature: ako držite pritisnut taster, ona će simulirati kao da se, u stvari, radi o uzastopnom pritiskanju (ali ne i otpuštanju) istog tastera.

Postoje dirke koje „ispuštaju“ i po nekoliko kodova. Na primer, sivi tasteri između osnovnog i numeričkog dela tastature (*kursori*, *insert*, *home*...) daju po dva znaka, a dirka Pause čak šest.

Razlika između pritisnutog i puštenog tastera je u sedmom bitu, koji je 0 ako je taster pritisnut, a 1 ako je otpušten. Na primer, pritisnuta dirka „A“ daje kod 1Eh, a otpuštena kod 9Eh. Kad sam već ranije pomenuo dirku Pause, moram da napomenem da je ona izuzetak, jer svih šest znakova daje odjednom: prvo ide niz od tri znaka da je taster pritisnut, a odmah zatim i da je otpušten. Ovo je verovatno zato što kod ove dirke *autorepeat* ne funkcioniše.

Sve „zanimljive“ kombinacije se posle prevođenja smeštaju u bafer. Bafer je kružni i postoje dva vektora, na početku (HEAD) i kraj (TAIL) niza. Kad se sustignu, onda je bafer pun ili prazan – jedina razlika je kad se to desilo: da li pri dodavanju ili pri uzimanju znaka. Kako ova rutina (INT 9) samo dodaje, onda to znači da je u pitanju pun bafer i u tom slučaju javiće se zvučni signal upozorenja.

Kontrolni tasteri (Shift, Ctrl i Alt) utiču na stanje statusne promenljive koja se nalazi na adresi 0040:0017. Pored ovoga, postoje još tri statusna bajta iz kojih se mogu uzeti

još neka obaveštenja. Vrlo je bitno da INT 9 može biti pozvan, a da se u baferu ne nađe ni jedan znak. Znači, dovoljno je proveriti sadržaj tih lokacija i iz njih detektovati da li je pritisnuta neka od navedenih dirki.

Neki tasteri izazivaju i posebne aktivnosti u INT 9. To su:

- Pause: čeka pritisak na ispravan ASCII znak
- Ctrl-Break
- Ctrl-Alt-Del: (obično) resetovanje računara
- Print screen: poziv INT 5
- System request: poziv INT 15h

Uz Ctrl-Break namerno ne stoji nikakvo objašnjenje, jer želim nešto više da kažem o tome. Kao prvo, to je, po meni, veoma loše rešeno pitanje. Dešava se mnogo nepotrebnih stvari, ali nećemo sad o tome. Iako se kaže da od viška glava ne boli, to ovdje nije slučaj. Evo šta se sve desi po pritisku na pomenutu kombinaciju:

- startuje INT 9

● INT 9:

- briše sve što se nalazi u baferu (izjednačava HEAD i TAIL)
- postavlja reč 0000h u bafer
- setuje bit 7 na lokaciji 0040:0071
- poziva INT 1Bh

● INT 1Bh:

- pre učitavanja operativnog sistema, ovaj poziv nema nikakvu funkciju. U BIOS-u se nalazi instrukcija IRET, na adresi na koju ukazuje ovaj vektor.
- po učitavanju DOS-a, ovaj vektor ukazuje na rutinu koja će podesiti određenu zastavicu tako da kasnije sam DOS pri pozivanju nekih podfunkcija proverava da li je potrebno prekidati proces. Ako treba, onda se poziva INT 23h.

Jednostavnim preusmeravanjem vektora INT 1B (ili INT 23h) na „praznu“ rutinu (da se sastoji samo od jedne IRET naredbe) može se sprečiti prekid u toku rada sa DOS-om, no to nema puno smisla. Ako već ne želimo prekidanje programa (i delovanja neke druge vrste), zašto se onda opterećivati onim izmenama koje se standardno dešavaju u toku izvršavanja INT 9?

Potpuna zabrana prekida i totalno ignorisanje svih aktivnosti najlakše se postiže ako se detektuje da je pritisnut Ctrl-Break pre standardne obrade INT 9. U primeru se može videti najčešće primenjivano rešenje ovog problema. Zapravo, radi se o tome da se „preuzme“ obrada ovog prekida, a na samom početku te nove rutine ispita da li je u pitanju Break ili nešto drugo. Posle ovoga treba omogućiti da stara rutina obavi posao koji je i do sada radila.

Tabela 1

word getkey(void)
int read(void)

– uzima i briše znak iz bafera; ako ne postoji onda će sačekati sledeći
– ova funkcija je osnova za interpretiranje makroa; treba je pozivati kad god vam u programu zatreba ulaz sa tastature

void stop(void)
void kill(void)
int list(void)

– prekid učenja ili izvršavanja makroa; reset
– poništavanje svih makroa

int save(FILE *out)
int load(FILE *in)

– primer za listanje definisanih makroa
– primer za snimanje definisanih makroa u datoteku

int start_key(word k)
int char_key(word k)

– učitavanje snimljenih makroa
– definicija tastera za početak/kraj učenja makroa
– definicija tastera za omogućavanje unosa jedne dirke u toku rada makroa (da se sačeka intervencija korisnika pre nastavka, eventualno za odgovor na neko uobičajeno pitanje)

int str_key(word k)

– definicija tastera za komandu „sačekaj prvi ENTER“ (slično kao prethodna mogućnost, ali se radi o nizu završenom sa ENTER)

void inkey_start()

– primer za rutinu koja treba da obavesti korisnika o početku učenja makroa i o tome za koji taster se vezuje makro

void inkey_end(void)
void test_break(void)

– slična funkcija kao prethodna, ali za obaveštenje o završenom učenju
– takođe „idejno rešenje“, a pokazuje kako obraditi Ctrl-Break.

Listing 1

```

//
// Računari 88
// 8/9/1992, Miljan M. Jovanović (mjova)
//
// delimičan kod, kompletna verzija se nalazi na Sezamu.
//

#include <dos.h>
#include <bios.h>
#include <stdlib.h>
#include <conio.h>

#include "macro.h"

void interrupt (*oldint9)(...);
#ifndef DISABLE_BREAK
void interrupt (*old_brk)(...);
#endif // DISABLE_BREAK

Keyboard Key;

char *code[512]= {
    "Null", "A", "B", "C", "D", "E", "F", "G", "H",
    .....
    "@BackSpc", "@Tab", "@Q", "@W", "@E", "@R", "@T", "@Y", "@U",
    "@I", "@O", "@P", "@[", "@]", "@Enter", "285", "@A", "@S",
    "CHR", "STR", "BackSpace", "Tab", "Enter",
    .....
};

//
//-----
#ifndef DISABLE_BREAK
void interrupt new_brk(...) {
    BREAK= TRUE;
}
#endif // DISABLE_BREAK

//
//-----
void interrupt newint9(...) {
    static word far *tail;
    static int i;
    static word subst[14][2]= {
        0x01b, 0xad00, // shift-esc
        0x52a0, 0xf500, // shift-ins
        0x47e0, 0xf700, // shift-home
        0x49e0, 0xf900, // shift-pgup
        0x53e0, 0xf600, // shift-del
        0x4fe0, 0xf800, // shift-end
        0x51a0, 0xfa00, // shift-pgdn
        0x48e0, 0xf200, // shift-up
        0x4be0, 0xf100, // shift-left
        0x50e0, 0xf300, // shift-down
        0x4de0, 0xf000, // shift-right
        0x3920, 0xc600, // shift-space
        0x1c0d, 0xa600, // shift-entar
        0, 0 // kraj tabele
    };

    #if(defined(DISABLE_RESET) | defined(DISABLE_BREAK))
    byte b;
    asm { cll;
        in al, 0x60;
        mov b, al;
        sti; }
    #endif

    #ifndef DISABLE_RESET
    if((KBDSTATUS & (KBD_CTRL | KBD_ALT) && b == 0x53) {
        ok_8255(); // Obaveštenje za PIC 8255 da je znak preuzet
        reset_8259(); // resetovanje 8259
        return;
    }
    #endif

    #ifndef DISABLE_BREAK
    if((KBDSTATUS & KBD_CTRL) {
        switch(b) {
            case 0x46: // Break pritisnut
            case 0xc6: // Break otpušten
                ok_8255();
                reset_8259();
                return;
            }
        }
    #endif

    tail= (word far *)MK_FP(0x40, KEYBD_TAIL);
    (*oldint9)();

    if((KEYBD_HEAD == KEYBD_TAIL && IBREAK)
        return;

    // upisan je znak u buffer, treba videti šta je...
    if((KBDSTATUS & KBD_CTRL) {
        switch(*tail) {
            case 0x01b: // ctrl-esc
                *tail= 0xee00;
                break;
            case 0x3920: // ctrl-space
                *tail= 0xc500;
                break;
            }
        }
    }

    if((KBDSTATUS & KBD_ALT && *tail == 0x3920) // alt-space
        *tail= 0xc400;

```

```

if((KBDSTATUS & KBD_SFT) { // pritisnut shift
    for(i= 0; subst[i][1]; i++)
        if(*tail == subst[i][0]) {
            *tail= subst[i][1];
            break;
        }
    }
}

/*****
/*
/* Clanovi klase Keyboard */
/*
*****/

// konstruktor objekta
Keyboard::Keyboard(void) {
    union REGS regs;

    c_brk= OFF;
    position= 0;
    status= MAC_OFF;
    mac= 0;
    ws= 0;

    start_macro_key= F1;
    get_char_key= F2;
    get_str_key= F3;

    if((KEYBRDMODE & 0x10)
        key_type= ENHANCED;
    else
        key_type= NO_ENHANCED;

    kill();
    stop();

    oldint9= getvect(0x09);

    #ifndef DISABLE_BREAK
    old_brk= getvect(0x1b);
    #endif // DISABLE_BREAK

    disable();
    setvect(0x09, newint9);

    #ifndef DISABLE_BREAK
    setvect(0x1b, new_brk);
    #endif // DISABLE_BREAK

    enable();
}

// destruktor
Keyboard::~Keyboard(void) {
    disable();
    setvect(0x09, oldint9);

    #ifndef DISABLE_BREAK
    setvect(0x1b, old_brk);
    #endif // DISABLE_BREAK

    enable();
}

word key;
union REGS regs;
word subst[7][2]= {
    0, 0xfc00, // break
    0x0e08, 0xcc00, // bs
    0x0f09, 0xcd00, // tab
    0x1c0a, 0xce00, // enter
    0x1c0d, 0xcf00, // enter
    0xe00a, 0xd000, // KP_enter
    0, 0 // kraj tabele
};

BREAK= FALSE;
while((KEYBD_HEAD == KEYBD_TAIL && IBREAK) {
    // dok program čeka na taster, ovde može postojati poziv
    // rutina za ispis sata i za čuvanje ekrana.
}
test_break(); // proveriti ctrl-break

regs.h.ah= key_type;
int86(0x16, &regs, &regs); // uzimam znak iz buffer-a
key= regs.x.ax;

for(i= 0; subst[i][1]; i++)
    if(key == subst[i][0]) {
        key= subst[i][1];
        break;
    }

if((key & 0xff) == 0xe0 || (key & 0xff) == 0xf0)
    key&= 0xf00;
return(key&0xff ? key&0xff : (0x0100|(key>>8)));
}

int Keyboard::read(void) {
    int key;
    while((key= _read()) >= 0xf00;
        return(key);
}

word Keyboard::_read(void) {
    if(status == MAC_PLAY) { // da li se izvršava macro?
        if(ws) { // da li treba uzeti znak za umetnuti niz?
            i= getkey();

