

izdaje BIGZ

izlazi jedanput mesečno

# računari 29

Specijalno izdanje časopisa „Galaksija“

avgust 1987.

Cena 500 din.

*katalog ms dos softvera*

## 77 pc programa



novi  
softverski projekat

### hakeri vole devpak

periferijska oprema

### hercules u boji

## razbarušeni sprajtovi

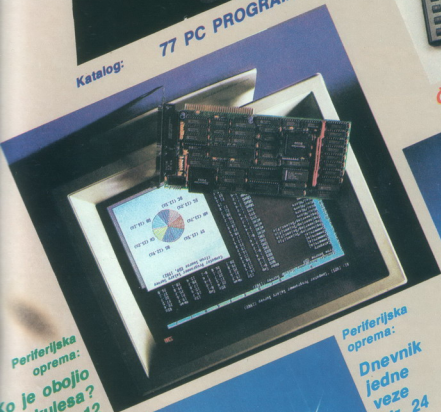
**U  
ovom  
broju**

Microsoft QuickBASIC 2.0

Katalog: **77 PC PROGRAMA str. 29**



**Četvrta generacija  
džepnih računara str. 8**

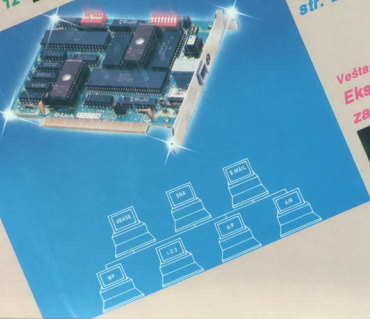


**Periferijska  
oprema:  
Ko je obojio  
Herkulesa?  
str. 12**

**Periferijska  
oprema:  
Dnevnik  
jedne  
veze  
str. 24**



**Računari u izlugu:  
„Komodor“  
u novom ruhu str. 9**



**Veštačka inteligencija:  
Ekspertni sistemi  
za početnike str. 19**





**Izdaje**  
Beogradski izdavačko-grafički  
zavod  
11000 Beograd  
Bulevar vojvode Mišića 17

**Telefoni**  
650-161 (redakcija)  
653-748  
650-528 (prodaja)  
651-793 (propaganda)

**Generelni direktor**  
Dobrosav Petrović

**V.d. direktor sektora izdavačko-  
novinska delatnost**  
Antun Martić

**V.d. glavni i odgovorni urednik**  
Gavrilo Vučković

**Odgovorni urednik**  
Jova Regasek

**Likovno-grafičko uređenje**  
Mirko Popov

**Redakcija časopisa „Galaksija“**  
Tanasije Gavranović, urednik  
Esad Jakupović, v.d. zamenik glav-  
nog i odgovornog urednika  
Aleksandar Milinković, urednik  
Jova Regasek, odgovorni urednik  
Zorka Simović, sekretar redakcije  
Srđan Stojančević, novinar  
Gavrilo Vučković, v.d. glavni i od-  
govorni urednik

**Stručni saradnici**  
Radomir A. Mihajlović, Dejan Rista-  
nović, Dušan Slavić, Nevenka Spa-  
lević, Anđelko Zgorelec

**Spoljna redakcija**  
Branko Đaković, Dejan Ristanović,  
Jelena Rupnik, Jovan Skuljan, prof.  
dr Dušan Slavić, Nevenka Spalević,  
Vlada Stojiljković, Zoran Životić

**Stalni saradnici**  
Nada Aleksić, Ninoslav Čabrić,  
Branko Đaković, Voja Gašić, Bran-  
ko Hebrang, Željko Jurić, Radomir  
A. Mihajlović, Zvonimir Makovec,  
Blazimir Mišić, Dejan Muhamedagi-  
ć, Ivan Nador, Zoran Obradović,  
Miodrag Potkonjak, Dejan Ristanović,  
Jelena Rupnik, Dušan Slavić,  
Jovan Skuljan, Nevenka Spalević,  
Zvonimir Vistrička, Žarko Vukosa-  
vljević, Anđelko Zgorelec, Zoran  
Životić

**Izdavački savet „Galaksije“**  
Dr Rudi Debijadi, prof. dr Branislav  
Dimitrijević, (predsednik), Radovan  
Dražković, Tanasije Gavranović, Ži-  
vorad Glišić, Esad Jakupović, Veliz-  
ar Maslač, Nikola Pajić, Željko  
Perunović, prof. dr Momočilo Ristić,  
Vlada Ristić, dr inž. Milorad Teofilo-  
vić, Vidoklo Veličković, Velimir  
Vasović, Milivoje Vuković

**Štampa**  
Beogradski izdavačko-grafički  
zavod  
11000 Beograd  
Bulevar vojvode Mišića 17

**Adresa redakcije**  
11000 Beograd  
Bulevar vojvode Mišića 17/III

**Rukopisi se ne vraćaju**  
**PRETPLATA ZA RAČUNARE**  
za Jugoslaviju  
RO BIGZ 60802-603-23264  
za jednu godinu — 6.000  
za pola godine — 3.000  
za inostranstvo:  
RO BIGZ 60811-620-16-101-820701-  
999-03377

22 UŠ, 39 DM, 13 lsg, 33 Švr,  
136 Švr., 131 Ftr. ili 12.000 din.  
Na osnovu mišljenja Republičkog  
sekretarijata za kulturu broj 413-  
77/72-03 i „Službenog glasnika“  
broj 26/72, ovo izdanje oslobođeno  
je poreza na promet.

## Sadržaj

- 4 Šta ima novo  
8 Računari u izlogu  
„Komodor“ u novom ruhu  
10 Računari u izlogu

### Plavi patuljak

- 12 Periferijska oprema

### Ko je obojlo Herkulesa?

- 14 Programeri govore

### Softver četvrte vrste

- 16 Peek & poke show  
18 Veštačka inteligencija

### Ekspertni sistemi za početnike

- 20 Mikroprocesori

### Matematičar gvozdene kova

- 22 Dejanove pitalice  
24 Interfejsi

### Dnevnik jedne veze

- 26 Svakodnevni algoritmi

### Ni manje ni više

- 29 Katalog PC softvera

### 77 PC programa

- 41 Mali oglasi  
42 Algoritmi

### Kraj Artur ne zna bezjki

- 44 Loto na računaru  
46 Tehnike programiranja/„amstrad“

### Žmurke sa Z80

- 48 Klub Z80  
50 Biblioteka programa

### Hakeri vole Devpak

- 53 Komercijalni softver/atari st“

### Emulacija ili imitacija

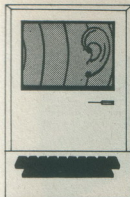
- 54 Matematički softver

### Metodi matrica

- 56 Put u središte ROM-a

### Makro asembleri

- 60 Razbarušeni sprajtovi



Šta ima novo

## Super memorija

U oblasti DRAM-a (Dynamic Random Acces Memory) ili dinamičke memorije sa direktnim pristupom, vodi japanska firma NTT (National Telegraph and Telephone).

16 Mbit/s DRAM zaista zasluđuje epitete kao što su „ultra“, „super“ i sl. Debljina linija u čipu je 0.7 mikromelara. Kondenzatori koji služe kao memorijski elementi su dimenzija manjih od 5 kvadratnih mikromelara. Vreme pristupa podacima je četiri puta manje od standardnog vremena pristupa. Naučnici koji su razvijali ovaj čip morali su, između ostalog, da „ratuju“ i sa alfa česticama, koje su izazivale slučajne defekte pojedinih ćelija, tako da je bilo potrebno razviti potpuno novi sistem kontrole kvaliteta za ovaj čip.

Napajanje je samo 3.3 volta a dimenzije 8.9x16.5x16.6 mm. Čip sadrži 40 miliona komponenti. (I. N.)

## Velika desetorica

Prošla, 1986. godina je bila, izgleda, prelomna i za japansku i za američku industriju čipova.

Svetsko tržište, koje je u 1986. bilo „teško“ 31 milijardu dolara, zabeležilo je rast od čitavih 25 % u odnosu na prethodnu godinu. Lavovski deo kolača uzele su japanske firme sa 14.2 milijarde dolara, Amerikancima je pripalo 10.2 milijarde, a

presotali deo — „ostatku sveta“. Porast proizvodnje u odnosu na prethodnu godinu je za Japan bilo rekordnih 40%, za evropske firme 15 % dok u SAD samo 6.4 %.

Velika desetorica su u 1986. zauzela sledeća mesta:

1. NEC (Nippon electric company) — Japan — 2.6 MLRD \$
2. Hitachi — Japan — 2.3 MLRD \$
3. Toshiba — Japan — 2.2 MLRD \$
4. Motorola — SAD — 2 MLRD \$

5. Texas Instruments — SAD
6. Philips-Signetics — Holandija
7. Fujitsu — Japan
8. Matsushita — Japan
9. Mitubishi — Japan
10. Intel — SAD

Najveći porast proizvodnje zabeležila je Toshiba od 54 % u odnosu na 1985, dok je najbolniji pad bio za Motorolu, koja je sa drugog pala na četvrto mesto. (I. N.)

## Napravi na Zapadu

Kada vidite reklamni slogan — „napravi bolje, napravi da traje, napravi jeftinije, napravi da bude kompatibilno“ — možete pomisliti da se radi o nekoj japanskoj ili tajvanskoj firmi. Američki proizvođač Western Computer je stekao priličnu reputaciju koristeći se ovim principima, ali je dodao i svoj najjači adut koji, iz razumljivih razloga, može da ima dobar propagandni efekat — „napravi na Zapadu“. Njihov najnoviji proizvod koji je posledica sopstvenog razvoja, sasvim je skladu sa dosadašnjim proizvodnom praksom i predstavlja nastavak serije kvalitetnih računara zasnovanih na IBM PC filozofiji.

Ne treba mnogo pogodati na kom mikroprocesoru je baziran ADVANTAGE 386. Intel 80386 na 16MHz srce je sistema koji na istoj ploči uključuje 512KB memoriju, 8 slotova za proširenja sa različitim adresnim arhitekturama (od 8 do 32 bita), RS232 i paralelni interfejs, baterijski napajan časovnik i podnožje za 80387 koprocesor (ili 80287 sa adapterom). Računar se standardno isporučuje sa flopijem 5 1/4 od 1, 2 MB i kombinovanim flopi/tvrđi disk kontrolerom, ispravljačem od 200W i tastaturom u IBM Enhanced Keyboard stilu (102 tastera). Pored klasičnog oblika, kutija se izrađuje i u TOWER stilu sa dodatnim prostorom za veći broj tvrdih diskova. O softverskoj kompatibilnosti se brine PHOENIX BIOS.

Firma nudi i veliki izbor proširenja. Pored standardnih video adaptera sa poboljšanim karakteristikama (delux EGA sa rezolucijom 752x410 tačaka) i monitora visoke rezolucije, mogu se nabaviti i hard diskovi kapaciteta 20—280 MB, memorijska proširenja do 15MB kao i poseban adapter kojim sistem postaje višekorisnički orijentisan sa mogućnošću priključenja do 32 ryčunara.

Za detaljnije informacije možete se obratiti na adresu: SH COMPUTER SYSTEMS, D7750 KONSTANZ, IM LOH 12, POSTFACH 5644.

4 računari 29 • avgust 1987.



## Veliki plavi ide dalje

IBM ne gubi vreme. Za eksperimentalni računarski sistem koji raspoznaje reči razvijen je rečnik od 20000 reči. IBM kaže da su u rečnik uvrštene reči koje se najčešće koriste u poslu.

Računar je predviđen da bude dimenzija stonog računara, a njegov korisnik ga „obučava“, tj. upozna je sa specifičnostima svog govora na taj način što mu pročita, reč po reč, tipski tekst u mikrofon. Ova „obuka“ traje oko 20 minuta.

Iako se izgovorena reč „gotovo trenutno“ ispisuje na ekranu, proces prepoznavanja reči se odvija u dve faze.

U prvoj fazi računara bira reči koje se „kandiduju“ da budu prepoznate, tj. liče na reči koja će na kraju biti ispisana.

U drugoj fazi ove reči se porede sa bazom podataka od 25 miliona reči, tj. tekstova iz IBM-ove poslovne korespondencije. „Kandidovana“ reč se poređi u kontekstu sa dve prethodne, tj. da li logično zvuči u tekstu gde je ona treća reč. Na taj način ovaj sistem razlikuje engleske reči „too“ od reči „two“ ili reč „know“ od reči „no“, koje se inače na engleskom veoma slično izgovaraju.

Interpunkcija se unosi verbalno.

Hardver je baziran na dva veoma brza podsistema, od kojih svaki sadrži specijalni čip (razvijen u IBM-u) za digitalnu obradu signala. (I. N.)

## Brzinski prvak sveta

Rekordi koji se postiču u elektronskoj industriji služe, izgleda, samo tome da bi već sutra bili oboreni.

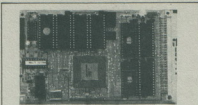
Trenutni prvak sveta u brzini u klasi digitalnih integrisanih kola je čip na bazi galijum-arsenida (GaAs) čiji je klok 18 gigaherza ili, laički rečeno, 18 milijardi perioda u sekundi. Proizvođač „malopoznat“ firma HJuz (Huges Aircraft Company). Funkcija ovog čipa je da radi kao brojač frekvencije u režimu deli-sa-dva. Prethodnik ovog čipa je radio na 13 gigaherza, ali na kriogenim temperaturama, dok ovaj čip radi na sobnoj temperaturi. Najbrža, komercijalna, silicijumska kola sada rade na učestalostima ne većim od 2 gigaherza.

Primena se vidi u superračunarima, fiber optičkim komunikacionim vezama, satelitskim komunikacijama i najmodernijim radarima. (I. N.)

## Link za „Turbo paskal“

Turbo ASM je novi i ne mnogo reklamirani Borelandov proizvod; uz njegovu pomoć možete da povežete programe pisane na čuvenom „Turbo paskalu“ sa assembler-skim rutinama. Nešto je slično, naravno, bilo moguće i do sada, ali je assembliranje trebalo obaviti unapred i nekako prepisati ASCII oblik objektnog koda u datoteku sa paskal programom. „Turbo ASM“ omogućava da u paskal program uključite COM datoteku, ili da za trenutak napustite kompajliranje i assemblerate tekst izvornog programa.

Cena „Turbo assemblera“ je 100 funti, a bliža obaveštenja možete da dobijete preko američkog telefona (415) 366 2062.



## Fort u čipu

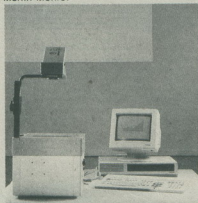
Novi Novixov mikroprocesor koji nosi oznaku NC4000P sadrži mašinski kod programskog jezika Fort. U čip je ugrađeno 40 fortovih instrukcija, koje su sjedinjene u jednu homogenu celjinu i predstavljaju fort kompajler. Čip radi na frekvenciji od 4 Mhz, a ugrađuje se u posebne fort-sisteme za IBM PC kompatibilne kompjutere.

U Novixu najavljuju nove procesore NC5000 i NC6000, koji će takođe predstaviti kompajlere viših programskih jezika. (I. M.)

## Digitajper za 100 funti

Džoj Šarp (Joy Sharp) i njegova kompanija Rombo (britanski telefon (506 39046 — kompanija je negde u Škotskoj) za svega 100 funti prodaju dvosmerni digitajper koji se povezuje sa Amstradovim kompjuterima iz serije PCW.

Šta omogućava ovakva dvosmerna komunikacija? Slike sa ekrana možete da prenosite na video traku, ali i slike sa video trake možete da digitalizujete, prenesete na ekran i, po potrebi, snimite na disk ili štampate. Na digitalizovanoj slici je, naravno, sam pronalazač, gospodin Šarp; prošla su vremena kada je digitalizovana samo Merlin Monrol



## Projekcije iz kompjutera

Projektor LCD omogućuje da se podaci iz kompjutera direktno preko ekrana projektuju na grafoskop, odnosno iz, i to u proizvoljnim dimenzijama. Nova LCD tehnologija radi s grafičkom rezolucijom od 640x200 tačkica i pojačanim kontrastom.

Grafoskop i kompjuter su povezani serijskim interfejsom, koji u sebi sadrži nešto savršeniji video interfejs, koji posebnom tehnikom kontrasta povećava čitljivost teksta. (J. M.)

## Legalna ili ilegalna magija

Firma Data Pacific (609 E Speer Blvd, Denver, Colorado 80203) je počela da reklamira novu Magic Sac, karticu koja omogućava Atarjevima računarima iz serije ST da izvršavaju programe pisane za „mekintoš“. Obućana je stoprocentna kompatibilnost, iako neke aplikacije mogu da budu sporije. Prvi se, naravno, pobunio Apple: da bi Magic Sac funkcionisao, treba ga dopuniti kopijom „mekintoševih“ ROM-ova. No, firma Data Pacific se i tu snala — ovi ROM-ovi se ne isporučuju uz karticu, što znači da kupci treba sami da se snadu. A Apple dobro zna kako će se snai!

Kad budete pisali u Denver, Kolorado, pozdravite Aleksis. Sigurno je umešana u ovaj marifetluk!

## Tandi emulira „epi II“

Dani u kojima su se Tandy i Apple borili za prevlast na računarskom tržištu pripadaju prošlosti — iako se obe kompanije sasvim solidno drže, IBM im je pomračio slavu. Teško je, međutim, zamisliti različite poslovne strategije: dok Tandy pravi PC i AT klonove, Apple ne odustaje od sopstvenog „mekintoša“ i stare „jabuke II“.

Nedavno je Tandy počeo da prodaje Trackstar 128, hardverski dodatak uz pomoć koga „tandi 1000“ može da izvršava (skoro) sve programe pisane za „epi II“. Trackstar 128 se sastoji od mikroprocesora 65C02, 128 K RAM-a, 16 K ROM-a, video interfejsa i „epi II“ kompatibilnog porta za proširenje — pravi računar u malom. Komunikacija sa štampačem, diskom i serijskim interfejsom je preuzeo sam „tandi 1000“.

Tandy tvrdi da ispitivanja nezavisnih stručnjaka pokazuju da preko 90% od fantastične „eplove“ biblioteke programa radi na Trackstar-u 128; probleme prave samo neke zaštićene igre. Jedina je nevolja cena — Trackstar 128 košta 400 funti, a ni originalni „epi II“ nije mnogo skuplji!

Američka adresa na koju možete da pišete za dalja obaveštenja je Tandy Corp, 1800 One Tandy Center, Fort Worth, TX 76102

## Novi IBM RT

Paralelno sa serijom PS/2, IBM je lansirao tri nova modela RT personalnih računara zasnovanih na RISC (Reduced Instruction Set Computer) procesorima. IBM RT 6151 Model 115 je, da počnemo od najodličnijeg, smešten u kutiju personalnog računara, dok modeli 125 i B25 predstavljaju čitave ormare koji stoje na podu. U centru svakog od njih je 32-bitni RISC procesor sa jedinicom za upravljanje memorijom (MMU), aritmetičkim koprocesorom i 4 megabajta brzog CMOS RAM-a. U blizini je i novi ESDI disk kontroler koji podržava tri interna hard diska. Smatra se da ovi modeli unapređuju performanse ranijih IBM-ovih RISC mašina za faktor 2 ili 3 u čemu pomaže i nova verzija operativnog sistema AIX.

Cene, na žalost, nisu baš niske: \$10.600 za model 115, \$16.100 za model 125 i \$17.600 za model B25. Razne IBM-ove adrese smo više puta objavljivali pa da objavimo i ovu: IBM Corp. (Information System Group), 900 King St., Rye Brook, NY 10573.

Šta ima novo

## Još jedan „komodor“

Komodor je nedavno najavio novu verziju popularnog C128 koja je dobila sufix D. Tastatura je najzad odvojena od centralne jedinice u koju je ugrađen disk 1571 (350 K po disketi). RAM je i dalje 128 kilobajta, ali je predviđena njegova dalja ekspanzija do 640 K prostim dodavanjem čipova i pomicanjem mikroprekidača. C128D je opremljen serijskim i paralelnim interfejsom, portom za kartridže i priključcima za džojstike. Sa softverske strane, C128D može da emulira C64, da radi pod CP/M-om ili da omogući korisniku da programira u 128 jeziku.

Cena je 550 dolara a adresa *Commodore Business Machines Inc., 1200 Wilson Dr., West Chester, PA 19380, USA*. Računar se još par meseci neće pojavljivati u Evropi.

## Sa lispom na C

Ne radi se, verovali ili ne, o najavi za umetak koji bi poznavaoce lisp uputio u programiranje na C-u (ovakav bi umetak imao vrlo široku publiku od pet ljudi) — govorimo samo o programu firme *Engineering for Logic and System Application (ELSA) (Z.I. de Velizy Villacoublay, 32, rue Grange Dame Rose, 92360 Meudon-La-Forêt, France)* koji programe pisane na lispu prevodi u C. Ovakav „cross compiler“ je mnogo potrebniji nego što mislite: Lisp je veoma pogodan jezik za pisanje nekih aplikacija iz oblasti veštačke inteligencije, ali ima jednu veliku manu: spor je. Zato ima smisla razvijati program na lispu, a onda ga prevesti na C i doćnjeci kompajlirati — rezultat može da bude veoma brz.

CROSS LISP je kompatibilan sa MS DOS-om (12.000 francuskih franaka) i UNIX-om (35.000 FF).

## 2\*80386 = ?

Personalni računari sa Intelovim 32-bitnim procesorom 80386 niču kao pečurke posle kiše: firma *PC Discount (2000 Travis, Suite 630, Houston, TX 77002, USA)* je predstavila Nobbler 386 koji, zajedno sa megabajtom RAM-a, 40 M hard diskom i 1.2 M flopijem košta 4000 dolara. *American Micro Technology (14751-B Franklin Ave., Tustin, CA 92680, USA)* svoj model AMT 386 prodaje za hiljadu dolara manje, ali je ova cena varljiva — u nju nije uračunat hard disk. Model AMT 386, međutim, može da se pohvali brzim klokom (16 MHz) i superbrzom cache memorijom koja bitno ubrzava rad čitavog sistema.

6 računari 29 • avgust 1987.

## Lični mašinista

Firma *Computervision, München* izbacila je na tržište jedan novi programski paket, koji bi trebalo da olakša rad inženjerima strojarstva. Softver je veoma fleksibilan, a napravljen je sintezom ideja iz nekoliko poznatijih programa.

Program „*Personal Machinist*“ je izvrsna simulacija modeliranja pojedinih strojnih dijelova. Omogućuje njihovo prikazivanje u nekoliko različitih projekcija te kotiranje u svim smjerovima. Hardverski dio sačinjava jedna grafička tabla, koja služi kao olakšica pri unošenju podataka za konstrukciju strojnih dijelova, što je moguće s tastaturu, pomoću miša ili svjetlosnog pera. (I. M.)

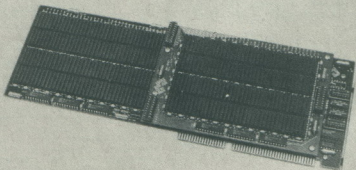
## Osam megabajta u jednom slotu

Ako ste pristalica tvrdnje da memorije nikada nema previše, obratite se firmi *Nonolithic Systems Corp, 84 Inverness Circle E, Englewood, CO 80112, USA* i naručite JustRAM karticu. Ova se čudna naprava priključuje u bilo koji slobodan slot vašeg AT-a i proširuje njegovu memoriju za sitnicu od 4 ili 8 megabajta: da bi stvar bila još lepša, u drugi slobodan slot možete da

ubacite još jednu ovakvu karticu i tako se proširite do teorijski maksimalnih 16 megabajta.

JustRAM može da se koristi kao *expanded* ili *extended* AT RAM (zamislite AT RAM disk od 8 megabajta), dok drugi operativni sistemi poput Xenix-a ili (budućeg) OS/2 mogu da ga konfiguriraju kao radni segment pristupaćom bilo kom komercijalnom programu.

Kartica od 8 megabajta košta 2690 dolara — ne baš koliko AT all...







## Kart(ic)e su podeljene

Na poznatom hamburškom sajmu predstavljene su RAM kartice kojima, po opštim procenama, pripada velika budućnost. Radi se o malim štampanim pločama koje su zaštićene čvrstim plastičnim kutijama; u kutiji se kriju RAM čipovi, poneko logičko kolo i, naravno, NiCd baterije. Toshiba kartice mogu da „zapamte“ 128, a Mitsubishije čak 512 kilobajta informacija — sasvim dovoljno za sve podatke o nosiocu, oznaku njegove krvne grupe, istoriju bolesti pa čak i stanje bankovnog računa. Kartica košta dvadesetak dolara, dok za čitač treba izdvojiti još \$100.

Ako jednog dana kompjuterske kartice zamene kreditne, hakerima će skočiti cena: svaka od ovih kartica može da se poveže sa PC-jem koji, naravno, može da menja njen sadržaj. Na primer podatak o stanju računa.

## Disk ili čip

Taman ste skupili pare i dobavili hard disk od 20 megabajta, a na sve strane je počelo da se priča da su diskovi mrtvi i da je samo pitanje kada da ih sahranimo.

Šta to konkuriše magnetnim memorijama? Verovali li ne, silicijum — RAM čipovi su postali toliko jeftini da kutija formata diska može da sačuva i čitavu stotinu megabajta. Samo se po sebi razume da je RAM stotinama puta brži, ali kako rešiti problem njegovog napajanja? Naravno, pomoću NiCd baterija koje, navodno, omogućavaju višegodisnje čuvanje podataka. Za sada su, istini za volju, ovakva rešenja nepraktična, ali su već najavljeni dinamički RAM-ovi od 64 megabajta (u prodaji se, istini za volju, tek stidljivo pojavljuju 16 megabitni čipovi), dok će se za samo pet-šest godina proizvoditi i RAM-ovi od 256 megabajta — zamislite pola gigabajta u 16 čipova!

Disk je, dakle, mrtav, ali će njegova sahrana morati da priček. U međuvremenu će se hard disk koji ste kupili itekako amortizovati!

## Fantastika iz svakodnevnog života

Douglas Adams, pisac čuvene knjige *Hitch-Hiker's Guide to the Galaxy* (u slobodnom prevodu „Autostopom kroz Galaksiju“) je ponovo došao u centar pažnje. Nije ga, istini za volju, stigla policija neke od devet američkih saveznih država (za gospodinom Adamsom je u ovim državama raspisana poternica zbog raznih kriminalnih dela ali on mirno živi u San Francisku), ali je njegova softverska firma *Activision* izbacila novu igru avanturu nazvanu *Bureacracy*. Igra je zasnovana na stvarnom događaju što je, uzgred budi rečeno, sasvim prirodno za čoveka koji je u životu doživeo toliko avantura.

Daglas Adams je u nekoj prilici prodao svoj mali stan i preselio se u novu kuću koju je, naravno, kupio na kredit. Banka je svu obavestjenja o ratama slala na Adamsovu staru adresu, što njega baš i nije ljutilo — tamo se doselila privlačna mlada dama. Kada je dama nekuda otišla, Adamsu se više nije lišto u stari kraj: napisao je pismo banki i rekao da oni, pre svih drugih, treba da znaju gde se nalazi imovina na koju drže hipoteku. Banka se uljudno izviniła za ovu grešku i poslala izvinijenje na — staru adresu!

Cilj igre je, bar formalno, da navedete banku da registruje vašu pravu adresu. Kažemo „bar formalno“, jer ćete mnogo više festički dobiti ako uspete da izbegnete otplatu kredita — sada valjda znate zašto Daglasa Adamsa traže u devet država.

Da vam otkrijemo jednu malu tajnu igre: vrata od garaže treba da udarite nogom tek kada iznad vas proleti Boeing 747 sa obližnjeg aerodroma. Svi ostali pokušaji da otvorite vrata će dovesti do daleko neprijatnijih i komičnijih situacija.

Umalo da zaboravimo: igra košta 30 funti i izvršava se na PC-ju ili Atariju ST.

## Nonsense in BASIC

## Gvoždurija druge vrste

Jednog trenutka mi se učinilo da mogu dati ozbiljan i značajan doprinos opštim naporima za razvoj domaće računarske terminologije (veštačkog govora). Sinula mi je originalna ideja (osobita zamisao): da zavirim u neki stari engleski rečnik iz doba kad ni Englezi ni Amerikanci nisu imali računare. Tako bih utvrdio na koji su način oni stvorili svoju terminologiju. Posle bih, kako sam u prvi mah zamišljao, lako primenio isti metod (smišljeni i planski postupak pri radu) na naše uslove i stvorio domaću terminologiju.

Pokazalo se da je sve to bio račun bez krčmara. Prvobitno značenje reči **HARDWARE** je gvoždurija. Englezi očigledno mogu za hardver da kažu „gvoždurija“, ali to kod nas ne ide. Zamislite, na primer, sebe kako izgovarate ovu rečenicu: „Moj računar ima bolju gvožduriju nego tvoj!“

Reč **SOFTWARE** nisam našao, očigledno je kasnije skovana po uzoru na gvožduriju. Tamo gde bi po abecednom redu trebalo da piše **SOFTWARE** stajalo je **SOFT-HEADED**, što će reći — slabouman. Nisam našao ni **BYTE**, ali sam pomislio da je ta reč možda postala od izraza **BIT BY BIT**, to jest mic po mic, malo po malo. **S** drugie strane, **BIT** ima dva značenja: 1. komadić, parče, 2. žvala. Ne moram da napominjem da je ovo drugo značenje očigledno neprikladno kad su u pitanju računari.

Vidim da i drugi imaju probleme sa bajtom. Neki muzički nastrojani i novinari pokušali su da lansiraju (*hitnu*) reč **OKTET**. Pojavio se i sportski kontra-predlog **OSMERAC**, što bi već moglo da ima nekog smisla. Bajt sa setovanim nultim bitom mogao bi u zapadnoj varijanti (nešto što ima različit oblik od **nečega**, a što ima se tim isto poreklo) da bude **OSMERAC** SA KORMILAROM, u istočnoj varijanti **OSMERAC** SA KRMAROM, a u severoistočnoj **OSMERAC** SA KRMANOŠEM.

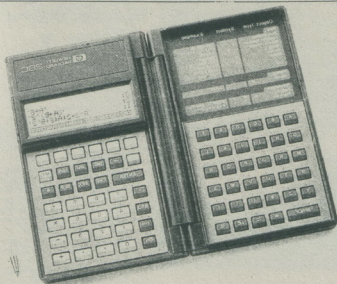
Totalno (celokupno) sam se zapetljao kad sam počeo da tražim ekvivalente (stvari iste vrednosti) za nešto složenije izraze, kao što su **DESK TOP** ili **RUN-TIME LIBRARY**, a još više kad sam pokušao da prevedem reč **MIKROPROCESOR**. To mi je ujedno bio i poslednji napor. Prekopao sam nekoliko rečenica i ništa pametnije nisam smislio od izraza **SITNOPOSTUPALO**.

Sad vam je svakako jasno zašto sam na kraju odustao. Badaiva i plienitosti zamisli, i spremnosti na samopozivani rad. Nisam ja za to, pustilicu lingviste (jezicari) da rade svoj posao na miru. Mora biti i da njima nije lako!

Što se mene tiče nisam se sasvim oslobodio more u koju sam zapao listajući rečnike. Sada mi po glavi kolaju i druge strane reči koje bi trebalo prevesti. Nijednim rešenjem nisam zadovoljan, pa me to još više muči. I vama se, verovatno, ne bi sviđalo da se televizor zove *slikogled ili mrdostik*, cigareta — *rakotvor*, kasetofon — *vrtozvek*, telefon — *daljninglas*...

Ako mislite da su reči i izrazi u zgradama mojih ruku delo, varate se! Sve što je u zgradama doslovno sam prepisao iz Vujaklijinog „Rečnika stranih reči“.

**Bata Bajt**



## Treća generacija

Dve generacije kalkulatora su za nama. Prvu čine kalkulatori sa velikim brojem ugrađenih numeričkih funkcija, ali bez mogućnosti programiranja. Drugu, započetu sa HP-65 čine programabilni kalkulatori koji su sa HP-41 prešli u ručne računare. Treća generacija upravo započinje Packardovim modelima HP-18C i HP-28C i bazira se na menijima, višelinjskom displeju, „soft-tasterima“, simboličkoj manipulaciji izrazima i „rešavaču formula“.

HP-18C, poznatiji kao Business Consultant je neprogramabilan, poslovno orijentisan kalkulator, izbačen u junu prošle godine sa ciljem da isplati troškove razvoja novih tehnoloških rešenja. Prvo od njih je **dvodelno klučite** koje se otvara poput knjižica i ima **dve tastature** i savitljive kontakte na pregibu. Na ovaj način je rešena potreba za više tastera uz male dimenzije mašine. Sledeći novitet je **displej od 4 reda** po 23 karaktera ili  $32 \times 138$  tačaka. Poslednji red se koristi za ispis menija koji čine sledeću novost. Prvi put kalkulatori rade sa menijima i njima pridruženim „**soft-tasterima**“, što dosta olakšava rad ne samo „biznismenima“ nego i tehničkim profesionalcima. Poslednji novitet koji je HP-18C doneo je „**rešavač formula**“, koji omogućava da zadamo proizvoljan izraz san nepoznatih, zatim preko manija za n-1 odredimo vrednosti pa tražimo rešenje za n-tu neovisno o tome gde se one u izrazu nalazi. Podrška poslovnim proračunima ne spada u teme vredne pomena, jer na programersko-korisničkom planu ne donosi ništa.

HP-28C se pojavio početkom ove godine i pored svih već pomenutih noviteta sadrži sve numeričke funkcije serije 10 (ali sa 12-to cifarskom tačnošću), aritmetiku kompleksnih brojeva, kao i računanje sa matricama i vektorima; realnim i kompleksnim. Rešavač formula pokriva

sve numeričke operacije koje HP-28C poznaje, a proširen je mogućnošću **crta-nja grafika proizvoljnog izraza**. Ugrađena je i **konverzija mernih jedinica** za 120 raznih veličina, uz mogućnost definisanja novih.

Glavni novitet ovog „kalkulatora“, koji je u stvari računar, predstavlja **simbolička matematika** (da ne kažemo algebra). Zadaemo simboličke varijable u izrazima i imamo na raspolaganju: skraćivanje i proširivanje izraza, smenu varijabli, diferenciranje i integriranje polinoma i izdvajanje varijabli.

Sve je to moguće zahvaljujući tome što ovaj računar radi sa **objektima** međ koje spadaju i **liste**. Da stvar bude lepša, radi se sa **stogom negrančene dubine** koji sadrži **bilo kakve objekte**, a programiranje je potpuno strukturirano. Sve u svemu, čovek ima utisak da radi na **fortru** proširenom rečima za rukovanje listama i objektima.

Čini vam se mnogo za ručni računar izrastao iz kalkulatora? Podatak da HP-28C ima isti **processor i internu organizaciju kao HP-71B sa 128K ROM-a** objašnjava mnogo toga, a njegova počka ina cena od 240 dolara (više od polovine cene HP-71B) takođe. Znaajući Packardovi filozofiju, za očekivati je da će se pojaviti modul koji će „**sve lepe stvari**“ raditi na HP-71B. Popularno ime za HP-28C koje se nalazi ispisano na klučistu je **Scientific Professional i ne naplaćuje se posebno**. Za HP-18C i HP-28C (možemo ih već nazvati serijom **XB**) postoji terminalni printer koji daje 24 karaktera na traci širine 58 mm, a sa mašinom je „povezan“ IC-zracima što predstavlja još jedan novitet. U nekom od sledećih brojeva očekujte potpuni prikaz HP-28C.

Ž. Berberski

Komodor 64C je, zapravo, prepakovani C64: stara štampana ploča je ugrađena u kutiju koja neobično podseća na C128, tastatura je poboljšana i svemu tome pridodata disketa sa GEOS-om, nekom vrstom osobitnog programa **Windows**. Za računarske početnike ćemo reći da je C64 (a samim tim i C64C) zasnovan na osobitnom mikroprocesoru 6510 (verzija popularnog 6502) koji radi na frekvenciji od 1.02 MHz, da je opremljen sa 64 kilobajta RAM-a, 20 kilobajta ROM-a i kolor grafikom 320\*200 koja se svodi na 40 znakova u redu. Jaka strana C64C je trokanalni generator zvuka, a slabe bežik i spor disk (nije mnogo brži od kasetofona).

## I dalje škrta

Komodor 64C košta 230 dolara, ali čete uz njega pozeleti da nabavite i neke dodatke. Tu je, prve svega, disk jedinica 1541C koja je identična sa 1541, ali je prepakovana u novu kutiju tako da po dizajnu odgovara „novom“ kompjuteru; cena ove disk jedinice je 230 dolara. Kolor monitor 1802 košta sledećih 250 dolara, a miš, koja GEOS intenzivno koristi, 50 dolara. Komodor je za jesen obećao proširenje RAM-a od 256 K koje će nositi oznaku 1764 i koštati oko 150 dolara; ovaj RAM, međutim, mogu da koriste samo dobri poznavoci asemblara, jer je, kao i mnogi drugi specijaliteti C64, nepristupačan iz bežika.

Uz C64C dobijate novu knjižicu nazvanu **Commodore 64C Introductory Guide** koja na samo 32 strane opisuje povezivanje računara sa televizorom (monitrom), diskom i štampačem i učitavanje komercijalnih igara. Korisnici koji planiraju još nešto osim igara će rado pročitati 200 strana knjige **Commodore 64C System Guide** koja opisuje bežik i operativni sistem; radi se, zapravo, o proširenom uputstvu za C64. Knjiga, na svu sreću, už čitačca da, koristič PEEK i POKE, kontrolise generator zvuka i sprajtove; smatramo, ipak, da je ovakvo programiranje izvan domena većine početnika. Najambiciozniji korisnici svakako treba da nabave knjigu **Commodore 64 Programmer's Reference Guide**; ako vam original nije pri ruci, svaki ozbiljniji pirat će vam rado prodati kopiju ili prevod.

## Prozor na „komodoru“

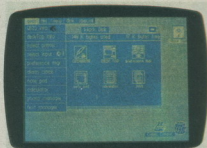
GEOS se, kako rekamos, isporučuje uz C64C, ali i vlasnici modela 64 mogu da ga koriste ako plate 60 dolara. Radi se, kratko rečeno, o operativnom sistemu koji bi trebao da približi kompjuter apsolutnom početniku koji ne može da zapamti gomilu komandi, ali zato može da dovede kursor do vinjete koja predstavlja program, prenese tu vinjetu do kante za dubre i tako je obriše. GEOS je u mnogome sličan odgovarajućim programima na „mekintoshu“ i „atariju 520 ST“; ali ne može da se meri sa njima po upotrebljivosti — „usko grlo“ nije 6510 (koji je, jasno, daleko spirilji od besnaestobitnih Motorolaovih procesora) već očajno spori disk koja program stalno poziva. GEOS, dakle, treba shvatiti kao neku vrstu praktikauma za upotrebu modernih operativnih sistema.

Integralni dio GEOS-a su tekst procesor **geoWrite** i program za crtanje **geoPaint**;

# „Komodor“ u novom ruhu

Redovni čitaoci kompjuterskih časopisa dobro znaju koliko je Alan Šugar preduzimljiv — malo, malo pa se pojavi neki novi „spektrum“! Sa druge strane okeana stvari teku mnogo sporije — Komodor očito smatra da „tim koji dobro igra ne treba menjati“, pa je stari dobri model 64 oslobođen bilo kakvih novotarija.

Početak 1987, međutim, Komodor je odlučio da ubrizga injekciju staroj „šezdeset četvorci“ i tako je nastao model 64C. Verujući da mnogi čitaoci „Računara“ tek treba da nabave svoj prvi kompjuter, ukratko ćemo se osvrnuti na ovaj Komodorov potez.



iaiko su ovi programi integrisani, ubacivanje crteža u tekst nije baš sasvim jednostavno.

Najglupija osobina GEOS-a je što su ga njegovi autori zaštitili i to prilično dobro — dok se ne pojave piratske kopije, moraćete svaki put da učitate GEOS sa originalne diskete. Ukoliko se, dakle, originalna disketa ošteti, pišete u Ameriku pa ćete (jednog dana) dobiti novu. Da bi stvar bila još neprijatnija, na disketu ne smete da lepите write protect nalepnicu što znači da neko neoprezno brisanje može da onesposobi jedinu kopiju sistemskog diska! Sva je sreća što ovaj program sam po sebi nije posebno koristan — čuvaćete ga i pokazivati prijateljima koje treba da ubeđete da „Komodore 64“ može sve što i mnogo skuplji „Atari 520 ST“.

## Ponovo u igri

Kada se sve sabere i oduzme, jedina prava prednost „Komodora 64C nad „starijim bratom“ je bolja tastatura što, istini za volju, i nije mala stvar. Glavni razlog koji može da vas navede na kupovinu ovoga računara je svakako izvanredna softverska podrška — oglasi su puni najnovijih igara i komercijalnih programa koji se prodaju po najnižim zamislivim cenama. „Komodore 64 C“, međutim, nije naročito upotrebljiv kao poslovni sistem: četrdeset znakova u redu je premalo za obradu teksta, dok spor disk otežava rad sa bazama podataka. Što se edukacije tiče, loš i zastareo bezjzik teško da predstavlja preporuku, ali je operativni sistem C64 pisan vrlo fleksibilno, što znači da je njegovo proučavanje zanimljivo iskustvo za buduće sistemske programere. Cena je, najzad, umereno visoka ali će pojava C64C naterati prodavnice da obore cenu C64 i tako se oslobode zaliha — ukoliko iole duže boravite u inostranstvu, možda ćete za sasvim malo para doći do C64!

Dejan Ristanović

## Commodore 64 C — tehničke karakteristike

Mikroprocesor	6510
Clock	1.02 MHz
ROM	20 K
RAM	64 K
Tastera	64
Funkcijskih tastera	4
Ekran	TV, monitor
Tekst	40*25
Grafička rezolucija	320*200
Flopi disk	1541 C (opcija)
Časovnik realnog vremena	ne
Ekspanzija	2 džojstik porta, 44-pinski bus expansion, user port
Operativni sistem	Basic 2.0 u ROM-u, GEOS 1.3
Cena	\$ 230 + \$249 (1802 kolor monitor) + \$229 (1541C disk) + \$49 (1351 miš).



# Plavi patuljak

**Pre dva meseca smo predstavili IBM-ovu novu seriju računara nazvanu PS/2 baveći se uglavnom modelima 50, 60 i 80. Nije ni čudo — ovi će računari ubrzo raditi pod novim i (navodno) revolucionarnim operativnim sistemom OS/2. Iako će IBM-ov model 30 zauvek raditi samo pod MS DOS-om, ne treba sasvim zaboraviti na njega — radi se, ako ništa drugo, o prvom IBM-ovom PC klonu! Model 30, ujedno, treba da pruži odgovor na legendarno pitanje koliko je IBM zapravo IBM kompatibilan?**

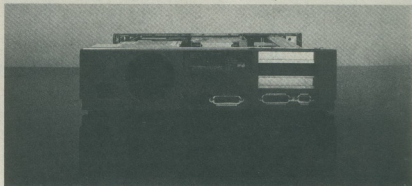
Model 30 je, pre svega, znatno manji od PC-ja ili XT-a, ali ovo smanjenje verovatno nećete primetiti — računar zauzima praktično isti deo stola ali je bitno niži (10"41"39.6 cm). Smješten je u lepo dizajniranu sivu plastičnu kutiju sa čije su prednje strane prekidač, tradicionalna ključaonica i disk jedinica od 3.5 inča. Što se masovne memorije tiče, model 30 se isporučuje u dve verzije: model 30-002 ima dve disk jedinice i košta 1100 funti, dok model 30-020 ima jednu disk jedinicu i hard disk od 20 megabajta, ali zato košta čitavih 1560 funti. U ovu cenu nisu uračunati nova i izvanredna tastatura (190 funti), IBM-ov monohrom monitor (200 funti), miš (70 funti) niti diskete sa PC DOS-om 3.3 (čitavih 70 funti). Kompletan sistem sa hard diskom, dakle, košta nekih 2000 funti, što svakako nije malo.

## Konstruktor široke ruke

U središtu računara je Intelov mikroprocesor 8086 koji je sa softverske strane potpuno kompatibilan sa „starijim bratom“ 8088, ali su mu magistralne šesnaestobitne što rezultira bržim zahvatanjem podataka iz memorije i, samim tim, bržim izvršavanjem programa. Zanimljivo je da 8086 koji je ugrađen u model 30 radi na 8 MHz, što je za IBM veliki pomak — PC i XT su radili na samo 4.77 MHz dok originalni AT radi na 6l Ne može se, istini za volju, očekivati da model 30 bude brži od AT-a (mikroprocesor 80286 je mnogo bolji i brži od 8086), ali će ubrznani klock svakako dobro doći svim ozbiljnim komercijalnim programima.

IBM se ovoga puta uzdržao od svog uobičajenog škrtačenja na RAM čipovima: model 30 je opremljen sa 640 K RAM-a, što znači da ni po čemu ne zaostaje za raznoraznim tajvancima koji su se do sada uvek hvatali velikom memorijom. Samo se po sebi razume da dalje proširenje RAM memorije nije moguće i da mikroprocesor 8086 radi sa segmentima — u bežikvu ili paskalu čete i dalje raspolagati sa samo 64 slobodna kilobajta, dok glavnu memoriju možete da koristite kao RAM disk.

Što se grafike tiče, na ploču modela 30 je ugrađena nova grafička kartica nazvana MCGA (Multi Colour Graphics Array). Iako je MCGA naslednik stare CGA kartice, ne morate da se bojite ružnih slova: svaki je znak iscrtan na matrici 8"16 što, na primer, ne zaostaje za Herkules standardom. Uz sve grafičke modove CGA kartice, MCGA može da se pohvali sa dva nova: 320\*200 tačkica u 256 boja i 480\*640 u dve boje. Sve se boje



### PS/2 model 30 — karakteristike

Mikroprocesor	8086
Koprocesor	8087 (opcija)
Clock	8 MHz
ROM	64 K
RAM	640 K
Tastura	102
Grafička kartica	MCGA, EGA, Herkules (opcije)
Rezolucija	640*480, 320*200
Boja	256
Paleta	262144
Hard disk	20 M (model 30-020)
Flopi disk	2*720 K (model 30-002), 1*720 K (model 30-020)
Interfejsi	RS 232, miš, Centronics
Časovnik realnog vremena	da
Ekspanzija	3 PC slota
Operativni sistem	PC DOS 3.3
Cena (funt)	1156 (model 30-002), 1558 (model 30-020), + 189 (tastatura), + 201 (monohrom monitor), + 505 (kolor monitor), + 69 (miš), + 70 (PC DOS 3.3).

biraju iz fantastične palete od 262144 nijansi (odakle ovaj broj? 262144 je, jednostavno, 2 na 18).

Jedan od dobrodošlih noviteta je što model 30, „oseća“ grafičku karticu koja je eventualno udenuta u neki od ekspanzionih slotova i prepusta joj kontrolu — Amstrad bi mogao mnogo da nauči od IBM-a! Model 30 se, dakle, može dopuniti EGA ili Herkules karticom, pa čak i novim IBM-ovim VGA kontrolerom koji emulira EGA, CGA i Herkules i dodaje nekoliko interesantnih modova o kojima smo pisali pre dva meseca. VGA kontroler se standardno ugrađuje u model 50, 60 i 80.

Pogled na zadnju stranu kutije otkriva

portove za tastaturu, miša, dvosmerni osmofitni Centronics, RS 232C i RGB. Tu su, naravno, i „prozorići“ koji obezbeđuju pristup ekspanzionim portovima kojih ima tri — model 30 odstupa od ostalih računara iz serije PS/2 utoliko što se na njega priključuju stare kartice. Prvi testovi pokazuju da je hardverska kompatibilnost sa ovim proširenjima stotocentna.

Ostalo je još da pomenemo masovnu memoriju: model 30 koristi disk jedinice od 3.5 inča, pri čemu se na svaku disketu upisuje po 720 kilobajta informacija. Format je, dakle, novi i nekompatibilan kako sa starij PC-jem tako i sa serijom PS/2; olakšavajuća je okolnost što je priključenje eksternih disk jedinica od 5.25 inča maksimalno pojednostavljeno, pa prenos nezastučenih programa ne predstavlja poseban problem.

## Sistemska softver

Model 30 nikada neće moći da radi pod novim operativnim sistemom OS/2, što znači da je BIOS pretrpeo minimalne izmene — IBM je tako izbegao razne probleme sa kompatibilnošću. Model 30 savršeno radi pod bilo kojom verzijom MS DOS-a ili PC DOS-a, ali je uz seriju PS/2 lansirana nova verzija koja nosi oznaku 3.3. DOS 3.3 nije, kao što se u prvi mah mislilo, identična sa DOS-om 3.2, ali razlike nisu naročito primetne. Uvedena je, pre svega, interna CALL komanda koja omogućava da jedna BAT procedura pozove drugu. Ovakav je poziv, izvršen za volju, i do sada bio moguć, ali je zaštetak pozvane BAT procedure značio kraj rada; sada se kontrola vraća prvoj proceduri.





Korisnici hard diska će pozdraviti komandu APPEND koja omogućava da produžite PATH spisak — do sada je dodavanje novog direktorijuma zahtevalo kucanje komande koja se često prostire u par redova i koja je puna naopakih kosih crta i čudnih skraćenica.

FASTOPEN je razidentni uslužni program koji ubrzava rad sa diskom — svako otvaranje neke datoteke se upisuje u internu tablicu, pa se zatim otvaraju kanali prema često korišćenim datotekama. Docnije otvaranje ovih datoteka je bitno brže što se, jasno, odražava na ukupnu brzinu rada računara. Vlasnici flopi diskova verovatno neće koristiti ovu rutinu jer bi njeno aktiviranje oduzimalo po malo prostora na svakoj radnoj disketi.

Koliko je model 30 kompatibilan sa

ranijim PC računarima? Do sada nije pronađen ni jedan jedini program koji pravi probleme; rade čak i igre poput „Zaxxon“-a ili „Digger“-a. Najveće iznenađenje predstavlja bežik program 730 colors koji direktno pristupa registrima 6845 CRT kontrolera koji se ugrađuje na CGA karticu. Iako model 30 uopšte nema 6845 kontroler, program radi savršeno! IBM-u se često može ponešto zameriti, ali niko ne sme da kaže da je „veliki plavi“ nemaran i da ne obavlja poslove od kraja!

#### Na dve stolice

Podrobna analiza pokazuje da model 30 teško možemo da ubrojimo u seriju PS/2: tastatura i kutija su, istina, novi ali je grafika stara, format zapisa na disketama

savim poseban, konektori za kartice nisu promenjeni... Model 30, jasno, ne spada ni u staru generaciju PC-ja: disk jedinica od 3.5 inča i nova grafička kartica čine priličnu razliku. Ima se utisak da model 30 predstavlja neku vrstu prelaznog oblika između PC-ja i PS/2 a prelaznih se oblika uvek treba čuvati — preti vam da vaš računar ostane softverski napodržan. Osim toga, cena modela 30 je takva da za iste pare možete da kupite nekog renomiranog AT klonu i tako dodate do mnogo bolje mašine; ne treba, naravno, isključiti ni kupovinu modela 50 koji košta „samo“ 700 funti više. Ne zaboravite da će OS/2 raditi na bilo kom AT-u, ali ne i na modelu 30!

Dejan Ristanović

# Ko je obojio Herkulesa?

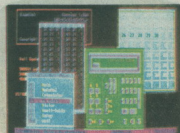
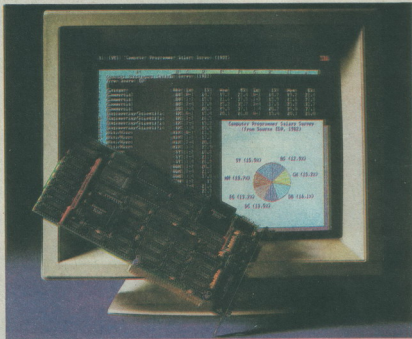
*Niko, izgleda, nije savršen, pa čak ni sam IBMPC — originalni model nije imao grafiku što je mnogo nezavisnih firmi začas iskoristilo. IBM je u međuvremenu izbacio čak četiri grafičke kartice, ali sav taj trud nije mogao da anulira početnu nemarnost — PC će, pored ostalog, ostati zapamćen kao računar sa najviše (međusobno nekompatibilnih) grafičkih standarda u istoriji. Među mnogobrojnim proizvođačima grafičkih kartica izdvaja se ime relativno male američke kompanije Herkules — Herkules kartica je stekla pravo građanstva u više miliona PC računara. Posle ne naročito zapaženog poboljšanja nazvanog Herkules Plus, letnji meseci 1987. nam donose novu Herkules karticu — InColor GB222.*

Da bismo shvatili filozofiju Herkules grafičkih kartica, moramo da se upoznamo sa razlozima zbog kojih originalni PC nije imao grafiku. IBM je, pre svega, proizveo bezbroj velikih računarskih sistema i stekao vrlo dobar utisak o tome kakvi su terminali potrebni korisnicima — posetite bilo koji računski centar i pronaći ćete terminale poput VT 50, VT 100 ili, u boljem slučaju, VT 200. Grafičke mogućnosti ovih terminala su, verovali ili ne, gotovo nikakve — ekran je monohromski, radi se isključivo sa slovima i (eventualno) grafičkim karakterima, skrolovanje teksta nije baš impresivno brzo... Razlog za ovakve karakteristike nije samo težnja da se cena snizi — racionalni rad velikog sistema zahteva da se na komunikaciju sa terminalima troši što manje procesorskih sekundi dok rad sa grafikom zahteva prenos ogromnih blokova podataka (da li je lakše poslati ASCII kod slova 'A' ili njegovu bit mapu?). Samo se po sebi razume da je grafika neophodna za neke specijalne aplikacije, ali korisnici kojima su ovakve aplikacije potrebne rado investiraju prilična sredstva u grafičke terminale, plotere i sličnu opremu.

## „Lotus“ je kriv za sve

Pošto je PC zamišljen kao jeftin poslovni sistem, IBM-ovi inženjeri su smatrali da on može sasvim lepo da živi bez grafike — na kraju krajeva, na raznim velikim računarima su implementirane sve moguće poslovne aplikacije, pa ipak ni jedan korisnik nije zakuckao za grafikom. IBM se, najzad, potrudio da set karaktera bude začuđujuće kompletan (koriste se kodovi 32—255 a ne, kao u ASCII standardu, 32—127) što je omogućavalo nekakvu rudimentarnu grafiku — obezbeđene su jednostruke i dvostruke linije, i svi njihovi neophodni preseci. Prvi poslovni programi su zaista funkcionalni bez grafike. A onda su se pojavile prve grafičke kartice pa su autori softvera počeli da dopunjavaju svoja remek-dela okvirima, linijama, osenčenim površinama i sličnim specijalitetima. Grafika je u početku bila samo dekorativna, a onda je paket „Lotus“ izazvao pravu malu revoluciju — po prvi put je personalni računar upoznao integrirani program za unakrsna izračunavanja i poslovnu grafiku.

Najrazličitije firme su jednostavno poludele za „Lotusom“ — što je značilo da stotine njihovih kompjutera treba dopuniti grafičkom karticom — pitanje je samo kojom. Najprilagodnije je obratiti se IBM-u, ali je ovakvo obraćanje davalo sumnjive rezultate — IBM je prodavao samo glupost zvanu



CGA. CGA kartica, istini za volju, otvara PC-ju grafičke perspektive, ali njena ugradnja podrazumeva odvratno ružna slova, nisku rezoluciju, spor rad i relativno ružne slike. Možda bi sve to nekako prošlo; CGA kartica, međutim, zahteva da kupite nov kor monitor, što znači da stari (skup ali izvanredan) IBM-ov monohrom monitor možete da bacite. Herkules je povukao mnogo bolji potez — kada kupite Herkules karticu na starom IBM-ovom monohrom

monitoru imaćete lepa slova i sasvim pristojnu crno-belu grafiku! Mnoge su se firme odlučile za ovakvu nabavku, pa je Herkules kartica postala gotovo oficijelni standard; najbolji način da neka softverska firma propadne je da uz svoje programe ne isporučuje dražer za Herkules karticu!

Herkules kartica je, rekosem, u svetu prilično zastupljena, ali je u Jugoslaviji njena zastupljenost praktično stoprocentna — autor ovoga teksta koji ima (najreću da koristi EGA karticu je morao da preinstalira svaki program koji mu je došao do ruke! Da stvar bude još gora, čitava domaća softverska produkcija je zasnovana na Herkules kartici, što proizvelo mnogo komičnih efekata — pošto sam u svim prilikama kukao da za IBM nije napisan ni jedan pristojan editor prilagođen programerima (ne, pominjete mi *IBM Professional Editor* na koji sam avertirao), jedan od saradnika „Računara“ mi je poklonio svoj originalni editor, koji je, koliko sam na licu mesta primetio, lepo radio. Editor je, istini za volju, radio i kod mene — uz njegovu se pomoć čak i mogao ispravljati program. Nevolja je, međutim, što se na ekranu nije videlo baš ništa —

ako znate da je peti znak osme linije pogrešan, možete da ga promenite ali ovaj rad teško mogu da preporučim. Domaći programeri, izgleda, smatraju da je *sasvim prirodno* poukovati znakove direktno u memorijske ćelije ?? — ?? — niko ne pomišlja da koristi BIOS ili MS DOS rutine!

Posle velikog uspeha Herkules kartice, Herkules Plus je izjavio relativno malo interesovanja — nova kartica je bila sasvim slična „starijem bratu“, s tim što je omogućavala definisanje znakova i rad sa dodatnim setovima karakterima koji su nazvani *Ramfont*. Cena je, međutim, bila takva da je malo ko smatrao za potrebno da kupi Herkules Plus — definisanje karaktera je lepa stvar, ali se vrlo često na zadovoljavajući način rešava promenom EPROM-a. Herkules je početkom 1987. zaključio da je vreme za generaciju grafičkih kartica i tako je nastala *InColor GB222*.

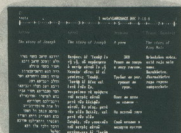
## Anatomija jedne kartice

InColor GB222 zauzima bilo koji PC, XT ili AT slot i povezuje se sa EGA kompatibilnim kolor monitorom (frekvencija horizontalnog skeniranja 22 KHz, frekvencija vertikalnog skeniranja 60 Hz). Poput ostalih Herkules kartica, InColor GB222 nije parlamentaran — ako već imate neku drugu grafičku karticu, moraćete fizički da je odstranite. PC klonovi koji na osnovnoj štampanoj ploči imaju ekvivalent neke grafičke kartice najverovatnije neće podneti InColor GB222 — „amstrd PC1512“, na primer, ne radi sa novom Herkules karticom.

