



izdaje bigz izlazi jedanput mesečno

računari 36

časopis
za popularizaciju
informatike
i računarstva
mart 1988.
cena 1.000 din.
YUISSN 032552-7271

veštačka inteligencija

**ekspertni
sistemi
u kitu**

komodor

**buket
grafičkih
naredbi**

spektrum

**najbrža
elipsa u jugu**

ibm pc

**reci
dos-u
ne**



programski jezici

objektno programiranje

hardver

katalog 100 pc klonova

**igre
na
10
strana**



9 778603 527279

ŠKOLSKI RAČUNAR

TIM 011

TIM 011 je najnoviji model iz familije TIM računara namenjen opštem i profesionalnom obrazovanju u oblasti informatike i računarstva kao i unapređenju nastave.

TIM 011 je pogodan za efikasno vođenje školske administracije kao i za povezivanje sa centrima koji se bave obrazovnom problematikom (biblioteke, univerzitete itd.).

TIM 011 — do 10 računara povezanih sa profesionalnim školskim računarom TIM 020 kompatibilnim sa PC XT, predstavljaju snažnu laboratoriju za informatiku i računarstvo.

TIM 011 je usvojen u Beogradu kao standard za osnovno i usmereno obrazovanje.

PROGRAMSKA OPREMA:

Operativni sistem:

Usavršen i proširen, CP/M kompatibilan, disk operativni sistem

Programski jezici:

BASIC INTERPRETATOR
BASIC PREVODILAC
FORTRAN
COBOL
C
FASCAL
MODULA 2
PROLOG
LOGO

Veliki broj uslužnih programa.

procesor teksta,
baza podataka,
kalkulacije,
školski aplikativni programi.

Naš školski TIM dobija

Institut „Mihajlo Pupin“

počeo je od 15. februara isporuku računarskih učionica beogradskim školama. Tokom dva meseca, šezdeset pet škola će dobiti po deset računara:

TIM 011

po redosledu koji je utvrdila Zajednica obrazovanja SR Srbije.

TIM 011

predstavlja zaokružen računarski sistem velikih mogućnosti.

Za TIM 011 nije potreban kasetofon.

TIM 011 je opremljen brzim i kapacitetnim diskom.

Za TIM 011 nije potreban televizor.

TIM 011 uključuje i kvalitetan monohromatski monitor.

Deset računara TIM 011 povezanih sa PC XT kompatibilnim računarom TIM 020 predstavlja moćnu laboratoriju za informatiku i računarstvo.

Uđite i vi u svet informatike sa TIM računarima.

INSTITUT „MIHAJLO PUPIN“ RI, RAČUNARSTVO,

Beograd — 11060, Volgina 15,

Telex: 11584 YU imp bg.

Tel. 011/772-876

Tehničke karakteristike:

Processor: HD64180 (HITACHI),

osnovni takt 6, 144 MHz

— ugrađeni kontrolori interapta

(INTC)

— kontroler za upravljanje memorijom (MMU)

— dva kanala direktnog pristupa memoriji (DMA)

— časovnik realnog vremena

— programabilni sat/brojca

Memorija: ROM — 4 kB sa mogućnošću proširenja od 64 kB

RAM — 256 kB korisničke memorije

32 kB grafičke memorije

Spoljna memorija: Disketna jedinica kapaciteta 3,5 inča 800 kB

(ugrađeni kontrolori omogućavaju priključenje 4 jedinice)

Tastatura: — profesionalna, ergonomski oblikovana sa YU setom

— 95 alfanumeričkih i funkcionalnih tastera

— programiranje zvuka — jedan kanal 3 oktave

Monitor: Monohromatski-zeleni

Prikazivanje teksta:

— 24 reda po 80 simbola sa atributima

— četiri skupa simbola: cirilica, latinica, ASCH, korisnički

Prikazivanje grafičke:

— 512 x 256 tačaka bit-mapirano

— četiri intenziteta

Interfejs: — Serijski RS-232-C, za povezivanje u mrežu

— CENTRONICS, za priključenje štampača

Fizičke karakteristike:

Dimenzije: Računar sa monitorom: 375 x 380 x 315 mm

Tastatura: 480 x 245 x 40 mm

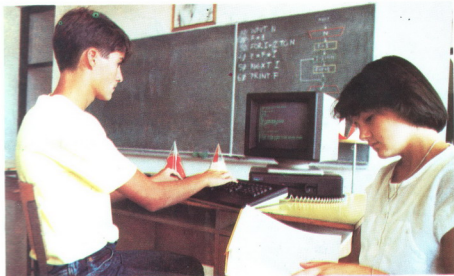
Masa računara: 12 kg

Uslovi eksploatacije:

Napajanje: 220 V (±10%, +15%), 50 Hz, 35 VA

Temperatura ambijenta: 10—35°C

Relativna vlažnost: 20—80%



tim 011



ŠKOLSKI RAČUNAR

časopis za popularizaciju
informatike i računarsva
izlazi jedanput mesečno
YUISSN 0352-7271

Izdaje
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvođe Mišića 17

Generelni direktor
Dobrosav Petrović
Glavni i odgovorni urednik
časopisa „Računari“
Stanko Stojiljković

Glavni i odgovorni urednik
Jovan Račanski

Stručna redakcija
Zorka Barčarić, Vola Gašić,
Slobodan Perović, Dejan Ristanović,
Jovan Skujanić, prof. dr Dušan Slavnić,
Nevanka Spalević, Zoran Živić,
Anđelko Zgoniec
Ekspertni redakcije
Zorka Bimović
Likovna grafičko uređenje
Mirko Popov
Marketing
Bergija Marčenko
Štampni saradnici
Nada Alakalić, Slobodanka Art, Zorko Berber-
Viktor Čerčević, Zoran Ovičarić, Nicoslav
Čebrić, Vola Gašić, Željko Jurčić, Blažimir Mile,
Zoran Obradović, Slobodan Perović, Miodrag
Potkonjak, Aleksandar Radovanović, Dejan Ri-
stanović, Jelena Ruznić, Duško Savić, Dušan
Slavnić, Jovan Skujanić, Nevanka Spalević, Vlada
Stojiljković, Sela Šviltica, Milan Tadić, Žarko
Vukosarijević, Anđelko Zgoniec, Zoran Živić
Izdavački savet „Računari“
Dr Rado Babić, prof. dr Branislav Dimitrijević
(predsednik), Radovan Drašković, Tanasija Ga-
vanović, Zivorad Gilić, Esad Jakupović, Vel-
izar Maslač, Nikola Pajić, Željko Penunović,
prof. dr Karmelo Ristić, Vlada Ristić, dr inž.
Milorad Teofilović, Vidolko Vajlićević, Valmir
Vasović, Milivoje Vuković

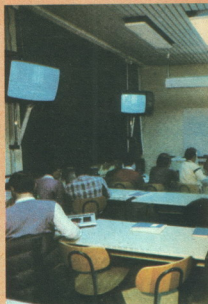
Adresa redakcije
11000 Beograd
Bulevar vojvođe Mišića 17/18
Telefoni
650-161 (sekretarijat)
653-748 (redakcija)
650-528 (prodaja)
651-793 (prodajna)
Raspisuje se na vrata
Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvođe Mišića 17

Preplata
Jugoslavija
— Za jednu godinu — 12.000,—
— Za šest meseci — 6.000,—
Na žiro račun: RO BIGZ 90802-620-
23264

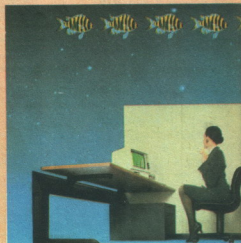
Inostranstvo:
— Za jednu godinu 27 US \$, 48 DM, 16 Letp,
40 Šv. 173 Šv. 184 Fr. ili 34.000 din. Na žiro
račun: RO BIGZ 60911-620-18101-620701-999-
03377

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

- 4)Šta ima novo
- 8)Obrazovanje
Računarske zvezde San Franciska
- 10)Računari u izlogu
Psion Organiser II
- 11)Skica za portret/Janko Mršić-Fišogel
Violinista na kompjuteru
- 12)Obrazovanje
Osnovna škola za odrasle
- 14)Dejanove pitalice
Najbolji rešavači
- 17)Stono Izdavaštvo (3)
Unos i priprema teksta
- 20)Izlog knjiga
21)Računari u izlogu
Ilustrovani katalog 100 PC klonova
- 26)Veštačka inteligencija
Ekspertni sistemi u kitu
- 28)Programski jezici
Objektivno programiranje
- 31)Komerijalni softver/CED
- 32)Komerijalni softver/CodeView
Ne daj bubu da te gricka
- 36)Unix bez muke (3)
Editori teksta
- 38)Algoritmi
Sve po listi
- 40)Klub Z80/Bežik klub
- 42)Help
- 44)Dejanove pitalice
Sanja voli Madonu
- 46)Tehnike programiranja/GUS
GUSan gače, ga, ga, ga . . .
- 47)Štampači/epson
NLQ u TOM-u
- 48)Tehnike programiranja/CoreWar
Dva se vola u programu bola
- 51)Tehnike programiranja/IBM PC
Reci DOS-u NE!
- 52)Tehnike programiranja/komodor
Buket grafičkih naredbi
- 54)Tehnike programiranja/spektrum
Najbrža elpsa u Jugi
- 56)Tehnike programiranja/spektrum
- 59)Uradi sam/Mali kućni roboti (2)
Put za periferiju
- 62)Mali oglasi
- 68)Pet plus
- 73)Razbarušeni sprajtovi



Osnovna škola za odrasle str. 12



Računarske zvezde san Franciska str. 8



Šta ima novo

Časopis
za slepe korisnike računara

Da slipe osobe nisu isključene iz korišćenja računala opće je poznata činjenica. Veliku pomoć slijepim osobama u Šri Njemačkoj pružile su redakcije magazina Chip, Computer Persönlich, c't i mc, izdavanjem magazina na kaseti nazvanog COMBIT. Na kaseti su snimljeni najzanimljiviji članci i novosti koje se nalaze u spomenutim magazinima. Cijena je 7,50 DM, a magazin na kaseti izlazi tri puta godišnje. Nadamo se da će ova informacija možda biti zanimljiva i našim slijepim korisnicima računala koji razumiju njemački jezik. Molimo sve čitaoce koji poznaju kojeg slijepog korisnika računala da ga upoznaju sa ovom humanom gestom. Adresa na kojoj se može dobiti više informacija je: Bit-Zentrum, Bayerischer Blindenbund e. V., Arnulfstrasse 22, 8000 München 2, BRD. (Z. V.)



MC 68020 i 32 korisnika

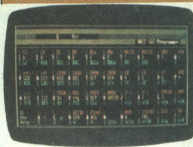
Višekorisnički računari su sve češće zasnovani na mikroprocesorima — Kowin Three firme Kowin Computer (američki telefon (800)445-8946) je zasnovan na Motorolinom MC 68020 i može da opsluži 32 grafičke radne stanice! 32-bitnom Motorolinom super mikroprocesoru pomažu dva „obična“ MC 68000 — jedan je zadužen za komunikaciju sa korisnicima, a drugi brine o pristupu hard disku od 40 megabajta. Centralna jedinica sa 4,5 megabajta RAM-a košta 11,990 dolara, a svaki terminal po 1,190 dolara. Dobra prilika da, uz minimalna ulaganja, proverite znanja koja ste stekli čitajući našu malu seriju tekstova o Junksu!

Profi tastatura
za „atari st“

Na tržištu se pojavila nova profesionalna tastatura za ATARI ST. Sadrži svu potrebnu tastaturu logiku, te utičnice za dojtik i miša. Priključni kabl je dug 2 metra, a u tastaturu je ugrađen i Reset taster. Cijena je 800 DM, a više informacija se može dobiti od: Binnewies, Bergfeldstrase 37, 3000 Hannover 91, BRD. (Z. V.)

HP-16C na PC-ju

Sećate li se „Pakardovih bombončića“ („Računari 21“)? HP-16C je kalkulator namenjen programerima — obezbeđuje rad u raznim brojnim sistemima, logičke operacije, konverzije... U blagodatima HP-16C ojednako mogu da uživaju i vlasnici PC-ja: XACT 16C firme CalcTech Inc (američki telefon (206) 6431682) je rezidentni program koji simulira poznati Hewlett Packardov kalkulator i to kako vizuelno tako i funkcionalno. Cena od 50 dolara zvuči sasvim razumno.



Tošiba ne staje

Japanski gigant Tošiba zasniva kompjuterski deo svoje reputacije na prenosivim računarima. Najnovija mašina namenjena poslovnim ljudima zasnovana je na mikroprocesoru 80386 (16 MHz), ima RAM od 2 megabajta, disk jedinicu od 3,5 inča, hard disk od 40 megabajta i specijalni plazma ekran koji obezbeđuje rezoluciju 640*400 (EGA). Cena, kako proizvođač kaže, „ne prelazi 6500 dolara“.

Prva animirana igra

Zar nisu sve arkadne igre animirane? Jesu, ali se ta animacija ne može poradićti čak ni sa osrednjim crtanim filmom. Vlasnici „atarja ST“ ojednako mogu da uživaju u potpuno animiranoj igri Disk Specil. Probajte — nećete verovati dok ne vidite!



Muke po Acornu

Da li se Acorn ponovo nalazi u krizi? Sa jedne strane, stvari stoje izvanredno: „arhimed“ je svuda dobro primljen, prodaju se sve mašine koje se proizvode, univerzitet u Kembridžu je najavio kupovinu 1000 modela iz serije 400, stari Master 128 i dalje dominira britanskim školama... Ipak, Acorn je nedavno ostao bez izvršnog direktora, otpustio 49 službenika i „privremeno“ odložio razvoj Arc PC pločice koja je Arhimeda trebalo da učini PC kompatibilnim. Zašto? Probleme, po svemu sudeći, pravi Olivetti!

Pokazalo se da se interesi italijanskog giganta koji drži većinu Acornovih akcija nekoliko kose sa dajim razvojem „arhimeda“ i da bi Olivetti želeo da strategiju svog razvoja gradi na intelovim procesorima, ne „cepajući“ tržište na RISC i CISC. Zanimljivo je pitanje kakvu će politiku voditi novi Acornov direktor Harvey Coleman koji je do skora radio u britanskoj propagandnoj ispostavi Olivettija.



Džek Trbosek i dame

Kada već pominjemo igre pune nasilja, ne smemo da zaboravimo na britansku Školu za dame (St Brides School for Ladies), čije se polaznice sve češće probijaju na softversko tržište. Novi hit je igra „Džek Trbosek“ (Jack the Ripper) koja, po mišljenju britanskih cenzora, „sadržji previše nasilničkih scena da bi bila prikladna za mlade od 18 godina“. Pa neka neko posle govori o „nažnjen polu“!

Čik pogodi

U gužvi oko jednog piratskog štanda na „Yu video šou“ čujemo dva hekeka kako međuju:

— Zašto „Računari“ stalno objavljuju ženske na naslovnim stranim — čudi se jedan.

— Zato što svaki haker ima računar, a žensku smo po neki — odgovara drugi

Tuc muc

— Sedeo sam pred tim čovekom i desetak minuta lupetao na neuvezbanom engleskom sve što mi je palo na pamet — pričao nam je Miodrag Potkonjak prilikom nedavne posete redakciji o svom prijemnom ispitu na Berkliju. — Nakon desetak minuta Ričard Karp je bio ubeđen da pred sobom ima totalnog idiota.

Američki profesori su, međutim, izgleda soj i ne predaju se tako lako. Karp nije poverovao sopstvenim ušima i dao je Miodragu teško rešiv zadatak da ga „reši kad može“. Miodrag je zadatak rešio za jedan dan. Iznenadeni profesor mu je odmah dao drugi, a potom i treći, ovoga puta takav koji ni sam nije znao da reši. Ostalo je istorija o kojoj Miodrag iz skromnosti ne želi da priča.

Bilo bi lepo kada bi Miodrag mogao da nam prenese barem malo atmosfere sa Berklija, ali on to ne stiče — njegov profesor obzavda da zadaje domaće zadatke, a Miodrag ih sa zadovoljstvom rešava.

Kockica je bačena



Dragan Mošulović (sed) i Srđan Mijanović na zimskoj školi Informatike u Petnici

Kao što seljaci u duši nikada ne izlaze iz svojih opnaka, tako ni hakeri koji su barem jednom osetili slast programiranja procesora Z-80 ne mogu da zaborave mašinu na kojoj su odrasli. Njih čete poznati po tome što od svojih skupocenih PC-ja i „amiga“ prave nikad prežajene „spektrume“, „galaksije“ i „amstrade“. Za ovakav podvig je dovoljno ugraditi u mašinu koprocesor i prebaciti u njega nešto osnovnog softvera i vaš AT će se nepogrešivo buditi sa porukom „Sinclair Research Ltd. 1982.“ ili „Hi Soft GENS 3M21 1983.“

Hakcerski svet je, međutim, prepun iznenađenja. U zimskoj školi informatike u Petnici upoznali smo momka kome jedan koprocesor Z-80 na PC-ju nije bio dovoljan, pa je projektovao hiperkockicu sa — 8-

snaset procesora Z-80 u paraleli! Ovaj poduhvat srednjoškola Srđana Mijanovića iz Zrenjanina, pored sentimentalne, ima, naravno, i istraživačku vrednost — zadatak projekta je da se ispituju potencijali paralelne arhitekture sa (sasvim primereno srednjoškolskom džepu) jeftinim konfiguracijama. Srđanu, koji je projektovao hardver i napisao sistemski softver, pomaže Dragan Mašulović, srednjoškolac iz Novog Sada. Dragan je napisao kompjoler posebnog programskog jezika, zasnovanog na fortu, za komunikaciju sa mašinom.

Otkuda Srđanu ovakve „prozapadnjačke“ računarske ideje? Verovatno otuda što živi u poznatom zrenjaninskom naselju Ma-la Amerika.

Zagrebačka veza



Ojla Lazarević — Javorina

— Originalni programi se nabavljaju uglavnom iz SR Nemačke i Engleske i stižu u Zagreb, gde se, izgleda, nalazi „centrala“ za celu Jugoslaviju — kaže sociolog Ojla Lazarević-Javorina (na slici) koja poslednjih meseci istražuje sociologiju piratstva. — Odatle se distribuiraju raznorodnim potrozičima, najčešće zaposlenim ljudima — inženjerima, ekonomistima i tehničarima, kojima programi veoma korisno služe, prvenstveno zbog prirode poslova koje obavljaju. Iako mlade ljude zanimaju pretežno igre, poslovni programi su ipak najtraženiji. Veoma dobro se prodaju i programi za studente, kao i različiti programi za magistarske radove. Cena po jednom programu je minimum 5.000 dinara, a za programski jezik 15.000 dinara.

— Ova pojava jednom sociologu ne bi bila naročito interesantna da, kao i u svakom sumnjivom poslu, ne postoje „krupne“ i „sitne“ ribe — kaže Ojla. — U piratstvu ove „sitne ribe“ ne bivaju uvek pojedene, nego nekako opstaju sa svojim sitnim interesima. Pošto „krupne ribe“ ugrabe svoj veliki zaloga, sitni preprodavci, koji i nisu „pirati“ u pravom smislu te reči, nude programe jeftino, uglavnom bez uputstava. Većina njih zapravo i ne zna čemu služe uputstva, nego ih samo u cilju zarade, omogućava i preprodaje. Zbog velike količine novca koji se nalazi u igri, preprodaja softvera nepovratno gubi svoje druge strane i polako prerasta u pravi kriminal — organizaciju velikih bosova i njihovih malih podanika.

Za vlasnike „Komodora“ i BBC-ja

Čitaocima „Galaksije“ i „Računara“ već smo nekoliko puta predstavljali Klub programera Elektrotehničkog fakulteta u Beogradu, po svemu suđeci programerski klub sa najdužom tradicijom u zemlji. Smešno bi bilo reći da kroz godine koje čine tu tradiciju Klub nije imao razne probleme — najveći problem je sekcija vlasnika „komodora 64“ koja godinama nije uspevala da organizuje rad koji bi zadovoljio članove. Znao se, čak, događati da Klub članovima ove sekcije ponudi besplatnu članarinu i varljivoj nadi da će sledeće godine sve biti bolje.

Početak 1988. C sekcija je, zahvaljujući svom novom koordinatoru Bojanu Zanoškuru, konačno proradila — formiran je veoma ozbiljan i opširan katalog komercijalnih programa i literature koji, po svemu suđeci, predstavlja najpotpuniju domaću zbirku „ozbiljnog“ softvera za C-64. Što se prodaje igra na kasetama tiče, Klub ne pokušava da konkurise piratima. Ostale promene se odnose na BBC sekciju koja je, takođe pod novim rukovodstvom, počela da prodaje programe i literaturu (na raspolaganju je praktično sav ione vredan softver koji se prodaje u Engleskoj), doduše po prilično visokim cenama.

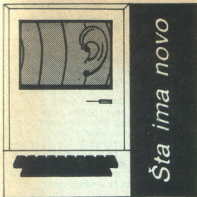
Najjednostavniji način da se učlanite u Klub je da u koverat ubacite drugi na sebe adresirani koverat sa potrebnim markama i da sve to pošaljete na adresu Klub programera, PKK SSO ETF, Bulevar Revolucije 73, Beograd. U prapratnom pismu možete da pomenete računar koji posedujete.

TIM bolji

Razvoj domaćeg računara se, rekostmo, nikada ne završava — konstruktori školskog računara TIM 011 su, u saradnji sa institutom „Mihajlo Pupin“, ispravili dobar deo zamerki koje smo uputili u prošlim „Računarima“!

Rešeni su, pre svega, razni hardverski problemčići koje je izazvao prekidaički PSU: slika je savršeno stabilna, čak i dok računar intenzivno opti sa diskom. Softverska poboljšanja su mnogo značajnija: TIM 011 je dobio komandni editor, program DISKCOPY, imenovane direktorijume, komande za redirekciju i još poneku sitnicu koja uključuje i bržu komunikaciju sa diskom. Što se tastature tiče, „Pupin“ je delimično usvojio sugestije javnog mnjenja: iveri SHIFT i DELETE su povećani, „mrtvi“ tasteri ukinuti a kutija zakašena tako da ne smeta pri umešanju disketa. Dalje izmene rasporeda tastera lome se o stenu zvanu YUSCII standard; kada sledeći put budete gledali zavrnutu scenu „Brijantina“...

Prva serija TIM-ova će se, kada ovi „Računari“ budu u vašim rukama, najverovatnije već nalaziti u ulčionicama — isporuča je planirano za 15. februar. Uz računare se isporučivati dve diskete (od kojih je jedna prazna) i relativno kratko (pedesetak strana) uputstvo za upotrebu. O softveru koji će biti upisan na drugu (bolje reći prvu) disketu pišaćemo u sledećim „Računarima“. Možda vas tamo dočeka još jedno iznenađenje...



Šta ima novo

Borland ide u školu

Obrazovne ustanove koje školuju programere su izvanredno tržište za razne interpretatore, kompajlere i debagere. Shvatajući to, Borland je ponudio specijalni popust za škole: Turbo Pascal, Turbo C i Turbo Prolog se, na primer, prodaju za po 30 umesto za 100 funtil Škola koja se interesuje za ovaj popust treba da potpiše specijalni sporazum sa Borlandom precizirajući broj nastavnika i učenika koji će koristiti pojedine pakete.

Zli jezici bi rekli da domaće škole (koje su se nekada opremile PC klonovima) kompajlere i interpreatore dobijaju sa još većim popustom i to bez ikakvog ugovora!

Za ljubitelje TSR programa

TSR program je, za neupućene, program čiji je deo neprekidno u memoriji i koji je, prema tome, stalno pristupačan. Mnoge softverske kuće povela su se za Borlandovim primerima i tako je nastalo mnoštvo rezidentnih programa za koje se ne bi uvek moglo reći da međusobno saraduju. Rezidentni programi, osim toga, oduzimaju određenu količinu memorije — nije redak slučaj da korisnik mora da menja AUTOHEC-BAT proceduru i resetuje računar da bi učitao neki komplikovaniji paket!

Jedno od rešenja je *Fxtra Memory Manager* firme *Delta Technology* (američki telefon (715) 8327575). Svi rezidentni programi se smeštaju u virtuelnu memoriju (naročito zgodno za vlasnike AT-a opremljene megabajtom RAM-a) i pozivaju po potrebi. Glavni RAM je, dakle, savsimo slobodan? Skoro — u njega je upisan *Fxtra Memory Manager* koji je, pogađate, i sam TSR program!

Za šaku disketa

Uz priču o „PI Editoru“ i AutoCAD-u pomenućemo jedan razgovor sa Kenom Mitchellom koji je sredinom maja 1987. borio u Jugoslaviji i ugovorio pisanje programa. Kena je interesovalo koliko se u Jugoslaviji koriste specijalizovani programi poput AutoCAD-a. Dobio je odgovor nalik na „Pa, video sam AutoCAD, ali nisam smatrao da ga vredi imati“. Ken je tužno klimnuo glavom, priznajući da je nekih 700 dolara zaista previše za nekoga ko živi u jednoj siromašnoj zemlji i ko nije siguran da mu je AutoCAD zaista potreban. Odgovor „AutoCAD za mene ne košta 700 dolara; pitam se da li vredi pet praznih disketa“ trebalo je dugo obrazlagati!

6 računari 36 • mart 1988.



Softver za „arhimeda“

Na britanskom tržištu je sve više programa za „najbrži mikro na svetu“ — posebno se trudi firma *Computer Concepts* koja, uz Inter seriju, prodaje i razna hardverska proširenja koja omogućavaju da sve potrebne programe držiže u EPROM-ima.

Naročitu pažnju i dalje izaziva *PC Emulator* uključen u osnovnu cenu „arhimeda“. Testovi su pokazali visok stepen kompatibilnosti, ali se brzina baš nije proslavila — „najbrži mikro na svetu“ jedva dostiže običan PC na 4.77 MHz!



Bankova izjava

Kako da postanete dobar novinar? Kažite šta mislite i ne brinite kome ćete se zameriti, kaže Martin Banks, koji je po drugi put proglašen za najboljeg „tehničkog novinara“ u Britaniji. Bankova rubrika *Banks Statement* koju redovno objavljuje najstariji evropski kompjuterski časopis *Personal Computer World* je zaista najutitila mnoge proizvođače i korisnike razne opreme (dokaz? Pregledajte pisma čitalaca), ali se u ogromnom broju slučajeva pokazalo da su njegovi prikazi apsolutno objektivni a kritike opravdane.

Domaći program u SAD

Čak i veoma pažljivi čitaoci „Bejta“ verovatno nisu primetili program „PI Editor“ koji za 150 dolara prodaje *Programmer's Paradise*. Ne bismo ga, naravno, primetili ni mi da se ne radi o delu domaćih programera!

„PI Editor“ je, pre svega, *Language Sensitive* editor namenjen isključivo programerima. Po ekstenziji datoteka program prepoznaje korišćeni jezik (tekuću verziju čine preprocessori za bejzik, fortran, turbo pascal, C i assembler) i u toku rada upozorava programera na sve očite greške pomažući mu istovremeno da uklopi kontrolne strukture i, uopšte, osveži poznavanje sintakse određenog jezika. „PI Editor“, uz to, nudi istovremeno ispravljanje nekoliko programa u raznim prozorima, makro naredbe sa parametrima, automatsko prilagođavanje raznim konfiguracijama... Za kraj smo ostavili najlepše: komande editora su uglavnom zasnovane na VAX-ovom EDT-u koji je, po mišljenju mnogih korisnika, gotovo idealan editor za programere.

Iako dvojica autora „PI Editor“ iz raznih čudnih razloga ne žele da se predstave javnosti i da izlože finansijsku konstrukciju čitavog projekta, nadamo se da će, ako već piratujete program, izbegavati slanje njegovih kopija u inostranstvo!

Letraset softver

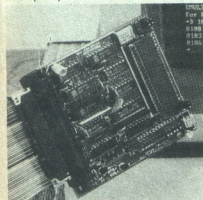
U proizvodnju Desktop-Publishing softvera upustila se i tvrtka Letraset poznatija po produkciji preslikača slova, brojeva, raznih znakova, simbola i ostale grafike. Program „Ready, Set, Go! 4.0“ predstavlja cjelovit program za stono izdavaštvo. Za sada se može nabaviti za računalo „mekintosh“ po ceni 1750 DM. I po ceni i po kvaliteti pogodan je za urede i privatno izdavaštvo koje kod nas uzima sve više maha. Stručnjaci grafičari koji su imali prilike raditi s tim programom kažu da se to trenutno najbolji odnos kvaliteta cijena koji se može naći na tržištu, te da grafičke mogućnosti znatno premašuju cijenu. Više informacija sa adrese: Letraset Deutschland GmbH, Mergenthalerstrasse 6, 6000 Frankfurt am Main 60. (Z. V.)

Ada na PC računaru

Programski jezik qda dostupan je sada i vlasnicima PC računara. Ada-Compiler u PC verziji tvrtke Meridian Software Systems nosi naziv AdaVantage. Programski ada paket zauzima 1,5 MB na disku. Cijena od 2850 DM je dosta visoka, ali ozbiljnijim korisnicima će paket itekako biti koristan, pa će naći načina da ga nabave. Zainteresiranima više informacija stoji na raspodisnu na adresi: ICS Information and Communication Systems Marketing GmbH, Kronbergerstrasse, 27, 6000 Frankfurt 1, BRD. (Z. V.)

Podaci sa PC-a na PC

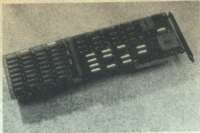
Mnogi računarski klubovi, kao i neke škole koje su nabavile računala klase PC; muku muče sa međusobnim prijenosom informacija sa računala PC na drugo PC računalo. Veliko olakšanje, a relativno i jeftino, (400 DM) predstavlja softverski paket pomoću kojeg se podaci mogu prenositi serijalnim ili paralelnim interfejsom brzinom od 115 kilobauda. Program, nazvan Direct-link, pisan je u assembleru. Odgovara za upotrebu kod varijanti PC/XT ili AT sa serijskim interfejsom, 256 K RAM-a i DOS verzijom preko 2.0. Moguće je nabaviti i varijante ovog softverskog paketa. Adresa: H+B EDV, Ölgrasstrasse 4,7992 Tettang 1, BRD. (Z. V.)



Emulator za 8051

Hardverarši nam često zameraju što se među vestima ne nađe i neki uređaj neophodan onima koji se bave razvojem mikroprocesorske opreme. Evo jednog: EMUL31-PC švedske firme Nohau Elektronik (tel. (040) 922425) emulira Intelove mikroprocesore 8031, 8032, 8051, 8052, 8344, 80535, 80C152, 80C252, 80C452 i 80C451; sam emulator ima 64 kilobajta RAM-a i specijalni trace bater od 16 kilobajta. Kontrolishe ga, jasno, bilo koji IBM PC, XT ili AT kompatibilac, dok se programiranje može vršiti u assembleru, PL/M-u 51, paskalu 51 ili C-u 51. Cena od 2700 dolara nije baš niska ali nikada ne treba porediti cenu amaterske i poluamaterske opreme sa profesionalnim napravama.

Svi pomenuti Intelovi mikroprocesori ugrađuju se u razne kontrolere i komunikacione adaptere.



Sa PC-ja na 386

U prošlim „Računarima“ pomenuli smo PC Elevator koji vlasnicima PC-ja ili AT-a omogućava da uživaju u blagodetima procesora 80386. Tu je i konkurentna: Master 386 firme Aox (američki telefon (617) 8904402) je pločica zasnovana na 80386 (16 ili 20 MHz) i opremljena RAM-om od 2—10 megabajta. Cena se, zavisno od konfiguracije, kreće između 1600 i 2200 dolara.

Treća mogućnost je 386C-PAC firme Specialix Systems Ltd (britanski telefon (01) 3989422). Karakteristike su sasvim slične a ni cena od 1995 funti nije baš neuskadna.

Sa „komodora“ na PC

„Menjam komodor za PC-ja uz doplatu“. Ako vam sitnica zvana doplata otežava davanje ovakvog oglasa, firma Financial System Software (Anbrlan House, St Mary's Street, Worcester WR1 1HA, England) će vam pomoći da napravite prvi korak: Sogwap Big Blue Reader omogućava „komodoru 128“ da čita i snima podatke u formatu kompatibilnom sa PC-jem. Da li program radi na običnom C-64? Taman poslat!

„Elita“ PC-ju

Sećate li se „Elite“, igre koja je proslavila Acornove računare i koja je dočimje prilagođena „amstradu“. „Komodoru 64“ pa čak i „spektrumu“? „Elita“ je od skora na raspolaganju i vlasnicima Amstradovog PC-ja. Iako je igra veoma dobro realizovana i brza, mnogima smeta potpuno „gušenje“ rezidentnih programa — čak ni CTRL ALT DEL ne resetuje računar! Odgovor firme Firebird da nije bilo dovoljno memorije ne treba uzimati ozbiljno: zar igra koja se izvršavala u BBC-jevih 32 K RAM-a više nije dovoljno 512 K?

Ako se pokaže da Elita radi i na drugim PC kompatibilcima i da rejtniji snimljeni na BBC-ju važe i na PC-ju, setičemo se gde smo stali sa „Elitom“ — galaksica 3, plane-tee Raceedit!

„Varvarin“ nepoželjan u Nemačkoj

Kompjuterska igra Barbarian koju smo pominjali u nekom od prethodnih brojeva „Računara“ je u Nemačkoj proglašena za „nepoželjnu“. Razlog nije omet koji ... afirmishe žensko telo kao predmet koji muškari mogu po volji koristiti! već previse nasilja. Varvarin, dakle, može da se prodaje samo u nemačkim seks-šopovima.

Rotringov cad

Program RDS-M Assistant proizvod je tvrtke Rotring, svjetski poznate po crtačem priboru. Program se isporučuje na 5 diske-ta, a odgovara za upotrebu pod MS-DOS operativnim sistemom, kao i za PC računare. Za rad je potreban PC sa 512 K glavne memorije, te EGA ili Hercules grafička kartica. Program omogućuje upotrebu najraznovrsnijih simbola kao i izradu kompletnih konstrukcijskih crteža. Uz program dolazi i priručnik od 150 stranica. Paket cijeni nije dostupan amaterima (cca 35000 DM), ali je itekako pogodan za manje konstrukcijske biroe, koji i kod nas sve više koriste računarske CAD sisteme. Informacije: Rotring euroCAD GmbH, Haferweg 46,2000 Hamburg 50, BRD. (Z. V.)

Nova akcija

U saradnji časopisa „Računari“ i Instituta Mihajlo Pupin

Tim 011 u samogradnji

Detaljnija obaveštenja u sledećem broju

MC 68000 na pločici

U stara dobra vremena kontroleri su najčešće zasnovani na mikroprocesorima Z-80, 8080 ili 6502. Nova dobra vremena donose nove standarde — sve su češći i jeftiniji single board kompjuteri zasnovani na Motorolinom mikroprocesoru 68000. Jednu ovakvu pločicu za samo 250 dolara prodaje firma Marion Systems (američki telefon (213) 451-8910).

MS68K je pločica 133x203 mm opremljena mikroprocesorom (radi na 8 MHz), RAM-om od 512 K, serijskim i SCSI portovima i podnožjima za EPROM-e od 128 K. Da bi pločica postala kompjuter, treba je dopuniti izvorom za napajanje i nekakvim terminalom. MS68K će, međutim, mnogo češće raditi bez ikakvog terminala, kontrolišući neki industrijski proces.

Za kompjuterske tekst poliglote

Poliglota je, sećamo se, čovek koji govori više jezika. Kompjuterski poliglota je čovek koji koristi više programskih jezika. A kompjuterski tekst poliglota? Naravno, onaj ko koristi više tekst procesora.

Svako od nas ponekad mora da koristi razne tekst procesore — neki editor je jednostavno zgodniji za neku primenu! Konverzija tekstova pisanih posredstvom jednog editora u format nekog drugog često je prava mora — pročitahe napis o stonim izdavaštvu u ovim „Računarima“. Word-by-Word firme Corporate Software (britanski telefon 07357 5361) obezbeđuje konverziju između formata WordStar, WordPerfect, Word, PFS Write, PC Writer, Volkswriter, MillitMate, Ventura ... Zgodan način da uložite 150 dolara!

računari 36 • mart 1988. 7

Računarske zvezde San Franciska

Kada su u decembru 1987. godine birali gradonačelnika u San Francisku, svi kandidati su budućnost gradstva u predizbornim kampanjama videli u turizmu (prihod San Franciska od turizma je tri puta veći od Jugoslovenskog), finansijskim transakcijama oko osiguranja, vrhunskoj elektronskoj i bio tehnologiji i, verovali ili ne, na prvom mestu, provoklasnom školstvu. Smatra se da grad nema budućnosti bez prestiža Stenforda i Berklija.

Nije lako biti zvezda na Stenfordu. Pogotovu ne na Odeku za računarstvo, koji je u službenoj proceni američkog NSF (koji odlučuje o finansiranju američke nauke) proglašen za najbolji odevak ne samo u računarstvu nego uopšte u SAD. (Prosečna ocena profesorskog kadra u škali od 0 do 5 je 5.00.) Tu radi i Donald Knut, čija je siva eminencija „Umetnost programiranja“ istovremeno biblija, manifest i Wall Street Journal računarstva. Četrdeset odsto radova objavljenih iz oblasti algoritama u vodećim svetskim časopisima citira tu knjigu. Što neko reče: ako hoće da ti rad bude ozbiljan, stavi Knuta u literaturu. Knjiga se, uzgred budi rečeno, prodaje i posle 20 godina od prvog izdanja, koje je štampano na posebnom papiru, kao i sva ostala na engleskom jeziku, koji omogućava viševoknu nepromenljivost, u mesečnom tiražu od 2.000 primeraka samo na engleskom jeziku. U SSSR-u je prodato preko 100.000 primeraka na ruskom jeziku. U mnogim drugim jezicima tiražu su basnoslovno za knjigu tako visokog nivoa. Ali što reče Knut: mnogi je kupuju, ali nisam siguran da je baš svi kupci čitaju.