```

```

if(i == ENTER) { // kraj umetnutog niza, označi kraj
    ws= OFF;
    if(position >= macro[mac].len)
        status= MAC_OFF;
    }
return(i);
} else {
i= macro[mac].keys[position++];
if(i == WAIT_CHAR) {
    i= getkey();
} else if(i == WAIT_STRING) {
    ws= ON;
    return(MAC_STRING); // vraća da treba da pokupi str
}
if(position >= macro[mac].len) { // makro je odsvirao ;
    status= MAC_OFF;
    }
return(i);
}
}

// ne treba odrađivati snimljenu sekvencu, treba obraditi
// običan znak sa tastature.

i= getkey();

// prvo provera da li pritisnuti taster neki od 'internih komandi'
if(i == start_macro_key) { // makro start/end REC taster
    if(status == MAC_REC) { // snimanje je već u toku (završi ga)
        macro[mac].len= position;
        status= MAC_OFF;
        inkey_end();
        return(REC_END);
    }

    status= MAC_REC; // znak da počinje snimanje
    position= 0; // od prvog znaka
    inkey_start(); // obaveštenje korisniku
    i= getkey(); // snimati pod ovim znakom
    if(status != MAC_REC) // pritisak na break prekida snimanje
        return(REC_END);
    inkey_start(i); // odštampa i znak koji se definiše

    // specijalni tasteri ('start/end', 'get char', 'get string' i ESC)
    // se ne mogu predefinisati jer nema smisla.
    if(i == start_macro_key || i == get_char_key || i == get_str_key ||
        i == ESC) {
        status= MAC_OFF;
        inkey_end();
        return(REC_END); // ne može sve da bude mac taster
    }
}

// da li je datom tasteru već pripisan niz znakova?
for(j= 0; j<MAX_MACROS; j++)
    if(macro[j].key == i)
        break;
if(j >= MAX_MACROS) { // makro nije definisan ranije
    for(j= 0; j<MAX_MACROS; j++) // nađi slobodno mesto
        if(macro[j].len == 0)
            break;
}
if(j<MAX_MACROS) { // nađeno prazno mesto ili
    mac= j; // mesto već postojećeg makroa
    macro[j].key= i;
    status= MAC_REC;
    return(REC_START);
}
status= MAC_OFF; // nema mesta za novi makro
inkey_end();
return(REC_ERR);
}

if(status == MAC_REC) {
// učenje makroa je u toku
// ako je korisnik naznačio da želi da ostavi prostor za
// kasnije dopunjavanje makroa (rupa u makrou za niz znakova)
// onda te znake ne treba beležiti.
if(ws) {
    if(i == ENTER) {
        ws= OFF;
        macro[mac].keys[position++] = WAIT_STRING;
        if(position >= MACRO_LEN) {
            macro[mac].len= position;
            status= MAC_OFF;
            inkey_end();
        }
    }
    return(i);
} else {
    if(i == get_char_key) {
        macro[mac].keys[position++] = WAIT_CHAR;
        i= getkey();
    } else if(i == get_str_key) {
        ws= ON;
        return(MAC_STRING);
    } else {
        macro[mac].keys[position++] = i;
    }
    if(position >= MACRO_LEN) {
        macro[mac].len= position;
        status= MAC_OFF;
        inkey_end();
    }
}
return(i);
}
}

```

```

// nije učenje, počinje izvršavanje makroa, ako treba
// prvo ide provera da li je taster makro?
for(j= 0; j < MAX_MACROS; j++) {
    if(macro[j].len != 0 && macro[j].key == i)
        break;
}

// ako jeste makro onda počinje izvršavanje
if(j<MAX_MACROS) {
    status= MAC_PLAY;
    mac= j;
    position= 0;
    i= macro[mac].keys[position++];
    if(i == WAIT_CHAR) {
        i= getkey(); // uzmi znak sa tastature umesto koda WAIT_CAH
    } else if(i == WAIT_STRING) {
        ws= ON; // treba pokupiti niz znakova (do prvog ENTER-a)
        return(MAC_STRING);
    }
    if(position >= macro[mac].len)
        status= MAC_OFF;
    }
return(i);
}

void Keyboard::stop(void) {
    if(status == MAC_REC) {
        macro[mac].len= 0;
        inkey_end();
    }
    status= MAC_OFF;
    ws= OFF;
}

void Keyboard::kill(void) {
    for(int j= 0; j < MAX_MACROS; j++)
        macro[j].len= 0;
    mac= MAC_OFF;
}

// provera da li korisnik zahteva prekid rada i
// obrada eventualnog prekida
void Keyboard::test_break(void) {
    if(brk)
        Key.stop();
    if(!c_brk) {
        brk= FALSE;
        return;
    }
    if(brk) {
        printf("\nL BREAK into program, 10:1");
        if(KBDSTATUS & KBD_CTRL)
            while(KBDSTATUS & KBD_CTRL);
        exit(0);
    }
}
}

```

Sažeti program za tastaturne makroe

Listing 3

```

#ifndef __MACRO
#define __MACRO

#include <stdio.h>

#include "keys.h"

typedef unsigned char byte;
typedef unsigned int word;

// direktive za prevodioc:
// ako je DISABLE_RESET definisano onda će resetovanje
// računara preko tastature (ctrl-alt-del) biti nemoguće
#define DISABLE_RESET

// ako je DISABLE_BREAK definisano onda će pritisak na
// ctrl-break kombinaciju biti bez efekta
#define DISABLE_BREAK

// bios promenljivie
#define KBDSTATUS (*(byte far *)0x4171u) // A C I S rS
#define KEYBRDMODE (*(byte far *)0x4961u)
#define KEYBD_HEAD (*(word far *)0x41A1u)
#define KEYBD_TAIL (*(word far *)0x41C1u)

#define inkey() (Key.read())
#define BREAK (Key.brk)
#define reset_8259() asm { cli; mov al, 0x20; out 0x20, al; sti; }
#define ok_8259() asm { in al, 0x61; mov ah, al; or al, 0x80; \
    out 0x61, al; mov al, ah; out 0x61, al; }

struct MACRO {
    int key;
    int len;
    int keys[MACRO_LEN];
};

/*****
*/
/* Zaglavlje klase Keyboard */
/*
*****/

class Keyboard {
    word i;

```

```

byte j;
byte ws; // čeka li niz?

word start_macro_key; // početak učenja makroa
word get_char_key; // 'rupa' za jedan znak
word get_str_key; // 'rupa' za više znakova do ENTER-a

byte position; // pozicija u makro
byte status; // off, play, rec
byte mac; // makro koji se obraduje (po redu)
MACRO macro[MAX_MACROS];

word read(void);
public:
byte brk; // da li je pritisnut break?
volatile byte c_brk; // da li je dozvoljen break
byte key_type; // tip tastature...

Keyboard(void); // konstruktor
Keyboard(void); // destruktor

int read(void); // ovom fn treba čitati tastaturu
word getkey(void); // interno, uzima znak iz bafera

```

```

void stop(void); // zauzavlja izvršavanje makroa
void kill(void); // poništavanje svih makroa
int list(void); // listanje makroa

int save(FILE *out); // snimanje makroa u datoteku
int load(FILE *in); // učitavanje makroa iz datoteke

int start_key(word k); // imenovanje tastera za 'snimanje/reprod. makroa'
int char_key(word k); // 'ubacivanje znaka'
int str_key(word k); // 'ubacivanje niza'

void inkey_start(unsigned key=0); // ispis da je počelo snimanje makroa
void inkey_end(void); // ispis (brisanje) poruke da je završeno snimanje

void test_break(void); // da li je pritisnut break?
};

extern char *code[512];
extern Keyboard Key;

#endif // _MACRO

```

Zaglavlja i definicije

Dakle, cilj nam je da ne „propustimo“ staroj rutini ono što ima loše posledice. Evo kako to možemo da uradimo: korisnik prvo pritisne Ctrl, i to izaziva (osnovni INT 9) setovanje bita 2 na lokaciji 0040:0017, zatim pritisak na Break treba pročitati sa adrese (port) 60h. Ako je ustanovljena opisana situacija, može se preći na resetovanje kontrolera i time završiti obrada INT 9. (Ovo je već kodirano u primeru, pa ako to rešava vaše probleme jednostavno treba omogućiti prevodiocu da i taj deo prevede. Direktiva za definisanje DISABLE_BREAK se nalazi u *macro.h* datoteci, jednostavnim uklanjanjem „/“ sa početka reda biće ignorisan zahtev za obradom Break iz stare INT 9 rutine.)

Činilo mi se u početku da je dobro onemogućiti Break i RESET. Ali, pokazalo se da je pitanje za diskusiju, jer to može imati i loše strane, naročito ako se radi o zabrani resetovanja računara sa tastature. Mislim da je bolje pravilno reagovati na zahtev za prekidanjem nego ga potpuno onemogućiti. Osim toga, vrlo je značajno kako će se ovo rešenje odraziti na ostale programe ako treba da rade zajedno (npr. *DesqView*).

PRATIMO BREAK

Sve ono što INT 9 radi u ovoj situaciji (pražnjenje bafera itd) nećemo dirati, već ćemo iskoristiti pogodnost da se na kraju poziva INT 1Bh. Ta rutina treba da signalizira da je Break zatražen, a mi ćemo iz programa, kad nama to bude odgovaralo, izvršiti prekid. Signal se svodi na postavljanje vrednosti promenljive **brk** (koja je član klase **Keyboard**) na TRUE. Primer za proveru tog signala (**Keyboard::test_break()**) ima dva dela: prvi, koji se uvek izvršava (prekidanje učenja/izvršavanja makroa – kontinuitet jer se bafer prazni) i drugi, koji obezbeđuje izlaz iz programa.

Ovde se koristi stanje promenljive **c_brk** (član klase **Keyboard**), koja je TRUE ako je dozvoljen izlaz iz programa i FALSE ako je zabranjen. Možda izlaz nije ni potreban, ali dovoljno je pritisakom na Ctrl-Break prekinuti neki unutrašnji proces. Ako se iz programa vrši npr. sortiranje baze, a to može da potraje, onda bi želja za prekidom značila da korisnik hoće da odustane od sortiranja, a ne napuštanje programa. U glavnoj petlji potprograma za sortiranje treba ispitivati promenljive **Key.brk** pa, ako je TRUE, vratiti kontrolu na mesto odakle je sortiranje zatraženo. Naravno, prekid treba obaviti u sigurnom trenutku, a ne u pola posla...

PROŠIRENA OPSERVACIJA

Često u programu zatreba da se upotrebi kombinacija Shift-CrUp ili Shift-CrDown (što je već standard za markiranje teksta) itd. Kako to rešiti? Ima više načina, no, predimo na stvar...

Osnovna rutina INT 9 ne beleži uvek stanje tastera ALT, CTRL i SHIFT. Ako postoji neka ASCII kombinacija (npr. shift+slovo), onda će ona biti pravilno zabeležena kao veliko slovo. Međutim, u nekim uslovima se ignoriše stanje pomoćnih tastera i pamti samo glavni. Takve kombinacije su Shift-Cursor (na nenumeričkoj tastaturi) – u baferu ostaje samo znak da je pritisnut kursor, a stanje Shift-a se ignoriše.

Svaka dirka (ili kombinacija sa Alt, Ctrl ili Shift) ima jedinstven broj dužine jedne reči (dva bajta). U tim uslovima ima „mesta“ za puno raznih kombinacija, a sve, naravno, nisu zauzete, pa ih možemo koristiti za svoje potrebe. Da

bi izbor bio lakši, prvo treba pogledati kako se „dešifruje“ taj kod i ima li nekog smisla u njemu?

Jednostavno, postoje dve vrste dirki: jedne imaju svoje mesto u ASCII tabeli – to su slova, brojevi i simboli – a ostale se koriste za posebne namene (funkcijski tasteri, ESC, kursori itd). Ako dirka ima neku ASCII vrednost, onda je ta vrednost, jednostavno, zabeležena u nižem delu reči koji je opisuje. U tom slučaju, viši bajt nosi uvek isti broj, bez obzira na stanje kontrolnih tastera. U drugom slučaju, niži bajt je nula, a viši predstavlja kompletnu informaciju o pritisnutom tasteru i indirektno javlja da li je upotrebljen Alt, Ctrl ili Shift.