Osim RGB izlaza, InColor GB222 dopunjava zadnju stranu vašeg PC-ja standardnim Centronics konektorom — ovaj port se zove LPT1; i ne može jednostavno da se primenjuje u neko drugo logičko ime. Ukoliko ste, dakle, do sada poveljavali štampač sa RS 232/Centronics karticom, moraćete da preместite njene mikroprekeidra i „nazovite“ ovaj port LPT2: — ako vaša kartica nema mikroprekeidra, moraćete da je odstranite i tako ostanete bez serijskog interfejsa!

InColor GB222 nudi rezoluciju 720\*348 u 16 boja koje se biraju iz palete od 64. Karakteri su definisani na matrici 9\*14, što znači da se u svakom redu ispisuje po 720/9=80 znakova. Karaktere, međutim, možete da zuzite (8\*14) i tako radite sa 720/8=90 slova u redu. Povećanje broja slova u redu je vrlo zanimljiv fenomen — nekada se smatralo da su 64 slova sasvim dovoljna, onda su korisnici zaključili da poneko slovo više ne škodi i tako se stiglo do 80. Osamdeset slova u redu je zaista dovoljno, ali se pojavljuju sve više paketa koji ovaj broj do krajnosti iskorišćavaju. Tako se ponovo pojavljuje želja za nekim slovom više — paket koji radi sa 80 slova u redu postaje daleko moćniji kada se ekran proširi ponekom marginom. Tako će 90 slova u redu biti dobrodošao luksuz — korak po korak, personalni računari će za koju godinu svakako dostići profesionalne terminale koji odavno nude 132 znaka u redu.

Kartica InColor GB222 je opremljena sa 256 kilobajta sopstvenog RAM-a koji može da se konfiguriše na razne načine. U tekst modu koriste se samo 4 kilobajta za ASCII mapu ekrana; oblici znakova su upisani u ROM, dok se ostatak RAM-a jednostavno



ne koristi. Ovaj je mod kompatibilan sa svim komercijalno raspoloživim programima koji rade sa tekstom.

U Ramfont modovima prvih 16 kilobajta memorije sadrže mapu ekrana, dok ostatak RAM-a eventualno sadrži definiciju znakova. Prvi Ramfont mod podrazumeva postojanje jednog seta od 256 znakova koji mogu proizvoljno da se predefinišu — možemo, na primer, da promenimo samo deset znakova i tako uvedemo YU latinšica slova ali možemo i da promenimo kompletan set znakova i uvedemo ćirilicu, grčka slova ili nešto drugo. Ovaj mod je kompatibilan sa većinom komercijalno raspoloživih programa.

Drugi Ramfont mod je specijalitet kartice InColor GB222: 48 kilobajta memorije je rezervisano za ekran, a ostatak za najviše 12 različitih setova znakova. Osim bita, na Zalost, više nije dovoljno da se jednoznačno označi neko slovo — potrebna su i četiri bita za izbor seta znakova što znači da za svako slovo mesto potrebno 12 bita. Ovaj „novi ASCII“ omogućava da se na ekranu istovremeno ispisuju tekstovi na raznim jezicima (naša slika ilustruje hebrejski, grčki, ruski i sanskrit) ali, naravno, nije kompatibilan sa postojećim tekst procesorima i drugim programima. U poslednje se vreme, međutim, pojavljuju „poliglotski“ tekst procesori (pogledajte „Računare 27“) koji će se očito dobro slagati sa InColor GB222 karticom.

U grafičkom modu kompletna memorija je podeljena na dve stranice; svaka stranica sadrži po jedan kompletan opis ekrana, što znači da jedna može da se prikazuje dok drugu modifikujete. Opis ekrana se sastoji od četiri bit mape, od kojih svaka odgovara jednoj od boja — boja neke tačke se dobija superpozicijom odgovarajućih elemenata bit mape. InColor GB222 ne sadrži grafički koprocensor niti bilo kakav hardver koji bi podržao tzv. *blitter* grafiku koju smo upoznali kada smo predstavljali „amigu“; Herkules kartice su uvek bile prilagođene relativno konzervativnom PC standardu. U centru InColor GB222 kartice se, naime, nalazi 6845 CTRL kontroler koji je za ovu priliku dopunjen sa dva ULA čipa koja automatižuju pojedina pomeranja delova ekrana, ali je ova automatika daleka od mogućnosti pravih grafičkih koprocensora koji, nezavisno od glavnog procesora, mogu čak i da kontrolišu sprajtove.

## Obojeni „SuperKey“

Uz InColor GB222 karticu dobijate i disketu sa nekoliko programa. Tu su, pre svega, drajveri za razne poznate softverske

pakete kao što su „AutoCAD“, „Lotus“, „Symphony“, „Manuscript“, „Word“, „Windows“, „Framework II“ i „Javelin“; uz pomoć ovih programa nova, Herkules kartica može da preuzme funkciju bilo koje grafičke kartice koju ste ranije koristili. Programerima su, međutim, potrebne i neke alate, pa su se tako na disketi našli programi SETCOLOR i PALETTE: prvi omogućava da pregledate raspoložive boje i odaberete one koje ćete u daljem radu najčešće koristiti. Druga omogućava da promenite 16 boja koje koristi neki aplikativni program i tako taj program prilagodite svom ukusu ili svom monitoru. Ljudima je, naime, vrlo teško ugoditi: neko voli bela slova na tamnoj pozadini, neko drugi bela slova na zelenoj pozadini, treći volje *IBM Professional Editor* i bela slova na plavoj pozadini, autor ovog teksta uvek koristi žuta slova na crnoj pozadini (ah, taj „Turbo Paskal“) ... Pokazalo se da PALETTE.COM uspeva da „prevari“ čak i osetljive programe kao što su „SideKick“, „SuperKey“ i „ProComm“.

Uslužni program RAMFONT se koristi za učitanje i promenu setova znakova; na disketi je dato 25 primera koji se kreću od korisnih internacionalnih setova znakova (jedan od njih je ćirilica ili ruska) do komične imitacije rukopisa. Utisak dopunjava program FONTMAN koji, uz uobičajeni editor znakova, omogućava prenošenje delova jednog seta znakova u drugi, pa čak i komponovanje znaka koji se sastoji od nekoliko drugih znakova — koriste se, jasno, Bulove funkcije AND, OR, NOT i XOR.

## Hardverski glamor

Kupovina nove grafičke kartice nužno donosi određene probleme sa kompatibilnošću: i pored priloženih drajvera, naići ćete na probleme pri instaliranju nekih programa. Najmesniji problem nastaje kada pokušate da pokrenete „SideKick“: ako mu kažete da imate kolor monitor, program će krahirati. Ukoliko, sa druge strane, odaberete monohrom displej, sve će raditi savršeno, a slika će uz to biti u bojil!

Da li će InColor GB222 unaprediti performanse programa? Firma Herkules tvrdi da je skrolovanje bitno ubrzano i da je tekst stabilniji a znaci lepši nego na EGA kartici. Što se oblika znakova tiče, Herkules je verovatno u pravu. Slika je, dalje, izvanredna i stabilna, ali se ni na EGA karticu ne treba žaliti. Skrolovanje je minimalno ubrzano kada radite na bejziku, paskalu ili WordPerfect-u, ali su dobici u PC fort-u praktično stoprocentni!

Veće nevolje će vam pričiniti hardverska kompatibilnost — morate da kupite EGA kompatibilan kolor monitor koji košta bar 500 funti što, uz 400 funti za grafičku karticu (na obe cene tek treba dodati VAT) predstavlja impresivan zbir. Kartica InColor GB222 se, osim toga, pojavila u IBM-ova serija koja afirmiše novi VGA grafički standard koji je, u uslovima približne cene, u svemu superioran Herkulesu. Sve u svemu, utisak koji stičete proučavajući Herkules InColor GB222 podle običnog Herkulesa je sličan utisku koji stičete kada, posle Odiseje 201, gledate Odiseju 2010 — mnogo više sjaja, mnogo više specijalnih efekata, ali i mnogo manje istorijskih ideja!

Dejan Ristanović



# Softver četvrte vrste

**U svojoj izuzetno zanimljivoj biblioteci za Prave Programere „Microsoft Press“ je nedavno objavio zbornik tematskih razgovora sa petnaest najbriljantnijih programera današnjice. U toku nekoliko narednih meseci, u okviru serije „Programeri govore“, prenećemo najzanimljivije intervjue iz ove knjige — sa Garljem Kidalom autorom CP/M operativnog sistema, Džefom Raskinom, vodom projekta „mekintoš“, Džonatanom Saksom, tvorcem „Loutsa 1-2-3“, Bilom Gejtsom, tvorcem bajzika i mnogim drugim živim programerskim legendama. Ima li boljih učitelja da nas uvedu u svet programskih ideja, tajni programerskog zanata i tajni uspeha u programerskom poslu?**

Skot Kim (Scott Kim, rođen 1955) želi — uz pomoć računara — da nas vrati u vreme pre Gutenberga, kada su ilustracije i slova bila jedno te isto... Studirao je matematiku, kompjutersku nauku i muziku na Stanfordu, da bi zatim svu svoju ljubav poklonio grafičkom dizajnu, tačnije — inverzijama. Taj termin Kim koristi da bi opisao umetničku realizaciju reči na taj način da se one mogu crtati iz različitih pravaca, slično anagramu. Početkom osamdesetih je izdao knjigu „Inversions“. Istovremeno je razvio softver četvrte vrste — „Inversions for the Macintosh“. Taj softver, priključen na MacPainta, nudi mnoge vežbe, tehnike i igre za stvaranje inverzija. Za svoju doktorsku tezu sada istražuje radikalno nov dizajn za korisnikov interfejs i — ne želeći da „zakasni na tržištu“ — već je osnovao svoju kompaniju „Look Twice“.

— Šta vas je, gospodine Kim, podstaklo da objavite umetnost inverzija kao posebnu knjigu? U vreme mojih studija na Stanfordu, istovremeno su se dogodile tri stvari koje su mi dale inspiraciju za tu knjigu: počeo sam da radim s računarima, sprijateljio sam se s Daglasom Hofstaterom (Hofstadter), mojim profesorom i mentorom, i upisao kurs za grafički dizajn.

U osnovi je bila kompjuterska muzika koju sam počeo da učim 1975. na muzičkom odelu. Imao sam sreću da budem u dobroj klasi i, što je još važnije, kursjevi su se održavali u stanfordskoj Laboratoriji za veštačku inteligenciju. Laboratorijske se nalazila u podnožju jednog brežuljaka, okružena zelenilom, izolovana od vreve kampa. U toj idealnoj sredini u meni se rodio kompjuterist.

— Dakle, to je istina da ste preko muzike stigli do računara?

U stvari, prvi dodir s računarima sam imao već u srednjoj školi, ali pravo iskustvo počeo sam da stičem na Stanfordskom univerzitetu. Kursjevi kompjuterske muzike su mi omogućili da upoznam najšire mome, koji su uvek bili spremni da vam naizgled objasne na čemu rade. To je divan način učenja.

Ne manje uzbuđujuće iskustvo bilo je i otkriće procesora reči i video-igara, naročito ranih verzija igre rata zvezda. U to vreme ljudi još nisu imali kućne računare, pa nam je Laboratorija bila i dom i škola. Tu čudnu prvu godinu u Stanfordu neću nikad zaboraviti.

— Bili ste opsesivni već od početka... Čini mi se da ste učili sve što se tamo nudilo?

Da, od izvesnih granica. Računarske kurseve sam savladavao kao da sam pomahnitio. Potkrpaj dve godine sam upisao kurs LISP, a drugu godine prijavio se na Knutovu seriju kurseva o



**„Nema fundamentalnih razlika između programiranja na računaru i korišćenja računara... Kad otkucavate svoje ime na tastaturi računara, to je svojevrsno programiranje.“**

strukturama podataka. Savladao sam sve računarske jezike. Tako, kad sam 1975. ušao u svet kompjuterske muzike, već sam bio prekaljen u računarima, ili sam bar ja tako mislio.

— Kada ste dobili ideju da napravite Inverzije? Već na Stanfordu i opet na kursu, jedinom koji je imao veze sa umetnošću, kada sam oviđavao bazičnim dizajnom. Moju pažnju je privukao vizuelni aspekt; oduvek sam bio zainteresovan za vizuelnu igru, naročito za animaciju... Radeći jedan zadatak u okviru kursa, napravio sam prvi inverziju.

Te iste školske godine upoznao sam Daglasa Hofstatera, baš u vreme kad je pisao svoju fascinantnu knjigu „Gödel, Escher and Bach“. Imao sam tu čast da radim sa njim dve godine dok nije završio svoju knjigu.

— Šta ste radili za profesora Hofstatera?

Bila je to saradnja prijatelja. Sedeli smo po celu noć u sobi s terminalom i diskutovali o raznim idejama; onda bismo neke stvari proveravali na računaru... Dok sam mu pomagao da napiše svoju knjigu, sticao sam samopouzdanje

za pisanje sopstvene. Daglasova knjiga je izvanredna, ništa nije sakrio.

— Što se tiče vaše knjige, shvatio sam da je cretze uradio Džon Vornok (Warnock)?

Da, on mi je u tome dosta pomogao. Vornok je sada kod „Adobe“-a, a u ono vreme bio je i moj mentor kod „Xerox“-a.

Za vreme studija bio sam i neplaćeni konsultant kod „Xerox PARC“-a; Stanford je blisko saradivao s njihovom istraživačkom laboratorijom. Prvobitno, nameravao sam da se malo zabavim sa „Metafontom“, jezikom za dizajnerski slog. Međutim, ubrzo sam počeo da kreiram slike na jeziku J&M, prethodniku PostScripta. (J&M = John Warnock and Martin Newell: autori jezika).

Dok sam pravio ilustracije za moje „inverzije“, sinula mi je ideja da koristim J&M za stvaranje slika. Za te dizajne nisam morao da se oslanjam na računar, pa ga nisam koristio bar u dve trećine radova. Ali za neke od tih dizajna — na primer, bajzitoru spiralu — istraživač sam mogućnosti na računaru iz prostog razloga što sam htio da vidim šta će se dogoditi. Pošto mi je programiranje išlo od ruke, nisam imao nikakvih problema. U slučaju bajzitorne spirale, napisao sam program koji je stvarao slova. Najpre sam uradio sve skice na grafičkom papiru, numerisao tačke i sastavio program za reč „infinity“ — saglasno onome kako sam je dizajnirao. Tada sam objasnio Džonu Vornoku šta treba da uradi, pa je on napisao program koji transformiše prave linije u spirale... Od tog vremena mene živo interesuje šta se može prikladno učiniti na relaciji računari-vizuelna ekspresija.

— Čuo sam da sada završavate jedan softver koji treba da nauči ljude kako da stvaraju inverzije? Kako ste razvili taj softver?

Reč je o „Inversions for the Macintosh“. Taj softver ide sa knjigom i obe stvari se prodaju u paketu. Inače, taj softver je koncipiran tako da funkcioniše sa osloncem na MacPaint.

Mogao sam lako da napišem program, ali umesto toga odlučio sam da razvijem poseban softver, što je bila neobična strategija. Morate znati da je po pravilu izbegavam programiranje kad god je to moguće.

— Zašto nismo izlazili programiranje?

Pisanje velikog kompleksnog programa predstavlja jedan indirektni postupak. Ako možemo da učinim na direktniji način, onda dajem prednost takvom postupku. To je za mene izazov: šta bi mogao učiniti na računaru da bi bio interaktivan bez programa?

Inverzije kod „mekintoša“ su nešto što ja nazivam „četvrta vrsta softvera“. Postoji već mnogo ikustvenih primera. Recimo, to su šabloni za apredštovanje a MacPaint uključuje mogućnost montaže. Šabloni nude lepa rešenja za konkretne potrebe. Umetnost montaže je prenet iz jednog drugog medijuma; to je korisno, ali ne otvara puteve za korišćenje nedjuma na jedan doista nov način.

Moj softver je u stvari MacPaintova datoteka i tip slova; stvoren je tačno onako kao što bi i neko drugi firmirao datoteku sa MacPaintom —



kucanjem i crtanjem. Datoteke sadrži vizuelne zagonetke i vežbe, koje, u početku jednostavne, postupno postaju sve složenije. Ali, moji softver je samo polovina MacPainta: ostatak morate sami zamisliti. Praktično, ja postidim korisnike da razvijaju mentalnu fleksibilnost, da zamisljavaju šta će se dogoditi pre nego što nešto učine. Ta veština — predvideti šta će se dogoditi — po pravilu se ne uči u školama. Igraючи se blokovi, ma se kuće, deca se navikavaju da zamisljavaju prostor.

— Pomenuli ste softver četvrtre vrste? O čemu je tu reč?

Softver četvrtre vrste je onaj koji stvaraju kompanije, koristeći konvencionalne programerske jezike. A softver četvrtre vrste stvaraju sami korisnici, bez programiranja, bar ne u uobičajenom značenju te reči; on se nagraduje na softveru treće vrste, i dobro je imati taj softver u bogatom izboru.

Nije danas lako sa softverom; većina je veoma skupa. Dugoročno gledano, cene bi trebalo da padaju, ali se to ipak ne događa. Tržište još nije razvijeno, a troškovi proizvodnje su visoki. A tu su i mnogi drugi faktori...

Zato smatram da je najpешe kod softvera četvrtre vrste što svako ko ima računar može da ga formira. Nagrado, gradite na postojećim programima, a neki su posebno koncipirani za tu svrhu — kao, recimo, "Pinball Construction Set".

— Da li smatrate da će se postupak stvaranja softvera toliko pojednostaviti da će svako ko raspolaže računarnom moći da napravi svoj sopstveni softver?

Naravno, ja smatram da će se u perioda programiranja promeniti. Sada u glavi držite sve moguće ideje i algoritme, pa radite celu noć, pokušavajući da ih grupišete u neku zbirku zagonetki. Kroz nekoliko godina — ja se nadam — biće dostupna većina segmenata potrebnih za konstruisanje programa, bar onih u čestoj upotrebi. Na primer, ako želite da napravite neki svoj program, svratite se u obližnji "Radio-Stack" i kupite nekoliko prefabrikovane delove potrebne za njegovu izgradnju.

Takođe, očekujem da većina ljudi neće programirati na način kako se to sada shvata. Čak i danas nema fundamentalnih razlika između programiranja na računaru i korišćenja računara. To jesu različite aktivnosti, ali one u kontinuitetu na istoj razini. Kad otkucavate svoje ime preko tastature računara, to je svojevrsno programiranje. Drugi tip programiranja — kroz jezike koji su danas u upotrebi — predstavlja pretežno indirektnu akciju. A kad budemo raspolagali progra-

## **"Ja želim računar koji je kao list papira."**

merskim jezicima koji su direktniji, tada će gotovo svako moći da gradi program. To, u stvari, više neće biti programiranje. Niti će jezici biti ono što su danas.

— Pa kako ćete onda nazvati kompjuterske jezike? Ako ih ne priznate kao jezike, da li to znači da će postati nešto poput engleskog?

Promenile se, pre svega, pojmovi koje sada koristimo za opisivanje tih stvari. Vizija Alana Keja (Kay) je da će računari potpuno uspeti kad prestanemo da koristimo reč "computer". On često podseća da se u ranim danima elektriciteta verovalo da će doći vreme kada će svako imati neki električni motor u svojoj kući. Slično vam je danas sa računarnima. Kad postanu dovoljno mali i dovoljno jeftini, računari će naprosto "iskaznuti" u opštem ambijentu. Neće vam pasti na pamet da svom gostu kažete: "Evo, to mi računari!". Jer će oni biti svuda po stanu... Nestaće iz prvog plana i to će biti znak njihovog vrhunskog trijumfa.

— Da li je vaš rad sa inverzijama povezan na neki način s vašom doktorskom disertacijom? Sada radim istovremeno na nekoliko projekta-

ta inverzije su indirektno uključene u moja istraživanja jedino u smislu njihovih uloga u grafičkim slikama prikazanim u nekom kompjuterskom medijumu. Šta se tiče mojih teza za doktorsku disertaciju, one se odnose na grafičko-interfejni dizajn u duhu Alana Keja, ali u sasvim novom pravcu.

Ja se uresudređujem na nešto doista fundamentalno... Pretpostavimo da stajujemo od nule, da odbacimo sve što znamo o računarnima i

**"U to vreme nisam voleo programiranje, jer mi nije padalo na um da ve možda bude drukčije. Sada kada sam u stanju da to zamislim, ja uporno nastojim da programiranje bude drukčije."**

onda projektujemo jedan iz osnova nov na koji će se uzvišenim raditi vizuelno orijentisane osobe. Hoću reći, ja želim računar koji je kao list papira. Ne morate sebe pitati: "Kako da dođem do tekstualnih modula?", "Koji laster da pritisnem" i slično. List papira je krajnje direktan. Ono što vidite na njegovoj površini, to je ono što je, to kod kod računara morate da razmišljate šta se nalazi ispod njega. Računar i jeste kao list papira, samo što ispod ima skrivene žice i hardver.

— Kako pristupate tom problemu — da stvorite računar kao list papira?

Pa, najpre razmišljate o tome šta mi se ne sviđa kod sadašnjih računara. Potom se koncentrišem na grafički dizajn i rad sa slikama, na one što tu najviše volim. A zatim nastojim da to sve stvori približno. Postoje vizuelni umetnici koji rade na računarnima, to je poznato. Većina ih koristi programe za bojenje; ti programi su direktno direktni, u velikoj meri poput lista papira, ali računar dodaje tanana poboljšanja. Lako je zamisliti kako bi se moglo poboljšati. Sve je fluidno i možete učiti da kombinujete slike. Rezultat nikad ne mora da bude konačan.

Muzičari — gledani u celini svoje profesije — nemaju averziju prema programiranju; navikli su da rade sa apstraktnim notacijama, bar oni koji su klasično školovani. Dakle, njima nije stran indirektni postupak. Oni najpre koncipiraju u svojoj glavi, a zatim jele prebacuju na papir. Time se muzičari svrstavaju u labor vizuelnih umetnika. Konačno, ovde nije toliko reč o vizuelnom i zvučnom, već o stilu rada...

Sve u svemu, ja razmišljam o tome šta je u stvari računar i zašto nam se toliko nametnuo. Postoji mnogo problema sa računarnima. Naravno, ja u ovom trenutku nemam rešenja, ali nastojim da računare trenovno preispitam, jer sam uveren da će se ukazati obećavajuće mogućnosti.

— Znači li to da vi razmišljate o računaru koji će koristiti svako, ili o onome koji će biti specifično namenjen umeticima?

Umetnik je deo inspiracije, ali u krajnjem ishodu ja razmišljam o nečemu što bi služilo svakome. Ne postoji jedinstven pravac. Način na koji ja radim uključuje nekoliko projekata koji se donekle razlikuju ali se u nekim tačkama i dodiruju — iz svega toga reba da se nešto radi.

Povrtnim sam nameravao — za koju lezu — da formiram jezik za vizuelno programiranje. Zamisljavao sam da bi on morao biti više direktan. Izgradio sam sićučne projekte da bih testirao specifične odete. Taj jezik — umesto nizova reči — imao bi simbole raspoređene u nešto što bi ličilo na grafikon o kretanju proizvodnje, ili, za sve vreme me je bilo jako osećanje da to što radim nije dobro. Zašto ne bih pisao na paskalu?

Razgovarao sam s mnogim ljudima kod konačno nisam "uhvatio" Larja Teslera (Tessler) koji je u to vreme pisao članak o programerskim jezicima za "Scientific American". Tesler, iskusni programer, radio je nekog na Eplovom računaru "Izla". Dao mi je nekoliko dobrih saveta koji su mi pomogli da shvatim koliko ja u stvari nisam voleo programiranje! Nisam ga voleo jer u to

vremne nije mi padalo na um da ve sve može da bude drukčije, sada kada sam u stanju da to zamislim, ja uporno nastojim da programiranje bude drukčije!

— I šta se dogodilo kad ste shvatili da ve može da bude drukčije?

Postao sam nestrpljiv. Govorio bih poznanicima: "Ne uzimaj stvari zdravo za gotovo, one mogu biti i drukčije." Osećao sam obojnost prema konkretnom programiranju. Znaite, računara je veoma zavodljivo, stalno vas izaziva da učinite još neku stvar... Dobro je znati kad se treba zavustaviti.

— Da li je neko posebno uticao na vas, to jest na ono što sada radite?

To su tri čoveka: Džef Raskin, Dejvid Torenborg (Thorenborg) i Ted Nelson. Sva trojica su davali prednost jednostavnosti, beskompromisno. Na primer, "pet" je najveći broj koji vam je dozvoljen za bilo šta na čemu radite.

Dejvid Torenborg je pisao najčistije. Izdao je jedan mali priručnik — "Zero Mass Design" — čiji je moti: ako hoćete da oblikujete neki softver, napisite knjigu ili se upustite u bilo koji drugi projekt koji zahteva planiranje, počinite od krajnje jednostavnostijednog dizajna. Neka bude toliko uprošćeno — da ne funkcioniše. To zahteva veliku disciplinu, ali dok ne osetite neuspeh, nikad nećete znati deo su granice uprošćenosti.

— Pa, sa "Zero Mass Design" na umu, kako mislite da doćete do tačke kada će računar biti poput lista papira?

Suštnina moje teza je u tome da je takva pretpostavka već uključena u kompjutersku nauku. Bilo mi je potrebno mnogo vremena da to shvatim, jer je posredni nešto veoma suptilno. Ono što vidite na ekranu, to je ono što Alan Kay naziva "korisnikova iluzija". Ekran daje prilično vernu prezentaciju, ali to nije ništa drugo nego struktura podataka iz memorije. Računar je taj koji gleda u podatke, a vi ne; daje vam ovu ili onu sliku na ekranu, a vi onda gledate u njenu i pitate se šta li to računar "misli" tamo iz ekrana. Sistem MacPaint je u tom pogledu veoma upečatljiv, jer ja ličim u nekim slučajevima veoma dobro, to jest bliska stvarnosti.

Svako ko razmišlja o vizuelnom programiranju u stvari ima na umu vizuelno predstavljanje programiranja. Samo programiranje ostaje isto ali ga vi nadgradujete slikom. Rekao bih da je to samo preliv preko torte, jer se suština ne menja. Karika koja nedostaje je tu — između računara i vas. Računar ne vidi ono što vi gledate na ekranu. Kad bi kompjuter postupao sa ekranom tačno na onaj način kako to čini korisnik, interakcija bi bila direktna. Ja bih želeo da

## **"Ja bih želeo da računar misli vizuelno..."**

računari misle vizuelno... Shvatio sam da je to jedna tako čudovišna ideja da sam morao napraviti radni uzorak na "Xerox" — da bih celu stvar obradilo.

— Kako je to moguće da se sve stavi na ekran?

Ne znam ni da li je to upšte moguće. Ali ja ipak sada kreiram jedan veoma uprošćen program koji bi dopustio da bukalvno sve stane na ekran. Pre nego što jedan takav program postane stvarnost, morate imati sredinu u kojoj on može žveti. Ono što sam ja napravio je jednostavan tekst i grafički editor. To je, uglavnom, u istoj kategoriji kao MacPaint, i je daleko uprošćenije. Kod MacPainta, kad otkucate jednu reč, ne možete je menjati; gubite mogućnost redaktura, jer su slova otkucane reči uskiedštena kao bitovi, a ne kao karakteri.

MacPaint predstavlja sliku na ekranu posredstvom bitova koji su tu. Da bi se nešto izmenilo, možete odatle uzeti bilo šta, čak i polovinu slova i pomerati je amo-tamo. Zaboravite da su to slova i s njima ćete postupati kao da su slike, što

je najlepše kod MacPainta. Ali tu je i ono što ne valja: nije moguće poći od bitova na ekranu natrag ka onome što je predstavljalo slova.

— Šta se posliže time što je sve prikazano na ekranu? Da li će računari biti manje kompleksan za korisnika?

Prototip koji ja izgrađujem nije odgovor na vaše pitanje. To je samo korak napred ka krajnjem cilju. Ali ako sada nisam u stanju da pružim odgovor, to ne znači da treba da odustanem, čak ne istražujući mogućnosti...

Kad programirate, vi bezrezervno prihvatate činjenicu da ste u stanju da manipulirate svim onim kompleksnim apstraktnim elementima. Na-prosto, smatrate da drukčije i ne može da bude. Ja u to ne verujem: radiči sa grafičkim dizajnom, shvatilo sam da postoje i drugi načini. Kad koristite neku grafičku dizajnera, vi ne razgovarate sa mnom pokazujete slike ili tražite da vidite njegove — to je ono što je važno. Nije sve u apstraktnim koncepcijama.

— A kako je sa muzikom? Muzika se nikad nije odvajala od apstraktnog postupka. Neće valjda notacije iščuznuti?

Pa, većina muzičara ne koristi note. Ja lično sam otkriven kao muzičar zbog svog akademskog obrazovanja; otkriven sam strojnim pravilima muzičke teorije, tehnik i notacije. Nisam u stanju da improvizujem što mnogi mogu. Ja im zavidim.

Početkom pedesetih godina, Grejs Hoper (Hopper) — majka programskih jezika visokog nivoa — govorila je da ljudi ne treba da pišu skovanim jezicima, već pre nekim koji je poput engleskog. Odgovarali su joj da računari nije u stanju da razume engleski, to je naprosto nemoguće. Ali, stvar nije u tome. Tu se radi o kompromisu između onog što računari može da razume i onoga što je čovek u stanju da razume.

„Računar je veoma zavodljiv, stalno vas izaziva da učinite još neku stvar...“

Želio bih da ljudima omogućim da razgovaraju sa računarima putem nečeg što je blisko slikama. To ne znači da sam protivnik apstraktnih notacija. Ja samo želim da uključim red i simbole kao specijalnu vrstu slika. Da bi odgovorilo tom zahtevu, računari mora da bude osposobljen za prelaz od bitova na slova, da postane nešto kao sistem za prepoznavanje optičkih znakova.

— Kakve veze imaju „inversions“ s tom idejom?

Pa da vidite — i postoji jedna lepa veza. Za svoju knjigu je karakteristično da nije ozbiljna. U stvari, ona je luckasta. Kao neka igra, a iz Igranja dolazi mnogo novih ideja. Ali ta knjiga nameće i jedno pitanje: „Je li ovo predstavlja sliku ili reči?“ Teško je na to odgovoriti. To je pomalo i jedno i drugo. Mnogi zaboravljaju da su te male crte i bele tačke na stranicama knjige rezultat nečijeg crtanja. Slova su forme — to se gubi iz vida a vi jedino zapazate da su nekako drukčiji a ona su takva jer je način na koji se stvaraju — drukčiji.

Pre Gutenberga, ilustracije i slova bili su jedno te isto: nerazdvojlivi. Kasnije, nastale su dve discipline koje su se razvijale u posebnim pravcima. Sada kada smo dobili „mekinotš“, ja vidim medijum koji ih ponovo spaja. Kod MacPainta ne postoje razlike između redi i slika.

Slova u našem alfabetu začela su se kao crteži. Te forme napravljene su ljudskom rukom. One nisu stihle niotkuda i same se uklesale u kamen. Tokom mnogih stoleća te forme su se menjale i evoluirale. Važno je da shvatimo — ili da se posvetimo — da je sve notacije, muziku, naš jeziki li računarske jezike napravio čovek. To su simboli koje je moguće menjati. Uvek postoji izbor. Ta sposobnost promene notacije ili simbole daje silnu snagu ljudskim bićima.



## Iz života u legendu

Prelistavajući stručnu literaturu, otkrili smo zanimljiv filološko-komputerski rad izvesnog dr Muada Konjhotički pod naslovom „Komputerske bajke i legende prigradskih neetičkih grupacija“.

Taj do sada nedovoljno poznati rad otvara nove pravce za tumačenje odnosa širih pa i najširih narodnih masa prema kompjuterima i sličnim napravama.

Ovde prenosimo samo najzanimljivije i najinformativnije teze.

Po autoru, u severnim predgradima Beograda postoji razvijena narodna priča o tome da svi pravi kompjuterski stručnjaci završavaju svoj buran život ili u ustanovama za duhovni oporavak ili u prosvetnim savetima.

U ljubljanskom prigradskom naselju Džiberšćici ukorenjena je domaća legenda o tome da je SIDA nastala pod uticajem raznih faktora, od kojih je najznačajniji zračenje sa kompjuterskog ekrana.

U prištinskom naselju lduku postoji uvreženo narodno predanje o tome da rad sa kompjuterima izaziva smanjenu potenciju i hroničan nedostatak želje za polnim opštenjem i produžavanjem vrste.

Interesantno je da u pojedinim krajevima Novog Beograda i Banja Luke kruži ista priča o nekom izmišljenom kompjuterskom stručnjaku koji je bio frustriran u svom malom i glupom računarskom centru neke u našim krajevima i onda je lepo otišao u Ameriku i tamo postao milioner. Programirajući.

## Meteorološki izveštaj

Sa početkom ovog meseca može se očekivati da će na severu zemlje doći do povremenog uticaja bitnog, dok će u svim ostalim delovima biti izrazito bajno. Povremeni prodori osvetljenja mogu sasvim lako dovesti do sporadičnih bogovanja u višim predelima, kao i u zatvorenim kotlinama. Ne treba se iznenađivati ni povremenim linkovanjima, iako ih ne treba očekivati. Ako u drugoj polovini meseca ne dođe do značajnih promena u oštroj situaciji, može se očekivati pojačan uticaj antibazikone sa zapada, kao i nestabilnih novopirivanja sa istoka.

Na žalost, prosečna kairanja će biti oko 64K, iako se mogu očekivati pojedini udari i do 512K.

Detaljnija prognoza u regionalnom izdanju Peek & poke show-a.

U pojedinim manjim gradovima istočne Srbije postoji legenda o tome da najbolje računare na svetu nemaju nadobudni Amerikanci nego Rusi, koji duboko u Sibiru imaju najbolji kompjuter na svetu, samo neće da ga pokažu da bi zavarali sve živo.

U još manjim gradovima istočne Srbije postoji legenda o tome da najbolji računari

na svetu nemaju ni Amerikanci ni Rusi nego ga imamo mi. Dobro je sakriven i čuvan i zato se pravimo kao da smo toliko glupi da ne bismo umeli da napravimo dobar računari — da neko ne bi nasluo da ga imamo.

U skoro svim većim gradovima postoji identična legenda o tome da prvi interaktni ne uče programiranje. Ona se logički nastavlja na narodno predskazanje o tome da treba samo malo sačekati, jer će se uskoro stvari rešiti same od sebe.

## Važno, tiče se bezbednosti i love

Zahvaljujući informacijama koje smo dobili za nedavni Dan Bezbednosti, napredovanje računara u našem društvu prati i kriminal. Kako saznajemo, računarski kriminal je od prošle godine svoj obim povećao za preko 200%. Te zabašujuće cifre govore same za sebe.

Znači, od prošlogodišnjeg jednog slučajaja sad imamo čak tri komada sa najavom deli da eskalacije, pa se može sa sigurnošću tvrditi da će sledeće godine biti i pet pa i šest slučajeva kompjuterskog kriminala.

Prošlogodišnji slučaj, kada je Gaga Minčić iz Beograda svom tastu „komodorom“ razbio lovu, brzo je zataškano, da ne bi bacao loše svetlo na računare.

Ovogodišnji slučajevi su mnogo komplikovaniji. Kao prvo, tu je velika afera „Frka“, u kojoj su se Direktor računskog centra Radović i samostalni projektant sistema ing. Jovan Simić, vatreni navijači Zvezde i Partizana, u sred računskog centra gušali i razmenjivali mučke ali neveste udarce.

Onda, tu je monstruozna afera Kajneć—Marković, koja je nedvosmisleno pokazala da je ljubomorni suprug pokušao da umori svoju nevernu suprugu podmećući joj da se igra na „spektrumu“ po kojem je prethodno prolio 7,4 litara vode. Na svu sreću, Olga Kajneć—Marković nije preterano ljubila „Impossible mission“, pa je ostala živa.

Poslednji i najtragičniji slučaj odigrao se nedavno. Izvesni Z. M., po zanimanju provanik, je u centru Zemuna razbio jedan izlog i iz njega ukradio strano-domaći računarski frafciger porekla. Tužno delo bi samim tim bilo već dovoljno veliko, ali je jadni Z. M., kad je ustanovio šta je ukradio i koliko taj kompjutor vredi, pokušao da izvrši samoubojstvo, što mu nije uspelo ali je zato došao u ruke pravdi.

Kao što vidite, kompjuterski kriminal je u porastu. Čuvajte se!!! Pogledajte oko sebe! Ako vidite nešto sumnjivo, javite nam.

## Informatička (ne)pismenost

Na temelju ankete utvrđeno je da najmanje problema s informatičkom pismenošću u nas imaju studenti, slijede srednjoškolski, a u osnovnoj školi je još lošija situacija. Najzaostajali smo po tome u privatni.

Slikovito rečeno, u SAD 85 posto rukovodećeg kadra usvojilo je osnovne elemente informatičke pismenosti. Kod nas, u najboljem slučaju, može biti obratno.



Nastavljajući sa svojom humanističkom koncepcijom (ta reč znači začeo, ali neka vas to ne zbuni) Peek & poke show je pronašao još jedan način da iskoristi potencijal koji kompjuteri imaju za poboljšanje i uljepšavanje ljudskih života. Tako već od sledećeg meseca preko naše specijalne Daš-Dam kompjuterske mreže počinje sa radom Peek Agencija za kompjuterska brakove. Mi smo spariti nesparene mlade ljude pomoću sofisticiranog kompjuterskog programa razvijenog u našim laboratorijama, a sad svetski poznatog pod imenom Daš-Dam Very Soft. Pozivamo sve mlade kojima je baš ovo potrebno da nam se jave preko modemske veze.

### Mali, malečki, ovolišni dopis

PA Marvin, koji nam sa svojih cunja-nja po tajanstvenom kontinentu Severne Amerike redovito šalje poverljive izveštaje, ovog puta nas je obradovao ovim malim dopisom koji govori o životu mladanih Amerikanina.

„Najnovija izreka u bogatom reper-tuaru njihovih sarkazama označava vreme poznato kao  $\pm$  nikada, a izgovara se

kao: „Kad u Jugoslaviji naprave kompjuter“.

Koliko je nama poznato, našem ambasadoru u Ujedinjenim Nacijama naloženo je da oštro protestuje zbog ovog izrugivanja našoj nemoći.

Marvin se i dalje cunja, samo je sve depresivniji, pored ovakvih viceva.

Na zahtev frustriranih čitalaca Samoupravna redakcija PEEK & POKE SHOW-A RASPIŠUJE

posle duže pauze još jedan VELIKI NAGRADNI KONKURS na temu „Kako biste vi sprečili da jugoslovenska deca uče programiranje?“

Konkurs je javan i registrovan kod jugoslovenske zajednice za javne skupove i fudbalska prvenstva. Uzimaće se u obzir samo prilozima na programskim jezicima naroda i narodnosti Jugoslavije. Samo u nedostatku boljih oblika prima-će se i likovni i muzički radovi.

Kao i uvek, pripremili smo vredne i obilne nagrade koje ćemo podeliti samo najboljim učesnicima i našim rođacima.

Prva nagrada je simbolična i neobično vredna. Maštoviti čitalac koji je dobi- (čitateljke?) imaće pravo da krsi računar koji će deca ubuduće koristiti u školama. Zato molimo sve koji učestvuju na konkursu da odmah dostave svoj predlog naziva za ljubimca budućih razreda.

Druga nagrada nije simbolična i sa- stoji se iz primanja srećnog recipijenta (to znači primaoca) za stručnog saradnika u cenjeni časopis „Računari“. Njemu će u „Računarima“ godinu dana biti isplaćivan prosečni honorar stručnog saradnika, pod uslovom da ništa ne piše i ne objavljuje.

Treća nagrada je sitna, polusimbolična i skoro ilegalna. Saostoji se iz kućne adrese Saveznog sekretara za unapređenje računarstva i poskupljenja cigareta. Šta će dobitnik uraditi sa ovom nagradom ne znamo i ne tiče nas se. Pored ovih glavnih nagrada, pripremili smo i pravu gomilu utešnih nagrada koje bi trebalo ako ne da zadovolje učesnike ono bar da ih nateraju da nas manje mrze.

Adresa za slanje priloga je kao i uvek

Peek & poke show  
Računari  
Galaksija  
BIGZ  
Bulevar vojvode Mišića 17  
Beograd

### Mi o Vuku...

Svetska kompjuterska asocijacija je prilično nepovoljno reagovala na predlog jugoslovenskog ogranka da se povodom velike Vukove godišnjice koja se slavi u našoj zemlji ove godine u sve kompjutere u svetu uvede i čirilični set znakova. Kako saznajemo, odgovor je jedno kurtoazno „Ne!“.

### Ko doživi, pričaće

Radujte se i veselite, pucajte i pevajte, spremamo nešto lepo za vas. Od jeseni ćete svi moći da znate šta to pišu ti čudni i nerazumljivi ljudi po raznim kompjuterskim časopisima. Da, tako je, od jeseni Peek & poke show priprema za vas svoj „Mali rečnik malo, još manje i nimalo poznatih računarskih termina“. Tad će se biti objašnjeno.

### Zen je stigao do nas

Hip hip hureji! Ponovo smo najbolji. Na legendarnom konkursu časopisa „Ubijam kompjuter njuz“ jedna naša igra je dobila prvu nagradu u kategoriji najracionalnijih igara.

Naša igra je zasluženo bila hit čitavog konkursa. Pod produkcjskom palicom tima „Da da gu gu“, igru je uradio Mr Štrok i to na nadasve originalan način.

Da se ne bi remetila koncentracija igrača, muzika je izbačena. Takođe, da se igrač ne bi nepotrebno opterećivao spoljnim efektima, igra je urađena u samo crno-beloj verziji. Iz istih razloga je scenario radikalno pojednostavljen, pa se igra sastoji samo iz jednog ekrana. Originalno rešenje predstavlja i to što su u cilju konciznije priče ukinuti ograničavajući elementi, kao i neprijatelj. Nakon takog radikalnog čišćenja igre, bilo je sasvim logično što je određeno da figura glavnog junaka bude nepomična. Trenutno se razmišlja o novoj verziji ovog programa, koji bi bio isti kao stari, osim što bi bila izbačena figura glavnog junaka. Uglavnom zato da ne bi remetila koncentraciju.

### Rule Yugoslavia

Nasi specijalizovani računarski udžbenici doživljavaju pravi bum u belom svetu. Nekoliko njih se upravo prevodi u Americi, a jedan od stručnijih je izšao u specijalnom prevodu i opremi pod naslovom „Šnupi i kompjuteri“. Pretpostavlja se da će imati velikog uspeha kod čitalačke publike tog uzrasta. Saznajemo da se prema specijalno izdanje iste knjige za one koji još ne umeju da čitaju.

### Glavice kupusa

Radionica za egocentrična i mentalna istraživanja Peek & poke show-a oličena u vrhlin stručnjacima dr Šnuku i Šiksi najzad je došla do nepobitnih dokaza da intenzivno igranje video igara ne samo da oštećuje moždane vijuge i time dovodi do vegetiranja njega (što je preveđeno u analiza patološkog kompjuterisiranja) i radikalno menja lični opis igrača u skladu sa teorijom o vegetiranju.

Dokumentarni snimak koji dokazuje ovu našu prastaru tezu je načinjen u jednom novobogradskom domu posle seanse od 73 sata neprekidnog igranja za uludnih i zaglupljujućih igara. Mi smo vas opomenuli. Sad neka vam sam interpret pomogne.



# Ekspertni sistemi za početnike

*Krajem sedamdesetih i početkom osamdesetih godina iz veštačke inteligencije na tržištu se moglo naći jedva ponešto softvera (pri tom na Balkanu, naravno, bez prapratne dokumentacije) i svega dva udžbenika za prvo čitanje, od kojih je jedan bio lošiji od mnogih priručnika, i, što je najgore, u ovim krajevima niko o veštačkoj inteligenciji nije znao skoro ništa. Sada nam je na raspolaganju više stotina knjiga, dosta časopisa, programskih jezika prilagođenih raznim mašinama, softverskih alati za razvoj ekspertnih sistema, mnogo raznovrsnih kurseva (čak i na video trakama) i raznih drugih stvarčica koje život čine udobnijim.*

Najbolji način da uđete u novu oblast da prvo pogledate da niste možda već stigli tamo: jednostavno treba apsorbovati najnovije trendove i ideje dotične oblasti i to, po mogućnosti, iz prve ruke. Dakle, nastojte da što brže saznate „ko su glavne zveri“, a potom čitajte njihove radove i stupite u direktniji kontakt sa njima.

Brz uvid u oblast dobićete ako se učlanite u Društvo za veštačku inteligenciju pod nazivom American Association For Artificial Intelligence (AAAI), koje četiri puta godišnje izdaje časopis „The Artificial Intelligence Magazine“, kao i godišnji katalog sa imenima i adresama preko 4000 istraživača u oblasti veštačke inteligencije. Članovi društva imaju i razne finansijske povlastice prilikom učestvovanja na konferencijama i prilikom preplate na mnoge stručne časopise. AAAI društvo se javlja i kao sponzor, ili jedan od sponzora, velikog broja tehničkih skupova.

## Veštačka inteligencija za direktore

Komercijalno nastrojenim osobama (da ne kažemo baš trgovcima) preporučujemo da počnu od knjige „Expert Systems: A Management Guide“ koja se na zahtev može dobiti od PA COMPUTERS.

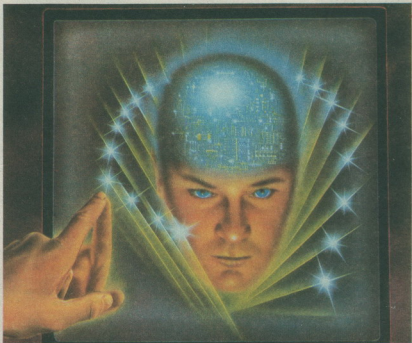
Nakon toga, dotični bi mogli da se zainteresuju za trenutno stanje na tržištu. Spomenućemo pet žurnala koji veću pažnju posvećuju novitetima na pijaci veštačke inteligencije nego situaciji u laboratorijama.

Časopis iz primenjene veštačke inteligencije „Applied Artificial Intelligence Report“ u izdanju Instituta za inteligentne računarske sisteme sa Univerziteta u Majamiju ima akcenat upravo na aktivnostima na polju biznisa. Kao vrlo bitan momenat u korist ovog časopisa reći ćemo još i to da je od svih pet koje predlažemo biznismenima ovaj najjeftiniji.

Artificial Intelligence Research Services izdaje časopis „AI Mass Market Newsletter“. U časopisu, na vrlo osoban način, čuveni ekspert za nove trendove računarstva Carl Sipe (Charles Sippl), daje vlastito mišljenje o novim proizvodima.

Za časopis „Knowledge Engineering“ ili, što bi se reklo, Inženjerstvo Znanja, u izdanju Richmond Publishing, smatra se da ima najkvalitetnije članke sa najobjektivnijim i najdužim ekspertizama proizvođača.

Vrlo interesantan je i časopis „The Spang Robinson Report“ kuće Spang Ro-



binson. On pokriva oblasti za koje se ostali veliki izdavači nisu preterano zainteresovali: poput veštačke inteligencije na IBM PC računarima, ili, na primer, primene paralelizma u veštačkoj inteligenciji. Njihova specijalnost je i da detaljno prate napredak na polju veštačke inteligencije u Japanu.

Ekspertni sistemi, podoblast veštačke inteligencije koja je već postigla zapažene rezultate i time na velika vrata ušla u svet biznisa, imaju i specifične menadžerske časopise. Jedan od najpoznatijih takvih strateških časopisa je, svakako, „Expert Systems Strategies“, u izdanju kuće Cahners. Za razliku od prethodnih časopisa, ovaj je razumljiv i laicima koji ne znaju baš sve tajne trgovačke struke.

Pored pomenutih časopisa, s vremena na vreme se pojavljuju i sjajni vodiči u svet biznisa veštačke inteligencije. Neke od njih su napisali ljudi koji se jako dobro razumeju i u tehnologiju i u zakonitosti tržišta, pa su kao takvi na naročitoj ceni. Da samo

pomenemo neke od takvih priručnika: „The AI Business: The Commercial Uses of Artificial Intelligence“ u izdanju MIT Press, čiji su autori Patrick Henry Winston i Karen A. Prendergast; ili „How to Make „Real“ Money with Artificial Intelligence“ u izdanju Artificial Intelligence Research Services (AIRS), čiji je autor Charles Sippl. Poslednja izdavačka kuća je i najveći izdavač specifičnih studija za primenu raznih podoblasti veštačke inteligencije na tržištu.

Konačno, polu-tehničkim menadžerima kojima treba jednostavna osnova veštačke inteligencije dobro mogu da posluže knjige Miskoffa i Sippla koje navodimo u nastavku teksta.

## Veštačka inteligencija za programere

Programerima koje žele tek samo da se dobro informišu na tehničkom nivou dobro će doći časopisi poput novog magazina „AI Ekspert“ iz kuće CL Publications. Radi se o



časopisu pisanom u popularnom stilu, baš poput starijeg izdanja iste kuće „Computer Language“. Za prvo čitanje odlična je i knjiga Han Miskoff „Understanding Artificial Intelligence“. Realno je očekivati da će čitalac koji prvi put ulazi u oblast imati i terminološki problema. Iz tih razloga nije loše imati pri ruci i specijalizovani stručni rečnik. Preporučujemo Charles Sippl „Expert Systems / Natural Languages Dictionary“ u izdanju Artificial Intelligence Research Services.

Programerima koji žele da se ozbiljno bave veštačkom inteligencijom trebaće dosta tehničkog osposobljavanja. Najbolji način da se takve veštine steknu su intenzivni seminari. E, tu su Balkanci u lošem položaju!

Naime, u svetu je praksa da takve seminare drže firme koje inače prodaju alkatke za razvoj softvera veštačke inteligencije. Običaj je da firma kupcima održa kurs o kupljenom proizvodu, kao i o mogućim načinima upotrebe. Kod nas nije baš ustaljena praksa da se softver kupuje, a proizvođači iz sveta su uglavnom tako okružni da sa piratima uopšte ne žele da imaju posla, a kamoli da ih obučavaju upotrebi ukradene robe!

Moraćemo, dakle, da počnemo i sa kupovinom softvera, ili da alternativno ubedimo neke majstore da nas obučavaju čisto volenterski! Postoji i treća, sirotinjska mogućnost: da skupimo nešto malo dolara i pošaljemo nekoliko naših hakera na „licino mesto“, u Ameriku, sa zadatkom da „pokradu“ neka znanja, a potom da obuču i široke programerske mase Balkana. Pretpostavljam da bi se dosta toga dalo naučiti i u seminarima koji traju relativno kratko. Na primer, na seminaru o projektovanju ekspertnih sistema „Building Expert System“. Ovaj seminar je u veoma dobrom glasu, a drže ga momci iz Software A&E, kompanije koja proizvodi ljuske ekspertnih sistema. Nesumnjivo da bi još veća korist bila od naših ljudi koji su doktorirali na nekom od eminentnih svetskih univerziteta i žele da se vrate u Jugu, ali ih, bar zasad, ovde nerado dočekuju (prirodno, jer ljudi mnogo znaju, pa ovde mogu samo da smetaju!)

### *Veštačka inteligencija za pirate*

Seminari su, dakle, vrlo pogodan način da se brzo ude u svet veštačke inteligencije. U našim uslovima to, doduše, nije lako izvodljivo, ali nikad se ne zna ko ima dolara u izobilju. Ipak, mislimo mi i na one koji su se svojevremeno obogatili preprodajom softvera, a sad su se uozbiljili i žele da rade prave stvari!

L, kite, evi i informacije za bivše pirate a sadašnje milijardere željne znanja. Amerikanci imaju i mnoštvo „instant“ seminara — traju svega dva do četiri dana u zgusnutom obliku, a cena je za dobro situiran svet sasvim prihvatljiva. Nismo bili, ali rekoše nam da su odlični seminari firme Integrated Computer Systems. U pitanju su tri kursa pod nazivima „Knowledge-Based Systems & Artificial Intelligence“, „Programming in LISP and PROLOG“, i „PC Expert System Shells“. Pored pomenute firme, od kojih kuća institut U.S. PROFESSIONAL DEVELOPMENT INSTITUTE ima kvalitetne kurse-

ve „Knowledge Engineering“ i „Programming Expert Systems in PROLOG“, a Institut za napredne tehnologije (Institute For Advanced Technology) firme Control Data drži „Artificial Intelligence and Expert Systems“, „Knowledge Engineering for Expert Systems Workshop“, „LISP programming Workshop“ i „PROLOG programming Workshop“.

Programerima koji nemaju „vremena“ da putuju po seminarima i da na licu mesta pokupe nova znanja, ne preostaje ništa drugo nego da čitaju kvalitetnu literaturu. Dobar uvid u ideje veštačke inteligencije može se dobiti iz knjige G. L. Simmons „Introducing Artificial Intelligence“ (izdanje NNC Publications). Sličan uvid, dopunjen i delom o načinu pisanja programa veštačke inteligencije, čini kostur, verovatno najviše citirane knjige iz ove oblasti. Radi se o knjizi Patrika Winstona „Artificial Intelligence“ u izdanju kuće Addison-Wesley. Čitalac koji želi sličan uvid, ali na široj osnovi, verovatno će biti zadovoljan priručnikom o tri toma pod naslovom „The Handbook of Artificial Intelligence“. Ovaj priručnik delo je čitave grupe autora, a urednici su Avron Barr, Edward A. Feigenbaum i Paul R. Cohen.

### *Veštačka inteligencija za eksperte*

O ekspertnim sistemima na tržištu verovatno postoji više knjiga nego o svim ostalim podoblastima veštačke inteligencije zajedno. Većina takvih knjiga ne zahteva neko veliko predznanje. Jednu od knjiga koje objašnjavaju kako ekspertni sistemi funkcionišu i koja se, uz to, vrlo lako čita izdala je kuća John Wiley and Sons. Autor ove knjige, koju preporučujemo kao lako štivo pred spavanje, je Peter S. Sell, a knjiga se zove „Expert Systems — A Practical Introduction“. Kod knjiga poput pomenute je nezgodno što čitaocu daju bolju predstavu o ekspertnim sistemima, ali mu glavne „cake“ i dalje ostaju mutne. Što je još gore, ovakve knjige ne daju ni osnovna znanja o načinu projektovanja ekspertnih sistema.

Specijalno za hobiste ista kuća izdala je i knjigu C. Naylor „Build Your Own Expert System“. U ovoj knjizi autor nas uči baš ono što u prethodno pomenutoj knjizi nedostaje. Akcenat je na tehnikama projektovanja ekspertnih sistema. Pri tom je sve napisano jednostavnim jezikom, primerenim upravo hobiistima.

Spomenimo još dve knjige iz grupe onih koje više objašnjavaju kako ekspertni sistemi funkcionišu nego što ude čitaoca kao da ih i sam pravi. Paul Harmon i David King su napisali vrlo interesantnu knjigu „Expert Systems: Artificial Intelligence in Business“ u kojoj čitalac, pored načina rada ekspertnih sistema, može pročitati i mnoštvo podataka i spekulacija o potencijalnom tržištu za ekspertne sisteme. Da nas neka od izdavačkih kuća ne bi proglasila pristrasnim kritičarima, nećemo vam reći da je i ova knjiga izašla u izdanju John Wiley and Sons. U izdanju Addison-Wesley izašla je knjiga Donald Watermana „A Guide to Expert Systems“ III, što bi se kod nas reklo, vodič kroz ekspertne sisteme. Radi se o vrlo katalogiziranoj knjizi. Tu nalazimo katološki sredene: softvere za razvoj ekspertnih sistema; postojeće ekspertne sisteme; termi-

nologiju ekspertnih sistema; kompanije koje prave ekspertne sisteme i razne slične korisne informacije.

Knjiga „Developments in Expert Systems“ čiji je urednik M. J. Coombs, a izdavač Academic Press, bavi se nekim od najnovijih rezultata na polju ekspertnih sistema. Pretpostavka je da čitalac poznaje ekspertne sisteme od ranije prilično dobro. Oni istraživači koje interesuju najnoviji rezultati u veštačkoj inteligenciji, baš kao i u drugim naukama, upućeni su na časopise, monografije i konferencije.

Kad su u pitanju časopisi, tu je Elsevier, odeljenje North-Hollanda, nenadmašan. Oni publikuju, između ostalog, i časopis „AI: Artificial Intelligence“ koji se smatra najvažnijim opštim časopisom iz veštačke inteligencije. Kuća izdaje i više usko specijalizovanih časopisa (na primer za fazi skopve).

Ugledni časopisi su preopterećeni količinom radova koje im autori šalju, tako da iz tehničkih razloga od prijema naučnog rada pa do publikovanja prođe dosta vremena, pa su i rezultati po časopisima relativno zastareli. Najsvježiji radovi izlazu se na konferencijama, pa je za istraživače vrlo korisno da se nađu na takvim skupovima.

U korist takvih izleta u svet teško je ubediti naše direktore i poglavare republičkih zajednica nauke. Jedno od kompromisnih rešenja je čitanje zbornika radova (proceedings) sa važnih konferencija. Od kuće Norgan Kaufmann moguće je kupiti skoro sve najvažnije zbornike radova sa konferencija posvećenih veštačkoj inteligenciji. Ruku na srce, takva izdanja nisu baš jeftina, ali sigurno da su daleko jeftinija od troškova boravka na konferencijama.

### *Veštačka inteligencija za spektumovce*

U svetu su u poslednje vreme vrlo popularni računari specijalizovani za veštačku inteligenciju. Na žalost, računalske veštačke inteligencije (npr. TI Explorer firme Texas Instruments) Regan čuva kao oči u glavi, (njihov izvoz je pod zabranom) pa su nedostupne i u našim krajevima.

Srećom, veštačku inteligenciju moguće je razvijati bez specijalizovanog hardvera. Sasvim je uobičajeno da ljuske ekspertnih sistema rade na računarsima opšte namene. Primera radi, spomenimo da čuveni Rule Master firme Radian može da radi praktički svuda gde imamo Unix operativni sistem.