Inače, Knut je knjigu napisao dok je kao student postdiplomskih studija matematike na Kalfornijskom Tehnološkom Institutu pišući tezu o projekktivnim geometrijama rešo da zaradi neku paru. Da ironija bude veća, Knut je uspeo da nađe primenu u računarstvu za skoro sve oblasti matematike, ali ne i za projekтивne geometrije. O Knutovim zaslugama u oblasti digitalne tipografije i računarskih koreliranih problema ne treba ni pričati. (Iako ponekad izgleda suludo provesti šest meseci istraživačkog rada da bi se dizajnirao oblik slova S, kad to radi Knut to je vrhunska nauka.)

3.500 doktora nauka

Čovek sa možda najveće objavljenih algoritama je Floyd. Ko nije čuo za Ulmana i njegove knjige o kompajlerima, VLSI, algoritima i formalnim jezicima? Valjda jedina knjiga koja može da se iole ozbiljno mari sa Knutovom „Umetnošću programiranja“ po citiranju, dugovečnosti i uticaju na razvoj računarstva je „Kreiranje i analiza računarskih algoritama Ahoa, Ulmana i Hopkrofta.

U častom teorijskom računarstvu glavnu reč vode Mana i Valding. O projektima i zvezdama veštačke inteligencije na Stenfordu ne treba ni govoriti: u vrhunskim 8 računari 36 • mart 1988.

časopisima i na najboljim konferencijama iz oblasti veštačke inteligencije više od polovine autora je sa Stenforda, ako pod Stenford računamo i SRI. Papadimitrou je istovremeno jedan od najboljih u paralelizmu i u operacionim istraživanjima i u teorijskom računarstvu i ko zna u čemu još. Henesi i njegova MIPS grupa i kompanija su jedini ozbiljni konkurent RISC-u na Berkliju.

Na ulazu na Stenford stoji tabla: „Ulazite u opštinu koji ima 3.500 doktora nauka“. Godina bez Nobelove nagrade na Stenfordu nije godina. Za one koji misle da se u računarstvu ne može dobiti Nobelova nagrada, moramo reći da greše. Dobio ju je svojevremeno Sortiaf za medicinski program. Sortiaf je, naravno, profesor na Stenfordu.

Najbolji profesor na svetu

Nije, međutim, lako ni biti zvezda na Berkliju: 14 nobelovaca, stotinak akademika, i bezbroj nosilaca raznih nagrada, uz službeni naziv NSF kao najbolje škole za postdiplomsku studiju i istraživanja obezbeđuju Berkliju prestizno mesto među svetskim univerzitetima.

Možda je još teže biti zvezda na Odeku za elektrotehniku i računarstvo. Ričard Karp je dobio Tjuringovu nagradu za zasnivanje nekoliko oblasti, kao što su NP zadaci, probabilistički algoritmi, poliedarska kombinatorika. Da ne spominjemo neslužbeni naziv najboljeg profesora sveta u računarstvu. Prošle godine se 37 studenata oficijelno prijavilo da sluša kurs kod Karpa. Na prvom času njih nekoliko stotina. Zapranešeni Karp pita: „Koliko od vas sluša oficijelno kurs?“. Otrpilike svaki osmi student diže ruku. Na pitanje šta je sa ostalima niko ne odgovara. Na kraju se jedan od studenata osmeli i kaže: „Vi ste toliko slavni da ja nisam mogao da propustim priliku da odušljam vaš kurs.“ Tada je Karp posvetio prvi čas objašnjenjima zašto studenti ne treba da slušaju njegovu predavačnja. I to je bilo prvi put da nije uspeo da uveri svoje studente u ono što je izlagao. Uzgred budi rečeno, Karp je na glasu da daje ogromnu količinu domaćih zadataka i to je razlog što više studenata nije oficijelno upisalo njegov kurs.

U kriptografiji nema većeg autoriteta od Manuela Blama. Isto važi i za random generatore. Lovler je neprikosnoveni autoritet u primenjenoj kombinatorici, a Barlekamp za kodove. U bazama podataka pro-

jektu tipa INGRES i POSTGRES obezbedili su počasno mesto Vongu, Stounbrjekeru i Roveu. Ferari i Anderson, posle svetskog standarda za juniks (BSD UNIX), nadiru sa distribuiranim operativnim sistemima.

Zašto Japance boli glava

Deespejn i Pat su konstruisali prolog mašinu od koje Japance boli glava. Brza je od njihovih najbržih mašina za dva reda veličine. Paterson i Sekvin posle RISC-a spremaju nova iznenađenja na polju mikroprogramiranja. Zadeh je, pored osnivanja teorije fazi sistema, veoma uticajan u veštačkoj inteligenciji, posebno u oblastima analize teksta i prevodjenja. U formalnim jezicima i kompajlerima „istorijskim veličinama“ kao što su Gil, Harison i Suzan Grejem u profesorskom kadru se ističe Hilfinger, jedno od vodećih imena u funkcionalnim jezicima i okruženjima.

Njutn, Sandjovani, Meesermit i novopridošli direktor IBM Tomas Vukson centra za algoritme Brajton, pored buvalno stotina radova u najboljim svetskim časopisima, svojim projektima u CAD-u učinili su da se tu škole tipa Karnegi-Melton, Urbana-Sampanj i Masacuetski Tehnološki Institut osećaju bezmalo inferiornim.

Brodersen, pored službenog priznanja za najbolje radove u oblasti CAD-a, ima i neslužbena priznanja za najbolje vodene (i finansirane, što je u Americi korelisano) projekte. Među njegovim projektima ističe se dvodnevni VLSI. Potpuno nezadovoljan brzinom izgradnje svojih računara od dva meseca (računajući vreme od trenutka kreiranja nacrti čipa do završenog proizvoda) sa svojim grupom je skovao plan za dvodnevni završetak. (Zašto dvodnevni, a ne jednodnevni? Razlog je trivijalan: potrebna mi je noć da bi mreža sastavljena od par „krojova“, „koneksne mašina“, i PSC superkompjutera i stotinak VAX-ova (sa akcentom na VAX-ove 8800) i brdo specijalizovanog hardvera naročito konstruisanog na Berkliju izoptimizirala hardver, arhitekturu i softver novog kompjutera.)

Svi komercijalni projekti iz CAD koji nešto znače u svetu počivaju na programu SPICE koje sveto nešto više od desetak godina uradila grupa studenata kao seminarski rad. Mlade zvezde nadiru u svim oblastima: Vaziriani i Zajdel u algoritmima, odnosno računarskoj geometriji, ... haj i LI u specijalizovanim arhitekturama. Ko u senzorima, Osterhaut u sistemskom programi-



ranju, Vilenski u veštačkoj inteligenciji. I dok je Stenford u proceni NSF najbolji po profesorima u računarstvu, Berkli u istoj proceni ima najbolje studente elektrotehnike u SAD.

Mnogo bi se još moglo pisati o stotinak profesora UC Berkli odeska za računarstvo i elektrotehniku i pedesetak Stenford odeska za računarstvo, ne pominjući Stenford odesk za elektrotehniku i zvezde tipa Helman (začetnik kriptologije) i Kajlat (čovek koji je sveo čuvene sistolike na specijalni slučaj klase arhitekture). Ali kad upitate nekog na ovim školama ko je stvarno postigao uspeh, na Stenfordu i, posebno na Berkliju, će vam dati zaista neočekivan odgovor: jedan student redovnih studija, i to koji je poduze studirao, i jedan propali student postdiplomskih studija. Dok je kod Stiva Vozniaka objašnjenje za popularnost jednostavno i krije se u milionima dolara koje je poklonio svom odesku, koji ga je toliko namučio i zadužio, slučaj Bila Djoja i njegovog kompanjona Endi Behtolshajma zaslužuje posebnu priču. (Naravno, Pakard, trećeg čoveka po bogatstvu u SAD, sa 3 milijarde dolara i sa reputacijom čoveka koji je počeo seriju više desetomilionskih poklona univerzitetima, darujući 50 miliona dolara svome Stenfordu, kao pripadnika stare garde nećemo ni spominjati.)

Sanjarenje Bila Djoja

Omiljeni hobi Bila Djoja je sanjarenje. A tema računari. Posebno računar 2001 godine. Kao, on će biti 128.000 puta brži od

današnjih, imaće memoriju dovoljnu da pohrani biblioteku od nekoliko stotina hiljada knjiga i biće veličine kutije šibica. Naravno, reč je o personalcu. Ne verujete? Mnogi nisu verovali Bilu Djoju kada je napustio doktorske studije na Berkliju da bi osnovao svoju kompaniju. Danas „Sun Microsystems“ sa milijardu dolara godišnjeg prometa i pozicijom broj jedan u prodaji radnih stanica predstavlja odgovor nevernim Tomama. A on im je sve lepo još na početku objasnio uz pomoć svog, Djojovog, zakona: „Brzina računara se svake godine duplo uvećava. Ko ne može da prati taj trend, propada. Ko je brži od tog trenda, uspeva.“ I dok većina kompanija pokušava da kreira zatvorene standarde, SUN se koncentriše umesto na zaštitu postojećih na razvoj nove tehnologije! „Sunov“ pristup je jednostavan: kreiran je otvoren standard, koji svaka kompanija može slobodno da prihvati, oni ga čak reklamiraju, tako da su mečerni AT&T i Xerox odučili da mu se priklone.

I dok su za Regana mračne sile Rusi, Kubanci i Nikaragvanci, za Djoja su to IBM, DEC i Apple. Oni su ti koji koče progres. I dok bi mnogi poželeli takve kočilače, Djoj čini sve da SUN izbegne njihovu sudbinu. SUN vodi korene iz kompjutera koji je kao dev studentskog istraživačkog projekta na Stenfordu konstruisao Endi Behtolshajm. Kompanija je uspeła da napravi pare zahvaljujući računaru koji je konstruisan za naučnike i inženjere, a zasnovan na industrijskim standardima: mikroprocesorima Motorola 68.000, Eternet lokalnoj mreži i

operativnom sistemu UNIX. Kombinujući legendarni kompjuterski konstruktorski talent Stiva Vozniaka, poduhvatničke sposobnosti Stiva Džobsa i širinu vizije Alana Keja, Djoj je tiho pretvarao svoj san u stvarnost.

Otvoreni procesori

Djojovo radno mesto oficijelno nosi naziv „zamenik predsednika za istraživanje i razvoj.“ Međutim, on ne prima nikakve izveštaje i ne predaje ih. Umesto toga, on se šeta kroz kancelarije i radi sve: od prepravljavanja i testiranja softvera, do primanja poslovnih partnera. Njegov najveći talent je, bar po rečima njegovih saradnika, u širini vizije. On ima izuzetnu moć da vidi ulogu svakog detalja u celokupnom projektu i da istovremeno realizuje svaki od kritičnih detalja. I pored toga, on važi za jednog od retkih ljudi koji ne misle u računarskim terminima. Ili, kakor sa kaže Donald Paterson, njegov profesor sa Berklija: „Ako se ikad bude vratilo da završi svoje doktorske studije, to neće biti u računarstvu, već u engleskoj literaturi.“ I nastavlja: „Djoj je jedan od najboljih ljudi koji su se ikada upisali na postdiplomsku studiju računarstva na Berkliju. Za vreme doktorskog ispita Djoj se bezmalo posvadio sa jednim od profesora oko jednog pitanja. Niko u komisiji nije znao tačan odgovor. Članovi komisije su se složili: Ako je Djoj u pravu dobiće čistu desetku, a ako ne devet.“ Djoj je dobio desetku. Ali nije to sve čime je Djoj zaslužio slavu na Berkliju. Sa kolegama sa kursa on je detaljno prekonstruisao UNIX i ponudio ga po ceni od 150 dolara svakom ko je zainteresovan. A takvih je bilo 100.000.

Godina 1987. bila je po rečima Bila Djoja, posebno opasna za SUN. Dostigli su milijardu dolara prometa, a to je trenutak kada su nekada lokomotivno računarskog razvoja, kao DEC i Data General, postale suviše velike da bi dalje bile progresivne. Ali odgovor na tu opasnost u SUN-u je stvarno bio neočitan. Dok drugi proizvođači baziraju svoje kompjutere na mikroprocesorima Intela i Motorole, snagu novim SUN-ovima davaće sopstveni čip SPARC. SPARC je skraćenica za Scalable Processor Architecture. Po opštem mišljenju, to je prvi kočkarski potez, ali bar dve stvari daju dobre izgleda da uspe. Prvo, pošto je „scalable“, čip će vremenom postati sve brži i brži; i drugo, SPARC će biti dostupan svim proizvođačima čipova koji žele da proizvode svoje sopstvene verzije. To je do sada bilo nezamislivo. Ali u SUN smatraju da šansa da SPARC postane prvi otvoreni standard vredi više od bilo kog uloga. Do sada su tri kompanije potpisale ugovore za izgradnju čipa. Sve je ovo imalo takvog odjeka da je MIPS, kompanija čuvenog profesora Henesija, koju inače finansiraju Japanci, povukla potpuno isti potez i ubedila tri druge kompanije, uključujući i LSI Technology, da joj pristupe u njihovom otvorenom standardu.

Ali sve to ne uzbuđuje previše Bila Djoja. „Sve što interesuje Djoja je sanjarenje“, kaže jedan njegov kolega. A on sada ima milijardu dolara vrednu kompaniju kao osnovu za svoje snove.“

Nada Aleksić

Elektronski rokovnik

Živeći već desetak godina u svetu personalnih računara, ljudi su osmislili raznorazne primene ovih „inteligentnih“ naprava — nekih od njih se, naravno, nije bilo lako setiti. Od samog početka se, međutim, govorilo da će personalni ili prenosivi računar zameniti imenik, adresar i rokovnik, a ta se zamena nikako ne ostvaruje. Pojava nove generacije prenosivih računara, u kojoj „psion organajzer II“ zauzima počasno mesto, predstavlja korak ka ostvarenju dugog sna.

Jedan od programa koji se isporučuje na demo kaseti ili disketi svakog računara je, svakako, telefonski imenik — unosište imena, adrese i brojeve telefona vaših prijatelja i saradnika, a onda ta imena po želji pronalazite, sortirate i ispisujete. Pošto se gašenjem računara podaci upisani u RAM nepovratno uništavaju, treba ih snimiti na neki medij spoljne memorije kao što je kasetna ili disketa. Kada vam načiji telefon zatreba, treba da učitate program, a zatim i podatke što će, zavisno od brzine komunikacije sa kasetofonom, potrajati dobrih pet minuta. Nije mnogo bolje ni sa računarima koji koriste disk jedinice: treba, najpre, pronaći disketu sa programom, onda učitati program, pa pronaći disketu sa podacima, otkucati ime prijatelja koga tražite... Moderniji personalni računari su samo naoko pogodniji: podaci neće zauzimati preveliki deo vašeg hard diska, ali ćete po svakom uključivanju kompjutera čekati tridesetak sekundi da se disk zavrti! Uporediti ovo sa prostim pokretom ruke koja otvara „papirni“ telefonski imenik i oka koje za trenutak pronalazi potreban broj — računar očito nema mnogo šansi!

Beležnica ili elektronika

Slična je stvar i sa rokovnikom — svaki personalni računar lako može da prikuplja podatke o vašim sastancima, ali tako prikupljeni podaci ostaju skriveni dok je računar ugašen. Da biste se, osim toga, podsetili kada neki sastanak počinje, morate da se nađete u blizini računara, tj. u stanu ili kancelariji. Papirnatii rokovnik je, sa druge strane, uvek sa vama!

Prvi ozbiljniji pokušaj prevladavanja problema napravio je Tandy svojim modelom 100 iz serije TRS 80 (pogledajte „Računare 25“). TRS 100 staje u tašnu, ima profesionalnu tastaturu, ekran sa četrdeset slova u redu, dvadesetak kilobajta RAM-a čiji se sadržaj čuva i dok je računar isključen, časovnik realnog vremena i, što je možda najvažnije od svega, ROM sa ugrađenim bežikom, adresarom, rokovnikom i tekst procesorom; kvalitet ugrađenih programa, na žalost, nije baš visok. Osnovna mana TRS 100 je što ne „ume“ da vas upozori na sastanak koji se približava: ako ne pogledate u rokovnik, propustili ste sastanak! Slično se, istini za volju, ponaša i papirni rokovnik, ali bi se od kompjutera moglo



očekivati da unese bar nekakva poboljšanja.

Šinklerov Z-88 je, zapravo, poboljšani TRS-100: težina je smanjena, ekran povećan, softver doteran gotovo do savršenstva... Z-88 aktivno saraduje sa vama: bez vaše intervencije zvučno će vas upozoriti na svaki redovni ili specijalni posao koji treba da obavite. TRS-100 i Z-88, ipak, ne mogu da zamenju rokovnik: preveliki su i preteški. Svaki od ovih računara zauzima polovinu tašne, što znači da ga možete nositi samo ako nosite i tašnu u kojoj, uz to, neće biti previše mesta za druge papire. Papirni rokovnik, sa druge strane, možete da nosite u bilo kom džepu!

„Psion organajzer“ je naprava koja bi zaista trebala da pošalje rokovnike u istoriju; posebno je interesantna revidirana verzija uređaja nazvana „psion organajzer II“.

Mini tastatura...

Pogled na sliku pokazuje da je „psion organajzer II“, zapravo, naprava veličine prosečnog kalkulatora: debljine od 3 cm i težina od 250 g omogućavaju nošenje u džepu. Računar je snabdeven mini LCD ekranom koji u svakom od dva reda prikazuje po 16 znakova i specijalnom tastaturom koja se sastoji od 36 džirki.

Posle veličine, najveći problem svih prenosivih računara je tastatura. Što se „psion organajzer II“ tiče, kvalitet tastera je sasvim zadovoljavajući, ali je raspored prilično čudan: umesto standardne QWERTY konfiguracije, slova su poredana redom A, B, C... Y, Z. Ovakav raspored je dobro poznat vlasnicima starijih džepnih računara kao što je HP-41C — da bi kalkulator imao vertikalni dizajn koji omogućava da ga čvrsto držite u jednoj ruci dok drugom unosite tekst, redovi slova moraju da budu kratki. Ljudima koji su naviknuti na standardne tastature vertikalni dizajn značajno otežava unošenje teksta, ali treba znati da „psion organajzer II“ i nije računar kome je potrebno previše kucanja — unosićete uglavnom kratke notice. Olakšavajuća okolnost je mogućnost povezivanja sa „pravim“ računarom o kojoj ćemo tek govoriti.

Trideset šest tastera, jasno, nije dovoljno za sva slova, brojeke i specijalne znakove. Zato se cifre i interpunkcija dobijaju korišćenjem raznih SHIFT-ova u kombinaciji sa kursorskim tasterima — većina džirki ima po tri funkcije što svakako podesna na

starog dobrog „spektruma“. Smisao ostalih dirki odigledan je iz natpisa na njima: ON, MODE, DEL i EXE.

Što se ekrana tiče, dva reda od po 16 znakova ne zvuče previše raskošno, ali se dobro napisan softver svakim lepo analizirao sa njima — duže poruke pregledamo uz pomoć kurzorskih tastera.

... i maks RAM

RAM čipovi su sve manji i sve jeftiniji, pa džepni računari najzad mogu da imaju memoriju koje se do skora nije stideo ni neki „spektrum“, „komodor“ ili BBC: osnovna verzija „psion organajzera II“ (model XP) opremljena je RAM-om od 32 kilobajta, dok se priključivanjem posebnih modula radna memorija može proširiti do fantastičnih 280 K — više nego dovoljno da sakupite sva imena i adrese koje će vam ikada u životu zatrebati van kuće, a da ipak ostanete dovoljno prostora čak i za najpopuljniji rokovnik!

U RAM većine računara upisuju se programi. „Psion organajzer II“ zaista može da se programira na jeziku nazvanom OPL, koji je nastao kombinovanjem dobrih svojstava bejzika i C-a; programi se izvršavaju izuzetno brzo, na primer sedam puta brže nego na „spektrumu“! Znanje programiranja, naravno, nije neophodno za korišćenje „psion organajzera II“ — u ROM je upisan „kalkulator“, „adresar“ i „rokovnik“; posebno je zanimljiv rokovnik koji, uz standardne opcije, podržava dnevne, nedeljne i mesečne alarme koji vas sat ili dan unapred upozoravaju na poslove koje treba da obavite.

Ambiciozniji korisnici mogu da kupe ROM module koji se povezuju sa jednim od dva ekspanziona slotova. Trenutno su posebno popularni *Pocket Spreadsheet*, program za unakrsna izračunavanja, i *Superchig* koji, priključen u port B, nudi moćnu bazu podataka koja je mnogo fleksibilnija od ugrađenog adresara, mini tekst procesor i uslužni program za dodeljivanje funkcija tasterima.

U vezi sa „velikim bratom“

Posredstvom ugrađenog serijskog interfejsa i programa *Comms Link*, „psion organajzer II“ se jednostavno povezuje sa bilo kojim personalnim računarom; vaša telefonski imenik možete, dakle, da unesete pomoću standardne tastature i da ga onda prekopirate u „organajzer“. Komunikacija je, jasno, dvosmerni: zabeleške koje ste uneli „na terenu“ lako prenosite na disketu nekog PC-ja, ili „amstrada“ da biste ih uključili u izveštaj o projektu na kome radite.

„Psion organajzer II“ košta 100 funti, pri čemu je u ovu cenu uključeni i VAT od 16%. Cena raznih aplikacionih i memorijskih modula kreće se između tridesetak i sto funti, što približava čitavu konfiguraciju čak i korisnicima sa relativno plitkim džepom. Jedinu prava konkurenciju „psion organajzeru II“ su minijaturni japanski rokovnici koji se, istini za volju, ne mogu programirati, ali koji su zato teški jedva pedesetak grama i ne prelaze dimenzije vizit karte. Tržište džepnih računara je, očito, sve aktivnije!

Dejan Ristanović

Violinista na kompjuteru

Vlasnicima QL-a sigurno je poznata igra „Cavern“, ali malo ko zna da iza nje stoji Janko Mršić-Flögel, Zagrepčanin, trenutno stanovnik Londona i vlasnik „Paradox software“-a, čovek koji svoju pažnju deli između dve podjednake ljubavi — violine koji svira od malih nogu i računara kod kojih je na najbolji način spojio lepo sa korisnim.



Programer i biznismen: Janko Mršić-Flögel (levo) sa našim saradnikom

Janko je rođen u Zagrebu 1968. godine. Živio je na Salati blizu mjesta gdje je sada sportski kompleks izgrađen za Univerzitet. Pohadao je osnovnu školu „Rudi Čajavec“. Po završetku sedmog razreda osnovne škole preselio se u London, gdje je pohađao „Highgate school“ — srednju školu, jednu od najstarijih u Britaniji (počinje iz 16. stoljeća). Trenutno je student druge godine „Imperial college“ u Kingstonsu (jedan londonski kvart). To bi, ukratko, bila Jankova biografija. Kao i mnoge druge — bezlična i suhoparna, ni jedne riječi koja bi mogla dati naslutiti da se radi o prijaznom, srdačnom i nadasve inteligentnom čovjeku. Ni riječi o smješku kojim obično počinje objašnjavati zašto i kako se počeo baviti računarima. A počelo je slučajno.

Na nagovor učitelja srpskohrvatskog jezika (i), krenuo je u Multimedijski centar — tadašnju zagrebačku oazu računarstva. Bilo je to u petom razredu osnovne škole. „Tada sam se samo igrao“ — priznaje Janko, ali nije zapostavljao ni programiranje. Istina, bilo je to u bejziku, ali ga to nije spriječilo da kasnije postigne prave rezultate (šta bi na ovo rekli bejzkomrci). I tako je Janko počeo svraćati u Multimedijski, gdje ga je poslije nekog vremena zapazio Branimir Makanec — jedan od osnivača tog centra. Janko je u Zagrebu ostao do svoje 13-te godine, kad je sa završnim sedmim razredom osnovne preselio u London jer je kao pobjednik na jugoslovenskom takmičenju mladih violinista dobio muzičko stipendiju.

Prešavši u London, došao je u sredinu u kojoj je talas kompjuterizacije bio u punom zamahu. Njegova ljubav za računare je tek sad mogla doći do izražaja. U Londonu je kupio i svoj prvi računar — „apl II“. I ubrzo je počeo pisati za njega. Ti prvi programi nisu bili namjenjeni tržištu, ali je Janko '84

godine posudio jednog BBC-a i napisao program koji je bio i te kako komercijalan. Bila je to igra „Lifty-Lifty“ za firmu „Visions“. Igra je dobila vrlo dobre kritike, koje su naročito isticala grafiku i „playability“ (u slobodnom prevodu, igrainost). Ubrzo su slijedile i druge stvari. Igru „1994“ je predao za „spektrum“ i C64. Poslije je napisao i gruču „BIG BEN“ za firmu „Interceptor“.

A onda se pojavio QL koji mu se odmah dopao. Bilo je to početak jedne nove faze za njega — prelazak na Motorolu 68000. Janko je za QL-a odmah napisao gore pomenutu igru „QL-cavern“, koja je postala najprodavaniji program za QL svih vremena i ujedno prvi program koji je Sinclair research kupio od nekoga za QL od izbacivanja QL-a na tržište (Psion-ov poslovni paket je bio naručen prije no što je QL i napravljen). Ali, to nije bilo sve. Janko je za QL napravio još desetak igrā.

A onda je u julu 1985. godine osnovan „PARADOX SOFTWARE“. U to doba se već naveliko pričalo o Džeku Tremleju i novom Atariju. Naravno da je i „Paradox software“ pozvan na prezentaciju ST serije računara. I tada je proradila Jankova hakerska žica.

„Nisam mogao da odolim“ — kaže Janko. „Odmah sam se bacio na pisanje programa“. I rezultati su ubrzo bili vidljivi. Izbacili smo dva programa: igru „Mission house“ i uslužni program „Desk diary“, koji je, ujedno, bio i prvi program za ST-a koji su u GEM-u napisale nezavisne softverske kuće“. Janko je do sada ukupno napisao dva poslovna programa i oko petnaestak igrā. I pored programiranja, Janko nije zapostavio ni muziku i violinu. Još uvijek je član „Young musicians symphony orchestra“ — simfonjskog orkestra sastavljenog isključivo od mladih muzičara.

Saša Svitlica

Osnovna škola za odrasle

Kada posetite „Iskra Delta“ i njihov novi proizvodno-razvojni centar u Stegnama kod Ljubljane, reći će vam da to nije sve što vredi videti. Sa ponosom će početi priču o svom obrazovnom centru u Novoj Gorici. Ne bez razloga. Ovde je odavno shvaćeno da se sa golim hardverom ne može ništa učiniti. Kupcu, uz mašine, treba dati i odgovarajući softver i treba ga — obučiti.

Centar „Iskra Delta“ je nastao pre dve godine, i to slučajno — kaže Oton Mozetič, direktor osnovne jedinice „Iskra Delta“ u Novoj Gorici. — Ovde je bio propal hotel „Argonauti“, koji je bio dosta zapušten, i u kome su boravili, uglavnom, turisti iz istočnih zemalja. Nije bilo ni velikih poslovnih efekata. Pojavila se „Iskra Delta“ i uradila ono što niko u okolini nije bio ni spreman ni sposoban da učini. U saradnji sa lokalnim hotelskim preduzećem i uz pomoć opštine i banaka, projekat je počeo.

Čarobna moć granice

U Ljubljani smo već imali obrazovanje, ali ono naučnije, priča Mozetič. Ovakvo nešto bilo je novina. Od predstavnika sa jednim čovekom dobili smo veliki obrazovni centar. Ne treba kriti, ističe Oton, da je i blizina granice prednost ove škole, jer zastupnici stranih firmi imaju svoje „mamece“ — razna putovanja koja su u pozadini velikih prodaja.

— Mi ljude iz Makedonije, na primer, dovodimo u Novu Goricu da im pokažemo ovaj centar gde će učiti kako se najbolje koristi računar, ali i da pogledaju naš proizvodni program, koji je ovde postavljen i koji koriste naši polaznici. Nižmu, izmedu ostalog, nije tajna, privlači mogućnost izleta u susednu Italiju.

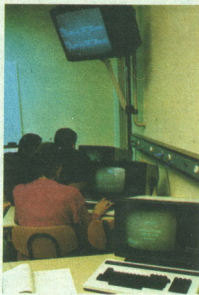
Tako u renovirani školski deo — hotel je pretrepo manje prepravke — stižu korisnici „Deltinim“ proizvoda iz cele zemlje. U strahotini ovog proizvođača računara obrazovne, kažu, ima izuzetan značaj. Po njima, nije problem samo kako popuniti kapacitete već i kako računar opslužiti da iz njega „istisneš“ sve što nudi. Zato se tako kupci i spremaju za to.

— Jer, kakva bi nam reklama bila da imamo računare koje ljudi ne znaju da koriste. Istina je da da toga ima kod nas u zemlji, jer dolaze nam ljudi koji nemaju ni ono osnovno znanje da bi počeli kurs u našem centru — kaže Mozetič.

— Uz obuku pruža im se mogućnost da vide šta proizvodi „Delta“, pa je značajna i tržišna funkcija obrazovnog centra u Novoj Gorici. Zato je u tom kraju „Iskra Delta“ napravila dva jaka referentna centra. Jedan je u energetici, daljinsko upravljanje hidroelektranom na Soči (komandni centar je opremljen „Deltinim“ mašinama), a drugi je zdravstveni informativni centar u Novoj Gorici.

Kupci nas usavršavaju

Ali, znanja u Novoj Gorici ne dobijaju samo gosti ili polaznici kurseva. Znanja dobija i „Iskra Delta“, vrlo zanimljivo razmišlja Mozetič. Jer, kupac dolazi sa svojim znanjima i potrebama, iz drugih područja i 12 računari 36 • mart 1988.



oblasti rada, toliko potrebnih „Delti“.

— Udovoljavajući zahtevima kupaca, sami sebe usavršavamo. Bez vođenja kupca, naročito u razvoju, ne može se daleko doći. Naš razvoj usmeravamo, pre svega, tržište i kupac. Lutanja ne sme da bude. Ako ga je bilo, bilo ga je upravo zbog toga što nisu pronađene potrebe kupaca — naglašava direktor jedinice u Novoj Gorici.

Zahvaljujući velikoj frekvenciji ljudi kroz centar, uspešli su da krenu sa predavanjima iz rukovođenja. Ta druženja im omogućavaju i da prate potrebe širom zemlje, da bi kasnije okupili zainteresovane i predstavili im svoj program. Ko dolazi u školu „Iskra Delta“ u Novoj Gorici?

Od ponedeljka do petka

Pre svega, kupci. Kada zaključuje ugovor, u kome postoji član koji određuje koliko troškova za obrazovanje daje „Iskra Delta“, dobija bonove i katalog sa programima obrazovanja i centru. Sam bira termin, program koji će pohađati... naravno, u zavisnosti od opreme koju je kupio. Seminari traju različito. Po pravilu, uči se od ponedeljka do petka. Ali, kursevi su tako podešeni da se mogu nastavljati, u neke veće „module“, koji obrazuju operatore, programere, sistemske inženjere... Naravno, to je samo dopuna njihovih znanja.

Vrata centra otvorena se preko cele

godine, sem mesec dana preko leta, kada je remont opreme. A nje ima zaista dosta. U računskom centru su mašine „Delta“, u 22 učionice je 150 terminala (u svakoj učionici najviše 16 ljudi)... Iza toga stoje dva „Delta“ računara 4860. Tu su i „partneri“, „triglav“, a uskoro će biti i AT. Predavanja drže instruktori proizvođača, ali i drugi, jer je u Gorici samo 12 zaposlenih. Dolaze stručnjaci iz ljubljanskog dela „Delta“, ali i ljudi iz „Digitala“, jer nije retkost da stručnjaci „Iskra Delta“ gostuju u „Digitalu“.

U centru se usavršavaju i ljudi iz same „Iskra Delta“. Na primer, hardversko obrazovanje, obuka serviseru za hardver, postoji samo u Novoj Gorici, i njega dobijaju samo zaposleni u „Iskra Delta“. Planovi rada centra postoje za šest meseci unapred, ali prva tri u tom planu su fiksirana, sigurna. Kada se oduče za termin, kupci se jave, i time rezervišu mesto i hotelu obrazovnog centra i na samom kursu. Nastava traje od 9 ujutru do 13 časova, da bi se nastavila posle ručka. Grupu vodi instruktor, koji sa njima živi, jede, prati ih i maltene ispunjava želje. Ukoliko su polaznici povučeni, objašnjava Mozetič, u svakoj sobi se može postaviti terminal povezan sa našim računskim centrom. Ako su zadovoljni nastavom, u slobodnom vremenu se mogu opustiti u bazenu, na teniskim terenima, trim stazi... Fakultetski obrazovni čovek zadržava za se rekreaciju kupaca-polaznika. Ipak, u petak je, kaže Oton, najpopularnija rekreacija obilazak prodavnica u italijanskoj Gorici.

Poligon za testiranje

Da li u zemlji postoje slični obrazovni centri?

— „Intertred“, zastupnik IBM-a. Imalo nešto slično u Radovljici. Znam da EI-Hanvel nešto priprema u Izoli, nekoliko učionica... Okolina centra „Intertreda“ je izuzetna i objekat im je nov, ali mislim da nemaju ovakvu koncepciju rada ni ovakvu tehničku podlogu, jer su kod nas instalirani poslednji proizvodni noviteti, za razliku od njih. Nama je, praktično, ovaj centar poligon za testiranje, naša prva probna instalacija.

Mozetič naglašava da su vrata centra ptovorena i za druge, ne samo za kupce. Školske centre, doduše manje, „Iskra Delta“ ima i u većim gradovima u zemlji, koji su u nekoj hijerarhiji, ispod ovog u Novoj Gorici. Ako neki kupac, ili u nekom delu zemlje ima veći potreba za specijalnim kursovima, veli Mozetič, onda šaljemo instruktore u bilo koji od ovih centara — Zagreb, Beograd, Novi Sad, Skopje... Možemo ih imati i u fabrikama, tu smo dosta fleksibilni.

Momčilo Cebalović



POSTANITE ČLAN KLUBA ČITALACA BIGZ-a „KULTURA”

BIGZ je osnovan 1831 godine. Bila je to prva štamparija u Srbiji i njeno osnivanje predstavlja izuzetan događaj u istoriji srpskog naroda u borbi za nacionalnu slobodu tokom nekoliko decenija, a za razvoj kulture sve do današnjeg dana. Pored knjiga kao što su „Život i priključenja“ Dositeja Obradovića, „Srpske narodne pesme“, „Kovčević za istoriju, jezik i običaje Srba“ Vuka Stefanovića Karadžića, u ovoj štampariji štampani su prvi srpski bukvar, prve srpske novine i časopisi, prvi srpski zakon.

Od početka rada, prva srpska štamparija deluje i kao izdavač. To što se začelo pre više od veka i po — zajedništvo štamparstva i izdavaštva — svoj puni izraz nalazi danas u Beogradskom izdavačko-grafičkom zavodu.

Zadatak BIGZ-a je:

- da izdaje vredne knjige iz svih oblasti savremenog stvaralaštva kao i celokupnog književnog i kulturnog nasleđa;
 - da izdaje dela pisana na svim jezicima naroda i narodnosti Jugoslavije, kao i celoga sveta, radi boljeg međusobnog upoznavanja naroda, kultura i civilizacija
 - BIGZ posebno neguje knjigu iz oblasti humanističkih nauka;
 - svojom džepnom knjigom, jeftinom i dobrom, BIGZ se prepoznaje među jugoslovenskim izdavačima.
- Pošto je knjiga sve skuplja i sve se teže kupuje BIGZ vam nudi željenu knjigu s popustom, koji možete dobiti ako

postanete član Kluba čitalaca BIGZ-a. Članom Kluba čitalaca mogu postati lica ili organizacije koje zele i mogu da saraduju na ostvarenju zadataka BIGZ-a.

Član Kluba čitalaca BIGZ-a postajete uplatom godišnje članarine, u svim BIGZ-ovim knjižarama širom Jugoslavije. Visina članarine utvrđuje se odlukom Radničkog saveta BIGZ-a i iznosi za 1988 godinu 15.000 dinara, a za studente i učenike 5.000 dinara i može se uplatiti u tri mesečne rate.

Za inostranstvo članarina je dvostruka, uz posebnu naplatu poštarine. Za pravna lica: radne organizacije, biblioteke, članarina je 65.000 dinara.

ČLANOVI KLUBA ČITALACA BIGZ-a ZA SVOJU ČLANARINU DOBIJAJU:

— PRAVO DA U TOKU GODINE ČETIRI PUTA MOGU KUPITI KNJIGE U IZDANJU BIGZ-a, PO SVOM IZBORU, SA 50%, A ZA DŽEPNU KNJIGU SA 40% POPUSTA;

— 30% POPUSTA NA SVA IZDANJA BIGZ-a;

— 20% NA PREUZETA IZDANJA, OSIM UDŽBENIKA I ENCIKLOPEDJSKIH IZDANJA;

— BILTEN BIGZ-ovog KLUBA ČITALACA KOJI IZLAZI KVARTALNO;

— PRAVNA LICA ČLANOVI KLUBA ČITALACA BIGZ-a DOBIJAJU:

— PRAVO DA U TOKU GODINE MOGU DVA PUTA KUPITI KNJIGE PO SVOM IZBORU SA POPUSTOM OD 40%, A UDŽBENIK I LEKTIRU SA POPUSTOM DO 20%; BILTEN BIGZ-a;

— SVI ČLANOVI UČESTVOJU U NAGRADNIM IGRAMA TOKOM GODINE.