Pogledajmo to na primeru nekih slova:

dirka	+Shift	+Ctrl	+Alt
A	1E61	1E41	1E01
B	3062	3042	3002
C	2E63	2E42	2E03

Ako je pritisnuto slovo „A“, u baferu će se nalaziti reč 1E61h. Dakle, niži bajt je različit od nule i predstavlja ASCII kod malog slova „a“. U kombinaciji sa dirkom Shift, ta reč će imati vrednost 1E41h, itd. Kompletna tablica se može sačiniti uz program koji bi vršio očitavanje tastature preko BIOS-a, a zatim ispisivao kodove na ekran. Preporučujem ovakav način čitanja:

```

regs.h.ah = 0x10; // broj funkcije
int86(0x16, &regs, &regs); // poziv rutine
printf(„%04x\n“, regs.ax); // ispis na ekran

```

Onaj deo tabele koji nas zanima je vezan za tzv. sive dirke (između numeričke i osnovne tastature):

dirka	+Shift	+Ctrl	+Alt
Esc	011B	011B*	011B*
Space	3920	3920*	3920*
Enter	1C0D	1C0D*	1C0A
Ins	52E0	52E0*	92E0
Home	47E0	47E0*	77E0
PgUp	49E0	49E0*	84E0
Del	53E0	53E0*	93E0
End	4FE0	4FE0*	75E0
PgDn	51E0	51E0*	76E0
Up	48E0	48E0*	8DE0
Left	4BE0	4BE0*	73E0
Down	50E0	50E0*	91E0
Right	4DE0	4DE0*	74E0

Zvezdica označava kodove koje treba zameniti jer već opisuje neku dirku. Ispitivanjem sadržaja bafera, i da li je pritisnut neki kontrolni taster (shift, ctrl ili alt), možemo postaviti naš kod. Ova zamena može biti obavljena u različitim delovima programa, ali samo dva rešenja zadovoljavaju:

- 1) pri preuzimanju koda iz bafera
- 2) posle svakog starog INT 9

Prvi način ima olakšavajuću okolnost jer zaobilazi nezgodnu promenu i doterivanje INT 9 rutine, a ne vrši se ni direktno upisivanje u bafer (to treba izbegavati zbog programama koji će raditi u istom okruženju). Međutim, ovo rešenje ima i mane. Ako je uzimanje znaka zakasnilo, jer je program bio zauzet nekim drugim stvarima, može se desiti da pojedini kodovi budu pogrešno protumačeni.

Drugi način je onaj čije rešenje se nalazi u priloženom primeru. Ukratko, napisali smo novu INT 9 rutinu koja posle izvršenog dekodovanja podataka pristiglih sa tastature (poziv starom INT 9) jednostavno proveru da li nešto treba zameniti. Tako će se zamena izvršiti u sledećim situacijama:

	stari	novi
Shift-Esc	011B	ED00
Ctrl-Esc	011B	EE00
Shift-Space	3920	C600
Ctrl-Space	3920	C500
Alt-Space	3920	C400
Shift-Enter	1C0D	A600
Shift-Ins	52E0	F500
Shift-Home	47E0	F700
Shift-PgUp	49E0	F900
Shift-Del	53E0	F600
Shift-End	4FE0	F800
Shift-PgDn	51E0	FA00
Shift-Up	48E0	F200
Shift-Left	4BE0	F100
Shift-Down	50E0	F300
Shift-Right	4DE0	F000

NEŠTO VIŠE O PRIMERU

Možda je čudno zašto baš C++? Pa jednostavno, upotrebom konstruktora i destruktora olakšano je preuzimanje i vraćanje vektora INT 9 (čitaj: manje briga), a promenljive ostaju vidljive samo u okviru objekta. Objekt se kreira statički i treba da postoji samo jedan. Nije nikakav problem da se ovaj primer prebaci u „običan C“, ali treba voditi računa da se izaberu prikladnija imena promenljivih.

Klasa **Keyboard** se sastoji od nekoliko funkcija, koje su prikazane u tabeli 1.

Programer je dostupan makro **inkey()** definisan u **macro.h** samo radi čitljivijeg izvornog koda. **inkey()** će vratiti kod pritisnutog tastera ili će sačekati da se u baferu pojavi neki znak. U okviru funkcije **getkey()** vrši se zamena koda koji imaju istu ASCII vrednost, ali se radi o sasvim drugom tasteru. Zato se vrši zamena za BackSpace, koji je u stvari Ctrl-H, zatim za Tab, itd. Ova rutina vraća neki broj od 1 do 511. Svi brojevi nisu dodeljeni, a tačan raspored nalazi se u datoteci *keys.h*.

Zanimljivo je da se postavljenjem nekih rutina u petlju označenu u **getkey()** funkciji može dodati par vrlo lepih mogućnosti. Ako se odatle poziva rutina za ispis sata i rutina za čuvanje ekrana, onda vaš program dobija još zanimljivih karakteristika. Na primer, u redu za čuvanje ekrana: ako prođe neki vremenski period neaktivnosti, onda se lepo snimi sadržaj ekrana, a zatim obriše. Međutim, možda se pitate kako će raditi sat iz jedne rutine za obradu tastature. Pa, vrlo lepo! Ako se vaš program oslanja na unos sa tastature, onda će brza rutina za štampanje sata biti neprimetna, a korisnik neće ni znati da se sat obnavlja samo dok čeka na taster.

Na kraju, skrećem vam pažnju na neka ograničenja: pri prevodenju ovog primera mora se isključiti provera steka *stack warning* i nije dozvoljena upotreba „registar“ promenljivih. Sve ovo vezano je za upotrebu službene reči *interrupt*. U okviru ovog teksta dati su, zbog uštede u prostoru, samo najneophosniji primeri, a kompletan listing potražite na Sezamu, u direktorijumu RSOFT.

U ZNARU BKUJA /

Godišnje takmičenje završili smo jednom zbilja lakom pitalicom i bili nagrađeni sa 49 tačnih odgovora – dobar deo je stigao preko Sezama što pokazuje da je elektronska pošta dostupna sve većem broju naših čitalaca!

Trebalo je, da se podsetimo, pronaći bar jedan par prirodnih brojeva A, B takav da $AB(A+B)$ nije deljivo sa 7 dok je broj $(A+B)^7 - A^7 - B^7$ deljiv sa 7^7 . Najpre ćemo $(A+B)^7 - A^7 - B^7$ napisati u obliku $7AB*(A+B)*(A^2+AB+B^2)^2$ – kada vam neko kaže ovaj izraz, lako ćete pokazati da je identičan polaznom, ali transformisati prvi u drugi... za to je trebalo biti na matematičkoj Olimpijadi, na kojoj je zadatak i postavljen! Obzirom da ni A, ni B ni $A+B$ nisu deljivi sa 7, ostaje da je A^2+AB+B^2 deljivo sa 7^3 , tj. sa 343. Prvi par koji zadovoljava ovaj uslov je očito (1,18) što je ujedno i rešenje zadatka.

Rešenja poput izloženog obično ćete naći u raznim zbirkama, ali ne treba zbog njih da se kompleksirate – setiti se opisane transformacije zbilja deluje volšebno, ali stvar uopšte nije tako teška kao što izgleda. Naime, čim se radi o deljivosti treba se pozabaviti nekim razvojem na faktore, a kada pogledate izraz $(A+B)^7 - A^7 - B^7$ sa jedne i izraz $AB(A+B)$ sa druge strane, gotovo da se nameće ideja da se drugi može izvući kao faktor prvog – čim se pominje A ili B na neki stepen, očekuje se

da se u rastavljanju toga pojave faktori oblika $(A+B)$, AB i slični! Dakle, ima osnova pretpostaviti da se $(A+B)^7 - A^7 - B^7$ može napisati u obliku $AB*(A+B)*P(A,B)$, gde je $P(A,B)$ nekakav polinom u kome učestvuju A i B. Stepen tog polinoma očito treba da bude za dva manji od stepena polaznog (pošto je $AB*(A+B)$ izvučeno ispred zagrada), a pošto je stepen polaznog polinoma 6 (neka vas ne zbuni pominjanje sedmih stepena – kada se $A+B$ digne na sedmi, tu će se pojaviti A^7+B^7 , ali će se to skratiti sa ostatkom izraza) rešeni stepen će biti 4. Dakle, $P(A,B)$ se u najopštijem obliku može napisati kao $pA^4+qA^3B+rA^2B^2+sAB^3+tB^4$ gde su p, q, r, s i t konstante koje treba odrediti. Zatim treba obaviti sva potrebna množenja i doći do dva polinoma koja treba da budu identična za svako A i B. Na osnovu toga, recimo zamenom nekih pažljivo izabranih konkretnih vrednosti A i B, može se konstruisati sistem jednačina čijim se rešavanjem, ako je polazna pretpostavka bila dobra, određuju željeni koeficijenti.

Slika 1 pokazuje kako se problem mogao rešiti uz malo manje

matematičke intuicije i malo više oslanjanja na resurse računara – najzad, „Računari“ su računarski a ne matematički časopis! Do rešenja se primenom programa nalik na izloženi zbilja dolazi za trenutak.

Obzirom da je većina primljenih rešenja bila relativno slična, nagrade su dodeljene izvlačenjem. 98000 dinara dobija **Miodrag Martinić** iz Arandelovca,

73500 dinara **Tatjana Velimirović** iz Beograda a 49000 dinara **Jan Slička** iz Padine. Pohvale su zaslužili i **Aleksandra Stanić**, **Petar Beličev**, **Igor Ikodinović**, **Jozef Kratica**, **Stojan Miloradović**, **Dejan Mitić**, **Nikola Nedović** i **Miloš Pruvlović**.

Rezultate godišnjeg takmičenja objavljujemo u sledećim „Računarima“ – dopustimo još par dana roka za zakasnele odgovore.