Dakle, na vama je da prema vlastitim potrebama i dostupnom vam računaru na tržištu pronađete adekvatne razvojne alkatke veštačke inteligencije. Jasno, spektumovci ne mogu očekivati da će pronaći profesionalni razvojni softver za ekspertne sisteme namenjen njihovoj mašini. Ako vam trenutno nije dostupan neki VAX, ne gubite nadu! Amerikanci za svega desetak hiljada dolara prodaju moćne personalne namenjene profesionalnim istraživačima na polju veštačke inteligencije. Još samo da Tajvani uzmu stvar u svoje ruke i to pravo razvojno sistema u vašoj sobi i, to, što je najvažnije, za male pare. U međuvremenu, dok čekate na Tajvanec, uradite svoj deo posla — teorijski se dobro potkujete!

Zoran Obradović

# Matemaričar gvozdenog kova

**Mnogi vlasnici PC kompatibilnih računara razmišljaju o aritmetičkim koprocesorima — svi znamo da se uz njihovu pomoć strahovito ubrzava rad sa racionalnim brojevima, što se blagotvorno odražava na razne CAD pakete, kompajlere i mnoge druge programe. Nevolja sa aritmetičkim koprocesorima je što su skupi — da li će se pedesetak funti uloženi u 8087 uopšte isplatiti? Zbog toga smo odlučili da u ovim i sledećim „Računarima“ detaljnije upoznamo performanse Intelovih aritmetičkih koprocesora; osnova za napis je tekst Stivena Frieda The 8087/80287 Performance Curve koji je objavljen u jednom od specijalnih izdanja renomiranog časopisa Byte.**

Čip 8087 je specijalni 80-bitni koprocesor koga proizvodi isključivo Intel (u stranoj literaturi ćete pronaći termin *solesource* koji znači da neka firma ima monopol na proizvodnju nekog čipa). Opremljen je sa 80-bitnih registara u koje se, pod kontrolom mikroprocesora 8086 odnosno 8088, upisuju racionalni brojevi. Koprocesor ove brojeve može da sabira, oduzima, množi, deli, korenuje i poredi, pri čemu se svaka od ovih operacija obavlja stotinak puta brže od odgovarajućeg solidno napisanog računarskog programa za osnovni mikroprocesor. Čip 8087, uz to, može da se programira za izračunavanje transcendentálnih i trigonometrijskih funkcija, pri čemu je faktor ubrzanja u odnosu na normalne programe približno 50.

## Nežan kao maslačak

Ovakvo unapređenje performansi očito ima svoju cenu — 8087 ima gotovo tri puta više tranzistora nego 8086! Veći broj tranzistora označava veću potrošnju, pa će aritmetički koprocesor na našem PC-ju disipirati 1,5 W, što će se, jasno, odraziti na njegovo zagrevanje — temperatura keramičkog kućišta je tipično 45–55 stepeni Celzijusa. Ukoliko je koprocesor okružen čipovima koji dosta troše i ukoliko je ventilator slab, ova temperatura može da poraste i izade izvan specifikacija, što daje izuzetno neprijatne rezultate — pri računarnju se gubi po neki bit i to na nepredvidljiv način! Dalji porast temperature dovodi do kraha čitavog sistema, što znači da je kvalitet izrade najvažnija karakteristika koju aritmetički koprocesor treba da zadovolji — verovatno zbog toga samo Intel proizvodi ove čipove i verovatno su zato oni toliko skupi. Zanimljivo je da PC-jev BIOS ne sadrži rutinu koja testira aritmetički koprocesor, što znači da će prisustvo feleričnog primerka izazvati greške koje ni na koji način neće biti prijavljene!

Čak i renomiranoj firmi poput Intela proizvodnja aritmetičkih koprocesora donosi mnogo problema — procenat škarta je 85 odsto, što znači da tek svaki sedmi čip prolazi stroge testove i izlazi na tržište. Čipovi koji prođu testove bivaju podeljeni u dve kategorije u zavisnosti od kloka koji mogu da podnesu — 5 MHz odnosno 4 MHz. Maksimalna temperatura prema specifikacijama ne sme da pređe 70 stepeni.

Karakteristike aritmetičkih koprocesora, međutim, nisu veće — svake godine se 20 računari 29 • avgust 1987.

## Generacije Intelovih aritmetičkih koprocesora

HMOS 1			HMOS 3		
Oznaka	Clock	Temperatura	Oznaka	Clock	Temperatura
C8087-4	4	70	C8087	5	70+
C8087-6	4.77	50	C8087-2	8	70+
C8087-3	5	70	C8087-1	10	70+
C80287-3	5.33	70	C80287-8	8	70+

### Slika 1

pojavljivala po neka nova generacija koja se odlikovala manjom disipacijom snage, većom brzinom ili nižom cenom. Poslednji važan događaj se zbio tokom 1985. kada su promene bile toliko velike da je Intel zaklju-

čio da čip zaslužuje novo ime — tako je nastao 80287 koji se standardno ugrađuje u IBM PC AT. Slika 1 prikazuje oznake Intelovih aritmetičkih koprocesora raznih genera-

- Aritmetički („savage“) test — bejzik
 

```
10 A=1
20 FOR I=1 TO 2499
30   A=TAN(ATAN(EXP(LOG(SQR(A*A)))))+1
40 NEXT I
```
- Aritmetički („savage“) test — fortran
 

```
A=1
DO 10 I=1,2499
   A=TAN(ATAN(EXP(LOG(SQRT(A*A)))))+1
10 CONTINUE
```
- Aritmetički („savage“) test — Lotus (čellia A21)
 

```
5 TAN (5ATAN (5EXP (5LN (5SQR (A20*A20))))+1
```
- Zajednički element — fortran sa jednodimenzionalnom matricom
 

```
DO 10 N=1,1000
DO 10 I=1,100
  W(I)=Z(I)+X(I)*Y(I)/Z(I)
  R(I)=W(I)+Z(I)*Y(I)
10 CONTINUE
```
- Zajednički element — fortran sa dvodimenzionalnom matricom
 

```
DO 10 J=1,1
DO 10 N=1,1000
DO 10 I=1,100
  W(I, J)=Z(I, J)+X(I, J)*Y(I, J)/Z(I, J)
  R(I, J)=W(I, J)+Z(I, J)*Y(I, J)
10 CONTINUE
```
- Zajednički element — fortran sa trodimenzionalnom matricom
 

```
DO 10 K=1,1
DO 10 J=1,1
DO 10 N=1,1000
DO 10 I=1,100
  W(I, J, K)=Z(I, J, K)+X(I, J, K)*Y(I, J, K)/Z(I, J, K)
  R(I, J, K)=W(I, J, K)+Z(I, J, K)*Y(I, J, K)
10 CONTINUE
```
- Megalopolis test — svaki izraz po 1800 puta
 

```
1.2+3.4 2.3-4.5 2.3+4.5 2.3/4.5
```

Slika 2

	Standardan PC		9.54 MHz kartica	
Frekvencija 8086/8088	4.77 MHz	4.77 MHz	9.54 MHz	9.54 MHz
Frekvencija 8087	0 MHz	4.77 MHz	0 MHz	9.54 MHz
BASICA (bez 8087)	891.	—	757	—
BASCOM (bez 8087)	169.50	—	64.7	—
87 BASIC (sa 807)	—	5.74	—	2.53
87 BASCOM (sa 8087)	—	3.35	—	1.56
True BASIC	160	10	—	35
Better BASIC	873.	12	—	5.20
Professional BASIC	420.	15	335.	6.
MS FORTRAN	—	6.92	—	3.26
RMCF FORTRAN	—	3.85	—	1.81
Lotus 1-2-3	374.	10.2	177.	4.80
Symphony 1.1	478.	17.5	—	10.2

Slika 3

Tabela je podeljena u dve kolone, jer se HMO51 tehnologija bitno razlikuje od HMO53 — dok su HMO51 čipovi „umirali“ kada temperatura pređe specificiranu za jedan jedini stepen, HMO53 koprocesori često rade i na 130 stepeni Celzijusa — otuda plusveć Pažljivi pogled na sliku 1 otkriva jednu neobičnost — čipovi C8087-6 su verovatno trebali da budu C8087-3, ali nisu prošli testove. Utvrđeno je, sa druge strane, da oni rade u običnom PC-ju ukoliko im temperatura ne pređe 50 stepeni, pa je uvedena nova oznaka. Intel, međutim, tvrdi da ovakvi koprocesori nikada nisu izabli na otvoreno tržište, premda je 20.000 primeraka prodato OEM-u koji ih interno koristi. Ukoliko vam, dakle, neko ponudi C8087-6, glatko ga odbijte!

### Na brzinskom testu

Pостоje samo dva načina da upoznate brzinu aritmetičkog koprocesora — teži (i sigurniji) zahteva kupovinu logičkog analizatora i brojanje ciklusa dok se lakši (i nesigurniji) svodi na pisanje brzinskih testova, takozvanih *benchmark* programa.

Pisanje programa koji testiraju brzinu nekog računara je vrlo osetljiva operacija — ukoliko znamo kako neki sistem funkcioniše, lako ćemo napisati program koji će na njemu biti mnogo brži nego na drugim mašinama, kao i program koji će na njemu biti mnogo sporiji — zato mnoge firme biraju brzinske testove koji odgovaraju baš njenim proizvodima! Čak i ako je brzinske testove pisao nepristrasni programer, u njihove rezultate ne smete previše da se pouzdate — možda su isprobane baš st-

vari koje su vama retko potrebne. Sve brzinske testove ćemo, dakle, uzeti sa velikim rezervama, ali bez njih ipak ne možemo — bolje je, na kraju krajeva, imati bilo kakve nego nikakve podatke!

Slika 2 prikazuje dva testa kojima ćemo podvrgnuti aritmetičke koprocesore: prvo smo nazvali aritmetički test (u stranoj literaturi se zove *savage* zbog divljeg ritma kojim se generišu brojevi) i kodirali na fortranu, bajziku i paketu „Lotus 1-2-3“, dok je drugi (*zajednički element*) isproban na bajziku i raznim verzijama fortrana. Varijantu drugog testa koja je prilagođena *spreadsheet* paketima Lotus 1-2-3 i *Symphony 1.1* smo, u čast poznate trkačke staze, nazvali *Megalopolis* test — pogodite li zašto? U daljem tekstu će nas interesovati prevenstveno vreme izvršavanja programa sa slike 2; testiranje tačnosti je daleko osetljivije pitanje kojim ćemo se baviti drugom prilikom.

Testiraćemo tri okruženja aritmetičkog koprocesora: prvo je 8088/8087 na 4.77 MHz (standardni PC), drugo 8086/8087 PC kartica koja obezbeđuje rad na 9.54 MHz i treće 80286/80287 na frekvencijama od 6 do 9 MHz. Za početak ćemo porediti prva dva okruženja i pokušati da proniknemo u razloge zbog kojih 9.54 MHz kartica daje dvostruko ubrzanje u nekim slučajevima i četvorstruko u drugim.

Sve operacije mikroprocesora možemo grubo da podelimo u dve grupe: registarske manipulacije koje obavljaju ALU (*arithmetic and logic unit* ili, u domaćoj literaturi, aritmetičko logička jedinica) i ulazno-izlazne operacije čiju brzinu ograničava vreme koje je procesoru neophodno da pristupi

RAM-u ili nekoj drugoj periferiji. Pošto su sve registarske operacije sinhronizovane sistemskim klokom, njihova se brzina linearno povećava ubrzavanjem oscilatora — povećajte dvostruko radnu frekvenciju, pa će registarske operacije biti dvostruko brže! Obzirom da 8086 i 8088 imaju jednaku internu arhitekturu, dvostruko ubrzanje kloka i promena procesora će izazvati dvostruko ubrzanje registarskih operacija. Ovi se zaključci odnose na programe koji uglavnom operišu sa podacima u registrima, a takvi su vrlo retki — svaki iole ozbiljniji program pristupa radnoj pa čak i masovnoj memoriji. Ulazno-izlazne operacije su, dakle, veoma važan parametar za procenu brzine nekog procesora.

Brzina pristupa memoriji kod 8086 i 8088 je određena širinom magistrale za podatke (8 bita na 8086, 16 bita na 8086) i dužinom takozvanog *prefetch* reda (*fetch* je faza pripreme instrukcije kada se iz memorije čita njen operacioni kod. *Prefetch* označava deo faze pripreme neke instrukcije koji se pokrpa sa fazom izvršenja prethodne ali ovaj izraz ne umemo da prevedemo). Obzirom na unapređenu konstrukciju 8086 i širinu njegove magistrale za podatke, brzina prenosa podataka na 9.54 MHz pločici je povećana četiri puta, što znači da će programi koji se intenzivno obraćaju RAM-u biti oko četiri puta brži nego na standardnom PC-ju.

Ostalo je još da pomenemo programe koji se intenzivno obraćaju masovnoj memoriji, na primer hard disku. PC-jev operativni sistem je jednokorisnički, što znači da procesor koji je podneo zahtev za ulaz odnosno izlaz nema nikakva druga dela nego da čeka da se taj zahtev ispuni; dužina ovog čekanja je određena karakteristikama samog hard diska, što znači da se vremena na PC-ju i 9.54 MHz kartici praktično ne razlikuju.

Stiven Fried u već pomenutom tekstu *The 8087/80287 Performance Curve* pokušava da izvede formulu koja daje očekivanu matematičku brzinu 9.54 MHz table u odnosu na standardni PC. Posle umereno jednostavnog izvođenja kojim se ovde nećemo baviti, dobija se da je relativna brzina jednaka  $4 \cdot (1+N)/(2+N)$ , gde je sa N označen količnik Nio/Nreg — broja ulazno-izlaznih i registarskih operacija u nekom programu. Ukoliko posmatramo program sa približno podjednakim brojem ulazno-izlaznih i registarskih operacija ( $N=1$ ), relativno ubrzanje bi trebalo da bude 2.67.

Koju vrstu operacija može da ubrza aritmetički koprocesor? On svakako neće učiniti da mikroprocesor brže pristupa RAM-u ili hard disku, ali će pretvoriti ogromne segmente manipulacija sa brojevima (jeste li nekada videli program koji deli brojeve u pokretnom zarezu?) u par instrukcija koje se izvršavaju veoma brzo — tipično za stotinak taktova. Zato očekujemo da prvi test (*savage*) koji se intenzivno bavi „žvakanjem brojeva“ pokaže ogromne dobitke u brzini.

Pošto smo razvili neku vrstu matematičkog modela, prelazimo na *benchmark*. Možda ćete se zapitati zbog čega je bilo potrebno potrošiti ovoliko prostora za teorijska razmatranja kada već planiramo eksperimentalne testove. U nauči (a testiranje performansi računara zalazi duboko u nauku) se, međutim, uvek počinje od teorije, jer



ona omogućava da se procene rezultati i značaj eksperimenata — ukoliko eksperiment daje nešto što teorija ne može da objasni, mora da se promeni ili eksperiment ili teorija!

Slika 3 prikazuje vremena izvršavanja aritmetičkog testa sa slike 2 u raznim okruženjima — na raznim (kompajliranim i nekompajliranim) varijantama bejzika, — na fortranu, na Lotusu i na programu Sympy. Program je izvršavan na standardnom PC-ju i na 9.54 MHz kartici, pri čemu je aritmetički koprocessor bio uključan ili isključen — situacije u kojima je koprocessor isključen smo opisali navodeći da on radi na 0 MHz. Crtice u tabeli označavaju nemoguće situacije — „MS Fortran“ na primer, odbija da radi bez aritmetičkog koprocessora (verovatno se ga zbog toga davno obrisali) dok „Basica“ ovaj koprocessor jednostavno ne konstatuje. Sva vremena na slici su u sekundama.

Slika 3 pokazuje spektakularno ubrzanje kako u interpretiranom tako i u kompajliranom bejziku (faktori su 155 odnosno 50 na standardnom PC-ju i u *Better Basicu* (faktor 72), dobiti na paketu *Sympy* su nešto manji (faktor 30) dok se na *True Basicu* i *Lotusu* dobijaju sasvim umereni rezultati. Treba da kažemo da se razne verzije verzije bejzika više ponose svojim mogućnostima debagovanja i kontrolnim strukturama, a manje brzinom i tačnošću aritmetike, što znači da će se korisnici kojima je aritmetika najbitnija opredeliti za fortran ili (možda) novi Borlandov „Turbo bejzik“.

Zanimljivo je uporediti kompajlere za fortran i bejzik. Ukoliko ste smatrali da je fortran mnogo brži, pogrešili ste — stari Microsoftov BASCOM daje brži kod od izvaniknih varijanti najstarijeg kompjuterskog jezika! Treba, naravno, da se čuvamo dalekosežnih zaključaka — iz rezultata jednog testa nikako ne možemo da izvedemo konačne zaključke o kvalitetu nekog kompajlera ili, još manje, jezika.

Najbolje prolaze kupci aritmetičkih koprocessora ako se bave raznim CAD primenama — nije lako staviti na papir test koji bi pokazao koliko se crtanje linija ili bojenje površina ubrzava dodavanjem aritmetičkog koprocessora, ali će svako ko dokupi 8087 a onda startuje „AutoCAD“ biti u pravom smislu fasciniran. Važi, dakako, i obrnuta logika — neko ko intenzivno radi sa „AutoCAD“-om, a zatim pređe na mašinu bez aritmetičkog koprocessora će se začas priseliti najstrašnije jeverjske kelte „Dabogda imao p nemao“. Pre nego što izvedemo konačne zaključke o upotrebljivosti aritmetičkih koprocessora na PC-ju moraćemo, međutim, da se pozabavimo AT-om i koprocessorom 80287. Za mesec dana ćemo vas, dakle, iznenaditi činjenicom da je 8087 daleko spektakularniji dodatak od novog i modernog 80287!



Dejanove  
pitalice

## Telefonska zagonetka

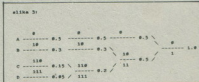
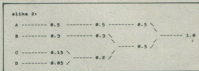
*Posle nekoliko lakih problema, poželili smo malo da iskušamo čitaoc i tako je nastala dvadeseta (jubilara) Pitalica koju smo ocenjivali kao relativno tešku. Dodatni faktor koji utiče na manji odziv je mesec jun koji uvek pročiće u znaku ispitata, kraja školske godine i drugih iskušenja. Rezultat — za prvih desetak dana juna primili smo svega nekoliko pisama! Rokovi zaključenja avgustovskih računara su, međutim, takvi da nismo mogli da čekamo dvadeset peti jun — ovoga ćemo puta izložiti rešenje Telefonske zagonetke dok ćemo imena nagrađenih i komentari primljenih odgovora objaviti kroz mesec dana.*

Podsetimo se, po običaju, Pitalice koja se bavila problemima telefonskog saobraćaja u nekom gradu. Grad ima tačno 30.000 telefonskih pretplatnika koje smo numerisali brojevima 1, 2, 3, ..., 29.999, 30.000. Ova numeracija nije slučajna: pretplatnik čiji je redni broj manji prima više poziva: broj 1 je verovatno neka javna služba (npr. obaveštenja) dok je broj 30.000 neki pretplatnik koji retko posećuje svoj stan. Raspodelu telefonskih poziva smo ilustrovali programom sa slike 1 koji, za zatado N, daje broj poziva koje mesečno prima pretplatnik čiji je redni broj N.

Trebalo je pretplatnicima dodeliti telefonske brojeve trudeći se da broj cifara koje svakog meseca moraju da se okrenu (otkucaju) bude minimalan: prirodno je dodeliti često pozivanim pretplatnicima dvocifrene ili trocifrene brojeve dok pretplatniku čiji je broj 30.000 možete mirne duše da dodelite vrlo dugačak broj. Upozorili smo vas, međutim, na jednu sitnicu: ako nekom pretplatniku dodelite broj 95, nikom drugom ne možete da dodelite broj 952 ili 95248 ili, uopšte, bilo koji broj koji počinje sa 95!

Rešenje ovoga zadatka se zasniva na takozvanim Hafmanovim kodovima koji su opisani u većini ozbiljnih programerskih udzbenika; čitaoci „Računara“ su ih upoznali kroz tekst „Platonove ljubavne muke“ iz 17. broja našeg časopisa. Da ne biste preurali po starijim brojevima, upoznaćemo Hafmanove kodove na jednostavnom telefonskom primeru.

Pretpostavimo da grad ima samo četiri telefonska pretplatnika A, B, C i D; A mesečno prima 50 poziva, B trideset, C petnaest a D samo 5. Telefonski brojevi su, osim toga, binarni — sastoje se samo od nula i jedinica. Bilo bi prirodno dodeliti



slika 61:

```

10 REM
20 REM Redni broj ----> telefonski
30 REM
40 REM Prva programa Milana Orlica
50 REM
60 REM
70 READ I:
80 DIM A(I*10)
100 READ A(I)
110 NEXT I
120 A=7: A(I*10)=0
130 FOR I=1 TO I*10
140 A(I)=A
150 A(I*10)=A
160 NEXT I
170 INPUT "Redni broj pretplatnika " ; B
180 PRINT "Telefonski broj: ";
190 I=1: B=99: B10=0
200 IF B=100 THEN GOTO 210
210 B10=B\10: B=B%10: I=I*10+1: GOTO 170
220 IF B=1 THEN PRINT STR$(B10);I:
230 IF B=0 THEN PRINT STR$(B10);I:
240 GOTO 170
250 DATA 9,0,0,0,0,1000,200,200,277,27,20,10
260 B10=B\10: B=B%10
270 IF I=10 THEN B10=10*B10+7: B10=10*B10+7: B10=10*B10+7: B10=10*B10+7: B10=10*B10+7
280 RETURN
  
```

slika 51:

```

Redni broj pretplatnika: 1
Telefonski broj: 0000
Redni broj pretplatnika: 2
Telefonski broj: 0001
Redni broj pretplatnika: 10
Telefonski broj: 0009
Redni broj pretplatnika: 200
Telefonski broj: 0199
Redni broj pretplatnika: 1077
Telefonski broj: 1076
Redni broj pretplatnika: 21000
Telefonski broj: 93584
Redni broj pretplatnika: 31111
Telefonski broj: 9999999991275
  
```

slika 51:

```

10 REM
20 REM Telefonska pitalica
30 REM
40 REM Input
50 DIM A(30000)
60 DIM B(30000)
100 FOR I=1 TO 30000
110 B(I)=I: A(I)=0
120 NEXT I
130 NEXT I
140 GOTO 170
150 INPUT "Redni broj pretplatnika " ; B
160 IF B=100 THEN GOTO 170
170 I=1
180 IF B=10 THEN I=I*10+1: GOTO 150
190 B10=B\10: B=B%10: I=I*10+1: GOTO 150
200 PRINT "Telefonski broj: "; B10;I:
210 PRINT
220 GOTO 150
230 DATA 9,0,0,0,0,1000,200,200,277,27,20,10
240 B10=B\10: B=B%10
250 IF I=10 THEN B10=10*B10+7: B10=10*B10+7: B10=10*B10+7: B10=10*B10+7: B10=10*B10+7
260 RETURN
  
```



# Dnevnik jedne veze

*Nijedan uređaj vezan za računare ne može se ni zamisliti bez nekog interfejsa. Stvar je prosta — interfejsa sa jedne, interfejs sa druge strane, dva konektora, par žica, nešto softvera i uređaji mogu da razgovaraju do mile volje! Ni izbor nije mali: CENTRONICS, RS232, BCS, SDLC, IEE488, HPIB, GPIB... Jeste li nekad stvarno probali? Ovo je dnevnik jednog delikatnijeg pokušaja koji je (početnička sreća!) nakon silnih peripetija ipak završen sa uspehom.*

Problem se pojavio iznenada. Pored PC računara našla se jedinica trake. Nedostajao je samo mali korak, bilo kakva veza ova dva uređaja i PC-u se otvaraju nova vrata — ogromna količina podataka akumuliranih na velikim sistemima mogla bi se koristiti bez naročitih investicija. Zbog svoje pouzdanosti i cene, traka je i dalje jedan od važnih skladišnih medijuma. I ne samo to. Sva indirektna razmena podataka između računara odvija se preko trake. Unos podataka najčešće se vrši na traci. Na trakama su zaštitne kopije svih drugih magnetnih medijuma. Na trakama je... Da ne dužimo. Pitanje: „Da li se jedinica za unos podataka na traku može vezati na PC?“. Odgovor: „Pa... ako ima neki interfejs... nema problema. Interfejsi zato i postoje!“

## Hardverske spekulacije

**Formulacija zadatka 1:** Ako na jednoj strani imaš IBM PC kompatibilni računar a na drugoj Singer 4510 jedinicu trake koja je opremljena komunikacionim adapterom za BSC protokol, šta treba uraditi da bi prenos podataka između ova dva uređaja bio moguć?

### ODGOVOR 1.

Potrebno je: pod a) nabaviti BSC adapter za PC, pod b) nabaviti softver za komunikaciju i pod c) nabaviti odgovarajući kabl. Prednosti ovakvog rešenja su nesumnjive — korisnik ne mora da zna ni šta je to BSC ni kako se povezuje.

BSC adapter je nabavljen. Divna mala pločica košta neku stitnicu od dvadesetak miliona (deviznih, jasno), ali šta je to prema onome što se dobija — ubodeš ploču, zatvoriš računar, priključiš kabl i RADIII U stvari, zaboravili smo kabl, ali to sigurno nije problem, stemićemo tih par žica već nekako. Dakle još samo softver. Ima li neko softver za BSC? (Bolje neka se sam javi!) Počinju da pristizuju diskete. PERA.COM (ne, to nije, sigurno), ZIKA.EXE (možda je to?), BBBBCC.CFG (ovo je neka konfiguracija, što postoi!) BSC.EXE, evo gal Kucaj samo BSC! Počinje konverzacija (prevod, nap. prev.):

Adresa data porta 1) 0BC0 2) 08C0 3) 06C0? Odgovorite sa 1. 2 ili 3

Kako da znamo na kojoj adresi je data port? Uz karticu nema nikakvog uputstva! Da probamo 4. možda možemo da ga... BEEEEP:

24 računari 29 • avgust 1987.



Odgovorite sa 1. 2 ili 3

Ne može. Preturamo uputstvo za originalnu IBM BSC karticu, sve je to isto. Data port, data port... 08E2! Samo što to nema u spiskul četvorka je znači bita jedina šansa. Nema veze, preformulišemo zadatka.

**Formulacija zadatka 2:** Ako na jednoj strani imaš IBM PC kompatibilni računar SA BCS adapterom i BEZ softvera i na drugoj Singer...

### ODGOVOR 2.

Potrebno je samo: pod a) napisati program. Prednosti rešenja su očigledne — korisnik usvaja i u potpunosti savladava nova znanja.

Sva dokumentacija je tu. Originalno IBM uputstvo za originalnu BSC karticu i odgovarajuća literatura za Singer. Idemo prvo sa PC strane — hardverski opis, shema, i najvažnije, uputstvo za programiranje. Inicijalizacija adaptera počinje postavljanjem porta A kao izlaza, porta B kao ulaza, prva tri bita porta C kao ulaza, bit 3 kao izlaz. Ostali bitovi se ne koriste. Postavljanje ovih parametara se postiže upisivanjem ODGOVARAJUĆE kontrolne reči u statueni registar... Doduše, nedostaje konkretna vrednost, ali pronaći ćemo mi već to — 256 pokušaja i gotovo. Dakle, inicijalizaciju smo završili. Nastavljamo: „Nakon inicijalizacije treba ponovo omogućiti interapt koji je inicijalno onemogućen. Čip automatski

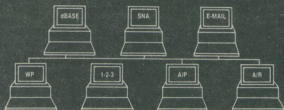
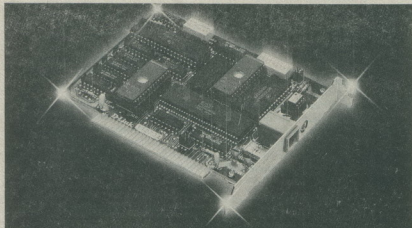
ulazi u „hunt“ modalitet i nakon sinhronizacije sa spoljnim signalom postavlja bit u statusnom registru... ukoliko se koristi interni klok... ako se sinhronizacija vrši eksternim klokom ovaj mora biti prisutan na nožici... zatim se upisuje komandna reč... isti registar daje status... bajt sa porta mora biti pročitan pre nailaska sledećeg kako bi... interapt se džemperom može prebaciti na IRQ4... Dakle, moraćete ozbiljno!

## A sada sasvim ozbiljno

BSC BINARY SYNCHRONOUS COMMUNICATION je protokol za serijski prenos podataka. Spolja, po mnogo čemu, podseća na RS232 interfejs: naponski nivoi su isti, brzine prenosa, raspored linija na kontaktima 25-pinskog konektora je takođe isti, pa i sami nazivi linija i njihova namena. Međutim, postoji veoma bitna razlika. Start i stop bit RS232 protokola ne postoje, već prenos teče sinhrono — bitovi jedne poruke spojeni u neprekidni niz u zavisnosti od svoje vrednosti menjaju naponsko stanje na liniji u -12 +12 V. Pri tome, prijemni uređaj, koristeći bilo interni ili eksterni tekst predajnog uređaja kojim je definisana brzina prenosa, u istim vremenskim razmacima očitava stanje linije i vraća naponske nivoe u poverku bitova. Princip je veoma jednostavan, ali da bi funkcionisao mora biti ispunjeno nekoliko uslova. Sinhronizacija mora biti precizna za čitavo vreme trajanja poruke. I veoma mala promena u intervalu u kome se uzimaju uzorci sa prijemne linije može dovesti do gubitka sinhronizacije, tako da će u najboljem slučaju čitava poruka biti pomerena za 1 bit. U gorim slučajevima uzimanje uzorka može pasti negde oko ivice promene vrednosti bitova u poruci, što će dati raznorazne rezultate, uvek daleko od stvarnih vrednosti.

Zbog ove osjetljivosti, BSC protokol predviđa da se podaci šalju isključivo u kraćim blokovima uz razne provere — bit parnosti se generiše za svaki bajt, kontrolna vrednost se generiše za čitav blok podataka, prijemni uređaj nakon svakog primljenog bloka potvrđuje da li je redni broj bloka paran ili neparan itd. Kritičan trenutak komunikacije je na početku poruke. Prvi bit koji naiđe pokreće merenje vremena kojim se obezbeđuje uzimanje uzorka u preciznim intervalima. Zbog toga poruka uvek započinje „rasterom“ bitova — dva bajta AAH (LPAD kod) kod kojih su bitovi nazmenično postavljani na 1 i 0. Sličan bajt, 22h (ASCII





SYN kod) se koristi na više mesta u poruci za dodatnu proveru sinhronisanosti. Bit 7 svakog bajta se koristi kao bit parnosti. Kraj jednog bloka podataka se obeležava bajtom 255 (sve binarne jedinice). Ovaj bajt je, u stvari, dobija kada na liniji u osam intervala nema promene naponskog nivoa i inače ne postoji u sedmobitnom ASCII setu sa parnom parnošću — broj 127 (01111111) ima neparan broj bitova, tako da bit 7 ostaje na nuli.

U toku stvarnog prenosa odvija se prilično živa komunikacija. Ako PC prima podatke a Singer (u daljem tekstu SG) šalje, razgovor teče ovako:

SG — svake tri sekunde poruka ENQ  
 PC — na prvi nalazak ENQ odgovara sa ACKO  
 SG — šalje 1. DATA blok  
 PC — odgovara sa ACK1  
 SG — šalje 2. DATA blok  
 PC — odgovara sa ACKO

i tako do kraja datoteke na traci, kada SG šalje jedan prazan blok (DUMMY BLOCK) i odmah zatim EOT (END OF TAPE) poruku.

Svaka od poruka iz ovog spiska (ENQ, ACKO, ACK1 itd) se sastoji iz niza bajtova — početnog rastera (LPAD), SYN bajtova, samog bajta poruke i na kraju jedan 255. DATA blok takode ima sličnu strukturu: nekoliko LPAD, pa nekoliko SYN, pa STX (START BLOCK), stvarni podaci, pa ETX (END BLOCK), kontrolni bajt i na kraju 255.

BSC adapter je konstruisan tako da najveći deo posla na sebe preuzima hardver. Početna sinhronizacija na LPAD bajt se obavlja potpuno hardverski. O čitavoj sinhronizaciji poruke brine se timer koliko tako da softveru na izgled ostaje manji deo posla. Na žalost, nije tako. Pošto je upotre-

ba interpat tehnika za prijem i slanje gotovo obavezna, po našoj početnoj proceni ne bi bilo moguće, pošavši potpuno od početka, proći bez 7 do 10 dana rada na razvoju. A ako je već potrebno toliko vremena, onda može daleko jednostavnije i jeftinije, BSC adapter može da se zameni za nešto korisnije i da se upotrebi ono što već postoji u računaru. Dakle, ponovo smo na formulaciji zadatka broj 1.

Ako se pitate šta to postoji u računaru a ima veze sa BSC adapterom, odgovorićemo vam odmah — Centroniks interfejs! Centroniks je najjednostavniji moguć interfejs — niz izlaznih linija i nekoliko ulaznih — čista digitalna I/O kartica koja se slobodno može nazvati USER portom PC!

## Negde sasvim blizu

Ideja je jednostavna — iskoristićemo jednu ulaznu liniju za prijem podataka, jednu izlaznu za predaju, masa i to je sve. Naponski nivoi ne odgovaraju (-12, +12 prema TTL) ali se to lako rešava sa malo hardvera. Da ne bi baš sve bilo tako jednostavno, iskoristili smo još jedan ulazno-izlazni par — na pločici se našao takt generator čiji je izlaz ulazio u PC, a iz PC-a se mogao preko izlazne linije proizvoljno startovati i zaustavljati.

Dakle, program za prijem mora da radi ovako:

- 1) U petlji se čita ulazna linija
- 2) Na prvu promenu se startuje takt generator
- 3) Čita se izlaz takt generatora i kada pokaže promenu ...
- 4) Čita se bit sa ulazne linije koji se ugurava u bajt
- 5) Kada je primljen bajt=255, cela poruka je primljena i

## 6) Sledi odgovor

Slanje je još jednostavnije — startuje se takt generator i po njegovom taktu šalje biti po bit poruke.

Pošto se ne sme propustiti ni jedan takt, svi PC interaptovi moraju biti sprečeni. To, doduše, krije jednu malu opasnost — kada se spreči interapt, nema čitanja tastature pa ne radi ni resetovanje sa ALT+CTRL+DEL. A vaš program se „vrti“ (pod tačkom 1) u neprekidnoj petlji i ako ste zaboravili da priključite kabl ili sa predajnim uređajem nešto nije u redu pa nema promene stanja na ulaznoj liniji ... Da ne biste stalno isključivali računar, dobro je u petlju ubaciti jedno brojanje ponavljanja, pa ako se ništa ne desi dok izbrojite, do recimo, 100000, možete izaći iz petlje uz odgovarajuću „TIME OUT“ poruku. Sve u svemu, jedna INLINE TURBO PASCAL mašinska rutina od dvestotinjak bajtova dovoljna je za ceo posao slanja i prijema niza bajtova iz nekog stringa. Šad još samo da se ukloni po protokol.

Prvo čitanje i poruka je tu! Primili smo ENQ poruku, ali ona baš nije po strukturi identična sličici iz uputstva. Umesto dva SYN signala dobijali smo bar po deset. Nije problem ako se zna da je tako, ali se naglo ruši poverenje u uputstvo, a bez njega ...

Kad se već SG tako raspituje za nas, bio je red da mu odgovorimo. Sastavimo sledeću poruku: LPAD LPAD SYN SYN SYN SYN SYN (kad on nama toliko, valjda i njemu treba) SYN SYN ACK 0 i 255, pošaljemo je, predemo brzo u prijem očekujući konačno blok podataka, vidimo na kontrolnom panelu SG kako se poruka šalje i ... ništa. Sledeća poruka je potpuno „raštímavana“! Sada već nešto nije jasno. Ako je prijem u redu, a jeste jer je prva poruka primljena kako treba, ako je predaja u redu, a jeste, jer je SG shvatio da treba da pošalje blok, onda nema razloga da svaki sledeći par prijem/predaja ne bude u redu. Da biste stekli realnu sliku kako se stvar dalje odvija ostavite ovaj tekst, zamislite se i nastavite sutra.

## Kraj, ili još malo do kraja

Jasno, rešenje je uvek prosto. Šaljući bajt 255 na kraju poruke, prošili smo vreme na nešto što se inače dešava i bez toga — linija miruje na logičkom stanju 1. SG je već na nulu nakon ACK shvatio našu poruku i odmah počeo predaju, a mi smo se to vreme još uvek bespotrebno slali ostatak bajta 255. Kad smo prešli u prijem, bilo je kasno — propustili smo nekoliko prvih bitova.

Greška je ispravljena i podaci su, konačno, krenuli! A zašto nije još uvek kraj? Kada je prijem sa SG rešen, ostalo je da se proces okrene. Sada mi šalje ENQ, SG šalje ACK 0, sve je u redu, kad odjednom ... a kako se računa kontrolna suma bloka. SG se ne može prevartiti — ako na karju bloka pošaljete nešto što se ne slaže vraća poruku NAK (NEGATIVE ACK) i odbija da zapiše podatke na traku. Da smo probali zdravorazumskom metodom i upotrebili ono što se inače upotrebljava, ovde bi stvarno bio kraj. Na žalost, prvo smo pogledali uputstvo a tamo piše: kontrolna suma je osmobitna kumulacija vrednosti polinoma  $X^{p+1}$ . Morali smo još malo da se zamislimo.

Zoran Životić

# Ni manje, ni veće

**Poređenja i ispitivanja sadržaja procesorskih registara predstavljaju polazne operacije u toku formiranja razgranatih programskih struktura. Neki mikroprocesori pružaju programeru tom prilikom izuzetan komfor u radu, dok drugi čak oskudevaju i u nekim osnovnim operacijama, zahtevajući od programera da se služi svakojakim trikovima kada poželi da izvede neki uslovni skok. Z80 je negde na granici između ove dve krajnosti: programer se ni u kom slučaju ne oseća ugroženim kada naiđe na grananje, ali ima situacija u kojima mora dobro da razmisli dok ne naiđe pravo rešenje.**

## Statusni registar

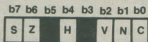
Svako grananje u programu odvija se na osnovu sadržaja statusnog registra *F*. To je jedan osmobaritni registar, koji uvek nosi informaciju o tome kako je proteklo izvršavanje prethodnih instrukcija programa. Na primer, ako je upravo bila izvršena naredba **ADD A, #30**, a rezultat sabiranja je veći od  $\#FF$ , automatski će biti setovan *multi bit* registra *F*, kao znak da je u operaciji sabiranja došlo do prenosa. Otuada se *bit 0* registra *F* zove *indikator prenosa (carry flag)*.

Može se desiti da nas uopšte ne interesuje kako je proteklo izvršenje operacije **ADD a, #30**, a u tom slučaju ćemo jednostavno nastaviti program drugim naredbama. Međutim, čest je slučaj da rezultat neke operacije neposredno određuje dalji tok programa. Na primer, u slučaju prenosa treba skokovati na neku adresu **EXIT**. Tada će program izgledati ovako:

```
ADD A, # 30
JP C.EXIT
```

Upotrebili smo naredbu **JP C** (Jump on carry), koja označava „skok u slučaju prenosa“. Ukoliko prenosa ne bude, program će se nastaviti normalno od naredbe iza **JP**.

Isto tako smo mogli da upotrebimo i naredbu **JP NC, EXIT** (Jump on non carry), ako smo želeli da se skok vrši u slučaju da nema prenosa.



F

Slika 1. Statusni registar

Od osam bitova koliko ih registar *F* ima, samo šest se koriste kao indikatori. Na slici 1. prikazan je raspored ovih bitova, a oznake su sledeće:

- S — indikator znaka (sign flag)
- Z — indikator nule (zero flag)
- H — indikator poluprenosa (half carry flag)
- V — indikator parnosti/prekoračenja (parity/overflow flag)
- N — indikator oduzimanja (subtract flag)
- C — indikator prenosa (carry flag)

U najvećem broju slučajeva programer ne mora da zna kako je koji indikator smešten u registru *F*. Očigledan primer smo upravo imali u naredbi **JP C,EXIT**. Nigde tu nije bilo potrebno da znamo da je indikator *C* baš *bit 0* u statusnom registru. Ali, kada dođe trenutak za izvođenje trikova, slika 1. će nam i te kako biti potrebna.

26 računari 29 • avgust 1987.

Nećemo ovde ponovo objašnjavati ulogu svakog indikatora, jer smo o tome u „Računarima“ više puta pisali. Umesto toga, preći ćemo na praktične probleme oko korišćenja statusnog registra u programiranju.

## Osmobaritna poređenja

Mikroprocesor *Z80* raspolaže naredbom za poređenje osmobaritnih brojeva. To je instrukcija **CP (compare)**, koja ima samo jedan argument (drugi je uvek akumulator). Na primer, **CP B** znači da će se porediti sadržaji registara *A* i *B*. U suštini, statusni registar *F* postaviće se potpuno isto kao da se radilo o naredbi **SUB B** (oduzimanje *B* od akumulatora), s tom razlikom što do oduzimanja zapravo neće doći. Ako su brojevi koji se poredi jednaki, biće setovan indikator nule ( $Z=1$ ). U protivnom je  $Z=0$ . Ako je sadržaj akumulatora manji od navedenog argumenta, indikator prenosa će biti setovan ( $C=1$ ). U protivnom je  $C=0$ .

U tabeli na slici 2. date su sve kombinacije indikatora *C* i *Z* po izvršenju naredbe **CP B**. Kombinacija  $C=1$  i  $Z=1$  nije moguća, jer pri rezultatu nula ( $Z=1$ ) indikator prenosa je obavezno resetovan.

C	Z	relacija
0	0	A > B
0	1	A = B
1	0	A < B

Slika 2. Indikatori *C* i *Z* po izvršenju naredbi **CP B**.

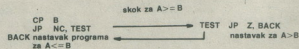
Obavljanjem poređenja i postavljanjem statusnog registra, naredba **CP** je završila svoje. Na nama, kao programerima, je da na osnovu toga obavimo grananje u programu.

Skok u slučaju  $A=B$  obavimo naredbom **JP Z**, jer je indikator nule setovan samo pri  $A=B$ , a resetovan u svim ostalim slučajevima. Slično tome, skok pri  $A<B$  obavljamo sa **JP C**, jer je indikator prenosa setovan jedino i samo za  $A<B$ . Kako, međutim, da izvedemo skok pri  $A>B$ ? Da li sa **JP NC**? ili možda sa **JP NZ**?

Na žalost, iz tabele sa slike 2. nije teško videti da je uslov  $NC$  ( $C=0$ ) ispunjen i pri  $A>B$  i pri  $A=B$ , dakle pri  $A>=B$ . A uslov  $NZ$ , s druge strane, ispunjen je pri  $A<>B$ .

Očigledno, treba nam neka naredba za uslovni skok oblika **JP NC, NZ** koja bi uzela u obzir *oba* indikatora. Želje su jedno, a realnost sasvim nešto drugo: *Z80* ne poznaje ovakvu naredbu. Vlasnici *PC-a* se sada zadovoljno smeškaju i trljaju ruke, jer njihov procesor *8088* ima sve moguće uslovne skokove koji bi programeru pali na pamet.

Ipak, nećemo se tako lako predati. Jedna mogućnost je primena uzastopnih testiranja. Obavićemo skok pri  $A>=B$ , a onda ćemo se vratiti ako je  $A=B$ , kao što to prikazuje slika 3. Nije baš neka elegancija, ali vrši posao.



Slika 3. Višestruko testiranje

Gotovo uvek ćemo, međutim, izbeći ovakve neprijatne situacije na malo lukaviji način. Kada poredimo dva broja, recimo  $m$  i  $n$ , a želimo da obavimo skok pri  $m<n$ , onda ćemo  $m$  ubaciti u

akumulator, izvršiti poređenje CP n i skok JP C. Ali, ako vršimo skok pri  $m > n$ , stavimo n u akumulator i obaviti CP m, a zatim opet skok JP C. Tablica na slici 4. daje i sve ostale mogućnosti za uslovi skok u zavisnosti od odnosa brojeva m i n.

uslov za skok	sadržaj akumulatora	poređenje	naredbe za skok
$m = n$	$A = m/A = n$	CP n/CP m	JP Z
$m < n$	$A = m$	CP n	JP C
$m > n$	$A = n$	CP m	JP C
$m < = n$	$A = n$	CP m	
$m > = n$	$A = m$	CP n	JP NC
$m < > n$	$A = m/A = n$	CP n/CP m	JP NZ

Slika 4. Poređenje u zavisnosti od skoka

**Da li je nula?**

Specijalan slučaj osmootbitnih poređenja jeste ispitivanje da li je zadati broj nula. Obično se taj zadati broj nalazi u akumulatoru, a ako nije, prvo se tamo dovode.

Dakle, problem svodimo na ispitivanje sadržaja akumulatora. Postoje dva osnovna načina da se proveri da li je A nula:

1) CP #00  
JP Z, ...

Ovo je najprirodniji, ali i najglupiji način da se postigne ono što želimo. Operacija CP #00 zauzima dva bajta i izvršava se za sedam mašinskih ciklusa, što je čisto rasipanje memorije i vremena.

2) AND A  
JP Z, ...

Ovo je daleko elegantniji i, zapravo, najčešći način testiranja akumulatora. Operacija AND A zauzima samo jedan bajt i traje četiri ciklusa. Na prvi pogled izgleda malo besmisleno, jer se akumulator dovodi u logičku konjunkciju (logičko „i“) sa samim sobom. Ako je neki bit bio setovan, ostaće setovan, jer je  $1 \text{ AND } 1 = 1$ . A ako je bio resetovan, ostaće resetovan jer je  $0 \text{ AND } 0 = 0$ . Dakle, sadržaj akumulatora se neće izmeniti, ali statusni registar će biti postavljen u skladu sa tim sadržajem.

**Ispitivanje znaka**

Kada radimo sa brojevima u opsegu između  $-127$  i  $+127$ , onda je sedmi bit uvek rezervisan za predznak: jedinica označava negativan broj, a nula pozitivan (ili nulu).

Predznak akumulatora može se najjednostavnije proveriti sa AND A, a zatim testirajući indikator znaka (sign flag). Recimo, skok u slučaju negativnih brojeva obavićemo sa:

AND A  
JP M, ... ;skok ako je A negativno

Može se sedmi bit akumulatora neposredno dovesti i u indikator prenosa C, nekom od naredbi transiranja i rotiranja uljevo, mada to automatski menja sadržaj:

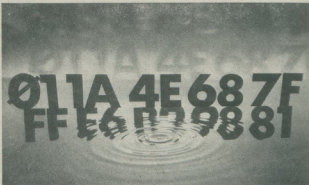
RLCA  
JP C, ... ;skok ako je A negativno

I, naravno, bit 7 akumulatora može se i direktno testirati (uz nepotrebno tračenje vremena i memorije):

BIT 7,A  
JP NZ, ... ;skok ako je A negativno

Ukoliko broj koji se testira nije u akumulatoru, a nije ekonomično prenositi ga u akumulator, onda je operacija BIT nezamenjiva. Na primer, za slučaj nekog broja u memoriji na adresi sadržanoj u HL, testiranje znaka bismo obavili sa:

BIT 7,(HL)  
JP NZ, ... ;skok ako je broj negativan



**Šesnaestobitna poređenja**

Mikroprocesor Z80 jeste osmootbitni, ali to nikako ne znači da se svaki program svodi na obradu isključivo osmootbitnih brojeva. Ne treba zaboraviti da su sve memorijske adrese upravo 16-bitne, i veoma često je programer u prilici da radi sa 16-bitnim registarskim parovima mikroprocesora. Problem nastaje kada treba testirati sadržaj registarskog para, pošto ne postoji neka CP naredba sa šesnaestobitnim argumentima.

Praktično jedino rešenje u takvim situacijama je upotreba operacije oduzimanja SBC HL, rp, koja oduzima sadržaj registarskog para rp (to može biti BC, DE, HL ili SP) od sadržaja registarskog para HL. Oduzima se istovremeno i trenutno stanje indikatora prenosa (nula ili jedan) i o tome stalno moramo voditi računa. To znači da se, pre upotrebe SBC instrukcije u cilju poređenja dva broja, moramo uveriti da je indikator prenosa resetovan, ili, radi svake sigurnosti, možemo ga uvek resetovati neposredno ispred SBC HL, rp nekom logičkom operacijom (opet je najpodesnija AND A jer ne menja sadržaj akumulatora).

Konkretno, uzećemo da porođimo sadržaje HL i DE. Recimo da se skok vrši na adresu FINISH samo u slučaju da je HL manje od DE:

```
AND A           ;resetuje se indikator prenosa
SBC HL, DE     ;poređenje (tačnije: oduzimanje)
JP C, FINISH   ;skok ako je HL < DE
```

Ovim je problem rešen sve dotle dok nam je svejedno šta se dešava sa sadržajem registarskog para HL. Jer, po obavljenoj operaciji SBC HL, DE u HL će se smestiti rezultat oduzimanja. Od dva broja koja smo poredili, ostaće očuvan samo onaj u DE.

Ako nam je važno da i sadržaj HL ostane neporemećen, moramo se snalaziti na razne načine. Postoji jedno veoma prosto, mada estetski ne i naročito privlačno rešenje: odmah posle SBC HL, rp obaviti sabiranje ADD HL, rp, čime se obnavlja sadržaj HL. Jedina primedba je što i operacija ADD sama po sebi deluje na statusni registar, a nas interesuje efekat prethodne operacije SBC. Međutim, nema razloga za brigu. Operacija ADD HL, rp zamišljena je upravo tako da minimalno poremeti sadržaj statusnog registra. Pre svega, uopšte ne deluje na indikatore znaka S i nule Z. Na indikator prenosa C deluje, ali na isti način kao i SBC: Ako je posle SBC HL, rp indikator prenosa bio resetovan, ostaće resetovan i posle ADD HL, rp. A ako je bio setovan, ostaće setovan. Prema tome, sva tri indikatora S, Z i C ostaće posle ADD HL, rp praktično nepromenjena. Preostala tri indikatora programeru ipak nisu naročito važna.

Dakle, naš prethodni primer bi sada izgledao ovako:

```
AND A           ;resetuje se indikator prenosa
SBC HL, DE     ;porede se sadržaji HL i DE
ADD HL, DE     ;obnavlja se HL
JP C, FINISH   ;skok ako je HL < DE
```

Dalje sledi ista priča kao i u slučaju osmootbitnih poređenja: ne možemo na ovaj način ostvariti skok za bilo kakav odnos sadržaja HL i DE. Na primer, za  $HL = DE$  primenićemo operaciju JP Z, FINISH. Ali, za  $HL > DE$  nećemo moći tako da uradimo ništa. Opet moramo voditi računa o redosledu brojeva koji se poredi. Na sreću, stvar se olakšava, jer mikroprocesor raspolaže jednom veoma korisnom i brzom operacijom EX DE, HL, koja za samo četiri otkucanja takta razmenjuje sadržaje registarskih parova DE i HL; ono što je bilo u jednom preći će u drugi.



### Šesnaestobitna nula

Kada nas interesuje da li neki registarski par sadrži nulu, najjednostavnije je proveriti da li svaki od osmoinbitnih registara u tom paru sadrži nulu. Na primer, uslov  $BC=0$  znači istovremeno i  $B=0$  i  $C=0$ . Razume se, nema potrebe da ispitujemo i  $B$  i  $C$  posebno. Elementarno poznavanje logičkih operacija tu će nam omogućiti uštedeti trud. Treba se samo setiti da logička disjunkcija („ili“) daje rezultat nula samo ako su oba argumenta nula. Dakle, treba nekako izvršiti  $B OR C$  i onda testirati rezultat:

LD A, B ;sadržaj B prenosi se u akumulator.  
OR C ;da bi se doveo u disjunkciju sa C  
JP Z, ... ;skok ako je  $BC=0$

Skok u slučaju  $BC<>0$  obavili bismo, jasno, sa JP NZ.

### Ispitivanje opsega

Kao primer ispitivanja i poređenja uzećemo problem utvrđivanja opsega u kome se nalazi zadati broj. Konkretno, napravivši potprograme koji ispituju da li zadati broj predstavlja ASCII kod neke od cifara dekadnog brojnog sistema, ili neko od slova abecede. Sve što nam je u tu svrhu potrebno jeste tabela ASCII kodova.

Cifre dekadnog sistema 0 — 9 kodiraju se heksadekadno sa #30 — #39. Ako je broj koji se ispituje u akumulatoru, samo treba videti da li je on manji od #30 ili veći od #39, u kom slučaju ne predstavlja dekadnu cifru (primer 1). Kada ispitujemo da li je A veće od #39 vršimo u stvari CP #3A. U slučaju vrednosti do #39 (zaključno sa #39) biće setovan indikator prenosa. U slučaju kodova od #3A pa na više, indikator prenosa će biti resetovan. Nama odgovara komplementarno stanje indikatora (nula za #39 i manje i jedinica za #3A i više) pa zato koristimo operaciju CCF.

#### Primer 1.

Potprogram NUMER ispituje da li akumulator sadrži ASCII kod neke od cifara dekadnog sistema (#30 — #39). Na izlazu je indikator prenosa resetovan ( $C=0$ ) ukoliko akumulator sadrži dekadnu cifru, a setovan ( $C=1$ ) u protivnom.

```
NUMER CP #30 Da li je kôd manji od bilo koje cifre?
RET C Ako jeste, povratak ( $C=1$ ).
CP #3A Da li je kod preko #39? Ako jeste, indikator
prenosa će biti resetovan, a to je suprotno
od onoga što želimo da imamo na izlazu.
CCF Korekcija indikatora prenosa.
RET Povratak.
```

Testiranje slova abecede je nešto malo složenije. Imamo, zapravo, dva opsega: velika slova između #41 i #5A i mala između #61 i #7A. Primećujemo da se mala i velika slova razlikuju samo u jednom bitu. Na primer, veliko „A“ se kodira sa #41 (BIN 01000001), a malo „a“ sa #61 (BIN 01100001). Jedinica razlika je u tome što malo „a“ ima setovan bit 5. Mi unapred ne znamo da li akumulator sadrži malo ili veliko slovo, ili bilo kakvo slovo uopšte, ali možemo odmah jednostavno resetovati bit 5 operacijom AND #DF i onda samo proveriti da li se dobijeni broj nalazi u opsegu velikih slova. Program bi u svemu bio sličan sa onim za dekadne cifre. Međutim, obično će nam biti potrebno da sadržaj akumulatora ostane neizmenjen. Zato je bolje ispitivati posebno oba opsega za mala i velika slova, kao u našem primeru 2.

#### Primer 2.

Potprogram ALPHA ispituje da li sadržaj akumulatora odgovara ASCII kodu nekog slova abecede. Na izlazu je indikator prenosa resetovan ( $C=0$ ) ako akumulator sadrži slovo, a u protivnom je setovan ( $C=1$ ).

```
ALPHA CP #41 Da li je kod manji od bilo kog slova?
RET C Ako jeste, povratak ( $C=1$ ).
CP #5B Da li je to veliko slovo?
CCF Korekcija indikatora prenosa.
RET NC Povratak ako je veliko slovo ( $C=0$ ).
CP #61 Da li je karakter ispod slova „a“?
RET C Ako jeste, povratak ( $C=1$ ).
CP #7B Da li je to malo slovo?
CCF Korekcija indikatora prenosa.
RET Povratak.
```

### Trikovi sa statusnim registrom

Za kraj smo ostavili jednu zanimljivu ideju kako izbeći višestruka suvišna ispitivanja, brige oko redosleda argumentata koji se porede, stanja indikatora pre i posle poređenja i slične gljovolje. Ideja je primenljiva i u sve probleme vezane za grananja u programu, a mi ćemo je ilustrovati na primeru testiranja nekog realnog broja zapisanog u formi pokretnog zareza. Napravivši potprogram koji testira zadati broj  $x$ , a onda nam omogućuje grananje za svaki od mogućih uslova:  $X=0$ ,  $X<0$ ,  $X>0$ ,  $X<=0$ ,  $X>=0$  i  $x<>0$ .

Koristićemo sva tri indikatora: S, Z i C i zahtevati sledeće: Ako je broj jednak nuli ( $x=0$ ) indikator nule je setovan (Z, zero), a u protivnom ( $x<>0$ ) indikator je resetovan (NZ, non zero). Ako je broj negativan ( $x<0$ ) indikator znaka je setovan (M, sign negative), a u protivnom ( $x>=0$ ) indikator je resetovan (P, sign positive).

Ako je broj strogo pozitivan ( $x>0$ ) indikator prenosa je setovan (C, carry), a u protivnom ( $x<=0$ ) indikator je resetovan (NC, non carry).

Svi uslovi, zajedno sa odgovarajućim sadržajem statusnog registra F dati su u tabeli na slici 5.

vrednost broja	indikator			uslov za skok	statusni registar	
	S	Z	C		bin	hex
$x=0$	0	1	0	Z	0 1 0 0 0 0 0 0	#40
$x<0$	1	0	0	M	1 0 0 0 0 0 0 0	#80
$x>0$	0	0	1	C	0 0 0 0 0 0 0 1	#01
$x<=0$	X	X	0	NC	X X 0 0 0 0 0 0	#40/#80
$x>=0$	0	X	X	P	0 X 0 0 0 0 0 X	#40/#01
$X<>0$	X	0	X	NZ	X 0 0 0 0 0 0 X	#80/#01

Slika 5. Priprema statusnog registra

Potprogram koji obavlja ovaj posao veoma je jednostavan. Njegov zadatak je samo da postavi u statusni registar vrednost #40 u slučaju  $x=0$ , #80 u slučaju  $x<0$  ili #01 u slučaju  $x>0$ . Jedinica poteškoća je u tome što ne raspolazemo mašinskom naredbom oblika LD F, nm. Zato ćemo prethodno pripremiti registar C, a onda kombinacijom PUSH BC, POP AF preneti željeni sadržaj iz C u F. Time se, istina, i B prenosi u A, ali to je usputni efekat koji nam ne smeta.

#### Primer 3.

Potprogram TEST ispituje vrednost realnog broja u memoriji. Na ulazu HL sadrži adresu broja. Na izlazu su indikator S, Z i C postavljeni u zavisnosti od vrednosti broja, prema tabeli sa slike 5. HL na izlazu ne menja vrednost, ali C i A nisu očuvani.

```
TEST LD C, #40 Pretpostavka da je  $x=0$  (status: P, Z, NC).
LD A, (HL) Uzimanje eksponenta realnog broja.
AND A Ako li je broj jednak nuli?
JR C, T.END Ako jeste, vrši se skok na izlaz.
LD C, #80 Pretpostavka da je  $x<0$  (status: M, NZ, NC).
INC HL Pomeranje na prvi bajt mantise.
BIT 7, (HL) Da li je broj manji od nule?
DEC HL Vraćanje HL na početnu vrednost.
LD R, NZ, T.END Ako je broj manji od nule, skok na izlaz.
JR C, #01 Broj je veći od nule (status: P, NZ, C).
T.END PUSH BC Koristićemo mašinsko steka sadržaj registra C
POP AF prenosi se u statusni registar F.
RET Povratak.
```

Kompletni potprogram je dat u primeru 3. Realan broj koji se testira nalazi se u memoriji u standardnom formatu: prvi bajt zapisa predstavlja eksponent. Ako je on nula, to znači da je ceo broj nula. Na drugom bajtu počinje mantisa, ali je bit 7 rezervisan za predznak: jedinica znači minus, a nula plus.