Članovi Kluba čitalaca BIGZ-a mogu kupiti knjige na kredit — otplatu u najviše devet mesečnih rata, s tim što rata ne može biti manja od 5.000 dinara. Kupovinom knjiga na otplatu do pet meseci, ne zaračunava se kamata. Do devet meseci zaračunava se kamata od 16,4%. Kod kupovine knjiga na otplatu potrebna je overa radne organizacije; penzioneri prilažu pretposljednji ček od penzije.

BEGRADSKI IZDAVAČKO-GRAFIČKI
ZAVOD
11000 BEOGRAD

Bulevar vojvode Mišića 17/6
Klub čitalaca BIGZ-a
Tel. 653-763; 653-899

OBLASTI INTERESOVANJA

1. Beletristika
2. Filozofija, sociologija, psihologija
3. Meritizam i društveno politička literatura
4. Istorioigrafija
5. Lekalioigrafija
6. Umetniost
7. Pionirsko i film
8. Knjige za decu
9. Publicistika
10. Posebna izdanja

RAČUNARI — 36

(prezime i ime)

(zanimanje — adresa stanovanja)

(datum rođenja, broj i/k, matični broj)

Na žiro-račun BIGZ-a broj 69082-603-23264 uplatilo sam

dana

dinara

Članski ulog može se uplaćivati u ratama (najviše 3 rate). Sve prava članova teku po uplati članskog uloga.

(datum)

(potpis)

OKRUŽITE OBLASTI KOJE VAS ZANIMAJU!

PRISTUPNICU I PETI PRIMERAK UPPLATNICE POŠALJITE NA ADRESU KLUBA!

DOBRO DOŠLI U KLUB ČITALACA BIGZ-a!

Dejanov trio

Dejanove pitalice već više od dve godine zaokupljaju pažnju ljubitelja teško rešivih zadataka. Tokom prošle godine težina pitalica je oscilovala — od izuzetno lakih, na koje smo dobijali stotine odgovora do onih kod kojih je i sam Dejan morao da se pomuču da pronađe rešenje — ali su čitaoce ove rubrike i u najtežim trenucima ostajali sa Dejanom do kraja. Ko su rešavači koji su u prošlogodišnjem ciklusu pokazali najviše uspeha? Sticajem okolnosti, od osam trećeplasiranih rešavača predstavljamo samo Sinišu Stamatovića. Nadamo se da ćemo tokom sledećih meseci uspeti da predstavimo i ostale rešavače, kojima je za dlaku izmaklo jedno od prva dva mesta.

Milan Grbić

„Nisam rob kompjutera“

Ne živim samo za i oko kompjutera, kaže Milan Grbić, dobitnik prve nagrade na „Dejanovim pitalicama“. Nekada i po deset dana ne pipnem kompjuter.

Milan Grbić, iz Sremčice kod Beograda, pokazao se kao najuspešniji rešavač naših pitalica. Izgleda, međutim, da je verovatno bilo lakše nalaziti rešenja za zadatke iz

jesam, bio sam četiri godine stariji — iznosi Milan svoje srednjoškolske utiske.

Ali, izgleda da nije samo za taj period vezan neki Milanov nedefinisan pesimizam, u ovim vremenima verovatno prisutan i kod velikog broja drugih mladih bica.

— Od studija ništa nisam očekivao. Ako sam nešto dobio, ne znam šta je to. Još ne razmišljam o smeru na daljim godinama studija. Ma ne, mašinstvo me uopšte ne interesuje — iznenađuje nas Grbić.

— Možda elektrotehnika, pitamo ga.

Odgovara nam sleganjem ramenima. Izgleda da ga ni to ne privlači. Možda tom pomenutom nedefinisanom pesimizmu koji smo pročitali iz njegovih očiju, odgovara i ponašanje za onih sat vremena druženja sa njim, razlog i neke poteškoće na studijama. Kaže da ne razmišlja ni šta bi jednog dana mogao da radi. Možda, programiranje, ali... Ovo zvanje programera sa srednjom školom, po njemu, nije ništa.

Kompjuter mu, bar za sada, ne pomaže mnogo na studijama.

— Imam „Komodor PC10“. Programiram razne stvari — istiche. — Radim na turpaskalu. Video igre? Ne, nemam ni jednu. Možda će uskoro biti i toga. Ne robujem kompjuteru. Dešava se da danima, čak i po desetak dana, ne sednem za mašinu, ne pipnem je. Rad i sedenje za računarnom zavisi od moje inspiracije. Ponekad sedim po tri, četiri sata. Ali, tu nema pravila. Kompjuter je za mene samo mašina, ali i izazov. Ne, nemam društvo za kompjutere, obično su to drugi sa fakulteta. Ja nisam rob kompjutera, ne živim za njega, oko njega ili zbog njega.

— Kompjuteri se uvode i u škole. Klinci se sve ranije upoznaju sa tim mašinama. Da li je to dobro?

— Ja ne vidim korist od toga. Mislim da je to uzaludan posao — smatra Milan Grbić.

Uz računare, posvećuje se i naučnoj fantastici. Trenutno čita „Neuromanse“ Billa Gibsona.

— Da li si zadovoljan postavljenim pitalicama?

— Mislim da nisu bile teške. Onu igru „Brojevi i slova“ nisam rešio. Bile su zanimljive, ali ne i toliko teške. U svakom slučaju, uživao sam radi takmičenja i iskušavanja sebe van a ne radi nagrada — kaže naš pobeđnik.

Milan Kovačević

„Iz mašine izvlačim maksimum“

Nepoznanica šta i koliko mogu računari i kolike su im mogućnosti, voli Milana Kovačevića iz Šida, drugoplasiranog u „Pitalicama“, u potragu za njihovim granicama. Diplomirani matematičar, star 24 godine, traži posao.

Poznatim pravcem kojim idu oni kojima su jača strana prirodne nauke hodio je i Milan Kovačević iz Šida. Srednja škola, prirodno-matematički smer, praksa i susret sa računarima u računskom centru u Šidu, maturski rad iz programiranja, upis na Prirodno-matematički fakultet u Novom Sadu, su Milanove stepenice na tom usponu do diplome koju je dobio u 23. godini. Sve u roku. Trenutno je nezaposlen, u potrazi za poslom. Ima planova, koji su, veli, vezani za Novi Sad.

I dok tako traga, sa velikim šansama da potragu ubrzo završi, zanimao se, između ostalog, i našim pitalicama. Ali, voli i programiranje, matematiku, šah, „Galaksiju“ preko koje se upoznavao sa kompjuterima-šahistima... Tako je i kupio „Chalenger 7“. U svoje vreme mučio se i sa madarskom kockom, nabavio džepni računar HP41CV, saradivao u „Klubu programera“, „Galaksiju“... „Računare“ čita od prvog broja. Dobro je što dostaje prenose iz strane literature, kaže.

— Interesuju me granice računara; šta mogu da urade, kolike su im moći, koliko mogu da nam pomognu. Navikao sam da iz svake mašine izvlačim maksimum. Valjda sam zato i danas na „spektrumu 48K“, koji sam kasnije nabavio — objašnjava nam Milan Kovačević. — Inače, po meni, neke granice računara se već naslućuju. Poznata je knjiga „Šta računari ne mogu“.

Ali, mnogo je važnije da i ove mašine koje imamo, smatra Kovačević, koristimo najviše što one pružaju. Da se ljudi osposobe da ih koriste, a kasnije se mogu ustvari i u programiranje.

I on, kao i Grbić, ponekad danima ne sedne za kompjuter. Ali, kada ima posla,



„Računara“ nego „otvoriti“ pobeđnika, „raskraviti“ ga za razgovor o sebi.

— Nisam se stegao, ali... ne znam šta bih pričao o sebi. Ja sam sasvim prosečni čovek, ništa specifično — reče nam negde pri kraju razgovora kada je naš novinar iscrpeo sva pitanja.

Ipak, nešto smo saznali. Milan je star 21 godinu, nalazi se na drugoj godini mašinstva. „Nema još smera“, kaže, a kompjuterima se zanima nekoliko godina. Koliko?

— Ne sećam se, veli, bilo je to valjda na nekom sajmu, kada sam prvi put došao u dodir sa tim „mašinama“.

Kao po nekoj školi, a mladi valjda tako žive, srednja škola i matematičko-tehnički smer u 13. gimnaziji, smer programera, važe i za Grbića.

— Srednja škola mi je bila dosadna. Da li sam izašao bar pametniji? Pa, valjda

Siniša Stamatović

„Mrzim bubanje napamet“

kada sedne. . . Najviše se, što je normalno, bavi programiranjem, a matematika je prva stepenica koju treba savladati i dobro je poznavati za bavljenje kompjuterima, kaže.

— U osnovi je matematika, jer i danas se traže brzi i efikasni algoritmi. Izabrao sam ono što volim, što me zanima, a mislim da sam na fakultetu dosta toga naučio, ističe Kovačević. Bio je i na mnogim takmičenjima, posebno iz matematike. Volim da se takmičim, još od osnovne škole posećujem takve „svečanosti“.



Ali, nisu samo periodi zainteresovanosti i čekanje trenutak inspiracije za rad sa računarom ono što povezuje naše dobitnike. I Kovačević čita naučnu fantastiku, jer „tu ima interesantnih ideja“. Dobro je što se sve više piše o PC mašinama, ali nije dobro kada se veća pažnja posvećuje hardveru nego softveru, smatra on. Nagrada ni njemu nikada nije bila cilj kada je polazio „u boj“ sa pitalicama. Takmičenje je, po njemu, bilo zanimljivo, mali izazov za one koje interesuju takvi problemi. Kako kaže, zadaci su bili vezani za proste brojeve. Sve u svemu, bilo je fino, zaključuje.

Nije tačno da se mi, koji volimo kompjutere, zanimamo samo njima. Slobodnog vremena može da se nađe, kada se čovek dobro organizuje, pa ja, veli Milan, odlazim i u pozorište, bioskop, čitam knjige. . . Mislim da je važno da svako nađe sebe, ono što ga privlači.

Bubanje napamet i društveni predmeti prava su mi muka, kaže Siniša Stamatović. Zanima ga primena računara u radio-tehnici. Šta je video na Mesecu, Jupiteru, Saturnu. . .

Takmičenja nisu nikakav bauk za Sinišu Stamatovića, trećelasiranog u „Pitalicama“. Često se pojavljuje na njima, a prvi izlasci u svet igara bili su još u trećem razredu osnovne škole, kada je dobio čokoladu kao nagradu za najbolju igračku. U petom razredu postoje „prvak“ iz matematike, kada od 20 mogućih osvaja 19 bodova. Sa godinama dolazi i važnost nastupa, pa se pojavljuje i na pokrajinskim takmičenjima. Kao dobar matematičar dobija Alasovu diplomu. I kada su ga profesori nagovarali da krene na matematički smer, kada su svi očekivali da je to normalna stvar, on se odlučuje za informatiku. To je tada bila novost, kaže. U početku je bilo malo teže, jer sam se morao navići na novi način razmišljanja, ali kada sam shvatio neke principe programiranja, moja dalja profesionalna orijentacija bila je određena, objašnjava nam.

Danas, 25-godišnji Siniša Stamatović, ima 11 ispita od kraja na Prirodno-matematičkom fakultetu u Novom Sadu.

Bilo je nekih problema, ali sada sam „otkako“ glavne prepreke, pa mi je put do diplome, čini mi se otvoren, kaže. Muka mi je bubanje napamet, društveni predmeti. . . kod programiranja i struke, nema problema. Pre nešto više od tri godine nabavio sam „spectrum“.

— Oduvek sam maštao da se tako neka mala spravica nađe u mojoj blizini. Prave mogućnosti te mašine nisam znao i počeo sam da ulazim u njegove tajne — priča nam Stamatović.

— Za „spectrum“ je bilo najviše uslužnih programa. Znanje sa fakulteta sam mogao da primenjujem na njemu — naučio sam paskal. lisp, prolog. . . Trenutno znam 10—15 jezika, ali mi nije problem da naučim i ostale koji se pojavljuju, kaže Siniša. Zanimam se uglavnom domaćom računarskom literaturom, jer do strane se teško dolazi. Sećam se kada sam preko „Galaksije“ dolazio do igara, slao ih i dobio. Mađarskom kokcom sam se bavio za vreme vojske. Na fakultetu sam dosta vremena provodio, čak i ono slobodno, pored njihovog računara PDP11. Na radnoj akciji Lenka 87 bio sam instruktor za mikroručinare.

Siniša se, uz računare, bavi i radio amaterstvom. Ali, ni tu ne može bez kompjutera. Posebna mu je poslastica primena računara u radiotehnici. Trenutno je aktuelan „radio packet“, kaže, pa čemo u skorijoj budućnosti, zahvaljujući i računarima, komunicirati sa celim svetom. I on, kao i Grbić i Kovačević, za mašinu sedne s vremena na vreme.

— Ponekad sedim po pet-šest sati, kaže. Ali, ovih dana i meseci bio sam angažo-



van samo na ispitima i pitalicama iz „Računara“.

— Pa, kakve su bile?

— Bilo je izuzetno teških. Onih, što ih zovem „sitna creva“, bilo je i zadatka sa cakom, ali i izuzetno lakih — kaže Stamatović. — U izuzetno teške ubrojio bih „najveći prost broj“ i „brojke i slova“. U teške, „šahovski problem“ i „maglični sestougo“. „Sitna creva“ su bila „proste sekvence“, „prijani mornari“ i „telefonska zagonetka“. Zadaci sa cakom su bili „monetarna pitalica“, „četiri četvorke“ i „fotografi i modeli“. Nikakav problem nisu predstavljali zadaci „putevi“ i niz brojeva deljivih sa 99. Najviše sam se zadržao na „telefonskoj zagonetki“. Mislim da bi taj program mogao da ima upotrebnu vrednost i u nekoj pošti.

Ipak, problem za Sinišu bilo je i što su laki zadaci izlazili kada je imao najviše slobodnog vremena, dok su se oni teži pojavljivali kada je bio u stisci sa vremenom. Možda je i to uticalo što sam treći, kaže on.

Naučna fantastika je ona spona koja spaja naše dobitnike. I Siniša Stamatović čita naučnu fantastiku, interesuje ga svemir i njegova nepoznanica, ali on, za razliku od drugih, ne ostaje samo na čitanju. Zahvaljujući jednom drugaru i jednom teleskopu, često vida „prava čudesa“. — Video sam krater na Mesecu — kaže. — To je bio doživljaj. Zatim, sam posmatrao Jupiter, Saturn, Mars. . . Video sam i druga, nesvakidašnja i teško objašnjiva događanja. Ali, o tome se obično čuti, pa u to mnogi i ne veruju, kaže Siniša Stamatović.

Razgovor vodilo: Momčilo Cebalović
računari 36 • mart 1988. 15

AMIGA 500

— Računar godine 1987

U organizaciji stručne publikacije CHIP, međunarodni žiri iz 8 zemalja izabrao je mikroračunar godine za 1987. U kategoriji kućnih računara sa velikom prednošću pobedio je računar AMIGA 500.

Ocene stručnjaka su jednoglasne u tome da AMIGA 500, kao 16/32 bitni multitasking računar nove generacije i pravi naslednik računara C 64, postavlja nove standarde u toj kategoriji računara za 90-te godine.

AMIGA 2000

U osnovnoj varijanti performanse računara su slične performansama AMIGE 500. Kao otvoren sistem daje mogućnost modularne dogradnje računara po željama kupaca:

- dodatna ugradnja 1x3,5 inča disketne jedinice i 1 hard disk, ili
- dodatna ugradnja 1x5,25 inča disketne jedinice i 1 hard disk
- sa ugradnjom dodatne kartice postiže se kompatibilnost
- 5 AMIGA dodatnih mesta za proširenja
- 4 PC XT/AT mesta za proširenja
- proširenje RAM do maksimalno 9,5 MB

Cena*
Osnovna izvedba \$2.118,72



Tehničke karakteristike

Mikroprocesor:	MC 68000
Clock (MHz):	7.159
Dodatni procesori:	8086, 4,77 MHz 80286, 8 MHz 68020
Tastatura:	profesionalna, sa 95 tastera (10 funkcijskih)
ROM:	256 K
RAM:	512 K, maksimalno proširljivo na 8,5 MB
Grafika:	320x256 (32 boje) 640x256 (16 boja) 320x512 (32 boje) 640x512 (16 boja) 320x256 (4096 nijansi)
Paleta:	4096 boja
Ugrađena disketna jed.:	3,5 inča, 800 K
Dodatne disketne jedinice:	3
Dodatni hard disk:	20—80 MB
Interfejsi:	IBM PC kompatibilni RS 232 i Centronics — analogni RGB video ulaz i izlaz — stereo audio ulaz i izlaz, UHF izlaz, 2 palice
Ostala proširenja:	AMIGA sistemski bus (vodilo)
Operativni sistem:	AMIGA DOS MS DOS (opcija)
Dobavljeni softver:	AMIGA DOS, Basic, Workbench, CLI
Cene:*	
AMIGA 500	\$676,90
Color monitor 1084	\$385,22
Printer MPS 1200P	\$269,11

Zastupnik: KONIM
61000 Ljubljana, Titova 38
Telefon: 061/312-290

* Na cenu u devizama dodati 60% dinarskih troškova

Unos i priprema teksta

Do nedavno se primena računara u obradi teksta uglavnom zaustavljala u trenutku kada je rukopis gotov. Prilikom štampanja sve se odvijalo kao da računara u fazi pisanja nije ni bilo. Dakle, tekst je ponovo prekućavan, štampač se probni primerak na kome su vršene ispravke i prolazilo kroz još mnogo faza i ruku da bi se dobila konačna forma. Danas je situacija u mnogo čemu drugačija. Pojavom laserskih štampača, uz dovoljno snažne računare iz personalne klase, postaje sasvim moguće ceo proces do samog trenutka umnožavanja držati pod kontrolom.

Pošto smo u prošlim nastavcima izneli najveći broj argumenata za i protiv stonog izdavaštva, možemo da pređemo na konkretne stvari. Prvi korak je, svakako, unos teksta i njegova priprema za slaganje. Ova tema, na prvi pogled, ne deluje ni malo interesantno i najčešće se potpuno preskače pri razmatranju uvođenja stonog izdavaštva. Najčešći komentar je da tu i nema šta posebno da se priča, jer postoji nebrojno mnogo tekst procesora koji mogu poslužiti za unos. Programi za slaganje su u stanju da raspoznaju format gotovo svih poznatih tekst procesora, tako da problem izgleda potpuno rešen. U čitavom lancu radnji od rukopisa do konačnog otiska, ovakav stav predstavlja najveću moguću grešku i zapravo protiče iz već opisanog i kod nas često primenivanog metoda izvlačenja zaključaka na osnovu smešno malog uzorka. Dakle, ako želite da pripremite za štampu jednu stranicu propagandnog materijala za neki proizvod, upotrebićete bilo koji tekst procesor (WordStar, WordPerfect i njime slično) i uneti jednu stranicu teksta, startovati program za prelom, označiti koji ste program za unos koristili i nastaviti sa igrom oko rasporeda ilustracija i teksta.

Primenite isti metod na knjižicu od dvadesetak strana koja ima nekoliko naslova i podnaslova sa par tabela i imaćete dobre izgleda da posvo završite u razumnom roku. A zatim uzmete, recimo, zbornik radova sa nekog kongresa — stotinu strana sa stotinu naslova, tri stotine podnaslova, stotinu tablica sa stotinu zaglavlja, linija, potpisa, fusnota... Neće trebati mnogo vremena da shvatite da je odgovarajuća priprema — od unosa do ulaska u program za prelom — zapravo najveći i najzobitnijiji deo posta, kome se mora pristupiti na krajnje ozbiljan način.

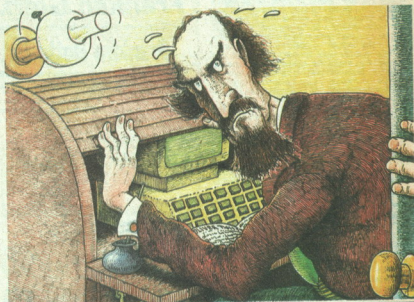
Odmah da napomenemo da ovide neće biti reči o „amaterskom“ unosu teksta — ako autor ima mogućnosti da piše uz pomoć računara — što se, nadamo se, brzo postati uobičajena praksa — onda unos teksta nema mnogo veze sa ovim o čemu ćemo govoriti. Nastanak rukopisa i prednosti koje računar u ovoj fazi pruža su tema za sebe o kojoj je već bilo reči. Međutim, kao što ćemo nešto kasnije pokazati, bez obzira na dizajnerski materijal, tekst, nakon što se nađe na disketu, zahteva dodatnu obradu da bi se sam prelom obavio na što lakši način.

Unos teksta

Pretpostavka da unos teksta da obavlja profesionalni daktilograf sa ciljem da se tekst nađe na disketi za što kraće vreme i sa što manje grešaka postavlja određene zahteve pred program koji se koristi. Otvorenost zahteve pred program koji se koristi za unos teksta na neki način postaju u ovom delu nedostatak. Koncept koji se koristi za unos treba da zadovolji samo nekoliko kriterijuma:

1) Brzina prihvatanja tačera i rada upotrebom mora biti velika, što podrazumeva odsustvo raznih formatiranja u toku pisanja, a pogotovo potrebe da se dodatnim tasterima tekst nakon korekcija dovodi u red.

2) Jednostavnost rukovanja i odsustvo (ili barem mogućnost blokiranja) složenih opcija iz kojih se teško vraća u režim za unos. Posebno je važna jednostavnost opcija za snimanje i upis teksta.



PISMO - FONT

U razgovoru sa ljudima iz struke i često se javljaju nesporazumi oko termina PISMO i FONT koji se izjednačuju. Kada nekom saopštite da raspolazete sa na primer trideset fontova, verovatno će ostati impresioniran vašim mogućnostima izbora jer on može da bira smo između tri ili četiri pisma. Na kraju će se verovatno ispostaviti da je njegov izbor veći jer vaših trideset fontova u stvari mogu biti i samo jedno pismo. Jedno pismo u jednoj veličini čini jedan FONT. Engleski izraz za pismo je TYPEFACE.

AaBbCcDdEeFf Times 12pt Normal

AaBbCcDdEeFf Times 12pt Bold

AaBbCcDdEeFf Times 12pt Italic

Kod laserskih štampača ovo su najčešće sretna varijante u jednoj veličini pisma

3) Mogućnost unošenja veće količine teksta bez „gubljenja“ — granica od 64 K se može smatrati kao najmanja vrednost koja zadovoljava. U praksi se pokazalo da preizak sa pisacima mašine na računar nije ni malo težak ako su zadovoljeni prethodni kriterijumi. Program može, ali i ne mora, da poseduje i neke dodatne opcije — mogućnost „zavirljanja“ u sadržaj datoteke

TIMES HELVETICA

Dva pisma koja su najčešće u upotrebi i koja su ujedno najpoznatiji predstavnici grupa serifnih i beszerifnih pisma

sa štampe bez upotrebe za upisivanjem u memoriju, kreiranje makro naredbi za automatsko ponavljanje reči koje se često pojavljuju u nekom tekstu i slično.

Pronaći idealan program ovog tipa za PC računar nije ni malo jednostavno jer, koliko je nama poznato, i ne postoji. Situacija je sasvim razumljiva, jer je teško očekivati da bi ovako

„siromašan“ program imao dobru prođu na tržištu. Možda je najbliži po svojim performansama „star“ IBM-ov „Personal Editor“, ali njegova upotreba ne rešava problem do kraja.

Dakle, ako je cilj unosa da se na disketi nađe „čist“ tekst, onda treba upotrebiti najjednostavniji i najbrži editor. Čist tekst je, ipak, samo deo posla. Daleko je bolje i brže ako se u toku unosa obavljaju i neke obaveštavaju teškote, koje mogu izazvati od programa za preloj kao se koristi. Da bismo objasnili o čemu se radi, moraćemo opet da krenemo od kraja i započnemo veliku temu oko koje se sve i okreće — formatiranje teksta.

Pogled na stranu

Ko li ste ikad, čitajući neku knjigu, zapitali koliko faktora utiče da stranica koju gledate izgleda baš tako? Jasno, ne treba mnogo pažnje da se primeti da slova u svim knjigama nisu ista. Ili da je pored između redova veći ili manji, ali veliki broj detalja vam je sigurno promakao. Zapravo, u tome i leže suština grafičkog oblikovanja neke knjige — različite forme stvaraju različite ugođaje kod čitaoca, a da toga nije svestan. Bez obzira što najvise deo štampanog materijala često zadovoljava samo gotu funkciju nosioca informacije, ostaje činjenica da grafičko oblikovanje tretirano kao umetnost može presudno da utiče na zadovoljstvo čitanja.

Pokušaćemo da racionalno (dakle „hladno računarski“) sagledamo koji elementi čine stranu, ograničavajući se, za sada, na najjednostavniji slučaj — stranicu knjige koja sadrži isključivo pasuse teksta. Pri tome se, jasno, nećemo baviti estetskim kriterijumima, već mogućnošću realizacije pojedinih zahteva na računaru.

Prv element je format strane. Iako je format snažan kreativni element, uglavnom se iz veoma „prizemnih“ razloga (odpad pri isecanju iz velikih tabaka) koriste nekoliko standardnih formata. Najčešće je to B5, koji je gotovo standard za udžbenike i slična izdanja i A5, koja uglavnom koriste oni kod kojih estetika nije naročito da se pojavi i neki nesterodni format. Se onda uglavnom radi ili o blagom dodatnom opsecanju neka od pomenuta dva, ili o luksuznijem izdanju deo se format do kraja poštoje kao kreativan element.

Sledeći element su margine teksta, odnosno površina koju tekst na ukupnoj strani zauzima kao i pozicija numeracije strane i eventualnog dodatnog teksta vrhu ili dna. Karakteristično je da se leva i desna strana knjige ponašaju kao slika u ogledalu pa je na levoj strani (ili parnoj strani, kao što se često naziva) leva margina jednaka desnoj margini na desnoj (neparnoj) strani. Što se slova tiče, strane se mogu slagati identično, s obzirom da je sama površina teksta ista, i kasnije pri montaži izvršiti odgovarajuće pomeraње. Međutim, pozicija broja strane koji je retko kad u sredini, već se „meta“ iz levog ugla na levoj strani u desni ugao na desnoj strani, zahteva da se strane različito tretiraju. (Napomena: argument da se i ovaj deo može obaviti u montaži odbacujemo jer odstupa od postavljenoj onoj cilja — kada je strana složena, onda je složena u potpunosti)

Četvrti element su unutarnje margine teksta, tek tu nas čeka pravi posao.

Koliki razmak treba da bude od vrha strane do naslova poglavlja, ili će možda naslovi poglavlja biti na posebnim stranama? Položaj naslova, podnaslova i sličnih redova teksta takođe zahteva veoma precizno definisanje razmaka u odnosu na tekst sa gornje i donje strane, kao i da li će biti centrirani ili ne.

Posus kao celina zahteva preciznu definiciju: da li je prvi red uvučen unutar margine ili je izbačen van, ili je prva linija na margini, a ostatak pasusa uvučen. Koliki je razmak između pasusa? Unutar pasusa je situacija još komplikovanija. Koliki je linijski razmak? Da li ste znali da veći linijski razmak čini deo slova izgledaju veći i obrnuto? Pasus može biti poravnat na tri načina — u takozvani levi, desni ili donji počinju na levoj margini, a završavaju se, u zavlačenoj od dužine poslednje reči, desni brok (desna margina poravnata, a leva slobodna) ili obostrani blok

18 računari 36 • mart 1988.

Franklin Goth. Book Ital.

Schrift hat vornehmlich die Aufgabe, lesbar zu sein; nur dadurch erfüllt sie

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ÄÖÜ 1234567890
abcdefghijklmnopqrstuvwxyzäöü
? ! " # \$ % & ' () * + , - . : ;

Franklin Goth. Med. Ital.

Schrift hat vornehmlich die Aufgabe, lesbar zu sein; nur dadurch erfüllt sie

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ÄÖÜ 1234567890
abcdefghijklmnopqrstuvwxyzäöü
? ! " # \$ % & ' () * + , - . : ;

Franklin Goth. Orig. Schindow

Schrift hat vornehmlich die Aufgabe, lesbar zu sein; nur dadurch erfüllt sie

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ÄÖÜ 1234567890
abcdefghijklmnopqrstuvwxyzäöü
? ! " # \$ % & ' () * + , - . : ;

Franklin Goth. Ital.

Schrift hat vornehmlich die Aufgabe, lesbar zu sein; nur dadurch

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ÄÖÜ 1234567890
abcdefghijklmnopqrstuvwxyzäöü
? ! " # \$ % & ' () * + , - . : ;

Franklin Goth. Ext. Cond.

Schrift hat vornehmlich die Aufgabe, lesbar zu sein; nur dadurch erfüllt sie ihres Zweck, das

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ÄÖÜ 1234567890
abcdefghijklmnopqrstuvwxyzäöü
? ! " # \$ % & ' () * + , - . : ;

Šrok izbor za đubok džep - katalogi pisama naponizantiji proizvođača foto slog uređaja sadrže po stotinjak stranica sa različitim pismima kao što je ova

TEPAV TEPAV

Slika A. „Nezgodni“ sudari slova - Poštujući razmak koji je difinisan širinom slova, neke kombinacije mogu da daju loš izgled odvojenosti slova od reči. U desnom primeru je (namerno prenašajeno) izvedeno „podsecanje“ (eng. kern) koje pri većim gradacijama neapodno da bi se popravilo utisak

(redovi počinju na levoj margini a poslednja slova izabrati razdvajanje reči na slovima na kraju reči da bi poravnavanje desne margine izazvalo što manji poremećaj) razmaka između reči i slova u redu, ali i ne mora.

Izbor pisma

Izboru pisma, odnosno vrsti slova koja će biti upotrebljena, posvećuje se posebna pažnja. Kreacija novih pisama je vrlo strogo umetnička disciplina u kojoj nije ni malo lako izmisliti nešto novo. Do danas je napravljeno nekoliko hiljada različitih pisama, koje se po nekim karakteristikama koje ih povezuju dele u familije. U svakodnevnoj upotrebi je, pak, svega nekoliko pisama koja se smatraju osnovnim (u pitanju nije netaštvošt, već cena jednog pisma za foto-slog). Najčešći su tajma, koji spada u grupu takozvanih serifnih pisama (serife su mala zaočnjavanja koja čine stupu ili vrh slova), i helvetika iz grupe beserifnih pisama (sans serifna pisma od francuskog — bez serifa) kod kojih svaki stub slova završava ostru.

Određivši pismo, namo, zapravo, uradili ništa konkretno. Oblik pojave svakog pisma može da varira u nekoliko elemenata: veličini, univo zacrnjenja (zadebljanja), deformisanosti u vertikalnom ili horizontalnom pravcu, zaočnjavanju ili obnošenosti u odnosu na vertikalnu itd. U praksi se ipak gotovo redovno variraju samo prva dva elementa.

Veličina pisma, iako bi trebalo da bude nešto što se može veoma precizno izmeriti, nije ni to blizu tako jednostavno pitanje. Koje slovo meriti? Ako uzamemo za meru veličinu, racimo veličinu slova A, onda ćemo dobiti vrednosti iz koje se ne može zaključiti ništa o visini malog slova a i

monoton

Slika B. Prvo slovo o u ređi je tačno poravnato po baznoj liniji ostalih slova (m,n) pa ipak deluje kao da je izdigđuto. Obične pri dnu slova zahtevaju blago spuštanje da bi se optički postigao utisak poravnatosti (ostala dva o i u su spuštani kako treba)

obrnuto. Svako pismo ima svoju proporciju, a stvar se dodatno komplikuje kod spuštanih slova kao što su g i i itd. Veličina pisma je mera koja se nastala još u počecima štampe i bazirala se na tadašnjoj tehnoloziji. Svako slovo i kao reljef bilo izrađeno na drvenoj ili metalnoj pločici i njihovim slaganjem se dobijao red. Veličina pločica je bila takva da se naslanjajom jednog reda na drugi dobijao linijski razmak koji je bio karakterističan za određenu veličinu pisma. Zato je upravo ova veličina usvojena kao mera koja se izražava u takozvanim tipografskim tačkama ili punktima. Punkt se dobija deljenjem inča na 72 dela, ali zbog razlike u američkom i ranijem evropskom inču postoje dve različite mere. Jedna je američka, takozvana „pika-point“ i druga evropska i kod nas u upotrebi „ciocero-punkt“ koja je približno 6% veća. Pika i cintero su veće jedinice i u oba sistema ih čini 12 punkt (points).

S obzirom da se danas linijski razmaci često biraju nešto veći nego što pismo zahteva, veličina pisma na gotovom materijalu je postala egzaktno nemerljiva i obično se procenjuje. Na osnovu veličine lika slova moguće je odrediti veličinu pisma samo ako se tačno zna koje je pismo u pitanju i ko je proizvođač foto-sloga na kome je tekst siagan, jer se isto deklarisanu veličinu, doduše naznato, razlikuju od proizvođača do proizvođača.

Nivo zacrnjenja odnosno zadebljanja pisma se obično izražava u stepenima kao što su crno, polucрно, belo, svetlo itd.

Zaočnjavanje pisma se obično označava kao italik forma ili kurziv, ali treba praviti razliku između ova dva pojma. Na primer, tajma italik, iako je zaočnjen, ne dobija se prostim zaočnjavanjem običnog tajma (pogledajte na primer malo slovo f).

Iako smo to za ovako kratko vreme pobrojali mnogo elemenata koji utiču na ukupan izgled strane, spisak njih ni približno iscrpljen. Čitavu novu temu čini zalaženje u najmanje detalje izgleda i pozicioniranja svakog pojedinačnog slova, pa sam ovu temu saslavljujemo sa nekoliko detalja na slikama A, B i C.

Slike A, B i C

Za tehničkog urednika

U prvom tekstu iz serije smo rekli da tehnički urednik obeležava tekst dajući sve relevantne napomene za slaganje. S obzirom da je izbor širok, pogledaćemo kako to u praksi izgleda.

Uvek na početku rukopisa se zadaje glavna napomena koja ima važnost za ceo tekst i može izgledati ovako:

„Tekst slagati helvetikom, 10/12 pt, na 20 ciera

Ovakva napomena, iako deluje prilično nekompletna, uglavnom je dovoljna jer se u praksi mnoge stvari podrazumevaju (vrednost 10/12 određuje veličinu pisma — 10 pt, i linijski razmak — 12 pt). Tako se, recimo, podrazumeva da će tekst biti slagani poravnato po obe margine sa deobom reči, jer je to najčešći oblik koji se koristi.

„Problemi“ nastaju unutar teksta. Svaka lokalna promena koje odstupa od generalne napomene mora biti obeležena pa tako svaki naslov, podnaslov, uvučeni pasus i slično dobijaju napomenu o tome kako se slažu. Malo složeniji slučaj dobijaju čitavu šumu teksta po marginama. Logika obeležavanja sledi logiku zadanja komandi na foto-sloju.

Analizom ovih oznaka može se lako ustanoviti da za jedan tekst, bez obzira na njihov broj, broj kombinacija koje se pojavljuju i nije tako velik, iako će uz svaki naslov pisati da treba da bude složen recimo veličinom 18pt, crno, malo je verovatno da će biti naslova istog nivoa značenja koji će se slagati drugačije. Dakle, svi naslovi poglavlja se uvek slažu na isti način, svi podnaslovi takođe, i tako dalje. Tekst se uvek sastoji iz određenih logičkih celina kojih nikad nema puno i koje se kroz čitav tekst tretiraju grafički na isti način.

Obeležavanje teksta

Da bismo pokazali kako se problem obeležavanja teksta može posmatrati na drugačiji način, počemo sa konkretnim upoznavanjem programa za prelom „Ventura Publisher“.

Prvo što treba reći je da je na računaru, kada je tekst u pitanju, jedina „uhvatljiva“ i tačno definisana celina pasus. S obzirom da svi programi koji formiraju tekst uzimaju slobodu da raspodeljuju reči po redovima, pasus je jedina celina koja uvek ostaje, s obzirom da se jedina i obeležava pri unosu (najčešće tasterom RETURN).

Podela teksta na pasuse je sasvim zgodna sa stanovišta slaganja jer se kao pasus može obeležiti proizvoljno dugačak (ili kratak) tekst koji predstavlja logičku celinu. Tako se pasusom smatra i naslov od samo jedne reči. Pasus, ujedno, predstavlja i najmanju celinu podložnu promeni formata. Korisnih posledica ovakve organizacije ima mnogo više nego mana. Za svaki se pasus u tekstu može zadati čitav niz parametara koji određuju kako će biti složen:

- pismo i veličina,
- linijski razmak unutar pasusa,
- razmak između dva pasusa,
- dodatni razmak iznad i ispod pasusa,
- uvlačenje/izvlačenje prvog reda,
- lokalna promena leve i desne margine,
- tip poravnavanja (levi, desni, obastrani blok),
- razdvajanje reči na kraju reda,
- fino podešavanje razmaka između slova i reči, i

Čitav niz specijalnih efekata koji uključuju:

- Podvlačenje/nadvlačenje/ukoviravanje
- jednom do tri linije sa proizvoljnim debelinama i rastojanjima kao i tipom rastera,
- veliko početno slovo koje zauzima proizvoljni broj redova po visini itd.

Pri tome se skup ovih karakteristika naziva „TAG“ (engleski tag — privratak, najčešće ćudilica koja se kači na nešto radi obeležavanja) i dodeljuje mu se neko proizvoljno ime. Tako tag „NASL-1“ može definisati sve parametre po kojima se slaže naslov poglavlja, „NASL-2“ parametre za podnaslove itd.