WINDOWS PITALICA

Na slici 2 data je tabla neobičnog oblika koja se sastoji od 33 polja koja su na očigledan način obeležena sa A3-G5. Na svakom polju osim centralnog (D4) nalazi se po jedna kuglica. U svakom potezu jedna kuglica se može prebaciti preko (horizontalno ili vertikalno) susedne na prazno polje, pri čemu „preskočena“ kuglica nestaje sa table. U početnoj poziciji se, na primer, kuglica sa B4 može prebaciti na prazno polje D4, pri čemu kuglica sa C4 nestaje (potez obeležavamo sa B4-D4). Zatim se, na primer, kuglica sa C2 može prebaciti na C4 (obeležavamo to sa C2-C4), pri čemu nestaje kuglica C3 i tako dalje. Vredi pomeniti da „dijagonalno“ prebacivanje nije dopušteno – u poziciji na-

stoj posle prethodnih poteza se, na primer, ne sme odigrati E5-C3. Igra prestaje kada stanje na tabli bude takvo da ni jedan potez nije moguć – nema kuglica pokom potezu jedna kuglica se može prebaciti preko (horizontalno ili vertikalno) susedne na prazno polje, pri čemu „preskočena“ kuglica nestaje sa table. U početnoj poziciji se, na primer, kuglica sa B4 može prebaciti na prazno polje D4, pri čemu kuglica sa C4 nestaje (potez obeležavamo sa B4-D4). Zatim se, na primer, kuglica sa C2 može prebaciti na C4 (obeležavamo to sa C2-C4), pri čemu nestaje kuglica C3 i tako dalje. Vredi pomeniti da „dijagonalno“ prebacivanje nije dopušteno – u poziciji na-

Slika 1

```

/*
 * Rešenje 80. Pitalice
 * Prema programu Miloša Pruvlovića
 * "Računari 89"
 */
#include <stdio.h>
#include <stdlib.h>

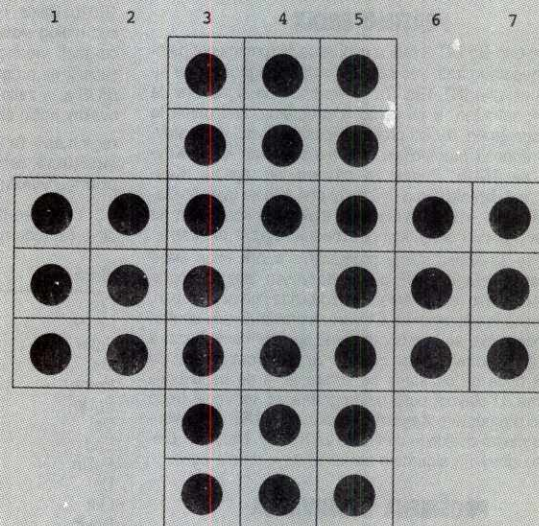
/* LIMIT predstavlja maksimalnu vrednost zbira A+B do
 * koje program treba da traži brojeve A i B */
#define LIMIT 500

void main()
{
    const unsigned long int S=7*7*7*7*7*7*7; /* 7**7 */
    unsigned long int Zbir,A,Pow7[LIMIT];
    Pow7[1]=1; /* i**1=1 */
    for(Zbir=2;Zbir<LIMIT;Zbir+=((Zbir%7==6 ? 2 : 1)))
    {
        /* Menjaj Zbir=A+B od 2 do limita, ali tako da nije deljiv sa 7 */
        Pow7[Zbir]=((((Zbir*Zbir)%S)*Zbir)%S)*Zbir%S;
        Pow7[Zbir]=((((Pow7[Zbir]*Zbir)%S)*Zbir)%S)*Zbir%S;
        for(A=1;Zbir-A>=A;A+=((A%7==6 ? 2 : 1)))
        {
            /* Probaj sve A takve da A bude manji od dva broja i da A%7!=0 */
            if((Zbir-A)%7!=0) /* Ako ni B nije deljiv sa 7 */
            if((((Pow7[Zbir]+2*S)-Pow7[A]-Pow7[Zbir-A])%S==0))
            {
                /* Ako brojevi ispunjavaju uslov zadatka, ispiši ih i kraj */
                printf("Traženi brojevi su: A=%ld, B=%ld\n",A,Zbir-A);
                exit(0);
            }
        }
    }
}
    
```

Programersko rešenje pitalice 80

KAKO DOSTAVITI REŠENJE

Rešenja pitalice, zajedno sa obrazloženim programima koje ste napisali, pošaljite na uobičajenu adresu: „Računari“ (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd tako da pristignu pre 25. februara 1993. Sva pisma sa korektnim rešenjima konkuriraju za novčane nagrade od 98.000, 73.500 i 49.000 dinara dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete poslednje tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj. Osim običnom rešenju ove pitalice možete da pošaljete i elektronskom poštom posredstvom Sezama – koristite menije REDAKCIJA, PITALICE, REŠENJE ili, iz komandnog moda, pošaljite ličnu poruku korisniku pitalice (mail write pitalice).



82: Do rešenja se dolazi u _____ poteza.

Ime i prezime _____

Adresa _____

Mesto _____

Kompjuter _____ Vreme _____

Identifikacioni broj:

iz broja tel. god. rod. po izboru

Virtual Library of Faculty of Mathematics - University of Belgrade

Rezultat funkcije

Do sada smo više puta raspravljali o prenosu argumenata funkcija, redosledu po kom se on vrši, promenjivom broju argumenata i slično. Vraćanje rezultata funkcije je nekako uvek delovalo očigledno, najčešće zato što C-ovske funkcije u najvećem broju slučajeva vraćaju običnu *int* vrednost, ređe *long* i još ređe pointer. Rezultat funkcije se zapravo najčešće koristi kao indikator uspešnosti rada po konvenciji: ako je 0, funkcije je završila posao korektno, ako je rezultat različit od nule, negde je došlo do greške i tada je rezultat sam kod greške. Ova konvencija je nastala iz biblioteka u kojima je 99% funkcija pisano na ovaj način. Kada funkcija treba da vrati bilo šta složenije, redovno se koristi mehanizam gde se kao argument funkciji prenosi pointer na područje u koje će ona smestiti rezultat.

Pretpostavimo da imamo funkciju koja vrši pomeranje neke tačke P, pri čemu je tačka opisana strukturom:

```
typedef struct POINT {
    int x;
    int y;
};
```

U C-u se skoro nikada takva funkcija ne piše na sledeći način:

```
struct POINT fun( struct POINT p ) {
    p.x += N;
    return p;
}
```

već po pravilu:

```
int fun( struct POINT *p) {
    p->x += N;
    return 0;
}
```

Prvi slučaj je toliko neobičan za C da verujem da se mnogi pitaju da li je uopšte tako nešto moguće. Ili, da malo zaplašimo veličinom, da li je nešto ovako moguće:

```
typedef struct bufs {
    char buf1[16000];
    char buf2[8000];
};
struct bufs fun( struct bufs b ) {
    // obavi neki posao sa baferima..
    return b;
}
```

Pored poruke o prekoračenju steka koju možete lako eliminisati parametrom **-F 8000** (stek 32k), program će raditi sasvim korektno. I zaista, rezultat funkcije nije ni na koji način ograničen – sve što može biti argument funkcije, može biti i njen rezultat.

Oni koji imaju više iskustva u programiranju na PC računarima sigurno osećaju da ovdje ipak nešto nije u redu. Sve one barijere od 64K, skroman inicijalni stek (4K) ukazuju da sa ovakvim funkcijama može biti problema. Problema zaista ima, ali ne na nivou ispravnog funkcionisanja, već na nivou performansi programa pisanog na ovaj način.

Ako pokušate da zamislite šta se dešava u toku izvršavanja poziva ove funkcije, brzo ćete spoznati razloge zašto se ovako nešto retko koristi. Prvo, pre poziva funkcije, potrebno je parametar preneti na stek. U našem slučaju to znači da će stek pointer biti "spušten" za 24.000 i sadržaj cele strukture biti prekopiran u ovu zonu. Potom sledi poziv funkcije koja na steku vidi ovaj argument, nad njim obavi svoj posao i do te tačke je sve potpuno standardno. Trenutak **return b** je zanimljiv – šta da se uradi sa tih 24.000 bajtova koje treba nekako vratiti nazad?

Postoji nekoliko mogućnosti. Pošto se na steku nalazi kopija originalne strukture, funkcija bi prosto mogla da završi rad vraćajući adresu parametra. Kompajler bi po povratku generisao kod koji sa steka kopira ovaj sadržaj u promenljivu kojoj dodeljuje rezultat funkcije i potom "očisti" stek, što je ina-

če standardna procedura za funkcije C tipa. Ova mogućnost zvuči lepo, ali je nemoguća iz dva razloga. Prvi je što se funkcija može deklarirati kao **passcal** tip, pri čemu će stek očistiti sama funkcija pre povratka, čime parametar ostaje nezaštićen od prepisivanja u toku interapt rutina koje mogu nastati između povratne instrukcije i završetka kopiranja sadržaja u promenljivu.

Druga, naravno očiglednija je što funkcija ne mora imati ni jedan argument istog tipa kao što je i rezultat, pa na steku neće biti prostora, recimo:

```
struct bufs fun( void ) {
    struct bufs b;
    // obavi neki posao sa baferima
    return b;
}
```

Kompajler može pre poziva funkcije odvojiti na steku prostor za rezultat, ali bi to značilo prilično opterećenje steka, koji je i ovako ograničen. Zato se koristi inicijalni segment za podatke, u kome se rezerviše potreban prostor koji služi kao privremeno skladište za povratak rezultata funkcije. Dakle, u našem primeru imamo prvo kopiranje 24.000 bajtova na stek, potom kopiranje 24.000 bajtova u rezervisanu zonu i, na kraju, još jedno kopiranje iz te zone u samu promenljivu kojoj smo dodelili rezultat funkcije. Kada se ovo uporedi sa prvim primerom, u kome se prenosi pointer na strukturu i koji zahteva samo jedno smeštanje jedne adrese na stek, jasno je koliko program ima "posla preko glave" (*overhead*) da udovolji našoj želji.

Ipak, u nekim situacijama ovaj mehanizam ne možemo izbeći. Najtipičniji slučaj je rad sa realnim promenjivama, koje su dovoljno "velike" da ne mogu biti prenete u registrima, pa se koristi pomenuta tehnika. Bilo bi veoma naporno obavljati operacije na sledeći način:

```
void sqr(float *r) {
    *r *= *r;
}
```

iako je on sa stanovišta brzine izvršavanja efikasniji od standardnog:

```
float sqr(float r) {
    return r*r;
}
```

Prvim metodom bismo jednim potezom izgubili mogućnost pisanja složenih numeričkih izraza, što je, naravno, previska cena čak i za značajnije dobitke u brzini.

Život promenjive

Nepažljivim prihvatanjem samo uopštene ideje da treba koristiti prenos parametara i rezultata po referenci (pointerima), a ne po vrednosti, možete lako upasti u jednu grešku koja se često dešava početnicima. Pogledajte sledeći primer:

```
char *getname(char *prompt)
{
    char name[80];
    printf("%s: ",prompt);
    gets(name);
    return name;
}
main()
{
    char *username;
    username = getname("Vase ime");
    printf("Vase ime je: %s",username);
}
```

Ovaj primer će se ponašati po svim poznatim zakonima koje važe za ozbiljne bagove – proba će dati uvek korektan rezultat, a kad program počne stvarno da se koristi, dobijaćete u potpuno nepravilnim intervalima umesto imena korisnika "kuke i kva-ke" i naravno, kad ponovo krenete da testirate opet će sve biti u najboljem redu. Ako imate baš izuzetnu sreću, greška će isplivati odmah kod prvog testira-

nja, ali će se potom opet sve odvijati "kako treba" – ponovljene probe neće dati nikakve sumnjive rezultate. Gde je greška?

Greška se krije u promenljivoj **name**, koja je deklarirana kao dinamička, odnosno automatska promenljiva. Posledica ovakve deklaracije nije samo u tome da se promenljiva "ne vidi" van funkcije, već i da je njen život vezan za funkciju – rada se po ulasku i umire na kraju. Ono što zbunjuje je to da je teško pojmove "rađanje" i "umiranje" vezati za nešto kao što je sadržaj memorije mašine. Nema, naravno, nikakve poetike u onome što se dešava. Na ulasku u funkciju, prostor za promenljivu se otvara na steku – po izlasku iz funkcije prostor se oslobađa, ali zaista važi da je u memoriji i dalje sadržaj promenjive **name**. Pitanje je samo dokle. Do trenutka kada pozovemo novu funkciju koja će iskoristiti isti prostor i po njemu ispisivati svoje podatke, ili, što je teže uhvatljivo, dok se ne desi inače česta pojava u računaru, neki od interaptova koji takođe koristi stek da sačuva vrednost registara.