Jovan Skuljan

# 77 PC programa

*Kada su mi iz redakcije „Računara“ predložili da priprelim katalog programa za PC, moram priznati da u prvom trenutku nisam ni slutio šta me čeka. Uzor za ovaj katalog trebalo je da bude katalog programa za džepne računare koji je svojevremeno bio objavljivian u časopisu „Galaksija“. Samo, PC nije isto što i HP 41C, a „dBASE III PLUS“ što i „Gomilice“.*

*Biblioteka programa za PC računare je najbogatija na svijetu, što jednom „softverskom selektoru“ može samo da oteža posao. Prilično veliki problem bio je kako, i koje, programe izabrati za ovaj katalog. Teško je suditi o vrijednosti i značaju nekog programa ako je oblast njegove primjene daleke oblasti vašeg interesovanja. Ono što je za nekog od neprocjenjivog značaja, za drugog nema nikakve važnosti. U pokušaju pomirenja tih suprotnosti, težište izbora je palo na one programe koji bi mogli biti interesantni za širi krug korisnika. Zato je opisano dosta procesora teksta, programskih jezika i programerskih pomagala, integrisanih paketa... Koliko je to bilo moguće, opisani su i neki specijalistički paketi.*

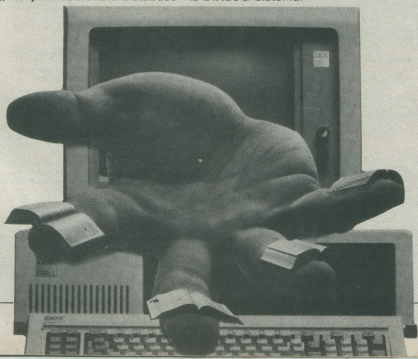
*Postavljalo se pitanje i kako, i koje programe izdvojiti unutar svake kategorije. Osnovni kriteriji su bili kvalitet i rasprostranjenost programa na našem „tržištu“.*

*Klasifikacija programa u kategorije takođe bi se mogla shvatiti uslovno, jer nije urađena ni po kakvim standardima. Uz to, neki programi bi se mogli svrstati u razne kategorije („dBASE III/PLUS“ je relacions baza, ali i programski jezik).*

*Uz programe, gdje je to bilo značajno (i poznato), navedeni su kritični zahtjevi na hardver.*

*Ovaj katalog bi trebalo da omogući lakše snalaženje u sve većoj ponudi i, bar u prvom koraku, pomogne u izboru odgovarajućeg softvera. Obradeno je 77 programa (važno je da je broj okrugao), a ako se pokaže da je ovakav katalog koristan, objavljivaćemo njegove dopune iz broja u broj.*

*Izvori su bili lično iskustvo u radu sa pojedinim programima, članci iz časopisa „Byte“ i informacije dobijene putem „Computer Software Database“ iz DIALOG sistema.*



## Biblioteke potprograma

## Turbo Database Toolbox

**Borland International**  
Biblioteka potprograma

Ovim programom dobijate kompletnu biblioteku procedura za „Turbo Pascal“ namijenjenu pretraživanju i sortiranju datoteka. Ove procedure su jako zgodne za razvoj aplikacija zasnovanih na bazama podataka. Program će raditi sa minimalno 128 K RAM-a.

## Turbo Editor Toolbox

**Borland International**  
Biblioteka potprograma

Tu je sve što vam treba da napravite svoj sopstveni tekst editor. Odgovarajući moduli se sklapaju prema sopstvenim potrebama. Sve procedure su date u izvornom kodu. U sklopu programa, kao primjer, dolazi i „MicroStar“ editor teksta sa propadajućim menijima i prozorima. Taj se editor može, na žalost, koristiti samo za pisanje, jer nije predviđena opcija za slanje teksta na štampač. Potrebna je memorija od minimalno 192 K.

## Turbo GamesWorks

**Borland International**  
Biblioteka potprograma

Zbirka izvornih kodova igrara pisanih u „Turbo Pascal“-u. Analizom se može dosta naučiti o tajnama i strategijama raznih igara i o teoriji igara uopšte, a procedure se mogu upotrijebiti za pisanje vlastitih igara. Priloženi su kompletni programi za šah, briđž i go-moku, spremni za prevođenje. Neophodno je najmanje 192 K RAM-a.

## Turbo Graphix Toolbox

**Borland International**  
Biblioteka potprograma

Ovim programom se „Turbo Pascal“-u dodaju nove rutine za rad sa grafikom visoke rezolucije, uključujući i kompleksnu poslovnu grafiku i prikaze. Omogućeno je odlaganje slika u memoriju i njihovo ponovno prikazivanje. Sve rutine su u izvornom kodu spremne za prevođenje. Potrebno je minimalno 128 K RAM-a.

## Turbo PROLOG Toolbox

**Borland International**

Zbirka vrlo korisnih rutina za programere na prologu. Sve rutine, a ima ih preko 80, date su u vidu izvornog koda, tako da se lako mogu ubaciti u sopstvene programe. Tu je i oko 40 programa koji demonstriraju njihovu primjenu. Priložene su rutine za generisanje poslovne grafike, konverziju podataka iz „Reflex“, „DBASE III“, „Lotus 1-2-3“, „Symphony“ programa, pravljenje korisničkih interfejsa sa raznim vrstama menija (pull-down, pop-up, tree, line, box), kompletan komunikacioni paket i parser generator. Ukoliko neko želi da razvija aplikacije koristeći „Turbo PROLOG“, ovaj paket će mu sigurno trebati. Da bi se koristio potrebno je 512 K RAM-a, dvije 360 K disketne jedinice i „Turbo PROLOG“ verzija 1,10 ili kasnija.

30 računari 29 • avgust 1987.

## Baze podataka

## dBase III

**Ashton-Tate**  
Relaciona baza podataka

Vjerovatno najpopularniji sistem za kreiranje i održavanje relacionog sistema baza podataka. Istovremeni rad sa do 10 datoteka. Datoteka može sadržati do milijardu zapisa. Maksimalna dužina zapisa je 4000 bajta i može sadržati do 128 polja fiksne dužine. Na raspolaganju je pet vrsta polja: Character, Numeric, Date, Logical i Memo. Rad sa programom je interaktivan ili programski. Za ljubitelje programiranja, na raspolaganju je bogat i snažan jezik četvrte generacije. Pošto se programska datoteka interpretira, „dBase III“ je izuzetan alat za razvoj aplikacija zasnovanih na bazama podataka. Postoji nekoliko kompajlera kojima se ti programi mogu ubrzati. Omogućeno je i lako kreiranje maski za unos i automatsko kreiranje izlaznih izvještaja. Program je snabdjeven „on-line helpom“ i dodatnom pomoći u interaktivnom radu. Bez izlaska u DOS moguće je pokrenuti druge aplikacije. Program zahtijeva minimum 256 K RAM-a, i dva 360 K disketna pogona. Tvrdi disk se preporučuje.

## dBase III PLUS

**Ashton-Tate**  
Relaciona baza podataka

Najnovija verzija popularnog „dBase III“ programa. Uvedena su mnoga poboljšanja u odnosu na prethodnika, a uz to je zadržana potpuna kompatibilnost. Nisu potrebni nikakvi konverzioni programi da bi se datoteka i programi kreirani „dBase III“ sistemom upotrebljavali sa PLUS verzijom. Kompatibilnost u drugom smjeru nije potpuna, jer je dodato oko 50 novih komandi i funkcija i tri nove vrste datoteka. Komande za sortiranje i indekisanje su optimizirane i brže su 2 do 10 puta. Poboljšanje je i interfejs prema korisniku u interaktivnom radu. U sklopu programa sada je i editor maski za unos. Podržana je kontrola pristupa datotekama i rad u mreži. U paketu je priložen generator aplikacija za automatsko pisanje programa. Na žalost, editor za pisanje programa je ostao isti, dakle ožban, pa se i dalje preporučuje korištenje nekog vanjskog editora. I ovaj program podržan je odgovarajućim kompajlerima, koji omogućuju stvaranje izvršne verzije aplikacije neovisno od „dBase III PLUS“ sistema. Program je snabdjeven „on-line helpom“, a pridodata je i tutorialni disketa.

## R:base System V

**Micronim**  
Relaciona baza podataka

Program je evoluirao iz R:base 5000 i urađen je po uzoru na „manifram“ sistem koji se naziva Relational Information Management i koji koristi NASA. Tim proizvođač Micronim želi da osvoji primat koji, za sada, drži „dBase III Plus“. Program je impozantne dužine. Dolazi na 11 disketa i na disku zauzima oko 3.6 Mbi! Za korištenje je neophodno imati bar 512 K RAM-a i tvrdi disk. Za rad u mreži potrebno je 640 K RAM-a. Program se sastoji od nekoliko, međusobno povezanih, cjelina. Njima se može pristupiti preko zajedničkog menija ili direktno. Zapis može sadržati do 800 polja, ukupne dužine do 4096 bajta. Polja, pored uobičajene definicije imena, tipa i dužine, imaju i attribute. Ako neko polje ima atribut „Key“, pretraživanje po tom polju biće znatno brže, bez potrebe za kreiranjem posebnih indeksnih datoteka. Preciznost računanja je na 15 cifara, u rasponu 10(e-307) do 10(e+307). Instrukcijski set je bogat i obuhvata oko 70 aritmetičkih, trigonometrijskih, statističkih i finansijskih funkcija. Najimpresivniji dio programa je Application Express koji automatski piše programe za korisnika. Na raspolaganju je nekoliko univerzalnih maski za unos i izlaznih izvještaja, a korisnik može definisati i sopstvene. Posebno korisna opcija je ukrašavanje podataka u tabeli (cross-tabulation). „R:base System V“ može da prima podatke iz „Lotus 1-2-3“ i „dBase III“ programa, ali i da ih eksportuje u „Lotus 1-2-3“ format. Verzija 1.1 zna to da uradi i za „dBase III“.



**Reflex****Borland International***Baza podataka*

Ovo, u stvari, i nije program za kreiranje i održavanje baza podataka u uobičajenom smislu riječi. Bolji naziv za „Reflex“ bi bio analitička baza podataka, jer na veoma jednostavan način omogućava analizu i prezentaciju podataka u bazi. Podaci mogu biti predstavljeni na pet različitih načina. Form View, služi za jednostavno definisanje strukture datoteke. Na raspolaganju je šest vrsta polja. List View daje kolonski prikaz sadržaja datoteke i omogućava brisanje, dodavanje, promjenu i sortiranje zapisa. Graph View daje grafički prikaz podataka u svim standardnim načinima. Crosstab View daje tabelarnu prezentaciju podataka u numeričkoj formi. Izvanredna je međupovezanost ovih prikaza. Ako se koriste „prozorske“ mogućnosti (do tri prozora), promjena parametara u jednom prozoru momentalno će se odraziti na ostale. Preostali način prezentacije je Report View i on predstavlja poseban program. Omogućava kreiranje izvještaja i niz drugih stvari, kao što su konverzija podataka iz „Lotus 1-2-3“, „Symphony“, „dBASE III“ i drugih standardnih programa. Moguća je konverzija i u suprotnom smjeru. Datoteka može sadržati do 32.500 zapisa. U zapisu može biti do 250 polja. Maksimalna dužina polja je 254 znakova. Programom se može upravljati tastaturom ili mišem. Komande se biraju iz menija. Za rad je potrebno bar 384 K RAM-a i dva disketna pogona.

slika se čuva u vidu datoteke elemenata. To omogućuje manje zauzeće memorije, ali i sporije iscrpanje prilikom pomjeranja. Slika može biti veća od jednog ekrana. Osim korištenja preddefiniranih elemenata, moguće je i crtanje slobodnom rukom. Dosta je bogat izbor uzoraka za popunjavanje zatvorenih površina. Ko nema naročite umjetničkih ambicija, može koristiti bogatu zbirku već urađenih slika. Ispis na štampaču je dosta dobar.

**MS Paint****Microsoft***Program za crtanje*

Program paint tipa (slika se zapisuje kao bitna mapa, a ne kao datoteka elemenata), koji radi pod „MS Windows“ sistemom. Obično dolazi u sklopu paketa, ali to zavisi od prodavca. Softver je veoma dobar za programe ovoga tipa. Na raspolaganju je dosta korisnog crtačkog alata. Podržane su sve opcije i mogućnosti koje se očekuju od jednog ovakvog programa, ali ima i specifičnosti. Ako se crteži prikazuju samo na ekranu, radna površina će biti čitav ekran, ali slika neće moći biti veća od ekrana. Ako želimo da pravimo štampane kopije crteža, radna površina će biti malo uža, poput papira u štampaču, ali ćemo se slobodno moći pomjerati i crtati po čitavoj dužini takvog „papira“, koja je višekratno veća od ekrana.

**Crtranje****Freelance****Lotus Development Corp.***Program za crtanje*

Program koji omogućava pravljenje efektnih crteža i grafova (mogu i dijapozitivi) za prezentacije ili neke druge prilike. Kome su dosadili standardni stubični ili kružni dijagrami, moći će dati mašti na volju i svoje izvještaje potkrijepiti zanimljivim i originalnim grafičkim prikazima. Na raspolaganju je biblioteka od oko 400 raznih simbola, a korisnik može definisati i sopstvene „Freelance“ prihvata dijagrame uređene programima „Lotus 1-2-3“, „Symphony“ i „Graphwriter“ i omogućava njihovu promjenu i doradu. Dijelovi slike mogu se pomjerati, kopirati, rotirati, odslikavati u ogledalu i slično. Mogu se crtati osnovni geometrijski likovi kao što su linije, lukovi, pravougaonici, krugovi itd. Tekst se može dodavati direktno, ili iz datoteke u ASCII formatu. Upravljanje je tastaturom, mišem, ili preko grafičke table. Podržani su svi najpoznatiji ploteri, matricni i laserski štampači. Za rad sa programom potrebno je minimalno 256 K RAM-a.

**Editori****IBM Professional Editor****IBM***Editor*

Jedan od starijih, ali još uvijek aktuelnih programa. Osnovna namjena mu je korištenje za pisanje programa. Editor je ekranski, ali je prilagođen manipulacijama sa programskim linijama. Linije mogu biti dugačke do 140 karaktera, što je zgodno za nazubljene programe. Manipulacije su vrlo lagane i elegantne. Snabdjeven je moćnim komandama za pretraživanje i zamjenu riječi u tekstu. Omogućen je pristup svim zakonima iz karakter seta. Podržana su dva tipa makros. Zahtjevi na hardver su jako mali. Program će biti zadovoljan i sa 64 K RAM-a u jednom disketnom jedinicom. Mana mu je da tekst koji obrađuje ne može biti duži od 64 K, ali se i to ograničenje može prevazići obradom u segmentima. U svakom slučaju, program neophodan svakom ko puno programira.

**GEM Paint****Digital Research***Program za crtanje*

Vrlo dobar program za crtanje „paint“ tipa. To znači da se slika čuva kao bitna mapa, a ne kao datoteka elemenata. Slika se crta na centralnom dijelu ekrana, a lijevo i desno su prostori za sličice crtačkog alata i uzoraka za popunjavanje. Alat je dosta raznovrstan i koristan. Može se crtati olovkom, četkicom ili sprejem. Na raspolaganju je 21 preddefinirani uzorak za popunjavanje. Korisnik može kreirati sopstvene uzorke. Na slike se može dodavati tekst u raznim fontovima. Ono što se ne može alatom, može se funkcijama iz menija. Slika se može isjecati, zumirati, lijepti, kopirati, održavati u ogledalu — dakle, sve ono što se i očekuje od jednog „paint“ programa.

Podržane su i boje, ali tek ako imate kolor karticu malo bolju od CGA.

**GEM Draw****Digital Research***Program za crtanje*

Radi pod GEM-om i koristi sve prednosti takve radne okoline. Omogućuje crtanje jednostavnijih slika. Kao i kod svih „draw“ programa,

**Grafičko projektovanje****AutoCAD****Autodesk Inc.***CAD program*

Najpopularniji CAD program za PC izuzetnih mogućnosti. Omogućava izradu crteža iz svih oblasti tehnike. Crtanje je brzo i jednostavno, potpomognuto velikim brojem opcija i komandi, a izmjene su lake i bezbolne. Slika se čuva u vidu datoteke, tako da može biti proizvoljne veličine. Podržano je automatsko skaliranje. Program raspolaze i mnoštvo „egzotičnih“ funkcija, kao što je automatsko zaobljavanje oštih ivica. Moguće je praviti „slide show“ za prezentacije (automatsko kontrolisano smjenjivanje slika na ekranu). Sa ovim programom se može uraditi gotovo sve (bar što se tiče crtanja), ali za dobre i efektne rezultate potrebna je i odgovarajuća oprema. Program zna da radi sa dva monitora, pri čemu jedan koristi kao grafički, a drugi kao tekstualni. Zbog obimnih preračunavanja preporučuje se matematički koprocesor. Tvrdi disk nije obavezan, ali se svakako preporučuje. Dodatna memorija konfigurisana kao RAM-disk bitno će ubrzati rad. Crteži se mogu odštampati na matičnom štampaču, ali je za profesionalni rad ploter neophodan. Za upravljanje programom može se koristiti tastatura, ali je podržan i čitav niz drugih spravnica, kao što su miš, grafička tabla, svjetlosna olovka i slično

## Energraphics

**Enertronics**

*CAD program*

Ovo nije samo CAD, već i program šire „crtačke“ namjene. Pomoću „Energraphics“ paketa mogu se grafički analizirati podaci i crtati poslovna grafika u dvije i tri dimenzije. Podaci se mogu prenositi iz svih najvažnijih programa za tabelarne poručne. Može raditi i osnovne statističke analize programa za i crtati matematičke funkcije u dvije i tri dimenzije. Pri tome zna za okretanje, zumiranje i uklanjanje skrivenih linija. Moguće je, naravno, raditi i CAD stvari u arhitekturi, mašinstvu, elektronicima i drugim oblastima. Raspolaze bogatom bibliotekom simbola i fontova kojoj korisnik može dodavati vlastite kreacije. Na kraju, moguće je praviti vrlo efektne prezentacije (slide show). Za to postoji poseban komandni jezik. Crteži se mogu oštampati i nacrtati na većini poznatih štampača i plotera. Za rad zahtijeva bar 128 K RAM-a i dvije 360 K disketne jedinice.

## HiWIRE

**Wintek**

*CAD program*

Nakon popularnog „smARTWORK“-a, programa za dizajniranje štampanih pločica, Wintek je nedavno izbacio svoj novi proizvod koji služi za kreiranje elektroničkih šema. Novi program donosi i dosta novosti. Više je prilagođen radu sa mišem, komande se biraju iz menija, a podržani su i prozori. Program raspolaze bogatom bibliotekom od preko 700 elektroničkih elemenata, uključujući i mikroprocesore. Korisniku je dopušteno da i sam kreira sopstvene simbole i ubacuje ih u biblioteku. Simboli se mogu premijestati, kopirati i rotirati. Dodatna pomoć pri projektovanju je automatsko pravljenje liste korištenih elemenata i proračun cijene iskorisćenog materijala. Podržana je kolor grafika. Šeme se mogu veoma kvalitetno oštampati na većini matičnih štampača i plotera. Program će raditi sa minimalno dvije 360 K disketne jedinice i 320 K RAM-a.

## smARTWORK

**Wintek**

*CAD program*

Program za profesionalno i poluprofesionalno dizajniranje štampanih pločica. Pločice mogu biti sa jednostranom ili obostranom štampanom veličine do 10x16 inča. Istovremeno može biti prikazana jedna ili obe strane štampe. Program „razumije“ električne veze i sam traži najkraći put između dva kontakta. Veze se ne preklapaju. Na raspolaganju su dvije širine veze. Prikaz je u boji ili crno-bijel. Crtež može biti veći od jednog ekrana. Komande se zadaju u komandnoj liniji. Može se koristiti i miš. Kreirani crtež može se oštampati na matičnom štampaču ili na ploteru. Program zahtijeva dva disk pogona i minimalno 192 K RAM-a.

## Integrirani paketi

### IBM Assistant Series

**IBM**

*Paket za obradu podataka*

Programi starije generacije. Paket obuhvata četiri programa. „Filling Assistant“ je mala baza podataka. „Reporting Assistant“ je pomoć pri objedinjavanju podataka i pisanju izvještaja. „Graphing Assistant“ je namijenjen crtanju poslovne grafike. Rezultati su vrlo dopadljivi. „Writing Assistant“ je dosta zgodan procesor teksta, ali bez opcije poravnavanja desne margine. Stranice teksta su jasno razgraničene i jako zgodno prikazane. Prihvata tekstove samo u sopstvenom formatu. Programi mogu raditi pod „TopView“ i „MS Windows“ sistemima.

32 računari 29 • avgust 1987.

## CA Executive

**Computer Associates International, Inc.**

*Integrirani paket za obradu podataka*

Ovo je, u stvari, kompletna radna stanica. Program je impresivne veličine. Dolazi na 16 disketa i na disku zauzima oko 4.5 Mb. Sa radnog stola se biraju prozori sa programima. Podržane su boje. Tabelarni kalkulator je malo nespretna za korištenje, ali tabela može biti u boji. Tekst procesor je sasvim dobar. Baza podataka je standardna za ovaj tip programa. Posjeduje jednostavni komandni jezik. Na raspolaganju su programi za kreiranje maski za unos podataka i automatsko generisanje izvještaja. Komunikacioni dio omogućuje povezivanje na velike sisteme (npr. IBM 3270) i konverziju podataka za CA-Universe — relaciju bazu podataka koja radi na „mainframe“ sistemima. Grafički dio paketa je odličan. Crta 2D i 3D poslovnu grafiku i omogućava pravljenje „slide show-a“ za prezentacije. Tu su još editor za pisanje programa, note-pad i drugi zabilježnik i tutorial, koji se tretira kao zaseban program. Vjerovatno ne treba ni napominjati da se podaci mogu jednostavno razmijeniti između programa. U svakom trenutku na raspolaganju je i on-line help. Mada postoje bolji pojedinačni programi, snaga ovog sistema je u potpunju integraciji. Za potpuno korištenje njegovih prednosti potrebno je imati tvrd disk.

## Framework I

**Ashton-Tate**

*Integrirani paket za obradu podataka*

Ovaj izvanredan program objedinjuje tabelarni kalkulator, bazu podataka, tekst procesor, poslovnu grafiku i komunikacije. Rad se zasniva na prozorima, koji se u ovom slučaju zovu okviri (frames), a po filozofiji je jako sličan radu sa GEM-om. Okviri se mogu preklapati, pomjerati i može im se mijenjati veličina. U okvirima je moguće raditi sa bilo kojom vrstom dokumenta (tabela, datoteka, tekst, grafika). Svi dokumenti se tretiraju na isti način, tako da se sve opcije iz tekst procesora (npr. razni tipovi ispisa) mogu sasvim ravnopravno koristiti i u bazi podataka ili tabelarnom kalkulatoru. Tekst procesor je prilično dobar i omogućuje prikaz u bold, italic i underline načinu. Tabelarni kalkulator je izvanredan i lagan za korištenje. Baza podataka je standardna za ovaj tip programa. Grafički dio paketa zaslužuje sve pohvale. Graf se crta prilično brzo i moguće mu je po volji mijenjati veličinu. Komande se izvršavaju pritiskom na funkcijske tipke ili se biraju iz menija. Na raspolaganju je i čitav programski jezik nazvan FRED, koji omogućava potpunu kontrolu sistema i kreiranje sopstvenih aplikacija vođenih menijima. Program je snabdjeven on-line helpom i tutorialom. Nedostatak u odnosu na slične programe je relativno mala veličina pojedinačnih dokumenata.

## Framework II

**Ashton-Tate**

*Integrirani paket za obradu podataka*

Poboljšana verzija koja je otklonila mnoge mane programa-prenosa. Poboljšan je interfejs prema korisniku, a mnoge operacije koje je bilo potrebno programirati u FRED programskom jeziku sada se biraju iz menija. Podržano je korištenje virtualne memorije, tako da sada dokumenti mogu biti daleko veći. Tabela može, na primjer, biti veličine do 32000x32000 ćelija. Pridodata je i opcija provjere pravopisa (spelling checker) koja se može koristiti kako u tekst procesoru, tako i u tabeli ili bazi podataka. I dalje su podržane makro instrukcije. Tehnikom „ukvirivanja“ i zahvaljujući jednakom tretiranju dokumenata, veoma lako se dobijaju efektni izvještaji u kojima se kombinuju tekst, tabela, izvještaji iz baze podataka i poslovna grafika.

## Lotus 1-2-3

**Lotus Development Corp.**

*Integrirani paket za obradu podataka*

Nezabavljivo ime u oblasti poslovno orijentisanih programa. Sastoji se iz tri dijela: tabelarni kalkulator, baza podataka i poslovna grafika. Tabelarni kalkulator je svakako najjači dio paketa i ima najviše zasluga za njegovu popularnost. Dimenzije table su 256x2048 ćelija. Komande se

biraju iz menija. Podržane su makro instrukcije. Baza podataka je standardno jednostavna, bez posebnih prednosti. U toku rada sa tabelom, moguće je odgovarajući grafik vidjeti na ekranu. Da bi se vidio i na papiru, potrebno je preći u poseban program — PrintGraph. Može se birati između porednog i uzdužnog štampanja. Tekst procesor ne ulazi u sastav paketa. Iako postoje programi za integrisanu obradu podataka sa većim mogućnostima, „Lotus 1-2-3“ ipak i dalje ostaje kralj među programima ove vrste.

## Lotus 1-2-3 Release 2

Lotus Development Corp.

*Integrirani paket za obradu podataka*

Nova, usavršena verzija popularnog „Lotusa“. Nisu poduzeti neki bitniji zahvati, koji bi se uočili na prvi pogled. Pažnja je posvećena usavršavanju samih funkcija programa. Tabela je povećana do veličine 4096 x 256 ćelija. Brojevi mogu biti u opsegu od  $10e^{-99}$  do  $10e^{99}$  za ulazne podatke, i  $10e^{-308}$  do  $10e^{308}$  za rezultate operacija, i mogu biti dugi do 240 znakova. Dodane su i neke nove komande, kao što su komande za rad sa stringovima, ili komande za rad sa matricama. Usavršeni su i makroi, koji su i inače bili jaka strana ovog programa. Sada je moguća direktna razmjena tabela sa „Symphony“ programom. Podržana je dodatna memorija po Lotus/Intel standardu i matematski koprocesor. Za rad sa programom potrebno je minimalno 256 K RAM-a.

## Open Access II

Software Products International

*Integrirani paket za obradu podataka*

Jedan od boljih „programa za sve“. Objedinjuje funkcije tabelarnog kalkulatora, baze podataka, tekst procesora, grafike, komunikacija i planiranja vremena. Tabelarni kalkulator je manje-više standardan. Tabela je veličine 3000 x 216. U ovakvim paketima baza podataka je više reda radi, ali ne i u ovom slučaju. Baza je relacionala, sa sopstvenim programskim jezikom koji se naziva „Programmer“. Mogu se praviti maske za unos podataka. Tekst procesor može u tekst izvještaja uključivati podatke iz tabele, ili baze, i grafiku. Postoji i opcija za pisanje cirkularnih pisama (mail merge). Grafički dio paketa je, takođe, za svaku pohvalu. Crta 2D i 3D postovnu grafiku, pa čak i površine na osnovu funkcija. Dio za planiranje vremena omogućava vođenje kalendara, ima sat sa alarmom, automatski telefonski imenik, notes za zabilješke i priručni kalkulator. Komunikacioni dio paketa podržava VT 100, XMODEM i KERMIT protokole, kriptografsku zaštitu podataka i postavljanje parametara modema. Rad sa programom je preko menija. Podržani su prozori i boje.

## PC Four

Pison

*Paket za obradu podataka*

Poznati XCHANGE paket sa QL-a prerađen za PC. U paketu dolaze četiri programa i tutorijal disketa. Programi mogu raditi nezavisno i međusobno razmjenjivati podatke. „Abacus“ je tabelarni kalkulator bez posebnih prednosti. „Easel“ je program za crtanje 2D i 3D poslovne grafike, dosta dobrih karakteristika. Priložena je i GEM verzija. „Archive“ je programski jezik za relacionalu bazu podataka, sa nekim poboljšanjima u odnosu na QL verziju. Ima fantastičan editor za pisanje programa. Procesor teksta „Quill“ je stabilniji već na QL-u jer nije dosljedno primijenjena filozofija WYSIWYG (What You See is What You Get). Na ekranu se direktno ne vide posebni načini ispisa (bold, italic, underlined, subscript, superscript), već su drugačije obojeni. Ostale karakteristike su veoma dobre. Posebna prednost je način kontrole štampača i mogućnost pravljenja translacije tabele znakova.

## Symphony

Lotus Development Corp.

*Integrirani paket za obradu podataka*

Symfonia je trebala da naslijedi „Lotus 1-2-3“, ali i pored očiglednih prednosti nikad nije doživjela popularnost svog prethodnika. U paketu su sada integrirani i procesor teksta i komunikacioni program. Poboļjšani su

algoritmi za rekalkulaciju, usavršen grafički dio paketa i dodani prozori. Može ih biti maksimalno četiri. Ukoliko je u prozorima grafika, promjenom podataka, pored rekalkulacije, automatski se ažuriraju i grafički prozori. Komande se biraju iz menija kao i u „Lotusu“. Komandni set je proširen, pa postoje dva menija. Da se u grafiku odštampa, i dalje je potrebno preći u poseban program. Izgled grafike i kontrola štampača bolji su već u „Lotusu“. Minimalna konfiguracija na kojoj će simfonia „prosivirati“ je IBM PC kompatibilan računar sa jednom 360 K disketnom jedinicom, 320 K RAM-a i DOS verzijom 2.0 ili kasnijom. Naravno, da bi se vidjela grafika, potrebna je i neka grafička kartica.

## Kompajleri

Clipper

Natucket

*dBase III kompajler*

Najpopularniji kompajler za „dBase III“ sistem. Omogućava stvaranje izvršnih programa (sa .EXE nastavkom) na osnovu programskih datoteka radenih u „dBase III“ programskom jeziku. Treba obratiti pažnju na verziju programa, jer neke verzije nemaju direktno implementirane sve funkcije. Neke nisu uopšte podržane (join), a neke su simulirane programima (total). Kompajler, međutim, raspolaze i nekim dodatnim komandama i funkcijama. Sve datoteke treba nanovo indeksirati, jer indeksi fajlovi nisu kompatibilni. Starije verzije kompajlera su jednodozne. Verzija jesen '86 je potpuni „dBase III PLUS“ kompajler. Podržane su sve funkcije, ali su indeksi fajlovi i dalje nekompatibilni. Podržan je rad u mreži. Kompajler je sada dvoprolazan i znatno brži. Moguće je povezivanje sa objektnim modulima radenim u drugim programskim jezicima. Kao linker koristi se „Plink86“. Kompiliranjem programi se izdvajaju 2 do 20 puta

## Komunikacije

Crosstalk XVI

Microstuf

*Komunikacioni program*

Najbolje je prodavani komunikacioni program, svakako ne bez razloga. Služi za komunikaciju dva računara putem modema ili preko lokalne veze, kada su računari jednostavno povezani preko serijskog interfejsa. Na taj način se resursi jednog računara (disk, štampač) mogu koristiti sa drugog računara. Ako su računari snabdjeveni sa po dva serijska kanala, može se napraviti jednostavna mreža u kojoj svaki računar može komunicirati sa svojim „lijevim“ i „desnim“ susjedom. Prilikom slanja datoteka, provjerava se ispravnost prenosa. Podržani su Kermit i Xmodem protokoli i emulacija VT-100 terminala. Parametri komunikacija biraju se iz menija. Program raspolaze sopstvenim komandnim jezikom i omogućava pisanje komandnih datoteka za specijalne namjene. Prilagodbe je da radi kao pozadinski proces, uporedo (istovremeno) sa drugim programima. Tako može da radi svoje uobičajene poslove dok „Crosstalk“ prima, npr., elektronsku poštu. Program radi i sa „TopView“ sistemom. Zahtjeva minimalno 96 K RAM-a.

## Kontrola štampača

Fancy Fonts

Softcraft Inc.

*Kontrola ispisa na štampaču*

Program omogućava kaligrafski ispis na matricnom štampaču bilo koje ASCII datoteke. Ima 20 preddefiniranih fontova. Fontovi se mogu mijenjati, i u tekstu se, posebnim komandama, određuje šta će biti ispisano na koji način. Fontovi se referiraju redoslijedom učitanjavanja, pa se mogu praviti razne kombinacije. Korisnik ima mogućnost da mijenja postojeće, ili da kreira sopstvene skupove znakova. Ispis na štampaču je veoma kvalitetan i



precizan. Na žalost, korisnik nije u mogućnosti da prije štampanja, na ekranu, vidi ni približan oblik teksta.

## Letrix

Hammerlab Corp.

Kontrola ispisa na štampaču

Rezidentni print procesor koji omogućava veoma kvalitetan ispis na matricnim štampačima raznih vrsta. Štampati se može u 20 predefinisanih fontova, koji uključuju grčka i hebrejska slova, ćirilicu (rusku) i posebne matematske simbole, a korisnik može definisati i sopstvene. Program se poziva sa jednim, ili više fontova, i ostaje u memoriji sve dok se računar ne resetuje. Zauzima 40 do 70 Kb. Svi izlazi na štampač sada idu preko njega. Istovremenim pritiskom na Shift i Return tipke poziva se meni pomoću koga se mogu postaviti dodatni parametri za rad programa. Postoje i posebne "backslash" (\) komande, koje će program otkriti u tekstu, i koje mogu poslužiti za dodatne efekte. Pored ovih, "Letrix" razumije i standardne "WordStar" komande za posebne načine ispisa (bold, underline, superscript, subscript). Kada poravnava desnu ivicu, dok proporcionalno štampanje, ne reformatira paragraf, već samo umeće blanko karaktere. Sa "Letrix"-om i "WordStarom" može se štampati i dvokolonski tekst. Program daje izvanredne štampane rezultate, ali i usporava štampanje. Mana mu je i što ne koristi bafer štampača, pa je računar "blokiran" sve dok se štampanje ne završi.

## Print Shop

Broderbound Software

Kontrola ispisa na štampaču

Program za kreiranje i ispis pozivnica, zahvalnica, plakata, zaglavija pisama i slično. Na raspolaganju je bogata biblioteka slika i fontova. Korisnik može mijenjati postojeće simbole, ili kreirati sopstvene. Slova mogu biti ispisana u solid, outline i 3D stilu. Ostavljena je i mogućnost slobodnog šaranja po ekranu (opcija Screen Magic), kojom se mogu postići zanimljivi efekti. Rad sa programom se odvija preko menija. Forma za ispis kreira se od slika i iz biblioteka i teksta. Pozivnice i zahvalnice mogu biti uokvirene raznovrsnim okvirima. Prije štampanja, na ekranu je moguće vidjeti izgled kreacije. Ispis je dosta kvalitetan.

## Sideways

Computer Associates International

Kontrola ispisa na štampaču

Koliko puta ste se mučili pokušavajući da veliku tabelu strpate na širinu A4 papira. Sideways rješava upravo takve probleme štampajući po dužini papira. Nova verzija programa (3.0) može, uz to, da štampa u 9 različitih fontova i omogućava izbor kvaliteta štampa. Niz drugih opcija, koje se biraju iz menija, pružaju dodatnu kontrolu ispisa. Program prihvata datoteke sa nastavkom .PRN i prepoznaje odgovarajuće kontrolne kodove za izabran štampač.

## Set-FX+

Soft Style

Kontrola ispisa na štampaču

Program koji može biti od koristi vlasnicima Epson i kompatibilnih štampača. Pomoću dva menija, moguće je postavljajući svih parametara štampača. Osim ovih uobičajenih zahvata, omogućava ispis teksta raznim vrstama fontova. Fontovi mogu biti korišteni na dva načina. Ukoliko štampač ima dovoljno RAM-a, primjenjuje se "downloading" tehnika, kojom se standardni karakter set iz ROM-a štampača zamjenjuje setom u RAM-u. Takvo štampanje je bitno brže od drugog načina, kada se slova štampaju u grafičkom modu. Korisnik može da mijenja postojeće, ili da definiše sopstvene setove karaktera, što je posebno bitno za naše prilike. Definišanje karaktera je preko matrice. Osim fontova, mogu se koristiti i grafički karakteri. Omogućeno je i štampanje po dužini papira (pivot printing).

34 računari 29 • avgust 1987.

## Kopiranje programa

### Copy II PC

Central Point Software

Program za kopiranje

Dok se ne pojave prvi rezultati domaće inventivnosti na polju skidanja zaštite u cilju "pravljenja arhivskih kopija", sasvim dobro će poslužiti i ovaj program. "Copy II PC" se koristi kao poboljšana Diskcopy komanda iz DOS-a (ima istu sintaksu), s tim što izlazi na kraj sa dosta zaštite. Može se čak i ubaciti u DOS umjesto standardne komande. To, međutim, nije preporučljivo, jer je kopiranje nezštićenih disketa bitno sporije. Kako se zaštite stalno usavršavaju i mijenjaju, ovi programi zastarjevu, pa je potrebno nabavljati nove verzije, kako bi se bilo u toku.

### Copywrite

Quaid Software Limited

Program za kopiranje

Programi ove vrste stekli su u našim krajevima neprolaznu popularnost. Još na samom početku računarskog buma. Oni su, naravno, namijenjeni samo pravljenju arhivskih kopija originalnih disketa. "Copywrite" ima lijepo urađen radni ekran. Opcije se biraju iz menija. Može da izade na kraj sa dosta zaštićenih programa, ali ga kopije uradne drugim koji programima znaju izbaciti iz takta. Zaštite se, s vremena na vrijeme, mijenjaju, pa se pojavljuju i nove verzije programa. Za sada nema programa koji bi, garantovano, sve kopirao.

### Nokey

Central Point Software

Pokretanje zaštićenih programa

Koristi se za pokretanje zaštićenog softvera sa kopija na tvrdom disku, ili arhivskih (piratskih?) kopija na disketama. Uz pomoć ovog programa pradike "Electric Desk", "Prokey", "Symphony", "Sidekick", "Think Tank", "Lotus 1-2-3", "dBASE III", "Framework 1.0" i drugi. "Nokey" je rezidentan i treba ga pokrenuti samo jedanput, prije rada sa nekim od ovih programa. Nakon toga ostaje u memoriji i djeluje sve dok se računar ne resetuje. Ponekad mogu nastupiti problemi u radu sa drugim rezidentnim programima. Na primjer, postaje rada sa "SideKick"-om, obično nije moguće pokrenuti "Prokey" ili "Framework". Rješenje je da se ponovo starta "Nokey". Ako neki programi, ipak, neće da rade, razlozi mogu biti različiti — od nestandardnog disk kontrolera (eventivni su problemi u radu sa tvrdim diskom koji se umeće kao kartica), do neodgovarajuće verzije samog "Nokey"-a. Ponekad je, osim startanja "Nokey"-a, potrebno poduzeti i neke dodatne korake komandom Assign. Kako se zaštite, s vremena na vrijeme, mijenjaju, tako se javljaju i nove, dopunjene, verzije ovakvih programa.

### Zerodisk

Quaid Software Limited

Pokretanje zaštićenih programa

Sretni vlasnici tvrdog diska kad-tad se suoče sa problemom rada sa zaštićenim programima koji uporno traže "originalne" diskete, kada se pokrene njihova radna kopija. Zerodisk omogućava da se neki od tih programa (ne svi) bez problema (i stalnog korišćenja "master" diskete) pokrenu i sa tvrdog diska. Ovaj program izlazi na kraj sa "Lotus 1-2-3 R1A", "Prokey 3.0", "The Harvard Project Manager", "R-base 4000", "Think Tank" i nekim drugim programima. Zerodisk je specijalna verzija drajvera koja preusmjerava neke sistemske pozive. Instalira se na taj način što se u datoteku CONFIG.SYS ubaci linija device = zerodisk.drv. Kako se, s vremena na vrijeme, mijenjaju zaštite, tako se pojavljuju i nove verzije ovog, i sličnih, programa.

## Logičke igre

### Psion Chess

Psion

Igra

Mada PC nije mašina za igranje, i ne može se pohvaliti dobrim programima te vrste, ovaj program, ipak, treba spomenuti. Prvobitno napravljen za QL-a, sa kojim je, u svojoj klasi, osvojio i prvenstvo Evrope 1984. godine, sada se pojavio i u verziji za PC. Bio je jedan od prvih programa koji su imali prikaz tablice u 3D. Ovo je i jedna od rijetkih igara koju će moći koristiti i vlasnici Hercules grafičke karte. Program pruža kvalitetnu igru na 10 nivoa, sa brzinama poteza od 2 sekunde do 4 minute. Postoje i opcije za jednako vrijeme i neograničeno vrijeme (pogodno za dopisni šah). Zna da rješava mat u 1 do 8 poteza. Pored neospornih igračkih kvaliteta, program pruža i dosta komfora. Brojne opcije omogućuju kontrolu toka partije, zamjenu pozicija, nudenje najboljih poteza, ponavljanje kompletne partije, ispis poteza na štampaču itd. Na raspolaganju su dvije vrste prikaza — 2D i 3D. Figure u 3D prikazu su veoma lijepo dizajnirane i meko se pokreću. Poteze nije potrebno upisivati već se figure pomjeraju kursoriskim tipkama.

## Matematika

### MathCAD

MathSoft

Matematički program

Da li ste ikad poželjeli da po ekranu vašeg PC-a pišete slobodno, kao po školskoj tabli, tekst, formule, grafike...? Upravo to vam omogućava MathCAD. Dovoljno je napisati, uobičajenom matematskom notacijom neku formulu, ili jednačinu, bilo gdje na ekranu, a program će automatski izračunati rješenje i prikazati ga u vidu brojeva ili grafika. Promjenom bilo kog parametra, automatski se vrši rekalkulacija. Može se sasvim slobodno dodavati tekst kao objašnjenja, a zatim se sve otšampati kao jedinstveni dokument. Program prihvata kako realne, tako i kompleksne brojeve, i vrši automatsku konverziju jedinica i dimenzionu analizu. Ugrađen je čitav niz funkcija, među kojima i one za integriranje i diferenciranje, a korisnik može definisati i sopstvene. Za rad je potrebno minimalno 512 K RAM-a.

## Obrada teksta

### Final Word

Mark of the Unicorn

Processor teksta i editor

Omogućava istovremeni rad sa deset tekstova koji se čuvaju u posebnim baferima. Prelazak iz bafera u bafer i prenos podataka su vrlo jednostavni. Ekran se može podijeliti na dva dijela (prozora), u kojima se može raditi sa istim, ili sa različitim tekstovima. Velika prednost ovog programa je što gubitak teksta sa kojim se radi čini praktično nemogućim, jer u kratkim vremenskim razmacima tekst automatski spasava u posebnu „swap“ datoteku na disku. Iznenađni pad sistema neće, dakle, uništiti višesatovni trud. Prilikom startanja programa automatski se inicijaliziraju svi baferi, tako da možemo nastaviti rad tamo gdje smo stali, bez ponovnog učitavanja teksta. U tekst ne ubacuje nikakve kontrolne karaktere, pa se može koristiti kao editor za pisanje programa. Podržane su makro instrukcije. Komande se biraju iz menija. On-line help postoji, ali je jako slab. Raspoložaće osnovnim komandama za formatiranje teksta, čiji se rezultat odmah vidi na ekranu. Pored toga, raspoložaće vrlo moćnim jezikom za formatiranje, čije se komande umeću u tekst, a rezultat je vidljiv tek kada se tekst odštampa. Podržano je formatiranje teksta sa hijerarhijskom

strukturom, automatsko formiranje sadržaja, indeksa pojmova i fusnota. Ove karakteristike ga čine pogodnijim za pisanje dužih, nego kraćih tekstova, jer tada njegove prednosti dolaze do punog izražaja.

## GEM Write

Digital Research

Tekst procesor

Tekst procesor koji radi pod GEM-om. Zasnovan je na tekstu procesoru „Volks Writer“. Mogu se koristiti razni načini ispisu i razni tipovi slova. Program prihvata grafiku urađenu drugim GEM programima. Nema drastičnih ograničenja na dužinu teksta. Pokretanje po tekstu i manipulacija sa njim su dosta brze. Ispis na štampaču je dosta dobar, i tekstovi napisani i obrađeni ovim programom izgledaju zaista efektivno.

## Manuscript

Lotus Development Corporation

Processor teksta

Program, prije svega, namijenjen onima koji se puno bave pisanjem tehničke dokumentacije. Omogućava miješanje teksta, matematičkih formula, tabela i grafike u istom dokumentu. Dokumenti mogu biti visoko strukturirani, maksimalne dužine do 800 strana. Precabjanje sa jednog dijela teksta na drugi je zapanjujuće brzo, zahvaljujući tome što se tekst ne čuva u vidu sekvencijalnog dotjeka, već se dijelovima teksta pristupa direktno preko pointera. Unutar jednog strukturnog nivoa (npr. poglavlja) može se vršiti sortiranje ili automatska numeracija podnivoa. Ima poseban editor za crtanje tabela i poseban komandni jezik za precizan i jednoobrazan ispis matematičkih formula. Grafika se u dokument ubacuje tek u trenutku štampanja i nije vidljiva u toku editovanja. Posebnom opcijom, prije štampanja, može se vidjeti konačan izgled kompletne stranice. Prilikom štampanja koristi se i tekstualni i grafički način. Tekst se, zbog brzine, ali i zbog kvaliteta, štampa standardnim fontovima kojima raspoložaće štampač. Grafičkim načinom štampaju se formule i grafika. „Manuscript“ prihvata ekstove u ASCII, DCA i Think-Tank formatu. Takođe, prihvata slike i table iz „Lotus 1-2-3“ i „Symphony“ programa. Automatski pravi indeks pojмова, slika i fusnota i sadržaj. Posebna opcija je uporedivanje dvaju verzija jednog dokumenta. Komande se biraju iz menija ili direktno sa par pritiska na tipke. Program zahtjeva minimalno 512 K RAM-a i tvrdi disk.

## Multimate

Multimate Int'l, Corp.

Processor teksta

Dobar procesor teksta profesionalnih karakteristika. Pomoću njega kreiramo biblioteku dokumenata. Imena dokumenata su dužine do 20 karaktera, tako da se može lijepo i komotno opisati šta dokument sadrži. Uz naslov, unose se i druge generalije, kao što su naziv autora, datum, ključne riječi za kasnije prepoznavanje i slično. Komande se biraju iz menija i uz pomoć funkcijskih tipki. Neke manipulacije sa tekstom su malo čudne (brisanje i umetanje slova, npr.), ali se korisnik brzo privikne. U tekstu su vidljivi kontrolni znaci kao što su CR ili TAB. Ima ugrađenu opciju za slanje cirkularnih pisama (Mail Merge). Kreiranje datoteke podataka je, istina, malo komplikovano, jer svaki podatak mora biti u posebnoj liniji i mora imati ime polja, ali je to olakšano dobrom podrškom makroa. Može se čak formirati biblioteka najčešćih dijelova teksta ili fraza. Program raspoložaće i jednostavnim matematičkim operacijama. Prilikom pisanja izvještaja više nije potrebno posezati za kalkulatorom ili nekim „spreadsheet“ programom da bi se u tabelama izračunali zbrojevi po kolonama i redovima. To će sada uraditi sam tekst procesor. Ugrađena je i pravilna pravopisna.

## MS Word

Microsoft

Tekst procesor

Tekst procesor velikih mogućnosti. Za one sa skromnijim zahtjevima, na raspolaganju su sve standardne opcije. Komande se biraju iz menija na dnu ekrana. Većina ima i svoje podmenije. Oni sa većim zahtjevima moći

će iskoristiti i napredne opcije kao što su: razni načini ispisa (bold, italic, underline, strikethru, superscript, subscript), razni fontovi, makro instrukcije i i makro definicije formata, automatsko numerisanje fusnota i pravljenje indeksa, rad sa prozorima itd. Bez dodatnih programa mogu se slati cirkularna pisma (mail merge). Ugrađena je i provjera pravopisa (spelling checker), ali samo za engleski jezik.

## MS Write

Microsoft

*Tekst procesor*

Program radi pod sistemom „MS Windows“ i u potpunosti koristi prednosti takve radne okoline. U ovu okolinu od distributera, ponekad dolazi i u sastavu tog paketa umjesto MS Paint programa, a nekada su oba programa uključena. Rad sa programom je veoma lagan, a kreirana dokumenta su atraktivna. Moguće je koristiti razne vrste fontova i slobodno ih miješati. Na raspolaganju su i uobičajeni načini ispisa — bold, italic, underline, superscript, subscript. Moguće je miješanje grafike i teksta, ali ne u istoj liniji. U samom programu može se mijenjati veličina i položaj grafike. Dokument se može sačuvati i kao tekst bez grafike. Prihvata i tekstove urađene drugim programima.

## NewWord 3

NewStar Software

*Tekst procesor*

Jedan od „WordStar“ kompatibilnih procesora teksta, koji donosi i neke novine u odnosu na uzor. Ima ugrađen „pop-up“ kalkulator sa aritmetičkim, logaritamskim i trigonometrijskim funkcijama. Jednačina i rezultat se tako mogu prenijeti u tekst. Osim toga, program zna da pravi totale po redovima i kolonama, numeričkih podataka unesenih u tekst. Podržani su i makroi. Ima šest predefinisanih, a 36 korisnik može definisati prema svojim potrebama. Ugrađen je program za provjeru pravopisa u toku rada, ali je priložen i još jedan poseban program sa istom funkcijom. U indeksa pojnova mogu biti uključene sve riječi iz dokumenta, sve riječi, izvještaji od svih liste, i samo određene riječi i fraze. Automatsko pravljenje sadržaja podržano je i nekim specijalnim „komandama sa tačkom“ (korisnici „WordStara“ znaju o čemu se radi). Tu su još mogućnosti poziva DOS komandi bez izlaska iz programa, konverzija bloka znakova u velika ili mala slova, izračunavanje broja znakova od početka teksta do kurzora pregled dokumenta u formi u kojoj će biti ođštampan. Ipak, Merge-Printing je, vjerovatno, najvažnija opcija. U toku štampanja može se manipulirati tekстом koristeći uslovne komande, tako da cirkularna pisma mogu izgledati kao da su specijalno pisana za svakog primaoca. Uz sve to program je veoma brz, posebno u operacijama sa diskom.

## VolksWriter Scientific

Lifetree Software Inc.

*Procesor teksta*

Procesor teksta specijalno namijenjen principu tekstova sa dosta matematičkih simbola i formula. Poštovan je princip W—Y—S—I—W—Y—G (What You See Is What You Get), tj. ono što se vidi na ekranu, dobija se i na štampaču. Podržano je nekoliko vrsta slova i mnoštvo specijalnih znakova. Komande i opcije se biraju iz menija i uz pomoć funkcijskih tipki. Tekst se obrađuje po stranicama. Nije moguće obrađivati tekstove pisane uz pomoć drugih programa. Štampanje je u grafičkom režimu. Program je u snabdjeven dobrim tutorijalom, koji je sasvim dovoljan za početak rada. Tutorijal se može i ođštampati, pa se dobije i uputstvo za upotrebu. Da bi se radilo sa programom, potrebno je bar 256 K RAM-a.

## WordStar

Micropro

*Tekst procesor*

Ime ovog programa postalo je sinonim za procesore teksta. U vrijeme kada je nastao bio je zaista dobar, vjerovatno i najbolji, program te vrste. Kasnije se pokazalo da se mogu napraviti i bolji tekst procesori, ali „WordStar“ je i dalje bio najpopularniji, tako da je praktično postao standard. Raspolaze bogatim setom instrukcija koje se biraju iz menija u gornjem dijelu ekrana i omogućuju korisniku da sa svojim tekстом uradi

gotovo sve. Samo nabiranje svih mogućnosti zauzelo bi dosta prostora. Nema, naravno, grafičkih mogućnosti, ali to su, ionako, osobine novijih programa. Može obrađivati velike dokumente, ali je skakanje sa jednog kraja takvog dokumenta na drugi prilično sporo, jer program dosta komunicira sa diskom. Tekstovi mogu biti formatirani, a mogu se čuvati i u ASCII formatu, pa se program može koristiti kao editor. Umjesto direktnog prikaza raznih načina ispisa i drugih specijalnih opcija, na ekranu se vide kontrolni znaci. Zbog toga je tekst pisan u „WordStar“-u prilično nečitak sve dok se ne oštampa. Štampa se sa diska, a ne iz memorije. Ovakav pristup sve više gubi bitku sa novim WYSIWYG (What You See Is What You Get) procesorima teksta koji omogućavaju puno bolju kontrolu izgleda dokumenta, pa čak i miješanje teksta i grafike. Možda je vrijeme za novi standard?

## WordStar 2000

Micropro

*Tekst procesor*

Poslije legendarnog „WordStar“-a, koji je počeo da gubi bitku sa novijim procesorima teksta, Micropro je izbacio ovaj program s ambicijom da postavi novi standard. Program donosi dosta poboljšanja u odnosu na prethodnika. Sada se može raditi sa dva teksta istovremeno, tako što se ekran podijeli na dva prozora. Moguće je kopirati tekst iz prozora u prozor. Komande se i dalje biraju iz menija koji se nalazi u gornjem dijelu ekrana. Dizajn programa je malo dotjeraniji. Uvedena je nova klasa blok komandi, kojima se dijelovi teksta mogu sortirati ili sa njima raditi matematičke operacije sabiranja, množenja ili dijeljenja. To bitno olakšava pisanje izvještaja koji sadrže tabele sa numeričkim podacima. Makroi mogu biti dužine do 560 znakova i mogu se čuvati kao posebne datoteke. Takođe, postoje posebne datoteke sa definicijama formata. Podaci za cirkularna pisma mogu se uzimati iz interne datoteke sa pravim poljima ili iz teksta gdje su podaci odvojeni zarezima. Program ima opcije za automatsko formiranje indeksa pojnova u četiri nivoa i sadržaja. Raspolaze „on-line“ helpom, a u paketu su priložene i dvije tutorijal diskete.

## Planiranje

### MS Project

Microsoft

*Planiranje*

Program je namijenjen poboljšanju planiranja. Njime se može planirati sve — od godišnjih odmora do velikih projekata. Paket je baziran na standardnoj metodologiji mrežnog planiranja i omogućava analizu svih relevantnih faktora u projektovanju — trajanja, troškova i korištenih resursa. Vremenska skala može biti u rasponu od minuta do godina. U okviru jednog projekta može se pratiti do 999 aktivnosti. Ukoliko je broj aktivnosti veći, ili ako se želi istovremeno pratiti više projekata, moguće je projekte povezati u jedan makro prečetak. Kalendar omogućava da se, u periodu rada na projektu, odrede radni i neradni dani. Takođe, moguće je definisati i satnicu radnog dana. U svakom trenutku se mogu pregledati resursi i cijena njihovog korištenja do kraja rada i dobiti informacije o pojedinačnim resursima i dijagrami korištenja. Črtaju se Gant i PERT dijagrami. Na PERT dijagramu naznačen je kritični put. Izborom analize može se upoređivati tekuća dinamika projekta sa planiranom. Štampani izvještaji iz projekta su prilično dobri. Neophodan alat svakome ko se bavi planiranjem.

### MS Project 3.0

Microsoft

*Planiranje*

U ovoj verziji ispravljani su propusti prethodnika i usavršene mnoge funkcije. Ukupno trajanje projekta može biti do 120 godina. Iako je broj aktivnosti po projektu ograničen na 1000, zahvaljujući mogućnosti povezivanja projekata u makro projekt, praktično je neograničen. Nazivi aktivnosti mogu biti dugi do 40 karaktera, tako da se ne moraju izmišljati svakakve skraćenice. Nije se štedilo ni na komentarima, koji mogu biti dugi do 200 karaktera. Maksimalan broj resursa je 255, i to maksimalno 8 po aktivnosti. Plan može biti prikazan u vidu Gant i PERT dijagrama sa naznačenim kritičnim putem. Moguće je zumiranje dijelova mreže. Od ostalih opcija tu su: odvojeni kalendar aktivnosti i resursa, analize tipa



„najprije“ i „najkasnije“, histogrami resursa itd. Podaci se izvoze u „Lotus 1-2-3“, „Multiplan“, „MS Chart“, „Primavera“, „dBASE II“, i druge programe. Program zahtjeva minimalno 256 K RAM-a. Postoji i verzija za rad u mreži.

## Primavera

Primavera Systems Inc.

Planiranje

Profesionalni paket za planiranje i vođenje projekata, namijenjen svima koji smatraju da je vrijeme novac i više se pouzdaju u čvrst i precizan plan nego u improvizaciju. Omogućava pravljenje plana aktivnosti i automatsko izračunavanje kritičnog puta u mreži, praćenje i kontrolu alokacije resursa, kontrolu troškova i analize tipa „šta ako...“. Dovoljava 96 resursa po projektu, a maksimalno 6 po aktivnosti. Daje automatski proračun troškova korištenja resursa, pri čemu uzima u obzir variranje cijene resursa u vremenu kao posljedice inflacije! Pored rad i Gant dijagrama, moguće je tabelarni prikaz plana aktivnosti, histograma resursa i troškova, stubičnih dijagrama i kumulativnih krivih. Na osnovu podataka o projektu, automatski se mogu uraditi veoma iscrpni izvještaji. Program je „command-driven“. Raspolaze dobrim „on-line helpom“, koji uzima u obzir poziciju u kojoj je pozvan (context sensitive) i korisniku odmah daje informacije o aktivnosti kojom se bavi. Unos podataka je jednostavan i brz. Podaci se, između ostalog, izvoze u „Lotus 1-2-3“, „dBASE III“ i ASCII formatu. Za rad sa programom potrebno je minimalno 512 K RAM-a i tvrdi disk od 10 Mb.

## Pomoćni programi

### dUTIL III PLUS

Fox & Geller Inc.