Dakle, postupak obeležavanja teksta se može odvijati i na neki način unazad. Nije potrebno odrediti kojim će pismom biti ispisani naslovi, pa zatim kroz tekst unositi odgovarajuću napomenu, već je dovoljno samo obeležiti neko slovo znakom sve naslove koji će biti složen na isti način. U praksi je za to dovoljno i jedno slovo, pa se uz sve naslove poglavlja upisuje slovo A, uz sve podnaslove slovo B i tako redom.

Ovakvo jednostavno obeležavanje ima mnogo prednosti. Jedna od značajnih je činjenica da ne predstavlja opterećenje pri unosu teksta i može se obaviti odmah u prvih fazama, slova ili oznake koje se koriste moraju se na neki način razlikovati od teksta, što zadiru u pitanje editora za unos kome ćemo se još jednom vratiti.

Najvažnije su pak sledeće prednosti.

- Pošto se već pre unosa zna koji se tagovi koriste, moguće je u računaru uneti karakteristike za svaki od njih i na nekom kraćem uzorku teksta proveriti izgled. Kraj unosa će tako biti sačekan sa već pripremljenim izgledom teksta.

- Menjanje neke od karakteristika jednog taga automatski se reflektuje na sve pasuse koji su njime obeleženi, pa provera raznih karakteristika spada u domene igrice a ne multikontrolnog posla

- Skup tagova sa svim karakteristikama se može sačuvati kao STIL slaganja i zatim lako primeniti na bilo koji drugi tekst — svi pasusi koji su obeleženi istom oznakom kao u ranijim tekstovima automatski se slažu na isti način. Tipika izdanja se na ovaj način slažu izuzetno brzo!

U toku unosa

Ogledno da čitav sistem stimuliše što sistematičniji i dosledniji pristup formatiranju teksta, što je i inače čiji. Jasno da se onda ne može očekivati lako snalaženje u situacijama koje „ispadaju“ iz sistema. Lokalna promena unutar pasusa kao što je promena pisma, zacrnenja, podvlačenja i slično, o kojima nije bilo posebno reči, takođe se lako uklopuju u sistem, jer zahtevaju ukoviravanje dela pasusa i dalje samo sa odgovarajućom oznakom, čiji je smisao kroz ceo tekst isti. Sistem „pada“ u situacijama u kojima se montažer u praksi snalazi na neki neprincipijelan način. Tipičan primer je slučaj sa višekolonskim tipisom koji svakodnevno možete sresti u našim novinama. Jedan od principa u slaganju koji se striktno poštuje je da više kolona moraju da završavaju na istoj visini, odnosno da moraju biti poravnate na dnu strane. A šta ako je tekst u nekoj koloni malo kraći pa nedostaje red ili dva ili pak ima nešto viška koji ne može da stane? U ručnoj montaži se stvar „lako“ doteruje — redovi se kraju (ili, ako je montažer izuzetno raspoloživ, u celoj koloni) malo razdrže ili sabiju što nije tako uočljivo kao kraća ili duža kolona koja bi delovala u najmanju ruku smešno.

Iz situacije nije nemoguće isplivati ni uz pomoć „Venture“. Poslednjem pasusu se dodeli novi tag u kome se definiše samo nešto već ili manji prored, ali bi ovakav rad već nakon nekoliko strana proizveo čitavu „šumu“ tagova jer kako je dokazano — ne postoje dva stupca u kojima je potrebno identično podešavanje razmaka.

Ako su nakon ovog objašnjenja stvari malo jasnije, možemo se vratiti tamo gde smo stali — na unos teksta. Ogledno je da se, s obzirom na jednostavnost oznaka, čitavo obeležavanje teksta može poveriti daktilografu koji tekst unosi. Kočna forma koja je potrebna „Venturi“ je ASCII datoteka u kojoj je tag zidat na sledeći način:

NAZIV TAGA = Sledi tekst pasusa

Ali o tome u sledećem broju.

Zoran Životić

Sto grama pameti

Diskusija je bila vrlo ozbiljna i na visokom nivou. Odlučivalo se o tome kako da se normira rad jedinog programera u agrokombinatu: Žike Nesrečkovića, poznatijeg po nadimku Fiopi.

— Ja se protivim da na ovakav način utvrđujemo normu za posao o kome ne znamo ništa — rekao je član komisije Steva, poljoprivredni tehničar. — Nije to kombajner ili traktorist. Znamo da se voda ne meri centimetrima, da se pamet ne meri kilogramima, a šta znamo o programiranju? Kakva je mera za programerski rad?

Umešao se Radoje magacioner: — Ne mistificiraj! Niko od vas ne zna kakav je moj posao u magacinu, pa št mi ipak odrezati normu, i to oho! Oh! smo i mi za kilobajt. Da mu oredimo kilobajt ili dva dnevno, pa neka čovek radi.

— Kako da mu izmerimo te kilobaj- te? — pitao je predsednik komisije. — Na kantar?

— Ja bih pre da norma bude izražena u stavkama — predložio je Đorđe knjigovodja. — Koliko stavki, koliko bodova . . .

Predsednik vezi je prekinuo: — To nema veze! Programer je jedino, a operater drugo. Pobrkao si ioničke.

— Pa onda neka mu norma bude broj programa. Ja mislim da bi bilo pravo da zacrtao jedan program dnevno.

Poslednji predlog potekao je od Vide daktilografkinje:

— Ne može to tako! On kucka, kucka, pa stane i kaže: „Gotov program!“ Otkud mi znamo da nije uradio samo pola programa? Ja bih njemu odredila normu na redove, kao svakoj daktilografkinji, a redovi mogu da se broje, nema zabušavanja.

— Možda je to najbolje — složio se predsednik. — Koliko redova dnevno predlažete?

— Vida ima normu od trista redova, ali njoj diktiraju drugi, ona ništa ne izmišlja. Drug Znesrečković sve sam izmišlja, pa bi bilo pravo da se taj broj smanji na polovinu — formulaiso je predlog Radoje.

Predlog o sto pedeset redova zamalo da bude jednoglasno usvojen. To mi je pre neki dan ispričao Žika Fiopi ičino. I dodao:

— Jedva sam uspeo da ih ubedim da se tako ne može. Inače, da su izglazali takvu odluku o normi, znaš šta bih uradio? Napisao bih program koji generiše sto pedeset programskih redova sa slučajnim naredbama. Čak sam smislio i algoritam koji obezbeđuje logičan sled naredbi, a ne njihovo besmisleno nizanje . . .

— Ti i Steva u celoj toj gužvi bili ste jedini koji ste razmišljali o besmislenosti — rekao sam mu. — Ali, neka te to ne brine. U nas se na sličan način ne odlučuje samo o programerskoj normi.

Izlog knjiga

Đorđe Zrnić i Dragoljub Savić

Dejan Ristanović

Simulacija procesa unutrašnjeg transporta

Mašinski fakultet Univerziteta u
Beogradu, 1987.
Strana 237, 7.900 dinara*

Uz obilje opšte računarske literature, na domaćem tržištu se sve češće pojavljuju specijalizovani priručnici koji se bave primenama računara interesantnim mahom za uži krug eksperata iz određene oblasti. Knjiga „Simulacija procesa unutrašnjeg transporta“, iako po nameni udžbenik za predmet „Projektovanje fabrika“ koji se izučava na Mašinskom fakultetu, zapravo je zanimljiv priručnik za svakoga ko se bavi transportnim problemom.

Sadržaj priručnika je orijentisan na primenu metoda teorije čekanja i simulacije u procesu projektovanja transportnih i skladišnih sistema. Primena ovih modela omogućava dinamički pristup rešavanju problema kretanja materijala i koristi se kao dopuna drugim grafičkim i analitičkim metodama — uz klasične metode u knjizi je izloženo i nekoliko originalnih modela uz objašnjenje njihovih dobrih strana i nedostataka. Svaki model je propraćen odgovarajućim programom koji je pisan na pomalo arhaičnom i prilično nestrukturiranom fortranu. Količina komentara u raznim programima je minimalna što znači da će potencijalni korisnik morati da pročita i značajne segmente okolnog teksta da bi razumeo koji problem program rešava i kako se njime rukuje. Finalni program („Simulacija procesa komisioniranja“) je zapravo delo nekoliko studenata Mašinskog fakulteta među kojima je autor ovoga teksta pronašao i jednog svog kolegu iz gimnazije — baš je svet mali!

Knjiga „Simulacija procesa unutrašnjeg transporta“, sve u svemu, predstavlja interesantan udžbenik za sve one koji se zanimaju projektovanjem transportnih sistema; korišćenje ovog udžbenika, ipak, zahteva određeno matematičko predznanje, što znači da će početnici uz ovaj udžbenik morati da nabave i raznu drugu literaturu.

*Cena u knjžarama. U skriptarnici Mašinskog fakulteta cena se može dobiti sa 30 odsto popusta.

20 računari 36 • mart 1988.

Mašinsko programiranje na mikroprocesorima Z80 i 6502



Tehnička knjiga, Beograd 1988.
Strana 256, cena 16.000 dinara

Neko ko prvi put pročita naslov ove knjige, a ima bar nekakvu predstavu o programiranju, možda će sa nevericom upitati: „Zar sve to može stati u jednu knjigu?“. Odgovor je: „Može, i te kakol!“: Stvar je, jednostavno, u pristupu.

O mašinskom programiranju samo jednog mikroprocesora može se napisati više knjiga, ali, uz odgovarajuću umešnost autora i ispravan pristup složenoj materiji, može se i na 256 stranica jedne knjige izložiti opis dva mikroprocesora, a da se time ni najmanje ne izgubi na kvalitetu.

Reč je o dva, svakako najpopularnija osobitna mikroprocesora: Z80 i 6502. Ko je imao prilike da programira oba, dobro

zna da, i pored nekih opštih sličnosti, među njima ima i prilično velikih razlika. Mnoge ideje koje je jednostavno ostvariti na jednom, nije baš tako lako realizovati na drugom, i obrnuto. Autor ove knjige to svakako najbolje zna. Dejana Ristanovića našim čitaocima nije potrebno posebno predstavljati. Moramo, međutim, reći makar to da iza knjige o kojoj govorimo stoji višegodišnje iskustvo i kilobajti ispisanih programa. Već ta činjenica dovoljna je garancija za kvalitet onoga što se u knjizi nalazi.

Celina je podeljena u pet poglavlja: Mašinsko programiranje, Mikroprocesor Z80, Mikroprocesor 6502, Biblioteka programa i Dodaci.

U poglavlju „Mašinsko programiranje“ upoznajemo se, na vrlo spontan i zanimljiv način, sa osnovnim pojmovima. Šta je to mikroprocesor, šta je binarna logika i šta je mašinski program... početnik će pročitati sa velikim interesovanjem, a iskusniji programer sa velikim zadovoljstvom. Naći ćemo tu i kratak uvod u rad sa asemblerima, a posebna pažnja posvećena je organizaciji memorije računara.

U poglavlju „Mikroprocesor Z80“ dat je sažet opis strukture samog mikroprocesora, kao i prikaz svih glavnih grupa instrukcija.

Na analogni način prikazan je i mikroprocesor 6502 u sledećem poglavlju.

Četvrto poglavlje rezervisano je za neke standardne primere mašinskih programa, svrstane u zajedničku biblioteku. Tu se, između ostalog, mogu naći rutine za vremensko kašnjenje, množenje, deljenje, konverziju brojnih sistema, pretraživanje, sortiranje itd.

U poslednjem poglavlju čitalac može naći tabelarni pregled svih instrukcija oba mikroprocesora, kao i njihovo programiranje na najpopularnijim kućnim računarima. Poseban „specijalitet“ predstavlja osvrt na nedokumentovane instrukcije, o kojima se inače retko piše.

Naročito bismo pohvalili tehničko rešenje knjige. To je jedno od onih izdanja koje svako od nas želi da ima, makar i kao ukras u svojoj biblioteci. Korice su odlično urađene. Svaka stranica odaje prijatan vizuelni utisak, jer su primeri i ilustracije vrlo lepo izdvojeni od teksta. Ova se pohvala jedino ne odnosi na dodatke sa tabelarnim prikazom instrukcija, koje, moramo reći, deluju pomalo nepregledno.

Knjigu preporučujemo svima koji se bave mašinskim programiranjem, bez obzira na iskustvo koje imaju. I, razume se, preporučujemo je onima koji tek žele da nauče mašinsko programiranje. Možda u toj knjizi neće naći baš sve ono što ih interesuje, ali će van svake sumnje dobiti izvanrednu osnovu za dalji rad.

Jovan Skuljan

Originalni i klonovi

Prošlu računarsku godinu pamtimo eo „tradicionalno“ skupim originalima i sve jeftinijim klonovima računara tipa PC. Cene, doduše, nisu baš najpristupačnije svim domaćim zaljubljenicima u kompjutere, ali nekako se snalazimo... Standard koji je nametnuo IBM i dalje se dobro drži. Svi proizvođači ga, uglavnom, poštuju, što je, svakako, mudar izbor.

Računar PC je još uvek prepoznatljiv, sa odvojenom tastaturom, zasebnim monitorom, dva flopi drajva ili hard diskom i gomilom standardnih kartica za proširenja.

Ono što bismo mogli okarakterisati možda kao hrabar korak jeste sve izraženija tendencija ka narušavanju hardverske kompatibilnosti sistema, ugradnjom novih integrisanih kola kao zamenu za hrpu „klasičnih“. Na primer, sve češće se može sresti minijaturna hercules kartica na jednom čipu, ili čak osnovna procesorska ploča na jednom čipu (!)

Razume se, funkcionalna kompatibilnost je očuvana, jer inače to više ne bi bio PC klon.

Drugi pravac razvoja klonova ide na povećanje brzine rada, što je posledica pojave sve boljih i bržih komponenti. Retki su sada klonovi XT-a koji rade na klasičnom kloku od 4,77 MHz. Novi standard je sada praktično 10 MHz (i 8 MHz se već retko sreće). Razume se, iz razloga opreznosti, svi klonovi uglavnom imaju mogućnost izbora kloka i jednostavnog vraćanja u osnovni režim rada. Slično važi i za AT klonove, gde je kloak od 6 MHz već odavno prevladan. Sada se ide do 12, pa i 16 MHz.

Grafičke mogućnosti računara i dalje se zasnivaju na CGA, Hercules i EGA standardu.

Katalog 96 PC klonova koji ovde donosimo, objavljen je u časopisu „MC“ i odnosi se na nemačko tržište, ali je i za naše kupce i te kako interesantan.



Abaco 18 E (Abacom) • 8086/4.77 MHz • Interfejs za monitor • Set znakova IBM • Magistrala 16 bita • RAM 256K • Flopi 360K • Tekst 80×25 • Grafika 640×200, opcija 720×348 • Proširenja: RAM do 2 MB, hard disk do 60MB • 884 DM.



Abaco 18 H (Abacom) • 8086/4.77 MHz • Interfejs za monitor i štampač • Magistrala 16 bita • RAM 640 K • Flopi 2×360K • Hard disk opcija do 120 MB • Tekst 80×25 • Grafika 640×200, opcija 720×348 • Proširenja: RAM do 8 MB, hard disk, časovnik, serijski interfejs itd. • 1197 DM.



Abaco 16 (Abacom) • 8086/4.77; 8 MHz • Interfejs, serijski, paralelni, za igre • Magistrala 16 bita • RAM 640K • Flopi 2×360K • Hard disk 30–120 MB opciono • Tekst 80×25 • Grafika 640×200, opcija 720×348 • Proširenja: Hard disk, RAM do 8 MB • 1710 DM.



Abaco 16-286 (Abacom) • 80286/6; 10 MHz • Interfejs za štampač i monitor • Magistrala 16 bita • RAM 1024K • Flopi 2×1.2 MB • Hard disk 30–120 MB opciono • Tekst 80×25 • Grafika 720×348 • Proširenja: RAM do 8 MB • 2822 DM.



AD-Turbo XT (AD Computertechnik) • 8 MHz • Operativni sistem MS-DOS 3.2 i GW-Basic • Interfejs: serijski, paralelni i za igre • Set znakova: nemački, Bolt, ASCII • Magistrala 16 bita • RAM 640K • Flopi 360K • Hard disk do 180 MB • Tekst 80×25 • 132×44 • Grafika 720×348 • 999 DM.



ARC-AT-Turbo (American Research Corp.) • 80286/8 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: RS 232 C, Centronics • RAM 512K • Flopi 1.2 MB • Grafika hercules-monochrome • Proširenja: RAM do 1 MB na ploči, 80287, hard disk 160 MB • 2995 DM.



ASC-XT-L (ASC Electronic) • 8086/4.77; 8; 10 MHz • Operativni sistem DOS • Set karaktera IBM • Magistrala 16 bita • RAM 256K • Flopi 360K • Tekst 80×25 • Grafika 600×200 • Osnovna verzija bez monitora • Proširenja: MI, hard disk, EGA, koprocesor, BTX adapter, IO karta, dodatni RAM • 998 DM.



AST Premium/286 (AST Research Inc., USA) • 80286/6; 8; 10 MHz • Operativni sistem MS-DOS 3.2, nemački • Magistrala 16 bita • RAM 1 MB • Flopi: max. 2 • Hard disk 20/40/70 MB • Tekst IBM kompabilan • Grafika 720×350 ili 640×200 • Osnovna verzija: 1 MB RAM, hard disk do 20 MB, EGA karta • 4850–10795.



PC 1 (Aartr) • 8086/4.77; 8 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: paralelni, serijski, miš • Set znakova ASCII nemački/IBM • Magistrala 16/24 bita • RAM 512–640K • Flopi 360K • Hard disk 20 MB • Tekst 80×25 • Grafika 720×348, u boji 640×350 • Proširenja: RAM do 640K.



A-200 II (Caron) • 4.77; 7; 15.9 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: paralelni i za igre • Set znakova: nemački • Magistrala 16 bita • RAM 256–640 K • Flopi: 2×5.25 inča • Hard disk 20 MB • Grafika 720×350 monochrome • Osnovna verzija: 256K RAM • 4315–7758 DM.



Amiga 500 (Commodore) • 7 MHz • Operativni sistem Amiga-DOS, intulacija • Interfejs: Centronics, RS 232 • Set znakova nemački/ASCII • Magistrala 16 bita • RAM 512K • Flopi: unutrašnji i spoljašnji, po 880K • Tekst 80×25, 80×25 • Grafika 320×200 do 640×512 • Proširenja: RAM do 1 MB, hardverski časovnik • 1259 DM.



Copro V 286-1 (Copro) • 80286/6; 10 MHz • Operativni sistem MS-DOS • Interfejsi: serijski, paralelni • Set znakova IBM • Magistrala 16 bita • RAM 1 MB • Flopi 1.2 MB • Hard disk 20 MB • Grafika 320×200 ili 640×200 • Proširenja: koprocesor 80287, hard disk 30–40 MB • 3980 DM.



DC-16 XT/2 (Dawicontrol) • 8086/4.77 MHz • Operativni sistem MS-DOS 3.1 • Interfejsi: 2× Centronics, RS 232 C, igre • Set znakova IBM • Magistrala 16 bita • RAM 256K • Flopi 2×360K • Tekst 80×25 • Grafika 720×348 hercules • Proširenja: hard disk do 20 MB, RAM do 640K • 1590 DM.



DC-16 AT/30 (Dawicontrol) • 80286/6; 10 MHz • Operativni sistem MS-DOS 3.1 • Interfejsi: Centronics-paralelni/video-TTL • Set znakova IBM • Magistrala 16 bita • RAM 512K • Flopi 1.2 MB • Hard disk 30 MB • Tekst 80×25 • Grafika 720×348 hercules • Proširenja: kao XT/2 • 3990 DM.



CDI AT60/286 (CDI) • 12 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: Centronics, V.24 • Set znakova nemački • Magistrala 16 bita • RAM 512K • Flopi 1.2 MB • Hard disk do 80 MB • Grafika hercules • Proširenja: hard disk do 80 MB, flopi 3.5 inča, EGA, monitor 14 inča, 1 MB RAM, koprocesor 80287 • 3357 DM.



Amiga 2000 (Commodore) • 68000
• Operativni sistem Amiga-DOS, opcija MS-DOS 3.2 • Interfejs: V.24, Centronics, miš, itd. • Set znakova nemački • Magistrala 16/32 bita • RAM 1—8 MB • Flopi 80K (do 4k) • Tekst 80x25 ili 80x62 • Grafika max. 640x512, max. 4096 boja • Osnovna verzija: Amiga Basic, tastatura, miš • Proširenja: 8088 ploča za MS-DOS, 80286 ploča, SCSI kontroler, ...



PS 20-II (Commodore) • 8086/4,77
MHz • Operativni sistem MS-DOS 3.2 • Interfejs: V.24 RS 232 C, Centronics • Set znakova nemački • Magistrala 16 bita • RAM 512K • Flopi 360K • Hard disk 20 MB • Tekst 80x25, 132x25, 132x44 • Grafika 640x200 u 16 boja, 720x382 monohrom • Osnovna verzija: nemačka tastatura, monohrom TTL monitor • Proširenja: 5 slotova, 8087 koprocesor, RAM do 640K itd.



PC 10-II (Commodore) • 8086/4,77
MHz • Operativni sistem MS-DOS 3.2 • Interfejs: V.24, RS 232 C, Centronics • Set znakova nemački • Magistrala 16 bita • RAM 512K • Flopi 2 • 360K • Hard disk opciono • Tekst 80x25, 132x25, 132x44 • Grafika 640x200 u 16 boja, 720x382 monohrom • Osnovna verzija sa nemačkom tastaturom i TTL monitorom • Proširenja: 5 slotova, 8087, RAM do 640K.



PC I (Commodore) • 8086/4,77
MHz • Operativni sistem MS-DOS 3.2 • Interfejs: serijski, paralelni • Set znakova nemački/ASCII • Magistrala 16 bita • RAM 512K • Flopi 360K • Tekst 80x25 • Grafika 720x348 monohrom, 320x200 u četiri boje • Osnovna verzija sa jednim flopijem • Proširenja: slotovi, drugi flopi • 1298 DM.



System 286/20 (DSM Digital-Service) • 80286/6, 10 MHz • Operativni sistem MS-DOS 3.2, GW-Basic • Interfejs: paralelni, serijski • Set znakova IBM • Magistrala 16 bita • RAM 640K • Flopi 1,2 MB • Hard disk 20 MB • Tekst 80x25 • Grafika 720x348 monohrom • Osnovna verzija sa hard diskom od 20 MB • Proširenja: hard disk, EGA, sve standardne PC/AT karte • 3850 DM.



XT-8 (DSM Digital-Service) • 8086/8 MHz • Operativni sistem MS-DOS 3.2, GBI-Basic • Interfejs: paralelni • Set znakova IBM • Magistrala 16 bita • RAM 640K • Flopi 360K • Hard disk 20/30 MB opciono • Tekst 80x25 • Grafika 720x348 monohrom • Proširenja: sve standardne kartice • 1498 DM.



ECD-Advanced (ECD Computertechnik) • 80286/6, 10; 12 MHz • Operativni sistem DOS 3.2 • Interfejs: paralelni, opciono serijski • Set znakova IBM • Magistrala 16 bita • RAM 512K do 1 MB • Flopi 1,2 MB • Hard disk 20 MB opciono • Tekst 80x25 • Grafika 720x348 • Proširenja: interfejsi, grafičke i specijalne kartice • 3873 DM.



ECD-Professional (ECD Computertechnik) • 8086/4,77; 8; 10 MHz • Operativni sistem DOS 3.2 • Interfejs: paralelni, serijski, igre • Set znakova IBM • Magistrala 16 bita • RAM 256—640K • Flopi 360K • Hard disk 20 MB opciono • Tekst 80x25 • Grafika 720x348 • Proširenja: interfejsi, grafičke i specijalne kartice • 1298 DM.



ECD-Modul (ECD Computertechnik) • 80286/6, 10 MHz • Operativni sistem DOS 3.2 • Interfejs: paralelni, opciono serijski • Set znakova IBM • Magistrala 16 bita • RAM 512K do 1 MB • Flopi 1,2 MB • Hard disk 20 MB opciono • Tekst 80x25 • Grafika 720x348 • Proširenja: interfejsi, grafičke i specijalne kartice • 2498 DM.



ECD-Trevel (ECD Computertechnik) • 80286/6, 10 MHz • Operativni sistem DOS 3.2 • Interfejs: paralelni, opciono serijski • Set znakova IBM • Magistrala 16 bita • RAM 512K do 1 MB • Flopi 1,2 MB • Hard disk 20 MB opciono • Tekst 80x25 • Grafika 720x348 • Proširenja: interfejsi, grafičke i specijalne kartice • 3499 DM.



ECD-Modul (ECD Computertechnik) • 8086/4,77; 8 MHz • Operativni sistem DOS 3.2 • Interfejsi: paralelni • Set znakova IBM • Magistrala 16 bita • RAM 256—640K • Flopi 360K • Hard disk 20 MB opciono • Tekst 80x25 • Grafika 720x348 • Proširenja: interfejsi, grafičke i specijalne kartice • 1245 DM.



PC XT Epson PC (Epson) • 8086/4,77; 10 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: RS232, Centronics • RAM 640K • Flopi 360K • Hard disk 20 MB • Grafika 640x200 • Osnovna verzija: 1 flopi, MGA karta, monitor 12 inča • Proširenja: TTL, MS-DOS 3.2, RAM do 8 MB • 3330 DM.



GAT-8/10/12 kompakt (Klaus Gorny) • 8; 10; 12 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: paralelni i serijski • Magistrala 16 bita • RAM 512K • Harddisk 20 MB opciono • Grafika 640x200 • 1280x1024 • Proširenja: grafička karta za grafičku visoku rezoluciju u boji • 2398 DM.



ATS Turbo (Import) • 80286/6; 10 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: Centronics • Set znakova IBM kompatibilan • Magistrala 16 bita • RAM 512K do 1 MB • Flopi 1,2 MB • Tekst 80x25 • Grafika 720x348 hercules • Proširenja: HEGA karta • 1995 DM.



XTS Turbo (Import) • 8086/4,77; 8 MHz • Interfejsi: sistem MS-DOS 3.2 • Interfejsi: Centronics • Set znakova IBM kompatibilan • Magistrala 16 bita • RAM 256K do 640K • Flopi 360K • Tekst 80x25 • Grafika 720x348 hercules • Proširenja: HEGA karta, drugi flopi • 1095 DM.



ATS 20 Turbo (Import) • 80286/6; 10 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: Centronics • Set znakova IBM kompatibilan • Magistrala 16 bita • RAM 512K do 1 MB • Flopi 1,2 MB • Hard disk 20 MB • Tekst 80x25 • Grafika 720x348 hercules • Proširenja: HEGA karta, drugi flopi • 2995 DM.



PC Mono Pack (Kaypro) • 8086—2,4/7,7; 8 MHz • Operativni sistem MS-DOS • Interfejsi: serijski, paralelni • Magistrala 8 bita • RAM 768K • Flopi 2 • 360 K • Proširenja: 80286 procesorska ploča, koprocesor 8087, EGA karta, hard disk • 3415 DM.



HP Vectra CS (Hewlett-Packard GmbH) • 8086/7; 16 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: serijski i paralelni • Set znakova IBM • Magistrala 16 bita • RAM 640K • Flopi 360K • Tekst 80x25 • Grafika 640x400 • Osnovna verzija: flopi 360K, monohrom monitor • Proširenja: hard disk 20 MB, flopi 3,5 inča, EGA • 4298 DM.



ATC-20 (Hösch) • 6; 12 MHz • Operativni sistem MS-DOS 3.1 • Interfejsi: paralelni, serijski, igre • Magistrala 12 bita • RAM 512K do 1 MB • Flopi 1,2 MB • Hard disk 21 MB • Tekst 80x25 • Grafika 720x348 • Osnovna verzija: monohrom grafička kartica • Proširenja: 132 znakova, super-EGA, IEEE-488, digitalizator, miš • 2890 DM.



Hösch Büro 200 (Hösch) • 6; 12 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: paralelni, serijski • Magistrala 16 bita • RAM 1 MB • Flopi 1,2 MB • Hard disk 21 MB • Tekst 40x40 do 132x44 • Grafika 640x480 do 1056x382 • Osnovna verzija sa tastaturom, super EGA, kartom i EGA monitorom • Proširenja: IEEE-488 kartica, miš • 4998 DM.



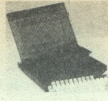
IBM PS/2, Modeli 30—002 (IBM) • 8096/9 MHz • Operativni sistem IBM DOS 3.1 • Interfejsi: serijski, paralelni • Set znakova Codepage 850 • Magistrala 16 bita • RAM 640K do 2.64 MB • Flopi 2x 720K (3.5 inča) • Grafika 640x480/720x400 (APA) • Proširenja: MGCA, mi, 8087—2 • 4793 DM.



Ameco ATF 20—10 (Import) • 80286/6, 10 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: RS232, Centronics • Set znakova IBM • Magistrala 16 bita • RAM 512K do 1 MB • Flopi 1.2 MB • Hard disk 20 MB • Tekst 80x25 • Grafika 720x348 hercules • Proširenja: HE-GA karta • 3989 DM.



PC Colour Pack (Kaypro) • 8086—2/4,77; 8 MHz • Operativni sistem MS-DOS • Interfejsi: serijski, paralelni • Magistrala 8 bita • RAM 768K • Flopi 2x 380 K • Proširenja: 80286 procesorska ploča, kopčopros 8057, EGA karta, hard disk • 4300 DM.



2000 (Kaypro) • 8086/4,77 MHz • Operativni sistem MS-DOS • Interfejsi: RS232 C • Magistrala 8 bita • RAM 768K • Flopi 720K (3.5 inča) • Grafika 640x200 • Proširenja: flopi ili hard disk • 4032 DM.



PC DS-500 (MAI) • 8086/4,77 MHz • Operativni sistem BOSS 6, MS-DOS 3.1 • Interfejsi: 2x RS 232 C, Centronics • Set znakova DN • Magistrala 8 bita • RAM 256K • Flopi 2x 360 K • Hard disk 20 MB • Grafika 630x200 • Proširenja: RAM do 768K • 4390 DM.



XT 18 SLC System Paket 1 (MCI) • 8086/4,77 MHz • Operativni sistem DOS 3.1 • Interfejsi: 1 paralelni, 1 serijski • Set znakova IBM • Magistrala 8 bita • RAM 640K • Flopi 380K • Hard disk 20/30 MB • Tekst 80x25 • Grafika 720x348 • Osnovna verzija: časovnik i TTL monitor 12 inča • Proširenja: flopi, hard disk, monitor, EGA sistem • 1449 DM.



AT 286 (MCI) • 80286/12 MHz • Operativni sistem DOS 3.2 • Interfejsi: 2 paralelni, 2 serijski • Set znakova IBM • Magistrala 16 bita • RAM 1 MB • Flopi 1.2 MB • Hard disk 30/60 MB • Tekst 80x25 • Grafika 720x348 • Čuvana verzija sa časovnikom i monitorom 14 inča • Proširenja: flopi, hard disk, interfejsi, EGA sistem itd. • 3999 DM.



AT 4 SLC System Paket 3 (MCI) • 80286/10 MHz • Operativni sistem DOS 3.1 • Interfejsi: paralelni, serijski, igra • Set znakova IBM • Magistrala 16 bita • RAM 640K • Flopi 1.2 MB • Hard disk 20/30 MB • Tekst 80x25 • Grafika 720x348 • Osnovna verzija sa časovnikom i TTL monitorom 12 inča • Proširenja: flopi, hard disk, interfejsi, monitor, EGA sistem • 2499 DM.



XT 16 PC (MCI) • 8086/10 MHz • Operativni sistem MS-DOS 3.1 • Interfejsi: paralelni, serijski, igra • Set znakova IBM • Magistrala 8 bita • RAM 640K • Flopi 2x380K • Hard disk 20/30 MB • Tekst 80x25 • Grafika 720x348 • Osnovna verzija sa časovnikom i monitorom 14 inča • Proširenja: hard disk, interfejsi, EGA sistem itd. • 2299 DM.



Hermon 20 sbc (MCT) • 66020/12; 16; 20; 25 MHz • Operativni sistem CP/M, P-DOS, OS-2/86 MB • Interfejsi: 2x serijski, 2x paralelni • Set znakova ASCII • Magistrala 32 bita • RAM 128-512K • Flopi: 2 • Hard disk 20-80 MB • Tekst 80x25 • Grafika 1024x1024 • Proširenja: RAM kartica itd. • 4389 DM.



Micromint Power AT 286 (Micromint Computer) • 80286-10/6; 8; 10; 12 MHz • Operativni sistem MS-DOS • Interfejsi: V.24, paralelni, igra • Set znakova nemčki • RAM 512K do 1 MB • Flopi 360K ili 1.2 MB • Grafika hercules • Osnovna verzija bez monitora • Proširenja: hard disk 20 MB, kontroler WD 1003 A2 • 2222 DM.



NCP-AT Baby (Niedermeier) • 80286/6; 8; 10; 12 MHz • Operativni sistem MS-DOS 3.21, GW-Basic • Interfejsi: paralelni • Set znakova IBM • Magistrala 16 bita • RAM 512K do 1 MB • Flopi 1.2 MB • Hard disk 20-150 MB • Grafika 720x348 • Osnovna verzija: hercules karta sa interfejsom za štampač • Proširenja: EGA, HE-GA, monitor 20 inča • 2299 DM.



M 15 (Olivetti) • 80C86/4,77 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: V.24, Centronics • RAM 512K • Flopi 2x 720K • Tekst 80x25, 40x25 • Grafika 640x200, 320x200 • Proširenja: flopi 5,25 inča • 4480 DM.



M 240 (Olivetti) • 8096/4,77; 10 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: V.24, Centronics • RAM 640K • Flopi 2x 720K (3.5 inča) • Hard disk 20 MB • Tekst 80x25 40x25 • Grafika 640x400, 840x400, 840x350 • Proširenja: flopi 5,25 inča • 8959 DM.



OST-D (Debrone) • 8086-2/8 MHz • Operativni sistem MS-DOS 3.21 • Interfejsi: 2 paralelni, serijski, igra • Magistrala 16/8 bita • RAM 256-640K • Flopi 2x 360K • Tekst 80x25 • Grafika 720x348 hercules • Proširenja: Hard disk, grafika u boji • 2568 DM.



MP 286 (Mitsubashi) • 80286/6; 12 MHz • Operativni sistem MS-DOS, Xenix • Interfejsi: RS232, paralelni • Set znakova: sve nacionalne verzije • Magistrala 16 bita • RAM 640K do 5.5 MB • Flopi 1.2 MB • Hard disk 20/40 MB • Tekst IBM • Grafika IBM • 5389 DM.



Acer 910 B (Multitech) • 80286/6; 10 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: 2 serijski, 1 paralelni • Set znakova ASCII • Magistrala 16 bita • RAM 512K do 16 MB • Flopi 1.2 MB • Hard disk 20/40 MB opciono • Tekst 80x25 • Grafika 720x348 ili 640x350 • Osnovna verzija: flopi, hercules karta, monitor 14 inča • Proširenja: 4 slobodna slot, EGA sistem, hard disk • 3995 DM.



ACER 500 S (Multitech) • 8086/4,77; 8 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: serijski, paralelni, igra • Set znakova ASCII • Magistrala 8 bita • RAM 256-640K • Flopi 360K • Hard disk 20 MB • Tekst 80x25 • Grafika 320x200, 640x200 ili 720x348 • Osnovna verzija: 256K RAM, CGA • Proširenja: Monitor, RAM do 640K, jedan slobodan slot • 1499 DM.



Acer 900 B (Multitech) • 80286/6; 10 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: 2 serijski, 1 paralelni • Set znakova ASCII • Magistrala 16 bita • RAM 512K do 16 MB • Flopi 1.2 MB • Hard disk 20/40 MB • Tekst 80x25 • Grafika 720x348 ili 640x350 • Osnovna verzija: 512K hercules, monitor 14 inča • Proširenja: hard disk, RAM do 1 MB na ploči, 5 slotova, EGA • 4495 DM.



Acer 710 B (Multitech) • 8086-1/4,77; 8 MHz • Operativni sistem MS-DOS 3.2 • Interfejsi: serijski, paralelni • Set znakova ASCII • Magistrala 8 bita • RAM 256-768K • Flopi 360K • Hard disk 20 MB • Tekst 80x25 • Grafika 320x200, 640x200 ili 720x348 • Osnovna verzija: 256K, TTL monitor 14 inča • Proširenja: 4 slota, hard disk 20 MB, RAM do 768K, EGA sistem • 2799 DM.



80T-H (Osborne) • 8086/8 MHz •
 Operativni sistem MS-DOS 3.21 •
 Interfejs: paralelni, serijski, igra •
 Magistrala 16/8 bita • RAM 256-
 640K • Flopi 360K • Hard disk 20
 MB • Tekst 80x25 • Grafika 720x348
 • Proširenja: grafički u boji, strešer
 • 3100 DM.



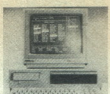
80-D (Osborne) • 80286/10 MHz •
 Operativni sistem DOS 3.21 • Inter-
 fejs: paralelni, serijski • Magistrala
 16 bita • RAM 640K • Flopi 2x 720K
 (3.5 inča) • Hard disk opciono •
 Tekst 80x25 • Grafika 640x400 •
 Proširenja: hard disk, monitor u boji
 • 4664 DM.



07-H (Osborne) • 80286/10 MHz •
 Operativni sistem MS-DOS 3.21 •
 Interfejs paralelni • Magistrala 16
 bita • RAM 512K • Flopi 1.2 MB •
 Hard disk 20 MB • Tekst 80x25 •
 Grafika 720x348 • Proširenja: stri-
 mer, grafika u boji • 4381 DM.



STT 286 (Olivetti electronics) •
 80286/6, 10, 12 MHz • Operativni
 sistem DOS • Interfejs: paralelni,
 serijski • RAM 1 MB • Flopi
 1.2 MB • Hard disk 20-80 MB •
 Grafika 720x348 • Proširenja: 6 slo-
 bodnih slotova • 5950 DM.