Jedno pitanje i tri odgovora

Da je povremeno listanje dokumentacije o bibliotekim funkcijama, čak i ako nemamo neki konkretan povod za to, korisna stvar pokazao je slučaj jednog programera sa Sezama. Kako od onoga što na pitanje **filename**: odgovori korisnik napraviti sredenu stazu do datoteke? Kombinacija koje korisnik može da smisli da bi se referisano na istu datoteku je zaista mnogo, od recimo `.\pera.txt` preko `c:\test\pera.txt` ili `c:.\pera.txt`, pa sve do pakosnog `.\..\..\..\pera.txt`. Za otvaranje datoteke nije mnogo važno kako je naziv zadat, pa čak i ako želimo lepo da ga ispišemo na ekranu (kad već hoće tako – neka mu!), ali ima situacija kada je ovo veoma bitno. Tipičan primer je funkcija **copy**, koja treba da odredi da li su naziv izvorne i odredišne datoteke isti – ovo možemo detektovati ako uporedimo putpuno uređene nazive (ima li drugog načina?).

Tri su rešenja bila ponuđena. Prvo, tipično za programera koji ima nešto slobodnog vremena je "daj da napišemo tu funkciju u par redova za sredivanje staze. Par redova u ovom slučaju nije baš par, mada nije, naravno, ni mnogo više. Drugo rešenje je opet tipično za programera koji nema ni malo slobodnog vremena:

```
getcwd(pamti_dir,64);
chdir(ono_sto_je_korisnik_zadao);
getcwd(kako_je_to_sredio_dos,64);
chdir(pamti_dir);
```

Ideja je da prepustimo DOS-u da pređe u direktorij koji je korisnik zadao, pa da onda od njega uzmemo tačan i sreden naziv. Primer ima malih nedostataka, jer se odnosi samo na direktorije (ako postoji i naziv datoteke, on bi morao prvo da se skine, a to zahteva par linija koda – prvo treba proveriti atribut onoga na šta staza ukazuje, da li se radi o direktoriju ili datoteci). Treba dodati i proveru da li staza ukazuje na neki drugi disk, preći na njega, pa tek potom uzeti **getcwd** (ili obratno, sve jedno).

Treće je, naravno, netipično rešenje. Jedan od korisnika Sezama je verovatno u listanju dokumentacije naleteo na funkciju **_fullpath** koja obavlja baš ovaj posao:

```
char *_fullpath(char *buf, char *path, int max_len);
```

Staza zadata u **path** će u sredenom obliku biti upisana u **buf** – pri tome se vodi računa i o oblicima `c:\pera.txt`, u kom slučaju funkcija umeće naziv aktivnog direktorija.

Da priča bude kompletna, prvi programer koji je postavio pitanje ipak ne spada u prvu kategoriju u onoj šaljivoj podeli, niti podela ima ikakve veze sa konkretnim ljudima iz ove priče. Zaplet je nastao zbog Borlanda, koji je funkciju **_fullpath** uveo tek u verziji BC 3.0, pa je i nije bilo moguće naći u dokumentaciji iz verzije 2.0, koja je bila konsultovana.

607 [OS.WINDOWS, TAB.EXCEL; PC]: Gang ekrani *Windows* aplikacija su sve popularniji. Evo jednog trika za *Excel 4.0*. Ostavite samo jedan aktivan ali prazan *spreadsheet* i osnovni *toolbar* ispod meni linije. Kliknite desnim tasterom miša na prazan prostor *toolbar*-a i izaberite opciju *Customize*. Nakon toga izaberite iz liste alata *Custom* i odvučite ikonicu za *Solitair* na *toolbar*. Kliknite na OK u prvom dijalogu prozoru, pa na *Close* u drugom. Držeći pritisnute Ctrl i Shift, kliknite na *Solitair*. U donjem levom uglu ekrana pojavice se ikona *Lotusa 1-2-3* koja će početi da se kreće ka sredini ekrana. Iznutra će početi da je razjeda roj insekata (*bugs*), koji će najzad izleteti na ekran. Tada kao Zoro osvetnik doleće *Excel 4.0* ikona i izbacuje Lotus sa ekrana, zajedno sa bagovima. Na ekranu se ispisuje ponedono *No Problem*, a ponekad i lista ljudi koji su stvorili *Excel* [Preuzeto sa Sezama, autor Zoran Kehler (*zkehler*)].

608 [OS.WINDOWS; PC]: Evo gang *screen*-a za *Norton Desktop for Windows*. Kliknite na *Help* iz menija, izaberite *About*, postavite kursor na desni crtež ikone NDW i kliknite 2-3 puta da nestane ispis iznad nje. Zatim istovremeno pritisnite i otpustite slova N D W (eventualne upozoravajuće signale ignorišite). Ponovo kliknite na NDW ikonicu i posle par sekundi pojavice se slika tvoraca ovoga paketa. Ako uokvirite sliku nekog od njih, ona će se automatski uvećati. Sve to vrlo lepo izgleda na skupljim video karticama, dok na *Hercules*-u ne izaziva naročit utisak [Preuzeto sa Sezama, autor Jurij Titov (*jtiov*)].

609 [P.J.TPASCAL; PC]: Možda će vas zanimati kako popularni programski jezici generišu pseudoslučajne brojeve. Na slici 1 je asemblerski listing nastao prevodjenjem *Turbo Pascal* programa u kome je pisalo samo *a:=random(255)*. Ostatak procedure, koji ovde zbog ograničenog prostora ne objavljujemo, je na klasičan način svodio 32-bitni slučajni broj na interval [0,255]. Kvalitetom ovog generatora slučajnih brojeva nismo baš oduševljeni.

610 [P.J.C.609; PC]: Tako to radi *Turbo Pascal*, a na slici 2 vidimo kako isti posao obavlja *Turbo C*. Za razliku od prethodnog listinga, koji je nastao disasembliranjem, ovaj je preuzet iz listinga *Run Time* biblioteke 5.0.

611 [OS.WINDOWS.556; PC]: Pre par meseci smo, govoreći o *swap* fajlu *Windows*-a 3.1, rekli da virtuelna memorija može da bude najviše četiri puta veća od raspoloživog slobodnog RAM-a, dok *swap file* ne može da bude veći od polovine slobodnog prostora na disku. Pokazuje se da se parametrom *PageOverCommit=nn* zadaje faktor kojim se (u trenutku startovanja *Windows*-a) množi količina raspoložive memorija, a zatim se dobijeni rezultat „odseca plafonom” od 32 megabajta. Dozvoljene vrednosti za NN su od 1 do 20, a podrazumevana vrednost je 4. Polovina raspoloživog prostora na disku je samo preporučena veličina *swap file*-a, ali se ona može učiniti i većom, i to iz *Virtual Memory* menija u okviru *386Enhanced* opcije *Control Panel*-a. *Swap file*, već smo naglasili, mora da bude kontinualan [Preuzeto sa Sezama, autor Viktor Obuljen (*viktor*)].

612 [P.J.TPASCAL; PC]: Zgodan trik za korisnike *Turbo Pascal*-a koji imaju potrebu da izvorni kod programa prevode na raznim konfiguracijama – u editoru dva puta pritisnite Ctrl O i na početku programa pojavice se spisak svih direktiva prevodiocu onako kako je podešen u *Options* meniju. Sada taj izvorni kod možete bez problema prevesti na bilo kojoj insta-

laciji *Turbo Pascal*-a gde su, recimo, izabrane sasvim različite opcije. Mogući izgled onoga što će se pojaviti na početku programa po primeni ovog trika je: (* \$A+, B-, D+, E+, F-, G-, I+, L+, N+, O-, R-, S+, V+, X- *) (* \$M 16384,0,655360 *) [Preuzeto sa Sezama, autor Ljubiša Adžemović (*adzem*)].

613 [OS.WINDOWS.571.518; PC]: Mnogi čitaoci su primetili da komande *IconTitleFacename=ime_fonta* i *IconTitleSize=velicina_fonta*, ugrađene u WIN.INI, ne funkcionišu ili delimično funkcionišu. Stvar je u tome što je njihovo dejstvo ograničeno na „čist” *Windows 3.1*, ali ne i na *Norton Desktop* koji mnogi koriste. Ako je *Norton Desktop* ipak instaliran, menja se samo *Control Panel*-u, sličice ostavljene na *Desktop*-u i neki dijalozi [Preuzeto sa Sezama, autor Zoran Kehler (*zkehler*)].

614 [OS.UNIX; PC]: Popularni *Coherent Unix* na nekim pločama neće da se instalira ako je hard disk definisan u *Setup*-u kao 47 (*user type*). U slučaju diska *Seagate 157A*, tip treba definisati kao 14 i sve će biti normalno. U slučaju drugih diskova... može biti problema. [Preuzeto sa Sezama, autor Davor Magdić (*d.m.*)].

615 [HARD; PC]: U poslednje vreme često se pominju VESA kompatibilne video kartice, a nije malo onih koji ne znaju o čemu se radi. VESA (*Video Electronics Standard Association*) je udruženje proizvođača koje se prvenstveno bavi *SuperVGA* karticama. Jedan od glavnih ciljeva udruženja bilo je usvajanje standarda za PC grafiku u rezolucijama preko 640*480, što bi omogućilo da aplikacije bez posebnih drajvera rade na svim VESA kompatibilnim karticama. VESA standardom je definisano više grafičkih modova na nivou softverskog interfejsa prema kartici (implementacija BIOS-a). U najvažnije stvari spada standardizacija *bank-switching*-a (u memorijskoj mapi se vidi samo 256 kilobajta video RAM-a), ravni za boje (*bit-plane*), pristup paleti, hardverski kursor itd. VESA asocijalija definisala je čak i svoj tip proširene ISA (AT-BUS) sabirnice svane VESA LOCAL BUS [Preuzeto sa Sezama, autor Danko Jevtović (*danko*)].

616 [HARD; PC]: Kada smo već počeli sa hardverom, da naučimo i šta je *local bus*. U pitanju su obične ISA ploče, sa jednim ili dva specijalna ekspanziona slota. Oni veoma podsećaju na „obične” 16-bitne slotove ali su „produženi” dodatnim konektorima sa izvodima. Tako, na primer, mikroprocesor može direktno da pristupa VGA kartici, disk kontroleru i drugim periferama koji, prirodno, moraju biti prilagođeni odgovarajućoj ploči. Ubrzanje koje se na ovaj način postiže može da bude spektakularno, 10 i više puta [Preuzeto sa Sezama, autor Dragoslav Ristić (*dragoslav*)].

617 [HARD; PC]: Govorili smo o video karticama i o disk kontrolerima, dakle „ostali” su nam još mikroprocesori. Dosta zabune u poslednje vreme izazivaju ploče sa *Cyrix*-ovim mikroprocesorom 486DLX čija cena dovodi u veliko iskušenje da posegnete u džep. Ipak, uzdržite se od toga – *Cyrix 486DLX* nije kopija *Intel*-a 80486DX nego 386DX klon za internim kešom od jednog kilobajta. Ime je samo *Cyrix*-ov reklamni trik.

Treba znati da je *Cyrix 486DLC* pin kompatibilan sa *Intel*-ovim 80386, dok je sa softverske strane dosta blizak „pravom” 80486, premda mu fali nekoliko instrukcija (*burst mode*...) Koprocesor, na žalost, nije njegov integralni deo, a umesto 8 ima samo kilobajt internog keša. *Cyrix*-ova marketinška strategija je usmere-

na na one koji već imaju 386 ploče i koji žele da unaprede performanse prostom zamenom čipa. Za one koji se tek odlučuju za kupovinu... mnogo ćete bolje proći sa „pravim” 80486 [Preuzeto sa Sezama, autori Danko Jevtović (*danko*) i Daniel Joskovski (*djoskovski*)].

618 [P.J.PASCAL; PC]: Već smo pisali o BUG-u mikroprocesora 80386/80486 koji pri korišćenju instrukcije SHRD u 16-bitnom modu daju pogrešan rezultat ako se šiftuje za više od 16 bita. Na žalost, ova instrukcija se koristi u okviru izvršne biblioteke (*run time library*) novog *Borland Pascal*-a 7.0 kada se on izvršava na 80386 (ili boljem) mikroprocesoru. Za uklanjanje greške treba izvršiti dve male modifikacije u datoteci *\BP\RTL\SYS\LONG.ASM*. Pronađite i prepravite odgovarajući segment prema slici 3. Zatim izvršite .BAT fajl sa slike 4 kako biste kreirali nove .TPL datoteke [Preuzeto sa BIX-a, autor terjem].