Pomoćni program

Ovaj program može biti od koristi ljubiteljima „dBASE III i III Plus“ sistema. Omogućava da se sa izvornim kodom programa urade mnoge korisne stvari, kao što su: provjera sintakse i pravljenje datoteke sa greškama, automatsko „nazubijavanje“ programa, ispis komandi velikim, a varijabilj malim slovima, pravljenje list datoteke sa linjskim brojevima, kreiranje „crossreference“ datoteke svih varijabli, programa i datoteka, formiranje stabla programa, uključivanje procedura u glavni program... i još mnogo toga. Opcije se biraju postavljanjem odgovarajućih parametara u meniju. Pri obradi izvornih datoteka ostaje nepromijenjena, a formira se nova sa željenim imenom. Program je dosta fleksibilan i korisnika može poštediti dosta dosadnog posla oko uređenja datoteka sa izvornim kodom.

### Norton Utilities

Peter Norton

Pomoćni program

Vrlo koristan program. Omogućava direktan pristup svim datotekama na disku i prikaz njihove fizičke pozicije. Može se pregledati i mijenjati sadržaj datoteka direktno na disku. Sadržaj je prikazan u heksadecimalnom i ASCII formatu. Editor za promjenu sadržaja je ekranski, a sve promjene su posebno istaknute. Promjene se ne upisuju na disk sve dok to korisnik ne odobri. Posebna prednost ovog programa je mogućnost vraćanja obrisanih datoteka. Vraćanje je potpuno uspješno ako se poduzme odmah nakon brisanja, prije nego što se na mjesto obrisane datoteke snime novi podaci. Sve opcije se biraju iz menija.

### Lotus HAL

Lotus Development Corp.

Pomoćni program

Po imenu podsjeća na računar iz Klarkove Odiseje 2001, ali je HAL, u stvari, skraćena od Human Access Language. Omogućava korištenje normalnog (engleskog) jezika za upravljanje „Lotus 1-2-3“ programom. Rezidentan je u memoriji, a iz „Lotusa“ se poziva kao i obični komandni meni. Korištenjem HAL-a bitno se ubrzavaju mnoge operacije, kao što su:

unos formula, formatiranje tabele, kopiranje i premještanje ćelija, kreiranje grafike i slično. Pored toga, dodane su i neke nove funkcije kao „replace“ za dodavanje komentara u formule, ili „undo“ za vraćanje obrisanih dijelova tabele. Da bi se stekao neki utisak kako izgleda rad HAL-om, uzmimo primjer sumiranja kolona tabele. Normalno bi trebalo povući crtu, napraviti formulu za jednu kolonu, a zatim tu formulu iskopirati potreban broj puta. HAL to sve uradi sam na komandu „total all columns“. Da bi se kreirao grafik na osnovu podataka iz tabele kucate se, na primjer, „graf col b to col a“. Jednostavno, zar ne? Ako puno radite sa „Lotus“-om, HAL će vam dobro doći.

### PC Tools

Central Point Software

Pomoćni program

Veoma upotrebljiv i koristan program, koji može da olakša svakodnevne poslove u operativnom sistemu. Prilično je ljubazan za korisnikom. Opcije se biraju iz menija. Za rad sa diskom program crta stablo direktorija, koje omogućuje lakše snalaženje. Mnoge operacije su poboljšane i olakšane. Ako se, na primjer, kopiraju datoteke, jednostavno se pravi lista za kopiranje. Isto važi i za brisanje datoteka. U listu se mogu uvrstiti i skrivene (hidden) datoteke. Prilikom kopiranja disketa (discopy) proces kopiranja je predstavljen grafički. PC Tools omogućava traženje uzorka teksta u datoteci ili na čitavom disku. Takođe, moguće je na disku, tragaajući po svim poddirektorijima, pronaći sve datoteke sa zadanim imenom. Postoji i opcija za rekonstrukciju i vraćanje obrisanih datoteka, kao i za promjenu atributa. Sadržaj diska može se direktno pregledati i mijenjati. Prikaz je u heksadecimalnom i ASCII formatu. Program daje i korisne informacije o sistemu i grafički prikazuje alokaciju prostora na disku. U odnosu na slične programe (Norton Utilities npr.), može se reći da je kompletniji i upotrebljiviji.

### SideKick

Borland International

Priručni pomoćni program

Ovaj program spada u grupu „load-and-stay-resident programa“, tj. u one programe koji se jednom učitaju u memoriju i tamo ostaju sve dok se sistem ne resetuje. Korisnik ih jednostavno poziva pritiskom na odgovarajuće tipke. Program se može pozvati u bilo kom trenutku i iz bilo kog drugog programa. Rad se odvija u prozorima. Na raspolaganju su „Wordstar“ kompatibilan tekst procesor za kraće zabilješke (Notepad), kalkulator sa decimalnim, binarnim i heksadecimalnim modom, kalendar, ASCII tabelu i telefonski imenik sa mogućnošću automatskog poziva. Program će raditi na svakom PC-u sa minimalno 128 K RAM-a.

### Turbo Lightning

Borland International

Pomoćni program — provjera pravopisa

Izuzetno brz i moćan Spelling Checker, snabdjeven rječnikom od 80000 (osamdeset hiljada) riječi i tezaurusom (rječnikom sa hijerarhijskom pojmovna) od 50000 (pedeset hiljada) riječi. Program radi u pozadini, a ispravnost pisanja provjerava se u toku rada sa nekim tekst procesorom. Gotovo trenutno možete dobiti i sinonime za problematičnu riječ. Rječnik je, naravno, engleski, tako da će vam program koristiti jedino ako pisete na tom jeziku. Verzija na našem jeziku još nije ni u pripremi. Zahtjevi na memoriju su 256 K.

## Programski jezici

### Turbo BASIC

Borland International

Programski jezik

Smatrajući da nije obavezno da bejzik bude spor, u „Borland“-u su u svom stilu, napravili program koji stvari postavlja naglavce. Rezultat je

jezik kompatibilan sa BASICA dijalektom, tako da ne treba učiti novi jezik, odličan interaktivni editor sa četiri prozora i menijima u stilu „Turbo Prolog“-a i lizuzetno brz prevodilac. Dužina programa je ograničena samo raspoloživom memorijom. „Turbo BASIC“ podržava rekurzije, lokalne, statičke i globalne varijable, duge cjelobrojne varijable (32 bita), 80 bitnu preciznost itd. Podržan je i matematički koprocesor, koji se, ako nije raspoloživ, emulira programski. Nije ja da i do sada nije bilo jezik kompajlera, ali gotovo svi su zahtijevali prepreku originalnog koda i žosta muke dok program ne proradi, zato je ovo program koji su ljubitelji jezika dugo čekali. Program zahtijeva minimalno 256 K RAM-a.

## QuickBASIC (verzija 2.0)

Microsoft

Programski jezik

Ovaj kompajler će ubrzati programe pisane u BASICA dijalektu jezika 3 do 10 puta. Eventualne modifikacije izvornog koda su minimalne. Osim toga, na raspolaganju su i nove mogućnosti. Podržano je potpuno strukturano programiranje. Brojevi linija nisu neophodni, ali su zadržani zbog kompatibilnosti sa BASICA. Labele mogu biti alfanumeričke. Potprogrami se pozivaju imenom. Omogućen je prenos parametara. Podržane su lokalne i globalne varijable. Korisnički definisane funkcije mogu se prozitati na više linija. Program se kompilira u memoriju brzinom od oko 6000 linija u minut! Omogućena je posebna kompilacija svakog modula, pa se može praviti biblioteka kompiliranih rutina. Program može biti dugačak koliko mu slobodna memorija dopušta. Pojedinačne numeričke tablele mogu biti veličine do 64 K. Ugrađeni editor je ekranski sa automatskim postavljanjem na otkrivenu grešku. Ima ugrađen dibager. Komande se biraju iz menija na vrhu ekrana. Prelazak sa pisanja programa u editoru na prevodjenje i izvršavanje je brz i jednostavan.

## GW BASIC 3.2

Microsoft

Programski jezik

Jedna od najmoćnijih verzija jezika interpretira ih tržištu. Podržana je potpuna redakcija standardnih ulaza i izlaza. Tako, program može da prima podatke iz datoteke, umjeso sa tastura, i da ih opet smješta u datoteke, umjesto da ih šalje na ekran ili štampač. Korisnik može da instalira i koristi svoje specifične uređaje. Ulazno/izlazne operacije na disku podržavaju velike datoteke (do 16.777.215 slojeva). Datoteke mogu biti sekvencijalne ili sa direktnim pristupom. Direktno iz programa mogu se izvršavati DOS komande ili neki drugi programi. Moguće je dodatno opsluživanje događaja u sistemu, izvještaji o greškama su precizni. Transcendentne funkcije mogu se računati u dvostrukoj preciznosti. Skup instrukcijski je prilično bogat (174 instrukcije). Grafika i zvuk su dobro podržani. Raspoložive prozornice sa skaliranjem. Komandama PUT i GET može se brzo uraditi jednostavna animacija. Korisnik može pozvati i sopstvene rutine u mašini. Mana je relativno mala veličina korisničkih programa (do 60332 bajta).

## Turbo C

Borland International

Programski jezik

Borland polako zaokružuje ponudu „turbo“ kompajlera. C kompajler zadovoljava ANSI standard. Prevodilac je jednodrozni i prevodi više od 7000 linija u minut! Prevodilac ne stvara izvršne verzije, već objektnne module, pa je u paketu i „Turbo Linker“. Objektni moduli su kompatibilni i sa PC-DOS linkerom. Moduli su udruživi sa modulima razvijenim u „Turbo Prologu“ i mogu se povezati u jedinstveni program. Matematski koprocesor se emulira programski. Editor je ekranski i interaktivni. Kada prevodilac pronađe grešku, automatski se prelazi u editor, a kurzor se postavlja na mjesto gdje je greška otkrivena. Uporodni brzinski testovi sa najnovijim verzijama „Microsoft“ i „Lattice C“ kompajlera su zaista impresivni. Impresivan je, naravno, i odnos cijena. Program zahtijeva bar 320 K RAM-a.

38 računari 29 • avgust 1987.

## C (verzija 3.0)

Microsoft

Programski jezik

Jezik poznat i popularan po svojoj prenosivosti, kompaktnosti koda i brzini izvođenja. Dvoj svojih osobina naročito je pogodan za pisanje sistemskih programa. Kompajler je optimizirajući. Implementirane su registarske varijable i mali, srednji i veliki memorijski modeli. Modeli se mogu mijesati uz pomoć NEAR i FAR pointera. Zadržana je kompatibilnost na nivou izvornog koda između MS-DOS i XENIX 286 sistema. U biblioteci rutina sadržana je većina rutina UNIX V sistema. Ugrađena je direktna podrška ili emulacija matematskog koprocesora i brz računanje bez koprocesora. Podržan je rad u mreži, dijeljenje datoteka i „zaključivanje“ zapisa i datoteka. Moguće je povezivanje sa modulima razvijenim u drugim „Microsoft“-ovim jezicima.

## DEMO

Software Garden

Programski jezik

Ovaj program samo uslovno spada u kategoriju programskih jezika. Najviše liči na DLAN, poznat sa „spektruma“, ali manjih mogućnosti. Namijenjen je praviljuju prezentaciji sa animiranim tekstom i jednostavnom grafikom. Omogućava kreiranje, promjenu i pregled slajdova, tj. slajdica. Pod kontrolom programa ti slajdovi se smjenjuju na ekranu i daju utisak animacije, ili simuliraju odvijanje programa. Pomoću ovog programa programi mogu proraditi (na ekranu) i prije nego što su napisani. Na žalost, ne raspoloža sa različitim vrstama fontova, a grafika se svodi na crtanje jednostrukih, ili dvostrukih linija. Komande se biraju iz menija, ili uz pomoć funkcijskih tipki. Nema neki poseban komandni jezik, ali se uz malo mašte i spretnosti mogu dobiti simpatični i efektni rezultati.

## FORTRAN (verzija 3.3)

Microsoft

Programski jezik

Jezik koji je, uglavnom, bio rezervisan za veće sisteme, i koji se dosta uči na našim fakultetima. Uglavnom se koristi za rješavanje inženjerskih, i, uopšte, računskih intenzivnih problema. Ovaj program sadri većinu opcija predviđenih ANSI 77 standardom, ali ima i dodatata. Podaci, tablele i COMMON varijable više nisu ograničeni na 64 K. Programi mogu biti dužine do 1 Mb. Podržana je uslovna kompilacija i „overlay“ strukture. Može se birati između direktne podrške, ili emulacije, 8087/80287 matematskog koprocesora, računanja bez matematskog koprocesora po IEEE standardu i BCD matematike. Zadržana je kompatibilnost između MS-DOS i XENIX 286 sistema, na nivou izvornog koda. Objektni moduli razvijeni i u drugim „Microsoft“-ovim jezicima, kao što su C (verzija 3.0, ili kasnija), „Pascal“ (verzija 3.3, ili kasnija) i makro assembleru, mogu se povezati u jedinstveni program. Prilagođen je radu u mreži.

## Macro Assembler (verzija 4.0)

Microsoft

Programski jezik

Alat neophodan svakome ko želi da se bavi mašinskim programiranjem. Ova verzija je tri puta brža od prethodnih i predstavlja zaokruženu cjelinu. Zadržana je kompatibilnost nagore sa starijim verzijama MS i IBM makro assemblya. Prepoznaje instrukcije 8086/8087, ali i 80186/80286/80287 mikroprocesora, pa se može koristiti i na XT i na AT računarima. Već iz imena se vidi da su podržane makro instrukcije, pomoćnih programa. Tu je, prije svega, izvanredan interaktivni simbolički dibager, koji se može koristiti i sa izvornim i objektnim kodom drugih „Microsoft“-ovih jezika (FORTRAN 3.3, Pascal 3.0, C 3.0). Library Manager omogućuje kreiranje biblioteka objektnih modula. Object Code Linker podržava „overlay“ strukture. Cross Reference Utility pravi listu definicija i lokacija svih korištenih simbola u programu. EXE File Compression Utility komprimuje .EXE datoteke, tako da zauzimaju manje prostora na disku, brže se učitavaju i izvršavaju. EXE File Header Utility omogućava prikaz i promjenu zaglavlja .EXE datoteka, podešavanje inicijalne memorijske alokacije i veličine steka.

## Pascal (verzija 3.3)

Microsoft

Programski jezik

Ko voli da piše programe u paskalu duže od 64 K, pa mu zbog toga ne odgovara „Turbo Pascal“, može slobodno da se odluči za ovaj kompajler. Programi mogu biti dužine do 1 Mb, sa višestrukim segmentima za podatke. Oni kojima ni to nije dovoljno, mogu koristiti „overlay“ strukture. Moguće je prevodjenje po modulima. Posjeduje 4 biblioteke matematskih funkcija (8087/80287 podrška, IEEE standard, 8087/80287 emulacija i BCD matematika). Podržan je rad u mreži, dijeljenje datoteka i „zaključavanje“ zapisa i datoteka. Zadržana je kompatibilnost na nivou izvornog koda između MS-DOS i XENIX 286 sistema. Omogućeno je povezivanje modula razvijanih pomoću drugih „Microsoft“ ovih jezika — na fortranu (verzija 3.3 ili viša), C-u (verzija 3.0 ili viša) ili makro asembleru.

## Turbo Pascal

Borland International

Programski jezik

Legendarni prvi Borland-ov proizvod i prvi prevodilac za paskal ispod 100 dolara. Zahvaljujući cijeni, ali i kvalitetu, stekao je puno poklonika. Jezik je, osim osnovnog seta instrukcija, pojačan procedurama za rad sa grafikom i zvukom. Verzija 3.0 ima i Turte grafiku. Prevodjenje je veoma brzo, a kod je kompaktan i brz. Izvršna verzija dobija .COM nastavak. Dužina programa je ograničena na 64 K. U sklopu programa dolazi i editor. Prilično je neljubazan i zahtjeva pamćenje niza kontrolnih sekvenci. Dobra strana je da korisnik sam može da izabere koje će se sekvence biti. Programiranje i ispravljanje grešaka je jednostavno. Ako se otkrije greška u toku prevodjenja, sistem nas automatski prebacuje u editor, na mjesto gdje je greška otkrivena. Ako imate neke posebne zahtjeve, na raspolaganju je niz dodatnih biblioteka procedura na izvornom ili objektnom nivou. Da biste uživali u „Turbo Pascal“-u, potreban vam je PC sa minimalno 128 K RAM-a.

## Turbo PROLOG

Borland International

Programski jezik

Jezik pete generacije namijenjen razvoju ekspertnih sistema i vještačke inteligencije. U prologu (PROgraming in LOGic), programiranje je puno jednostavnije i prirodnije nego na „klasičnijim“ jezicima. Osnove jezika možete slediti iz tradicionalno dobre popratne dokumentacije. Interfejs prema korisniku zasnovan je na prozorima i menijima. Editor je dobar. Prevodjenje je izuzetno brzo. Nema ograničenja na dužinu izvornog koda. U sklopu programa dolazi izvorni kod za GeoBase — bazu podataka o geografiji SAD pretraživu prirodnim jezikom. Ako vas zanima vještačka inteligencija i novi pristupi programiranju, ovo je pravi program za vas. Za rad sa programom potrebno je minimum 384 K RAM-a.

## Proširenja DOS-a

## MS Windows

Microsoft

Proširenje DOS-a

Koristan dodatak DOS-u, koji na PC uvodi grafičku radnu okolinu i još niz novosti kao što su: rad sa prozorima i mišem, rad sa više programa u memoriji, prenos podataka između programa, dobre i korisne pomoćne programe (sat, kalkulator, kalendar, emulator terminala, tekst procesor, baza podataka, program za crtanje itd.). Prozori se ne preklapaju i uvijek optimalno pokrivaju radnu površinu, ukoliko korisnik ne želi da mijenja veličinu. Sve aplikacije specijalno dizajnirane da rade sa ovim programom rade u prozorima, ali u prozorima mogu da rade i neke od standardnih aplikacija. Prelazak iz programa u program i prenos podataka su veoma jednostavni. Sve komande se biraju iz menija. Da bi se sistem efikasno koristio, miš nije prijeko potreban. Sve je više aplikacija koje koriste prednosti ove nove grafičke radne okoline, tako da se MS Windows

nameće kao novi standard. Novi Apricot-ov XEN računar ima „MS Windows“ kao osnovnu radnu okolinu. Potrebno je minimalno 256 K RAM-a (za istovremeno rad sa više programa preporučuje se 512 K).

## GEM Desktop

Digital Research

Dodatak DOS-u

Program koji stari dobri DOS zaodjeva u grafičko ruho, dodaje mu sličice, prozore i neizbježnog miškonu. Rad na GEM radnom stolu je zaista prijatan. Prozori su efektni i lijepo dizajnirani, ali ih može biti maksimalno četiri. Komande se biraju iz propadajućih menija. Upravljanje je prilagodeno mišu, pa će korisnici tastature biti malo hendikepirani. Uz program dolaze sat sa alarmom, kalkulator i „Spooler“ — program za štampanje. Moguće je pokretanje standardnih aplikacija, ali ne i rad sa više njih istovremeno. Preporučuje se korištenje specijalnih GEM aplikacija (Paint, Draw, Write itd.) koje u potpunosti iskoristavaju novu radnu okolinu. Za rad je potrebno bar 256 K RAM-a za DOS 2. xx, odnosno 320 K RAM-a za DOS 3.xx, i dva 360 K disketna pogona.

## TopView

IBM

Dodatak DOS-u

Program omogućava pokretanje i rad sa više programa istovremeno. To nije multitasking sistem, već su programi istovremeno prisutni u memoriji i može se lako prelaziti iz jednog u drugi. Programi rade u prozorima. Prozori se mogu preklapati, pomjeriti i može im se mijenjati veličina. Da bi se prešlo iz preklopa u program, dovoljno je preći iz prozora u prozor. Na sličan način mogu se kopirati dijelovi teksta iz jednog prozora (programa) u drugi. Prije pokretanja neke standardne aplikacije pod „TopView“ sistemom, potrebno je popuniti neku vrstu formulara podacima koji se čuvaju u posebnoj PIF (Program Information File) datoteci. Postoji veća zbirka već urađenih PIF datoteka za najpoznatije standardne aplikacije. Upravljači se može tastaturom ili mišem. Komande se biraju iz menija koji se pojavljuje na pritisak odgovarajuće tipke. Program podržava boje. Ne podržava prenos grafike iz programa u program.

## Stono izdavaštvo

### Click Art Personal Publisher

T/maker Company

Stono izdavaštvo

Jedan od prvih programa namijenjenih PC-u, u oblasti stonog izdavaštva. Poštovani su standardni principi u kreiranju ovakvih programa. Mogu se mijesati tekst i grafika. Tekst se može kucati direktno, ili prenositi iz ASCII datoteke, s tim da blokovi ne budu duži od 5 Kb. Na raspolaganju je više vrsta fontova, koji nose imena poznatih gradova (New York, Monaco, Athens...). Fontovi (ne svi) mogu biti ispisani u normal, bold, ili italic načinu. Posebnim programom (FontMove) postojeći fontovi se mogu mijenjati. Tekst može biti ispisan u maksimalno četiri kolone. Grafika može biti kreirana scanner-om ili prenijeta iz drugih programa. Za to može poslužiti poseban Snapshot program, kojim se pravi kopija ekrana bilo kog drugog programa. Omogućena je i jednostavnija obrada i dorada takve grafike, dodavanjem teksta i crtanjem linija ili pravougaonika. Program zna sam da tekstom popuni prostor oko grafike. U svakom trenutku, vrlo brzo je moguće vidjeti izgled kompletne stranice. Upravljanje je kombinacija menija i funkcijskih tipki, koja u prvom trenutku može zbuniti korisnika. Može se koristiti tastatura ili miš.

### PageMaker

Aldus Corporation

Stono izdavaštvo

Nije trebalo čekati dugo da se ovaj izvanredni program, prvobitno namijenjen „mekintošu“, pojavi i u verziji za PC. „PageMaker“ radi pod



„MS Windows“ sistemom. Program omogućuje potpunu kontrolu nad izgledom dokumenta, poštujući princip WYSIWYG. Omogućeno je miješanje teksta i grafike. Tekst se može unositi direktno, ali se prihvataju i zapisi gotovo svih poznatijih procesora teksta, kao i dokumenti u IBM DCA formatu („WS 2000“, „Volkswriter 3“ ...), ili obične ASCII datoteke. Naravno, na raspolaganju je mnoštvo fontova i načina ispisa. Grafika može biti urađena nekim od programa za crtanje, bilo „draw“, bilo „paint“ tipa („MS Paint“, „MS Draw“, „PS Paint“, „PC Paintbrush“, „In a Vision“ ...). „AutoCAD“-om (ADI format) i „Lotus“ ili „Symphony“ programima (PIC format). Prihvata i digitalizovane slike sa scanner-a, ako su u TIFF (Tag Image File Format) formatu. Tekst na stranici može biti podijeljen u maksimalno 20 stubaca. Stranica može biti podijeljena u maksimalno 20 stubaca. Stranica može biti umanjena za 50% ili 75%, ili uvećana 200%. Dokument može sadržati maksimalno 128 stranica. Podržana su i dva standardna jezika za opis izgleda dokumenta — PostScript i DDL (Document Description Language). Naravno, ovakav program ima i posebne zahtjeve na hardver. Potrebno je minimalno 512 K RAM-a disk od 10 Mb. Ne preporučuje se ni CGA, već EGA, Hercules, ili neka slična grafička karta. Za kvalitetan ispis neophodan je laserski štampač, mada su podržani i matični.

## Statistika

### Statgraphics

Statistical Graphics Corporation

Statistički i matematički paket

Izuzetan program za sve one koji se bave statističkom i numeričkom analizom. Iz glavnog menija bira se neka od 22 raspoloživih oblasti. Svaka od tih oblasti ima svoj meni sa opcijama, tako da je broj mogućih funkcija

jako veliki (oko 250). Neke od njih su: analiza vremenskih serija, ekstrapolaciona analiza, analiza varijance i regresiona analiza, brza Fourier-ova transformacija, množenje i inverzija matrica, linearno programiranje (simplex metoda), nule polinoma, sistemi jednačina, numeričko diferenciranje itd. Tu je i opcija za deskriptivnu statistiku. Pošto se grafika spominje i u imenu, na raspolaganju u opcije za crtanje 2D i 3D grafike na osnovu diskretnih podataka i/ili matematičkih funkcija. Grafika može biti u boji ili crno-bijela. Crtanje je malo sporo, ali dosta kvalitetno. Za unos podataka koristi se ugrađeni editor, a program prihvata i podatke u ASCII formatu, podatke iz „Lotus 1-2-3“ tabela i podatke u DIF formatu. Program raspolaze „on-line“ helpom. Za rad zahtjeva dvije 360 K disketne jedinice i minimalno 384 K RAM-a. Preporučuju se matematički koprocesori i 512 K RAM-a.

## Tabelarni kalkulatori

### SuperCalc 4

Computer Associates International

Tabelarni kalkulator

Program može dobro doći onim korisnicima koji puno rade sa tabelama, a ne vole da koriste velike integrisane pakete. Omogućava sve vrste tabelarnih proračuna i prikaz grafike na osnovu podataka iz tabele. Grafiku crta prilično brzo, ali je ne može odštampati. Poziv menija, izbor komandi i princip rada su gotovo isti kao i u programu „Lotus 1-2-3“, tako da neće biti problema sa privikavanjem. Kretanje po tabeli i unos podataka su poboljšani i ubrzanji. Rad sa programom je dosta komforan. Kao mana bi se moglo navesti da prilikom štampanja prikazuje i tabele redova i kolona. Program može da prihvata i snima podatke u raznim formatima, tako da nema problema sa razmjenom.



Microsoft QuickBASIC 3.0

# Mali oglasi

Ako ne možete da podnesete da drugi nemaju ono što vi imate, objavite svoj mali oglas u „Računarima“.

Ako ne možete da podnesete da drugi imaju ono što vi nemate, javite se na neki od malih oglasa u „Računarima“.

Prva stvar koju treba da uradite jeste da se odlučite da li želite običan ili uokviren mali oglas.

## Obični oglasi

Cena običnog malog oglasa do dvadeset reči je 1800 dinara. Svaka naredna reč košta još 150 dinara. Veznici, predlozi, priloge, zamenice, brojevi i ostale „male“ reči se računaju u cenu. Adresa oglašivača se ne računa u cenu. Tri važna ograničenja: mali oglas ne može biti duži od 50 reči, mali oglas ne može biti štampan velikim slovima i uz mali oglas se ne mogu objavljivati crteži i fotografije. Mali oglasi koji se ne uklapaju u ove okvire pripadaju kategoriji uokviranih malih oglasa.

## Uokvireni mali oglasi

Cena uokvirenog malog oglasa je 2000 dinara i po visinskom centimetru u stupcu širine 9,5 citera ako oglas nije viši od pet centimetara i 3000 dinara po visinskom centimetru ako je mali oglas visok između pet i deset centimetara. U sklopu uokvirenog malog oglasa mogu se objavljivati fotografije i crteži i mogu se birati veličina i tipovi slova (belo, polucrno, kurziv). Fotografije i crteži se plaćaju prema prostoru kao da se radi o tekstu. Jedno važno ograničenje: uokvireni oglasi preko 10 centimetara ne spadaju u kategoriju malih oglasa. Za njih važe pune komercijalne cene i oni se ugovaraju sa oglašnim odeljenjem BIGZ-a.

## Priprema malih oglasa

Poželjno je da mali oglas počinje sa Prodajem, Kupujem, Držim Časove, Menjam... i sličnim što ukazuje na njegovu sadržinu. Adresa oglašivača se kuca u produžetku teksta malog oglasa, a ne odvojeno. Ova ograničenja, razume se, ne važe za uokvirene male oglase.

Uz mali oglas treba navesti njegovu vrstu (običan, uokviren) i kategoriju u kojoj će biti objavljen „spektrum“, „komodor“, „hardver“, „literatura“...)

## Prijem i plaćanje malih oglasa

Mali oglas treba dostaviti na adresu redakcije „Računari“ — BIGZ (za male oglase), Bulevar vojvode Mišića 17, 11000 Beograd najkasnije do petog u mesecu. Svi oglasi koji do ovog roka pristignu u redakciju poštom, lično i, uz određena ograničenja, telefonom, biće uvršćeni u sledeći broj.

Mali oglasi se, po pravilu, plaćaju unapred bankovnom uplatnicom na račun 60802-603-23264 BIGZ, Bulevar vojvode Mišića 17, 11000 Beograd, sa obaveznom naznakom: „Računari“, mali oglasi. Kopiju uplatnice treba, obavezno, dostaviti zajedno sa tekstom malog oglasa.

## Prijem malih oglasa telefonom

Mali oglas se, u opravdanim slučajevima, može redakciji dostaviti telefonom (011/650-161), ali samo pod uslovom da nije duži od dvadeset reči ili da se naručuje ponavljanje oglasa iz prethodnog broja „Računara“. U tom slučaju oglas se može platiti i naknadno, uplatnicom koju će oglašivaču uputiti redakcija.

## Važno upozorenje

Sadržaj malog oglasa predstavlja diskreciono pravu oglašivača, pod uslovom da ne dolazi u koliziju sa zakonom i da ne vrjednja javni moral. U tom smislu, u „Računarima“ se ne vrjednja mali oglas u kome se naglašava ili veliča pornografska, mijltaristička ili slična društvenoneprihvatljiva osobina pojedinih igara.

## Skraćeni rokovi — malo oglasa

Iz tehničkih razloga bili smo prisiljeni da avgustovski broj „Računara“ zaključimo dvadesetak dana ranije nego što je to uobičajeno i znatno ranije nego što je to najavljeno (20. juna, a ne 20. jula kako je, štamparskom greškom, objavljeno). To je omešlo oglašivača da nam dostave oglase na vreme, pa ovaj broj „Računara“ izlazi, praktično, bez „Malih oglasa“. Redakcija se izvinnjiva čitaocima što ova puta ostaju bez svoje uobičajene berze softvera i oglašivačima — ako im je ovaj nesporazum sa rokovima poremetio planove.

## Spektrum

ORIGINAL FROM JAPAN! Dolazi JVC SOFT. Ako želite naučiti, najkvalitetnije, najnovije i najjeftinije programe brže se javite na adresu: Miloš Nikodijević, Jug Bogdanova 26, tel. 018/69-653

### OLDTIMER SOFT

SPECTRUM programi u kompletima ili pojedinačno, vrhunski snimci. Besplatan katalog tražite telefonom 011/436-137 svakodnevno od 10 do 16 h ili pišom na adresu: Miroslav Radosavljević, Braće Nedića 2, 11000 Beograd

46: Uskorilo!!! 45: Saboteur 2, Storm, Vulcan, Indoor sports ... 44: Sidney affair, Pippo, Lil alien ... Naručite besplatan katalog!!! Petra Drapšina 21, Zemun, 011/102-727

## Komodor

C-64: Madness, Grand Prix, Feud, Odyssey, Turbo Espint, Hyppaball, Tarzan, Police Cadet, BMK, Flash Gordon, Terminator, Great Escape, Camelot Warriors ... 20 igara + kasetna + pt = 1700 dinara. Igrač Boris, Petre Dokića 2-C, 71000 Sarajevo, 071/647-730

Asterix and Obelix soft i ovog meseca vam nude najnovije programe u kompletima i pojedinačno. Veliki popusti, saveti za početnike, besplatan katalog, Milan Jakovljević, Konice 67, 11506 Draževac

Apolo Soft! Svezi programi, povoljne cene, popusti, pokloni. Za katalog kompletno poslati 500 din. Spisak besplatan. Moguća razmena Mutavdžić Vladimir, Petra Matkovića 153, 81000 Titograd, tel. 081/38-550

## Amstrad

Bajaset!!! Sve najnovije programe koji se pojavju na YU tržištu posedujemo i mi. Ako ste čuli za neke nove igre slobodno zovite nas, ili mi sigurno već imamo. Prodaja pojedinačno i u kompletima. Tražite besplatan katalog. Bojić Blažo, Ante Zuanica 15A, 88000 Mostar, tel. 088/415-203

Biser soft: nudi vam širok izbor usluhnih programa i igara veoma jeftino. Najnoviji programi: Sigma 7, American Ninja i Aphelst, Saboteur 2, B. M. X simulator itd. Katalog besplatan. Radović Miroslav, tel. 068/412-323. Moše Pijade 36, 88000 Mostar



AMISOFT YU CP/M Software predstavlja najnovije CP/M programe: micro COBOL, XLISP, FORTH-83, Small-C (floating point), NEWCPM 83 k, TURBO PASCAL, ROS 3.3, FX-Character Generator, CP/M Machine Code Input-Output Help, WRITE HAND MAN, CBASIC 80, EXBASIC, DR DRAW, DR GRAPH, CP/M Igre: MEGANS MONOPOLY, BACARRAT, REVERSI, 3D CLOCK CHESS. Amsoft yu, Spincičeva 5, 41000 Zagreb, tel. 041/315-478.

Kompleti CP/M i utility programi: Kompleti LANGUAGES: FORTRAN, PISTOL, JRT PASCAL, micro PROLOG. Komplet TEXT: WORDSTAR, MAILMERGE, PROSPELL, ROTATE, Komplet STAT: AMSTAT 1-3 (statistički paketi). Komplet Plus: dBASE II, SUPERCALC 2, WORDSTAR 3.04, ZIP, SDI, Komplet 22: MICROSRIPT, MICROPEN, MICROSPREAD, CP/M Utilities: dbase II Utilities, Architecture Utilities, C-Archive, SPOOL, XDRIVE 10 (Ram-disc), TURBO PASCAL Graphic TOOLBOX. Pokloni: CAMBASE DATABASE, Novi AMSDOS programi: MASTERFILE III 6128, TASCWORD 6128 YU, TASPPELL, MINI OFFICE 2, PROFY PAINTER. Hardware: proširenje 464 na 6128 (CP/M 3.0), RAMDISC 256 k, EPROMi sa YU slovima za DMP-2000 i NLIQ-401. Amsoft yu, Spincičeva 5, 41000 Zagreb, tel. 041/315-478

## IBM

IBM programi, Kupovina, prodaja, zamena. Tražite besplatan katalog. Marija Klaić, Pavićeva 4, 54400 Đakovo

## Razno

Prodajem programe (crtanje, muziku, igre, ACC, text editore, ...) i literaturu, diskete. Proširenje 1Mb i modulator za YU. Katalog 300 din. Trudić Marjan, Greenwickska 12, 62000 Maribor

# Kralj Artur ne zna bejzik

*Dokaz da ni kraljevi ne žive baš lagodno i da su prinuđeni da razrešavaju različite probleme je nova muka kralja Artura. Redosled korišćenja Grala je (u „Računarima 26“) utvrđen. Međutim, sastan i okruglog stola ne održavaju se redovno. Naime, svaki od vitezoza živi u svom dvorcu, prilično izdvojen od ostalih. Dvorci su, po pravilu, okruženi gustom šumom kroz koju prolaze putevi koji ih povezuju. Mreža puteva (tj. šumskih putejljaka) je takva da se iz bilo kojeg dvorca može stići do bilo kojeg drugog bez presedanja — odnosno bez usputnog prolaska pored nekog trećeg dvorca. Nevolje, međutim, počinju u proleće, kada usled otapanja snegova i čestih kiša nabujaju šumski potoci, zbog čega pojedini putevi postaju neprohodni. . .*

Kralj Artur je rešio da u ovom nezgodnom periodu vitezove poštedi čestog sastančenja (preplivavanja potočica u gvozdeno uniformi nije baš preporučljiv sport). No, vitezovi žele da i dalje neometano razmenjuju Gral. Tačnije, trenutni vlasnik Grala i nije baš zainteresovan za saradnju, ali su ostali u većini. Da bi umirio svoje vitezove, kralj Artur je preuzeo na sebe zadatak da utvrdi između kojih dvorca je komunikacija moguća. Nema šta, velikodušno od strane jednog kralja! Vitezovi su mu napravili spisak prohodnih puteva, pa je kralj Artur zaseo jedno veče i pokušao da razreši problem.

## „Kraljevski metod“ kralja Artura

Kao i svaki kralj, kralj Artur je pokušao da ovaj problem razreši „kraljevski“, na što lakši način. Na žalost, ni u računarstvu, kao ni u matematici, nema kraljevskih puteva, što je kralju Arturu ubrzo postalo jasno. Kako je on to rešavao problem? Odlučio je da prvo utvrdi da li je moguće iz dvorca ser Keja stići do dvorca ser Tristana. Sa spiska je utvrdio da među ovim dvorcima ne postoji prohodan put. Međutim, rezonovao je kralj Artur, možda se oni mogu posećivati korišćenjem zaobilaznih puteva. Potragu za mogućim putem treba nastaviti od onih dvorca do kojih se direktno može stići iz dvorca ser Keja, a to su dvorci ser Lanselota, ser Gavena i ser Ovena. Brzom proverom je utvrdio da se ni iz jednog od ova tri dvorca ne može stići direktno do dvorca ser Tristana. Dakle, isti postupak se mora primeniti i na njih. I tako se za tilil čas broj mogućih puteva višestruko umnožio.

No, kraljevsko obećanje je kraljevsko obećanje i ono se mora poštovati. Do duboko u noć kralj Artur je sedeo rešavajući problem, da bi ga na kraju ipak razrešio. Poučen ovim mučnim iskustvom rešio je da zapiše algoritam kako bi sledeći put sličan problem mogao da natovari na leđa svojih pomoćnika. Kralj Artur nije znao ni bejzik, ni paskal, niti bilo koji drugi programski jezik. Zato je algoritam napisao na čistom engleskom, a mi smo ga malo ubolili da bi bila čitaoica „Računara“ bio razumljiviji (slika 1).

Algoritam razrešava pitanje da li između 42 računari 29 • avgust 1987.

```
if k=1
then ( potraživanje je za put dužine 1 )
  if povezan(a,b)
  then postoji_put:=true
  else postoji_put:=false
else begin
  postoji_put:=false; ( pretpostavka da put ne postoji )
  for c:=prvi_vitez to poslednji_vitez
  do if povezan(a,c)
  then if postoji_put(k-1,c,b)
  then postoji_put:=true
end ( else begin )
```

Slika 1 Algoritam za ispitivanje postojanja puta određene dužine

dvorca a i b postoji put dužine k, pri čemu se za neki put smatra da je put dužine 1 ako je to direktni put između dva dvorca, da je put dužine 2 ako je to put sastavljen od dve puta dužine 1, itd. Vrednost funkcije *povezani(x, y)* je *true* ukoliko su x i y dvorci direktno povezani putem, u suprotnom je *false*.

Kralj Artur ne bi bio pravi kralj da se nije pohvalio svom čarobnjaku Merlinu kako je elegantno razrešio problem. Umesto pohvale, kralja je dočekala kritika. Pa zar on, jedan kralj, ne zna jednostavnije rešenje? Tako obilato korišćenje rekurzije u algoritmu koji treba da bude efikasan je upravo nepodustivo! Da li je kralj ikada čuo za grafove? Zbunjeni kralj je samo odmahivao glavom. Nije mu preostalo ništa drugo nego da sasluša Merlinovo objašnjenje i na brza k nauči nešto o teoriji grafova. Naravno, Merlin je kao pravi čarobnjak sve ove stvari imao u malom prstu. Prenosimo skraćenu verziju Merlinovog izlaganja.

## Merlinovo predavanje

„Graf se sastoji od skupa čvorova i skupa lukova (veza). Svaki luk čvorova je određen parom čvorova. Za dva čvora datog grafa kažemo da su direktno povezani ukoliko postoji luk koji ih spaja. Ako parovi čvorova koji određuju luk čine uređene parove, onda je graf orijentisan. U suprotnom, graf je neorijentisan. Čvorovi x i y su

povezani putem dužine k ako postoji niz od k-1 čvorova  $z_1, z_2, \dots, z_{k-1}$ , tako da su x i  $z_1, z_1$  i  $z_2, \dots, z_{k-1}$  i y direktno povezani.

Graf (neorijentisani) je upravo idealna stvar da se njime predstavi mreža dvorca i puteva između njih. Jednostavnosti radi, dvorce označavamo brojevima 1, 2, ..., N i oni predstavljaju čvorove grafa. Putevi između dvorca su lukovi grafa. . . .

Ostatak Merlinovog predavanja smo upriličili tako da bude razumljivo današnjim hakerima. Graf se najčešće predstavlja uz pomoć kvadratne matrice čiji su elementi vrednosti *true* i *false*. Tako, u slučaju matrice  $A = a(i, j)$ , vrednost elementa  $a(i, j)$  je *true* ukoliko su čvorovi i, j direktno povezani, u suprotnom je *false*. Kada je u pitanju neorijentisani graf, matrica A je simetrična, tj.  $a(i, j) = a(j, i)$  za sve vrednosti i, j. Deklaracija grafa je data na slici 2.

Čvorovi i, j su povezani putem dužine l ukoliko je vrednost od  $a(i, j)$  jednaka *true*. Ovi čvorovi su povezani putem dužine 2 preko trećeg čvora k ukoliko je vrednost izraza  $(a(i, k) \text{ AND } a(k, j))$  jednaka *true*. Posmatrajmo, sada, malo komplikovaniji izraz

$$(a(i, 1) \text{ AND } a(1, j)) \text{ OR} \\ (a(i, 2) \text{ AND } a(2, j)) \text{ OR} \dots \text{ OR} \\ (a(i, N) \text{ AND } a(N, j))$$

(N je broj čvorova grafa, odnosno broj vrsta /kolona matrica A.) Vrednost ovog izraza je *true* samo ukoliko postoji put dužine 2

```
const max_cvorova = 50;
type tip_cvorova = 1..max_cvorova;
matrica_povezanosti = array[tip_cvorova, tip_cvorova] of boolean;
var grafi: matrica_povezanosti;
```

Slika 2 Deklaracija grafa



```

procedure proizvod
  (a,b: matrica_povezanosti; var c: matrica_povezanosti);
var vred: boolean;
  i,j,k: integer;
begin
  for i:=1 to max_cvorova (prolazak kroz vrste)
    do for j:=1 to max_cvorova (prolazak kroz kolone)
      do begin {izračunavanje c[i,j]}
         vred:= false;
         for k:= 1 to max_cvorova
           do vred:= vred or (a[i,k] and b[k,j]);
         c[i,j]:=vred;
       end {for...do begin}
     end {for...do begin};
end {procedure proizvod};

```

Slika 3 Procedura za određivanje bulovskog proizvoda matrica

```

procedure tranzitivno_zatvorenje
  (mat: matrica_povezanosti; var put: matrica_povezanosti);
var i,j,k: tip_cvorova;
  novi_proizvod, mat_proizvod: matrica_povezanosti;
begin
  mat_proizvod:= mat;
  put:= mat;
  for i:=1 to max_cvorova-2
    ( predstavlja broj anizacije mat samom )
    ( seba da bi se dobila mat_proizvod )
    ( put je matrica putova dužine n-2 )
    ( i i jednaka i )
  do begin
    proizvod:=mat_proizvod, mat, novi_proizvod;
    for j:=1 to max_cvorova
      do for k:= 1 to max_cvorova
        do put[j,k]:= put[j,k] or novi_proizvod[j,k];
        mat_proizvod:= novi_proizvod;
      end {for...do begin}
    end {for...do begin};
  end {procedure tranzitivno_zatvorenje};

```

Slika 4 Procedura za nalaženje tranzitivnog zatvorenja grafa

```

for i:= 1 to max_cvorova
  do for j:= 1 to max_cvorova
    do putk[i,j]:= putk[i,i] or
      ( putk[i,k] and putk[k,j] )

*****
Slika 5.1
*****

putk:= putk[i];
for i:=1 to max_cvorova
  do if putk[i,k]
    then for j:=1 to max_cvorova
      do putk[i,j]:= putk[i,i] or putk[i,k] and putk[k,j]

*****
Slika 5.2
*****

```

Slika 5 Algoritam za određivanje matrice putk na osnovu matrice putk-1

```

procedure tranzitivno_zatvorenje2
  (mat: matrica_povezanosti; var put: matrica_povezanosti);
var i,j,k: tip_cvorova;
begin
  put:=mat; {put pokriva od mat}
  for k:= 1 to max_cvorova
    do for i:= 1 to max_cvorova
      do if put[i,k]
         then for j:=1 to max_cvorova do
            put[i,j]:= put[i,j] or putk[i,k]
         end {procedure tranzitivno_zatvorenje2};

```

Slika 6 Procedura za nalaženje tranzitivnog zatvorenja grafa prema Voršelovom algoritmu

između čvorova  $i, j$ . Označimo sa  $A2$  matricu takvu da je  $a2(i, j)$  vrednost gornjeg izraza. Matricu  $A2$  zovemo matricom puteva dužine 2. Čitaoci kojima je blisko matricno množenje odmah će primetiti da je matrica  $A2$  dobijena kao proizvod matrice  $A$  sa samom sobom, pri čemu je numerička

operacija množenja zamenjena logičkom operacijom konjunkcije, a sabiranje logičkom operacijom disjunkcije. Zato ćemo  $A2$  zvati bulovskim proizvodom matrice  $A$  sa samom sobom. Procedura koja izračunava bulovski proizvod dve kvadratne matrice data je na slici 3.

## Tranzitivno zatvorenje grafa

Slično, kao što smo definisali matricu  $A2$ , matricu  $A3$  možemo definisati kao matricu puteva dužine 3. Nije teško utvrditi da je  $A3$  upravo bulovski proizvod matrice  $A2$  i  $A$ . Vrednost od  $a3(i, j)$  je *true* samo ukoliko postoji put dužine 3 koji povezuje čvorove  $i, j$ . Generalno, definišemo matricu  $A[K]$  kao matricu puteva dužine  $K$ . Ovu matricu možemo dobiti kao bulovski proizvod matrice  $A[K-1]$  i matrice  $A$ . Vrednost od  $aK(i, j)$  je *true* samo ukoliko postoji put dužine  $K$  koji povezuje čvorove  $i, j$ . Pretpostavimo da želimo da saznamo da li postoji put dužine najviše 3 koji spaja čvorove  $i, j$ . Ukoliko postoji takav put, on će biti dužine 1, 2 ili 3, pa vrednost izraza

$$a(i, j) \text{ OR } a2(i, j) \text{ OR } a3(i, j)$$

upravo određuje da li takav put postoji ili ne.

Pretpostavimo sada da želimo da konstruišemo matricu  $PUT$  tako da je vrednost od  $put(i, j)$  jednaka *true* samo ukoliko postoji put koji povezuje čvorove  $i, j$ . Jasno,  $put(i, j) = a(i, j) \text{ OR } a2(i, j) \text{ OR } \dots$

Na žalost, ova „jednakost“ se ne može koristiti za izračunavanje matrice  $PUT$ , jer ima beskonačno mnogo članova. Umesto toga, za izračunavanje se može koristiti jednakost  $put(i, j) = a(i, j) \text{ OR } a2(i, j) \text{ OR } \dots \text{ OR } a[N-1](i, j)$ .

Zašto je dovoljno samo prvih  $N-1$  članova da bi se odredila vrednost od  $put(i, j)$ ? Evo obrazloženja. Pretpostavimo da između čvorova  $i, j$  postoji put dužine  $M > N-1$ . Neka taj put prolazi kroz čvorove  $i, k_1, k_2, \dots, k[M-1], j$ . U ovom nizu ima  $M+1$  čvorova, što je, prema pretpostavci o vrednosti za  $M$ , više od  $N$ . Prema tome, u nizu moraju biti dva ista čvora, tj. jedan čvor u ponovljen dva puta. Jednostavnim uklanjanjem čvorova koji su između ta dva jednaka čvora, uključujući i jedan od njih, a zatim ponovnom primenom čitave operacije ukoliko je broj čvorova još uvek veći od  $N$ , na kraju dobijamo put koji prolazi kroz različite čvorove, pa je dužine najviše  $N-1$ . Dakle, ako su dva čvora povezana putem, tada sigurno postoji put dužine najviše  $N-1$  koji ih povezuje. Iz tog razloga je dovoljno da odredimo samo prvih  $N-1$  članova prethodno datog beskonačnog izraza.

Graf predstavljen matricom  $PUT$  nazivamo tranzitivnim zatvorenjem grafa predstavljenog matricom  $A$ . To je, u stvari, čvorna polaznog grafa lukovima za svari, dopuna koji nisu direktno povezani, ali su povezani nekim putem. Na slici 4 je data procedura *tranzitivno\_zatvorenje* koja kao ulaznu vrednost uzima matricu  $mat$ , a kao izlaznu vrednost daje matricu  $put$  koja predstavlja tranzitivno zatvorenje grafa predstavljenog matricom  $mat$ . U proceduri se poziva prethodno već definisana procedura *proizvod*.

Problem dvorca i puteva među njima se jednostavno razrešava tako što proceduru *tranzitivno\_zatvorenje* primenimo na spisak koji su napravili vitezovi, tj. na odgovarajuću matricu koja prikazuje postojanje ili nepostojanje direktnih veza između dvorca.

## Novi algoritam

U trenutku kada je Merlin došao do ove tačke u svom objašnjenju, kralj Artur se pobunio: „Pa kako je to sada pojednostava-

Loto po željama čitalaca (5)

# Nova skraćenja

**Pisma i telefonski pozivi svedoče o tome da ima dosta vlasnika „spektruma“ koji su se uključili u zajedničko razvijanje programa za izradu loto-sistema. Bilo je korisnih predloga i sugestija, našla se i po koja zamerka. Sve to će samo pomoći da program bude raznovrsniji i bolji.**

Graf u ovom broju nije namenjen prikazivanju nekih posebnih podataka. Na ekranu se vide dobite kombinacije, ali ne kao nizovi brojeva. Svaki red grafa ima 39 malih pravougaonika. Ispunjeni pravougaonici označavaju izvučene brojeve, i obratno. Ovakav pregled dobitnih kombinacija mnogi igrači prave na kartama, uz pomoć olovke i lenjira. Neki naoko sakriveni podaci na taj način se pojavje kao na dlanu: o ponavljanju brojeva, razmacima, učestalosti, gustini pojavljivanja, dugoj neizvučenosti, „belim“ poljima...

može očekivati prvi (najmanji) broj dobite kombinacije, zatim drugi, pa treći i tako dalje.

Kad je u pitanju prvi broj, dovoljno je da odredimo samo gornju dozvoljenu granicu. Za poslednji, najveći broj, određuje se donja granica. Rutina PRVI I POSLEDNI BROJ eliminiše iz sistema kombinacije kojima je prvi broj veći od 4, a poslednji manji od 13. (Pretpostavka je da sistem ima 15 brojeva, od 1 do 15, koji tek treba da se zamene.) To lako možemo izmeniti ako upišemo druge granične brojeve iz in-

```

1510
1520 PRVI I POSLEDNI BROJ
1530
1540 ELIM LD A,(RD+1)
1550 CP 3
1560 JP NC,GENER
1570 CP 3
1580 LD HL,(PR1)
1590 LD A,(HL)
1600 CP 13
1610 JP C,GENER
  
```

```

1510
1520 II OSTALI BROJEVI
1530
1540 ELIM LD A,(RD+1)
1550 CP 7
1560 JP NC,GENER
1570 CP 3
1580 JP C,GENER
1590
1600 LD A,(RD+2)
1610 CP 9
1620 JP NC,GENER
1630 CP 3
1640 JP C,GENER
  
```

```

220 BR0 EQU 15
230 GAR EQU 7
240 DK0 EQU 7
250 :---
260 GK0 DEFS DK0
270 ZK0 DEFS DK0
280 PGK DEFS 2
290 PZK DEFS 2
300 BRK DEFS 2
310 ADK DEFS 2
320 VAR DEFS 1
330 FIKS EQU 4
  
```

```

1530 :---
1540 ELIM LD HL,GK0
1550 LD B,BK0
1560 pf LD A,(HL)
1570 CP FIKS
1580 JR Z,NAST
1590 INC HL
1600 DJNZ pf
1610 JP GENER
  
```

```

40 PRINT PAPER 6: " L O T O
GRA F za 50 kola "
50 PRINT "Dobitne kombinacije"
60 LET y=150: LET x=6
70 FOR n=1 TO 51
80 PLOT 6,y: DRAW 234,0
90 LET y=y+3
100 IF n<41 THEN PLOT x,0: DRA
W 0,150
110 LET x=x+4
120 NEXT n
130 LET y=148
140 FOR i=1 TO 50
150 FOR j=1 TO 7
160 READ y
170 PLOT x+6+1,y: DRAW 4,0: DRA
W 0,1: DRAW -4,0
180 NEXT j
190 LET y=y-3
200 NEXT i
  
```

Graf je urađen na osnovu programske datoteke sa 50 dobitnih kombinacija, od kojih je poslednja izvučena u 22. kolu ove godine. Kako se datoteka može ažurirati objašnjeno je u dva-tri navrata u prethodnim brojevima.

## Granice raspona

U „Računarima“ broj 26 graf nam je pomogao da zaključimo u kom se rasponu

## Jedno ozbiljno pismo

### Formulom do dobitka

Mislim da se vaša rubrika može svrstati u sam vrh po kvalitetu u „Računarima“, pa vam stoga i pišem. Mada je u njoj više vidim svojevrsnu školu mašinskog programiranja nego mogućnost za brzo i lako bogatećenje, odušao sam da vam ipak malo „posolim“ pamet. Što se tiče same koncepcije programa,

Dakle, idući je u suštini dobar, ali ste napravili grešku u realizaciji programa. Rečimo da treba izvršiti selekciju kombinacija od 18 brojeva, ali ne prvih 18 (1—18), već po želji korisnika (dakle umesto 1 napr. 3, umesto 2 — 7, 3 — 15 ... 18 — 37) vaš program je nemoćan, jer treba izvršiti selekciju po stvarnim brojevima (korisnikovim) u sistemu, a ne njihovim rednim brojevima (u primeru 1—18).

Ovo važi ako sam dobro razumeo rad programa. Ako je tako, predlažem vam da pre testiranja kombinacija ubacite potprogram za „prevodenje“ generisanih kombinacija u stvarne kombinacije sa brojevima izabranim od strane korisnika.

Pošto ovih dana treba da položam Matematiku ili pozabavio sam se vežbe radi verovatnoćama u lotu. Saljem vam svoje rezultate (važje za izvlačenje 7 brojeva od 33):

(\*) BROJ BROJEVA KOJI NISU IZVUČENI NI JEDNOM U N IZVLAČENJA

vijenje i poboljšanje? Umesto kratkog algoritma — čitava teorija i te silne matrice Poispadaju ti oči iz glave dok ih sve ne izračunam! Merlin se zamislio i, sasvim neočekivano, priznao svoju grešku. Ali grešku je priznao samo zato što je u rukavu, kao i svi čarobnjaci, imao novu trik. Trik se sastojao u efikasnom računanju matrice PUT. Evo opisa Merlinovog algoritma.

Označimo se PUT[K] matricu čiji je element put K (i, j) jednak true samo ukoliko postoji put od čvora i do čvora j koji ne prolazi ni kroz jedan čvor označen brojem većim od K (osim, možda, samih čvorova i, j). Odmah se uočava da je PUT O polazna matrica A, a da je PUT [N] tražena matrica PUT. Jasno, cilj nam je da što brže izračunamo matricu PUT[N]. Postavlja se pitanje da li je moguće uo matricu dobiti iterativnim postupkom, odnosno da li postoji neka veza između matrice PUT[K-1] i PUT[K]. Odgovor je potvrđen. Jer, za bilo koja dva čvora i, j ukoliko je vrednost od put K-1 (i, j) jednaka true, tada je i vrednost od putK (i, j) jednaka true. Jedini slučaj kada je vrednost putK(i, j) jednaka true, a da je pri tom vrednost putK-1(i, j) jednaka false, je moguće kada postoji put od čvora i do čvora j koji prolazi samo kroz čvorove 1, 2, ..., K, ali ne postoji put koji prolazi samo kroz čvorove 1, 2, ..., K-1. To znači da taj postojeći put svakako prolazi kroz čvor K, pa ga možemo razbiti na put od čvora i do čvora K i na put od čvora K do čvora j. Ova dva „potputa“ usput prolaze samo kroz čvorove 1, 2, ..., K-1, pa vrednosti od putK-1 (i, K) i put K-1 (K, j) u ovom slučaju moraju biti true. Prema tome, put K (i, j) = true samo ukoliko je tačan bar jedan od uslova

1) putK-1 (i, j) = true

2) putK-1 (i, K) = true i putK-1 (K, j) = true.

To znači da je putK(i, j) jednako putK-1 (i, j) OR (putK-1 (i, K) AND putK-1 (K, j)). Algoritam za izračunavanje matrice PUT[K] na osnovu matrice PUT[K-1] je dat na slici 5.1. Ovaj logički izraz se može još pojednostaviti tako da ga ne računamo celog za sve vrednosti i, j. Jednostavnija verzija je data na slici 5.2.

## Voršelov algoritam

Konačno, stigli smo do cilja! Tu je brži i jednostavniji algoritam za izračunavanje matrice PUT i to bez primene rekurzije (slika 6). U knjigama se ovaj algoritam vodi kao Voršelov algoritam, po navodnom pronalazaču. Nije nam poznato da li je Voršel poznavao čarobnjaka Merlina i da li je od njega kopirao ovaj algoritam. Bilo ako bilo, kraj Artur je nakon svih muka bio zadovoljan rešenjem problema.

Usput valja primetiti da je, prema legendi, vitezova okruglog stola bilo samo 12 i da se običnim crtanjem mape vrlo brzo može ustanoviti koji su dvorci odsedeći od ostalih. Ipak, cela priča je poučna, jer za velike vrednosti N crtanje nije baš najkraći put za rešenje

### Nešto treće

Kad se pomenu loto i računar, oni koji se „ne kockaju s državom“ najčešće imaju jednu od ove dve asocijacije:

— Učitaš program, pritisneš tirkicu i računar ti odštampa sedam brojeva koji će biti izvučeni sledećeg utorka.

ili:

— Sve je to besmislica! Šanse da pogodis sedam brojeva su jedan prema petnaest miliona, sa računarnom ili bez nje.

Istina je, kao i obično, nešto treće. Niko i ne igra samo sa sedam brojeva. Po pravilima Jugoslovenske lutrije, igrač mora da uplati najmanje dve kombinacije. Krajnosti, to jest da se uplate dve istovetne kombinacije, ili da se odigra 14 različitih brojeva, obično se izbegavaju. Igrač će neke brojeve ponoviti, neke neće, pa će u dve kombinacije najčešće odigrati 10 do 12 brojeva.