Yas (Philips) • 80186/8 MHz • Oper-
 ativni sistem MS-DOS 2.11/3.10 •
 Interfejs: RS 232C, Centronics, miš •
 Set znakova IBM, nacionalni setovi •
 Magistrala 16 bita • RAM 128-
 640K • Flopi 2x 720K (3.5 inča) •
 Hard disk 20 MB • Grafika 640x250,
 640x350 • Osnovna verzija: 2 flopi-
 ja, DOS Plus, MS-DOS, GW-Basic •
 Proširenja: RAM do 640K • 3067
 DM.



Personal Computer P 3102 (philips)
 • 8086/5 MHz • Operativni sistem
 MS-DOS 3.1 • Interfejs: video, serijski
 i paralelni • Set znakova IBM •
 Magistrala 8 bita • RAM 640K •
 Flopi: 1x/2x 360K • Hard disk 20
 MB • Tekst 80x25 • Grafika 640x200
 ili 640x400 • Osnovna verzija: 2
 flopija, monohrom monitor, tastatu-
 ra • Proširenja: kolor sistem, kopro-
 cesor, miš • 3375 DM.



PT-AT (Plantron) • 80286/6, 8 MHz •
 Operativni sistem DOS 3.2 • Inter-
 fejs: 2 paralelni i 1 serijski • Set
 znakova IBM/nemački — DIN/ASCII
 • Magistrala 16 bita • RAM 640K •
 Flopi 1.2 MB • Grafika max.
 720x348 • Proširenja: 80287 kopro-
 cesor, 5 slobodnih slotova, RAM do
 1 MB • 4298 DM.



PT-ST (Plantron) • 80286/6, 8 MHz •
 Operativni sistem DOS 3.2 • Inter-
 fejs: paralelni • Set znakova IBM/nemački
 — DIN/ASCII • Magistrala 16
 bita • RAM 640K • Flopi 1.2 MB •
 Grafika max. 720x348 • Proširenja:
 6 slobodnih slotova, RAM do 1 MB,
 80287 • 3398 DM.



PT-286 AT (Plantron) • 80286/6, 10
 MHz • Operativni sistem DOS 3.2 •
 Interfejs: paralelni, serijski • Set
 znakova IBM/nemački — DIN/ASCII
 • Magistrala 16 bita • RAM 640K •
 Flopi 1.2 MB • Hard disk 64 MB •
 Grafika max. 640x480 • Proširenja:
 koprocessor 80287, 5 slobodnih slo-
 tova, RAM do 1 MB • 4998 DM.



MBC 17 Plus (Sanyo) • 80286/6, 8
 MHz • Operativni sistem MS-DOS
 3.2 • Interfejs: serijski, paralelni •
 Set znakova ASCII • Magistrala 16
 bita • RAM 512K • Flopi 1x/2x 1.2
 MB • Hard disk 20,30,40,60 MB •
 Osnovna verzija sa jednim flopijem
 • Proširenja: drugi flopi, hard disk
 • 3900 DM.



MBC 891 (Sanyo) • 80286/6, 10 MHz •
 Operativni sistem MS-DOS 3.2 •
 Interfejs: serijski i paralelni • Set
 znakova ASCII • Magistrala 16 bita
 • RAM 512K do 16 MB • Flopi: 1x/2x
 1 MB • Hard disk 20,30,40,60 MB •
 Tekst 80x25 • Osnovna verzija
 512K, 1 flopi • Proširenja: drugi
 flopi, hard disk • 4151 DM.



MBC 16 Plus (Sanyo) • 8086/4,77,
 8 MHz • Operativni sistem MS-DOS
 3.2 • Interfejs: serijski, paralelni •
 Set znakova ASCII • Magistrala 8/16
 bita • RAM 640K • Flopi: 1x/2x
 360K • Hard disk 20, 30 MB •
 Osnovna verzija sa jednim flopijem
 • Proširenja: drugi flopi ili hard disk
 • 1462 DM.



MBC 775 (Sanyo) • 8086/4,77 MHz •
 Operativni sistem MS-DOS 2.11 •
 Interfejs: paralelni • Set znakova
 ASCII • Magistrala 16/8 bita • RAM
 256-640K • Flopi: 1x/2x 360K •
 Hard disk 20 MB • Tekst 80x25 •
 Grafika 640x200 • Osnovna verzija
 sa dva flopija • Proširenja: hard
 disk, BTX verzija sa memorijom do
 640K • 5963 DM.



PT-XT (Plantron) • 8086/4,77, 8
 MHz • Operativni sistem DOS 3.2 •
 Interfejs: 1 serijski, 2 paralelni •
 Set znakova IBM/nemački — DI-
 N/ASCII • Magistrala 8 bita • RAM
 256K • Flopi 2x 360K • Proširenja:
 8087 koprocesor, 5 slobodnih slo-
 tova, RAM do 640K • 2398 DM.



PT-LC (Plantron) • 8086/4,77, 8
 MHz • Operativni sistem DOS 3.2 •
 Interfejs: paralelni • Set znakova
 IBM/nemački — DIN/ASCII • Magi-
 strala 8 bita • RAM 256K • Flopi
 360K • Grafika max. 720x348 •
 Proširenja: koprocessor 80287, RAM
 do 640K, 6 slobodnih slotova • 1798
 DM.



PT-286 AT Baby (Plantron) •
 80286/6, 10 MHz • Operativni si-
 stem MS-DOS 3.2 • Interfejs: pa-
 ralelni, serijski, igra • Set znakova
 nemački • Magistrala 16/8 bita •
 RAM 640K • Flopi 1.2 MB • Hard
 disk 64 MB • Tekst 80x25 • 132x25
 • Grafika 640x480 ili 1056x522 •
 Proširenja: 80287 koprocesor •
 3448 DM.



PC-116-1 Turbo (Promp-Computer)
 • 8086/4,77, 8 MHz • Operativni
 sistem MS-DOS 3.2 • Interfejs:
 Centronics, RS 232 C • RAM 256K •
 Flopi 360K • Grafika hercules mo-
 nohrom • Proširenja: RAM do 1 MB,
 sve standardne kartice, 8087, hard
 disk 160 MB • 1278 DM.



PC-116-x/20 (Promp-Computer) •
 8086/4,77/8 MHz • Operativni si-
 stem MS-DOS 3.2 • Interfejs: Cen-
 tronics, RS 232 C • RAM 256K •
 Flopi 360K • Hard disk 20 MB •
 Grafika hercules monohrom • Pro-
 širenja: RAM do 1 MB, sve stan-
 dardne kartice, 8087, hard disk 60
 MB • 2175 DM.



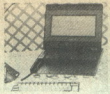
Schneider PC 1640 (Schneider) •
 8086/8 MHz • Operativni sistem
 MS-DOS 3.2 • Interfejs: serijski,
 paralelni • Set znakova IBM • Ma-
 gistrala 16 bita • RAM 640K • Flopi
 jedan • Hard disk 20 MB opciono •
 Tekst 80x25 • Grafika 720x348 •
 Osnovna verzija: jedan flopi, mono-
 hrom monitor, miš • Proširenja:
 drugi flopi ili hard disk, tri slobodna
 slot • 1698 DM.



Schneider PC 1512 (Schneider) •
 8086/8 MHz • Operativni sistem
 MS-DOS 3.2 • Interfejs: serijski,
 paralelni • Set znakova IBM • Ma-
 gistrala 16 bita • RAM 512K • Flopi
 jedan • Hard disk 20 MB opciono •
 Tekst 80x25 • Grafika 640x200 •
 Osnovna verzija: jedan flopi, mono-
 hrom monitor, miš • Proširenja:
 drugi flopi ili hard disk, kolor moni-
 tor • 1499 DM.



Goupil G5586 (SM7) • 8086/10 MHz
 • Operativni sistem MS-DOS 3.2
 • Interfejs: RS 232 C, Centronics, miš
 • Set znakova IBM, nacionalni set
 • Magistrala 16 bita • RAM 640K do 2 MB
 • Flopi 2x 360K • Hard disk 20/40 MB opciono • Grafika 640x200 ili 720x350 • Osnovna verzija sa DDGA karticom • Proširenja: 2 flopija, monitor -Z, Proširenje: Flopi 1 MB (5,25 inča), 720K (3,5 inča), strimer 60MB • 4400 DM.



Goupil Club (SM7) • 80C88/4,77 MHz
 • Operativni sistem MS-DOS 3.2
 • Interfejs: RS 232 C, Centronics
 • Set znakova IBM, nacionalni set
 • Magistrala 16 bita • RAM 768K • Flopi 2x 720K (3,5 inča)
 • Grafika 640x200 • Proširenja: koprocesor 8087, modem • 5575 DM.



Goupil G 4 (SM7) • 80186/8 MHz
 • Operativni sistem MS-DOS 3.1
 • Interfejs: 2x RS232C, Centronics
 • Set znakova IBM, nacionalni set
 • Magistrala 16 bita • RAM 512K • Flopi 2x 360K • Hard disk 20 MB opciono • Grafika 640x200, 720x350
 • Osnovna verzija sa 2 flopija
 • Proširenja: RAM do 640K, koprocesor, časovnik • 5118 DM.



Tandy 100 EX (Tandy) • 8086/4,77, 7,16 MHz
 • Operativni sistem MS-DOS 2.11, GW-Basic
 • RAM 256-640K • Flopi 360K • Grafika 640x200 • Osnovna verzija sa monohrom monitorom • Proširenja: flopi 360K (5,25 inča), ili 720K (3,5 inča) • 1485 DM.



Tandy 300 XL (Tandy) • 80286/4, 8 MHz
 • Operativni sistem MS-DOS 3.2
 • GW-Basic • Interfejs paralelni • RAM 512K do 4 MB • Flopi 360K • Grafika 640x200 • Osnovna verzija sa DDGA karticom • Proširenja: hard disk 20/40 MB • 3685 DM.



Tandy 200 (Tandy) • 80C85/2, 4 MHz
 • Interfejs paralelni • RAM 24K • Tekst 40x16 • Grafika 340x128
 • Proširenja: RAM do 72K, flopi 200 K • 1995 DM.



Tandon XPC (Tandon Computer) • 4,77 MHz
 • Operativni sistem MS-DOS 3.2
 • Interfejs Centronics paralelni • Magistrala 8 bita • RAM 256K • Flopi 2x 360K • Tekst 80x25
 • Grafika 720x48 • Proširenja: EGA grafički sistem • 2495 DM.



VID-AT 3094 (VID-Data) • 80286/6; 8, 10 MHz
 • Operativni sistem DOS 3.3, Unix/OS2 • Interfejs: Centronics, igre, 2x serijski • Set znakova DIN 2137 • Magistrala 16/32 bita • RAM 1 MB • Flopi 1,2 MB, 800K (3,5 inča) • Hard disk 40 MB • Tekst 80x25, 132x25 • Grafika 640x200, 720x48, 800x60 • Osnovna verzija sa monohrom grafičkom • Proširenja: hard disk 160 MB, optički disk, strimer • 4999 DM.



VID-X7 3084 (VID-Data) • 8086/4,77; 8, 10 MHz
 • Operativni sistem DOS 3.2 • Interfejs: Centronics, serijski, igre • Set znakova DIN 2137 • Magistrala 8/16 bita • RAM 640K • Flopi 360K • Hard disk 20 MB • Tekst 80x25, 132x25 • Grafika 640x200, 720x348 • Osnovna verzija sa kolor grafičkom • Proširenja: EGA, hard disk 160 MB, strimer • 3999 DM.



WY 1400-01 (Wyse) • 8086-1/9,54; 4,77 MHz
 • Operativni sistem MS-DOS 3.1 • Interfejs 2x V.24, Centronics • Set znakova IBM internacionalni • Magistrala 8 bita • RAM 256K • Flopi jedan • Hard disk opciono 20 MB • Tekst 80x25 • Grafika hercules, CGA • Osnovna verzija sa monitorom • Proširenja: drugi flopi, hard disk, koprocesor • 3363 DM.



WY 2108-01 (Wyse) • 80286/8MHz
 • Operativni sistem MS-DOS 3.2
 • Interfejs: V.24, Centronics • Set znakova IBM internacionalni • Magistrala 16 bita • RAM 512K • Flopi jedan • Hard disk opciono 20 MB • Osnovna verzija bez monitora • Proširenja: hard disk 20 MB, memorija do 15 MB, koprocesor • 4731 DM.



Toshiba 1000 Portable (Toshiba) • 80C86/4,77 MHz
 • Operativni sistem MS-DOS 2.11 • Interfejs RS 232 C, Centronics • RAM 512K • Flopi 720K (3,5 inča)
 • Grafika 640x200 ili 320x200 • Osnovna verzija sa jednim flopijem 720K • Proširenja: spojni flopi 5,25 inča • 2827 DM.



T 1000 (Toshiba) • 80C85/4,77 MHz
 • Operativni sistem MS-DOS 2.11 • Interfejs: RS 232 C, Centronics paralelni • Magistrala 8/16 bita • RAM 512K • Flopi 720K (3,5 inča) • Tekst 80x25 • Grafika 640x200 ili 320x200 • Proširenja: spojni flopi 5,25 inča • 2790 DM.



T 1100 Plus (Toshiba) • 80C86
 • Operativni sistem MS-DOS 3.2 • Interfejs: Centronics paralelni, RS 232 • Magistrala 16 bita • RAM 640K • Flopi 720K (3,5 inča) • Tekst 80x25 • Grafika 640x200 ili 320x200 • Proširenja: spojni flopi 5,25 inča, 5 slotova • 5680 DM.



Aphronic P 10/P 20 (TA Triumph-Adler) • 8086/4,77 MHz
 • Interfejs: Centronics, V.24 • Magistrala 16 bita • RAM 640K • Flopi 1x 360K, odnosno 2x 360K • Grafika 720x350 monohrom, 640x200 kolor • Osnovna verzija sa monohrom monitorom • Proširenja: 8087, strimer • 3630 DM.



Aphronic P50-1/P50-2 (TA Triumph-Adler) • 80186/8 MHz
 • Operativni sistem MS-DOS 2.11 ili 3.2 • Interfejs: Centronics, V.24 • Magistrala 16 bita • RAM 512K • Flopi 2x360K, odnosno 2x 720K • Grafika 640x400 monohrom i kolor • Osnovna verzija sa monohrom monitorom • Proširenja: hard disk, kolor monitor, RAM-disk, BTX • 4450 DM.



WY 2200-01 (Wyse) • 80286/10 MHz
 • Operativni sistem MS-DOS 3.2 • Interfejs: V.24, Centronics • Set znakova IBM internacionalni • Magistrala 16 bita • RAM 640K • Flopi jedan • Hard disk opciono 20/40/80 MB • Osnovna verzija bez monitora • Proširenja: memorija do 15 MB, drugi flopi, hard disk • 5928 DM.



Z-385 (Zenith) • 80286/8 MHz
 • Operativni sistem MS-DOS 3.2 • Interfejs: serijski, paralelni • Set znakova IBM • Magistrala 16 bita • RAM 640K • Flopi: 2 drajva • Hard disk 20 MB • Grafika 640x400 • Proširenja: flopi 3,5 inča • 4900 DM.



Z-145 Collage PC (Zenith) • 8088-2/14,77 8 MHz
 • Operativni sistem MS-DOS 3.2 • Interfejs: serijski, paralelni • Set znakova IBM • Magistrala 16 bita • RAM 512-640K • Flopi 360K • Hard disk 20 MB ili 80 MB opciono • grafika 640x200 u 16 boja, 720x350 monohrom • Osnovna verzija sa dva flopi drajva • Proširenja: jedan slot • 1995 DM.



Easy-PC (Zenith) • V40/7,1 MHz
 • Operativni sistem Easy-DOS • Interfejs Centronics • Set znakova IBM • Magistrala 16 bita • RAM 512K • Flopi 2x 720K (3,5 inča) • Hard disk 20 MB • Grafika 640x200 ili 640x400 • Proširenja: hard disk 21,4 MB, RAM do 640K • 1995 DM.

Ekspertni sistemi u kitu

U jednom od prethodnih nastavaka serije o veštačkoj inteligenciji obećali smo da ćemo ekspertnim sistemima posvetiti celu jednu priču. Pošto svoja obećanja uglavnom i ispunjavamo, ovoga puta pozabavićemo se ekspertnim sistemima i alatima za njihov razvoj.

Verovatno znate da su ekspertni sistemi kompjuterski programi koji poseduju znanja o nekoj relativno uskoj oblasti i da su sposobni da daju stručne savete, zamenjujući u određenoj struci nekog eksperta ili, u išiojjoj varijanti, pomažući dotičnom ekspertu konsultacijama.

Anatomija jednog eksperta

Ekspertni sistem se sastoji iz dve glavne komponente: **baze znanja** i **sistema zaključivanja**.

Baza znanja sadrži informacije o specifičnom subjektu. Znanje se pribavlja na razne načine: kroz knjige kao činjenice ili, na primer, heuristiku. Znanje se može predstavljati na razne načine, a moguće ga je i ažurirati u momentu kad se situacija promeni.

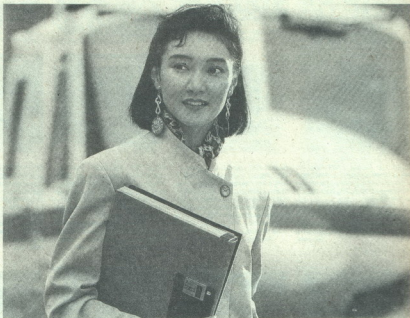
Sistem zaključivanja je softver koji je sposoban da informacije iz baze znanja sredi i odatle izvlači odgovore na razne upite. Naime, iz nekih od činjenica u bazi, kombinovanih sa informacijama dobijenim od korisnika, moguće je izvući razne specifične zaključke. Na taj način računar simulira ljudsko rezonovanje. Sistem koristi kontrolne strategije da bi odlučio u kom momentu je dobro primeniti neko od pravila na nove činjenice dobijene tokom rada — prilikom konsultovanja sa korisnikom.

Bilo bi greška reći da se ekspertni sistem sastoji samo iz baze znanja i sistema zaključivanja. Tačno je da su ovo njegove glavne komponente, ali ima tu još dosta toga. Spomenimo, reda radi, još dve bitne komponente ekspertnih sistema.

Interfejs prema korisniku omogućuje da korisnik u toku rada sistemu dostavi još neke važne informacije koje sistem iz baze nije uspeo da dobije, a pokaže se da u konkretnom zadatku imaju važnu ulogu. Takođe, koristeći ovaj interfejs, korisnik je u situaciji da za svaku od odluka ekspertnog sistema zatraži i dodatno objašnjenje o tome koji su rezoni rukovodili mašinu da donese takvu odluku.

Privremeni bafer za podatke čuva fakta i zaključke dobijene prilikom ekspertize koja je u toku. Ovaj bafer, koji često nazivaju i radnom memorijom, razlikuje se od baze znanja utoliko što sadrži informacije koje se odnose isključivo na tekući problem.

Svojevremeno, u prošlosti, projektovanje ekspertnog sistema je zahtevalo timski rad eksperta u oblasti za koju se sistem pravi (takozvani domenski ekspert) i osobe sposobne da ta znanja prenese mašini (takozvani inženjer znanja). U ta dobra stara vremena važno je pravilo da su se za svaki novi ekspertni sistem pisali i nova baza znanja i novi sistem zaključivanja. Ne treba ni spominjati da je iz tih razloga



trebalo ogromno vreme za pisanje ekspertnih sistema (mereno desetinama čovek — godina).

Onda su „jeretici“ sa Stanforda u okviru projekta za heurističko programiranje 1970. godine napravili MYCIN, ekspertni sistem za bakteriološku analizu krvi, koji je iz temelja uzdrmao ustaljeni pristup izgradnji ekspertnih sistema. Naime, MYCIN je projektovan tako da je njegov sistem zaključivanja (pod nazivom EMYCIN — valjda esencijalni deo MYCINA) bilo moguće izdvojiti i koristiti sa nekom drugom bazom znanja. Zaista, zahvaljujući EMYCINU, u rekordno kratkom vremenu napravljen je ekspertni sistem zvan PUFF za dijagnozu pulmonarnih disfunkcija ili, što bi se narodski reklo, plućnih smetnji. Da nije reč o nekom beskorisnom proizvodu dovoljno rečito govori podatak da PUFF još uvek rutinski koristi veliki medicinski centar Pacific Presbyterian Medical Center iz San Franciska. Dakle, EMYCIN je definitivno razbio zablude da se za novi ekspertni sistem uvek mora pisati i nov sistem zaključivanja.

Ekspert u delovima

Da bi se izgradilo ekspertni sistem od temelja, potrebno je dosta znanja iz računarstva i veštačke inteligencije i, naravno, iskustva u programiranju.

Koristeći programski jezik poput lisp ili prologa, moguće je projektovati sistem zaključivanja i izabrati način predstavljanja znanja. Jednom, kada struktura postane kompletna, može se u sistem uneti znanje i kreirati prototip ekspertnog sistema.

Ako bismo želeli da posao obavimo brže, morali bismo da startujemo od višeg nivoa. Neke od mogućih unapređenja su takozvana **programska okruženja** (programming environment). Takva okruženja omogućuju fleksibilnost jezika veštačke inteligencije, a da istovremeno raspoložu i raznim gotovim rutinama korisnim pri projektovanju ekspertnih sistema.

Alternativa predloženi pristupima je korišćenje **razvojnih sistema** za projektovanje ekspertnih sistema. Kod ovog tipa programa struktura sistema je unapred određena. To znači da je u paketu kupljen sistem zaključivanja sa praznom bazom znanja. Projektantu konkretnog ekspertnog sistema ostavljeno je da bazu znanja popuni informacijama njegovog domena, a da pri tome u sistemu zaključivanja ništa ne menja. To mnogima zvuči kao vrlo primamljiva mogućnost, jer umesto da se mušsa predstavljajam znanja i sistemom zaključivanja, sada su u poziciji da se posvete isključivo svom domenu i da brzo dobiju prototip željenog ekspertnog sistema.

Neki od projekatata ekspertnih sistema smatraju ovakvu strukturu previše spatzava-

judom. Sasvim je tačno da ovakvi razvojni sistemi (ekspertni sistemi u kitu) ne dozvoljavaju detaljnu kontrolu svakog procesa. Složeni sistemi koji su suočeni sa komplikovanim problemima često zahtevaju takvu potpunu kontrolu. S druge strane, prednosti ubacivanja znanja u vešte projektovane sistem zaključivanja su toliko da u mnogim situacijama odnose prevagu nad projektovanjem ekspertnog sistema od temelja. Ne treba posebno ni naglašavati da je cena sastavljanja ekspertnog sistema sa gotovim sistemom zaključivanja daleko manja od drugih pristupa.

Dobijanje prototipa u kratkom periodu vremena je vrlo korisno i da bi se odlučilo da li je konkretan problem pogodan za analizu primenom ekspertnog sistema (od nekih ekspertnih sistema se, kao od nepodesnih problema, odustajalo i u momentu kad je u njih već uloženo par miliona dolara). Takođe, brz i jeftin prototip je koristan pri predstavljanju ekspertnog sistema menadžerima koji vole da budu sigurni u ispravnost koncepta.

Ekspertni sistemi u kitu se po svojoj prirodi slažu sa današnjim trendom na tržištu projektovanja softverskih alati uopšte. To znači da će neki domenski ekspert mnogo radije razvijati ekspertni sistem u svojoj oblasti ako mu pri tom ne treba neka velika pomoć inženjera znanja. Dakle, alatka koju više ljudi ima šanse da razume, više će se i koristiti.

Tek dolaskom razvojnih sistema za ekspertne sisteme pružena je šansa i menadžerima da malo bolje shvate kako ekspertni sistemi zaista funkcionišu. Kad ovi strateški važni ljudi raščiste sa neodumicima i budu sigurni da ulazu kapital u stvar koja će raditi, situacija u veštačkoj inteligenciji će verovatno biti mnogo vrednija nego što je to danas.

Ekspert na lisp

Razvojni sistemi veštačke inteligencije se kategorišu po prirodi ugrađenog sistema zaključivanja i po načinu predstavljanja znanja u bazi. U komercijalnoj upotrebi mogu se sreći razvojni sistemi zasnovani na indukciji, na pravilima i frejm bazirani sistemi.

Kod indukcijских sistema znanje se stiče iz primera ranijih odlučivanja. U ovakvim sistemima koriste se razni algoritmi za kreiranje stabla odlučivanja. Ovo stablo se kasnije koristi za dobijanje odluka ili rezultata.

Sistemi sa pravilima koriste IF... THEN format za predstavljanje znanja. Razvojni sistemi ovog tipa međusobno se razlikuju po broju pravila koja podržavaju, kao i po tome da li pravila mogu da se grupišu hijerarhijski u cilju formiranja složenijih struktura.

Frejm bazirani sistemi, kao što i sam naziv govori, reprezentuju znanje u frejmovima. Kod ove tehnike objekti se opisuju zajedno sa svim informacijama koje imaju neke veze sa njim.

Od samih početaka ozbiljni ekspertni sistemi su se razvijali na lispu. Iz tih razloga i danas je činjenica da su veći i kvalitetniji razvojni sistemi ekspertnih sistema instalirani na lisp mašinama — specijalizovanim računarima namenjenim brzom izvršavanju lisp programa. Ekspertni sistemi nastali uz pomoć ovakvih alati imaju „manu“ što je



za njihovo korišćenje potrebno posedovati ništa manje nego neku lisp mašinu. Ovakav hardver Amerikanci nerado prodaju i zapadno-evropskim zemljama, a o nama i da ne govorimo. Dakle, za sada, alatke ovog tipa ostaju nam prilično nedostupne. Pored ove naše lokalne zamerke, sistemima o kojima upravo pričamo ponešto zameraju čak i Amerikanci. Naime, sistemi koji su izvršavaju na lisp mašinama ne mogu da koriste konvencionalne baze podataka, kao što ne mogu da koriste ni ne-lisповski softver!

Od nedavno su razvojni sistemi za projektovanje ekspertnih sistema počeli da se pojavljuju i za manje specijalizovane mašine od lisповskih. Na primer, za radne stanice poput Sun, Apollo, Tektronix ili Micro VAX. Nabrojane mašine nisu apstrakcija čak ni na Balkanu, a po Americi mogu da ih kupe i bolje stojeći studenti. Svi ovi računari imaju zajedničku osobinu da mogu da rade pod Unix operativnim sistemom. Inače, pod ovim operativnim sistemom na tržištu se može naći već sasvim pristojna ponuda razvojnih sistema veštačke inteligencije.

Između ostalog, postoje verzije i tako čuvenih sistema kao što je to EX-TRAN ili Rule Master. Interesantno je uočiti da ni jedan od ova dva sistema nisu razvijali Amerikanci. Oba su proizvod škotske firme Intelligent Terminals Ltd. videli smo njihove VAX verzije i u nastavku teksta na primeru EX-TRANa ilustrućiemo mogućnosti jednog pristojnog razvojnog sistema za projektovanje ekspertnih sistema u okruženju različitom od lisp mašine.

Ekspert na fortranu

EX-TRAN je istovremeno i indukcijски sistem i sistem zasnovan na pravilima. Korisnik EX-TRANa, tj. domenski ekspert, unosi znanje u sistem u obliku primera, a sistem odatle generiše pravila. Ili, ako pravila već postoje, moguće ih je direktno uneti u sistem. Stara pravila se mogu brisati ili menjati, a nova dodavati kad god se za to ukaže potreba. Zahvaljujući ovoj korisnoj osobini, sistem je lako održavati ažurnim.

Pravila u sistemu mogu da se grupišu u probleme i potprobleme. Interesantno je da se ovakve intervencije vrše izvan sistema! U običnom editoru se kreira eksterna tekstualna datoteka problema (problem text file) koji ne samo da specificira atribute

željenih grupacija pravila, već kaže i koje sve ekstremne programe korisnik želi da koristi unutar svog ekspertnog sistema. U ovoj datoteci se određuje čak i format koji će videti krajnji korisnik.

Kada je ekspert jednom zadovoljan stanjem sistema, tj. kad mu se učini da su generisana pravila ispravna, EX-TRAN će automatski proizvesti rutine za svako od pravila, i to, verovali ili ne, kao potprograme najobličnijeg fortrana-77!!! Nakon toga se dobijeni potprogrami prevode, a potom linkuju sa eksternom tekstualnom datotekom problema. Takođe, potprogrami se mogu koristiti potpuno nezavisno od sistema i moguće ih je linkovati sa drugim fortranskim programima.

EX-TRAN kao kontrolnu strategiju koristi takozvano povezivanje napred (forward chaining). Ova karakteristika za krajnjeg korisnika nema neki veliki značaj, ali je često dosta značajna za ljude koji razvijaju ekspertni sistem. Povezivanje napred znači da sistem radi tako što pretražuje pravila tražeći koje od njih je primenjivo na informaciju dobijenu konsultacijom sa korisnikom. Kad se pronade, pravilo se primenjuje, i zaključci se smeštaju u privremeni hafter za podatke.

Ekspertni sistem izgrađen uz pomoć EX-TRAN sistema omogućuje konsultacionu opciju u vidu dijaloga između korisnika i ekspertnog sistema. Korisnik je uvek u poziciji da pita ZAŠTO sistem od njega traži pojednu dodatu informaciju. Takođe, u svakoj situaciji moguće je postaviti i pitanje KAKO je sistem došao do pojednog odgovora, odnosno koji su ga argumenti rukovodili pri donošenju nekog zaključka. U zavisanosti za koga se razvija ekspertni sistem projektant je u poziciji da dostta lake pravi verzije koje će funkcionalisati uz veliku pomoć (u vidu konsultacije) krajnjeg korisnika, ili verzije koje će raditi za manje stručnog korisnika uz minimum konsultovanja.

EX-TRAN omogućuje da konsultacije idu ili dobijanjem potrebnih dodatnih informacija direktno od korisnika, ili da se za to koriste neki ranije pripremljeni podaci. U prvom slučaju korisnik informacije na zahtev sistema unosi preko tastature, dok u drugom opciji sistem direktno uzima iz neke eksternе datoteke. Sistem se može podesiti i tako da čita nekoliko eksternih datoteka bez bilo kakvih intervencija korisnika.

Ono što EX-TRAN izdvaja od ostalih razvojnih sistema za ekspertne sisteme je ▶

možućnost generisanja fortranskog koda. Kao takav vrlo je atraktivan u raznim naučnim i inženjerskim primenama (gde je fortran još uvek najčešće korišćeni jezik). Projektant EX-TRANa je bio svestan činjenice da će sistem najviše koristiti inženjeri, pa je njima za ljubav sistem opremljen vrlo jakim numeričkim delom. S druge strane, zahvaljujući njegovoj složenosti, EX-TRAN nije baš jednostavno koristiti (uputstvo za korišćenje ima nekoliko toмова teksta) što ga za manje primene čini inferiornim u odnosu na neke jednostavnije razvojne sisteme. Početnicima se možda neće sviđati ni to da EX-TRAN sam sebi nije dovoljan. Pored njega, trebaće vam svakako i fortran kompajler (jer sistem generiše fortranske potprograme koje, potom, treba još i prevesti).

Jedno je sigurno — EX-TRAN se zaista dokazao u praksi. Između ostalog, koristile su ga i firme poput Rockwell/Rocketdyne division pri razvoju ekspertnog sistema za analizu glavnog pogona na spejs šati, zatim multinacionalna naftna kompanija British Petroleum za analizu separacije sirove nafte, ili, na primer, zapadno-nemačka firma SEGAL za dijagnostiku štampanih kola. Interesantno je da je na EX-TRANu razvijen i ekspertni sistem za pronalaženje bagova u kompjuterskim programima!

Od malog ka velikom

Vrlo je važno shvatiti da nije baš svaki razvojni sistem podsejan za svaku primenu. Shodno tome, ne treba se čuditi što na tržištu postoji više raznih ponuda, a projektanti ekspertnih sistema i dalje muku muče da pronađu adekvatnu alatku za njihovu primenu.

Ista alatka zaista može da funkcioniše sa raznim bazama znanja, ali sa nekim od njih uspešnije nego sa drugima. Verovatno je najbolji način da se utvrdi da li će neki razvojni sistem dobro funkcionisati u nekoj konkretnoj primeni da se kreira prototip i na njemu eksperimentiše.

Ako ste se domogli nekog razvojnog sistema za projektovanje ekspertnih sistema, dobar savet je da za početak ne budete previše ambiciozni. Počnite da razmišljate o njegovoj primeni na male ekspertne sisteme. Posle za takve male inteligentne pomoćnike ima svuda oko nas (na primer u kancelariji: za pomoć u telefoniranju ili slanju pošte; za naručivanje potrošnog materijala; za analizu i proračune eventualnih troškova poslovanja; za otkrivanje kvarova u kancelarijskoj opremi itd.). Ključno je izabrati problem za koji smo sigurni da ima relativno mali skup pravila ili preporuke, a da pri tom ulazni podaci nisu nekog senzorskog tipa. Praksa pokazuje da se ovakvi jeftini ekspertni sistemčići u svetu odlično prodaju, pa eto šanse da se zarade dobre pare bez piratovanja softvera!

Čak i ako ne prodate svoj mali ekspertni sistem, sigurno ćete prilikom njegovog razvoja mnogo toga naučiti. Iskustvo stečeno na malim primenama od velike je koristi pri upoznavanju stvarnih karakteristika razvojnog sistema. Osim toga, pozitivni rezultati malih ekspertnih sistema deluju ubedljivo finansiraju mnoge velike projekte. A gde su veliki finansijski i veliki projekti, tu su i velike zarade! Dakle, na posao!

Zoran Obradović

Objektno programiranje

Programeri već decenijama pokušavaju da odgovore na jednostavno pitanje: „Kako pisati dobre programe?“ Pokazalo se da su dobri programi modularni, lako shvatljivi, da se mogu primenljivati i u situacijama za koje nisu specijalno pisani, da se mogu srazmerno lako menjati i dopunjavati. Sedamdesetih godina vodile su se žustre debate oko tzv. „strukturnog programiranja“ kao metoda za pisanje programa, a epilog danas znamo: struktuirano programiranje je ušlo u arsenal svakog programera koji lole drži do sebe. Uoporedo, mada bez pompe, evoluirale su i druge programske metodologije. Takozvano „objektno orijentisano“ programiranje (skraćeno OOP) predstavlja najznačajniji pomak u tom pravcu, a OO jezici danas doživljavaju neslućenu popularnost.

Najveći broj računarskih jezika podržava model izračunavanja koji se može sažeti ovako: aktivne procedure deluju nad pasivnim podacima. Svi najpoznatiji programski jezici su ovoj grupi — paskal, fortran, algol, PL/I, kobol, bezjik, i tako dalje. U njima postoji oštra razlika između podataka i informacija koje se obrađuju i procedura (potprograma, funkcija) koje ih obrađuju. Procedure i podaci su — što se tih jezika tiče — različiti entiteti, pa se programiranje svodi na konstruisanje programskih modula i njihovno pozivanje u toku izvršavanja programa. Svaki jezik definiše na svoj način prenošenje podataka u procedure, kao i u kojim se sve programskim blokovima može doći do jedne varijable.

Procedure ili poruke

Objektno orijentisani jezici zasnivaju se na pretpostavci da jedan jedini entitet — objekat — treba da obuhvati i podatke i procedure. Podaci u objektu mogu biti promjenjeni samo preko procedura (često ih nazivaju metodima) koje pripadaju tom objektu. Izračunavanja se izvode slanjem poruka (zahteva) objektima u programu. Objekti odgovaraju na poruke tako što stvaraju nove objekte i vraćaju ih kao rezultate. Svaki objekat je poput nezavisnog procesora čije se ponašanje može dokuciti samo na osnovu odgovora koje odaje. Na primeru običnog sabiranja dva broja uvidećemo ogromnu razliku između ova dva pristupa.

Pretpostavimo da treba sabrati dva broja:

2 + 4

U konvencionalnim programskim jezicima gornje izračunavanje bi se objasnilo na sledeći način. Naredba sabiranja, +, primenjuje se na dva celobrojna argumenta 2 i 4, i vraća rezultat 6. U OO jeziku rezultat će takođe biti 6, ali način na koji se do njega dolazi je drugojačiji. Poruka „+“ poslata je celobrojnom objektu 2. Objekat koji prima poruku (ovde 2) samostalno odlučuje kako će se naredba izvršiti, dok u jezicima tipa paskala aritmetički operator + komanduje izvršenjem. Komunikacija sa objektima

uvek je dvosmerna: rezultat, koji je i sam objekat (u gornjem primeru celobrojni objekat 6), vraća se kada se poruka pošalje objektu.

Programiranje u OO jezicima se, u krajnjoj liniji, svodi na svega dve operacije:

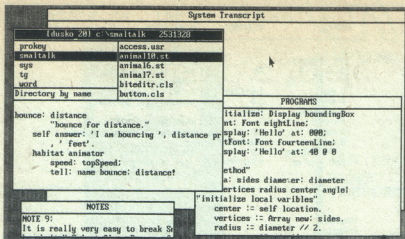
1. Prepoznavanje objekata i poruka u problemu o kojem želimo da pišemo program.

2. Određivanje odgovarajuće strukture podataka koja podatke interno predstavlja u okviru objekta, uz razvoj algoritama koji odgovaraju na poruku (tj. izvršavaju naredbu).

Drugim rečima, potrebno je prepoznati spoljašnji i unutrašnji vid objekta. Spoljašnji nas obaveštava šta dati objekat radi, a unutrašnji precizira kako se to postiže. Ne postoji način da se promeni stanje objekta osim slanjem poruka. Takva komunikacija je veoma prirodna i pruzeta je iz svakodnevnog života. O tome šta neka druga osoba misli može se suditi samo po porukama koje prima ili odaje, a svaka individua zna svoje unutrašnje stanje i kako reaguje na primljene poruke. Prema većini objekata u realnom životu ponašamo se na isti način — slanjem i primanjem poruka, tako da je objektno orijentisano programiranje prirodan nastavak uobičajenog ljudskog ponašanja. U stvari, u OO jezicima svaki program je simulacija nekog realnog procesa, i baš zato je programiranje na ovim jezicima najproduktivniji vid programiranja, takoreći — dečja igra. Ovak termin nije slučajno odabran, jer prvi OO jezik, „Smalltalk“, nastao je upravo iz želje da se napravi snažan računar za dečju.