619 [MRZ.NOVELL; PC]: Često se ukazuje potreba da se *Novell* mreža kontroliše sa nekog udaljenog računara, primenom modema. „Regularan” način da se to uradi je ROUTER koji se isporučuje uz *Novell Netware 2.2*. On omogućava povezivanje PC-ja preko modema sa bilo kojim serverom ili radnom stanicom. Na solo PC-ju treba, pomoću ROUTERGEN-a, generisati programe koji se učitavaju umesto IPX-a ili umesto NET\$OS-a. Najveći problem je brzina – ako se koristi modem i COM portovi, može se „izgurati” najviše 2400 bps, što je uglavnom nedovoljno sa iole komforan rad. Problem je, naime, u tome što ROUTER prvo prebaci program do odvojenog računara i tek onda ga izvršava, tako da obično LOGIN može da potraje par minuta obzirom da LOGIN.EXE ima 50 K. Korišćenjem posebnih *Novell*-ovih multiserijskih inteligentnih kartica (WNIM+) brzina se povećava na 19200 bps, ali su ove kartice, na žalost, kod nas jako retke a cena im prelazi par hiljada maraka.

Drugo rešenje je znatno jednostavnije i jeftinije, a opet u većini slučajeva može da „reši problem”. Na nekom od mrežnih računara treba postaviti modem i aktivirati HOST iz programa *PC Aterm Anywhere IV*. Posle toga sa bilo kog računara možemo pozvati taj čvor i pristupiti celokupnoj mreži kao što bismo joj pristupili sa kompjutera koji je primio poziv. I u ovom slučaju postiže se brzina 2400 bps, ali je sada ona dovoljna, jer se programi izvršavaju na računaru koji je primio poziv, tj. ne prenose se posredstvom modema [Preuzeto sa Sezama, autor Miodrag Zdravković (*mzdravkovic*)].

620 [COMM; PC]: Može li se na isti PC instalirati dva, tri, četiri ili više modema? Sve zavisi od tipa serijskog porta. Ukoliko se radi o serijskim portovima sa 8250 ili 16450 UART, instaliranje više od dva modema je gotovo nemoguće. Kada se radi o 16550A, njegov bafer od 16 bajta rasterećuje interapt sistem mikroprocesora, pa se može pokušati sa više modema. Nevolja je što 16550A serijske kartice nisu česte na PC računarima – ugrađuju se, međutim, na mnogim IBM PS/2 modelima. Svi noviji komunikacioni programi ih prepoznaju i koriste njihove dodatne mogućnosti.

Na PS/2 90 i 95 postoji serijski port sa DMA transferom, koji skoro da ne ometa procesor u radu, pa se (barem u teoriji) bez problema može raditi sa 4 ili čak 8 modema [Preuzeto sa Sezama, autor Miloš Prvulović (*prvul*)].

621 [KOMS; PC]: Pomoću *Turbo Debugger* mogu se testirati i veoma dugački programi, ali

Listing 1

```

data xx          dw      8405h
seedlow         equ     ...
seedhi          equ     ...
proc            near
mov             ax,ds:seedlow
mov             bx,ds:seedhi
mov             cx,ax
mul            cs:data_xx
shl            cx,1
shl            cx,1
shl            cx,1
add            ch,cl
add            dx,cx
add            dx,bx
shl            bx,1
shl            bx,1
add            dx,bx
add            dh,bl
mov            cl,5
shl            bx,cl
add            dh,bl
add            ax,1
adc            dx,0
mov            ds:seedlow,ax
mov            ds:seedhi,dx
retn
endp

random          endp

```

Kako Turbo Pascal generiše slučajne brojeve

Listing 2

```

-----*/
* filename - rand.c
* function(s)
* srand - initializes random number generator
* rand - random number generator
-----*/
/*
* C/C++ Run Time Library - Version 5.0
* Copyright (c) 1987, 1992 by Borland International
* All Rights Reserved.
*/

#include <stdlib.h>

#define MULTIPLIER 0x015a4e35L
#define INCREMENT 1

static long Seed = 1;

-----*/
Name          srand - initializes random number generator
Usage         void srand(unsigned seed);
Prototype in  stdlib.h
Description   see rand below
Return value  Nothing

-----*/
void srand(unsigned seed)
{
    Seed = seed;
}

-----*/

```

ne bez jednog „sitnog“ dodatka. U CONFIG.SYS treba ugraditi DEVICE=D:\SYS\TDH386.SYS -E3024. Ono -E3024 predstavlja veličinu *environment*-a i mora se navesti vrednost **veća** od stvarne. Ograničenje je i da se ne sme koristiti QEMM, 368MAX i ostali programi koji prebacuju 80386 u neki od *virtual* modova. *Turbo Debugger*, naime, kreira dva virtuelna XT-a i u jedan učitava TD.EXE, a u drugi 8086 program koji analiziramo.

Sve ovo, naravno, radi samo na 80386 (i boljim) računarima [Preuzeto sa Sezama, autor Miljan Jovanović (*mjova*)].

622 [TERM]: Ne patimo samo mi od terminoloških problema – u raznim publikacijama sa engleskog govornog područja naći ćete reči *disc* i *disk* – neki koriste jedan termin, neki koriste drugi a ima i onih koji u istom tekstu mešaju oba. Šta je „pravilnije“?

U prvim danima kompjuterske industrije *disc* je bio velika i masivna stvar koja se ugrađivala u još veći i masivniji dravj. Kada je IBM pronašao diskete (*diskette*) od 8 inča, pojavio se i termin *disk* (sa k) Docnije se zadržao i kada su se diskete smanjile na 5.25, 3.5 i manje inča. U američkoj varijanti se, dakle, koristi reč *disk* (svi, na primer, poznajemo program *Norton Disk Doctor*) ali se uređaj u koji se diskete umeću i dalje zove *disc drive* – nije baš logično.

U Britaniji se mnogo češće koristi termin *disc*, pre svega zbog giganta ICL koji uporno koristi baš taj izraz. BBC je tu nešto fleksibilniji, tako da njihovi školski računari (koje pravi *Acorn*) dopuštaju da se komanda kuca na oba načina – *DISC i *DISK su sinonimi. Kroz čitavu dokumentaciju koristi se, ipak, izraz *disc*.

Šta je, najzad, pravilnije? Kao i obično, sve

zavisi od varijante jezika koju koristite. Mi bismo, ipak, bili skloniji terminu *disk* – ako ništa drugo, više liči na našu reč! [Preuzeto sa Sezama, autor Dragan Petrović (*drakce*)].

623 [IGRE; PC]: Ako želite da nabavite još jedan život, dosta municije i druge lepote u igri *Commander Keen 5*, pritisnite zajedno tastere B, A i T [Preuzeto sa DECnet-a, autor Đorđe Herceg (UNSIM::IHERCEG)].

Priloge za ovu rubriku šalžite na adresu „Računari“ (za „Bajtove lične prirode“), Bulevar vojvode Mišića 17, Beograd ili preko Sezama u vidu lične poruke uredniku (*mail write dejanr*).

```

Name          rand - random number generator
Usage         int rand(void);

Related
functions     void srand(unsigned seed);

Prototype in  stdlib.h

Description   rand uses a multiplicative congruential random number
              generator with period 232 to return successive pseudo-
              random numbers in the range from 0 to 215 - 1.

              The generator is reinitialized by calling srand with an
              argument value of 1. It can be set to a new starting point by
              calling srand with a given seed number.

-----*/
int rand(void)
{
    Seed = MULTIPLIER * Seed + INCREMENT;
    return((int)(Seed >> 16) & 0x7fff);
}

```

Kako Turbo C generiše slučajne brojeve

Listing 3

```

LongShr: CMP     Test8086,2
          JB      @#1
          ; BUGFIX! SHRD AX,DX,CL fails if CL > 16!
          ; SHRD  AX,DX,CL
          ; SHR   DX,CL
          ; SHL  EDX,16 ; New code
          MOV   DX,AX ;
          SHR  EDX,CL ;
          MOV  AX,DX ;
          SHR  EDX,16 ; New code
          RETF

          .8086
          .....
LongShr: CMP     Test8086,2
          JB      @#1
          ; BUGFIX!
          ; SHLD  DX,AX,CL
          ; SHL  AX,CL
          ; SHL  EDX,16 ; New code
          MOV  DX,AX ;
          SHL  EDX,CL ;
          MOV  AX,DX ;
          SHR  EDX,16 ; New code
          RETF

          .8086
          .....

```

Ispravka бага u Borland Pasca-u 7.0

Listing 4

```

cd \bp\rtl
make
cd \bp\bin
copy *.tpl *.sav
copy ..\rtl\bin*.tpl *.

```

.BAT fajl za kreiranje novih .TPL datoteka



SISTEM ZA MODEMSKE
KOMUNIKACIJE

Hiljadu poruka u konferenciji SEZAM verovatno je rezultat teksta koji je objavljen u „Računarima 87” – najavljena su (premda ne još i realizovana) brojna poboljšanja Sezamovog korisničkog interfejsa, pa je bilo puno pohvala, pitanja, primedbi i sugestija. Dosta pažnje izazvala je i TV emisija „Codename Sezam” emitovana na NTV Studio B povodom Sezamovog trećeg rođendana – utisci su uglavnom bili veoma povoljni ma da je par korisnika imalo utisak da je emisija previše „lakonski” realizovana.

18

Priprema: Dejan Ristanović u saradnji sa moderatorima konferencija

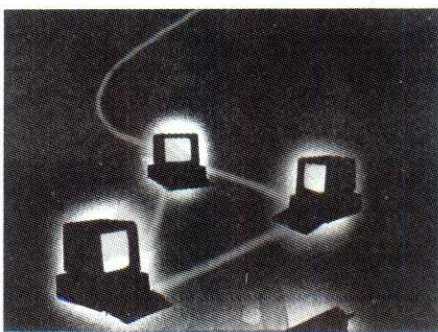
Moderatori: Dejan Balinda (CIVILIZACIJA), Nikola Bošković (PC.USER), Jovan Bulajić (FILES), Aleksandar Damjanović (PC.OS), Danko Jevtović (PC.HARD), Slobodan Kalezić (ORKA), Vladimir Kostić (miruje), Željko Krstić (VICEVI), Vladimir Krstonošić (miruje), Ranka Jovanović (RAČUNARI), Dalibor Lanik (ATARI.ST), Bojan Pavković (PC.UTIL), Pavle Peković (PC.PROG), Dejan Ristanović (NOVOSTI i KOMUNIKACIJE), Bojan Tepavčević (FORUM) i Zoran Životić (SEZAM)

Tokom novembra Sezam je „opslužio” 23,018 poziva, komunicirao sa korisnicima 5,743 časa, tj. preko 239 dana (obzirom da novembar ima 30 dan, u proseku je gotovo 8 nodova bilo neprekidno zauzeto) od čega je u chat-u provedeno 1065 sati ili 133 radna dana. Korisnici su sa Sezama download-ovali 1.2 gigabajta podataka dok je za isto vreme sistemu poslato 49 megabajta. Najviše poziva uputili su (ne računajući službena lica Sezama): Srđan Kusovac (*squsovac*), Aleksandra Petrović (*acka*) i Dušan Mihajlović (*zdule*), najviše vremena na vezi proveli su Vesna Denčić (*veca*), Željko Krstić (*zkrstic*) i Vesna Tulović (*vesna*) dok su najviše download-ovali Dražan Grbić (*dr.grba*), Slobodan Petrović (*feniks*) i Bora Živanović (*bor*). U javne konferencije pristiglo oko 5,750 poruka (preko 6.5 megabajta teksta) i 1.3 megabajta datoteka priključenih uz njih, 7,700 privatnih poruka (skoro tri megabajta privatne pošte) i 4,500 poruka u grupama (1.6 megabajta teksta ne računajući datoteke priključene uz te poruke).