### Prva pomoć

Ako uplatimo dve, ili 20, ili 200 kombinacija, računar nam može pomoći da ne bude „svaka vaška obaška“, nego da sve odigrane kombinacije čine jednu celinu koja će, pod određenim uslovima, imati jasno definisanu i čvrstu garanciju u odnosu na dobitak. Čovek treba da odredi brojeve i uslove, ostalo je posao računara.

strukcije CP u programskim redovima 1550 i 1600.

Kod ostalih brojeva treba odrediti i donju i gornju granicu. Drugi broj je na adresi GKO+1, treći na adresi GKO+2 i tako dalje. Kako se ovi rasponi ograničavaju vidimo u rutini OSTALI BROJEVI. Za štitače s manjim programerskim iskustvom možda će biti korisno malo poređenja sa bežiksom — programski redovi od 1540 do 1580 ovako bi se „preveli“:

$$K0=39 \binom{32}{39}^N$$

(\*) BROJ BROJEVA KOJI SU IZVUČENI JEDNOM U N IZVLAČENJA

$$K1=7 \cdot N \binom{32}{39}^{N-1}$$

(\*) BROJ BROJEVA KOJI SU IZVUČENI DVA PUTA U N IZVLAČENJA

$$K2 = \frac{32^{N+2}}{39} \binom{7}{39} \frac{N(N-1)}{2} \binom{39}{39}^N$$

(\*) BROJ BROJEVA KOJI SU IZVUČENI TRI ILI VIŠE PUTA U N IZVLAČENJA

$$K3=39 \cdot K2 - K1 - K0$$

Kako se ovo može primeniti: za 5 izvlačenja K05=14,5 K15=15,86 K25=6,94 K35=1,7, a za 6 izvlačenja K06=11,9 K16=15,62 K26=8,54 K36=2,94. Odatle možemo izračunati:

$$\Delta K0 = K05 - K06 = 2,6$$

$$\Delta K1 = K15 + \Delta K0 - K16 = 2,84$$

$$\Delta K2 = K25 + \Delta K1 - K26 = 1,24$$

$$\Delta K3 = 7 - \Delta K2 - \Delta K1 - \Delta K0 = 0,32$$

To znači da će iz grupe brojeva koji u poslednjih 5 izvlačenja nisu bili izvučeni ni jedan put biti izvučeni 2 ili 3 broja, iz grupe brojeva koji su izvučeni jedan put biće izvučeno 3 broja, itd. Ovo važi ako su odstupanja brojeva (1) od realnih izvlačenja relativno mala. Ako su veća mora se primeniti univerzalnije sredstvo — hipergeometrijska raspodela. One nam daje verovatnoća da je

### Druga pomoć

Brojeve možemo odrediti napamet, slučajnim izborom, izvlačenjem oduljica, skupljanjem datuma rođenana mlili i drugih, još na stotina drugih načina. Jedan od njih je statistička analiza do sada izvučenih dobitnih kombinacija.

Pitanje je samo može li statistika pružiti koliko-toliko pouzdane podatke? Odgovor je: može. Primer? Evo:

Otkako se u Jugoslaviji igra loto (dvadesetak godina) još se nije dogodilo da se ponovo dobijna kombinacija, to jest da svi izvučeni brojevi budu isti kao u nekom od prethodnih kola. Otkako se igra loto sa sedam brojeva, (pošto manje od dve godine), još se nisu pojavile dve dobitne kombinacije koje imaju više od pet istih brojeva. Zar nije logično da iz svog sistema izbacimo sve kombinacije koje imaju šest ili sedam istih brojeva kao bilo koja prethodna dobitna kombinacija? Uplatiti ćemo ostete smanjiti, a verovatnoća za dobitak će se gotovo neditnuti.

Naravno, postoje mnogo suptilnije analize, ali bez obzira na to koliko bi složen način da se to njih dode, kad se rezultati predoče i neupućenima, crno na belo, sve ojednom postaje očigledno i lako shvatljivo. Ovom stazom se dolazi do pravog putokaza za svakog igrača: pojedinačni brojevi se zaista ne

mogou prognozirati, ali se neki skopovi brojeva s velikom verovatnoćom mogu eliminisati, a neki favorizovati.

Ni tu nije kraj priče. Računar nam je pomogao, recimo, da favorizujemo nekoliko skupova koji zajedno imaju 25 brojeva — ili 480700 kombinacija. Zato nam je potrebna i

### Treća pomoć

U jednom prognoziranom skupu su, na primer, brojevi 1, 2, 3, 4 i 5. No statistika nam takode kaže da se od tih pet brojeva u dobroj kombinaciji mogu pojaviti najviše dva — ukoliko se ne desi neko čudo koje će igrači loto posle mesecima prepričavati.

Znači, treba da odigramo svih pet brojeva, ali tako da se u dobroj kombinaciji pojave najviše dva. Zar računar nije napravljen da bi obavljao upravo ovakve poslove?

Ovakvim i sličnim uslovljavanjem postizemo da se uplata višestruko smanji i postane primerena našem džepu. Što je uslova više, sistem se više skraćuje, ali se i gubi verovatnoća da se svi ti uslovi istovremeno steknu. No, ako se steknu, sistem koji smo napravili pomoću računara garantovano će nam doneti nagradu.

Neki kažu da je ipak najpametnije ne igrati loto. Možda su u pravu.

LET A=PEEK (GKO+1)  
IF A>=7 THEN GOTO GENER  
IF A<3 THEN GOTO GENER

Igrači koji vole sisteme sa fiksnim brojem imaju priliku da u program vvrstite rutinu FIKS. Fiksirani broj se može napisati iz naredbe CP u redu 1570, a može se definisati na početku programa, tamo gde su ostale promenljive i konstante, kao što je prikazano na listinju. Labelu NAST u ovom

izvučeno tačno K brojeva iz grupe od N brojeva, i glasi:

$$P[S=K] = \frac{N!}{K!} \binom{39-N}{7-K}$$

Primer: verovatnoća da se od N=13 brojeva izvuče K=4 broja (u sledećem izvlačenju) je jednaka P[S=4]=0,12 (u stvari, za N=13 je P<sub>max</sub>=P[S=2]=0,33, dakle svaki treći put možemo očekivati da iz grupe od 13 brojeva (bilo koji) budu izvučena tačno 2).

Možda možda pomoći i očekivati broja izvučenih brojeva za grupu od N brojeva: E(N)=N (primer: za N=13 E(13)=2,362, dakle možemo očekivati da se iz grupe od 13 brojeva izvuču 2).

Dakle, povoljnim uslovljavanjem kombinacije prako grupa brojeva iz kojih u kombinaciji mora biti tačno određen broj možemo umnogome smanjiti broj kombinacija, a time i uplatu.

Nadam se da sam vam bar donekle pomogao sa matematičke strane gledanja na problem. Siguran sam da vam programski deo posta neće biti svaki teško, ukoliko odlučite da ove formule pretočite u program.

Puno pozdrava programerima koji se bave programiranjem.

Jugoslav Durović, Beograd



silažu treba shvatiti uslovno. To je jednostavno početak neke druge, ma koje rutine.

### Redosled nevažan

Podsetimo se još jednom da redosled rutina za testiranje i eliminisanje kombinacija ne utiče na korektan rad programa. Važno je samo da sve te rutine budu upisane između labela TEST i MEM.

Znači, možemo izabrati rutine koje nam odgovaraju i samo njih vvrstiti u program ma kojim redosledom. Nešto ipak ne treba zanemariti. Neke rutine se izvršavaju brže, neke sporije. Korisno je da najpre upišemo one koje su brže i time odzujemo šansu sporijim rutinama da testiraju neke, već eliminisane, kombinacije. Program će tako ukupan posao obaviti za kraće vreme.

Kad je već reč o tome, rutine objavljene u ovom broju brže su od onih s kojima smo se upoznali do sada. A kakve će biti sledeće rutine i kako ćemo ih upotrebiti — o tom potom.

Žarko Vukosavljević



# Žmurke sa Z80

*Relokatibilnost programa potrebna je iz praktičnih razloga. Svaki računar doživljava mnogo promena i njegov operativni sistem i osnovni programski jezik (bejziki) se dopunjavaju nebrojeno puta. Nove naredbe i potprograme obično piše više ljudi, i nemoguće je da se svi oni sporazumeju oko toga gde će svaki od novih programa biti u memoriji. Zbog toga je neophodno da se sve rutine, čija je priroda takva da saraduju sa ostalim, znanim i neznanim, delovima sistema, naprave tako da budu pomerljive, tj. da bez razlike rade na bilo kom mestu u memoriji. Kao i uvek postoji više načina da se to postigne.*

Svaki procesor poznaje dva osnovna načina adresiranja: apsolutno i relativno. Kao primer, dobro će poslužiti instrukcije za skokove. Standardni format prve izgleda ovako:

JP NNNN

Broj NNNN je dužine dva bajta i predstavlja adresu od koje se nastavlja program posle ove instrukcije. Procesoru će biti sasvim svedeno na kom se mestu ova instrukcija nalazi — njeno značenje je uvek isto. Druga instrukcija se obično ovako predstavlja:

JR NN

U ovom slučaju broj NN je dužine jednog bajta. Procesor će ovu instrukciju interpretirati u zavisnosti od mesta na kom će naći na nju, pa je odatle potekao i naziv relativni skok (jump relative). Neka je broj NN jednak 10. Kada se instrukcija nalazi na adresi 10000, program će se posle instrukcije nastaviti od adrese 10012. A ako je instrukcija na adresi 10100, nastavak će uslediti na adresi 10112. Poslednja instrukcija se upotrebljava kad god je to moguće, jer zauzima manje prostora od apsolutnog skoka (JP), ali njeno ograničenje je u tome što može da dosegne samo 129 bajta veće adrese i 126 bajta manje. Na primer, ako pokušate da asemblirate instrukciju

10000 JR 10020

dobićete prilično energičan raport o grešci, i jedino što preostaje je da se slovo R pretvori u slovo P.

Kada su u pitanju pozivi potprograma, nije potrebno trošiti mnogo reči: kod procesora Z80 zastupljeni su samo apsolutni pozivi.

Još je preostao pristup podacima koji se nalaze u memoriji. Pridev „relativno“, i to sa velikom rezervom, mogao bi da se upotrebi kada se podacima pristupa posredstvom jednog od tri registra: para ili nekog od indeksnih registra:

LD (HL),A  
LD D,(Y+10)

Naravno, prethodno je potrebno napuniti određeni registar adresom na kojoj se nalaze podaci. Sa druge strane imamo primer apsolutnog adresiranja prilikom manipulisanja podacima

LD (NNNN),A

Pošto smo utvrdili koje instrukcije su nepomerljive, pokušajmo da rešimo problem na nekoliko načina.

## Za kratke programe

Kada su u pitanju relativno kratki potprogrami, dužine 40—50 bajta, postoje dva načina da se mašinc učini relokatibilnim. Jedan je vrlo jednostavan: potrebno je malo se napregnuti i izbeći sve apsolutne skokove, pozive potprograma i adresirati isključivo relativno sve podatke. Statistika pokazuje da otprilike jedna četvrtina mašinskih programa (prosečnijih) otpada na apsolutne pozive i skokove. To znači da je u programu dužine 40 bajta potrebno izbeći oko tri instrukcije koje u sebi sadrže adresni podatak. Između ostalog, moraćemo da se oprostimo od poziva potprograma koji se nalaze u okviru našeg mašica.

Drugi način nije posebno preporučljiv, naročito kada su u pitanju rutine kod kojih je važna

brzina izvršavanja. Dakle, bavićemo se samo pozivima potprograma i apsolutnim skokovima. Najvažniju ulogu će odigrati dve kratke rutine preko kojih bi trebalo da idu svi pozivi i skokovi. U pitanju su egzibicije sa steakom i adresnom aritmetikom. Autor ovog teksta je pripremio dve rutine, jednu za „amstrad“ i drugu za „spektrum“ — dva najpopularnija računara sa procesorom Z80:

Napomenimo još i da je neophodno da se rutine rel\_sr i rel\_ip uvek nalaze na jednom

da su oba programa iz pomenutog paketa relokatibilna (pomerljiva). Engleski programeri nisu koristili nijedan od dosad navedenih metoda. Jednostavno su pisali programe onako kako su navikli, a problem pomerljivosti su ostavili za kraj. A tada je to bilo rešeno na najbolji mogući način.

Na osnovni kod programa prilepijena su dva dela. Ispred se nalazi kratki program koji neto silno petlja za glavnim delom, i čovek ima utisak da su pokušali da zaštite svoj izum od kopiranja. Naravno, to je pogrešno, jer ljudi iz Hisoft-a

amstrad: rel_sr:	DI EX (SP),HL PUSH HL POP HL INC HL INC HL INC HL INC HL EX (SP),HL DEC SP DEC SP	spectrum: rel_sr	DI EX (SP),HL PUSH HL POP HL INC HL INC HL INC HL INC HL EX (SP),HL DEC SP DEC SP
rel_ip:	DI EXX POP HL PUSH AF INC HL LD E,(HL) INC HL LD D,(HL) INC HL ADD HL,DE POP AF PUSH HL EXX EI RET	rel_ip	EX (SP),HL EX AF PUSH DE INC HL LD E,(HL) INC HL LD D,(HL) INC HL ADD HL,DE POP DE POP AF EX (SP),HL RET
Evo i nekoliko primera:		Pisati	
Umesto CALL lab_sr		CALL rel_sr CALL lab_sr:nxt_sr nxt_sr: ... CALL Z_rel_ip JP Z_lab_ip:nxt_ip nxt_ip: ...	
JP Z_lab_ip			

mestu u memoriji. Moguće je napisati slične potprograme i za operacije premeštanja podataka, ali tada bi dužina programa već postala neprijatno velika. U svakom slučaju, ovaj način previjanja pomerljivih mašinskih programa nije naročito pogodan, a naročito ne kada su u pitanju lole duži mašinci, koji prelaze granicu od 255 karaktera.

Postoji još mnogo sličnih rutina i rutina, ali razlike među njima su minorne tako da bi bilo neprikladno trošiti prostor na njih. Zato ćemo preći na jedan drugučiji i bitno bolji način da se mašinski program učini relokatibilnim.

## Majstori svog zanata

Sigurno je da se u blizini svojih računara sa procesorom Z80 i atributom „kućni“ nalazi kasete (diskete) sa poznatim programskim paketom za mašinske programere Devpac. Tvorac ovog popularnog programskog alata je engleska kuća Hisoft. Ono što nas trenutno zanima je činjenica

nikad ne rade slične stvari. Sa druge strane, na višim adresama nalazi se izvesna tabeleta podataka, pomoću koje program na početku pravi čitavu stvar pomerljivom. A evo i na koji način.

U svakoj instrukciji koja sadržava adresni podatak, upisana je adresa kao da je sve asemblirano sa instrukcijom ORG 0. Na taj način dobijena je u stvari udaljenost od početka programa. A u tabeli podataka, koja se nalazi na kraju, upisana je adresa na kojoj se nalazi dotični adresni podatak. Sada već nije problem da se sve to iskombinuje u stvarnu memoriju. Pri tome je neophodna i mala, ali značajna pomoć od strane bejzika. Potrebno je da se prilikom poziva programa iz bejzika na neki način prosledi adresa na koju je program uticao. Postoji mnogo načina na koje se to može uraditi. Na „spektrum“, recimo, posle upotrebe funkcije USR NNNN, u registarskom paru BC nalazi se broj NNNN. Opet, projektanti „amstrada“ su se odlučili za to da korisniku propuste izbor i broj argumenata koji se šalju mašinskom programu. Inače, programeri

```

010 *
020 code=>defu *
030 op=>op2=>op2a**
040 RETURN
050
060 skip:=skip:=fig:=1+rec:=fig:=0: fig:=0: fig:=0: fig:=0: fig:=0
070 ON INSTR('skip,fig,code') GOTO 900,100,100,100,110
080 IF (code=>AND code=>code2) OR INSTR('code') THEN RETURN
090 skip:=1: fig:=1: fig:=1: fig:=1: fig:=1
100 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
110 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
120 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
130 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
140 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
150 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
160 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
170 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
180 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
190 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
200 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
210 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
220 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
230 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
240 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
250 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
260 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
270 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
280 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
290 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
300 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
310 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
320 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
330 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
340 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
350 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
360 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
370 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
380 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
390 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
400 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
410 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
420 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
430 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
440 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
450 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
460 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
470 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
480 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
490 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
500 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
510 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
520 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
530 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
540 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
550 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
560 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
570 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
580 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
590 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
600 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
610 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
620 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
630 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
640 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
650 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
660 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
670 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
680 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
690 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
700 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
710 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
720 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
730 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
740 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
750 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
760 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
770 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
780 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
790 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0
800 skip:=0: fig:=0: fig:=0: fig:=0: fig:=0

```

vrio cene ove programe, izmdu ostalog baš i zbog toga što su pomerljivi i tako da prema potrebi mogu da se učitaju na bilo koje mesto u memoriji.

A sada, evo i rutine koja se nalazi ispred programa, i koja prepravljaj program tako da ovaj radi svuda u memoriji:

```

LD HL,length
LD C,(X)
LD B,(IX+1)
ADD HL,BC
JR ENTRY
EX DE,HL
ADD HL,BC
PUSH DE
PUSH HL
LD E,(HL)
INC HL
LD D,(HL)
EX DE,HL
ADD HL,BC
POP HL
LD (HL),E
INC HL
LD (HL),D
POP HL

```

```

entry: LD E,(HL)
INC HL
LD D,(HL)
INC HL
PUSH BC
LD A,D
LD A,C
OR E
LD B,C
LD POP BC
LD JR NZ,lope
LD LD A,A
LD LD H,B
LD LD (HL),A18
INC HL
LD (HL),46
prog: ...

```

(Napomena: tabela length predstavlja zbir dužina glavnog programa i rutine koja je iznad ispisana. Ova rutina nije sasvim identična onoj iz Hisoft-ovih programa, ali radi potpuno isti stvar. Od labela prog počinje glavni program.)

**Tabela adresa**

Na ovom mestu se postavlja vrlo logično i neizbežno pitanje: kako formirati tabelu adresa?

Pitanje je vrlo neprijatno, ali postoje samo dva odgovora. Prvi način je da se zasađu rukavi, i da se pohaba tastatura računara. U dobro strukturiranom programu dužine pola kilobajta, potrebno je pronaći oko 45 instrukcija i svaku pribelježiti. Ovak način autor ostavlja drugovima naklonjenim administrativnim poslovima. Posao je krajnje dosadan, a poznato je, dosadni poslovi se prepuštaju računaru. U tu svrhu je napisan i uz tekst priloženi bejzik program. On zamenjuje čoveka gde god i koliko je to moguće, ali ipak ne može da obavi ceo posao sasvim sam.

Ali, prvo nekoliko reči o strukturi programa, tj. podataka. Program razlikuje tri osnovne vrste podataka: (1) instrukcije, (2) tablele adresa i (3) podatke. Na početku korisanik mora da pokaže gde se nalazi koja vrsta podataka. Tada računar među instrukcijama odabira one koje je potrebno relokatibilizovati (ovo liči na klejbe-zabjele), one koje nemaju adrese podataka u svom sastavu i, na kraju, one za koje nije siguran u koju grupu spadaju. Podaci se preskaču, a tablele adresa se uzimaju u obzir cele.

Čovekov posao se sastoji u sledećem: prvo je potrebno učitati neki assembler (Genu na primer) i tekst assemblerskog programa koji je potrebno obraditi. Na početku programa upisuje se instrukcija ORG 0. Zatim se vrši asembliranje, pri



ženu je neophodno izabrati opciju 16 da se program ne bi stvarno ugledao na nutu stranici. Potrebno je pratiti čitav proces asembliranja i zapisivati gde se nalaze podaci ili tabele adresnih podataka. Pri tom se zapisuju adrese prvog i poslednjeg bajta tabele ili grupe podataka. Na kraju treba zapisati i adresu prvog slobodnog bajta iz programa, jer će na tom mestu program formirati tabelu. Naposljetku, mašinar se anina na traku, pa dolazimo do učitavanja bajzik programa.

Pošto startujete učitani bajzik, bićete primorani da bacite pogled na svoje beleške iz prethodne seanse sa asemblerom. Zatim će biti učitani mašinar koji se obrađuje. Na kraju dva pitanja: jedno je estetske prirode — da li će brojevi da se prikazuju u heksa ili decimalnom formatu i drugo — da li da se program konsultuje sa korisnikom pre svake akcije. Sada započnite obradu mašinskog programa i opet pitanja za pojedine instrukcije, u smislu da li ih uvrstiti u tabelu ili ne. Na kraju će vas program izvestiti o dužini tabele i celog koda, na početak će biti prenesena rutina za relociranje i posle pitanja o imenu programa, sve će biti poslato na neki spojašnji medijum. Prilikom učitavanja potrebno je program pozvati sledećim nizom naredbi:

LOAD name\$, address: CALL address.

Na taj način, mašinar se može učitati na proizvoljno mesto i startovati, čime se vrši korekcija adresnih podataka u programu i, eventualno, inicijalizacija.

Rad nekog programa je najbolje prikazati kroz primer. Ovu puta, kao zamorče će nam poslužiti mašinski program objavljen u „Računarski“ 24 u okviru teksta o doprincu za crtanje kruga. Tekst linije 10 potrebno je promeniti u ORG 0, i asemblirati program uz opciju 16. Za podatak moraju se izabrati sledeći intervali: #6—#C, #1B—#37 i #24F—#280, a za tabele adresa: #D—#E, #1A4—#1AD, #1B1—#1BA, #1BE—#1C7 i #1CB—#1D4. Na pitanje o dužini potrebno je odgovoriti sa #2B1, a na preostala dva pitanja korisnik može odgovoriti prema raspoloženju. (Umesno pozivke — 4, sa bajzikom se mora koristiti ampersand —&, ako se koriste heksa brojevi.)

## Ništa nije sigurno

Na ovome se ne završava posao priprevedaka, jer se ispostavilo da ima isušiva mnogo zamki koje naš program nikako ne može sam da izbegne. Obratimo pažnju na naredbe u kojima se u neki registarski par ubacuje konstanta. Jedino autor programa i pažljivo posmatrač mogu znati kakav je smisao pomenute konstante — adresni ili aritmetički. Ako je u pitanju adresni podatak kao u sledećoj sekvenci

LD HL,16A  
LD A,(HL)

I ako ga ne uvrstimo u našu tabelu čitav trud će nam biti uzaludan, jer dovoljno je da se samo jedan podatak propusti. A opet, ako ga uvrstimo u tabelu i ispostavi se da je to bila konstanta potrebna za neki aritmetički proračun, težina greške će biti lita. Zato je potrebno biti vrlo oprezan, i nekoliko puta pratiti listing dok se program asemblira. Naravno, kada program nalde na ovakvu instrukciju, pre nego što bilo šta učini sa njom pitajte čoveka za mišljenje. Primeri ovakvih instrukcija, koje sadrže adresni podatak, nalaze se u našem program-primeru u sledećim linijama: 20, 30, 210, 1340, 2510, 3190 i 3210.

Prilikom obrade apsolutnih skokova, poziva potprograma, i apsolutnog pozivanja memorije program obrada pažnju na adresni podatak koji se nalazi u okviru instrukcije. Ukoliko je on (po vrednosti) u okviru samog programa postoji jak razlog da se njegova adresa unese u tabelu. Međutim, ne mora tako da bude. Poslužimo se opet ovim programom koji je asembliran na adresima 0000—#0280. Adresa #16 se svakako

nalazi u okviru programa, pa bi sledeća instrukcija morala po tom kriterijumu biti uzeta u obzir

JP #16

A šta ako je pisac programa u izvornom tekstu napisao

JP pcode?

Oni koji poznaju operativni sistem „amstrada“, znaju da se skraćenica pcode upotrebljava umesto adrese #16 i da se na njoj nalazi kratka rutina operativnog sistema koja obavlja skok na adresu iz registra DE. U ovom slučaju bila bi greška uzeti u obzir i ovaj podatak. Postoje dva načina da se to izbegne. Prvi je da se prilikom asembliranja zapami adresa instrukcije, pa da se predstavi bajziku kao podatak. Drugi način bi bio, da se na pitanje o kontroli odgovori potvrdo (sa Y), pa će tako i program kad nalde na pomenutu instrukciju pitati korisnika za savet. Ovakvi primeri su retki, pa u našem mašinaru i nema nijednog sličnog slučaja.

Tako je ispaldo da praktično nijedna instrukcija nije sasvim sigurna. Naravno, sve ovo ne bi trebalo da predstavlja problem korisniku koji je dobro uočio strukturu programa koji se obrađuje i koji krajnje oprezno prilazi ovom poslu. Doduše, verovatno će ovaj program u početku zadati dosta muka svakom ko ga bude koristio, ali, kada se stekne navika, sve će teći tako da bolje ne može biti.

## I poneka vrlina

Pošto je autor ovih redova nemilosrdno izneo svo laša svojstva svog programa i ovakvog načina pravljenja pomerljivih programa, vreme je da se upoznamo i sa njihovim vrlinama.

Prvo što se primjećuje je činjenica da mašinar i posle obrade zauzima prostor iste veličine i da se izvršava podjednako brzo. Prostor koji zauzima tabela adresa može se korisno upotrebiti posle obrade. Na taj način dobijamo program sa potpuno istim svojstvima, koja ima i onaj koji je tek asembliran. Takođe, ne treba zaboraviti ni programerski stil, a ovo je jedini način da se on sasvim sačuva. Tako se dobijaju mnogo kvalitetniji programi nego kada je programer primoran da zbog pomerljivosti izbegava pojedine konstrukcije.

Osvrnimo se i na program pisan na bajziku. Autor nije imao nameru da napravi neko čudo od veštačke inteligencije, već jednostavno pomodnika koji će čoveka osloboditi dosadnih dužnosti. Svakako da je mogao da se napravi program koji će analizirati mašinar i pokušavati da ga shvati da bi mogao da odredi koje instrukcije sadrže adresne podatke. Međutim, zašli bismo u krajnost i cilj više ne bi bio isti. Zato ovaj program verovatno ima najbolji odnos jednostavnosti i koristi.

Na kraju ćemo posvetiti i malo pažnje jednoj bajziku, koja je iznela najveći teret. U pitanju je instring funkcija IN\$TRA, čiji je zadatak da pronađe jedan string u nekom drugom nizu karaktera. Da nije ugrađena u Locomote bajzik „amstrada“, program bi bio mnogo duži i mnogo neeleгантiji. Tako je program, iako u bajziku, solidno strukturiran, pa se može protumačiti i bez komentara.

## Pomični ekran

Biblioteka programa našeg kluba polako se uvećava zahvaljujući sve brojnijim prilozima čitalaca. Ovu puta pažnju posvećujemo programu **Vladana Vučkovića** (Niš, Zetska 6/35). Reč je o rutini za skrol, odnosno o programu koji siliku na ekranu pomena nagore za određeni broj redova i to željenom brzinom.

Program se sastoji iz dva dela, glavnog programa i potprograma. Potprogram ADDR nalazi adresu prvog bajta svakog pojednog reda na ekranu. Koji će to red biti zavisi od sadržaja akumulatora. Smatra se da su redovi numerisani odzgod nadole, tj. broje se od vrha prema dnu ekrana. Izračunata adresa smešta se u registarski par HL. Sam princip izračunavanja objasnimo prilikom nekog detaljnijeg bajvljena organizacijom spektrumeove video-memorije.

Glavni program, koristeći adresu dobijenu od potprograma, prebacuju liniju za jedno mesto naviše. Promenom argumenta i naredbi labelirano sa T1 određujemo okvir u kome se vrši pomeranje. Ako umesto 0 napisemo, na primer, 64 program će dizati sadržaj donje dve trećine ekrana. Argument u naredbi obeležen sa T2 predstavlja brzinu podizanja (izraženo u brojevima redova) i uzima vrednosti od 1 do 16. Od značaja je i argument u naredbi obeležen sa T3. To je već spomenuta brzina umanjena za 1. Inicijalno zadati su brojevi 0, 3 i 2 tako da program podiže sadržaj celog ekrana. Vladan nam je dao i „pokice“ za svoj program, tako da se poukovanim adresama 50001, 60007 i 60021 može uticati na rad programa i iz bajzika menjati argumenti.

Vratimo se sada nešto elementarnijim stvarima i objasnimo upotrebu instrukcije LDIR (Load, increment and repeat) koja je korišćena i u Vladanovom programu. Instrukcija je namenjena prenosu sadržaja jednog memorijskog bloka u drugi. Za to se koristi sledeća sekvencija naredbi:

Dejan Muhamedagić



```

10 |+++++
20 |+ GLAVNI PROGRAM +
30 |+++++
40 | DRG 40000
50 |
60 |A=Gornj,granica skrota
70 |-----
80 | LD A,0
90 | LOOP CALL ADDR
100 | PUSH HL
110 |
120 |A=Brzina skrota (lin.)
130 |-----
140 |72 ADD A,3
150 | CALL ADDR
160 | POP DE
170 | LD BC,32
180 | LDIR
190 | CP 191
200 | RET NC
210 |-----
220 |arg=brz,skrota - 1
230 |-----
240 |73 SUB 2
250 | JR LOOP
260 |-----
270 |-----
280 |-----
290 |-----
300 |-----
310 |-----
320 | ADDR LD C,A
330 | OR 1
340 | RRA
350 | RRA
360 | OR A
370 | RRA
380 | XOR C
390 | AND 24B
400 | XOR C
410 | LD H,A
420 | LD A,C
430 | RLA
440 | RLA
450 | AND 224
460 | LD L,A
470 | LD A,C
480 | RET
490 |-----
500 |-----
510 |+LADAN VUCKOVIC 1987
520 |+++++
    
```

LD HL, adresa početka bloka  
LD DE, određišta adresa  
LD BC, broj bajtova koje treba preneti  
LDIR

### Nepostojeća naredba

Poznat je da Z80 ne zna za CALL (HL), iako je naredba JP (HL) potpuno legalna. Ipak, jednom programerskom dosetkom možemo CALL (HL) uspešno simulirati. Potrebno je da negde u programu imamo sledeći programski red:

LABEL JP (HL)

Time je ceo posao završen. Kasnije, kad zatreba, dovoljno je da napišemo:

CALL LABEL

i računar će izvršavati nepostojeću naredbu CALL (HL). Naredba CALL LABEL takođe može biti na gde u programu.

Evo kako izgleda program koji će sliku sa ekrana premestiti na neko drugo mesto u memoriji. (U konkretnom slučaju reč je o adresi 50000).

```

LD HL, 16384
LD DE, 50000
LD BC, 6912
LDIR
RET
    
```

Kada vam slika ponovo zatreba, u video-memoriju je vraća program:

```

LD HL, 50000
LD DE, 16384
LD BC, 6912
LDIR
RET
    
```

Pređajemo vam da razmislite i o sledećem problemu. Upotrebom instrukcije LDIR i zahvatjanjem što većeg memorijskog bloka napišite program za pomeranje atributa nagore i nadole. Verujem da ćete problem rešiti sa svega desetak instrukcija.

### Kvadratni koren

Klub ovoga puta otvaramo prilogom inž. Sergija Čerškova (Dalmatinska 16, K. Gomilica). Dajmo reč autoru priloženog listinga:

U svom radu upućen sam na česta izračunavanja, tako da sam morao da pišem brze matematičke rutine i da ih, po potrebi, uključujem u program. Do sada su na više mjesta objavivane rutine za brzo množenje i daljenje celobrojnih binarnih brojeva. Moj doprinos biblioteci kluba bit će rutina za izračunavanje drugog korena iz 16-bitnih nepredznačenih brojeva. Broj bitova može se odrediti za svaki program posebno (8, 16, 32, ...). Algoritam koristi svojstvo kvadrata sume dva-ju brojeva:

$$(x+y)^2 = x^2 + 2xy + y^2 = x^2 + (2x+y)y$$

Kao i činjenicu da u binarnom sustavu radimo sa nulama i jedinicama, te se množenje realizira običnim pomeranjem broja ulijevo.

U Sergiovom prilogu uočavamo i neke male programerske tajne koje čine rutinu bržom. U glavnoj petlji nalazimo instrukciju LD H,C. Pogleđamo li bolje, vidimo da je na početku programa registar C inicijalizovan na nulu. Umesto navedene instrukcije moglo je, dakle, stajati LD H,O. Ali, instrukcija LD H,C izvršava se u 4 takta, a instrukcija LD H,O u 7 taktova kloka. Kako kroz petlju

```

5 SDR EDU #
10 |-----
20 |Inicijal. registara
30 |-----
40 |broj u IX
50 |-----
60 | LD IX,(BROJ)
70 | LD HL,B
80 | LD DE,B
90 | LD L,D
100 | LD B,B
110 | XOR A
120 |-----
130 | GLAVNA PETLJA
140 |-----
150 |Prvo prebacujemo gornja
160 |dva bita u radni reg.
170 |-----
180 |SDR_1 ADD IX,IX
190 | ADC HL,HL
200 | ADD IX,IX
210 | ADC HL,HL
220 |-----
230 |Isucavaj medjurazultat
240 |-----
250 |-----
260 | EX DE,HL
270 |-----
270 |Htrenutni rezultat
280 |Isucaviramo (2x)y
290 |u binarnom obliku
300 |-----
310 | LD H,C
320 | LD L,A
330 | ADD HL,HL
340 | ADD HL,HL
350 | INC HL
360 | EX DE,HL
370 |-----
370 |Isputujemo da li je sve
380 |u redu
390 |-----
400 |-----
410 | SBC HL,DE
420 | JR NC,SDR_2
430 |-----
440 |jako nije, restauriraj
450 |medjurazultat
460 |-----
470 | ADD HL,DE
480 |-----
490 |SDR_2 CCF
500 |-----
510 |uključuj carry u rezul.
520 |-----
530 | ADC A,A
540 |-----
540 |ponovi petlju za ceo
550 |izadani broj
570 |-----
580 |DJNZ SDR_1
590 |-----
600 |-----
610 |-----
620 |NA IZLAZU:
630 |Imerazultat
640 |Hlostatak
650 |IX,BC=0
660 |DE=nije od važnosti
670 |dipl.ing S.Cerškov
680 |-----
    
```

brzina pokretanja dovoljno velika, mi čujemo ton. Visina tona određena je frekvencijom. U našem slučaju to je broj pokretanja membrane zvučnika u sekundi.

Konstruktori spektruma su u ROM računara, počevši od adrese 048E heksadecimalno, upisali frekvencije tonova osnovne lestvice:  
C — 261.63  
Djs — 277.18  
D — 293.66  
Dis — 311.13  
E — 329.63  
F — 349.23  
Fis — 369.33  
G — 392  
Gis — 415.30  
A — 440  
B — 466.16  
H — 493.8

Nećemo se baviti načinom na koji Z80 interpretira podatke iz ove tabelle, već ćemo se okrenuti praktičnoj primeni jednog potprograma iz ROM-a. Izvršavajući potprogram sa adrese 949 Z80 će proizvesti zvuk čija je visina data u registarskom paru HL, a dužina u registarskom paru DE. Zvuk se proizvodi sledećom jednostavnom mašinskom rutinom:

LD HL, visina tona  
LD DE, trajanje tona  
CALL 949  
RET

Treba još samo znati kako odrediti visinu i trajanje tona:

$$HL = INT(3 \cdot 500 \cdot 000 / (f \cdot 8) - 30 \cdot 125)$$

$$DE = INT (f \cdot t)$$

Nemojte da vas zbune ovi nalazigdel komplikovani izrazi. Upotrebite bezik da vam ih izračuna: f označava frekvenciju, a t trajanje tona. Na primer, za ton C iz tabelle nalazimo da je f=261.63. Neka je trajanje tona pola sekunde, tj. t=0.5. Ove dve vrednosti unesemo i izraze i dobijamo brojeve koje treba upisati u HL i DE registarske parove, da bi računar odsvirao ton C.

Ukoliko ste malo veštiji možete napisati bezik program koji će vam proračunati podatke za HL i DE za neku melodiju. Te podatke prepisite zatim u assembler iz DEFW instrukcija i napravite petlju koja će ih čitati, prenositi u HL i DE i pozivati rutinu sa adrese 949.

Priredili:

Aleksandar Radovanović  
Žarko Vukosavljević

# Hakeri vole DEVPAC

*Do sada je za „spektrum“ objavljeno barem desetak raznih asemblera i disasemblera. Nedavno se pojavio i famozni „LASER GENIUS“, ali je većina hakera ostala verna DEVPAC-u. Stoga smo novi programski projekat za ljubitelje mašinskog programiranja posvetili upravo ovom paketu. Program XEN predstavlja proširenje DEVPAC-a skupom naredbi koje pojednostavljuju i olakšavaju programiranje na mašinskom jeziku.*

Zašto hakeri vole DEVPAC? Svojevremeno su se uz DEVPAC javili i mnogi drugi asembleri i disasembleri (npr.: Infrared, Ultraviolet 48, Assembler Editor, Zeus Assembler, Aspect, Spectrum Assembler . . .), ali su svi ostali u sjeni DEVPAC-a. Jedino je Zeus Assembler bio svojevremeno popularan, ali ni on nije DEVPAC-u „ni do koljena“. I pored nekih nedostataka koji nisu odmah došli do izražaja, GENS je bio najbrži asembler sa najviše mogućnosti. Namjerno kažem bio, jer se javio LASER GENIUS. Da vidimo zašto će većina hakera ipak voljeti DEVPAC. . .

## . . . a mrze laserske genijalce

Autor ovog teksta je također nabavio taj LASER GENIUS i uvjerio se u činjenicu da je to stvarno najbolji paket (assembler i disassembler) za rad na mašinskom jeziku. Mogu slobodno da kažem da je GENS slab i spor assembler prema assembleru LASER GENIUS-a, ali. . .

LASER GENIUS, između ostalog, posjeduje ekranski editor, povečan broj slova u redu, tokenizovan sors fajl, odvojeni tablicu simbola, mogućnost učitavanja sors fajlova drugih asemblera (pa i GENS-a), makro instrukcije (ljubi ba majka), „inteligentni“ disassembler (gotovo kao neki viši programski jezik, npr. fort) itd, itd. Po svemu ovdje LASER GENIUS je van konkurencije spektrovih asemblera/disasemblera. Po svojoj profesionalnoj izradi i mogućnostima sličan je assemblerima jačih kompjutera (pa čak i IBM-u), međutim, par „sitnica“ čine ovaj paket i na običnom spektrumu za običnog YU hakera posve neupotrebljivim.

Najvažnija od tih sitnica je upravo proporcionalna kvaliteta programa, a zove se dužina programa. Samo assembler je dug čitavih 23 Ki! Da, dobro ste pročitali, dvadeset i tri kilobajta. To je oko tri puta duže nego GENS. Učitava se na adresu 30000 (doduše, relokatabilni je, ali to ne pomaže mnogo), RAM od kraja asemblera do adrese 65535 je rezerviran za assemblerov sors fajl, ispred sebe (i, od 30000 prema dole) smješta se tablica simbola. Postavlja se pitanje: gdje će naš pravi MC kod? Za njega su slobodne adrese do tablice simbola, a to iznosi (ukoliko nema bajzika) samo nekoliko K (to je stvarno „samo“, jer je to za neki ozbiljniji program malo). Ako ga relocirate na više adrese, imate manje mjesta za sors fajl, pa je to mač sa dvije oštrice. Dakako, ovo se može ublažiti korištenjem prave ili lažne (Sinklerove) disketne jedinice, ili sa dodatnim RAM-om do ukupno 80K. Na žalost, činjenica je da većina YU hakera nije ni pošten štampač, a kamoli disketnu jedinicu.

Ako vam prije navedeno nije dovoljan razlog da ostanete na DEVPAC-u, evo vam malo soli na kraju: assembler i disassembler LASER GENIUS-a ne mogu istovremeno biti u memoriji!!! Pročitajte posljednju rečenicu još jednom. Nazdravljaj Asembler je dug 23 K, disassembler čak 15 K (tri puta duži od MONS-a), pa kad se to zbroyi napamet i provjeri preko kalkulatora i kompjutera, izlazi čitavih 38 K, a „spektrum“ ima slobodno oko 40,5 K. Ovo računanje je bilo suvišno jer iako bi po dužini nekako stala ova u memoriju, njihova interna organizacija memorije ne podnosi prisustvo drugog programa te dužine. Mislim da je ova činjenica dovoljan razlog da program, ma kako bio dobar, ipak ostane samo na kazeti. Rad tipa „učitaj asembler — ukucaj program — asembleraj — snimi sors fajl — učitaj disassembler — nadi bago — učitaj sors fajl — ispravi grešku“ brzo bi dojadio svakom programeru, a kamoli neke hakera, koji ne želi da program od njega pravi budalu?, kako to kaže Vlada Kostić.

## Dobri, stari, DEVPAC . . .

I tako, primjenjujući onu našu „Hvali more, drž' se kopna“ preuređenu u „Hvali LASER GENIUS, drž' se DEVPAC-a“ vratimo se temi iz naslova.

Dakle, pred nama je program koji predstavlja dodatak DEVPAC-u, ali na drugačiji način nego kako je to riješio Kostić u „Računarima 6“. Prvo, čitav je program u mašinciu, što je u suprotnosti sa Kostićevim mišljenjem da je dobro da je dio programa u jeziku, jer je tada iako neku opciju prilagoditi trenutnim potrebama. Međutim, mašinski program sa više opcija može biti kraći od kombinacije bejzik-mašinciu sa manje opcija, a bejzik prostor je posve slobodan (često se pišu mašinci koji prčkaju po jeziku). Uz to, dodavanje većeg broja novih opcija je nemoguće (i zbog ograničenog bejzik prostora i zbog malog menija). Uz put, želim naglasiti da je ideja menija sa strelicom stvarno fantastična (i autor ovog teksta je često koristio), ali je praksa pokazala da je baš razlog dodavanja novih opcija ograničio primjenu takvih menija na uslužne programe tipa obrade podataka i sl.

## . . . sa dodatkom XEN-a

Posvetimo se samom programu (u daljem tekstu XEN). Pri nalaženju kompromisa između slobodno za bejzik / slobodno za mašinciu odluka je pala da RAMTOP bude pušten na 24499, što za bejzik ostavlja slobodno oko 700 bajtova. Smatram da je to sasvim pristojno, jer haker i onako uglavnom piše mašinske programe, tek ponekad par linija u jeziku (koje obično koriste mašinske rutine); osim toga, mašinski prostor je u svakom slučaju vrijedniji od bejzik prostora. Sa tih 700 bajtova mogu se napraviti neki programčići koji su u mašinciu kompliciraniji (npr. računanje u pokretnom zarezu). XEN počinje na adresi 24500 i dug je 6988 bajtova. Odmah nakon njega, na adresi 31600 nalazi se MONS, zatim još nekoliko bajtova XEN-a (vektor interapt rutine) i na kraju GENS na adresi 37121. Sve zajedno zauzima adrese od 24500 do 45383 (početak GENS-ovog teksta) što iznosi 20883 bajta. Za tekst ostaje slobodno 20153 bajta.

## Šugarove šećerne fore . . .

Prvo što se primijeti pri startovanju programa su drugačiji, ljepši karakteri. Većina hakera za svoje izvišavanje koriste mali portabi crno-bijeli televizor koji u kombinaciji sa PAPER 0:INK 7 daje vrlo loš izgled „spektrumovih“ karaktera, posebno vertikalnih linija. Hakeri obično dugo bulje u ekran što napreže oči. Zato smatram da je uvođenje odlično dizajniranog Amstradovog karaktera u program na uštrb 768 bajtova memorije opravdano. Neki vlasnici „spektuma“ su preuredili originalne ROM-ove, pa ako bude zainteresiranih biće objavljena verzija koja koristi ROM set i oslobađa dodatnih 768 bajtova RAM-a.

UDG set karaktera je premešten unutar XEN-a, što je oslobodilo kraj RAM-a za 168 bajtova namjenjenih sors fajlu. Definirana su naša slova, a unutar XEN-a postoji mogućnost definiranja novih karaktera (o tome kasnije). Ako vam treba adresa UDG seta, možete je dobiti u okviru jedne naredbe (i o tome kasnije).

Program se iz bajzika uvijek poziva sa RANDOMIZE USR 24500. Ne postoji hladan i topli start (cold i warm start) kao kod GENS-a, jer program „zna“ da li je startan prvi put. Posjeduje svoj interni stek.







# Emulacija ili imitacija

**„Atari ST“ (sam ATARI još manje) više ne muči muku sa softverom. Njega sada ima napretek. Da se jednim programom može osigurati na stotine drugih programa Atari je uvideo već odavno, a sada su toga postali svesni i programeri (i nezavisne softverske kuće). Nabavkom programa ST. IBM-PC ST se, bez ikakvih hardverskih prepravki i dodatka pretvara u MS DOS mašinu. Potrebne su vam samo 3,5 diskete od 3,5 inča sa programima i MS-DOSom.**

„Atari ST“ još od svojih prvih dana ima „nesretnu sudbinu“ kompjutera-emulatora. Ugrađeni mikroprocesor Motorola MC 68000 na frekvenciji od 8 MHz sposoban je emulirati (uz neznatne gubitke na brzini) sve postojeće osmo i osmo/šesnaestobitne procesore (Z80, 6502, 6510, 8088... ). Naravno, u pitanju je potpuna softverska emulacija. Prvi ovakav emulator bio je CP/M emulator. On je, u osnovi, 90% već bio realiziran samim operacionim sistemom, koji je, u ime kompromisa, podlegao CP/M-68 kompatibilnosti. Ovakav emulator bilo je jednostavno izvesti i firma GST je to učinila bez većih problema. Program je čist od bagova i emulira sav CP/M softver.

## Možda malo prerano

Poslije CP/M pojavili su se „mekintoš“ hardverski emulatori — Rogblov, koji nije radio sa 50% softvera (uglavnom zaštićeno, a taj je i najkvalitetniji), a poslije njega i Aladinov, koji emulira oko 95% softvera, što je zbilja impozantan broj, a uz to omogućuje i sve ono što nije imao Rogblov emulator (dvostrano formatiranje, i uopće, formatiranje diskete na samom ST-u npr.).

U međuvremenu, u ST-ovskim krugovima najprije, a zatim i u računarskim časopisima, na sva zvana je najavljen Atarijev

hardverski PC emulatora za računare ST serije. No, od toga se odustalo (razlozi nam nisu poznati) i barem u slijedećih pola godine ne bismo trebali očekivati emulator u trgovinama.

No, ostavimo Atari i (zar ga baš moram spomenuti) Jacka Tramiela da brinu o svom emulatoru i pogledajmo novu listu softverske emulacije na Atarijevom ST-u.

U pitanju je softverski emulator autora Daniela Rosengartena koji je namjenjen emulaciji MS ili PC DOS-a, a pred autorom (teksta) se nalazi preuranjena verzija 1. MS-EM je program dužine oko 50K. Učitavan u memoriju, priprema ST za učitavanje MS DOS-a (ova priprema je prijeko potrebna zbog različitosti memorijskih mapa dvaju računara). Pošto startujete MS EM, PRG i on se učita, ekran mijenja boju u bijela slova-tamna pozadina (na crno-bijelom monitoru) baš onako kako to čini i sam IBM-PC, a na ekranu se ispisuje poziv da ubacite sistemski disk (to je onaj na kojem se nalazi MS-DOS) u drav A i potom pritisnete neki taster. Tu se javlja prvi problem: za sada se MS DOS (preporučuje se verzija 3.2) ne isporučuje sa emulatorom i morat ćete ga sami nabaviti ili presnimati na disketi od 3,5 inča (ako imate 5 1/4 drav, za vas su svi problemi riješeni) što u dogledno vrijeme neće predstavljati veliki problem, jer je i IBM „konačno“ prihvatilo

3.5 inčni disketni standard (sada opozicija roni suze i grize nokte, ali što će — danas od konzervativizma živi samo Margaret Tacher).

## Automatski start

Pošto ste disketu sa MS DOSom ubacili u drav i pritisnuli bilo koji taster — slijede još dva upita: želite li kontrolu nad tastaturom i želite li interapte? U normalnim okolnostima na oba pitanja odgovorite negativno, osim u slučaju ako ste već prethodno uvidjeli da u radu sa nekim programom imate problema (neki strogo zahtjevuju interapte; kao TURBO PASCAL npr.). Poslije toga se učitava sam sistem. Sto će se u toku učitavanja pojaviti na ekranu zavisi od datoteke AUTOEXEC. BAT koja se prva učitava i prenosi vam poruke kao što su: verzija operacionog sistema, datum, vrijeme... Ako AUTOEXEC. BAT nije kreirana za start sistema, morate (pošto se drav drugi put zaustavi) pritisnuti RETURN i na ekranu će osvanuti originalne IBM-ove i MICROSOFT-ove poruke o kopiraju, a nešto zatim i prompt A, koji daje do znanja da je aktuelan drav A i da očekuje naredbu (sve o sistemu možete pročitati u „Računarna 18“ u tekstu „PC bukvar“).

## Radi, ne radi...

Pošto na sistemskoj disketi nema niti jednog programa, morati ćete se pobrinuti i nabaviti neke koji su vam potrebni ili su vam pristupačni. Nije sigurno da će svi programi raditi (neki npr. nekada prorade iz prve, a nekada ih bez uspjeha možete učitavati cijeli dan upravo je to razlog što vam nismo u mogućnosti sastaviti tabelu „sigurnih“ programa). Autor je emulator isprobao sa MS DOS 3.2 i programima „Turbo pascal“, „Turbo lighting“, „Wordstar“, IBM Assemblerom i nekim programima za komunikaciju i svi su radili bez većih problema.

Sam emulator prilično (20—30%) usporava program, a postoji i velika vjerovatnost bezrazložnog kraha. No to nije za zamjeriti pošto znamo da je u pitanju prva verzija emulatora, koja je prije vremena izbačena na tržište. Autor programa obećava i novu (bolju i sigurniju) verziju u skorije vrijeme. Nema razloga da ne vjerujemo u tvrdnju o mogućnosti 90 postotnog emuliranja uz znatno veću brzinu. I ova verzija, u suštini, radi korektno, a u pitanju su neki sitniji bagovi koji u nekim slučajevima kobno utječu na rad programa, a svakako ih je moguće ispraviti.

Goran Rukavina

### Važna upozorenja, korisni saveti

1. Ako imate dva drava obratite pažnju na to da će po učitavanju sistema disk A biti preimenovan u disk B, a disk B u A!
2. Također, ako imate dva drava, obratite pažnju na to da ih sistem neće prihvatiti u slučaju da je jedan od njih jednostrani, a drugi dvostrani.
3. Ako imate kolor monitor ili TV modulator, iskoristite instrukcije MODE 40 i MODE 80 i vaša ST će postati pravi PC sa kolor grafičkom karticom rezolucije 640x200 ili 320x200 (u modu 40 su slova plave boje). Ovo traži većina grafički orjentiranih programa.
4. Ako vam se na ekranu pojavi neka prijava greške pri radu sa diskom ili silčno, da biste nastavili sa radom, ponovili ili prekinuli operaciju, morate više puta uzastopno, ili u kombinaciji sa nekim od susjednih tastera, pritisnuti A, R ili I (u zavisnosti koju funkciju odabiramo) sve dok ih računar ne primi.
5. Ne pokušavajte formatirati disketu ili kopirati sistemski identitni MS-DOS-a. Disketu isformatirajte u ST-ovom GEM-u, jer su formati identični-365K po disku-jednostrano. Sistem je moguće kopirati pomoću PROCOPY programa, jer se na sistemskom disku mora nalaziti jedna „skrivena“ datoteka dužine oko 1K, a nju nije moguće kopirati ni iz GEM-a ni preko RAM DISKA.
7. Pošto će vam prelazanje iz MS DOSA u GEM biti prijeko potrebno otkrit ću vam jedan način da to što bezbolnije učinite (bez resetiranja). Otkucajte DIR i kada počne listanje disk-dijektorija pritisnite CONTROL+C i naći ćete se u GEM-u. Na žalost, proceduru učitavanja MS-EMa i MS DOSa morat ćete ponoviti iz početka.
8. MS EM. PRG odmah pohranite na neki disk i smjestite u folder (omot) AUTO. Na taj način će se poslije svakog BOOTovanja (naravno, ako disk sa MS EM-om bude u dravju) MS EM automatski učitati i bit će izbjegnuti neki od problema (npr. pozicija mape ekrana u memoriji, jer ako MS EM učitamo iz GEM-a automatski za sobom povećamo i područje koje je MS GEM već adresirao za svoje potrebe, samim tim i rizik od kraha daleko veći.

# Metodi matrica

„Metodi matrica“ su šesti nastavak serije „Računarski algoritmi“ u kojoj prof. dr Dušan Slavić daje niz algoritama sa programima za rešavanje odabranih numeričkih problema. U ovom tekstu razmatra se upotreba matrica i daju programi za: sabiranje, oduzimanje, množenje matrica, množenje matrice skalarnom, transponovanje i inverziju matrica.

Svedoci smo sve veće primene matrica u raznim oblastima delatnosti, od lingvistike i ekonomije do matematike i kvantne mehanike. Pogodnost koje pruža rad sa matricama već u svim otkrivenim. U seriji „Računarski algoritmi“ tema o matricama je najavljena već u prvom nastavku (Metodi gradjanja), a inverzija matrica bila je potrebna u drugom nastavku (Metodi tangente). Zbog algoritma za inverziju matrica koji je ovde korišćen, tema o matricama je u neposrednoj vezi sa prethodnim nastavkom serije (Metodi eliminacije).

## Operacije sa matricama

Matrica je uveo J.J. Sylvester (1851) kao uređenu pravougaonu shemu brojeva i definisao operacije sa njima. Neka matrica A ima n vrsta (redova) i m kolona (stubaca). Element u i-toj vrsti i j-toj koloni označava se sa  $a_{ij}$  ili (u računarskim programima) A(i,j). Matrica A obično se obeležava sa

$$A = [a_{ij}]_{n \times m}$$

i kaže se da matrica A ima dimenziju n\*m. Dve matrice A i B su jednake

$$A = B$$

ako su ima svi odgovarajući elementi jednaki

$$a_{ij} = b_{ij} \quad (i=1(1)n, j=1(1)m)$$

Jedna matrica jednakost povlači n\*m jednakosti. Tako definisana jednakost matrica je refleksivna (ako je A = B onda je B = A) i tranzitivna (ako su A = B i B = C onda je A = C).

Zbir matrica A i B je matrica

$$Z = A + B$$

čiji su elementi

$$z_{ij} = a_{ij} + b_{ij} \quad (i=1(1)n, j=1(1)m)$$

Sabiranje matrica dimenzije n\*m izvodi se pomoću n\*m sabiranja. Sabiranje matrica je komutativno

$$A + B = B + A$$

zato što je sabiranje brojeva komutativno  $a+b=b+a$ . Matematičari pretpostavljaju da je sabiranje matrica i asocijativno

$$(A + B) + C = A + (B + C)$$

ali ko radi na računaru itekako dobro zna da to ne važi ni za brojeve, pa razume se ne važi ni za matrice — zbog načina predstavljanja brojeva u računaru.

Razlika matrica A i B je matrica

$$R = A - B$$

čiji su elementi

$$r_{ij} = a_{ij} - b_{ij} \quad (i=1(1)n, j=1(1)m)$$

54 računari 29 • avgust 1987.

Za sabiranje i oduzimanje matrica potrebno je da su matrice istih dimenzija (n\*m).

Definiše se i nula-matrica 0 koja ima sve elemente nule

$$0 = [0]_{n \times m}$$

i za nju važi

$$A + 0 = 0 + A = A.$$

Ako vrste matrice A (dimenzije n\*m) postanu kolone (i obrnuto), dobijena matrica B (dimenzije m\*n) je transponovana matrica A.

$$B = A^T.$$

Tada važi i  $A = B^T$ .

Proizvod matrice A dimenzije n\*m i matrice B dimenzije m\*1 je matrica P dimenzije n\*1 čiji su elementi

$$p_{ik} = \sum_{j=1}^m a_{ij} b_{jk} \quad (i=1(1)n, k=1(1)1)$$

Element matrice proizvoda P u i-toj vrsti i k-toj koloni jednak je zbiru proizvoda elemenata i-te vrste prvog činioca (matrice A) i odgovarajućih elemenata k-te kolone drugog činioca (matrice B). Za proizvod AB matrica A i B potrebno je (i dovoljno) da broj kolona matrice A bude jednak broju vrsta matrice B. Već zbog toga, ako postoji proizvod AB, ne mora da postoji proizvod BA — pa, prema tome, ne može biti govora o jednakosti AB sa BA.

Početnike iznenađuje osobina matrica da AB obično nije BA, tj. da množenje matrica nije komutativno. Evo primera

$$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix}$$

medutim

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 1 & 5 \end{bmatrix}$$

U kvantnoj mehanici nekomutativnost množenja matrica ima odlučujuću ulogu. Ako postoje proizvod matrica AB i BC, onda je (AB)C = A(BC).

tj. množenje matrica je asocijativno.

Ako postoji proizvod matrica AB, onda važi

$$(AB)^T = B^T A^T.$$

Ako je broj vrsta jednak broju kolona (n = m), matrica je kvadratna. Elementi kvadratne matrice sa jednakim indeksima (i = j) nazivaju se dijagonalni. Ako su svi dijagonalni elementi matrice jednaki nuli, onda se matrica zove dijagonalna. Dijagonalna matrica kod koje su dijagonalni elementi jednaki 1 zove se jedinična matrica I. Dakle, jedinična matrica nije matrica kod koje su svi elementi jedinice (kako to početnici obično zamišljaju) — već matrica čiji su dijagonalni elementi jedinice a svi

ostali su elementi jednaki nuli. Množenje kvadratne matrice A sa jediničnom matricom I je komutativno

$$A I = I A = A.$$

Ne definiše se deljenje matricom. To ne znači da nije bilo pokušaja zasnivanja deljenja matricom. Biće još takvih pokušaja...

Postoji u matricnom računu nešto što podseća na deljenje. Pokazalo se da je moguće uvesti inverznu matricu  $A^{-1}$  matrice A tako da je njihov proizvod jednak jediničnoj matrici I

$$A^{-1} A = A A^{-1} = I.$$

Vidi se da je matrica A komutativna sa  $A^{-1}$  (naravno ako  $A^{-1}$  postoji).

## Inverzna matrice

Izračunavanje inverzne matrice  $A^{-1}$  treba izložiti sa više detalja.