Centar u Palo Altu

Pre skoro dvadeset godina čuvena kompanija XEROX stvorila je eksperimentalni naučno-razvojni računarski centar u gradiću Palo Alto, u Kaliforniji. Skraćeno, centar se zvao PARC (Palo Alto Research Center), i učinio je modernom računarstvu neprocenjive usluge. XEROX je skoro deceniju plaćao vrhunske računarske stručnjake da bi se isključivo bavili razvojem i eksper-



mentisijem — bez ikakve obaveze da nešto konkretno stvore i ponude tržištu. Pre desetak godina PARC je konačno zatvoren, ali je ostavio za sobom takve „svakodnevne“ stvari kao što su: miševi, ikone, bit-po-bit definisani ekrani (bit-mapped display), meniji, preklapajući prozori, projekat Dynabook i prvi objektno-orientisani programski jezik — „Smalltalk“.

Ako vas ovi pojmovi posećuju na računar „mekintosh“, u pravu ste — Stiv Džobs (Steve Jobs) je slučajno jednom prilikom video XEROX Star radnu stanicu i naprečak odlučio da sve to uključi kao standardan inventar, prvo u računar „Izma“, a potom i u njegovog ogojenog potomka po imenu „mekintosh“. Na ta dva računara su takođe bile realizovane verzije Smalltalk-a. Kasnije

su prozori, ikone, meniji i miš ovekovučeni u operativnim sistemima GEM i Windows, na računarima ST i IBM PC/XT/AT.)

Alan Kay, jedan od istraživača u PARC-u, stvorio je viziju računara Dynabook, koji je trebalo da ima sledeće osobine: cenu od svega nekoliko stotina dolara, veličinu školske sveske, ogromnu memoriju, ravan ekran osjetljiv na dodir, bit-po-bit programabilan grafički ekran, audio izlaze, i komunikacione mogućnosti. Smalltalk je trebalo da obezbedi softversku podršku.

Razvoj „Smalltalk-a“ produžio se na skoro deset godina, a završena verzija obnarodovana je 1980-te godine, pod imenom „Smalltalk-80“. Prva knjiga o „Smalltalk-u“ izašla je tek 1983. godine. Od tada, ovaj jezik, drugi OO jezik (npr. Actor, HyperTalk), i cela ideja objektno-orientisanog programiranja snažno dobijaju na zamahu. Na PC računarima veoma je popularna verzija „Smalltalk/V“, koja košta svega 99 dolara i pruža sve moguće pogodnosti svom korisniku (sadrži čak i kompletan Prolog). Actor je hibrid OO jezika i paskala, izvršava se na PC-u pod MS Windows-ima i košta oko 500 dolara. Za C postoje OO nadskupovi C++, PforC++, i slični. Najnoviji prodor OO jezika je HyperTalk, opet na novim verzijama računara „mekintosh“. Apple je čak odredio da HyperTalk bude glavni korisnički interfejs(i) i prodavače sa kao integralni deo svih novih „mekova“. S obzirom na lakoću korišćenja koja se time

Paskal VS Smalltalk

Evo primera na Paskalu (sa leve strane) i „Smalltalk“-u (sa desne). Pridodeljivanje izgleda isto:

```
a:=b+1          a:=b+c
```

jer je operator pridodeljivanja identičan u oba jezika, a i sintaksa varijabli je istovetna. Niz naredbi pridodeljivanja piše se ovako:

```
x:=0;          x:=0;
y:='odgovor'  y:='odgovor'
z:=w          z:=w
```

U paskalu se naredbe razdvajaju pomoću tačke i zapete, a u „Smalltalk“-u pomoću tačke. U oba jezika poslednja naredba u nizu ne mora da se razdvaja od sledeće naredbe. Varijabli y u oba jezika prima vrednost stringa, tj. niza slovnih znakova.

Pozivu funkcije u paskalu odgovara slanje poruke objektu:

```
b:=veličina(niz)      b:=niz.veličina
Funkcija veličina ima za svoj argument niz, a izračunata vrednost vraća se varijabli b. U „Smalltalk“-u poruka veličina šalje se sadržaju varijable niz. Evo primera funkcije sa dva argumenta:
```

```
x:=max(x1,x2);      x:=x1 max: x2.
y:=sum(p,q).        y:=p+q
```

U paskalu, argumenti funkcije moraju biti u obliku zagradama. U „Smalltalk“-u, argumenti se nalaze ispred i iza poruke. Sličan je princip i za pozive funkcije (poruke) sa tri ili više argumentata:

```
b:=između(x1,x2)      b:=x između: x1 and: x2
```

Ako poruka u „Smalltalk“-u ima tri ili više argumenta, ime poruke se razdvaja u nekoliko delova i deo poruke prethodi svakom argumentu (ne računajući prvi). U gornjem primeru, kompletna poruka zove se između:and: a argumenti su joj varijable x1, x2, i nizovi u „Smalltalk“-u savim su prepoznatljivi:

```
x:=a[];            x:=a at: 1.
a[]:=y;            a at: 1+1 put: y.
a[]+1:=a[]         a at: 1+1 put: (a at: 1)
```

Paskal koristi uglaste zagrade za oznaku elementa niza, a „Smalltalk“ koristi poruke at: i atput: (Primitivo da je + poruka, te da se u „Smalltalk“-u ne može pisati „slepijeno“ 1+1, kao u bejziku, Paskalu i skoro svim ostalim jezicima.) „Smalltalk“ dozvoljava da se samo skalarna veličina nalazi sa leve strane poruke :=, da se za sve ostale objekte mora koristiti atput: kao razdvojena poruka.

Naredbe grananja takođe su slične:

```
If a<b then
  a:=a+1
If atEnd(stream) then
  reset(stream)
else
  c:=next(stream)

a<b          IFTrue: [a:=a+1]
stream atEnd stream atEnd
IFFalse: [c:=stream next]
```

Ulovne naredbe pišu se pomoću uglastih zagrada. U prvom primeru, vrednost varijable a povećava se ako je ispunjen uslov a<b. U drugom primeru, učitava se neka datoteka u varijablu stream, pa ako je datoteka već pročitana do kraja, postavlja se na početak, a ako ne — učitava se sledeći znak, c.

Interaktivne naredbe su slične u oba jezika, na primer:

```
while 1<10 do begin [1<10]
  sum:=sum+a[];      while True: [
  i:=i+1              sum:=sum+(a at: i).
end;                 i:=i+1]
```

```
for i:=1 to 10 do   1 to: 10 do: [ : ]
  a[]:=0             a at: i put: 0]
for-petelj iz Paskala odgovara to do u „Smalltalk“-u. Varijabla označena su i: je interni indeks petlje, od 1 do 10 u drugom primeru.
```

Vraćanje rezultata izračunavanja obavlja se znakom stepenovanja,

```
ime Funkcije:=      rezultat;          rezultat
end;
```

U paskalu, rezultat funkcije pridodeljuje se imenu funkcije, koja ostatak programa služi kao varijabla iz koje se može pročitati vrednost izračunavanja. U „Smalltalk“-u, može postojati nekoliko vrednosti koje se u metodi izračunavaju, a ispred one koja se vraća ostalim objektima stavlja se :=. Kada „Smalltalk“ nalazi na taj znak, izvršavanje metoda se zaustavlja, a vrednost izraza vraća se kao vrednost metoda. (Podrazumeva se da se iz metoda vraća neki objekat.)

Dinamička alokacija promenljivih u memoriji zastupljena je u oba jezika:

```
new(p)             p:=Array new: 5
dispose(p)
```

U ovom primeru, i paskal i „Smalltalk“ u varijablu p smeštaju pokazivač na novocreovani objekat. Paskal zahteva da se objekti eksplicitno unište tokom izvršenja programa, naredbom dispose. U „Smalltalk“-u, uklanjanje dinamički alociranih objekata je automatsko i neprestano, tako da nema posebne naredbe za delokaciju memorije. Programi u „Smalltalk“-u zbog toga su manje podložni iznenađenim greškama u toku izvršavanja. Ugred, u „Smalltalk“-u sva imena varijabli samo su pokazivači na objekte negde u memoriji, tako da je svaki objekat (tj. strukturu podataka) moguće povećavati i smanjivati tokom izvršavanja programa!

Ne bi trebalo pomisliti da postoji takva paralela između paskala i „Smalltalk“-a. Većina programskih jezika pravi se oko minimalnog skupa tzv. rezervisanih reči. C je rekorder sa svega 28 rezervisanih reči; modula-2 ima 43, a kobil oko 300. „Smalltalk“ je sigurno najobiljniji: „Smalltalk/V“ za PC računare ima 200 (dve hiljade) naredbi! Drugim rečima, „Smalltalk“ je nadskup ne samo za paskal nego i sve druge programske jezike.

nudi, očekuje se da interesovanje za „mekove“ naglo poraste.

„Smalltalk“ je nastajao i kao jezik i kao operativni sistem, tako da uvek sadrži kompletnu radnu okolinu: interpreter, editor teksta, prozore, naredbe za rad sa datotekama (zamenjuje MS DOS na PC-u), komunikaciju sa mišem, dibager, grafiku visoke rezolucije, prevodilac, linker, menije, fontove, izvršavanje mašinskih programa, uopšte — sve što je programeru potrebno za svakodnevni rad. Na ovom mestu ograničimo se na osnovne pojmove OO programiranja, u terminima „Smalltalk-a.“

Osnovni pojmovi

„Smalltalk“ se sastoji od klasa. Pod klasom podrazumevamo sve informacije koje su potrebne za konstruisanje i korišćenje jedne grupe objekata. Objekti date klase obično se nazivaju primericima: svaki primerak pripada jednoj klasi, a klasa može sadržati više primeraka.

Klasa takođe sadrži radni prostor za metode. U OO jezicima metod je isto što i procedura u klasičnim programskim jezicima. Metod se pobuduje takozvanim selektorom, tj. porukom koja se šalje primerku klase. Metod fizički pripada klasi zbog štednje prostora, jer svi primerici jedne klase inakno treba da imaju iste ili slične odgovore na poruke. U metodu mogu postojati privremene varijable (nizovi, slogovi, matrice, ...) koji sasvim odgovaraju lokalnim varijablama u paskalu jer im se vrednost gubi po završetku metoda.

Svaki primerak rezerviše memoriju u kojoj se čuva njegovo stanje; tj. memoriji može se prići jer se vodi kao varijabla primerka. Za razliku od varijabli u metodima, varijabla primerka ne gubi svoju vrednost između dva izvršavanja metoda.

Programske klase

Najveći deo moći „Smalltalk“-a kao programskog jezika potiče iz grupisanja klasa u hijerarhiji. Svaka klasa ima bar jednu superklas (koja je, dakle, sadrži) i jednu ili više pod-klasa (koje ona sadrži). Na vrhu hijerarhije nalazi se klasa **Object**. Što je neka klasa bliža klasi **Object**, to je i opštija. Klase u programiranju mogu biti nova ideja, ali u drugim naukama, npr. u biologiji, sigurno nisu. Sva živa bića mogu se klasifikovati u skladu sa nekim svojim svojstvima. Slično, svi objekti u „Smalltalk“-u mogu se — a i moraju — klasifikovati, tj. pre početka stvaranja novog objekta programer mora da se odluči kojoj klasi (postojećoj ili nekoj) sasvim novoj, koju tek treba kreirati) novi objekat treba da pripadne. Objekti se tako formiraju kao osobine starih objekata, plus ono što ih razlikuje (tzv. operacionalna definicija). Evo primera.

Klasa **Magnitude** najčešće se koristi. Njoj pripadaju objekti koji se porede, mere, uređuju, i/ili prebrojavaju. Tipični predstavnici ove klase su slova, brojevi, datumi, vremena. Hijerarhija za sve **sub klase klase Magnitude** izgleda ovako:

Magnitude
Association
Character
Date
Number
Float

izraz	rezultat	komentar
4 / 3	1	celobrojni količnik
10 negated	-10	
2 + 3 * 4	20	svega udešno, striktno
3 - (2 + 2)	-1	zagrade
2 sqrt	1.414236	koren
4 sqrt	2.0	rezultat uvek iz Float
0 positive	true	
10 sign	1	znak broja
4 log: 2	1.4841316	logaritam u bazi argumenta
5 exp	2.4841316	
2.7182819 1n	1.0	prirodni logaritam
0.25 arcTan	0.24497866	ugao u radjanima

Fraction	LargeInteger	NegativeInteger
Integer	LargePositiveInteger	SmallInteger

Time

Magnitude nije ništa drugo do jedna apstraktna klasa koja definiše zajednički protokol za svoje sub-klase. U njoj se nalaze operacije poput =, <=, <, >=, >, =, kao i operacije za minimum i maksimum dva objekta iz ove klase. Sub-klase **nasleduju** sve varijable primerka, varijable klase i metode iz super-klase. Svaka sub-klasa može dodati svoje, specijalizovane primerke, varijable i metode, a takođe može neke od nasledenih elemenata preinaičiti. Sve to znači da se u velikoj doći izvršavanja programiranih delova programa može koristiti bez ikakvih izmena.

Klasa **Number** ne mora više da se bavi time koje su joj osnovne operacije porednje dozvoljene — sve ih je nasledila iz klase **Magnitude**. Međutim, klasa **Number** mora te poruke da redefiniše za slučaj brojeva, kao što i klasa **Date** mora poredjenja da redefiniše za datume (jer se oni drukčije poredje nego brojevi). Klasa **Number** razlikuje se od **Magnitude**, jer za svoje sub-klase obezbeđuje standardne aritmetičke operacije +, -, *, /, uz uobičajene numeričke metode: exp, cos, arcSin, tan, ln, sqrt, itd. Na primer:

Sub-klase **Float**, **Fraction** i **Integer** različito se odazivaju na operacije koje im je klasa **Number** postavila kao zajedničke. Na primer, već spomenuta poruka sabiranja, +, daje na znanje nekom objektu da treba da se izvrši sabiranje, ali **onako kako je za tu klasu interno definisano**. Brojevi — objekti iz klase **Float** sabiranjem daju takođe objekat klase **Float**; objekti klase **Fraction** (razlomci) sabiranjem daju razlomak; konačno, sabiranjem celih brojeva (**Integer**) dobija se opet objekat iz iste klase. To znači da je poruka sabiranja sva tri puta bila sasvim različito interpretirana, jer se sabiranje svake od ove tri vrste brojeva izvršava na sasvim različiti način. Ova osobina naziva se **polimorfizmom**, i omogućena je takozvanim „kasnim povezivanjem“, tj. „Smalltalk“ je interpreter i, slično lispu, može da u toku izvršavanja programa prepozna tip podataka sa kojim se radi.

Jezik bez mane

Objektno orijentisani jezici su u velikoj prednosti nad klasičnim, proceduralnim jezicima. Sve se događa posredstvom poruka, tako da su podaci i apstraktni i skriveni u isto vreme. Time programi dobijaju na pouzdanosti. Kasno povezivanje omogućava dodavanje novih klasa (objekata) a da se postojeće naredbe ne moraju menjati. Nasledjivanje, povezano sa polimorfizmom, omogućava da se veliki delovi već napisanih programa iznova koriste, pa programer piše manji broj naredbi, i time povećava svoju produktivnost. Konačno, naredbe koje se odnose na jednu funkciju nalaze se fizički samo na jednom mestu, a to bitno olakšava održavanje programa.

Kao koncept, „Smalltalk“ i drugi OO jezici su bez mane. U praksi, sve do pojave 16-bitnih računara OO jezici bili su retki. Čak i na PC/XT računarima „Smalltalk“ radi dobro ali ne idealno (AT je već sasvim dovoljan), dok se Actor guši i (bar u okviru 640 Kbajta) ne predstavlja ozbiljnu alternativu. Troškovi ostvarivanja kasnog povezivanja su, mereno računarskim resursima, skupi. Stanje poruke, po nekim merenjima, zahteva 1.75 više vremena nego prosto pozivanje potprograma. Na sreću, pokazalo se da programi pisani u OO jezicima u ukupnoj brzini izvršavanja ne zaostaju za proceduralnim jezicima, jer programer mora sam da u proceduralnom jeziku doda naredbe ispred i iz poziva procedure da bi postigao puni ekvivalent upućivanja poruke u nekom OO jeziku. Konačno, OO jezici sadrže veliki broj naredbi, pa početnik mora mnogo da uči pre nego počne da uživa u blagodatima objektno orijentisanog programiranja.

Sve u svemu, OO pristup programiranju predstavlja značajan korak unapred, i obuhvata sve dosadašnje ideale struktuiranog programiranja. „Smalltalk“ pripada grupi funkcionalnih jezika, te se može izvršavati paralelno ako računar ima više procesora. Takođe je jedan od tri glavna jezika veštačke inteligencije, pored prologa i lispa. Prema jednoj skraćenoj anketi časopisa *Dr. Dob's Journal for Professional Programmers* (najpopularniji programerski časopis u USA), „Smalltalk“ koristi čak 11% programera. Najpopularniju implementaciju, „Smalltalk/V“ opisao ćemo zato prvom sledećom prilikom.

Duško Savić

Novo izdanju Građevinske knjige

Priručnik za TIM 011 BASIC



40. GODINA



PRIRUČNIK ZA TIM-011 BASIC



Ovih dana se beogradskim školama vrše prve isporuke računara TIM-011 Instituta „Mihajlo Pupin“.

Iz dosadašnjeg iskustva znamo da su računari domaćih proizvođača oskudjevali u programskoj podršci, a naročito u pratećoj literaturi.

Nepotrebno je naglašavati da je za buduće korisnike računara TIM-11, dake i nastavnike, preko potrebno postojanje odgovarajućih udžbenika i literature iz programskih jezika za nastavu računarstva i informatike. U blizju budućnosti se očekuju programi i literatura za primenu računara i u nastavi iz drugih oblasti.

Do sada je na ovom planu kod nas urađeno veoma malo. Institut „Mihajlo Pupin“ se kao proizvođač školskog računara potrudio da u saradnji sa Građevinskom knjigom pokrene izradi i izdavanje niza neophodnih priručnika za operativni sistem i programске jezike računara TIM-011.

Iz štampe je već izašao priručnik za TIM-011 BEJZIK, a u štampi je priručnik za TIM-011 DOS. Namena ovih i budućih priručnika je da dokumentuju mogućnosti programske opreme koju proizvođač isporučuje uz računare.

Priručnik „TIM-011 BASIC“ urađen je po uzoru na svetske standarde. Pripremljen je pomoću PC-TEX programa i laserskog štampača.

Na 118 stranica formata A4 opisane su mogućnosti bežik interpretatora proširenog uvodnih instrukcijama. Priručnik je podeljen u pet poglavlja: Gfadi, Opšte karakteristike, Sastav bežika, Bežik komande i instrukcije i Bežik funkcije. U podacima su opisani rad sa datotekama, izvedene funkcije, poruke o greškama a tu je i obavezna ASCII tabela.

Utehu za one koji su u stranoj literaturi navikli na indeks predstavlja detaljan sadržaj.

U uvodu su navedene konvencije koje su upotrebljene pri pisanju priručnika.

U poglavlju „Opšte karakteristike“ objašnjeni su pojmovi „komandna linija“ i „programski red“ i navedeni kontrolni znaci TIM-tastature koje bežik prihvata.

Konstante, promenljive i njihovi tipovi, aritmetički, logički, relacijski operatori opisani su u poglavlju „Sastav BASIC-a“.

Komande i instrukcije bežika navedene su u abecednim redosledom i opisane na sledeći način (redom): sintaksa, namena, objašnjenje, elementarni primer.

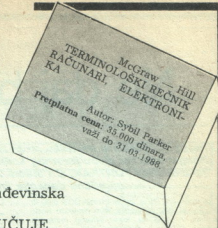
Ovaj način opisa, na žalost, nije pogodan za učenje bežika i podrazumeva da korisnik priručnika poseduje izvesno znanje iz programiranja i elementarno poznavanje bežika.

Bežik funkcije su opisane na način sličan opisu instrukcija.

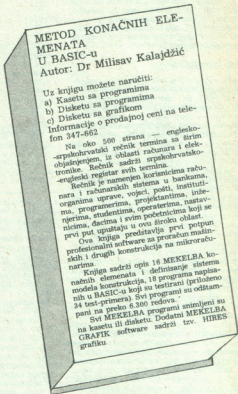
Prvi dodatak priručnika sadrži opisu rada sa datotekama, uz neophodne primere i čini se da bi bez njega priručnik bio prilično manjkav. U dodatku „Izvedene funkcije“ dati su primeri za izrađivanje funkcija koje nisu ugrađene u bežik interpretator. U porukama o greškama opisani su mogući uzroci grešaka i navedeni kodovi grešaka. Na kraju priručnika je ASCII tabela.

Naš zaključak: priručnik „TIM-011 BASIC“ urađen je veoma korektno. Njegovim korisnicima nesumnjivo će pomoći da lakše savladaju programski jezik bežik.

Knjiga ima 120 stranica A4 formata i može se nabaviti od izdavača po ceni od 7.000. dinara.



IRO „Građevinska knjiga“
PREPORUČUJE



METOD KONAČNIH ELEMENTA U BASIC-u
Autor: Dr Milisav Kalajdžić

- a) Kasetu sa programima
 - b) Disketu sa programima
 - c) Disketu sa grafikom
- Informacije o prodajnoj ceni na telefon 347-662

Na oko 500 strana — englesko-srpskohrvatski rečnik termina sa širim objašnjenjem, iz oblasti računara i elektronike. Rečnik sadrži srpskohrvatsko-engleski registar svih termina. Rečnik je namenjen korisnicima računara i računarskih sistema u bankama, organima uprave, vojsci, pošti, institutima, programerima, operativcima, nastavnima, studentima, operativcima koji se nećima, đacima i svim široku oblast. prvi put upućaju u ova široka područja. Ova knjiga predstavlja prvi potpun profesionalni software za proračun mašinskih i drugih konstrukcija na mikror računaru.

Knjiga sadrži opšte 16 MEKELBA konačnih elemenata i definisanje sistema mašinskih konstrukcija, 18 programa napisanih u BASIC-u koji su testirani (priloženo 34 test-primera). Svi programi su odštampani na preko 6.300 redova. Svi MEKELBA programi snimljeni su na kasetu ili disketu. Dodatni MEKELBA GRAFIK software sadrži tzv. HIRIS grafiku.

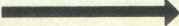
IRO „Građevinska knjiga“
11000 Beograd
Trg Marksa i Engelsa 8/II
tel. 347-662
PORUČBENICA — RAČUNARI 36

Poručuju knjige:
1. TERMINOLOŠKI REČNIK — RAČUNARI, ELEKTRONIKA, kom, u iznosu
2. METOD KONAČNIH ELEMENTA U BASIC-u (samo knjiga), kom, u iznosu
Uz knjigu možete poručiti:
a) Standardni paket MEKELBA na kaseti, kom, u iznosu
b) Standardni paket MEKELBA na disketi, kom, u iznosu
c) Dodatni paket MEKELBA GRAFIK na disketi, kom, u iznosu

Ukupni iznos od dinara uplatiti po prijemu uplatnica IRO „GRAĐEVINSKA KNJIGA“, ili na širo račun 69801-603-15416, sa navedenim nazivom knjiga.

U slučaju spora nadležan je odgovarajući sud u Beogradu.

Kupac:
Adresa:



Ne daj bubi da te gricka

Pravom programeru se teško može dati u ruke moćnija alatka od Microsoftovog dibagera CODEVIEW. Dibager radi na nivou izvornog koda, podržavajući istovremeno ni manje ni više nego tri prevodilaca (Microsoftov FORTRAN, C i Macro Assembler), i odlikuje ga veoma zreba koncepcija — u njega su ugrađene decenije Iskustva najboljih programera u borbi sa bagovima.

Za upotrebu CV-a trebate imati DOS 2.00 ili veći. Što se tiče memorije, sam program treba oko 384 K. Kad se tome pribroji memorija potrebna DOS-u i programu koji se testira, ispadne da se sa 512 Kb teško, a sa 640 Kb lakše diže. Hard disk nije potreban, iako je, svakako, dobrodošlo. Računar na kojem radite ne mora biti IBM-ov proizvod, čak ni stopotoni kompatibilac, ali mogućnosti CV-a ovise i o tome na kojem se računaru radi.

Lepota dibagiranja

Najveća snaga CV-a leži u njegovoj mogućnosti da dibagera na nivou izvornog koda. Što to znači? To znači da nam dibager omogućuje promatranje i kontroliranje izvršavanja programa u onom obliku u kom smo ga i napisali — u obliku skupa naredbi višeg programskog jezika i struktura podataka, a ne hrpe mašinskih instrukcija i adresa koje je generisao prevodilac. Zbog čega je to značajno? Svako od nas se sjeda svojih prvih koraka u programiranju. Većina je to činila u jeziku. Osnovna prednost jezika je u spoju lagane sintakse sa velikim mogućnostima za lako nalazjenje i ispravljanje grešaka koje pruža interpreter. Teško ispravljanje grešaka kod rada sa prevodilcima čini da je većini ljudi prelazak na neki drugi jezik, osim jezika, bolan i mukotpan. Osim toga, čak se i stariji okorjeli profesionalci desu da našu na situacije kad začale što ne postoji interpreter za onaj jezik u kojem rade, kako bi im traženje grešaka bilo olakšano. Simbolički dibageri mire dve nepomirljive koncepcije — udobnost interpretera i brzinu prevodilaca.

Sledeća stvar koja oduševljava korisnika CV-a je izuzetan izbor mogućnosti za kontroliranje i analiziranje rada programa koji dibagiramo. Na primer, tačke prekida mogu biti uvjetne, pri čemu uvjet za prekid može biti određeni broj prolaza kroz tačku, promjena vrijednosti varijable ili izraza, te ako neka varijabla ili izraz poprime tačno određenu vrijednost. Ove mogućnosti će znati cijeniti svi oni koji su se makar jednom zapitali zašto program ne prolazi kroz neku petlju onoliko puta koliko oni žele, kao i svi koji žele da doznaju zašto računaru uporno pokušava da podijeli neki broj s nulom, odnosno izvede drugi korijen iz negativnog broja ili mijenja varijablu koju nikad ne bi smio promijeniti. Uvjeti koji se mogu postaviti su razni i sve kombinacije koje su moguće će biti opisane.

Posljednja, ali ne i najmanje važna, vrijednost CV-a za programera je u tome što istovremeno podržava tri prevodilaca koji su, htjeli mi ili ne, u svijetu prihvaćeni standard i bez kojih profesionalni programeri ne mogu da žive (ako mislite da pretjerujem, isprobajte nove verzije). Tom unifikacijom je olakšan rad programerima, jer je sada dovoljno da zapamte samo jedan set komandi, dakle ne moraju se mijenjati dibageri svaki put kada se mijenja jezik na kojem se radi.

Prije početka

Da bi se neki program mogao dibagirati sa CV-om na nivou izvornog koda, mora se ponovo prevesti (odnosno asemblirati), uz nove opcije, što nas ograničava na dibagiranje programa čiji nam je izvorni kod dostupan. Nameće se, 32 računari 36 • mart 1988.

File	Search	View	Run	Watch	Options	Calls	Trace:	Go!	STATS.exe
									AX = 0000
									BX = 0000
									CX = 0000
									DX = 0000
									SP = 0800
									BP = 0000
									SI = 0000
									DI = 0000
									DS = 5B04
									ES = 5B04
									SS = 6753
									CS = 6115
									IP = 001F
									no carry
									no auxcy
									odd
									no carry

Slika 2. Izgled ekrana CODEVIEW-a sa izvornim kodom u dijalogu prozoru i otvorenom prozoru za registre

dakle, zaključak: CV nije namijenjen hakerima, već Pravim Programerima. Ovim prvima savjetujem da (ukoliko žele da budu glavni u svojoj ulici) nabave Atronus PC PROBE ili Intelov IZICE razvojni sistem. Cijena? Prava sitnica, 1500 dolara za prvi, deset puta više za drugi od njih.

Ukoliko to nije slučaj, programe ćemo moći dibagirati samo na nivou mašinskog koda. Prevodenje mora biti izvedeno samo sa onim verzijama prevodilaca koje podržavaju CV (za FORTRAN i C su to verzije od 4.00 naviše), a linkovanje sa verzijama linkera od 3.50 naviše. Dakle, sve ima svoju cijenu.

Mijenjanje izvornog koda radi dibagiranja sa CV-om nije potrebno, osim u slučaju kad koristite INCLUDE komandu za razvoj dužih programa. Tada CV neće moći, za fajlove koji su inkliudirani, dibagirati na nivou izvornog koda. Riješenje se sastoji u tome da ne vršimo inkliudiranje, već da module koji su inkliudirani prevedemo odvojeno i sve skupa kasnije linkujemo.

Kada dođe do prevodenja, ako želimo da razoplazemo sa brojevima linija izvornog koda i ostalim simboličkim informacijama, moramo koristiti specijalne opcije prevodilaca (za FORTRAN C to je /ZI opcija koja se mora doslovno ovako unijeti). Ne koristite ovu opciju u onim modulima gdje nije potrebna — štedimo memoriji i prostor na disku.

Preporučujem da kod dibagiranja „isključite“ optimiziranje (/OD u FORTRAN-u i C-u) jer, u suprotnom, mašinske instrukcije u prevedenom programu ne moraju odgovarati naredbama iz izvornog koda, a ni sve varijable ne moraju biti dostupne (mogu biti dodijeljene registrima). Optimiziranje „isključite“ ponovo nakon uspješnog dibagiranja. Normalno, ovaj savjet ne vrijedi ukoliko posumnate da od baga dolazi zbog prevodilaca. U tom slučaju bi „isključivanje“ optimiziranja moglo otkloniti vjerovatni uzrok greške.

Ovakvo pripremljeni moduli se linkuju sa /CO opcijom linkera.

CV može dibagirati samo ispravno prevedene

programe, što znači da vam on neće pomoći da ispravite sintaksne i slične greške. Za tu svrhu su vam potrebni drugi tipovi programerskih oruđa (ili oruđi?)

Dibagiranje modula obrađenih sa Library Managerom nije moguće, jer on iz modula odstranjuje sve informacije potrebne CV-u. Isto tako se ne mogu dibagirati ni programi koji koriste overlaye.

Ukoliko imate namjeru koristiti CV sa Macro Assemblerom, obratite pažnju na sljedeće stvari: 1. Verzije asemblera do 4.00 ne uključuju brojeve linija u objektni modul, što je ograničenje kod dibagiranja modula asembliranih s njima.

2. Sve simbole koje imate namjeru koristiti kod dibagiranja deklarirate kao PUBLIC. 3. Nije moguće dibagiranje izvornog koda. 4. Klasa za kod segment mora biti CODE'. 5. U programima asembliranih sa verzijama asemblera od 1.00 do 4.00 ne možete koristiti varijable u izrazima, ali zato možete konstante. Zbog ovoga se kod dibagiranja takvih programa ne mogu koristiti komande koje koriste varijable. 6. Ako koristite mala slova za simbole, isključite CASE SENSE opciju.

Predstava može da počne

Sada, kada je sve spremno (program je ispravno prevaden i linkovan sa svim potrebnim opcijama, CV se nalazi u tekućem direktoriju ili u putu), izvorna i izvršna verzija programa se nalaze u tekućem direktoriju, možemo startati dibager.

CV se starta sljedećom komandom: CV [opcije] ime izvornog fajla [argumenti]; gdje je:

ime izvornog fajla — ime fajla koji sadrži izvršni kod programa koji želimo dibagirati, argumenti — parametri koji su potrebni programu koji dibagiramo (pod uvjetom da ih on uzima sa komandne linije), opcije — jedna ili više sljedećih opcija: /W — CV se starta u prozorskom modu.

/B — upotrijebljiva se ako CV koristimo na PC-u koji ima CGA karticu i monitor sa samo dvije boje; ovlm se povećava čitljivost.

/S — CV koristimo za dibagiranje programa koji rade sa grafikom, ili sa više video stranica, te želimo vidjeti izlazni ekran; ne postoje nikakva ograničenja, ali je metod spor.

/F — dibagiramo programe koji ne koriste grafiku ili više video stranica, a želimo vidjeti izlazni ekran; video adapter ne smije biti monohromatski; metod je brz i šteti memoriji, ali postoje ograničenja.

/D koristimo IBM kompatibilni računar koji ne podržava određene IBM-specifične interapt funkcije.

/I — koristimo računar koji nije IBM kompatibilan, a želimo omogućiti prekid programa sa CTRL-C i CTRL-BREAK, te omogućiti NMI i maskiranje 9259 interapt kontrolera.

/T — CV se starta u sekvencijalnom modu (bez prozora). Ova opcija se često koristi kod redirekcijiranja.

/M — miš je instaliran, ali ga ne želimo koristiti; služi za dibagiranje programa koji koriste miša koji nije Microsoftov; mora biti instaliran drajver za miša koji dolazi uz CV.

/43 — imamo EGA karticu i želimo koristiti mod sa 43 rade na monitoru.

/2 — imamo 2 monitora i želimo koristiti oba kod dibagiranja; ovo je korisno kod izoliranja aktivnosti dibagera vezane uz ekran u odnosu na izlazni ekran programa; izlazni ekran programa će se pojavljivati na default monitoru, a dibagera na drugom; za vrijeme korišćenja ove opcije dibager ne može koristiti miša, dok dibagirani program može.

/Ckomande — CV nakon starta automatski izvršava seriju komandi navedenih u opciji; svaka komanda je odvojena od sljedeće sa tačka-zarezom; ako neka od komandi ima argumente koji traže blankove između njih, ciljeu komandi treba staviti u navodnike (" "); to isto važi i za svaku komandu koja koristi znakeve < i > (da ne bi došlo do neželjenog redirekcijiranja). Ako želimo da dibager izvede komande koje se nalaze u nekom fajlu, ova opcija ima izgled "/C>ime-fajla".

Ovim je startan CV i možemo početi sa dibagiranjem.

Organizacija ekrana

Ekran je (ukoliko se odluče raditi u prozorskom modu) po horizontalni podijeljen na dva prozora. U jednom dijelu komande dibagera (Microsoft ga zove dijalog prozor), a u drugom se nalazi program koji dibagirate. Linija koja je prikazana inverzno u ovom prozoru (plava na ekranu monitorima) je tekuća linija programa, odnosno ona će biti izvršena sljedeće. Linija na kojima se nalaze tačke prekida prikazane su svijetlije na ekranu.

Većina ova dva prozora se može mijenjati po vertikali. CTRL-D pomiče graničnik prema dolje (probuđuje prozor za program na račun dijaloga prozora), a CTRL-U čini obrnuto.

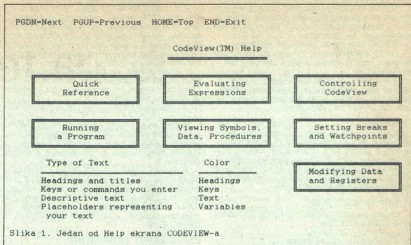
Kao odgovor na naše komande na ekranu se mogu otvoriti još dva prozora: prozor s registrima i prozor u kojem pratimo stanje varijabli ili izlaza u programu (Microsoft ga zove „watch window“).

Prozor s registrima se pojavljuje s desne strane ekrana. Dani su svi standardni registri mikroprocesora, a podržan je i matematički ko-procesor. Stanje zasticava je opisano punim riječima umjesto skraćenicama (no carry zamjenjuje NC). Prozor za pratiranje stanja varijabli, kada je uključjen, nalazi se odmah ispod menija koji je na vrhu ekrana. Odabrana opcija menija je ispisana svijetlije na ekranu.

Izbor iz jelovnika

Zadavanje komandi CV-u se može vršiti na 4 načina:

1. odabiranjem opcija menija tastaturom,
2. mišom,
3. funkcijskim tipkama i
4. zadavanjem direktnih komandi u dijalog prozoru.



Meniji su FILE, SEARCH, VIEW, RUN, WATCH, OPTIONS, CALLS, TRACE1 i GO! (zadnja dva su, u stvari, komande), a svaki od njih ima svoje opcije (osim zadnja dva). Otvaramo ih pritiskom na ALT i prvo slovo imena menija. Ukoliko otvoreni meni nije onaj koji smo željeli, možemo ga zatvoriti sa ESC ili (dok je još otvoren) otvoriti novi kursorskim tipkama za lijevo i desno.

Nakon što je željeni meni otvoren, izbor vršimo kursorskim tipkama za gore ili dolje i potvrđujemo s ENTER. Odabrana opcija je, prije nego što se potvrdi, ispisana svijetlije na ekranu.

Za izbor opcije se također može koristiti i istovremeni pretek na ALT i prvo slovo imena opcije. U tom slučaju ne moramo potvrđivati izbor pritiskom na ENTER.

FILE

Meni FILE služi za rad sa fajlovima. Sadrži tri opcije: LOAD, SHELL i QUIT.

LOAD opcija služi za učitavanje fajlova sa izvornim kodom i postavljanje tačka prekida u njima dok još uvijek radimo, recimo, na glavnom modulu. Tačke prekida u modulu čiji je izvorni kod prebrisao učitavanjem novoga neće biti izgubljene.

SHELL omogućuje privremeni izlaz u DOS radi izdavanja DOS komandi.

SEARCH

SEARCH meni služi za pretraživanje fajlova u potrazi za izrazima. Iznaz je način specificiranja varijabilnih stringova (slično kao wild card karakteri u imenima DOS fajlova), i biće opisano kasnije. Opcije su FIND, NEXT, PREVIOUS i LABEL.