U konferenciji RAČUNARI bilo je relativno mirno – tek nešto diskusije o novom broju „Računara”, njegovoj naslovnoj strani i ceni. Najviše interesovanja izazvala je nagradna igra, pa je bilo brojnih komentara na izvlačenje i iznenađenja što ni jedna nagrada nije došla u Beograd. Ipak, jedna od nagrada pripala je upravo pretplatniku Sezama, čije je korisničko ime *oki*.

Najviše poruka u konferenciji KOMUNIKACIJE stiglo je, kao i obično, u temu *modemi*. Diskutovalo se o (samo)gradnji transformatora za eksterne modeme (obično treba paralelno vezati dva trafoa za zvonce), zvanju modemom preko centrale uz prethodno okretanje „nule” da bi se dobio izlaz „u grad”, Intel Satisfaxion fax modem karticama (verovatno najbolji modeli na tržištu, ali koštaju 5-10 puta više od po specifikacijama sličnih modela drugih firmi) i mnogim drugim pitanjima. U okviru teme *kom.programi* više nego obično se diskutovalo o *Procomm-u*. Najpre je jedan korisnik poslao *Procomm 2.4* i u prvi mah izazvao veliko interesovanje. Docnije se, na žalost, pokazalo da je u pitanju stari *Procomm* (a ne *Procomm Plus*) iz 1986. godine. Bilo je reči i o programu koji računa broj utrošenih impulsa pri zvanju iz *Procomm-a*, o sekvenci koja aktivira eksterni DSZ/GSZ, chat modu *Telemate-a*, specifikacijama fosil drajvera i inicijalnim stringovima za *Remote Access 1.11*. U temama koje se bave elektronskom poštom dosta interesovanja izazvala je vest objavljena u „Politici” da će iz Jugoslavije uskoro ponovo moći da se šalje mail preko *Bitnet-a*. Pokazalo se, na žalost, da je vest u najmanju ruku preuranjena – Jugoslavija nije isključena iz članstva EARN-a, ali se pošta i dalje ne može slati ni primiti; kada će moći, videćemo. Diskutovali smo i o zvanju UBBG-a i BUEFxx računara iz sveta i o FON File Server-u koji izaziva sve veće interesovanje. Sa strane scene, tražili smo i našli brojeva BBS-ova u Silicijumskoj dolini i Češkoj, saznali email adresu novog američkog Predsednika (75300.3115@compuserve.com) i, kao svojevrsnu pikanteriju, pročitali informacije o hakerskom upadu na poznati Švajcarski BBS *Pegasus*.

Obzirom da se konferencija PC.PROG u novembru prepunila, programeri su dobili novo mesto za okupljanje – PC.PROG.2. U toj novoj konferenciji čitali smo o preslikavanju skupa datuma u skup celih brojeva. Lavinu poruka je izazvao problem korišćenja funkcija „findfirst”/„findnext” i izmene tako pronađenih datoteka. Videli smo kako se bezijk povezuje sa ostalim jezicima. Kao i mnogo puta do sada, najživlje je bilo među ljubiteljima paskala – poigravali su se slikama, tragalj za najboljim algoritmom za traženje zadate reči, rešavali veći problem kontrole miša, diskutovali o „OBJ fajlovima i Turbo Pascal-u, povezivali paskal sa drugim programskim jezicima, pravili komunikacioni



program, tražili načine da zaobiđu DOS-ove poruke o greškama i po ko zna koji put žalili zbog nekompatibilnosti unit-a Turbo Pascal-a 5.5 i 6.0. C programeri su demonstrirali lakoću konverzije između različitih tipova promenljivih, ali i koliko je lako zbuniti se u tome. I C ima svoju većitu temu, a to je manipulacija različitim grafičkim formatima. Novembar i nova PC.PROG.2 su doneli ljubiteljima C++ temu u kojoj su oni pisali o deklaracijama i vidljivosti promenljivih, nizovima proizvoljne veličine... U temi *clipper* pričalo se o linkerima, njihovim prednostima, manama, načinu korišćenja itd. Zanimljivo je bilo čitati i o pokušaju jednog korisnika da ukaže na probleme u vezi kompatibilnosti starog i novog *Clippera-a*, ali je, kao i par puta do sada, problem bio u korisniku, a ne u *Clipper-u*. Naravno, ni ovog meseca nije izostao pokušaj povezivanja C-a i *Clipper-a*.

Oktober je u konferenciji PC.USER prošao u znaku diskusije o tekst editorima, a naročito poredenju *QEdit-a* i *Multi Edit* – bilo je dosta argumenata i za jedan i za drugi. U temi *word.perfect* podsetili smo se zašto se gube naša slova pri izlasku iz *preview* moda – jedno od mogućih rešenja je domaći drajver FOG-GY *Gorana Bločić-a* (*gbiocic*). Dobili smo i makro koji vrši konverziju cirilice u latinicu i obrnuto. Tema *matematika* se bavila problemom rada paketa *Mathematica 2.0* na različitim mašinama – uz AMI BIOS program se uspešno pokreće ali sa DTK BIOS-om ima velikih problema kao i sa novijim verzijama *Phoenix BIOS-a*.

Ustanovili smo i da je verzija *Mathematice 2.0* za DOS višestruko (nekada i za faktor 5!) brža od *Windows* verzije. I u ostalim temama bilo je zanimljive diskusije – može li *AutoCAD 11* da radi pod *Windows-om*, kako se razmenjuju tekstovi između *Word-a* i *WordPerfect-a*, kakva je *Ventura for Windows...*

Glavna rasprava u konferenciji PC.UTIL vođena je u temi *arhiveri* – nastavili smo diskusiju o problemima ARJ-a pri kreiranju arhive na disketama (čak je i sam autor ARJ-a „priznao” da postoje problemi tog tipa, naročito ako se upis na flopi kešira) i predložili nekoliko načina da se problemi eliminišu; najlakše je da se arhiva kreira na hard disku, potom se prebaci na diskete i onda na samim disketama testira njena validnost – nije baš prebrzo ali je bar sigurno! Kašnjenje nove verzije PKZIP-a podstiče mnogo tvorce virusa da svoje viruse pakuju kao najnoviju verziju pomenutog programa. Tako je na Sezamu pominjana „najnovija” verzija tog programa pod oznakom 3.05. Obzirom da zvaničnih informacija o novoj verziji nije bilo, odmah se posumnjalo da je u pitanju prerađena kopija neke od ranijih verzija programa. Pitanje da li pomenuta verzija sadrži i virus ili ne, rešeno je dvanaestčasovnim testiranjem posle toga je pola postojećih datoteka bilo obrisano. Dakle čuvajte se „novih” verzija programa PKZIP. U temi *virusi* raspravljalo se o problemima detekcije virusa u programima koji virus uopšte ne sadrže. Lažnu uzbuđenu podigao je program VDEFEND iz paketa *PcTools*, a program koji u sebi navodno sadrži virus bio je LCOPY. Obzirom da je autor programa LCOPY (*dejanr*) na Sezamu, testirana

je originalna verzija programa koja ne sadrži nikakav virus, ali je VDEFEND i dalje uporno prijavljivao da program sadrži virus. Zaključak iz svega ovoga nije teško izvući: obratiti pažnju koji program koristite za otkrivanje virusa, jer može vam se desiti da „čistite” viruse kojih u stvari nema. Pored „većite” diskusije vlasnika 286 računara u temi *memory.mgr* o optimalnom korišćenju memorije, raspravljalo se još i o brzina i načinima testiranja programa za keširanje diska, o tome kako izbeći uvodne ekrane programa *HyperDisk*, ograničenjima programa *Norton Commander* pri radu sa direktorijima koji sadrže veliki broj datoteka (preko 500), kao i pri radu pod mrežnim operativnim sistemom i o programima za kovertovanje grafičkih formata.

Tema meseca u konferenciji ATARI.ST bilo je programiranje. Posle najave da uskoro stižu PURE C (nova verzija *Borland Turbo C-a*) i ACS (*Application Construction Set*), u ovoj temi se najviše pričalo o *Resource* editorima. Slična konverzacija se vodila i u okviru teme *emulatori*, gde se najviše pričalo o programu *Prototyper* za *Macintosh* koji je jako sličan *ST-ovom ACS-u* i automatski generiše C kod za osnovni *shell*. Tema *grafika* se bavila priključenjem dva monitora za rad sa *Calamus-om* a vest da je sada već legendarni program *Spectrum 512* koji u isto vreme prikazuje 512 boja (iako je *ST-ov* hardver ograničen na 16) postao *Public Domain* veoma je obradovala atariste – program se sada može preuzeti i sa Sezama. Ljubitelji kompjuterskih igara nisu stigli da se „odmore” od *OXYD-a 2*, a već je stigla vest da je *ASH* izbacio *OXYD 3*, koji radi i na *TT-u* i na novom *atari-*

SPECIJALNE USLUGE

PODSISTEM TANJUG

Podsistem „Tanjug” omogućava pristup profesionalnim servisima novinske agencije TANJUG. Radi se o ažurnim političkim, ekonomskim, društvenim, naučnim, sportskim i drugim vestima iz zemlje i sveta. Servisi neprekidno pristižu na Sezam preko stalno iznajmljene linije, a registrovani korisnici ih prema potrebi mogu čitati ili prenositi na svoje računare radi dalje obrade. Pristup ovom pod sistemu ostvaruje se iz menija (MENU TANJUG) ili komandnog moda (naredba NEWS).

Pristup pod sistemu „Tanjug” je opciona usluga Sezama i plaća se posebno. Za detalje oko preplate i uslova korišćenja obratite se Novinskoj agenciji Tanjug. Kontakt: Milena Sekulić, tel: 625-722.

PODSISTEM BETA

Podsistem „Beta” obezbeđuje praćenje domaćih časopisa u elektronskoj formi. Časopisi pristižu na Sezam pre ili u trenutku izlaska na kioske i docnije se prema potrebama korisnika mogu pregledati, čitati ili prenositi na sopstvene računare. Pristup ovom pod sistemu ostvaruje se iz menija (MENU BETA) ili komandnog moda (naredba BETA). Trenutno se na ovaj način distribuira nedeljnik „Vreme”, a pristup je u probnom periodu slobodan.

Pristup pod sistemu „BetaPress” je opciona usluga Sezama i plaća se posebno. Za detalje oko preplate i uslova korišćenja obratite se Marketingu lista „Vreme”. Kontakt: Vojislav Milošević, tel: 646-070.

PODSISTEM BERZA

„Berza” je „Banka poslovnih podataka privrede Srbije” – radi se o osnovnim podacima o radnim organizacijama u Srbiji i njihovoj delatnosti, zajedno sa nazivima artikala koje pojedine firme traže, odnosno nude. Različitim metodama pretraživanja mogu se izdvojiti firme koje traže ili nude ono što vam je potrebno, a zatim pronaći ostali podaci o njima, koji uključuju adresu, telefon, telex i faksi. Imate mogućnost da na sličan način date podatke o ponudi i tražnji vaše firme, koje će drugi korisnici docnije pregledati. Pristup ovom pod sistemu ostvaruje se iz menija (MENU BERZA) ili komandnog moda (naredba STOCK).

Pristup pod sistemu BERZA je opciona usluga Sezama i plaća se posebno. Za detalje oko preplate i uslova korišćenja obratite se firmi Electro Engineering, tel: 609-453.