Razložimo nepoznatu matricu  $A^{-1}$  po kolonama  $X_1, X_2, \dots, X_n$

$$A^{-1} = [X_1, X_2, \dots, X_n].$$

Jediničnu matricu I dimenzije n\*n takođe razlažemo na kolone

$$I = [E_1, E_2, \dots, E_n].$$

pri čemu  $E_j$  ( $j = 1(1)n$ ) su vektori (jedno-kolone matrice) čiji je j-ti element jedinica, a ostali elementi su nule. Sada važi

$$A [X_1, X_2, \dots, X_n] = [E_1, E_2, \dots, E_n]$$

ili

$$[AX_1, AX_2, \dots, AX_n] = [E_1, E_2, \dots, E_n]$$

ili

$$A X_j = E_j \quad (j = 1(1)n).$$

Ovo jednostavno izvođenje pokazuje da se izračunavanje inverzne matrice  $A^{-1}$  dimenzije n\*n svodi na rešavanje n sistema linearnih jednačina sa po n nepoznatih. U prošlom nastavku serije „Računarski algoritmi“ (Metodi eliminacije) izložena je modifikacija najefikasnijeg direktnog metoda za rešavanje sistema linearnih jednačina — Gaussovog. Kako je matrica sistema ista u svih n sistema po n jednačina, očigledno je dovoljno samo jednom primeniti metod eliminacije na matricu sistema A. Naravno, vektor slobodnih članova  $E_j$  se menja u svakom sistemu. Ukupan broj operacija množenja kod inverzije matrice dimenzije n\*n je samo tri puta veći nego kod rešavanja jednog sistema jednačina od n nepoznatih (ne mora da bude n puta veći). To je odlučni argument protiv pogrešnog gledišta da uvek treba rešavati sistem jednačina metodom eliminacije (nikada inverzijom matrica). Ako se sistem jednačina rešava za više od 3 raznih vektora



```

10 ----- Računari 29: Dušan Slavič, Metodi matrica
20 INPUT N: DEFDBL A-N,0-: NN=N*N: DIM A(NN),B(NN),P(NN),L(N),M(N)
30 DATA 1, 1, 1, 1, 2, 3, 4, 1, 3, 6, 10, 1, 4, 10, 20
40 FOR I=1 TO NN: READ A(I): NEXT I:PRINT "zadana matrica A":GOSUB 90
50 FOR I=1 TO NN:PRINT "I":NEXT I
60 GOSUB 200: PRINT "invzerna matrica, inv(A)": GOSUB 90:PRINT "det(A) = ";D
70 L=N+1:GOSUB 160:PRINT "proizvod matrica A * inv(A)"
80 FOR I=1 TO NN:A(I)*P(I)=I:GOSUB 90:END
90 FOR I=1 TO N:FOR J=1 TO N:STEP 1:PRINT USING "#####";A(J):NEXT J:PRINT :NEXT I: RETURN
100 ----- sabiranje matrica: A(N*M) vise B(N*M) je Z(N*M)
110 L=N*M: FOR K=1 TO L: I(Z(K)/A(K)+B(K)): NEXT K
120 ----- oduzimanje matrica: A(N*M) manje B(N*M) je R(N*M)
130 L=N*M: FOR K=1 TO L: R(K)=A(K)-B(K): NEXT K
140 ----- množenje matrice skalarom: A(N*M) puta s je C(N*M)
150 L=N*M: FOR K=1 TO L: C(K)=S*A(K): NEXT K
160 ----- množenje matrica: A(N*M) puta B(M*L) je P(N*L)
170 [R=0:IK=0: FOR K=1 TO L: FOR J=1 TO N:J=J:IB=IK:RI=0:IR=IR+I:FOR I=1 TO M:IB=IB+I:RI=(J)*IB+(I)+RI:J=J+1:NEXT I:P(IR)=RI:NEXT J:IK=IK+NEXT K:RETURN
180 ----- transponovanje matrice A(N*M) u matricu B(M*N)
190 L=I:FOR I=1 TO N:K=1: FOR J=1 TO M:B(L)=A(K):L=L+1:K=K+N:NEIT: NEXT I: RETURN
200 ----- izražunavanje inverzne matrice inv(A): matrica A
210 D=I:NK=0:FOR K=1 TO N:L(K)=K:M(K)=K:KK=NK+K:B(K)=A(KK)
220 FOR J=K TO N:L(K)=N+J-N
230 FOR I=K TO N:IK=L(K):IF ABS(GK)/ABS(A(IK)) THEN B(K)=A(IK):(L(K)=I:M(K)=J
240 NEXT I:NEXT J:L(K)=K:IF B(K)=A(K):IF B(K)=A(K):
250 I=M(K):IF I>K THEN LK=N+1-N:
260 FOR J=1 TO N:JK=NK+J:IK=L(K):HK=N-(J):A(JK)=A(IK):A(IK)=HK:NEXT J
270 FOR I=1 TO N:IF I<>K THEN IK=NK+I:A(IK)=-A(IK)/BK
280 NEXT I
290 FOR I=1 TO N:L(K)=NK+I:HK=A(LK):IK=I
300 FOR J=1 TO N:IF I<>K AND J<>K THEN JK=IK-I+K:A(IK)=HK+A(JK)+A(IK)
310 IK=IK+N:NEXT J:NEXT I:J=K
320 FOR J=1 TO N:IF J<>K THEN A(JK)=A(JK)/BK
330 JK=K+N:NEXT J:D=0:G:A(KK)=I/B/GK:NK=NK+NEXT K
340 FOR K=N-1 TO 1 STEP -1:I=L(K):IF I>K THEN LK=N-N+K+1-N:
350 FOR J=1 TO N:JK=L(K):HK=A(JK):IK=KK+J:A(IK)=-A(IK)+A(JK):HK:NEXT J
360 J=K(I):IF J<>K THEN IK=N-N:
370 FOR I=1 TO N:IK=L(N):HK=A(IK):JK=IK-K+J:A(IK)=-A(JK):A(JK)=HK:NEXT I
380 NEXT K:RETURN

```

```

RUN
? 4
zadana matrica A
1.0000000000000000 1.0000000000000000 1.0000000000000000 1.0000000000000000
1.0000000000000000 2.0000000000000000 3.0000000000000000 4.0000000000000000
1.0000000000000000 3.0000000000000000 6.0000000000000000 10.0000000000000000
1.0000000000000000 4.0000000000000000 10.0000000000000000 20.0000000000000000
inverzna matrica, inv(A)
4.0000000000000000 -8.0000000000000000 -11.0000000000000000 3.0000000000000000
-8.0000000000000000 14.0000000000000000 10.0000000000000000 -3.0000000000000000
4.0000000000000000 -11.0000000000000000 10.0000000000000000 -3.0000000000000000
-1.0000000000000000 3.0000000000000000 -3.0000000000000000 1.0000000000000000
det(A) = 1
proizvod matrica A * inv(A)
1.0000000000000000 0.0000000000000000 -0.0000000000000000 0.0000000000000000
-0.0000000000000000 1.0000000000000000 -0.0000000000000000 -0.0000000000000000
0.0000000000000000 0.0000000000000000 1.0000000000000000 0.0000000000000000
-0.0000000000000000 0.0000000000000000 0.0000000000000000 1.0000000000000000

```

ra slabodnih članova uz istu matricu sistema, celishodno je koristiti inverznu matricu za rešavanje tih sistema.

Izračunavanje inverzne matrice na računaru (a kako bismo drukčije?) skopčan je sa dodatnim teškoćama koje su posledica konačnog zapisa broja u računaru (ograničena tačnost) i uzanog opsega brojeva (posebno kritično kod matrica većih dimenzija). Zato program za inverziju matrice treba da bude krajnje oprezno načinjen, što ipak ne eliminiše uticaj pomenutih ograničenosti računara. Broj operacija množenja kod inverzije matrica je srazmeran trećem stepenu reda n. Raditi inverziju matrica u

standardnoj tačnosti za veće n je izraz avanturističkog duha. Dimenzija matrice ne mora biti suviše velika pa da opseg brojeva do 10<sup>10</sup> bude premalen. U tom slučaju treba raditi na drugom računaru (ili sa drugim osnovnim softverom) koji dozvoljava širi opseg brojeva, npr. do 10<sup>20</sup>. Mogućno je i napraviti program: kritične veličine pamti komplikovanje — odvojeno mantisu od eksponenta.

### Matrica i njihove determinante

U kritične veličine spada i determinanta. Determinanta nije vrednost matrice (kako se

ponegde pogrešno tvrdi), već broj pridružen matrici. Da je d determinanta matrice A dimenzije n\*n (odnosno, reda n), piše se

$$d = \det(A) = |a_{ik}|_{n \times n}$$

Ako je matrica A reda 1, onda je  $d = a_{11}$ .  
 Za matricu A reda 2, determinanta je  $d = a_{11}a_{22} - a_{12}a_{21}$ .  
 Matrica A reda 3 ima determinantu  $d = a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{23}a_{31} - a_{12}a_{21}a_{33} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}$ .

Determinante su znatno starije od matrica. G.W.Leibniz (1676) je prvi primetio da se pri rešavanju jednačina pojavljuju ove kombinacije koeficijenata. Godine 1678 objavljive tu u radu „Uzorak nove analize kojom se uklanjaju pogreške, koja kao rukom vodi duh i kojom se lako nalaze napreci“. Kao i svi vredni naučni rezultati, i ovo otkrice determinanta, ponovo je postignuto. G. Cramer (1750) u dodatku „Uvoda u analizu algebarskih krivih“ otkriva pravila koja sada obično nose njegovo ime, a bila su poznata Leibnizu i Newtonu skoro čitav vek ranije.

Teoriju determinanta razvili su E. Bezout (1764), A.T.Vandermonde (1772), P.S.Laplace (1772), J.L.Lagrange (1773). Naziv „determinanta“ uveo je u jednom posebnom slučaju K.F.Gauss (1801), a u opštem A.L.Cauchy. Cauchy je uveo množenje determinanta (1812). C.Jacobi (1841) teoriji determinanta daje sadašnji oblik.

Ako se u determinanti  $|a_{ik}|_{n \times n}$  izostavi i-ta vrsta i k-kolona, ostatak je determinanta koja se naziva subdeterminanta i obeležava sa  $A_{ik}$ . Determinanta reda n tada se može razviti po elementima i-te vrste

$$d = \sum_{k=1}^n (-1)^{i+k} a_{ik} A_{ik}$$

ili po elementima k-te kolone

$$d = \sum_{i=1}^n (-1)^{i+k} a_{ik} A_{ik}$$

Ovi razvoji „imaju veliki teorijski značaj“. Drugim rečima, ako vas neko nagovara da programirate izračunavanje vrednosti determinante na ovaj način, treba da znate da je broj operacija množenja ogroman (raste sa faktorijelom reda n). Razvoje navodim samo zato što se preporučuju — a ne bi trebalo. Mnogo je bolje koristiti Gaussov algoritam. U priloženom programu vrednost determinante matrice se uzged izračunava. Ne treba propustiti da se ta dobijene vrednost determinante i prikaže. Ako je vrednost determinante mala ili nula, to znači da je matrica (kojoj pokušavamo da izračunamo inverznu matricu) možda singularna. U matematici su singularne (neregularne) matrice čija je determinanta jednaka nuli i one nemaju inverznu matricu, kao što nula nema svoj recipročan vrednost. U računarstvu se i neke nesingularne (regulare) matrice — zbog toga što im je determinanta reda veličine mašinske nule — ponašaju kao singularne matrice. Izlaz iz teškoće je prelazak na računar veće tačnosti i šireg opsega brojeva.

U računaru se matrica pamti kao vektor dobijen nizanjem kolona matrice. Programi su tome prilagođeni u cilju izbegavanja suvišnih operacija pri računanju indeksa.

Dušan Slavič

# Makro asembleri

*Pre nekoliko meseci smo naučili kako se pišu „obični“ dvoprolazni asembleri kojima vlasnici kućnih računara sa osmibitnim mikroprocesorima najčešće moraju da budu zadovoljni. Ovoga ćemo se puta, kao što smo obećali, baviti makro asemblerima koji, premda ih obično vezujemo za veće sisteme, poslednjih godina stiču pravo građanstva i na personalnim računarima.*

Osnovna razlika između asemblera i kompajlera je što prvi prevodi jedan red izvornog koda u jednu mašinsku naredbu (koja, jasno, može da zauzme više od jednog bajta), dok drugi prevodi svaki red u veći broj (tipično 30—50) mašinskih naredbi. Makro asembler je (kao što mu ime i govori) asembler, a ipak ne zadovoljava ovu definiciju: njegov red izvornog koda može da bude transformisan u proizvoljno veliki broj mašinskih instrukcija. Pa ipak, makro asembler se uklapa u „duh“ definicije: ova zamenja je strogo kontrolisana, što znači da programer tačno zna kakav će mašinski program nastati. Za kompajlere ovo očito ne važi: možemo, doduše, da zamišljamo kako će neka naredba biti prevedena na mašinske, ali će ova nagadanja, ukoliko nismo odličan poznavalac kompajlera, verovatno biti vrlo daleka od istine!

## Za lenje programere

Makro asembleri su nastali prilično davno i na neki način predstavljaju preteču viših programskih jezika: ljudima je dosadilo da stotinama puta pišu iste sekvence naredbi, pa su počeli da ih zamenjuju skraćenicama; skraćenicu po skraćenicu, nastao je fortran. Na većim sistemima je danas svaki asembler **obavezno** makro asembler; Digital je, čak, ukinuo reč *assembler* i zamenio je sa *MACRO*, pa su programerke koje imaju posla sa PDP-jevima i VAX-ovim asemblerom („*makroom*“) izložene raznoraznim bezazlenim šalama.

.MCDEF Zbir		
push p1	ovaj	push p1
push p2	.	push p2
push p3	----	push p3
push p4	.	push p4
.MCEND	vratil	.
.MCDEF vprat		
pull p1	.	pull p1
pull p2	.	pull p2
pull p3	.	pull p3
pull p4	.	pull p4
.MCEND		

slika 1

Slika 1 će nam pomoći da steknemo osnovni osećaj o mogućnostima makro asemblera — program, istina, nismo prevodili na pravi mašinski jezik, ali smo izvršili takozvanu makro *ekspanziju*, tj. zamenili makro pozive nizovima asemblerskih instrukcija. Program započinje makro definicijama: reč *ovaj* predstavlja zamenu za niz asemblerskih naredbi koje prenose sadržaje prvih četiri registra (hipotetičnog) procesora na stek, dok reč *vratil* ima obrnutu

dejstvo. Ma kakav program da pišete, često ćete poželeti da sačuvate sadržaje registara na steku; umesto da pišete četiri reda, biće dovoljan jedan! Osim uštede u kucanju, program smo na ovaj način učinili daleko preglednijim — kao da smo koristili potprograme (makro naredbe sa slike 1, uzgred budi rečeno, ne bismo mogli tek tako da zamenimo pozivom potprograma, jer većina mikroprocesora koristi stek za prenošenje adresa povratka). Početnici, zaista, često mešaju makro naredbe i potprograme, pa nije loše odmah ukazati na razlike između njih.

Makro naredba se, pre asembliranja, zamenjuje nizom asemblerskih instrukcija koje rezultiraju generisanjem odgovarajućeg (tipično dugačkog) mašinskog koda. Poziv potprograma se, sa druge strane, prevodi u jednu mašinsku naredbu i dolazi do izražaja tek tokom izvršavanja programa. Ukoliko, dakle, prvenstveno koristimo potprograme, rezultujući program će biti kratak i relativno spor — svako pozivanje potprograma i povratka iz njega „troše“ ipak primetan broj mašinskih ciklusa. Ukoliko se, s druge strane, orijentisemo na makro naredbe, rezultujući kod će biti dugačak (svaki makro poziv se zamenjuje čitavim nizom naredbi), ali i brz — nema utroška vremena za razne skokove i povratke. Obzirom da je na modernim računarima memorija prostrana a brzina kritična, makro naredbe predstavljaju preovlađujuću soluciju.

## Makro procesori

.MCDEF zbir	
push af	
ld a, %1	
add a, %2	
ld %3, a	
pop af	
.MCEND	

slika 2

Slika 2 prikazuje mnogo zanimljiviju promenu makro asemblera: setovi instrukcija slabijih mikroprocesora su najčešće nesimetrični, što znači da su neki registri ravnopravniji od drugih — na Z80, na primer, možemo da napišemo ADD A, B (najčešće se piše samo ADD B) ali ne i ADD C,D. Uz makro definiciju sa slike 2 možemo

da pišemo ZBIR B C D što znači da se registri B i C sabiraju i da rezultat ide u registar D. Možemo, uzgred budi rečeno, da pišemo i ZBIR B C C; **ali ne** i ZBIR B A C (ovaj ćemo problem uskoro rešiti — za sada samo pokušajte da pogodite u čemu je prepreka).

Slika 3 prikazuje način na koji će makro asembler „prevesti“ red ZBIR B C D: sadr-

zbir b c d --->	push af
	ld a, b
	add a, c
	ld d, a
	pop af

slika 3

žaj A se prenosi na stek, sadržaj B se prenosi u akumulator, njemu se dodaje sadržaj C i rezultat smešta u D; na kraju treba vratiti A sa steka. Makro asembler, dakle, poznaje fiktivne argumente: kada smo napisali ZBIR B C D, prvi fiktivni argument (obežena sa %1) je dobio vrednost 'A', drugi (%2) vrednost 'B' a treći (%3) vrednost 'C' — primetimo da fiktivni argument nema vrednost koja će se nalaziti u registru C u toku izvršavanja programa; radi se prosto o slovu C što znači da je makro procesiranje obična obrada teksta. Što se makro asemblera tiče, mogli smo da napišemo i ZBIR B (&2000) B pa bi bilo izvršeno PUSH AF: LD A, B; ADD (&2000); LD B,A; POP AF — sadržaju B se dodaje sadržaj memorijeke čelije čija je adresa &2000. Z80, na žalost, nije opremljen instrukcijom ADD (nn), što makro procesor ne zna — makro procesiranje se, reklosmo, svodi na običnu obradu teksta. Ovakav problem, međutim, ne treba da vas zabrinjava — grešku koju je propustio makro procesor će prihvatiti asembler. Nevolja bi nastala kada bi Z80 posedovao i instrukciju ADD A, &20 i ADD A, (&20) — zamislite da programer napiše ZBIR B &20 B, a ustvari želi da se sadržaju B doda sadržaj *memorijske čelije &20 a ne konstanta &20*. Ovakve greške se teško pronalaze i ispravljaju, što znači da sa makro asemblerima treba biti vrlo oprezan — ukoliko program baš nikako ne radi, možda ga nije loše pogledati u razvijenoj formi!

Šta da se radi sa instrukcijama koje podržavaju nekoliko suštinskih različitih adresnih modova ili makro definicijama koje, zavisno od argumentata, zahtevaju

izvršavanje sasvim različitih mašinskih naredbi? Makro procesor bi očito bilo zgodno dopuniti nekakvim IF-ovima, zatim nekakvim petljama, lokalnim promenljivima, promenljivim brojem argumenata (zašto da ne možemo da napišemo ZBIR B C D E?) i tako dalje. Kada se jednom počne, nije lako stati; beskonačnim proširenjima nastaju čitavi mali programski jezici koji su, na žalost, često sami sebi cilj — udžbenici sistemskog programiranja obavezno opisuju makro procesore koji su toliko složeni da

se pomoću njih mogu rešavati 'Kula Hanova', računati faktorijel i mnogo sličnih stvari i to isključivo manipulacijama sa tekstom! Nevolja sa ovakvim makro procesorima je što je njihovo korišćenje izuzetno komplikovano i što se ne isplati investirati vreme u njihovo razumevanje ako baš treba da izračunamo faktorijel, napisaćemo bezijk program! Tako moderni makro asembleri imaju mogućnosti uslovne ekspanzije (o kojoj ćemo govoriti nešto donjije), vodi se mnogo računa da se uvedu samo zaista potreb-

ne stvari i da se njihova upotreba maksimalno pojednostavi. Mi ćemo početi od samoga početka — najjednostavnijeg makro procesora.

## Zrno po zrno . . .

Naš makro procesor će se baviti isključivo preprocesiranjem — na neki ćemo mu način dostavljati tekst, a on će liferovati asemblerske instrukcije koje donjije nekako treba dostaviti asembleru. Samo se po

```

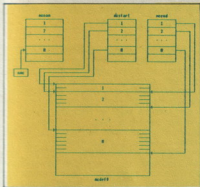
10 // *
20 // *
30 // * Osnovni makro procesor
40 // *
50 // * (C) 1987 by Dejan Ristanovic
60 // *
70 // * Rečunari 29
80 // *
90 // *
100 DIM mcname(100) OF 10
110 DIM mctstr$(100),mccnd$(100)
120 DIM mctstr$(100) OF 30
130 DIM aa OF 100
140 blankstr=""
150 mctstr$(1)=""
160 mccnd$(1)=""
170 mctstr$(1)=""
180 mctstr$(1)=""
190 sp11$(9,10)="",instr$(9)
200 WHILE instr$(1)="" AND DO
210 IF instr$(1)="" THEN
220   mctstr$(1)=""
230   mctstr$(1)=""
240   mctstr$(1)=""
250   mctstr$(1)=""
260   mctstr$(1)=""
270   sp11$(9,10)=""
280   WHILE instr$(1)="" THEN
290     IF instr$(1)="" OR instr$(1)="" THEN
300       PRINT "Label within macro"
310     ELSE
320       cptr=mctstr$(1)
330     IF instr$(1)="" THEN
340       mctstr$(1)=""
350     ELSE
360       mctstr$(1)=""
370     END IF
380   END IF
390   mctstr$(1)=""
400   sp11$(9,10)=""
410   END WHILE
420   mccnd$(1)=""
430   ELSE
440     expand$(1)=""
450   END IF
460   mctstr$(1)=""
470   sp11$(9,10)=""
480   END WHILE
490   END
500 // *
510 PROC sp11$(10,REF istr,REF instr) CLOSED
520   istr=""
530   istr=""
540   IF istr="" THEN istr=""
550   istr=""
560   istr=""
570   istr=""
580   istr=""
590   ELSE
600     istr=""
610     istr=""
620   END IF
630   istr=""
640   WHILE instr$(1)="" DO
650     istr=""
660   END WHILE
670   istr=""
680   WHILE instr$(1)="" DO
690     istr=""
700   END WHILE
710   istr=""
720   istr=""
730   istr=""
740   WHILE istr="" THEN
750     istr=""
760     istr=""
770   ELSE
780     istr=""
790     istr=""
800   END IF
810   ELSE
820     istr=""
830   END IF
840   END IF
850   END WHILE
860   END PROC sp11
870 // *
880 PROC anal(instr,REF mname,REF arg$( ),REF nargs) CLOSED
890   istr=""
900   istr=""
910   IF istr="" THEN
920     mname=""
930   ELSE
940     mname=""
950   END IF
960   WHILE istr="" DO
970     istr=""
980     istr=""
990     istr=""
1000    istr=""
1010    istr=""
1020    istr=""
1030    istr=""
1040    istr=""
1050    istr=""
1060    istr=""
1070    istr=""
1080    istr=""
1090    istr=""
1100    istr=""
1110    istr=""
1120    istr=""
1130    istr=""
1140    istr=""
1150    istr=""
1160    istr=""
1170    istr=""
1180    istr=""
1190    istr=""
1200    istr=""
1210    istr=""
1220    istr=""
1230    istr=""
1240    istr=""
1250    istr=""
1260    istr=""
1270    istr=""
1280    istr=""
1290    istr=""
1300    istr=""
1310    istr=""
1320    istr=""
1330    istr=""
1340    istr=""
1350    istr=""
1360    istr=""
1370    istr=""
1380    istr=""
1390    istr=""
1400    istr=""
1410    istr=""
1420    istr=""
1430    istr=""
1440    istr=""
1450    istr=""
1460    istr=""
1470    istr=""
1480    istr=""
1490    istr=""
1500    istr=""
1510    istr=""
1520    istr=""
1530    istr=""
1540    istr=""
1550    istr=""
1560    istr=""
1570    istr=""
1580    istr=""
1590    istr=""
1600    istr=""
1610    istr=""
1620    istr=""
1630    istr=""
1640    istr=""
1650    istr=""
1660    istr=""
1670    istr=""
1680    istr=""
1690    istr=""
1700    istr=""
1710    istr=""
1720    istr=""
1730    istr=""
1740    istr=""
1750    istr=""
1760    istr=""
1770    istr=""
1780    istr=""
1790    istr=""
1800    istr=""
1810    istr=""
1820    istr=""
1830    istr=""
1840    istr=""
1850    istr=""
1860    istr=""
1870    istr=""
1880    istr=""
1890    istr=""
1900    istr=""
1910    istr=""
1920    istr=""
1930    istr=""
1940    istr=""
1950    istr=""
1960    istr=""
1970    istr=""
1980    istr=""
1990    istr=""
2000    istr=""

```



sebi razume da bi u realnim uslovima trebalo integrisati makro procesor sa assemblerom (tako i nastaju makro assembleri), ali bi obim „Računara“ teško mogao da podnese sličnu integraciju. Zato će makro procesor koji pišemo predstavljati više ilustraciju algoritma nego program koji bi sam po sebi bio posebno koristan.

Pre nego što počnemo da pišemo program, treba da se odredimo za strukturu podataka koju ćemo koristiti, tj. da dizajniramo tablice. Potrebna nam je, pre svega, tablica imena makro naredbi koju ćemo nazvati *mcnam* i pointer na zadnji popu-



slika 4

njeni red ove tablice — *nm*. Obzirom da svaka makro naredba može da ima argumente, mogli bismo da uvedemo niz *narg* čiji bi i-te elemenat sadržao broj parametara i-te makro naredbe, ali ćemo od njega za sada odustati — broj argumenata ćemo odrediti u trenutku poziva. Potreban nam je, nazad, glavni niz *mcdef* *s* koji će sadržati same makro definicije. Pošto svaka makro definicija tipično zauzima više redova, uvešćemo nizove *mstart* i *mcend* — *mstart* (*i*) ukazuje na početak a *mcend* (*i*) na kraj i-te makrodefinicije. Slika 4 treba da vam pomogne da razumete strukturu podataka; sličnu smo metodologiju primenili kada smo, u Računaruina 10' i „Računaruina 12“, u okviru „Putovanja u središte ROM-a“, razvijali Logo interpretator.

Pošto je struktura podataka usvojena, razmislimo o algoritmu koji će program primenjivati. Učitavamo izvorni program liniju po liniju i analiziramo učitani tekst. Ignoriramo (eventualnu) labelu koja se završava dvotačkom i komentar koji počinje tačkom i zarezom i izdajvamo izvršni deo instrukcije sa adresom. Ukoliko instrukcija počinje tačkom, radi se o direktivi assembleru ili makroprocesoru, pa pokušavamo da prepoznamo reč MCDEF koja označava početak nove makrodefinicije. Ukoliko uspešno, kompletan tekst makrodefinicije (sve do reda. MCEND) prepisujemo u tabelu *mcdef* *s* i ne zaboravljamo da modifikujemo odgovarajuće pointere *mstart*, *mcend* i *nm*.

Ukoliko nismo prepoznali početak makrodefinicije, radi se o običnoj naredbi, pa vršimo njenu ekspanziju: poredimo mнемоник sa svim elementima table *mcdef* *s*, ako ga ne pronađemo, prepisujemo red na

```

.MCDEF ASEP                i pozicija bolje
.MCDEF X1 = 'A' AND (X2 = 'A') i naredbi
.MCEND
.MCDEF X1 <= 'A'
    JUMP A2
    AND A1, X1
.MCDEF X2 = 'A'
    JUMP B2
    AND B1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP C2
    AND C1, X1
.MCDEF X2 = 'A'
    JUMP D2
    AND D1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP E2
    AND E1, X1
.MCDEF X2 = 'A'
    JUMP F2
    AND F1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP G2
    AND G1, X1
.MCDEF X2 = 'A'
    JUMP H2
    AND H1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP I2
    AND I1, X1
.MCDEF X2 = 'A'
    JUMP J2
    AND J1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP K2
    AND K1, X1
.MCDEF X2 = 'A'
    JUMP L2
    AND L1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP M2
    AND M1, X1
.MCDEF X2 = 'A'
    JUMP N2
    AND N1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP O2
    AND O1, X1
.MCDEF X2 = 'A'
    JUMP P2
    AND P1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP Q2
    AND Q1, X1
.MCDEF X2 = 'A'
    JUMP R2
    AND R1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP S2
    AND S1, X1
.MCDEF X2 = 'A'
    JUMP T2
    AND T1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP U2
    AND U1, X1
.MCDEF X2 = 'A'
    JUMP V2
    AND V1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP W2
    AND W1, X1
.MCDEF X2 = 'A'
    JUMP X2
    AND X1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP Y2
    AND Y1, X1
.MCDEF X2 = 'A'
    JUMP Z2
    AND Z1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AA2
    AND AA1, X1
.MCDEF X2 = 'A'
    JUMP AB2
    AND AB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AC2
    AND AC1, X1
.MCDEF X2 = 'A'
    JUMP AD2
    AND AD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AE2
    AND AE1, X1
.MCDEF X2 = 'A'
    JUMP AF2
    AND AF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AG2
    AND AG1, X1
.MCDEF X2 = 'A'
    JUMP AH2
    AND AH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AI2
    AND AI1, X1
.MCDEF X2 = 'A'
    JUMP AJ2
    AND AJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AK2
    AND AK1, X1
.MCDEF X2 = 'A'
    JUMP AL2
    AND AL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AM2
    AND AM1, X1
.MCDEF X2 = 'A'
    JUMP AN2
    AND AN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AO2
    AND AO1, X1
.MCDEF X2 = 'A'
    JUMP AP2
    AND AP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AQ2
    AND AQ1, X1
.MCDEF X2 = 'A'
    JUMP AR2
    AND AR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AS2
    AND AS1, X1
.MCDEF X2 = 'A'
    JUMP AT2
    AND AT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AU2
    AND AU1, X1
.MCDEF X2 = 'A'
    JUMP AV2
    AND AV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AW2
    AND AW1, X1
.MCDEF X2 = 'A'
    JUMP AX2
    AND AX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP AY2
    AND AY1, X1
.MCDEF X2 = 'A'
    JUMP AZ2
    AND AZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BA2
    AND BA1, X1
.MCDEF X2 = 'A'
    JUMP BB2
    AND BB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BC2
    AND BC1, X1
.MCDEF X2 = 'A'
    JUMP BD2
    AND BD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BE2
    AND BE1, X1
.MCDEF X2 = 'A'
    JUMP BF2
    AND BF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BG2
    AND BG1, X1
.MCDEF X2 = 'A'
    JUMP BH2
    AND BH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BI2
    AND BI1, X1
.MCDEF X2 = 'A'
    JUMP BJ2
    AND BJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BK2
    AND BK1, X1
.MCDEF X2 = 'A'
    JUMP BL2
    AND BL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BM2
    AND BM1, X1
.MCDEF X2 = 'A'
    JUMP BN2
    AND BN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BO2
    AND BO1, X1
.MCDEF X2 = 'A'
    JUMP BP2
    AND BP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BQ2
    AND BQ1, X1
.MCDEF X2 = 'A'
    JUMP BR2
    AND BR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BS2
    AND BS1, X1
.MCDEF X2 = 'A'
    JUMP BT2
    AND BT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BU2
    AND BU1, X1
.MCDEF X2 = 'A'
    JUMP BV2
    AND BV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BW2
    AND BW1, X1
.MCDEF X2 = 'A'
    JUMP BX2
    AND BX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP BY2
    AND BY1, X1
.MCDEF X2 = 'A'
    JUMP BZ2
    AND BZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CA2
    AND CA1, X1
.MCDEF X2 = 'A'
    JUMP CB2
    AND CB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CC2
    AND CC1, X1
.MCDEF X2 = 'A'
    JUMP CD2
    AND CD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CE2
    AND CE1, X1
.MCDEF X2 = 'A'
    JUMP CF2
    AND CF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CG2
    AND CG1, X1
.MCDEF X2 = 'A'
    JUMP CH2
    AND CH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CI2
    AND CI1, X1
.MCDEF X2 = 'A'
    JUMP CJ2
    AND CJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CK2
    AND CK1, X1
.MCDEF X2 = 'A'
    JUMP CL2
    AND CL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CM2
    AND CM1, X1
.MCDEF X2 = 'A'
    JUMP CN2
    AND CN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CO2
    AND CO1, X1
.MCDEF X2 = 'A'
    JUMP CP2
    AND CP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CQ2
    AND CQ1, X1
.MCDEF X2 = 'A'
    JUMP CR2
    AND CR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CS2
    AND CS1, X1
.MCDEF X2 = 'A'
    JUMP CT2
    AND CT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CU2
    AND CU1, X1
.MCDEF X2 = 'A'
    JUMP CV2
    AND CV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CW2
    AND CW1, X1
.MCDEF X2 = 'A'
    JUMP CX2
    AND CX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP CY2
    AND CY1, X1
.MCDEF X2 = 'A'
    JUMP CZ2
    AND CZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DA2
    AND DA1, X1
.MCDEF X2 = 'A'
    JUMP DB2
    AND DB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DC2
    AND DC1, X1
.MCDEF X2 = 'A'
    JUMP DD2
    AND DD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DE2
    AND DE1, X1
.MCDEF X2 = 'A'
    JUMP DF2
    AND DF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DG2
    AND DG1, X1
.MCDEF X2 = 'A'
    JUMP DH2
    AND DH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DI2
    AND DI1, X1
.MCDEF X2 = 'A'
    JUMP DJ2
    AND DJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DK2
    AND DK1, X1
.MCDEF X2 = 'A'
    JUMP DL2
    AND DL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DM2
    AND DM1, X1
.MCDEF X2 = 'A'
    JUMP DN2
    AND DN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DO2
    AND DO1, X1
.MCDEF X2 = 'A'
    JUMP DP2
    AND DP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DQ2
    AND DQ1, X1
.MCDEF X2 = 'A'
    JUMP DR2
    AND DR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DS2
    AND DS1, X1
.MCDEF X2 = 'A'
    JUMP DT2
    AND DT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DU2
    AND DU1, X1
.MCDEF X2 = 'A'
    JUMP DV2
    AND DV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DW2
    AND DW1, X1
.MCDEF X2 = 'A'
    JUMP DX2
    AND DX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP DY2
    AND DY1, X1
.MCDEF X2 = 'A'
    JUMP DZ2
    AND DZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EA2
    AND EA1, X1
.MCDEF X2 = 'A'
    JUMP EB2
    AND EB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EC2
    AND EC1, X1
.MCDEF X2 = 'A'
    JUMP ED2
    AND ED1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EE2
    AND EE1, X1
.MCDEF X2 = 'A'
    JUMP EF2
    AND EF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EG2
    AND EG1, X1
.MCDEF X2 = 'A'
    JUMP EH2
    AND EH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EI2
    AND EI1, X1
.MCDEF X2 = 'A'
    JUMP EJ2
    AND EJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EK2
    AND EK1, X1
.MCDEF X2 = 'A'
    JUMP EL2
    AND EL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EM2
    AND EM1, X1
.MCDEF X2 = 'A'
    JUMP EN2
    AND EN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EO2
    AND EO1, X1
.MCDEF X2 = 'A'
    JUMP EP2
    AND EP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EQ2
    AND EQ1, X1
.MCDEF X2 = 'A'
    JUMP ER2
    AND ER1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP ES2
    AND ES1, X1
.MCDEF X2 = 'A'
    JUMP ET2
    AND ET1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EU2
    AND EU1, X1
.MCDEF X2 = 'A'
    JUMP EV2
    AND EV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EW2
    AND EW1, X1
.MCDEF X2 = 'A'
    JUMP EX2
    AND EX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP EY2
    AND EY1, X1
.MCDEF X2 = 'A'
    JUMP EZ2
    AND EZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FA2
    AND FA1, X1
.MCDEF X2 = 'A'
    JUMP FB2
    AND FB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FC2
    AND FC1, X1
.MCDEF X2 = 'A'
    JUMP FD2
    AND FD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FE2
    AND FE1, X1
.MCDEF X2 = 'A'
    JUMP FF2
    AND FF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FG2
    AND FG1, X1
.MCDEF X2 = 'A'
    JUMP FH2
    AND FH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FI2
    AND FI1, X1
.MCDEF X2 = 'A'
    JUMP FJ2
    AND FJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FK2
    AND FK1, X1
.MCDEF X2 = 'A'
    JUMP FL2
    AND FL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FM2
    AND FM1, X1
.MCDEF X2 = 'A'
    JUMP FN2
    AND FN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FO2
    AND FO1, X1
.MCDEF X2 = 'A'
    JUMP FP2
    AND FP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FQ2
    AND FQ1, X1
.MCDEF X2 = 'A'
    JUMP FR2
    AND FR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FS2
    AND FS1, X1
.MCDEF X2 = 'A'
    JUMP FT2
    AND FT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FU2
    AND FU1, X1
.MCDEF X2 = 'A'
    JUMP FV2
    AND FV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FW2
    AND FW1, X1
.MCDEF X2 = 'A'
    JUMP FX2
    AND FX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP FY2
    AND FY1, X1
.MCDEF X2 = 'A'
    JUMP FZ2
    AND FZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GA2
    AND GA1, X1
.MCDEF X2 = 'A'
    JUMP GB2
    AND GB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GC2
    AND GC1, X1
.MCDEF X2 = 'A'
    JUMP GD2
    AND GD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GE2
    AND GE1, X1
.MCDEF X2 = 'A'
    JUMP GF2
    AND GF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GG2
    AND GG1, X1
.MCDEF X2 = 'A'
    JUMP GH2
    AND GH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GI2
    AND GI1, X1
.MCDEF X2 = 'A'
    JUMP GJ2
    AND GJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GK2
    AND GK1, X1
.MCDEF X2 = 'A'
    JUMP GL2
    AND GL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GM2
    AND GM1, X1
.MCDEF X2 = 'A'
    JUMP GN2
    AND GN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GO2
    AND GO1, X1
.MCDEF X2 = 'A'
    JUMP GP2
    AND GP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GQ2
    AND GQ1, X1
.MCDEF X2 = 'A'
    JUMP GR2
    AND GR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GS2
    AND GS1, X1
.MCDEF X2 = 'A'
    JUMP GT2
    AND GT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GU2
    AND GU1, X1
.MCDEF X2 = 'A'
    JUMP GV2
    AND GV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GW2
    AND GW1, X1
.MCDEF X2 = 'A'
    JUMP GX2
    AND GX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP GY2
    AND GY1, X1
.MCDEF X2 = 'A'
    JUMP GZ2
    AND GZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HA2
    AND HA1, X1
.MCDEF X2 = 'A'
    JUMP HB2
    AND HB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HC2
    AND HC1, X1
.MCDEF X2 = 'A'
    JUMP HD2
    AND HD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HE2
    AND HE1, X1
.MCDEF X2 = 'A'
    JUMP HF2
    AND HF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HG2
    AND HG1, X1
.MCDEF X2 = 'A'
    JUMP HH2
    AND HH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HI2
    AND HI1, X1
.MCDEF X2 = 'A'
    JUMP HJ2
    AND HJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HK2
    AND HK1, X1
.MCDEF X2 = 'A'
    JUMP HL2
    AND HL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HM2
    AND HM1, X1
.MCDEF X2 = 'A'
    JUMP HN2
    AND HN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HO2
    AND HO1, X1
.MCDEF X2 = 'A'
    JUMP HP2
    AND HP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HQ2
    AND HQ1, X1
.MCDEF X2 = 'A'
    JUMP HR2
    AND HR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HS2
    AND HS1, X1
.MCDEF X2 = 'A'
    JUMP HT2
    AND HT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HU2
    AND HU1, X1
.MCDEF X2 = 'A'
    JUMP HV2
    AND HV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HW2
    AND HW1, X1
.MCDEF X2 = 'A'
    JUMP HX2
    AND HX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP HY2
    AND HY1, X1
.MCDEF X2 = 'A'
    JUMP HZ2
    AND HZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IA2
    AND IA1, X1
.MCDEF X2 = 'A'
    JUMP IB2
    AND IB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IC2
    AND IC1, X1
.MCDEF X2 = 'A'
    JUMP ID2
    AND ID1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IE2
    AND IE1, X1
.MCDEF X2 = 'A'
    JUMP IF2
    AND IF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IG2
    AND IG1, X1
.MCDEF X2 = 'A'
    JUMP IH2
    AND IH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP II2
    AND II1, X1
.MCDEF X2 = 'A'
    JUMP IJ2
    AND IJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IK2
    AND IK1, X1
.MCDEF X2 = 'A'
    JUMP IL2
    AND IL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IM2
    AND IM1, X1
.MCDEF X2 = 'A'
    JUMP IN2
    AND IN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IO2
    AND IO1, X1
.MCDEF X2 = 'A'
    JUMP IP2
    AND IP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IQ2
    AND IQ1, X1
.MCDEF X2 = 'A'
    JUMP IR2
    AND IR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IS2
    AND IS1, X1
.MCDEF X2 = 'A'
    JUMP IT2
    AND IT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IU2
    AND IU1, X1
.MCDEF X2 = 'A'
    JUMP IV2
    AND IV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IW2
    AND IW1, X1
.MCDEF X2 = 'A'
    JUMP IX2
    AND IX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP IY2
    AND IY1, X1
.MCDEF X2 = 'A'
    JUMP IZ2
    AND IZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JA2
    AND JA1, X1
.MCDEF X2 = 'A'
    JUMP JB2
    AND JB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JC2
    AND JC1, X1
.MCDEF X2 = 'A'
    JUMP JD2
    AND JD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JE2
    AND JE1, X1
.MCDEF X2 = 'A'
    JUMP JF2
    AND JF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JG2
    AND JG1, X1
.MCDEF X2 = 'A'
    JUMP JH2
    AND JH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JI2
    AND JI1, X1
.MCDEF X2 = 'A'
    JUMP JJ2
    AND JJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JK2
    AND JK1, X1
.MCDEF X2 = 'A'
    JUMP JL2
    AND JL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JM2
    AND JM1, X1
.MCDEF X2 = 'A'
    JUMP JN2
    AND JN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JO2
    AND JO1, X1
.MCDEF X2 = 'A'
    JUMP JP2
    AND JP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JQ2
    AND JQ1, X1
.MCDEF X2 = 'A'
    JUMP JR2
    AND JR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JS2
    AND JS1, X1
.MCDEF X2 = 'A'
    JUMP JT2
    AND JT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JU2
    AND JU1, X1
.MCDEF X2 = 'A'
    JUMP JV2
    AND JV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JW2
    AND JW1, X1
.MCDEF X2 = 'A'
    JUMP JX2
    AND JX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP JY2
    AND JY1, X1
.MCDEF X2 = 'A'
    JUMP JZ2
    AND JZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KA2
    AND KA1, X1
.MCDEF X2 = 'A'
    JUMP KB2
    AND KB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KC2
    AND KC1, X1
.MCDEF X2 = 'A'
    JUMP KD2
    AND KD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KE2
    AND KE1, X1
.MCDEF X2 = 'A'
    JUMP KF2
    AND KF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KG2
    AND KG1, X1
.MCDEF X2 = 'A'
    JUMP KH2
    AND KH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KI2
    AND KI1, X1
.MCDEF X2 = 'A'
    JUMP KJ2
    AND KJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KK2
    AND KK1, X1
.MCDEF X2 = 'A'
    JUMP KL2
    AND KL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KM2
    AND KM1, X1
.MCDEF X2 = 'A'
    JUMP KN2
    AND KN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KO2
    AND KO1, X1
.MCDEF X2 = 'A'
    JUMP KP2
    AND KP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KQ2
    AND KQ1, X1
.MCDEF X2 = 'A'
    JUMP KR2
    AND KR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KS2
    AND KS1, X1
.MCDEF X2 = 'A'
    JUMP KT2
    AND KT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KU2
    AND KU1, X1
.MCDEF X2 = 'A'
    JUMP KV2
    AND KV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KW2
    AND KW1, X1
.MCDEF X2 = 'A'
    JUMP KX2
    AND KX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP KY2
    AND KY1, X1
.MCDEF X2 = 'A'
    JUMP KZ2
    AND KZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LA2
    AND LA1, X1
.MCDEF X2 = 'A'
    JUMP LB2
    AND LB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LC2
    AND LC1, X1
.MCDEF X2 = 'A'
    JUMP LD2
    AND LD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LE2
    AND LE1, X1
.MCDEF X2 = 'A'
    JUMP LF2
    AND LF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LG2
    AND LG1, X1
.MCDEF X2 = 'A'
    JUMP LH2
    AND LH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LI2
    AND LI1, X1
.MCDEF X2 = 'A'
    JUMP LJ2
    AND LJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LK2
    AND LK1, X1
.MCDEF X2 = 'A'
    JUMP LL2
    AND LL1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LM2
    AND LM1, X1
.MCDEF X2 = 'A'
    JUMP LN2
    AND LN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LO2
    AND LO1, X1
.MCDEF X2 = 'A'
    JUMP LP2
    AND LP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LQ2
    AND LQ1, X1
.MCDEF X2 = 'A'
    JUMP LR2
    AND LR1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LS2
    AND LS1, X1
.MCDEF X2 = 'A'
    JUMP LT2
    AND LT1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LU2
    AND LU1, X1
.MCDEF X2 = 'A'
    JUMP LV2
    AND LV1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LW2
    AND LW1, X1
.MCDEF X2 = 'A'
    JUMP LX2
    AND LX1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP LY2
    AND LY1, X1
.MCDEF X2 = 'A'
    JUMP LZ2
    AND LZ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MA2
    AND MA1, X1
.MCDEF X2 = 'A'
    JUMP MB2
    AND MB1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MC2
    AND MC1, X1
.MCDEF X2 = 'A'
    JUMP MD2
    AND MD1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP ME2
    AND ME1, X1
.MCDEF X2 = 'A'
    JUMP MF2
    AND MF1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MG2
    AND MG1, X1
.MCDEF X2 = 'A'
    JUMP MH2
    AND MH1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MI2
    AND MI1, X1
.MCDEF X2 = 'A'
    JUMP MJ2
    AND MJ1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MK2
    AND MK1, X1
.MCDEF X2 = 'A'
    JUMP ML2
    AND ML1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MM2
    AND MM1, X1
.MCDEF X2 = 'A'
    JUMP MN2
    AND MN1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MO2
    AND MO1, X1
.MCDEF X2 = 'A'
    JUMP MP2
    AND MP1, X2
.MCDEF X1 <= 'A' AND (X2 <= 'A')
    JUMP MQ2
    AND MQ1, X1
.MCDEF X2 = 'A'
    JUMP MR2
    AND MR1, X2
.M
```

```

1880 DATA ".mdef swap"
1890 DATA "ld temp,r1"
1900 DATA "ld r1,r2"
1910 DATA "ld r2,temp"
1920 DATA ".mdef sort1"
1930 DATA ".mdef sort2"
1940 DATA "cmp r1,r2"
1950 DATA "jmp le,r3"
1960 DATA "swap r1 r2"
1970 DATA "j31 nop"
1980 DATA ".mdef sort2"
1990 DATA ".mdef sort2"
2000 DATA "sort1 r1 r2 r3"
2010 DATA "swap r1 r2"
2020 DATA ".mdef sort2"
2030 DATA "start1"
2040 DATA "sort1 r1 r2 dummy1"
2050 DATA ""
2060 DATA "sort2 r3 r4 dummy2"
2070 DATA ".end"
run
start1:
    cmp r1,r2
    jmp le,dummy1
    ld temp,r1
    ld r1,r2
    ld r2,temp
    ld r3,r4
    nop
dummy1:
    nop
    cmp r3,r4
    jmp le,dummy2
    ld temp,r3
    ld r3,r4
    ld r4,temp
    nop
dummy2:
    ld temp,r3
    ld r3,r4
    ld r4,temp
    .end
    
```

slika 10

### Makro procesor na delu

Slika 10 prikazuje naš makro procesor na delu — sastavili smo program koji, uz primenu dve makro naredbe, sortira sadržaje registara r1 i r2 u rastući, a registara r3 i r4 u opadajući poredak. Obe glavne makro definicije se zasnivaju na makro naredbi swap koja razmenjuje sadržaje bilo koje dva objekta (registri, memorijske ćelije...); druga makro definicija, uz to, poziva prvu, što je dobar test za primenjene rekurzije.

Slika 10 istovremeno prikazuje opasnosti kojih se treba čuvati pri radu sa makro assemblerima — pogled na razvijeni program pokazuje da je definicija sort2 daieko od racionalnosti, dok se pogledom na nju to ne mora uvek zapaziti.

Ostalo je još da primetimo da ovakve makro definicije ne bi bile moguće bez lokalnih labela, čija smo imena (dummy 1 odnosno dummy2) prosledili prilikom poziva.

Realizacija •MCIF naredbi nije naročito komplikovana ali je relativno obimna — treba, ako ništa drugo, napisati program koji izračunava vrednost logičkih izraza. Obzirom da nam obim „Računara“ ne dopušta da makro assembler razvijamo do detalja, zadovoljićemo se da pomenimo neke probleme uslovne makro ekspanzije i da vam njenu realizaciju ostavimo kao „domaći zadatak“.

Makro definicije su, verujte nam na reč, prilično besmislene bez mogućnosti da se u okviru njih programiraju skokovi — iako strukturiran programiranje negira potrebu za GOTO naredbama, ni jedan se mašinski jezik ne može zamisliti bez njih! Ukoliko bi skokovi bili upravljani na neko fiksno mesto u programu, sve bi bilo u redu. Probleme, međutim, ilustruje slika 7.

Vidimo da se u razvijenom programu pojavila dvostruka definicija iste labela, što nam assembler nikako neće oprostiti. Greška je, kada malo bolje razmislimo, sasvim prirodna — makro asembliiranje predstavlja obradu teksta, što znači da se fiziklno prepisivanjem makro definicije ponavljaju i labela definisane u njoj. Možete li da smislite kako da se ovaj problem prevaziđe? Postoje bar dva načina od kojih jedan, na račun saradnje korisnika, ukida bilo kakvu potrebu za dodatnim programiranjem!

### Lokalne labela

Slika 8 prikazuje trik koji treba da primenite kada god makro assembler koji koristite ne podržava lokalne labela ili (što je mnogo češći slučaj) kada ne znate kako se te labela koriste. Jedan od argumenata makro

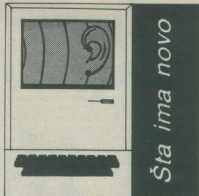
definicije je i labela što znači da ćemo sami pri svakom pozivu snabdeivati makro procesor imenom labela koju treba da koristi — ako dva puta upotrebimo isto ime, sami smo krivili! Samo se po sebi razume da pri izmišljanju imena ne moramo da budemo naročito originalni — obično se koristi ime makro definicije i broj 1, 2... Tako smo postupili i na slici 8.

Pravo rešenje problema su lokalne labela koje prikazuje slika 9. Obeležavaju se, na primer, sa %L1, %L2 itd, a makro procesor svaku od njih, u toku ekspanzije, zamenjuje jedinstvenom oznakom u kojoj figuriraju redni broj makro definicije i redni broj njenog poziva — na taj je način svaka labela garantovano jedinstvena!

Stigli smo do kraja ovog teorijskog opisa assemblera. Time, međutim, nismo rekli sve o ovim neophodnim alatima Pravih Programera — pripremili smo jedan primer takozvanih „humanih assemblera“ (pojam humanih assemblera smo opisali u „Računarima 24“) i detaljniji opis jednog konkretnog komercijalno raspoloživog makro assemblera — opredelili smo se za Micro-softov MASM 4.0 za IBM PC. Ovi će napisi, ukoliko prostor dopusti, biti objavljeni već u sledećim „Računarima“.

Što se „Putu u središte ROM-a“ tiče, uzdećemo par meseci odmora da bismo se ozbiljnije pozabavili ostalim sistemskim softverom — na redu su simulatori i, na kraju, kompajleri.

Dejan Ristanović

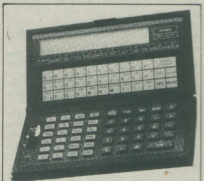


### Računarski Leonardo

Nova softverska firma Creative Logic (Uxbridge, Middlesex) zagolicala je radoznalost kompjuterske javnosti paketom Leonardo koji će se pojaviti u prodaji početkom jula.

Leonardo je program za IBM XT kompatibilne računare sa najmanje 512 kilobajta memorije koji rešava probleme. Kakve probleme? Bilo kakve!

Paket Leonardo treba da reši sve probleme koje biste mogli da mu postavite počevši od proračuna grejanja i vodovoda do društvenih analiza i planiranja izbornih kampanja. Kako je to moguće? Svi koji žele da saznaju moraću da plate 149, 695 ili 2995 funti, zavisno od verzije programa koju žele da nabave. Predlažemo da nabavite verziju od 149 funti i postavite joj sledeći problem: *kako bez previše para doći do najjače verzije Leonarda?* Odgovor će svakako biti: *raspijate se kod pirata!*



### Portabli za 100 funti

Ser Klajv Sinkler nije jedini ko se setio da proizvede portabli računare — Tandy je upravo lansirao model PC-6 koji prikazuje-mo na slici.

Za svega 100 funti dobijate računar sa 8 K RAM-a čija se radna memorija proširuje do 32 K dodavanjem posebnih modula (modul od 8 K košta 17 funti). Ekran je osamdesetokolonski, tastatura senzorsko mehanička, a litijumske baterije obezbeđuju fantastičnih 170 časova neprekidnog rada.

Tandy je istovremeno najavio mini 80-kolonski štampač koji košta 70 funti i 3.5 inčni disk koji može da se priključi i na stare modele 100, 102, 200. Disk košta 200 funti — duplo više od računara!

## Usijani, džojstik

**Dragan Brčanek** iz Pančeva, koji već postaje zvezda ove rubrike, još jednom je podstakao solidarnost igrača i čitalaca *Računara*. U pretpriklom broju raspravio se za cilj igre *Asterix & M.C.*; ja nisam znao da mu odgovorim, ali znao je Novobeogradanin **Ivan Kapičić**, koji piše: „Nije zadatak osloboditi druida (Aspiriniksa) iz rimskog logora (pa se, je li, on tamo i ne pojavljuje), nego sakupiti jedan delova kotla za čarobni napitak koji je Obeliks novomozno razbio. Ja sam dosad pronašao četiri komada (prvi na izlazu iz galskog sela, drugi kad vas legionar iz rimskog logora spровede u arenu gde vas čeka nimalo umiljan gladijator koga naravno pobeđujete, treći na ulasku u Rim i četvrti u čeliji broj šest u rimskom zatvoru, u koji dospevate tako što mirujete kad vas legionar napadne), i mislim da igra nije teška. Oni koji ne dele moje mišljenje lako će do cilja doći uz pomoć gore navedenog pouka (**unesenog u rubriku (Poukova mreža. V. St.)** Uzgred, zna li neko šifru za igru *Police Academy*?”

Sad bi bilo više nego lepo da upravo Dragan odgovori Ivanu; no ako se tu i ne uspostavi simetrija, odgovoriće neko drugi. U to sam siguran.

Još o *Asterixu*. Ivanovi poukovi za tu igru nisu jedini, ali su stigli prvi, tako da identični poukovi drugih čitalaca nisu objavljeni. Među tim drugim pošiljaocima poukova jeste i **Rusmir Arslanagić** iz Sarajeva; u on ih ispisuje na kraju svog prikaza, teksta veoma lepo napisanog... all nepotpunog. I sam Rusmir veli: „Ja nisam uspio sastaviti sve dijelove kotla. Sakupio sam ih 4.“ Pošto naša rubrika *Gotovo je, gotovo* zahteva opis igre pređene do kraja, Rusmirov tekst nismo mogli da upotrebimo. Uzdam se u to da će on naći i preostala tri komada (čovek sa lavljim prerezomno jamačino ima i lavlje srce), i da će nam onda poslati izveštaj o tome; tada će stvar biti i objavljena. Ako ga neko ne pretekne, razume se.

(Mada... čujem neke ozbiljne glasove koji sumnjaju u sedmi komad kotla. Ljudi koji su pronašli šest komada — i obili nego jureći od galskog sela do Rima i natrag — nikako ne mogu da nadu sedmi. Je li ga programer zaboravio? Ili previše dobro sakrio? Ili je negde bag? Ne znam.)

**Ivan Marović**, građanin Beograda, povrh poukova poslao je i jedno pitanje: „Da li za rubriku *EVERGREEN/NEVERGREEN* priloge pišu čitaoci ili redakcija?“

Čitaoci, Ivane. Njihovi prilozii, uostalom, ulaze i u druge rubrike: *Poukova mreža, LOAD „ja“, Gotovo je, gotovo!* i *Hvala, Arijadna*, a i mimo stalnih rubrika, ako su zanimljivi.

Drugim rečima: izvoli.

Još jednom se javlja **Miodrag Bugarčić** iz Plandišta, za koga sada vidim da je otičlan čučenik i da je za sobom ostavio peti razred (na čemu mu, naravno, čestitam). U pismu završenom u 15.30 (ništa ne izmišljaj, samo citiram), on traži više prostora za mape igara, a manje za reklame, a pita postoji li još koji nivo u igri *Rambo* (posebno, šta u prvom nivou znači ono golemo slovo H) i šta je cilj igre *Turbo*.

Što se tiče mape, saglasan sam 100%, ako ne i nešto više. One su zaista ovom bloku *Računara* potrebne kao hleb; neke nam čitaoci šalju sami, a neke naručuje mo... ali nikad ih nema u izobilju: teško ih je praviti. U ovom broju ipak ih ima nešto više nego obično.

Za reklame nisam nadležan, ali se usudujem da kažem: nepodnošne su kao izvor prihoda, naročito po ovoj skupoci. Da nema njih, *Računari* bi morali biti skuplji, a to ne bi radovalo čitaoce (pa, samim tim, ni redakciju). Možda je najbolje slegnuti ramenima i prihvatiti reklame kao neku vrstu više sile.

*Rambo* me odbija, a *Turbo* nisam ni video; odgovora nemam. Ali ništa za to: naći će se već neko ko ima. Kogod je dosad nešto pitao u *Usijanom džojstiku*, pre ili posle je dobio odgovor.

Mali vapa: kad šaljete poukove, nastojte da budu otvoreni na pisačoj mašini (ili printeru, ko ima). Ako baš moraju biti pisani rukom, onda neka budu čitki. Dok prekucavam ove poslednje, obilva me leden znoj: šta je nula a šta slovo O? Imam li pred sobom sedmici ili jedinicu? Je li ono što smatram osmicom zaista osmica ili slovo B? Jesu li neki razmaci namerni ili nehohotni? I tako dalje. U košmarima vidim legije igrača koji ređaju čitavu familiju, do u sedmo i deveto koleno, i meni i pošiljaocu poukova.