FIND opcija traži određeni izraz u fajlu.

NEXT traži sljedeće pojavljivanje izlaza u fajlu. Upotreba ove opcije ima smisla tek nakon što je jednom upotrijebljena FIND opcija.

PREVIOUS traži prethodno pojavljivanje izlaza u fajlu.

LABEL pretražuje izvršni fajl tražeći labelu na nivou mašinskog koda.

VIEW

Opcije VIEW menija su HELP, SOURCE, ASSEMBLY, OUTPUT i EVALUATE.

HELP poziva help ekran.

SOURCE prebacuje dibager u mod u kojem je program koji se dibagira na ekranu prikazan kao niz linija izvornog koda.

ASSEMBLY prebacuje dibager u mod u kojem je program koji se dibagira na ekranu prikazan kao niz instrukcije mašinskog koda.

OUTPUT će nam pokazati izgled izlaznog ekrana programa koji se dibagira.

EVALUATE izračunava vrijednost izraza koji želimo.

RUN

RUN meni sadrži opcije koje su potrebne za izvršavanje programa koji se dibagira. To su START, RESTART, EXECUTE i CLEAR BREAKPOINTS.

START starta program od početka (kao da smo upravo počeli sa radom). Sve prethodno postavljene tačke prekida ostaju važeće. RESTART priprema program za ponovno izvršavanje, ali ga ne starta. Sve prethodno postavljene tačke prekida ostaju važeće.

EXECUTE izvršava program instrukciju po instrukciju, ali usporeno. Izvršavanje se prekida pritiskom na bilo koji tipku.

CLEAR BREAKPOINTS briše sve prethodno postavljene tačke prekida.

WATCH

WATCH meni služi za praćenje rada programa. Opcije su mu ADD WATCH, WATCHPOINT, TRANCEPOINT i DELETE WATCH. Uz ovaj meni su usko vezani pojmovi naredba pratiranja, tačka pratiranja (watchpoint) i tačka praćenja (tracepoint), koji će biti opisani kasnije.

Opcija ADD WATCH dodaje novi izraz za pratiranje već postojećim izrazima.

WATCHPOINT dodaje novi watchpoint već postojećim.

TRANCEPOINT dodaje novi tracepoint već postojećim.

DELETE WATCH odstranjuje izraz čiju smo vrijednost pratirali.

OPTIONS

Postavljanje parametara koji diktiraju ponašanje CV-a vrši se kroz OPTIONS meni. Opcije su FLIP/SWAP, MIX SOURCE, SYMBOLS, BYTES CODED, REGISTERS, CASE SENSE, C i FORTRAN. Ako je neki od parametara „uključen“, kraj njegove se nalazi znak koji neodoljivo podsjeća na deseterastični čin.

FLIP/SWAP je parametar kojim se omogućuje, odnosno onemogućuje, čuvanje izgleda izlaznog ekrana programa. Ako vam nije potreban izgled ekrana koji dibagirate, preporučujem vam da isključite ovaj parametar. Dibagiranje će biti brže, a i ekran neće treperiti.

Rad sa ovim parametrom nam smisla ako CV nije startan sa opcijom /S ili /F.

MIX SOURCE nam, kada je postavljen, omogućuje da na ekranu dibagira istovremeno vidimo izvorni kod programa koji dibagiramo i instrukcije u mašinskom kodu koje su dobijene njegovim prevodjenjem.

Uz pomoć SYMBOLS parametra biramo da li će varijable i labela biti prikazane kao simboli ili kao adrese.

► BYTES CODED govori da li će instrukcije u mašinskom jeziku biti predstavljene samo u obliku memorija ili će u njih biti i bajtovi koji predstavljaju taj memorik.

REGISTERS otvara prozor sa registrima. CASE SENSE govori na koji će način biti tretirana razlika između velikih i malih slova u simbolima.

Parametri C i FORTRAN određuju da li će debugger kod izračunavanja izraza koristiti sintaksu za prvi ili drugi jezik. Odabiranje jednog od njih automatski isključuje drugog.

CALLS

CALLS meni, u stvari, i nije meni nego svojevrsni tekst sa imenima podprograma, i parametara s kojima su ti podprogrami pozvani, koji su doveli do nekog mjesta na kojem se trenutno nalazimo. Odabiranje jednog od članova ovog steka će nam pokazati iz kojeg je modula i na kojoj liniji taj potprogram pozvan.

Funkcijske tipke

Jedan dio komandi se može zadati debugeru preko funkcijских tipki.

F1 — Poziva Help ekran.

F2 — Otvara prozor sa registrima.

F3 — Radi isto što i opcije SOURCE i ASSEMBLY menu VIEW.

F4 — Radi isto što i OUTPUT opcija menija VIEW. U CV se vraćamo pritiskom na bilo koju tipku.

F5 — Starta program. Program se izvršava do slijedeće tačke prekida ili do njegovog kraja (ovisno o tome šta prvo naiđe).

F6 — Prebacuje kursor iz dijaloga prozora u prozor sa programom, i obrnuto. Pokretanje kursora po prozoru se vrši sa kursorskim tipkama. PGUP i PGDN pomiču prozor za jedan ekran prema gore ili dolje. Da bismo došli do kraja programa služi nam END. Na početak dolazimo sa HOME. Za vrijeme upotrebe ove dvije komande kursor mora biti u ekranu s programom.

F7 — Postavlja privremenu tačku prekida na liniju na kojoj je kursor i izvršava program do te linije, do iduće tačke prekida ili do kraja programa (ovisno šta prvo naiđe).

F8 — Izvršava slijedeću instrukciju. Pozive potprograma i interapti prati, cod pozive DOS funkcija ne prati.

F9 — Uključuje i isključuje tačku prekida na liniji na kojoj je kursor.

F10 — Izvršava slijedeću instrukciju ili cijelu proceduru (ukoliko je slijedeća instrukcija poziv iz potprogram ili interapt poziv). Pozive DOS funkcija ne prati.

Većina prozora za dijalog, odnosno za program, može se mijenjati po vertikalni. CTRL-D pomiče graničnik prema dolje (proširuje prozor za program na račun dijaloga prozora), a CTRL-U isti obrnuto.

Ukoliko nam se program „zaglavio“ u beskonačnoj petlji možemo ga prekinuti sa CTRL-BREAK, odnosno CTRL-C (ovo ne mora raditi u svim slučajevima). Ako koristimo IBM PC AT, ili kompatibilni računar, rad programa se uvijek može prekinuti pritiskom SYSTEM REQUEST.

CTRL-C također prekida izvođenje trenutno aktívne komande debugera (recimo: dugo daniiranje memorije koja nam je dosadilo) ili unošenje nove komande preko komandne linije u sekvencijalnom modu, dok CTRL-S privremeno zaustavlja sve ispisivanja na ekran. Ispisivanje se nastavlja pritiskom na bilo koju tipku.

Dijalog

Dijalogom Microsoft naziva zadržavanje komandi debugeru preko komandne linije, bez obzira da li se nalazimo u prozorskom ili sekvencijalnom modu.

Sekvencijalni mod je, najjednostavnije rečeno, mod CV-a u kojem nam nisu na raspolaganju prozori i meniji. U tom modu komandni interfejs i izgled ekrana je isti kao i kod SYMDEB-a ili DEBUG-a.

Ovaj mod se koristi kod rada sa IBM kompatibilnim računarima koji nisu dovoljno kompatibilni da bi se prozorski mod mogao koristiti na njima.

Čak i ako imate dovoljno kompatibilan (ili originalni) IBM računar, ponekad ćete zaželjeti da koristite sekvencijalni mod, jer tada CV upotrebljava standardne DOS pozive za ispisivanje na ekran pa, uz malu pomoć redirekcioniranja, možete imati rezultat cijelog rada sekvencijalno, ili odštampan na printeru.

Komande koje se zadaju preko komandne linije su daleko moćnije od onih zadanih na ostali način, jer su na raspolaganju svi parametri za pojedini komandu.

Komande se mogu unositi bez obzira u kojem je prozoru kursor.

Jedina stvar koja je na raspolaganju samo u prozorskom modu, a ne i u sekvencijalnom, jeste bafer koji služi za čuvanje zadnjih 4 K komandi i izlaznih rezultata koji su bili ispisani u dijalog prozoru.

Komande koje ću opisati će biti grupirane u logičke cjeline, ovisno o njihovoj svrsi. Ako vam se s vremena na vrijeme učini da je opis suhoparan i štur, imajte na umu da je to danak koji se mora platiti činjenici da je ovo članak sa prikazom debugera, a ne prevod njegovih 270 (1) strana uputa.

Izvršavanje programa

Trace komanda ima format:

T [brojac]

pri čemu 'brojac' kaže koliko puta zaredom treba automatski ponoviti trace komandu.

Ova komanda koristi hardverski trace (omogućuje ga trace flag procesora), zbog čega se može pratiti izvršavanje programa i kroz ROM. P komanda to ne omogućava, i ako se upotrijebi na pogrešnom mjestu, efekat će biti isti kao da je dana GO komanda (0).

Za razliku od trace komande, P izvršava program proceduru po proceduru, korak po korak, tj. ne prati pozive u potprograme nego ih samo izvršava. Format je:

P [brojac]

Ni jedna od ove dvije komande ne prati pozive DOS funkcija, nego ih samo izvršava.

GO komanda starta program od tekuće adrese. P komanda se zaustavlja kada dođe do svog kraja ili kad naiđe na prethodno postavljenu tačku prekida, watchpoint ili tracepoint. Ako program dođe do kraja, CV će nam javiti koji je „errorlevel“ kod vraćen (o ovom pogledajte u uputama za DOS). Format naredbe je:

G [adresa]

gdje 'adresa' predstavlja mjesto u programu (adresu ili simbol) gdje želimo postaviti tačku prekida.

EXECUTE komanda (E) je slična GO naredbi. Nema nikakve parametre. Program se izvršava brzinom od nekoliko instrukcija u sekundi, što izaziva dovoljno silan onem koji se dobije kad se promatra filmski „slow motion“ efekat. Program će se izvršavati sve dok ne dođe do kraja, tačke prekida, trace ili watchpoint-a. Izvršavanje počinje od tekuće adrese, a prekida se pritiskom na bilo koju tipku.

RESTART komanda ponovo starta program koji se izvršava (isto kao da smo ponovo startali CV). Varijable su reinicijalizirane, ali su sve tačke prekida, kao i tracepoint i watchpoint, ostale nepromijenjene. Broj prolaza kroz neku tačku prekida je postavljen na 1, a argumenti su zadržani.

Ovom komandom se ne može startati novi program. Jedini način na koji se to može učiniti je izlazak iz debugera i njeno ponovno startovanje sa imenom novog izvršnog fajla. Format je:

L [argument]

gdje je 'argument' niz argumenta koje, eventualno, program traži iz komandne linije.

Podaci i izrazi

CV nam omogućava da radimo sa raznim tipovima podataka (varijable, memorijske lokacije i registri) i izraza.

USE [jezik] odabire koju sintaksu (C ili FORTRAN) će CV koristiti u daljnjem računanju izraza. Ako je komanda dana sama (bez parametara), ispisuje se sintaksa koja se trenutno koristi. Prilikom njegovog startanja CV će automatski odabrati sintaksu na osnovu proizvođača u

imeni fajla koji sadrži izvorni kod (npr. FORTRAN sintaksu za fajlove sa proizvođačkom FOR 7. [razl.] formatu).

Izračunava vrijednost izraza i ispisuje je. Izraz je bilo koji izraz dozvoljen odabranom sintaksom, a format je jedan od slijedećih formata za ispisivanje brojeva:

a — decimalni sa predznakom,
i — decimalni sa predznakom,
u — decimalni bez predznaka,
o — oktalni bez predznaka,
x — heksadecimalni,
f — realni broj sa predznakom, 6 decimalnih mjesta,
F — realni broj sa predznakom, scientific notacija, 6 decimalnih mjesta,
g — broj zapisan prema E ili f formatu (ovisno o tome koji je od njih kompaktiniji),
c — znak,
s — string koji se završava NULL karakterom (ASCII string).

Prikazivanje simbola, ili grupe simbola, njihovih imena i adresa, te imena modula, se obavlja sa:
X? [modul] [rutina.] [simbol] [-] pri čemu zadnji parametar, ako se upotrijebi, označava sve simbole u nekom modulu ili rutini. Prikazivanje sadržaja memorijskih lokacija obavlja se nekom od dump komandi. Openditi formati tih komandi su slijedeći:
D[tip] [adresa : raspon]
Tip je jedan od slijedećih tipova podataka:
B — bait,
A — ASCII,
I — broj sa predznakom (2 bajta),
U — broj bez predznaka,
W — riječ,
D — dvostruka riječ,
S — realni broj jednostruke preciznosti,
L — realni broj dvostruke preciznosti,
T — realni broj dužine 10 bajtova.

Ako se tip izostavi, podrazumijeva se default tip (zadnji tip specificiran sa DUMP, ENTER, WATCH MEMORY ili TRACEPOINT komandom).

Registre procesora prikazujemo i mijenjamo sa R komandom. Ova komanda u isto vrijeme prikazuje i tekuću instrukciju programa.

Registri matematičkog koprocссора se prikazuju komandomom T (ne nije štamparska greška, ovo je zaista broj sedam).

Tačke prekida

Tačka prekida se može postaviti na bilo kojoj poziciji u programu (osim na linije sa komentarom). CV dopušta 20 tačaka prekida (redni brojevi od 0 do 19) istovremeno. Svaka nova kreirana tačka prekida dobija slijedeći slobodan redni broj.

Tačke prekida se mogu postavljati, brisati, isključivati, uključivati i listati. Osim toga, mogu se postavljati i uvjetne tačke prekida, što će biti objašnjeno kasnije.

Tačka prekida se postavlja naredbom:
BP [adresa [brojac] [,komanda]]

pri čemu je 'adresa' mjesto u programu (bilo adresa, bilo broj linije izvornog koda ili neki simbol) na kojem se želi postaviti tačka prekida, 'brojac' je broj prolaza kroz tačku prekida koji je potreban da se ostvari prije nego što se ova aktivira i prekine program, a „komanda“ je spisak komandi CV-u koje će se automatski izvršiti nakon aktiviranja tačke prekida. Komande su jedna od druge odvojene tačkom-zarezom.

Brisanje jedne ili više tačaka prekida izgleda ovako:
BC lista

Lista je spisak tačaka prekida (po njihovim rednim brojevima) sa razmakom među rednim brojevima. Ukoliko se umjesto liste stavši zvjedrica (*), izbrisi će se sve tačke prekida. Ako jednu ili više tačaka prekida ne želimo izbrisati, nego samo privremeno isključiti, koristimo komandu:
BD lista

Isključene tačke prekida se ponovo uključuju komandom:

BE lista

Listanje jedne ili više tačaka prekida se obavlja sa BL.

Lista sadržava redni broj svake tačke prekida, njen status (da li je uključeni ili ne), adresu u rutinu koja je sadrži, broj linija izvornog koda na kojoj se nalazi, broj prolaza kroz ruku i spisak komandi koje će se izvršiti kod njenog aktiviranja.

Promatranje rada programa

Komande koje spadaju u ovu grupu su, sigurno, najmnogiše od svih. Omogućavaju detaljno promatranje rada programa i postavljanje uvjetnih tačaka prekida.

Pri radu s ovim komandama treba razlikovati naredbe za promatranje, tačke promatranja (watchpoint) i tačke praćenja (trapoint). Naredbe za promatranje nam omogućuju praćenje vrijednosti varijabli ili izraza za vrijeme rada programa, dok su tačke promatranja ustvari uvjetne tačke prekida koje se aktiviraju ako vrijednost nekog (unaprijed zadanog) izraza postane TRUE ili različita od nule. Tačke praćenja su također uvjetne tačke prekida, ali se one aktiviraju samo ako dođe do promjene vrijednosti izraza.

Naredbe za promatranje izraza ne rade sa programima koji su asemlirani verzijama assemblera od 1.00 do 4.00! Umjesto toga, koristite naredbu za promatranje memorije.

Komanda za promatranje vrijednosti izraza je: **W [izraz,format]**

a za promatranje sadržaja memorije:

W[tip] adresa : raspon

pri čemu je:

izraz — izraz na osnovu kojeg želimo vršiti promatranje,
format — jedan od dozvoljenih formata za ispisivanje brojeva,

tip — jedan od dozvoljenih tipova podataka koji se nalaze na memorijskim lokacijama.

Tačke promatranje se postavljaju naredbom:

WP? [izraz,format]

a tačke praćenja sa:

TP? [izraz,format]

za praćenje izraza i

TP[tip] adresa : raspon

za praćenje memorijskih lokacija.

Naredbe i tačke promatranja, te tačke praćenja, se brišu komandom:

W broj

'Broj' je redni broj koji je tački dodijeljen kod njenog postavljanja.

Ako ne znate redni broj, možete izlistati sve tačke sa njihovim rednim brojevima i tekućim vrijednostima komandom **W**. Ovo je ujedno i jedini način da se u selektivnijem modu dozna tekuća vrijednost neke tačke ili izraza za promatranje ili praćenje. U prozorskom modu za to služi poseban prozor.

Analiziranje koda i podataka

Način prikazivanja debugiranog programa na ekranu se bira komandom:

S[1-18]

gdje svaka od pojedinih kombinacija znači sljedeće:

S+ — prebacuje debugger u mod u kojem je program koji se debugira na ekranu prikazan kao niz linija izvornog koda.

S- — prebacuje debugger u mod u kojem je program koji se debugira na ekranu prikazan kao niz instrukcija mašinskog jezika.

S# — prebacuje debugger u mod u kojem je program koji se debugira na ekranu prikazan kao istovremeni niz linija izvornog koda, uz njihov ekvivalent u mašinskom jeziku.

Disasembling dijela programa se obavlja sa:

U [adresa i raspon]

Za gledanje sadržaja nekog tekst fajla (bilo da je to fajl sa izvornim kodom ili neki drugi fajl) poslužiošće se komandom:

V [izraz]

ili **V [[metafajl]:brojlinije]**

Unutar samog fajla, čiji sadržaj gledamo, možemo se pomicati kursorskim tipkama.

Najobičnija tačka (.) od nam reči na kojoj se lokaciji ili liniji u programu trenutno nalazimo.

Za one koji se brzo zapletu u pokušaju da prate koje je sve potprograme program pozvao, da bi došao tamo gdje je sada, komanda **K** je dušu dala. Njenim zadavanjem se dobija, u obrnutom redoslijedu nego što su pozvani, lista imena potprograma koji su trenutno pozvani, s argumentima sa kojima su pozvani, kao i broj linije u programu sa koje su pozvani.

Modificiranje programa

Mogu se modificirati tri stvari: program (asemliranjem novih instrukcija), sadržaj memorijskih lokacija (najčešće sadrže podatke) i sadržaj registra.

Asemliranje se započinje komandom:

A [adresa]

a prekida uvođenjem jedne prazne linije (dovoljno je samo pritisnuti ENTER). Konvencije kod umješnosti memoriona su iste kao i za DEBUG i SYMDEB. Dozvoljeni su svi mnemonici za procesore 8088/86, 80188/86, koprocessore 8087 i 80287, kao i 80286 mnemonici koji ne služe za programski mod (CV ne poznaje mnemonike za protected-mode procesora 80286).

Modificiranje programa se ne može spremiti za kasniju upotrebu!

Modificiranje podataka na memorijskoj lokaciji se vrši komandom:

E[tip] adresa [lista]

gdje je:

tip — jedan od dozvoljenih tipova podataka koji se nalaze na memorijskim lokacijama.

adresa — adresa memorijske lokacije od koje počinje modificiranje podataka

lista — jedan ili više izraza s čijim vrijednostima modificiramo sadržaj memorijskih lokacija; rezultat izraza mora biti istolo tipa kao i tip lokacije koju modificiramo; ukoliko lista nije zadana CV će pitati za njen sadržaj; u tom slučaju može se preokretati neka tačka pritiskom na razmaknicu ili se vratiti na prethodnu sa backslash karakterom; modificiranje se prekida pritiskom na ENTER.

Prikazivanje sadržaja registra procesora i njihovo mijenjanje se obavlja sa:

R [imerregistar] [=izraz]

imerregistar — je ime jednog od registra procesora, izraz — rezultat ovog izraza će biti nova vrijednost registra.

Kontroliranje debugera

Ova grupa komandi kontrolira ponašanje debugera.

H daje help ekran. Jedan put kad smo ga dobili — kroz njega se možemo kretati odabiranjem prvog slova (ono je svijetlije ispisano na ekranu) u imenu područja u kojem želimo više doznati. Dodatne komande za rad sa help ekranom su:

PGUP — vraća nas u meni sa višim nivoom od onog u kojem se nalazimo,

PGDN — spušta nas u meni za jedan nivo niže od onog u kojem se nalazimo,

HOME — vraća nas u glavni meni,

END ili ESC — izlazak iz help ekrana.

Sadržaj help ekrana se nalazi u fajlu CV.HLP. Ukoliko se možemo odreći pomoći, poslužimo se prostorom koji je CV zauzima na disku. Sa O izlazimo iz debugera.

Naba postavlja bazu brojnog sustava u kojem će biti prikazivani rezultati izlaza i varijable. Dozvoljene baze su 8 (oktalni sistem), 10 (decimalni) i 16 (hexadecimalni). Meni lično nedostaje baza 2 (binarni sustav).

Ukoliko program koji debugiramo piše na ekran direktnim mijenjanjem sadržaja lokacija video memorije, izgled debugerovog ekrana će biti uništen. Da bismo ponovo vidjeli sadržaj debugerovog ekrana, potrebno ga je ispisati, što se čini komandom:

Ovo je takozvano „tudo a' illi „majmun“.

Da bismo vidjeli izgled izlaznog ekrana programa koji se debugira, moramo dati komandu:

Povratak u debuger se postize pritiskom na bilo koju tipku.

U DOS shell se izlazi komandom:

[komanda]

pri čemu je „komanda“ DOS komanda kojoj želimo izvršiti. Nakon njenog izvršenja dolazi do

povratka u debuger. Ukoliko ovaj parametar nije zadan, povratak u debuger nije automatski, pa se možemo ponasati kao da smo u DOS-u, ali zato za povratak u CV moramo dati naredbu EXIT.

Za više detalja tekući direktori za vrijeme rada pod DOS shellom, obavezno se vratite u originalni direktorij prije povratka u CV. U suptromnom, debuger neće moći naći sve fajlove koji su mu potrebni za rad.

Za izvršavanje shell komande potrebno je oko 200 K memorije, što iznosi dovoljno da se iz shella ne može izvršiti neki program čiji je apetit za memorijom dosta velik.

Pretraživanje

Pretraživanje se vrši na fajlovima sa izvornim kodom. Traže se stringovi koje Microsoft zove „regular expressions“ (ja ću ih zvati samo izrazi).

Izrazi su u ovom kontekstu Microsoftov metod specificiranja varijabilnih stringova (slično kao wild card znaci u imenima fajlova u DOS-u). Najjednostavniji izraz je, u stvari, string pa se pretraživanje može obavljati i bez da se razumije način specificiranja izraza.

Komanda za traženje izraza je sljedeća:

/[izraz]

Ukoliko izraz nije dat, debuger traži sljedeće pojavljivanje zadnjeg definiranog izraza.

Dok su za DOS specijalni znaci + 17, u ovom slučaju su to:

\ obuhvata kosa crta

* zvjedrica

[] uglaste zagrade

\$ znak dolara

↑ znak ekspanenciranja

— znak minusa

tačka

Značenje ovih specijalnih znakova je sljedeće:

Obrnuta kosa crta služi za prevaziženje značenja specijalnog znaka, tj. stavlja se ispred svakog specijalnog znaka kojeg želimo tretirati kao obično.

Tačka zamjenjuje bilo koji znak.

Znak ekspanenciranja označava početak linije, a znak dolara njen kraj.

Zvjedrica označava pojavljivanje prethodnog znaka proizvoljan broj puta.

Uglaste zagrade i znaci između njih su, u stvari, spisak karaktera koji se moraju nalaziti u traženom stringu na mjestu gdje su postavljene zagrade. Ukoliko unutar uglastih zagrada stavimo znak ekspanenciranja, onda to znači da se navedeni znaci ne smiju nalaziti na tom mjestu.

Ako navedemo samo dva znaka, a između njih stavimo znak minusa, onda smo time definirali raspon znakova koji se mogu nalaziti na tom mjestu.

Redirekcioniranje ulaza i izlaza

Redirekcioniranje ulaza za komande koje se zadaju debugeru se obavlja sa:

=lmeuredaja

pri čemu ime uređaja može biti i ime nekog fajla.

Redirekcioniranje izlaza za debuger se zadaje sa

[T]->[imeuredaja]

Ovim je sav izlaz debugera skrenut na uređaj. Opcija T nam omogućuje (ako je zadano) da izlaz bude ispisivan na ekranu bez obzira što je skrenut. Na taj način više ne moramo raditi „no slip“.

Drugi znak > u komandi služi ako je izlaz redirekcioniran u fajl koji ne želimo obristati (na primjer, jučer smo u taj fajl spremili rezultate jednog dijela posla koji danas želimo dovršiti). U slučaju da je komanda tako zadana, zapisivanje će se nastaviti u kraju fajla, a ko nije, datoteka će biti prebrisana novim izlazom.

Ukoliko želimo istovremeno redirekcionirati ulaz i izlaz na isti uređaj, nije potrebno da kucamo dvije komande već je dovoljno:

=lmeuredaja

Završit ću ovaj članak jednom rečenicom za koju se više ne sjećam gdje sam je čuo ili pročitao, ali sam siguran da nije imala nikakve veze sa debugiranjem. Ona glasi: „Ne dad bubti da te grickali“.

Zoran Cvjetičić, dipl. ing.

računari 36 • mart 1988. 35

Editori teksta

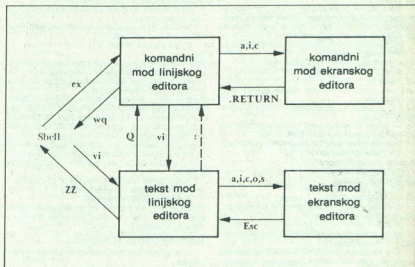
Treći nastavak naše male serije o juniks-u je uvod u upotrebu editora teksta vi i ex — videćemo da je na svakom moćnijem računaru editor teksta univerzalna alatka neophodna za svakodnevni rad, ali i da su juniksovi tekst editori često nepotrebno iskomplikovani i neprijatni za početnike. Dopravićemo, osim toga, priču o pravu pristupa datotekama, koju smo u prošlim „Računarima“ prekinuli na najzanimljivijem mestu.

Pre nego što pređemo na posao, pomećemo interesovanje koje je prvi nastavak ove serije izazvao među čitaocima „Računara“. Primili smo nekoliko pisama u kojima iskusniji korisnici juniksa, uz razne izraze podrške, kritikuju naš unekoliko kritički pogled na operativni sistem i, naročito, na mnemoniku komandi. Trebalo je da na samom početku kažemo da cilj ove serije nije upoznavanje čitaoca sa preciznom organizacijom operativnog sistema i sintaksom komandi juniksa a još manje regrutovanje novih zaljubljenika u ovaj operativni sistem — prostor nam omogućava samo da ponudimo minimum informacija koje će vam pomoći kada se budete našli ispred terminala neke juniks mašine. Ukoliko duže budete radili sa juniksom, na raspolaganju će vam biti brojne veoma obimne i kvalitetno pisane knjige, koje je možda trebalo nabrojati mnogo ranije. Kod nas se najčešće koriste Osborne McGraw-Hill-ovi priručnici čiji je prvi autor Rebeka Tomas (Rebecca Thomas); početnike posebno interesuje *A User Guide to the Unix System* (Rebecca Thomas & Jean Yates, Osborne McGraw-Hill, Berkeley, California, 1985). Sva pitanja i prilozima čitalaca su, jasno, dobrodošli i na njih ćemo se opširnije osvrnuti u poslednjem napisu iz ove serije.

Komanda *chmod*

Videli smo da svaka datoteka koja je upisana na neki od hard diskova juniks mašine ima svog vlasnika koji se u terminologiji juniksa zove *user* i obeležava sa 'U' — vlasnik datoteke je korisnik koji ju je kreirao. Jedino vlasnik (i upravnik računarskog centra) može da menja pravo pristupa za svoje datoteke. Korisnici koji zajedno sa vlasnikom rade na nekom projektu su grupa (obeležavaju se sa 'g'), dok su svi ostali korisnici centra jednostavno *other* ('o').

Juniks deli pristupe datoteci na čitanje (*read* ili 'r'), upis (*write* ili 'w') i izvršavanje (*execute* ili 'x'). Svaka datoteka ima desetak atributa koji govore ko je sve može čitati, ko izvršavati a ko menjati. Uz prikazivanje spiska datoteka upisanih u tekući katalog, komanda *ls* može da prikaže i prava pristupa za svaku datoteku — treba je samo dopuniti opcijom *-l*, tj. otkucati nešto poput *ls -l* *proba*, gde je 'proba' ime datoteke koju smo kreirali u „Računarima 35“. Zavisno od konfiguracije sistema, pojavice se tekst koji manje ili više liči na:



slika 1.

-rwx-r-x--- računari grupa 1 45 Jan 3 22:25 proba

'Računari' je ime vlasnika datoteke, 'grupa1' je ime grupe, 45 je veličina datoteke izražena u karakterima, 'Jan 3 22:25' označava datum kreiranja, 'odnosno zadnjeg modifikovanja datoteke, dok je 'proba' njeno ime. Tajanstveni znaci na početku ređu predstavljaju atribute prema shemi 'Tuuuggggooo', koja obično depirmira korisnike juniksa kojima je maternji jezik srpskohrvatski. T je tip datoteke, pri čemu crtica označava da je *proba* obična datoteka a ne katalog ('d'), 'uuu' označava prava koja ima vlasnik datoteke, 'ggg' se odnosi na grupu a 'ooo' na ostale korisnike. Vidimo da vlasnik ima sva prava, da grupa može da čita i izvršava datoteku a da joj ostali korisnici uopšte ne smeju pristupati.

Prava pristupa menjamo komandom *chmod* čija je sintaksa relativno komplikovana: *chmod klasa kod prava datoteka*. 'Klasa' je 'u', 'g', 'o' ili 'a' pri čemu je 'u' vlasnik, 'g' grupa, 'o' su ostali korisnici dok 'a' obuhvata sve ove tri kategorije; klase mogu da se kombinuju tako da, na primer, 'go' označava grupu i ostale korisnike. 'Kod' je plus, minus ili jednako; plus označava dodeljivanje novih prava, minus oduz-

imanje nekih postojećih, a jednako poništavanje postojećih i dodeljivanje novih prava. 'Prava', videli smo, mogu da budu 'r' (*read*), 'w' (*write*) ili 'x' (*execute*), a smisao imena datoteke je očigledan. Sada možemo da protumačimo komandu *chmod o+rwx proba* koju smo pomenuli u prethodnom „Računarima“: svim korisnicima centra ('o') se dodaje ('+') pravo da čitaju, menjaju i izvršavaju ('rwx') datoteku 'proba'. Ako želimo da oduzmemo pravo menjanja datoteke, otkučaćemo nešto poput *chmod o-w proba*. Ponekad je zgodno samom sebi oduzeti pravo izmene neke datoteke i tako sprečiti njeno nehotično brisanje ili menjanje — otkučaćemo *chmod u-rx proba*. Sa *ls -l* proba uvek može proveriti da li su prava zadata u skladu sa našim željama.

Vredi primetiti da nisu nezavisni svi parametri prava pristupa — ako se program u nekoj datoteci može čitati, on se može i izvršavati, što znači da je atribut 'r' praktično identičan sa 'rx'. Slično tome, ako je datoteka otvorena za svet, ona je samim tim otvorena i za grupu i vlasnika. Autor ovog teksta je jednom pokušao da ispita ponašanje datoteke na koju vlasnik nema nikakva prava, pa je otkučao nešto poput *chmod u-rwx proba*. Iznenadjenje nije bilo malo

kada se pokazalo da vlasnik i dalje ima pravo da ispisuje sadržaj datoteke; preciznije ispitivanje je pokazalo da je datoteka i dalje bila otvorena za „svet“ (trebalo je zapravo otkucati *chmod a-rwx proba*), pri čemu je „vlasnik“ sam sebi i „grupa“ i „svet“.

Pažljivo čitaoci primetiće jedan zanimljiv logički problem: datoteka može da bude otvorena za izvršavanje ('x'), a istovremeno zatvorena za čitanje. Ovakve datoteke su zgodne jer omeđuju korisniku da krade softver dopuštajući mu da ga izvršava. Izvršavanje je, istini za volju, neka vrsta čitanja ali je ovakva zaštita, iako naoko logički besmislena, praktično izvodljiva pri čemu korišćeni metodi prevazilaze ovo osnovnu školu juniksa. U napredne primene spada i dodatni atribut koji obezbeđuje *privremeno* dodeljivanje korisniku koji izvršava datoteku identifikacionog koda njenog vlasnika i, prema tome, dopušta nepreviđevanovim korisnicima da u kontrolisanim uslovima izvršavaju neke operacije rezervisane za upravnika centra.

Juniks tretira kataloge kao i obične datoteke, što znači da komandom *chmod* možemo da menjamo atribute bilo kog potkataloga koji smo kreirali. Ukoliko ostalim korisnicima centra zabranimo svaki pristup katalogu, zabranjujemo im i pristup datotekama koje su upisane u taj katalog, bez obzira na činjenicu da su same datoteke možda otvorene za čitanje. Da bi ostali korisnici mogli da izvršavaju naše datoteke, osnovni katalog i potkataloge u koje su datoteke upisane moramo da otvorimo bar za „izvršavanje“ ('x') — koristi se isti termin i isti atribut, premda „izvršavanje kataloga“ prilično smešno zvuči.

Kako se pišu programi

U ROM jednostavnijih kućnih računara ugrađen je bezik interpretator zajedno sa manje ili više kvalitetnim editorom koji obezbeđuje unošenje i ispravljanje programa i, eventualno, komandi. Ukoliko neki „spektrum“ ili „komodor“ opremite paskal kompajlerom, uz njega dobijate novi editor koji se koristi za unošenje paskal programa na sličan način treba da upoznate editor za unošenje asemblerkih programa, editor za C i tome slično. Veći kompjuterski sistemi, pa čak i moderniji personalni računari, programiraju se na mnogo različitih programskih jezika, što znači da bi bilo sasvim besmisleno očekivati da programer upozna desetak raznih editora i da ih u svakodnevnom radu racionalno koristi. Zato je sastavni deo svakog ozbiljnijeg operativnog sistema univerzalni editor teksta koji se koristi za unošenje i ispravljanje programa na svim postojećim jezicima, kao i za kreiranje i modifikaciju raznih datoteka kojima se računarski sistem konfigurise. Uz juniks se obično isporučuju editor *ed*, *ex* i *vi*. *ed* je jednostavan linijski editor, dok *vi* i *ex* zapravo predstavljaju dva segmenta istog editora koji je razvijen na Berkliju — *ex* je linijski a *vi* ekranski editor. Ostatak ovoga teksta posvećujemo osnovnim pojmovima neophodnim za korišćenje editora *ex*, odnosno *vi*.

Slika 1 prikazuje međusobne veze raznih modova editora. *Shell* je u terminologiji juniksa komandni mod, iz koga se *ex* datoteka odnosno *vi* datoteka prelazimo u

komandni mod linijskog odnosno ekranskog editora. Kucanjem neke od komandi *a*, *i*, *c* prelazimo u mod za unošenje i ispravljanje teksta, a iz njega se u komandni mod vraćamo unošenjem tačke (*ex*) odnosno pritiskom na taster *Esc* ili *interrupt* (*vi*). Iz linijskog u ekranski editor prelazimo komandom *vi*, dok je za obrnuti put zadužena komanda *Q* (bitno je da je slovo veliko). Napuštanjem linijskog editora, najzad, iniciramo sa *wq*, a napuštanjem ekranskog sa *ZZ*. Da bi stvar bila još malo moćnija i komplikovaniya, u komandnom modu ekranskog editora možemo da izvršimo bilo koju komandu linijskog editora, ako ispred nje stavimo dvoćatku — na ovaj način privremeno prelazimo u linijski editor ali se po izvršenju komande automatski vraćamo u ekranski.

Početak i kraj rada

Najjednostavniji način da startujemo ekranski editor je *vi datoteka* gde je *datoteka* ime datoteke čiji sadržaj treba ispraviti ili, ako datoteka ne postoji, kreirati. Kurzor će normalno biti postavljen na prvi znak datoteke, ali juniks obezbeđuje i promenu ove konvencije: sa *vi+datoteka* pozicioniramo se na kraj datoteke (zgodno ako program treba produžiti), dok sa *vi-n+datoteka* ispravke počinjemo od *n*-tog reda. Često se koristi i *vi+tekst datoteka*, što pozicionira kurzor na prvo pojavljivanje *tekst-a*. Samo se po sebi razume da po startovanju editora možete slobodno pomerati kurzor i pretraživati tekst, što znači da općiju minus tiče, na raspolaganju su vam obični *vi-r datoteka* (koristi se za „oporavljanje“ datoteka koje su editovane u trenutku kada je sistem pač) i *vi-wn datoteka*. Poslednji oblik pođešava veličinu prozora u kome se tekst obrađuje; prozor se, ukoliko drugačije ne odredite, postavlja na 8, 16 ili 24 linije u zavisnosti od brzine komunikacije terminala i centralne jedinice do 600, do 1200 odnosno preko 1200 bauda), što obezbeđuje komforan rad.

Ekranski editor napuštamo kucajući *ZZ*, a linijski komandom *wq*; nema, naravno, nikakve prepreke da ekranski editor napuštamo sa *wq* pošto smo videli da dvoćatka obezbeđuje izvršavanje komandi linijskog editora iz komandnog moda ekranskog. Ukoliko zaključimo da su ispravke koje smo uneli nepotrebne, editor napuštamo sa *q!* — izmene se ignorišu i datoteka ostaje nepromenjena.