PRESTUP S PREDUMIŠLJAJEM

Dejan odgovara: Ja sam, za razliku od gospodina Tešića, koji je očigledno zgrožen mojim delom, i te kako uživao u njegovom pismu (članku?). Retko kad se sreće tako lucidno objašnjenje nekih aspekta objektnog programiranja (krećući od teoretske podloge, dakako). No, uz to je g. Tešić naveo i niz mesta u mom tekstu za koja tvrdi da su netačna i nedorečena. Da sada za kaznu ne bih klečao na kukuruzu, dopustite mi da objasnim o čemu je reč:

Umesto da bacim na čitaoca sve što C++ ima da ponudi, što je pristup koji bi nesumnjivo zahtevao mnogo više prostora od jednog novinskog članka (nekoliko knjiga bi bilo prikladnije), potrudio sam se da iz tog obilja izdvojim one informacije koje bi čitaocu dale nekakvu osnovu u korišćenju objekata, a i da ga „zagolicam“, tako da po čitanju teksta pokuša da do kraja upozna C++. Odlična kritika na Sezamu, kao i to što je moj tekst završio kao #1 na tamošnjoj top-listi, samo su me učvrstili u uverenju da je ovakav pristup ispravan. Da ne bude zabune, niko se tamo nije javio i rekao „Hej, tekst je super, eto, postao sam C++ programer“, već su rekli da im je tekst razjasnio mnoge stvari koje im do sada nisu bile jasne, te da će se sada ozbiljno pozabaviti C++-om. Čini se ispravno...

ZATVORITE ME S KNUTOM

Što se tiče „gradiva“ koje sam za taj tekst izdvojio, ono se, paradoksalno, zasniva na Turbo Paskalu. Momci u Borlandu imaju zgodnu naviku da svoj paskal obogate sa 70 posto snage „onog drugog jezika“, zadržavajući pri tom jednostavnost. Tako je TP u verziji 4.0 dobio unite „pozajmljene“ iz Module, u verziji 5.5 su stigli i objekti veoma slični onome kako ih C++ predstavlja, a tek je verzija 6.0 dobila i **private** polja, implementirana, doduše, drugačije nego što to C++ radi. Kako ne znam nikoga ko ima problem sa shvatanjem objekata onako kako ih Turbo Paskal nudi, i ja sam se u svom tekstu „usudio“ da predstavim samo klase, virtualne funkcije, konstruktore/destruktore, i, naravno, preklapanje operatora kome nisam mogao da odolim. Za **friend** klase/funkcije i **protected** polja klasa tu jednostavno nije bilo mesta: iako su ove „opcije“ veoma korisne, ljudi friški u objektnoj metodologiji pisanja programa najčešće ih zloupotrebljavaju. Čini mi se da bi bilo pametnije „sakriti“ ih od novih programera, nego im dati nešto što u većini slučajeva ispadne kao tempirana bomba. Nije baš „information hiding“, ali tu je negde.

„Omaške i nekorektnosti“ iz mog teksta su tu (osim za „klasu kao sintaksnu jedinicu“) namerno. Smatrao sam da je mnogo bolje da kažem da klasa „nije ništa drugo nego malo poboljšana struktura“, nego da je, citiram g. Tešića, „uz objekat, osnovni koncept objektnog pristupa teoriji sistema i kao takva je preuzeta i koristi se i u objektnim jezicima.“ Ajde! Ne vidim kako bi prosečan C programer, koji nije ni prošao blizu teorije sistema mogao da shvati ovakvu rečenicu. Moj pristup je, iako možda netačan, empirijski gledano, uspeo da objasni šta je to klasa. Nekako se ne osećam loše zbog toga... Čak i Knut u predgovoru knjige o TeX-u kaže da su „neke tvrdnje u knjizi materijalno pogrešne, međutim da to predstavlja način da korisnik shvati neke složene stvari na određenom nivou da bi, kada njegovo znanje napreduje, mogao da ih shvati u potpunosti!“ Ako budemo robijali zbog naših gnusnih zlodela, nadam se da ćemo deliti ćeliju.



(Nagrada igra: Priznao sam da sam mislio jedno, a pisao drugo. Ako uspete da dokažete da sam u isto vreme radio treće, moći ćete da se okušate i u novom kolu: „Ko je zapravo poznati političar S.M.?”)

SILOVANJE JEZIKA

Pa, to je to. Bez ikakve želje da dalje raspijem ovakav način diskusije, u ostatku mog odgovora slede dve tačke u kojima moram da iznesem svoje neslaganje sa g. Tešićem:

U jedanestoj tački on iznosi svoje (blago rečeno) neslaganje sa mojom konstantacijom da objekte ne treba gurati svuda. Da li je to tačno ili ne, to još niko ne zna, ali većina svetskih autoriteta na polju objektnog dizajna i klasifikacije (Booch, na primer) tvrdi da ipak *nije objekat*. Siđemo li sa Olimpa teoretske diskusije, stvari postaju još gore: pokušajte samo da preusmerite neki interapt na **member** funkciju (pravu, ne statičku) neke klase. Ili se upitajte da li na funkcijama najnižeg nivoa za, recimo, crtanje po ekranu, u kojima se svaka instrukcija broji, možda treba uštedeti na skrivenom **this** parametru. Ili se upitajte zašto i sam Stroustrup u svojim programima koristi **friend** funkcije (!) da bi prevazišao neka inherentna ograničenja objektnih jezika? Insistiranje na nemešanju objektno i strukturirane metodologije razvoja programa ne samo da je teoretski nepoželjno, nego je i praktično nemoguće! Uvek ću dati prednost objektnoj metodologiji kad mogu, ali insistiranje na njoj prečesto se pretvara u silovanje samog jezika.

Usput, iako i meni ideja da neko od C++-a uzme samo najpotrebnije a da piše zapravo C programe zvuči loše, ne mogu se nikako složiti da je to gore nego da piše „suve“ C programe. Nije teško zamisliti nekoliko klasa (string, vektor i matrica padaju na um) koje bi svaki C programer mogao i te kako dobro da iskoristi u svojim programima, bez ikakvog poznavanja objektnog programiranja. Velika je šteta ne iskoristiti sve što C++ nudi, ali, koliko god parče da odgrizete, nikada niste u minusu.

NA LEDIMA C-a

U svom zaključku g. Tešić tvrdi da nije C++ u osnovi još uvek C. Moram da se i složim i ne složim (?) sa ovakvom konstantacijom. Reći, sa jedne strane, da je C++ samo malo bolji C je kao kada bismo rekli da je Andrija Artuković „čovek koji je voleo da radi sa ljudima“. Prednosti koje C++ donosi u odnosu na C su ogromne, radi se, u stvari, o kvalitativno novom jeziku. Na žalost, kada kažem da je C++ u osnovi još uvek C, to je zato što on ipak pati od mnogih ograničenja koja ima i sam C. Na primer, C++ ne dozvoljava da napravite operator ** za stepenovanje, a i ^ koje je takođe dobar kandidat ima problem sa prioritetima (^ slabije vezuje i od operatora +, pa tako izrazi ne bi bili „ono pravo“). Ili, recimo, klasa **LongLongInt** ne može direktno od parsera da dobije broj

```
12435473127123572654724,
```

već se dodela mora vršiti preko stringova:

```
a = LongLongInt ("12435473127123572654724");
```

Petljanje sa parserom bi predstavljalo veliku muku, tako da je C++, uz izuzetak nekoliko novih operatora, i ovde ipak napravljen na ledima C-a. To ne treba gurnuti ispod tepiha, već se treba u odnosu na to pravilno postaviti pri kreiranju programa.

(Usput, moja konstantacija da je C++ još uvek C je podrazumevala da se mnoge stvari u C++-u rade kao i u C-u. Nisam želeo da objašnjavam u tekstu koncepte klase i polimorfizma, pa da me posle presretne neki čitalac sa pitanjem: „Dobro, super su te klase, ali kako se u C++-u sabiraju brojevi?“ Verujte mi, to se dešava.)

Da zapakujemo stvar, g. Tešić i ja se (ma koliko to čudno zvuči) slažemo da je C++ odličan izbor. Kao što to obično biva, g. Tešić će iz ove diskusije izaći ubeđen da sam ja potpuni folirant i dileja, ja ću misliti da je njega žena pričepila da mi napiše repliku zbog mojih šovinističkih primera, a vi ćete, ukoliko ste ovo preživeli, verovatno nastaviti da koristite ovaj prelepi jezik. Živi bili pa videli.



Dure Jakšića 1a, 21000 Novi Sad, tel / fax: +38-21-26-251
26-275

PREDSTAVLJA SuitCASE

Ako želite:

- Automatizovan razvoj aplikacije;
- Bavljenje projektovanjem a ne programiranjem;
- Aplikacije koje automatski rade i u mreži; (višekorisnički ambijent);
- Aplikacije koje su transakciono orijentisane (rollback i sl.);

Ako NE želite:

- Da ste u začaranom krugu editovanje-kompajliranje-testiranje-debagiranje;
- Da ste rob a ne kreator Vaših aplikacija jer se najveći deo vremena bavite održavanjem aplikacija;
- Da Vam projektna dokumentacija leži u glavama projektanata;

APLIKATIVNI SOFTVER

- preko DVADESET kompletnih programskih paketa
- svi programi pisani u SuitCASE-u
- mogućnost vezane kupovine (aplikacija + SuitCASE)
- SAMOSTALNA dorada software-a (obj + lib + SuitCASE)
- dokumentacija i uputstva obezbeđeni

NOVELL mrežna oprema

- NETWARE 2.2, 3.11 i NETWARE LITE 1.1
- X.25 GATEWAY
- povezivanje udaljenih radnih stanica
- konsalting, servis

Virtual Library of Faculty of Mathematics - University of Belgrade

elibrary.matf.bg.ac.rs



- ultra brze ETHERNET kartice COAX (256/512 Kb RAMa, 80186 procesor)
- POCKET ETHERNET adapteri
- povezivanje sa HOST sistemima
- CDDI / FDDI (100 Mb/s)

- transiveri
- ripiteri
- bridževi
- pocket 3270 adapteri
- rešenja za olakšano kabliranje

NOVELL®

SYSTEM INTEGRATOR

BBS za podršku korisnicima SuitCASE i NOVELL®: 021 / 26 - 275

Prigrama: TIPOGRAFIKA

ComTrad

the **ART** of
computer making

ComTrad 386/40-105

Procesor 386/40MHz, 4MB RAM

Floppy uređaji 1.2MB i 1.44MB

Tvrđi disk 105MB IDE

Dva serijska i jedan paralelni port

SVGA grafički adapter 1024x768/1MB

SVGA kolor monitor

Mini tower kućište, ASCII tastatura, miš

2,795.-

**Te samo ComTrad može: poklon uz 386DX
i 486DX računare - ručni skener "A4 Scan"!**

ComTrad YU • Genex apartmani, V. Popovića 6, Beograd

Telefon: (011) 222-41-51, 222-26-51 • Fax: (011) 222-41-39

Radno vreme: 9-17, ponedjeljak-petak

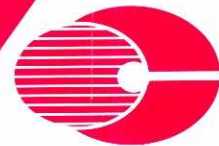
SPRINT
Novi Sad
Novosadskog
sajma 35
Tel. (021)
623-717
Fax (021)
623-901

ComTrad Shop
Beograd
Cvijičeva 104
Tel. (011)
752-663
Fax (011)
752-663

INFOTRADE
Pristina
Hekurudha 7
Tel. (038) 25-830
Fax (038) 25-822

COMTRAD
KRAGUJEVAC
Kragujevac
Kragujevačkog
oktobra 102
Tel. (034) 60-336
Fax (034) 67-117

MANIR
Ruma
Grobljanska 2
Tel. (022) 421-265
Fax (022) 421-265



ComTrad
COMPUTERS