## Mirisi iz kuhinje

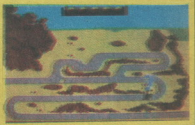
### TICA TRKAČICA

Još jedan crtač u kompjuteru. „Tica trkačica i onaj baksuz od kojoga postali su igra. U ulozii rečene ptice nastupate vi — i, naravno, jurcate kao bez duše ne biste li spasili svoj stalno ugroženi život. Trčanje, kao što znamo, guta energiju, pa morate usput zobati zrnevlje (i paziti da pri tom ne pozobljete i sačmu).

Kao i na filmu, brišete drumom, uzanim puteljkom kroz kanjone, i još užim mostićima preko provalija. (Uza sve to, ovide morate i kroz lavirinte — kojih na filmu nema). Kojot će smišljati svakoake trikove i nameštati zamke, a vaše je da

ga prozrete i njegove smicalice upotrebite protiv njega samog. Ako budete spretni, njega će gaziti kamioni, zatrpavati kamenje, itd. Ako ne budete, on će vas pojesti za ručak.

U čemu ima i neke pravde — jer u originalnoj verziji, na filmu, večno je ostajao izigran, izgnjavljen i gladan.



uređuje: Vlada Stojiljkovic

## LOAD „ja“ Odiseja 2020.

Lozinke jednog naučnika padaju u ruke bandi manijakalnih zlikovaca koji, po cenu svog života, nameravaju da unište čovečanstvo. Međutim, u igru ulaze i sile dobra...

Igra je naučno-fantastična i ima pet nivoa. Glavni lik nastao je u dalekom svemiru i igrom slučajja zatekao se na našoj planeti u trenutku kad je to bilo najpoptrebnije. Njemu, kao i nama, nije stalo do uništenja sveta.

U igri imate jedan život, jedan pokušaj. Toliko za početak.

**Prvi nivo:** uzbuza na kopnu: šifra: 1941. Banda je preuzela kontrolu nad svim tenkovi, šalje ih na sve strane, praveći opštu zbrku. Vaš je zadatak: udesiti da se tenkovi zaglibe. Na raspolaganju imate laser koji topi zemlju. Kad zauzavate 189 tenkova, banda prelazi na plan dva. Uz pomoć šifre prelazite na sledeći nivo.

**Drugi nivo:** uzbuza na moru. Šifra: Titanik. Sada su vaša veća plovila (dijalski dirigovana) u rukama bande. Od vas se traži da razbijate signale koje šalje centralni kompjuter. Oručje su vam ultrajubičasti zraci koje sami stvarate.

**Treći nivo:** Uzbuza u vazduhu. Šifra: likar. U ovom delu u igre naoružani ste samo svojim refleksima. Kad zasvetli neki broj, imate na raspolaganju samo pola sekunde da reagujete, naime da otkucate taj isti broj na tastaturi.

**Četvrti nivo:** kosmičke zavrzlave. Šifra: Vostok. Banditi čekaju satelite da se sudare. Vaše je da u pomoć svog broda sprečite te sudare. Satelita ima ravno 100. Pazite da se ne sudare dva satelita različita po boji. Kao i ranije, pri prelasku iz nivoa u nivo, ukucavate šifru.

**Peti nivo:** banda ulaže poslednje napore postavljajući tempiranu bombu u Palati mira u oazi Taka, gde kroz dan-dva treba da počnu mirovni pregovori. Treba da pronađete bombu, odnesete je u bazu bandita i njihovom legijom njihovom bombom dignete u vazduh.

Na kraju igre, čitate poslednje izdanje novina dok letite natrag ka svojoj planeti. Na naslovnoj strani piše: „PREGLED VESTI iz SVETA. U poslednjih nekoliko dana pri-mećene su čudne pojave...“

Vladimir Stakić





Aleksandra Pantica

## EVERGREEN/NEVERGREEN

## Evergreen

## 180

Pikadoooo! I to sa onim ostrim strelicama, nema šale! Doduše, „spektrum“ se malo muči da otpeva onih „a hundred and eighty“, ali nije loše. Ovo je sjajna igra iz prostog razloga što sam pogodio trostruku dvadeseticu k'o od šale, a u stvarnosti pogadam samo slučajne prolaznike.

Mada ne pijem pivo, sjajno mi prija kad jedan od pet-šest mogućih protivnika gada 3X20, a pogodi „single 1“. A ni šankerica nije loša.

## Nevergreen

## TOP GUN

Ja sam, inače, lud za simulacijama letenja, ali se nikako ne slažem sa prodajom imena. Firms „Oušn“ u zenicama su zasjale funte, i mi dobismo igru Top Gun. Naravno, tu igrač izigrava mladog Mr Cruise-a koji je potpuno sam u beznađnoj situaciji, jer ga napadaju desetine MIG-ova, a što je još gore, napolju je mrak, ne vidi se nič (da i ne govorimo o tome kako ga pretpostavljeni mrze, pa mu uvek uvala „fantom koji nema ni najosnovnije instrumente“). Ali ništa to ne smeta da vidimo sjajne lupinge tipa „baš me briga za fiziku“ slične.

## Randomize Game

## Buduća prošlost

Šta radite sa starim igrama? Jesu li vam u životu Tanks, Raiders, Centipede, Asteroids, Hungry Horace, Arcadia, Spectres, Calaxians? Ili ste ih bacili? Poklonili? Prebricali?

Ako jeste, mislim da ste pogrešili. Pomislite o tome koliko će vam one imati sentimentalnu vrednost kroz, recimo, deset godina. Otrpilike ovakvu kakvu imaju fotografije, suvenirni i spomenari; siguran sam u to. Igraćete ih i uzdišati.

Biće dobre i kao simboli statusa, slično ostalim antikviteta.

I najzad, imaće lepu tržišnu vrednost. One marke koje je vaš deda kupovao za bogatelu i lepio na pisma ne razmišljajući o njima, danas se plaćaju svim zlatom. Jedna predratna ploča (78 obrtaja u sekundi) skuplja je od dvadeset novih. I tako dalje.

Pa vi sad vidite.

## Odakle mi veštica?

Ne naročito inventivan, ali svakako efikasan što jeste: pozvati stručnjaka u časopis da komentariše igre iz svoje oblasti. Razni to rade; naročito mi pada u oči Commodore User, u kome simulacije letenja — i uopšte igre sa letelica — komentariše pravi iskusni pilot, kapetan aviona.

Uraditi to isto, ovde? Lako bih našao pilota, pa i razne sportiste za sportske simulacije, ne bi bio prevelik problem ni sa stručnjakom za kosmičke letove ili kriminologom...

... ali šta dalje? Gde da nađem nekog merodavnog za igre kao što su Warlock, Witch's Cauldron, Druid, Nosferatu?

Odstupao sam.

## Vanmarsovci

— Znam tačno kako će to izgledati — reče mi prijatelj. — Spustiće se

kosmonauti na Proksimu Kentaura, ili na Aldebaran; svejedno. I pronaći će male zelene kako igraju kompjuterske igre. A igre će se zvati Osvajaci sa zemlje, i Rat Trećoj planeti i Zemljani dolaze, i sve u tom stilu: pucačke igre u kojima treba slistiti što više nas. Na ekranima će ti mali zeleni tamaniti ljude kao muve.

— Ne brini — rekoh. — U pucačkim igrama na kraju uvek gubi igrač.

## Mala frustracija

Jeste li čuli? Izvesni John Keats (Dton Kits) napravio je prototip hardverskog dodatka pod radnim naslovom **Voystick** (reč u kojoj su pomešani **voice** i **joystick**; naprava, dakle, funkcioniše kao džojstik, ali reaguje na verbalne komande. Navodno je mala, jeftina i gotovo nepokvarljiva.) Neće više biti muka sa džojstikom: cimanja, čupanja, kvarenja. Samo ćemo govoriti: „Levo!“, „Skačil“, „Pucaj!“ itd. — razume se, na engleskom.

Ili bi bar tako bilo, da Kits nije ispaoo naivan. Ponudio je svoj izum firmi koja već proizvodi džojstike; firma je otkupila sva prava, a onda nacrt zaključala u sef. Džojstici su im kudikamo isplativiji: em su skuplji, em se često kvare (tj. prodaju se u većem broju). Profit je nađaćao nauku.

## Ne ponovilo se

Stiglo je leto, stiglo je vreme za odmor. Letujte, uživajte, nemojte da vam se desi ono što je snašlo kuma jedne moje poznanice.

Dotični kum, ozbiljan čovek i otac porodice, toliko se oduševio kompjutorom (kupljenim za sinove) da je čitav godišnji odmor proveo u Beogradu uz **Impossible Mission** i **Boulder Dash**. Ženu i decu je otprovao na more, da mu ne smetaju.

To je bilo prošle godine. Strah me je da pitam poznanicu šta kum planira za ovo leto.

## Džojstikom, ognjem i mačem

Gledajući igre u kojima je cilj uništiti Moskvu, ili poraziti sovjetske snage, ili likvidirati sovjetskog ambasadora, mislio sam da je to ipak prolazni napad paranoje: svako čudo za tri dana.

Ali ne; bolest se nastavlja. Pojavio se **Hacker II**: sovjetski naučnik ima planove oružja za uništenje sveta. Pojavio se **Gunship**: borite se protiv jedinica Varšavskog pakta. Pojavio se **SDI**: Sovjeti prede da nuklearnim projektilima zbrišu SATE. U njima se mahom traži da „crvene“ uništavaju na sat i na hvat. Uvek su oni najveća opasnost za čovečanstvo, i uvek su Amerikanci jedini braniloci slobode i mira. Poruka je jasna da ne može biti jasnija; nema nikakve dvojbe oko toga šta je pisac htio da kaže. U tu poruku igrač može i da ne poveruje odmah, ali... jedna takva igra, pa druga, pa treća — i lako će se razviti uslovljeni refleks kakav odgovara programerima igara i programerima programa.

Dok traje to ispiranje mozga, jedni govore: „Pa šta? To je samo igra“, a drugi: „Nema veze, gledaj kako je to sjajno urađeno“.

„Samo igra“? Šta onda u igri traži politika? Ima li politike u kikerima, lestižu, jambu, preferansu, dominama, biljaru? Postoji li šah u kojima su bele figure obeležene Davidovom zvezdom, a crne polumesec (ili obratno)?

„Sjajno urađeno“? Ne spirim, ali ta vrsta kvaliteta nije dovoljna. Pokušajmo da zamislimo igru sa savršenom grafikom, besprekornom animacijom, vanredno impresivnom muzikom i zvučnim efektima, apsolutno glatkim skrolovanjem i vrhunskim scenariom u kojoj bi cilj bio pobiti što više Jugoslovena. Šta bismo rekli o njoj?

Uostalom, ljudima koji od drveća ne vide šumu već je odgovorio neko umnjni od mene: Đurđor Orvel. U eseju o Salvadoru Daliju napisao je i ovo: „Prvo što tražimo od zida jeste da stoji. Ako stoji, onda je dobar zid, i pitanje kojoj svrsi služi sasvim je odvojivo od toga. Pa ipak, i najbolji zid na svetu valja srušiti ako okružuje koncentracijski logor.“

A možda grešim. Možda je baš dobro da bude što više igara čija je svrha da probude skotu u nama. Možda će biti još bolje ako se ne zadržimo samo na ovoj temi. Teren je još neistražen, mogućnosti su velike. Već vidim nove igre te vrste. Recimo: **Arbeit Macht Frei**, u kojoj ste komandant Aušvica i morate pobiti što više Jevreja koji beže od gasne komore (muškarci donose 100 poena, žene 50, a deca 25). Ili **Hijack Hero**, gde morate obići 16 zemalja, u svakoj uzeti po jednog taoca (ovde bi deca donosila najviše poena) i dići u vazduh po jednu robnu kuću da biste na kraju pobešli oteim avionom. Ili **Crusader**, u kojoj na početnom meniju birate religiju koju ćete zastupati, a potom krećete u krstaški rat da istrebite sve inoverce na planeti Zemlji. I tako dalje. Sve je prilika da bi takve igre oduševile publiku. Naravno, pod uslovom da budu sjajno urađene.

Vlada Stojiljković

# WHERE IN THE WORLD IS CARMEN SANDIEGO?

Kada se igra zove, u nešto slobodnijem prevodu. „Gde se po svetu skriva Karmen Sandiego“, razumljivo je da se odmah postavlja pitanje ko je uopšte ta cjenjena dama.

Najkraće rečeno, to je šef verovatno dosad na svetu nevidene bande lopuža koja žari i pali po svim kontinentima redom.

A šta krađu — to je posebna priča.

Mona Liza, Papin prsten, Kruna Ivana Groznog... samo su delić koji ilustruje kakvim se sve mračnim poslovima ta banda prevećanaca bavi.

Naravno da im ineko mora stati na put, a to ste vi, koji kao detektiv Interpola imate zadatak da celu bandu strpate u cizme — no na nevolju, poneka od ptičica, iako smeštene iza rešetaka, zna da i odleprša, a onda: Jovo nanovo.

Posebno je teškoća što imate posla sa izuzetno prefinjenim i obrazovanim lopovima, tako da vam samo solidno poznavanje istorije, geografije, biologije (i mnogo čega još) može pomoći da stvar isterate do kraja, i rešite ukupno 14 slučajeva, kada će se konačno i neuhvatljiva Karmen Sandiego naći tamo gde joj je i mesto.

Ali, počnimo od početka. Startujete u gradu u kome je izvršen zločin, tj. krađa.

Pošto vas je Interpol obavestio da imate šest dana da rešite zamršen slučaj, na raspolaganje su vam četiri opcije.

Najvažnija je, svakako, **Investigate** ili istraživanje. Gde god ste se zatekli, imate tri prilike da se raspitate da li je počinilac na tome mestu viden. Pri tom i skupljate podatke kako biste ga identifikovali, i potom, u ispostavi Interpola, uzeli poternicu bez koje je lopova nemoguće uhapsiti.

Opcija **Visit Interpol** (ili „Poseti Interpol“) omogućava vam da upoređujete podatke kao što su pol, boja kose, hobi, posebna obeležja, i tip automobila koji lopov vodi, eliminacijom dođete do imena počiniova — jer banda se sastoji od deset lopova, što ženskih što muških. Budite pažljivi, jer je zenskih lako pogrešiti.

Na redu je opcija **See connections** ili „Vidi veze“. Omogućava vam, bez gubljenja vremena, da, prema podacima koje ste prikupili u istraživanju otkrivenje u koji je grad lopov umakao.

I potom na aerodrom, pa avionom za njim.

Ukoliko ste na pravom tragu, kompjuter će vam to javiti; ukoliko niste, i traženu osobu niko nije video, odmah u avion i natrag, da biste potražili pravo rešenje.

Igru počinjete kao  **Rookie** ili „Regrut“, a potom (zavisno od uspeha) dobijate redom činove, da bi vaše ime, pod uslovom da uhvatite Karmen Sandiego, bilo zapisano zlatnim slovima u sve hronike Interpola, a svet našlčito bio osloboden jedne grдне napasti.

Uživajte u ovoj izuzetnoj lepoj igri, koja od vas ne zahteva nikakvo grčevito mučenje džojstika, a pritom vam pruža priliku da mnogo toga korisnog naučite.

Slobodan Perović

Rubrike **Gotovo je, gotovo!** i **Sveže učitano** (nove igre) nastaju u saradnji sa emisijom Radio-Beograda CIP i SEDAM JARICA, koja je na programu subotom u 14.02 (1 program)

## Gotovo je, gotovo!

### CURSE OF SHERWOOD

U ovoj igri morate skinuti prokletstvo bačeno na jedan zabačeni deo Šervudske bume, utoliko pre što šumski građani i ožvešli mrtvaci, pod dejstvom vradžbina, nasmrču na vas. Sve se to dešava u tri dimenzije i na 43 ekrana.

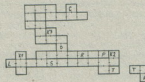
Imate četiri života, možete nositi najviše tri predmeta, za vaše avanture određena je polovina ekrana.

Vaši su protivnici: GRMALJ, koji vas gadi kamenjem, i koga likvidirate tek trećim pogotkom; PTICE, koje često menjaju smer i lete u jatima po četiri; KOSTURI, koji su brzi (na njih je najbolje ići batinom); STRELCI, koji se pojavljuju u trojkama i gađaju vas nemilice; VUKODLAK, koga možete ubiti jedino srebrnim bodežom (za sobom ostavlja lobanju); VATRA, od koje vas čuva čarobni napitak (ako ga imate); VILENJACI, čije oružje ostaje vama kad ih ukodate; MUNJE, okrugle i neumitne; i AŽDAJA, koga, poput GRMALJA, morate pogoditi triput.

Na drugoj strani, u kućama, žive: VEŠTICA (daje vam napitak ako joj date lobanju i ogledalo), BOGATAŠ (gada vas strelama, ima ogledalo koje vam ostaje kad ga ubijete), i STARAC, koji će vam prodati plan močvare.

Od startne pozicije krenite desno. U četvrtom ekranu, kad ubijete vilenjaka, uzmite **club** (batinu). Dva ekrana potom, nađi ćete na BOGATAŠEVU kuću. Uđite, ubijte ga, uzmite ogledalo. Onda natrag na startnu poziciju, pa tri ekrana levo: tamo je VEŠTICINA kuća. Uđite, dajte joj ogledalo. Izadite, idite do prvog prolaza naviše. Prvi protivnik na koga naiđete biće VILENJAK, koji za sobom ostavlja **ice wand** (leden i čarobni štapić). Zadržite batinu. Sledeća slika ima prolaz naviše. Prođite kroz njega s leve strane. GRMALJ sada postaje teži protivnik. Za sobom ostavlja ključ. Kad jednog prođete kroz kapiju, taj vam ključ više neće koristiti. KOSTURI su dve sobe

Legend:  
 0 = start  
 1 = boat  
 2 = village  
 3 = shop  
 4 = mine  
 5 = treasure  
 6 = shop  
 7 = cave  
 8 = cave  
 9 = cave  
 10 = cave



levo od kapije. Kad ih uništite, vratite se na startnu poziciju, idite desno do prvog prolaza naniže, pa opet desno. Stanite na proplanak okružen glijivama, u sam centar, i pojaviteće se na „drugom spratu“. Idite do vrata, razbijte ih batinom, krenite desno pa dole. Ubijte vilenjaka, uzmite krst i srebrni bodež koji on ostavlja za sobom. Još jednom se vratite na startnu poziciju, idite četiri ekrana levo, ubijte vukodlaka bodežom, uzmite lobanju i odnesite je veštici. Ona će vam dati napitak. Izdite iz njene kuće, idite desno do prolaza koji vodi naviše. Uzмите **ice wand**, idite desno do reke, i zaledite je pomoću ledene kuglice. Osvetljenu reku predite i idite prema vredi novca. Vratite se do prolaza koji vodi naviše. Idite do starčeve kuće; za novac ćete kupiti mapu, koja se poziva prilikom na laster 1 (pezite, ispravan je samo njen prvi deo). Mapu koristite da menjate položaj protivnika. Od kuće krenite ulivo, pa naviše. Sad ste u močvari. Blizim putem nastavite naviše, a onda okolnim putem krenite levo. Kad stignete do kraja ekrana, krenite naviše. Drugi deo močvare predite okolnim putem koji vodi naviše. Kad izadete, imate jedan jedini put; prođite pore kod ulaza u zamak. Ubijte aždaju, stanite na zvezdu i prođite kroz kip. Kad završite igru, dobićete čestitku što ste zatvorili vrata pakla.

Ako ste se u toku igre negde zaglavili, RESTORE vas vraća na početni meni. Igru ne možete završiti ako nemate krst. Kroz močvaru ne krećite bez napitka.

Vladimir Stakić

### Poukova mreža

CPC-464  
 HAUNTED HEDGES  
 BOMB JACK  
 LIGHT FORCE  
 JET BOOT JACK

MEMORY 32999-LOAD „HEDGES“-POKE 34871.n.CALL 33200  
 MEMORY 5999-LOAD „BOMB JACK“-POKE 6144.n.CALL 6000 (ni POKE 6653.0 za besmrtnost)  
 Prvo se unese „RUN“ (ENTER), potom ESC, a zatim: MEMORY 4079-LOAD „LIGHTFORCE“-POKE 27887.3.CALL 4080. (Ne znam zašto mora RUN) pa ESC, ali mora. Ako niko zna, nek mi javi)  
 Prekinite bezik loader i neposredno pre komande CALL 4900, u liniji 110, unesite POKE 32338.0 (besmrtnost) ili POKE 7503.n (za n života).

Vladan Matic, Milošević

„Spektrum“  
 TIGARD  
 ROGUE TROOPER  
 TARZAN  
 TOP GUN

POKE 27212, 0 (besmrtnost)  
 POKE 35091, 0 (municija)  
 POKE 30844, 231: POKE 30845, 3 (snaga)  
 POKE 51005, 0: POKE 51188, 0 (besmrtnost)  
 POKE 26460, 0 (besmrtnost)

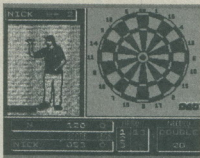
Ivan Kapičić

## PUBGAMES

Naslov je jasan: igre koje se igraju u pabovima. Ovdje ih ima sedam: pikado, kuglanje, blekdžek, poker, stoni fudbal, bilijar, domine.

### DARTS (pikado)

Ova mi se igra najmanje dopada, iz mnogo razloga: grafika je maltene zanemarena, a različitost i skrolovanje nisu ni za jedinicu. To je malo lošija kopija igre 180 u kojoj vam je zadatak da ruku, prikazanu na tabli za pikado, pomerate u određenom smeru, u zavisnosti od toga koliko vam poena treba, da biste naposljetku dobili konačan broj: nulu. Krećete sa 501 poenom, i kako gđate tablu, tako se broj poena smanjuje, da biste na kraju bili nagrađeni prigodnom muzikom i novom mogućnošću za igranje (naravno, sada možete birati nivo, broj igrača itd.)



određujete hoće li ta lopta imati unutrašnji ili spoljni felš. Cilj je da lopta ude u rupu na stolu. (Da, na stolu; ne po uglovima. I ja sam se iznenadio kad sam ih video usred stola). Morate paziti da ne srušite štapiće rasute po stolu u vidu Bermudskog trougla (zanimljivo), inače vam sleduju kazneni poeni.

### BOWLING (kuglanje)

Cilj je jasan: treba srušiti što više keglji (po mogućnosti 6). Treba da pazite pod kojim azimutom bacate kuglu, jer su i milimetri u ovoj igri važni. Vi ste prikazani u gro planu, a iznad keglji se nalazi nišan; njime određujete ugao pod kojim ćete izbaciti kuglu. Igra je grafički dobro urađena, a ni muzika ne zaostaje za onom iz DARTS-a.

### BLACK JACK (blekdžek)

Ova igra, zajedno sa narednom, po meni je vrhunac u kompletu PUBGAMES zbog izvrsne grafike i skrolovanja (samo malo iza igara kao što su SANXION i URIDIUM). Stari listić blekdžeka ovdje neće imati problema, a i novajlije će brzo ući u „fazon“, tako da o tome ne bih previše. Za razliku od mnogih kartaških simulacija, u ovoj su karte poređane realistički — lepezasto a ne horizontalno — a i znaci su zalista lepo urađeni...

### BILLIARDS (bilijar)

Kao što rekoх, biser PUBGAMES-a. Na početku igre prikazuje vam se samo deo stola; kasnije se slika širi. Odmah na početku birate kojom ćete loptom gđati: crvenom, belom, ili belom sa tačkom. Štap možete pomerati ulevo i udesno, tako da

### POKER

Sigurno ste igrali mnoge kompjuterske varijante strip-pokera (među njima, recimo, i Samantu Foks). Ovo je običan poker, i to malo lošija verzija, zato što nema slika koje dočaravaju ambijent. Međutim, ne očajavajte: ova će vas igra primorati da podosta sedite nad kompjuterom dok ne pronađete „zakut“.

### KICK OFF (početni udarac)

Uz bilijar i blekdžek, ovu bih igru svrstao u red boljih, prešteno zbog odlične grafike i, naravno, skrolovanja. Ideja je ocenjena desetakom: mislim da nema nijednog Jugovića koji još nije bio u luna-parku i igrao stoni fudbal. Imate četiri poluge kojima pokrećete igrače (naravno, i vaš protivnik isto toliko). Međutim, ako jedan igrač udari lopticu, igrači nanizani na drugoj šipki ne mogu se okretati sve dok im lopta ne ude u „fokus“. Posebnu plusiasticu predstavlja mogućnost „roštiljanja“, kao i na pravom stonom fudbalu u luna-parkovima...

(Inače, svaka od ovih 7 disciplina učitava se posebno, tako da možete preskočiti one koje vam ne krasi život).

Predrag Ivanović

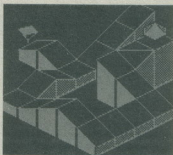
## SPINDIZZY

C64

Čigra se prvi put pojavila u Gyroscope-u, gde se kretala po određenom putu, zbog čega je verovatno mnogima brzo dosadilo. Međutim, na sceni stupaju verni nastavci: Spindizzy. Ova igra je takode rađena u 3D maniru, ali je lavirintskog tipa. Na početku, vreme iznosi 107 jedinica i smanjuje se tokom igre padova ili, u redim slučajevima, sa vašim suparnicima, a nadoknađuje se kupljenim dijamantima.



Pre starta igre na raspolaganju su vam sledeće opcije: S-pokazuje poslednji postignuti rezultat; O — dobijate novi meni tako što se sa F1 reguliše brzina kretanja, sa F3 birate igru sa jednom ili dve palice odjednom, dok sa F7 se vraćate na početni meni; H — vam daje na ekranu popis i opis vrste podloga i pomagaca u toku igre.



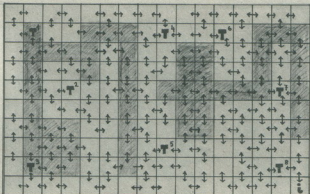
Pucanjem startujete igru. Igra ima 386 slika. Dijamanti su razbacani većinom po nemogućim mestima. Ni upravljanje nije nauvno, jer se čigra ili loptica kreće u deset različitih brzina. Položaj u lavirintu dobijate rotiraju lavirint sa F3 na levu ili F5 na desnu stranu. Ukoliko vam je dosadila četvrtasta čigra, sa I je promenite u tanku čigru ili, ako vam se više sviđa, u lopticu. C menja boju lavirinta i pozadinu ekrana. Pritiskom na P zaustavlja vreme na neki period. Informacije o postotku: pređenih slika, pokupljenih dijamantima i pređene igre dobijate na S. Ako ste se negde zaglavili, pritisnite Q pa počnete igru ponovo.

Uz pažljivo upravljanje, uz puke za vreme ili bez njega, trebaće još 5—6 sati naporne igre.

Vladimir Stakić

### BOMB SCARE

- T = telefonti  
 1. Zepha  
 2. Quart  
 3. Delta  
 4. Xylem  
 5. Nitro  
 6. Crypt  
 7. Ytron  
 8. Astra  
 i = izlaz



Saša Ampovski



# Šakom i pamom

## HEAD OVER HEELS

nepobedivost

```

1 CLEAR 44580
2 LET I=0;LET W=0
3 FOR F=#2880 TO 32178
40 READ A;FOKE F;A
50 IF A<=100;LET W=1
60 NEXT F
20 IF C<1744297 THEN PRINT "E"
DATA EROR#1;STOP
30 PRINT AT 18,1;"START 'HEAD OVER HEELS' TAPE"
30 RANDOIZE USR 32880
180 DATA 221,33,283,92,17,234
110 DATA 4,62,255,25,285,86,5
120 DATA 48,241,243,237,94,33
130 DATA 44,129,229,33,173,99
140 DATA 229,91,31,17,163,252
150 DATA 1,22,33,283,94,25,199
160 DATA 282,237,79,195,173,99
170 DATA 35,78,129,229,33,182
180 DATA 252,229,31,31,17,289
190 DATA 252,1,232,2,33,289,232
200 DATA 62,196,237,79,195,199
210 DATA 252,33,289,252,17,289
220 DATA 138,1,92,6,237,174,33
230 DATA 229,138,34,253,138,34
240 DATA 237,138,33,218,138,34
250 DATA 246,138,33,252,138,34
260 DATA 9,139,62,195,58,29,139
270 DATA 33,114,129,243,19,199
280 DATA 195,289,138,175,58
290 DATA 166,255,62,195,58,69
300 DATA 255,33,288,258,54,180
310 DATA 255,33,148,125,17,258
320 DATA 258,1,258,62,36,115
330 DATA 55,255,33,62,34,113
340 DATA 163,33,148,125,17,258
350 DATA 163,62,33,58,128,163
360 DATA 58,123,163,49,255,355
370 DATA 195,48,112
    
```

## SPEED KING 2

moć da prođete kroz ostale vozače

```

1 BORDER B; PAPER B; INK B
2 CLEAR 24599
30 PRINT AT 10,3;"START 'SPEED KING' TAPE"
40 LOAD "CODE" I;LOAD "CODE"
50 FOK BRN#24
60 RANDOIZE USR 58339
    
```

## SIGMA 7

neograničena količina svemirskih brodova

```

1 CLEAR 65535
2 LET I=0;LET W=0
3 FOR F=#6800 TO 65840
40 READ A;FOKE F;A
50 IF A<=100;LET W=1
60 NEXT F
20 IF C<133316 THEN PRINT "E"
RROR IN DATA";STOP
60 PRINT AT 10,6;"START 'SIGMA 7' TAPE"
70 RANDOIZE USR 65880
180 DATA 6,3,197,221,33,0,8
110 DATA 17,17,6,62,255,35,295
120 DATA 86,5,193,16,237,221
130 DATA 33,48,1,253,19,199
140 DATA 42,225,55,285,86,5
150 DATA 62,258,58,156,133,58
160 DATA 138,233,156,133,229
200 DATA 195,196,235
    
```

## NETHER EARTH

beskonačne zalih

```

1 BORDER B; PAPER B; INK 7
2 CLEAR 65535
30 PRINT AT 10,3;"START 'NETHER EARTH' TAPE"
20 LOAD "CODE" 64730
30 FOK 64735
40 FOR F=#6824 TO 68236
40 READ A;FOKE F;A;NEXT F
50 DATA 62,138,33,218,11,173
60 DATA 62,33,58,71,174
70 DATA 195,0,164,163,255,258
80 RANDOIZE I;LOAD 64730
    
```

## AMAUROTE

podšavate elemente po želji (INFINITE MONEY znači "beskonačno mnogo novca", INFINITE BOMBS "bezbroj bombi")

64 računari 29. e avgust 1987.

## a INVINCIBLE („nepobediv“)

```

1 CLEAR 20599
2 FOK #24500,0
10 PRINT AT 10,5;"START 'AMAUROTE' TAPE"
10 LOAD "SCREENS";LOAD "CODE"
20 INPUT "INFINITE MONEY (Y/N)";I
7 I=48
25 IF A="" THEN FOK 46301,1
20 I=0
30 INPUT "INFINITE BOMBS (Y/N)";I
7 I=48
35 IF A="" THEN FOK 40615,2
40 INPUT "INVINCIBLE (Y/N)";I
45 IF A="" THEN FOK 46312,3
50 RANDOIZE USR 26680
    
```

## GAUNTLET

beskonačna energija

```

1 CLEAR 64999
2 LET I=0;LET W=0
3 FOR F=#68000 TO 68032
15 READ A;FOKE F;A
20 IF A<=100;LET W=1
30 NEXT F
20 IF C<444441 THEN PRINT "ER"
RROR IN DATA";STOP
30 DATA 211,57,218,254,17
40 DATA 81,1,62,255,35,285
50 DATA 80,5,48,241,33,173,94,33
60 DATA 34,257,255,243,195
70 DATA 6,255,62,291,58,82
80 DATA 184,195,8,132
90 PRINT AT 10,5;"START 'GAUNTLET' TAPE"
90 RANDOIZE USR 65880
    
```

## INTO THE EAGLE'S NEST

beskonačno mnogo munjice i kijučave, nepobedivost

```

1 CLEAR 29599
2 LET I=0;LET W=0
3 FOR F=#48000 TO 48037
40 READ A;FOKE F;A
50 IF A<=100;LET W=1
60 NEXT F
20 IF C<25217 THEN PRINT "ER"
RROR IN DATA";STOP
180 DATA 33,114,129,243,19,199
110 DATA 8,237,176,190,8,91
120 DATA 33,255,227,17,255,255,1,8
130 DATA 129,237,184,175,38
140 DATA 32,163,58,176,168,58
150 DATA 64,195,8,128
160 FOK 25840,0
200 INPUT "INFINITE ARMS (Y/N)";I
7 I=48
210 IF A="" THEN FOK 64820,5
220 INPUT "INVINCIBLE (Y/N)";I
48
230 IF A="" THEN FOK 64829,9
240 INPUT "INFINITE KEYS (Y/N)";I
48
250 IF A="" THEN FOK 64832,9
260 PRINT AT 10,8;"START 'INTO THE EAGLE'S NEST' TAPE"
260 LOAD "CODE"
400 FOK 58336,26
410 FOK 58325,258
420 FOK 58649,198
430 RANDOIZE USR 58368
    
```

## URIDIUM

podšavate elemente po želji (INFINITE LIVES = „bezbroj života“, PASS OVER WALLS ETC = „prelaženje preko zidova i ostalog“ NO ALIENS, LAND NOW PRINTED STRAIGHT AWAY = „bez vanzemaljaca, zemlja se sada smesta štampa“)

```

1 LET I=0;LET W=0
2 FOR F=#48000 TO 48048
10 READ A;FOKE F;A
15 IF A<=100;LET W=1
12 NEXT F
15 IF C<3398781 THEN PRINT "E"
RROR IN DATA";STOP
200 DATA 221,33,39,244,17
    
```

```

25 DATA 129,2,62,255,55
30 DATA 285,86,237,94,33
35 DATA 283,62,48,58,48
40 DATA 245,33,195,86,42
45 DATA 254,24,184,245,58
50 DATA 188,245,33,0,8
60 DATA 34,145,195,8,8
65 DATA 245,33,114,254,17
68 DATA 8,41,148,62,237
70 DATA 174,195,48,44,33
75 DATA 255,237,17,255,395
80 DATA 1,6,145,237,18,6
85 DATA 62,34,58,79,138
90 DATA 62,195,58,99,138
100 DATA 195,58,235
105 FOK 25848,0
110 INPUT "INFINITE LIVES (Y/N)";I
15 I=48
160 IF A="" THEN FOK 65050,5
120 INPUT "PASS OVER WALLS ETC (Y/N)";I=48
130 IF A="" THEN FOK 65054,5
140 INPUT "NO ALIENS, LAND NOW PRINTED STRAIGHT AWAY (Y/N)";I=48
150 IF A="" THEN FOK 65061,5
160 PRINT AT 10,7;"START 'URIDI' TAPE TAPE"
200 RANDOIZE USR 64950
    
```

## ENDURO RACER

neograničeno vreme

```

1 CLEAR 64880
2 LET I=0;LET W=0
3 FOR F=#30000 TO 32155
10 READ A;FOKE F;A
15 IF A<=100;LET W=1
20 NEXT F
25 IF C<131517128 THEN PRINT "E"
DATA EROR#1;STOP
30 PRINT AT 10,3;"START 'ENDURO RACER' TAPE"
50 RANDOIZE USR 32880
180 DATA 221,33,283,92,17,234
110 DATA 4,62,255,25,285,86,5
120 DATA 48,241,243,237,94,33
130 DATA 44,129,229,33,188,99
140 DATA 229,91,31,17,163,252
150 DATA 1,29,33,283,94,62
160 DATA 282,237,79,195,188,99
170 DATA 38,78,129,229,33,199
180 DATA 252,229,31,31,17,289
190 DATA 252,1,232,2,33,289,232
200 DATA 62,196,237,79,195,199
210 DATA 252,33,289,252,17,289
220 DATA 138,1,92,6,237,174,33
230 DATA 229,138,34,253,138,34
240 DATA 237,138,33,218,138,34
250 DATA 246,138,33,252,138,34
260 DATA 9,139,62,195,58,29,139
270 DATA 33,114,129,243,19,199
280 DATA 195,289,138,175,58,163
290 DATA 166,255,62,195,58,69
300 DATA 58,123,163,49,255,355
310 DATA 34,188,255,33,148,125,17
320 DATA 188,259,1,58,6,237,174
330 DATA 195,255,286,38,58,68
340 DATA 161,164,195,8,91
    
```

## KNUCKLE BUSTERS

bezbroj života

```

1 BORDER B; PAPER B; INK 7
2 CLEAR 24575
15 PRINT AT 10,2;"START 'KNUCKLE BUSTERS' TAPE"
20 LOAD "SCREENS"
30 LOAD "CODE"
40 FOK 23991,45
50 RANDOIZE USR 33796
    
```

## LIGHTFORCE

neuništiv brod

```

1 BORDER B; PAPER B; INK 7
2 CLEAR 65535
3 FOR F=#48000 TO 48007
40 READ A;FOKE F;A;NEXT F
50 DATA 62,34,58,71,174
60 DATA 159,195,48,154
100 PRINT AT 10,4;"START 'LIGHT FORCE' TAPE"
110 LOAD "CODE"
120 FOK 68014,0
130 FOK 68016,76
140 FOK 68017,178
150 FOK 68020,258
200 RANDOIZE USR 68020
    
```

## BUTCH HARD GUY

bezbroj života

```

1 BORDER B; PAPER B; INK 7
2 CLR
50 PRINT AT 18,2;"START 'BUTCH HARD GUY' TAPE"
30 LOAD "CODE" 16384
20 FOK #23315 TO 23321
25 READ A;FOKE F;A;NEXT F
30 DATA 179,58,44,138
40 DATA 195,133,158
50 RANDOIZE USR 23796
    
```

## MARTIANOIDS

bezbroj života

```

1 INK 7; BORDER B; PAPER 0
2 CLEAR 65535
3 LOAD "CODE"
10 RANDOIZE USR 24576
20 LOAD "CODE"
30 FOK 46793,0
40 RANDOIZE USR 38481
    
```

## SHOCKWAY RIDER

bezbroj života

```

1 BORDER B; PAPER B; INK 7
5 CLEAR 24799
50 FOK=#68000 TO 64818
60 READ A;FOKE F;A;NEXT F
70 DATA 33,29,188,54,8,35,54
80 DATA 58,195,288,167
100 PRINT AT 10,7;"START 'SHOCKWAY RIDER' TAPE"
120 FOK 68024,0
130 FOK 68016,76
140 FOK 68017,178
150 FOK 68020,258
200 RANDOIZE USR 68020
    
```

## PAPERBOY

nepobedivost

```

1 CLEAR 65535
10 PRINT AT 10,5;"START 'PAPERBOY' TAPE"
20 LOAD "CODE"
30 FOR F=#2880 TO 32824
40 NEXT F
50 DATA 33,22,255,24,86,254
60 DATA 229,33,148,125,17,258
70 DATA 6,237,176,190,8,91
85 DATA 38,156,195,197,255,255
100 RANDOIZE USR 32880
    
```

## SHORT CIRCUIT

bezbroj života (potprogram važ samo za drugi deo)

```

1 CLEAR 44580
2 LET I=0;LET W=0
3 FOR F=#2880 TO 32169
40 READ A;FOKE F;A
50 IF A<=100;LET W=1
60 NEXT F
20 IF C<1744297 THEN PRINT "E"
30 PRINT AT 10,8;"START 'SHORT CIRCUIT #212' TAPE"
50 RANDOIZE USR 32880
180 DATA 221,33,283,92,17,234
110 DATA 4,62,255,25,285,86,5
120 DATA 48,241,243,237,94,33
130 DATA 44,129,229,33,188,99
140 DATA 1,42,33,283,94,62
150 DATA 282,237,79,195,188,99
160 DATA 38,78,129,229,33,199
170 DATA 252,229,31,31,17,289
180 DATA 252,229,31,31,17,289
190 DATA 252,1,232,2,33,289,232
200 DATA 62,196,237,79,195,199
210 DATA 252,33,289,252,17,289
220 DATA 138,1,92,6,237,174,33
230 DATA 229,138,34,253,138,34
240 DATA 237,138,33,218,138,34
250 DATA 246,138,33,252,138,34
260 DATA 9,139,62,195,58,29,139
270 DATA 33,114,129,243,19,199
280 DATA 195,289,138,175,58,163
290 DATA 166,255,62,195,58,69
300 DATA 58,123,163,49,255,355
310 DATA 34,188,255,33,148,125,17
320 DATA 188,259,1,58,6,237,174
330 DATA 195,255,286,38,58,68
340 DATA 161,164,195,8,91
    
```

Ako programeri nemaju milosti, imaju je redakcije časopisa. Sinclair User i Comodore User u poslednje vreme objavljuju dodatke sa mapama igara i savetima za igranje. Tiraž raste, igrači gube manje vremena; svima lepo.

Odnedavina je isprednjačio Sinclair User; objavio je kartice sa preprogramima za besmrtnost, nepobedivost, neograničenu količinu municije, bezbojno bušenje ušiju, jednačenje po zvučnosti itd. Ukucate preprogram, pritisnete RUN (potom, naravno, ENTER), i najzad, kad vam kompjuter javi da je sve izvršeno, učitate igru; život tada postaje lakši.

Naravno, ne znamo svi engleski, i ne čitamo svi engleske časopise... ali tu su Računari da posreduju. Spektrumovci, izvoliše.

**HEAD OVER HEELS**



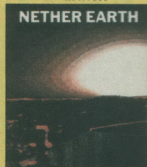
**SPEED KING 2**



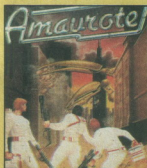
**SIGMA-7**



**NETHER EARTH**



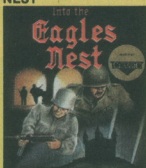
**AMAUROTE**



**GAUNTLET**



**INTO THE EAGLE'S NEST**



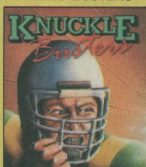
**URIDIUM**



**ENDURO RACER**



**KNUCKLE BUSTERS**



**LIGHTFORCE**



**BUTCH HARD GUY**



**MARTIANOIDS**



**SHOCKWAY RIDER**



**PAPERBOY**



**SHORT CIRCUIT**



# Spektrum

## ENDURO RACER

### TRKA IZDRŽLJIVOSTI

Jeste li preživeli posljednju motosimulaciju?

Vrlo dobro. Vaš vredni „spektrum“ opet vas zove u sedlo, nudeći vam nov motocikl i nove muke.

I nove efekte, ako ćemo pravo. Tako će lakše podneti sve one zidove, stene, kaktuse, kamione, peščane nanose i razne druge pogibelji. Vrlo često moraćete skakati (ubrzajte, pa dođite prema sebi), što je za bubreg prava stvar. Obilježje je manji problem — ako niste prevište brzi — a kad naidete na pesak nastojte da razvijete nešto šesto čulo, jer ovih regularnih pet neće vam biti od prevelike koristi.

Sve vreme će u na snazi biti ona stara jednadžba po kojoj je sposobnost manevriranja u obrnutoj srazmeri sa brzinom. Nemojte je zaboraviti, no pošto vam je brzina bitna, računajte sa tim da vam je u prvom delu



pametno biti na (ili oko) 195 kilometara na sat, a da je u drugom dovoljno samo malo predi stotku.

Život će vam posebno zagorčavati kolege, koji u vama osećaju najjače konkurenta pa gledaju da vas uklone (ovaj glasog možete shvatiti na razne načine). Ali ko će koga ako ne svoj svoga?

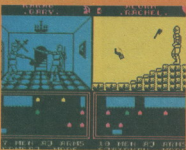
## THRONE OF FIRE

### VATRENI PRESTO

Bilo jednom jedan kralj, pa imao tri sina. Kad je kralj umro, tri sina su se počela tući za presto. Pošto na prestolu ima mesta samo za jednog, onaj ko hoće da na njega sedne mora prvo dobiti likvidirati ostala dvojicu. I tako je počela borba.

Lidi li vam ovo na bajku? Ne verujem, jer mnogo puta u istoriji na vlasti su dolazile bratubice. Tako će biti i u ovoj (kao što vidimo, realistično) igri.

Štvar se uglavnom svodi na šunjanje po zamku (8% odnosa), uključujući prestonu dvoranu i tuču sa protivnicima. Pretendent na presto ima uza se desetericu vojnika — i skener. Skener je u dnu ekrana, i pokazuje da li je ovaj ili ona odaja prazna (mrak) ili ne osvetljena bojom koja dotičnog identifikuje. U gornjem uglu levo ili desno (zavisno od opcije) kuca pretendentovo srce; brzo ako energije



ima dosta, sporije i sporije ako energije opada. Energiju obnavljaju napici.

Možete igrati udvoje (tada je kompjuter treći brat) ili sami (tada je kompjuter dvojica braca, što je interesantan oblik elektronske šizofrenije).

## MARTIANOIDS

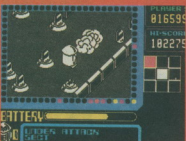
### MARSOVCOIDI

Malo vam je Marsovaca, sad su se pojavili još i Martovcoidi. I to u najgorem trenutku: dođete u kosmičkom brodu. Cilj im je da onesposobe vaš glavni kompjuter — a samim tim i vas.

Zlo biste prošli da nemate robota koji sakuplja akumulatore, štiti sve sektore broda i stara se da svaki program koji je iz sektorskih kompjutera krenuo u glavni procesor stigne na određeno neometano. Ovo poslednje veoma je važno: kad program stigne na mesto, aktiviraju se broski odbrambeni mehanizmi i neprijatelji počinju padati kao kruške.

U međuvremenu, robot se suzi svojim naručivanjem. Ako treba samo likvidirati protivnike, ispušta oblak otrovnog gasa, a ako je nešto krupnije u pitanju, puca iz megapisto-  
lja koji razara sve (čak i predmete koji su vam potrebni; neprecizan naliđanja može sam sebe upropastiti).

Protivnici su mu svakojaki: ima čak i miševa na navijanje, gladnih njegove energi-



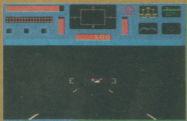
je. Svi oni nastoje da počupaju kratke prugaste bandere kojima se programi prenose do procesora.

Obaveštenje o njihovom napadu dobijate na mapi sektora (kvadratu podeljenom u devet kvadratića). Kad vidite da je neki kvadratić počeo da bleska crveno i žuto, smesta šaljite robota u taj sektor da raščisti stvar. Da je uspeo (kao i uspeo), požaćete po tome što će kvadratići poobeleti.

Kad svi pobele, ustanite i večerajte.

## STAR RAIDERS II

### NAPADAČI SA ZVEZDA II



Zaviđala nostalgija? Preslušate ideje? Ko bi to znao? U svakom slučaju, pred nama je stara i dobro znana igra, malo našminkana da joj se ne vide bore. Priču znamo: vanzemaljaca je mnogo, a vi ste usamljeni. Oni rade vama o glavi — vi radite njima o glavi.

Pomaže vam kompjuter, koji automatski bira oružje, već prema tome da li pucaete na letelice manjeg formata ili na njihov matični brod. Imate i energetske štitove, a možete doznati i ekran sa informacijama da biste videli treba li ići na remont ili ne (jer vaš će brod bivati oštećen, a i municija će se smanjivati). Ako vam dođe voda do poda, možete bežati i u hiperprostor (hyperspace), što u stvarnosti ne postoji, a u prevodu znači: znaš gde si uleteo, ali pojma nemaš gde ćeš izleteti.

Dobar deo igre dešava se i nad tuđinskom planetom: tu vam je zadatak da uništavate neprijateljske baze (koristeći veoma efikasne bombe).

Potrošićete grdo mnogo municije i — nadam se izbeći našu neudnu planetu.

## THRUST II PRODR II



Još jedna varijanta budućnosti.

I to prilično crne. Totalitarna imperija zavela je strahovladu širom galaksije, i — već je krvca iz zemlje provela i Zeman do'o, valja ustajati! Pobunjenici su osvojili jednu planetu; ona će im biti veoma korisna baza.

... ali treba je prvo dovesti u red. Prekrivena je crvenom prašinom, koja se mora počistiti; u te svrhe skupljate energetske kugle kojima se hrani prečišćivač vazduha. Kao da traganje već samo po sebi nije mučno, ispostavlja se da kugle ekspandiraju ako ih blagovremeno ne smestite dugo treba — i da nisu sve podjednako teške, što stvara transportne i druge probleme.

Pa kao da čak ni to nije dosta, napadaju vas svakojaki androidi — a njih samo kiselina može uništiti; pa morate skupljati i nju. (Ni to nije kraj). Svaki tip androida osetljiv je samo na „svoju“ kiselinu; morate, dakle, pronaći koja na koga deluje).

Ukratko, muke i nevolje. Uostalom, niste valjda verovali da je moćna galaktička imperija moguće srušiti tek tako, bez znoj, krvi i suza.





## Komodor

### NEMESIS THE WARLOCK VEŠTAČ ZVANI NEMEA



Nastov se baš nije proslavio. Onako sačinjen od naslova dveju drugih igara, deluje kao fudbalski klub **HAJDUČKA ZVEZDA**, ili kao roman **NA DRINI IDIOT**.

Zato je igra njena. Kao veštac zvani Nemeza, rešili ste da uništite carstvo zlog Torkemada (ne onog inkvizitora, mada izvesna sličnost postoji). Imate mač, imate pištolj, imate i razne bajalice i vradžbine, i krečete.

Na vas se obrađavaju Torkemadini ratnici, nazvani Terminatori (što ova igra duguje **Bojnom životu?**) i dok ne pobijete određenu kvotu, ne možete u sledeći ekran. Korisno je znati da vam i njihova tela mogu poslužiti kao platforma.

Ali... što više vremena utrošite, to je jača Torkemadina moć, tj. lice mu je sve vidljivije na ekranu. Na kraju, ako ste spori, pobijeni Terminatori postaju zombiji i napadaju vas.

Vajta i ovo znati: pištolj radi tek kao drugi priručnik **FIRE** (prvo pritiskanie samo znači da ste ga potegli iz futrole). Municiju morate tražiti po ekranima; ma koliko je bude, ne uzimate više od 12 metaka.

### AUF WIEDERSEHEN, MONTY DO VIDENJA, MONTI



Ala! vera na maloga batul!

Neumitlivi krt Monti, pošto je u trećem nastavku pobeo preko Lamansa, sad se sprema u Grčku, na ostrvo Montos, jer sa tog ostrva nema ekstradicije.

Ali ostrvo treba kupiti, a Monti je švorc; zato se, kao svaki pametan krt, počinje šunirati po Evropi — od Atlantika do Urala, što rekao De Gol — i pravi biznis. U Monaku se takmiči na auto-trkama, u Francuskoj nastoji da drpi Mona Lizu i uvaja je za debelu lov, u Italiji prodaje fudbalsku loptu Juventusu, itd. (Protazi i kroz Jugoslaviju — tačnije, kroz Zagreb. Da li da bi ošepario na Univerzijadi ili ukrao šta iz Mitinarne zbirke, ne znam). Usput, prirodno, skuplja korisne predmete, uključujući i avionske karte, jer će često leteti iz jedne zemlje u drugu.

Ima, doduše, i pogibelnih predmeta, i uz to mnogo neprijatelja. Povrh karakonzula u raznim oblicima, tu su, razume se, i oni nezbežni pištolji koji prete da od njega za sekundu naprave križnji hamburger.

Ako bude brz, precizan i, nadasve, pametan, pokupiće šta nu treba.

## Amstrad

### LIVINGSTONE, I PRESUME

#### LIVINGSTON, PRETPOSTAVLJAM

U pitanju je doktor Livingston, pretpostavljam.

Jer vi ste Stenli, onaj koji ga je najzd našao u sred Afrike i izgovorio dotičnu istorijsku rečenicu. Kad budete u situaciji da je i vi kažete, završičete igru kao pobednik.

Ali do kraja ima još mnogo, a vi tek treba da počnete.

Počinjete opremljeni bumerangom, bodžom, ručnom bombom i motkom za skakanje. Nastavljaite kroz 63 ekrana, kroz džunglu i hrmove u njoj, okruženi živim i neživim neprijateljima. Majmuni će vas gadati kokosovim orasima. Zamke će samo čekati da upadnete u njih. Vodena čudovišta, vešto gladna, naonošite zube da vas što dostojnije prime.



Ljudožderi — čija je deviza „Ništa ljudsko nije mi tuđe“ — gejače vodu u kazanima i prepirali se oko toga da li ste ukusniji sa mešanom slatim ili sa senfom.

I tako dalje.

Vaša energija, hrana i voda, kao što ste i mogli pretpostaviti, nisu neiscrpne. Morate se usput snalaziti da ih obnovite: ne bi valjalo da izdahnete iscrpljeni na dva koraka od čoveka koga hoćete da spasete.

## BARBARIAN

#### VARVARIN

Pucaju rebra, padaju glave, šiklja krv.

Neki put i malo previde, za nas koji nismo sadiati. Ali eto, takav je život — ili, jasnije rečeno, takvo je izdžito.

Pravozračnik čilave ove klanice jeste ali čarobnjak. Njemu je zapala za oko lepa Manjana; ako je ne dobije, uništiće njen grad i pobiti sve do miša u duveru. Ko hoće da je oslobodi, mora se tući s njegovim borcima (i pobediti, naravno). Bori su okrutni i odlično uvezdani, spasa nema nikogud.

Ali pojavljuje se... ko? Varvarin (Ne onaj što je bilo Stalada, nego onaj što ima naglasak na prvom slogu). On kreće u borbu, bijudi protivnike rukama, nogama, glavom i



mačem. Ako zada šest udaraca, pobeđuje; ako primi šest, odlazi u vednu lovnica. Događ je živ, tuđi će se ili u šumi ili na borištu u čarobnjakovom zamku.

Ne veliku radost ljubitelji nastoje

## PROHIBITION

#### PROHIBICIJA

Budalasta ideja.

Sedamnaest godina u SAD je bilo zabranjeno proizvoditi, uvoziti i konzumirati alkohorna pića. Od tog režima videla je vajdu samo mafija, koja je zgumla nezamislive pare praveći i švercujući razne vrste vatrene vode.

Vi treba da ih onemogućite, jer ste policajac (jedan jedini na ko zna koliko gangstera). Čekate ih u zasedi i pucate na njih kad se pojave. Imate 5 sekundi da ih uhvatiti u nišan. Ali nisu ni oni veslo sisali. Ne pojavljuju se onde gde ih možete odmah videti; pojavljuju se van ekrana. Vi, dakle, morate sve vreme šverkovati uljevo, odnosno, navide ih razniže. Donekle vam pomaže strelica koja pokazuje uljevo ili uldesno — ali kad dođete do pravog



ekrana, vertikalno traženje morate obaviti sami.

Ako i porad svega dobro nandante i pogodite gangstera, on će pasti, a vi ćete ići dalje. Ako promašite, nećete ići dalje, jer će on pogoditi vas.

Vrlo jednostavno, u suštini.

**JA U ŠKOLU IDEM I DOBAR SAM ĐAK!**

**MOJA PRVA ŠKOLSKA TORBA**



MAME, TATE,  
DEKE I BAKE  
ZA VAŠE  
PRVAKE! —  
MOJA PRVA  
ŠKOLSKA TORBA.

ĐAČE PRVAČE —  
— SAMO ZA  
TEBE!

BEZ FRKE I  
ZBRKE, BEZ  
GUŽVE I TRKE,  
JEDINSTVENO!

PRVI PUT, TORBA  
HITA U TVOJ  
DOM.  
ZA DEVOJČICE I  
DEČAKE —  
LEPE TORBE ZA  
PRVAKE!

Đače prvače, u tvojoj prvoj školskoj torbi naći ćeš sve za čitanje, pisanje, crtanje i računanje. Iz svake torbe male smeše se knjige, sveske, pisaljke i računaljke, crvene, žute i plave.

Sadržaj svih torbi je isti — cena je jedinstvena.

**samo 25.000**

— izbor torbe je vaš

RODITELJI NARUČITE KOMPLET ŠKOLSKE TORBE ZA VAŠEG ĐAKA PRVAKA I DOBIĆETE SVE ŠTO JE VAŠEM MALOM UČENIKU POTREBNO ZA PRVE ŠKOLSKE DANE KUPOVINOM NA OVAJ NAČIN — KOMPLET KNJIGA, PRIBORA, SVESAKA I ŠKOLSKU TORBU DOBIJATE

**10% JEFTINIJE**

NE MORATE DA SE BRINETE I MISLITE — MI MISLIMO ZA I NA VAS!

Pored ozbiljnog znanja svaka torba sadrži i mali poklon iznenađenja. U svakoj torbi iznenađenje slatko — „BAMBI“ KEKS prija i krepki glatko!

Đaku prvaku, brkati čika VUK. U prvoj školskoj torbi i bež sa likom tvorca naše azbuke — Vuka Karadžića.

— Raspored časova — poklon  
— BIGZ-ova knjiga poklon

**SADRŽAJ moje prve školske torbe**

- |                               |   |
|-------------------------------|---|
| 1. ĐAČKA TORBA                | 11. MATEMATIČKI RADNI LISTIĆI               |
| 2. KOMPLET ŠKOLSKI PRIBOR     | 12. POZNAVANJE PRIRODE I DRUŠTVA            |
| 3. KOMPLET SVEZAKA (8)        | 13. LISTIĆI ZA POZNAVANJE PRIRODE I DRUŠTVA |
| 4. AKVAREL BOJE               | 14. PEŠAK I VOZAČ U SAOBRAĆAJU              |
| 5. PLASTELIN                  | 15. LEKTIRA — KOMPLET                       |
| 6. RAČUNALJKA                 |   |
| 7. BUKVAR — MOJA PRVA KNJIGA  |   |
| 8. NASTAVNI LISTIĆI UZ BUKVAR |   |
| 9. ČITANKA — DOBRO JUTRO      |   |
| 10. MATEMATIKA                |   |

Svi udžbenici i lektira su usaglašeni sa nastavnim planovima za 1987. godinu

RAČUNARI 29

Molim da mi posuđem gotaljete kompleta „Moje prve školske torbe, sa torbom br. \_\_\_\_\_ po ceni od 25.000 dinara.

IME I PREZIME \_\_\_\_\_

POŠTANSKI BROJ I MESTO \_\_\_\_\_

ULICA I BROJ \_\_\_\_\_

Naručeni komplet platitiću poštaru prilikom preuzimanja kao i poštanske troškove.

(Svojeručni potpis)

Naručenicu slati na adresu: BIGZ — Agencija DUGA, Bulevar vojvode Mišića 17/III, 11000 Beograd