Opšti oblik komandi editora *vi je* (brojač) *operator* (brojač) *operand*; zagrade označavaju neobavezan deo komande. *Operator* govori šta treba uraditi: *d*, na primer, označava brisanje teksta. *Operand* označava objekat na koji treba primeniti operaciju: *c* je, na primer, znak (karakter), a *w* reč. Oba brojača obezbeđuju višestruko ponavljanje iste operacije: *d2w* će, kao i *2dw*, obrisati dve reči iza kurzora dok bi *5d2w* obrisalo $5 \cdot 2 = 10$ reči.

Pokretanje kurzora

Pri radu sa editorom ekran je zapravo prozor na potencijalno daleko duži tekst, što znači da vidimo samo 23 (ili manje) sukcesivne linije programa koji obrađuje-

mo. Komandama *CTRL U* i *CTRL D* pomeramo prozor nagore odnosno nadole i to za polovinu deklarisanе dimenzije: posle *CTRL D*, na primer, donja polovina ekrana zamenjuje gornju dok se ostatak ekrana popunjava daljim sadržajem datoteke; kurzor se pozicionira blizu vrha prozora. Za brzo pregledanje teksta ponekad je zgodno pomerati kompletan prozor (osim možda par graničnih linija) a na njegovu polovinu — za pokretanje unapred zaduženo je *CTRL F* a za pomeranje unazad *CTRL B* pri čemu se kurzor pozicionira na sam početak prozora.

Za pokretanje kurzora u okviru prozora zaduženi su tasteri na kojima su nacrtane strelice; obzirom da je juniks nastao na mašinama koje nisu imale naročito bogate tastature, ostavljena je i mogućnost pomeranja kurzora na desno, levo, gore i dole komandama *l*, *h*, *k* i *j* respektivno; u skladu sa pomenutim općim oblikom komandi, 5k će pomeriti kurzor 5 redova naniže. U samom editorском modu kurzor možemo da pomeramo i pritiscima na *CTRL H* (levo), *CTRL P* (desno), *CTRL N* (dole) i *SPACE* (desno) — ove komande se menjaju od terminala do terminala i upisane su u datoteku (*etc/termcap*) na način koji će vam možda objasniti neki iskusniji korisnik.

Kretanje znak po znak je prilično sporo, pa su autori *vi-a* obezbedili komande *A* (bezik znak za stepenovanje) i *\$* koje pomeraju kurzor na početak, odnosno kraj linije, kao i komande za „preskakanje“ reči. Reč je, što se juniksa tiče, bilo koji niz slova, cifara i znakova za podvlačenje (—) koji je omeđen znacima koji nisu alfanumerički — blanko simboli, interpunkcija, tabulatori... Komandama *W*, *b* i *e* pomeramo kurzor na početak sledeće, početak tekuce odnosno kraj tekuce reči dok sa *W*, *B* i *E* nalazemo sistemu da pri prepoznavanju reči ne obrača pažnju na znake interpunkcije — reči su tada nizovi simbola „omeđeni“ blankovima ili tabulatorima. Komande '*i*' i *j*' su, najzad, zadužene za pomeranje kurzora na početak sledeće odnosno tekuce rečenice pri čemu je rečenica definisana kao niz znakova koji se završavaju tačkom, uzvičnikom ili upitnikom. Ostalim komandama za pomeranje kurzora ovde se nećemo baviti — nisu baš neophodne za praktičan rad!

Osim pomeranja kurzora, *vi* mora da obezbedi unošenje, ispravljanje i pretraživanje teksta i da pruži korisniku mnoge usluge koje se prirodno očekuju od specijalizovanog editora teksta. Komandama koje ovo obezbeđuju bavimo se za mesec dana, kada ćemo upoznati i neke dodatne naredbe samog juniksa. Peti, poslednji nastavak ove serije planiramo za „Računare 38“ — upoznaćemo naredbe rezervisane za upravnika sistema!

Dejan Ristanović

Sve po listi

Ako ste, čitajući naslov, umesto poslednje reči mahinalno izgovorili reč „spisak“, to još ne znači da imate pokvarenu maštu, već da znate da jedna od najkorisnijih struktura podataka — lista — nije ništa drugo do običan spisak (reči, simbola itd.). Liste ne služe samo za „veštački inteligentne“ programe, već i za rešavanje sasvim ovzemaljskih programerskih problema.

Mada su skupovi kao pojam nastali u okviru matematike, danas ih srećemo u gotovo svim naučnim i tehničkim oblastima. Programiranje se takođe zasniva na skupovima i skupovnim operacijama, ali pojam skup morate svedom tražiti u programerskoj literaturi. Za razliku od kompjuterskog naučnika, programer će umesto „program za dodavanje i uzimanje elemenata skupa“ reći nešto kao „program za insert-/delete na AVL-stablu“ i začuđiti se kako to da ga nije razumeli. Veoma mali broj autora (samo oni najznačajniji kao što su Knuth ili Aho, Hopcroft, Ullman) nastoji da objasni programeru da su liste, binarna, 2—3 i AVL stabla samo strukture podataka i da služe predstavljanju jednog istog pojma — skupa.

Prosečnom poznavocu programerskih jezika Virthove kuhinje (pascal i modula-2) pitanje rada sa skupovima može se učiniti prostim i nepotrebnim, jer poznaju tip podataka SET koji služi za definisanje skupa. Radi se, međutim, o „optičkoj varci“. Tip podataka SET ima vrlo ograničen broj elemenata (obično 32 do 64), tako da je za bilo kakve ozbiljne primene potpuno neupotrebljivo. Pokušajte da deklarirate SET OF integer ili SET OF character, pa će vam postati jasna veličina ovog ograničenja.

Pri radu sa skupovima treba uvek rešiti dva osnovna problema. Kako u računaru predstaviti skup i kakve operacije na skupu definisati? Mada na prvi pogled potpuno različiti, ova dva problema su čvrsto povezana. Najjednostavnije rečeno, ne može svaka struktura podataka predstavljati skup (i ne mogu se na svakoj reprezentaciji skupa (ranije izabranom strukturom podataka) realizovati efikasno sve skupovne operacije). Sa uređenim skupovima se radi lakše i efikasnije nego sa neuređenim, ali algoritmi koji rade sa neuređenim skupovima imaju šire područje primene. Sve u svemu, svaka konkretna realizacija skupa i operacija na njemu predstavlja rezultat kompromisa, uz uvažavanje zakonitosti „posla“ na koji skup treba primeniti.

Operacije sa skupovima

Skupovi se predstavljaju na toliko mnogo načina da bi stvaranje uvek novih operacija u zavisnosti od predstavljanja stvorilo pravi haos. Stoga se kao polazna tačka uzima skup osnovnih operacija (sa skupovima), pa se za konkretno predstavljanje razmatra koje se od njih mogu a koje ne mogu izvoditi. Razni autori daju razne definicije osnovnih operacija u zavisnosti od toga čime se konkretno bave. Za čitaoca „Računara“ odabrao sam onu koju su dali 38 računari 36 • mart 1988.

Aho, Hopcroft, Ullman u svojoj čuvenoj knjizi „The Design and Analysis of Computer Algorithms“ zbog toga što nije mnogo dugačka (svega 7 operacija), a sadrži dovoljno moćne operacije da se njima može komforno raditi. Operacije su sledeće:

- 1 — **MEMBER(a,S)** — određuje da li a pripada skupu S (vraca flag: 0 ili 1)
- 2 — **INSERT(a,S)** — dodaje skup S element a, ukoliko on već nije u njemu
- 3 — **DELETE(a,S)** — uklanja a iz S ako je on u njemu postojao
- 4 — **UNION (S1,S2,S3)** — nalazi uniju skupova S1 i S2 i smešta je u skup S3; pri tom se smatra da su S1 i S2 disjunktni skupovi.
- 5 — **FIND(a)** — daje „ime“ skupa koji sadrži a
- 6 — **SPLIT(a,S)** — razdvaja uređeni skup S na dva skupa S1 i S2 tako da su elementi S1 manji ili jednaki a a elementi S2 veći od a.
- 7 — **MIN(S)** — daje najmanji element u S

Pri proceni algoritma za rad sa skupovima ne ceni sa složenost jedne operacije već niza operacija. Pri tom niz može sadržati sve operacije 1—7, ali ih ne mora sadržati sve. Tako se razlikuju tzv. UNION-FIND nizovi ili INSERT-DELETE-MEMBER nizovi, u zavisnosti od toga koje od operacija sadrže. Složenost se traži kao složenost za niz dužine n (recimo, za dodavanje n elemenata skupu)

Skupovi su naročito potrebni pri radu sa grafovima, gde se obično javlja nekoliko odvojenih nizova operacija (najčešće su to UNION-FIND i INSERT-DELETE-MEMBER nizovi) od kojih svaki radi sa drugim predstavljanjem skupa, što smanjuje složenost algoritma jer važi pravilo: što je manje različitih operacija u nizu, to je veća efikasnost (o efikasnosti i grafovima možete čitati u RA 30/55).

Predstavljanje skupova

Pošto su određene osnovne operacije, sledeći korak je određivanje predstavljanja. Pri tome osnovni faktor je struktura niza osnovnih operacija koji treba izvršiti na skupu. Neke kombinacije operacija se lako „usklađuju“, dok je neke skoro nemoguće uskladiti. Izbor predstavljanja se vrši tako da se dobije najbolja srednja složenost, što će reći da nema nikakve koristi od toga što se jedna operacija izvršava sa složenošću $O(n)$ ako je drugoj potrebno $O(n^2)$. Ideja vodilja je, dakle, da složenosti operacija budu približno iste, da ako neka baš „hoće“ da bude efikasna a to ne smeta ostalima — utoliko bolje.

Osnovni zahtev koji predstavljanje skupa mora da ispuni je dinamičnost. Pošto se nikada u vreme pisanja programa ne može znati koliko će elemenata biti potrebno, veličina skupa mora biti promenljiva i neograničena, što će reći da nikakvi nizovi i matrice ne dolaze u obzir. Izuzetak su interpretirani jezici poput bejzika, kod kojih

su nizovi i matrice često dinamičke strukture, no o tome kasnije. U opštem slučaju (kod kompajliranih jezika) se, dakle, koriste pointeri i tip podataka RECORD odnosno STRUCT u zavisnosti od toga da li su u pitanju paskal/modula-2 ili C. Pošto se

programi za rad sa skupovima ne pišu na mašinskom jeziku, to u razmatranje predstavljanja moraju ući i karakteristike jezika. Ukoliko se zahteva da elementi skupa mogu biti bilo kakvi raznorodni podaci, onda paskal/modula-2 apriori otpadaju zbog nepostojanja takozvane (cast) konstrukcije svojstvene algolu i C-u. Naravno, uvek možete raditi u interaktivnom fortu ili lispu, ali tada morate računati na upoznavanje nove sintakse ovih jezika (ukoliko na njima već ne radite).

Najdirektniji način predstavljanja skupa je lista, tj. spisak. Kad skup napišete na papiru, on nije ništa drugo do spisak elemenata, a kao takvog ga i računari može lako prihvatiti. Programi za rad sa listama se lako i brzo pišu. Liste ne zahtevaju nikakvu specijalnu obradu pri dodavanju i uklanjanju podataka, efikasno koriste memoriju i mogu sadržavati bilo kakve podatke. Mana im je, međutim, to da je vremenska složenost niza od n instrukcija uvek $O(n^2)$ za bilo kakav raspored instrukcija (čak i kad se radi o nizu samo jedne instrukcije), što će reći da je za velike skupove rad sa listama vrlo spor (ukoliko nemate pri ruci list-mašinu).

Drugo rešenje je korišćenje hash-tabele sa očekivanom složenošću $O(n)$ (za INSERT-DELETE-MEMBER niz). Ovakvo rešenje je na prvi pogled idealno, no kada se uzme u obzir to da i ovaj metod u „nesrećnim“ slučajevima „pada“ na $O(n^2)$, te da je za svako dodavanje/uklanjanje elementa potrebno mnogo dodatne obrade, pokazuje se da ovo rešenje i nije baš za neku preporuku.

Sledeće rešenje je binarno stablo. Složenost ovog rešenja je $O(n \log n)$ u srednjem i $O(n^2)$ u najgorem slučaju, po čemu bi se moglo zaključiti da je to još gore od hash-tabele. Binarno stablo međutim, slično listi, ne zahteva gotovo nikakvu dodatnu

obradu i to ga čini privlačnim, naravno u slučaju da radite sa uređenim skupom, budući da se binarno stablo zasniva na relaciji uređenja skupa. Najveća mana binarnog stabla je to što lako degeneriše u listu, svaki put kada dobija uređene ulazne podatke, a to se često dešava. Zamislite asembler koji za tabelu simbola (koja nije ništa drugo do skup parova ime-vrednost) koristi binarno stablo. Posle dužeg rada, uspeli ste konačno da disasemblerom „razbije“ neki mašinski program, potom ste ga malo modifikovali i prepustili asembleru. Pošto su labele kreirane od strane disasemblera uređene u rastući niz, binarno stablo će postati lista, a asembliranje će proteći u nedogled.

Da bi se iskoristila dobra svojstva binarnog stabla a otklonila „opasna“, koristi se balansirano stablo kod koga se, na račun dodatne obrade dodavanja/uklanjanja stalno broj čvorova u levom i desnom podstablu održava približno jednakim. Time se eliminiše opasnost degeneracije, a zadržavaju se dobra svojstva binarnog stabla. Postoje dve vrste balansirano stabla: AVL-stablo, nazvano po svojim tvorcima (Adeľ'som-Vľ'skil i Landis) i 2—3 stablo koje je stvorio Hopcroft i kod koga svaki čvor može imati 2 ili 3 sina, čime se, u stvari, vrši balansiranje.

Skup kao lista

Lista je struktura podataka koja u velikom broju slučajeva potpuno zadovoljava kao reprezentacija skupa, naročito ako se prvi put srećete sa skupovima na računaru, pa vam je jednostavnost i razumljivost programa daleko važnija od par sekundi dobika u brzini. Liste se jednostavno definišu na algotimik jezicima (Pascal/Modula-2 i C) pomoću pointera, ali to je prilaz koji ne možemo baš svi da isprobamo na svojim mašinama. Za onoga ko tek ulazi u ovo područje idealno bi bilo raditi na jeziku, koji je interaktivan i omogućava vam da probate i grešite bez opasnosti po pad sistema. Za početak, dakle, bezik dobija prioritet kao jezik za rad sa listama.

Liste se u njemu mogu realizovati na dva osnovna načina. Prvi je ekvivalentan realizaciji stabla datoj u RA 30/57, dakle pomoću dva niza od kojih prvi sadrži podatke, a drugi „link“, tj. redni broj sledećeg elementa. Ovakav način realizacije ima, međutim, bitan nedostatak, a to je ograničenost dužine liste. Na sreću, mnogo savremeni bezik interpreteri omogućavaju redimenzioniranje niza, tako da se lista može bilo kad produžiti ako je to potrebno. Mogu se, međutim, kao linkovi držati stvarne adrese, a sa podacima raditi koristeći PEEK i POKE tehniku. Tada, međutim, bezik odlazi u vode paskala i postaje prilično nerazumljiv za nekog ko tek treba da stiče osećaj rada sa listama.

Drugi način realizovanja je puno jednostavniji. Radi se o tome da lista postaje običan niz znakova sa odgovarajućim grafičnim znacima (delimiterima). Neka početak i kraj liste označavaju otvorena i zatvorena vitičasta zagrada. Unutar ti zagrada su elementi odvojeni zarezima. Dakle, izrazom:

```
AS=" A,B,CD,75,7Q,F9;"
```

se u alfanumeričku promenljivu AS upisuje lista od 6 elemenata. Savremeni bezik interpreteri su opremljeni bogatim i moćnim setom naredbi za rad sa stringovima, što u mnogome olakšava manipulisanje

listama. Nevojia je, međutim, što se sintaksa tih naredbi razlikuje od jedne implementacije do druge. U cilju razumljivosti programa, odlučio sam da zaboravim na monstrum-operacije tipa MID\$ i da usvojim standardnu notaciju koju je bezik nasledio od fortrana 77 (radi se, dakako, o jednom zvaničnom standardu ovog jezika: ANSI BASIC-u). Notacija je vrlo jednostavna i laka za korišćenje. Izvršavanje naredbe:

```
PRINT AS{3,7}
odštampaće „B,CD
tj. substingr od trećeg do jednog znaka iz
AS. Ako bi se sad izvršilo:
```

```
AS{3,7}="" i PRINT AS
```

dobilo bi se

```
„A,75,7Q,F9;"
```

što će reći da je tako izbrisan substingr iz AS. Izvršavanje:

```
AS{3,0}="" „R" i PRINT AS
```

daje pak

```
„A,R,75,7Q,F9;"
```

tj. ubačen je novi substingr u AS.

Nekoliko jednostavnih procedura

Ubacivanje novih i brisanje starih substingrova je, dakle, krajnje jednostavno, a to je ujedno i sve što je potrebno za rad sa listama. Od standardnih operacija potrebne su funkcije LEN(AS) koja daje dužinu stringa i POS(T\$,P\$,P), koja daje poziciju od koje unutar T\$ počinje P\$ (kao substingr). Pri tome se ispituju karakteri u T\$ počev od pozicije P (ako P nije navedeno podrazumeva se 1).

```
POS("APERABPERA","PERA") daje 2
```

```
a POS("APERABPERA","PERA",3) daje 7
```

```
1000 SUB UNION(SS,US)
1010 IF US="" THEN 1130
1020 IF SS="" THEN SS=US GOTO 1130
1030 US=US{2,LEN(US)-1}&
1040 B=1
1050 E=POS(US,;)
1060 WHILE E
1070 S=LEN(SS)
1080 PS=US{B,E-1}
1090 IF NOT POS(SS,PS) THEN SS{S,0}=""&PS
1100 B=E+1
1110 E=POS(US,;B)
1120 END WHILE
1130 END SUB
```

slika 1

Na slici 1 vidite proceduru koja daje uniju dva skupa predstavljenih listama upisanim u SS i US. Listiji 1010 i 1020 ispituju slučajeva kad je jedan od skupova prazan. Listiji 1030 eliminiše vitičaste zagrade uz US i dodaje zarez, čime se olakšava pretrazivanje. Ukoliko želite da sačuvate US, morate ga preneti u proceduru „by value“ a ne „by reference“. Listiji 1040 i 1050 postavljaju početne vrednosti „pointera“ na početak (B) i kraj (E) elementa liste iz US koji se ispituje. U listiji 1060 počinje WHILE petlja u kojoj se za svaki element liste uz US pretrazuje lista iz SS i dodaje joj se ukoliko već u njoj ne postoji. Listija 1080 „vadi“ element iz US-liste, a listija 1090 ga dodaje SS listi ako je to potrebno. Listiji 1100 i 1110 postavljaju nove vrednosti B i E „pointera“, a listija 1120 zatvara WHILE petlju.

Time je realizovana osnovna operacija UNION koja radi za nedisjunktnе skupove. Pri tome je pretpostavljeno da je SS dovoljno dugačak da primi sve elemente unike. U nekim bezik interpreterima je to tačno, jer su sposobni da produžuju string koliko je

to potrebno. Kod drugih se, pak, mora vršiti njegovo redimenzioniranje, što donekle komplikuje program, pa je ovog puta taj „kućni posao“ izostavljen.

```
1500 SUB INSERT(SS,E)
1510 IF NOT LEN(E) THEN 1550
1520 IF SS="" THEN SS{2,0}-E$ GOTO 1550
1530 S=LEN(SS)
1540 IF NOT POS(SS,E) THEN SS{S,0}=""&E
1550 END SUB
```

slika 2

Operacija INSERT se sada može realizovati tako što će u US biti jednodimenzionalna lista ili se program može modifikovati (čitaj uprostiti), čime se dobija ono što vidite na slici 2. Linije 1530 i 1540 su iste kao linije 1070 i 1090, što će reći da se UNION mogao realizovati pomoću INSERT-a, ali bi se tako izgubilo nešto od brzine. Ovo važi generalno na bilo kakvo predstavljanje skupova. Operacija UNION se uvek može realizovati kao niz INSERT operacija, ali se pri tome gubi efikasnost.

```
2000 SUB DELETE(L$,AS)
2010 B=POS(L$,AS)
2020 IF NOT B THEN PRINT „Element Not Found"
GOTO 2040
2030 L$B-(B#2)+LEN(AS)-(B#2))=""
2040 END SUB
```

slika 3

Na slici 3 je program koji realizuje DELETE operaciju. Listija 2030 koristi jedan mali programerski trik, tj. koristi vrednost testa (B#2; # se piše i kao <>) kao broj za računanje „pointera“. Ovaj test služi za razlikovanje slučajeva u kome se listi odzima prvi element (tada je B#2). Zamenite test konstantom 1, probajte listi da odzume prvi element, pa će vam postati potpuno jasna svrha ovog trika.

Operacija MEMBER nije ništa drugo do POS(SS,"a")

ili, ako želite „čistu“ vrednost testa (0 ili 1): NOT NOT POS(SS,"a")

Da bi se realizovala operacija FIND treba imati niz stringova, recimo DIM L\$(N)256!. Operacija FIND se onda realizuje tako što se izvršava MEMBER za svaku listu (FOR L=1 TO N) sve dok se ne pronađe ona u kojoj je element a. Može se, međutim, oformiti i posebna lista kojoj će se dodavati indeksi (L) svih lista koje sadrže a, čime se proširuje osnovna operacija FIND.

Pošto lista generalno sadrži bilo kakve objekte, to se operacije SPLIT i MIN ne mogu izvesti. Ukoliko pak (zbog prirodne realizacije) elemente smatramo stringovima, moge bi se i ove operacije izvesti, no to ne bi imalo nekog naročitog smisla činiti. Čemu vam može služiti podatak da je „PERA“ „veći“ od „LAZE“.

Realizacija lista koja je ovde prikazana sasvim je jednostavna, toliko da možete lako eksperimentisati, i stvorena samo za rad sa skupovima. Ukoliko želite da radite sa listama koje ne predstavljaju skupove, potrebne su vam operacije za spajanje i cepanje lista, kao i mehanizam za rad sa ugnježenim listama (na primer, „1,2,[A,B,C],3,4“). U ovde predstavljenju realizaciji to znači da bi trebalo stalno prebrojavati broj otvorenih i zatvorenih vitičastih zagrada i paziti na dubinu gnježenja. Za takve operacije se već koriste list, fort ili C, a u najgorom slučaju paskal/modula-2, ali, o tome u sledećem broju.

Žarko Berberski



Klub Z80

Spektrum

Softversko merenje frekvencije klock signala

Za razne primene, kao što su precizne vremenske petlje i generisanje tona, potrebno je tačno znati frekvenciju klocka koji dolazi na mikroprocesor. Ukoliko pri ruči imamo frekvencometar ili osciloskop, i još znamo gde treba meriti, problem je rešen. Ali ništa veći posao nije napraviti program koji obavlja to merenje koristeći poznata vremena za izvršenje mašinskih instrukcija.

Brzo množenje, i treći put

Množenje sa 100 je, kako izgleda, postala prava opsesija članova kluba Z80. Ovu puta radi se o zalatu genijalnog ideji Damira Mulaje iz Zagreba. „Kolega Krešimir Kos upao je u zamku decimalnih brojevnih sustava“, piše nam Damir. Čovjeku je najbrže množenje sa sto obaviti preko dva množenja sa deset, ali ne i računalicu. Prilazim rutinu za množenje sa sto koja je još brža. To je postignuto drugačijim obrascem množenja, čime su izbačene dvije LD r,r instrukcije. Ova rutina je za 8 T stanja kraća od rutine u R35, što iznosi 7% ubrzanja, a pri tome je i dva bajta kraća. Za punih 61 T stanja, odnosno 36% brža je od rutine objavnjene u R34, a duža je od ove za samo 5 bajtova. Ne mogu tvrditi da je to najbrža moguća rutina (jedan

moj kolega kaže: „INC A je VJEROJATNO najbrži način da se uveća A registar za 1“), ali je nisam uspio smisliti bržu“.

MUL 100 LD D,H

```
LD E,L
ADD HL,HL : 2
ADD HL,HL : 3
ADD HL,HL : 6
ADD HL,HL : 12
ADD HL,HL : 24
ADD HL,HL : 25
ADD HL,HL : 50
ADD HL,HL : 100
RET
```

Rutina iz R34 ne može nositi ni epitet najkraće, jer bi program napravljen DJNZ petljom bio najmanje 5–8 bajtova kraći. U zadnjem trenutku, u Redakciji je stiglo i pismo Daniela Nikolića iz Zagreba, koji je takođe došao do istog rešenja za brzo množenje, različit broj 100 na sledeći način: $100 = 4 \cdot (1 + 8 \cdot (2 + 1))$.

Princip je vrlo jednostavan: svaka mašinska naredba izvršava se za tačno određen broj otkućaja klocka. Ako neki program ima ukupno N otkućaja, a vreme izvršenja programa je T, onda frekvencija klocka iznosi $f = N/T$.

Najbolje rešenje je napraviti neku mašinsku petlju, koja će pri svakom prolazu uvećavati neki broj u memoriji. Konačno stanje brojača, pomnoženo brojem ciklusa same petlje, daće ukupno broj n

ciklusa za čitav program. Trajanje programa, s druge strane, merićemo bilo kojim časovnikom koji pokazuje sekunde. Startovanje i prekidanje programa vršićemo na otkućaj sekundare. Greška ne može preći par desetina sekunde.

Međutim, moramo obratiti pažnju na nekoliko detalja.

Pre svega, da bi program zalista korektno radio, moramo se obezbediti od bilo kakvih prekida. Drugim rečima, mikroprocesor mora izvršavati samo naš program i ništa drugo. Zato ćemo na početku staviti instrukciju DI.

Drugi problem potiče od malo neobične organizacije „spektruma“. Klock koji dolazi na Z80 ne ide direktno sa kristala, već prvo prolazi kroz ULA čip. A ULA ima nezgodnu osobinu da povremeno prekida klock, umrtvljujući tako procesor, za kratko vreme dok se vrši prenošenje silke na ekran. Na sreću, ovo se dešava samo ukoliko procesor radi sa prvih 16K RAM-a. Na višim adresama, klock je stabilan. Zato ćemo naš program smestiti negde iznad adrese 32767.

Mašinska petlja LOOP koja je korišćena u našem primeru izvršava se za ravno 100 ciklusa. Njen zadatak je samo da testira taster SPACE i da uveća 32-bitni brojač za jedinicu. Jasno, taj posao se može obaviti i na mnogo elegantniji način, ali cilj je ovde bio da petlja traje tačno 100 taktova, radi lakšeg računanja. Neke od instrukcija tu samo stoje da bi popunile vreme.

Bežik program će dati rezultat u kilohercima, jer je tu otprilike i tačnost merenja.

Jovan Skuljan

Ispravka

U listingu programa za ispis celih brojeva, objavljenom prošlog meseca u ovoj rubrici, potkrala se jedna štamparska greška. Linija 180 bi trebalo da glasi:

180 END: DEFB 0

Izvinjavam se svim čitaocima koje je ova greška zbulnila.

Amstrad

Konverzija brojeva

Logičan nastavak prošlog izdanja Kluba Z80, koji se bavio ispisivanjem brojeva, jeste rešavanje inverznog problema: niz bajtova koji čine zapis nekog broja u ASCII kodovima treba prevesti u konkretnu brojnu vrednost u nekom registru ili registarskom paru. Svi programi koji zahtevaju bilo kakav numerički unos sa tastature moraju sadržati deo koji se bavi ovim poslom. Zato nema assemblera, prevodioca, linijskog editora ili programa za poslovnu grafiku koji nema neku rutinu koja obavlja taj zadatak.

Opet ćemo izložiti jedno rešenje koje se uz minimalne izmene može prilagoditi proizvoljnom brojnom sistemu. Takav pristup omogućuje pisanje samomodifikujućih rutina, koje zamenjuju nekoliko potprograma potrebnih za ostvarivanje istog zadatka. Sem toga, na ovaj način se lakše uočava šta je suština nekog postupka, a šta je samo prilagođavanje posebnostima pojednog slučaja.

Brzi ispis teksta

Branko Nikitović (Zlatiboraka 18, Čačak) poslao nam je dve korisne rutine. Obe služe za vrlo brzo ispisivanje teksta u modu 2. Prva prikazuje jedan znak na proizvoljnom mestu na ekranu. Ulazni parametri su u akumulatoru i BC registarskom paru:

A: kod željenog karaktera
B: Y koordinata (0<=Y<=24)
C: X koordinata (0<=X<=79)

Listing 1

```
WR:CHR: LD H,O
LD L,B
LD B,H
ADD HL,HL
LD D,H
LD E,L
ADD HL,HL
ADD HL,HL
ADD HL,HL
LD H,24
ADD HL,HL
ADD HL,HL
ADD HL,HL
ADD HL,HL
EX DE,HL
LD H,7
LD L,A
ADD HL,HL
ADD HL,HL
CALL #B906
LD B,8
LD A,(HL)
LD (DE),A
INC HL
INC HL
ADD A,8
LD D,A
DJNZ LOOP
JP #B909
```

Program za merenje frekvencije klock signala. Pri svakom prolazu kroz petlju uvećava se brojač SPACE. Početak brojanja ostavlja se pritisikom na CAPS SHIFT, a prekid sa HIDE. Rezultat brojanja na izlazu se nalazi u memoriji pošto od adrese 60000.

```
ORG 50000 Početna adresa programa.
START DI Onemogućuje sve prekide.
LD HL,#0000 Početna vrednost
LD DE,#0000 brojača je nula.
WAIT IN A,#FE
RR A,#(FE)
RRA A,(#FE)
RRA A,(#FE)
LOOP LD A,#7F Priprema za tastere B,N,M,SYMBOL (SPACE.(7).
IN A,(#FE) Očitaj izabrani red tastature. (10)
RRA Da li je pritisnut SPACE? (4)
JR NC,END Ako jeste, idi na izlaz. (7)
LD BC,#0001 Brojač će se uvećati za jedinicu. (10)
EX DE,HL Niži bajtovi brojača prelaze u HL. (4)
ADD HL,BC Brojanje za jedan. (11)
EX DE,HL Uzimanje viših bajtova brojača. (4)
LD BC,#0000 Učazće se u ožbi samo prenos. (10)
ADC HL,BC Kompletiranje brojača. (15)
INC BC Tek da prođe vreme. . (6)
JR LOOP Nazad u petlju. (12)
END LD IX,#60000 Priprema za prenos brojača u memoriju.
LD (IX+0),H Prenosenje
LD (IX+1),L brojača
LD (IX+2),D u
LD (IX+3),E memoriju.
EI Obnavljanje prekida.
RET Povratka u bežik.
```

```
10 PRINT „Merenje frekvencije klock signala“
20 PRINT
30 INPUT „Vreme u sekundama?\":t
40 PRINT „Pritisnite CAPS SHIFT za startovanje programa.“
50 PRINT „Pritisnite SPACE, posle „:t“ sekundi.“
60 RANDOMIZE USR 50000
70 LET C=PEEK 60003+256*(PEEK 60002+256*(PEEK 60001+256*PEEK 60000))
80 PRINT
90 PRINT „Y=“:INT (C/10+0.5):„kHz“
100 STOP
```


Ideja je jednostavna: kada utvrdimo da je i sledeći bajt koji analiziramo cifra, do tada pročitan deo broja pomoćno osnovom brojnog sistema i dodamo mu brojnu vrednost te cifre. HL registar, u kome se čuva međurezultat, na početku ima vrednost 0. Ako, na primer, treba pročitat niz cifara „1257“, HL će dobiti vrednosti 1, 12, 125 i 1257. U programu je korišćena rutina #DBBE za množenje. Naravno, vi možete koristiti i neku od rutina za brzo množenje koje se zasnivaju na rotaciji HL registra.

Program koji je pred vama analizira bajtove počevši od adrese na koju na ulazu ukazuje HL registar i čita sve dok ne naiđe na znak koji nije cifra. Ugašen flag prenosa signalizira da je prvi bajt nije cifra ili da je zapisani broj veći od 65535. Po izvršenju programa, u BC se nalazi pročitan broj, a HL ukazuje na prvi neobrađeni znak.

Ako vam je potrebna rutina koja čita oktalne brojeve, treba samo uneti sledeće izmene:

Naravno, sada i umesto CALL T_DEC pišemo CALL T_OCT. Verujemo da vam neće pogadati da treba uraditi za binarne brojeve. Za čitanje heksadekadnih brojeva ima malo više posla:

Potprogrami T_DEC, T_OCT, T_HEX ispituju da li se u akumulatu-

Druga rutina ispisuje na celom ekranu 2000 znakova. Labela TEXT navedena u prvj liniji listinga 2 ukazuje na prvi od 2000 bajtova koji predstavljaju tekst za ispisivanje. Ovaj potprogram se može iskoristiti u nekom vašem tekst-processoru ili za ispisivanje help-krinja.

Listing 2

```
SCREEN: LD IV,TEXT ;U IV početak teksta za ispis
        DE,#C000 ;DE početak video memorije
        (OFFSET MORA BITI 0)
LOOP.1: LD L,(Y) ;U L znak iz teksta
        HL=#L+1+65530 ;HL=#L+1+65530
        ADD HL,HL ;U HL je adresa definicije znaka u ROMu
        ADD HL,HL ;čvorovi donji dole
        PUSH DE ;čuva DE na stoku
        LD B,B ;B bajta definicije znaka
        LD A,B ;U A bajt definicije
        LD (DE),A ;savaj ga na ekran
        INC HL ;sledači bajt definicije
        LD A,B ;U DE bajt ispod na ekranu
        (DE=DE+#800)
        ADD A,D ;ako znak nije prebačen, natrag u petlju
        LD D,A ;završni donji ROM
        INC IV ;sledači znak
        POP DE ;uzime DE
        INC DE ;promezjuje slededeći znak
        LF #D0 ;uzime D
        CP E ;
        JR NZ,LOOP.1 ;ako nije završen ispis, natrag u petlju
        LD A,#C7 ;'natrag u program
        SP ;
        JR NZ,LOOP.1 ;CRET
```

Obe rutine zbog brzine imaju i neka ograničenja: OFFSET mora biti 0, koriste se definicije znakova iz ROM-a i kontrolni znakove se ne izvršavaju (ispisuju se kao i svaki drugi simbol). Sem toga, moraju biti smeštene iznad lokacije #400. Ali, za nekoliko područje primene prevazilaze ti ograničenja nije ni potrebno.

```
10 GET BC: LD A,(HL) ;Sa (HL) se čita broj
          20 CALL T_DEC ;i smelja u BC.
          30 RET NC ;Na izlazu:
          40 PUSH DE ;BC=pročitani broj
          50 EX DE,HL ;(HL)=prvi znak
          60 LD B,D ;koji nije cifra
          70 LD H,B ;CY resetovan
          80 LD L,B ;-greška
          90 LOOP: PUSH DE
          100 LD C,A
          110 LD DE,10
          120 CALL #DBBE ;HL=HL-DE
          130 LD C,ERROR
          140 ADD HL,BC
          150 JR C,ERROR
          160 POP DE
          170 INC DE
          180 LD A,(DE)
          190 CALL T_DEC
          200 LD C,LOOP
          210 EX DE,HL
          220 LD B,D
          230 LD C,E
          240 POP DE
          250 SCF
          260 RET
          270 ERROR: POP DE
          280 POP DE
          290 OR A
          300 RET
          310 T_DEC: CP #9+1
          320 RET NC
          330 CP #0
          340 CCF
          350 RET NC
          360 SUB #0
          370 SCF
          380 RET
          110 LD DE,B
          310 T_OCT: CP #8
          110 LD DE,16
          390 T_HEX: CALL T_DEC
          400 RET C
          410 CP #F+1
          420 RET NC
          430 CP #A
          440 CCF
          450 RET NC
          460 SUB #A-10
          470 SCF
          480 RET
```

Bejzik klub

Nije crno
sve što
ima gumice

Nas stalni saradnik Željko Jurić, koji je pre nekoliko meseci obukao voljnik uniformu, poslao nam je iz Novog Sada interesantan članak o osobenostima „spektrumovog“ bejzika i o njihovim prednostima i manama. Članak je dosta obiman, pa okviru Bejzik Kluba u nekoliko tematskih celina. Ovuoga puta odabrao smo deo koji se odnosi na definisanje složenih funkcija.

Funkcija VAL je na „spektru“ vrlo neobično rešena. Kod većine računara ona kao rezultat vraća broj koji je napisan u obliku stringa. Na primer:

LET A=VAL("34.8")

daje broj 34.8 upisan u promenljivoj A. Ako se na početku navedenog stringa ne nalazi broj, rezultat je nula. „Spektrumova“ funkcija VAL ima više od toga. Njome možemo da izračunamo vrednost izraza koji je napisan po pravilima bejzika, ako su definisane sve promenljive koje su upotrebljene u tom izrazu. To znatno proširuje upotrebu vrednost funkcije VAL, što možete da vidite iz sledećih primera:

```
10 DEF FN a(X)=VAL X$
20 DEF FN b(Y)=VAL X$
30 DEF FN c(X,Y)=VAL X$
40 LET X=5: LET Y=8
50 LET X$=-4*X+5*Y-3
60 PRINT FN a(2)
70 PRINT FN b(2)
80 PRINT FN c(2,2)
```

Navedeni program daje rezultate 45, 27 i 15, iako je izraz u promenljivoj X\$ isti u sva tri slučaja. U funkciji FN a, argument je X, a Y je prethodno definisana promenljiva.

Kod funkcije FN b je stvar obrnuta, a FN c ima dva argumenta, pa se i za X i za Y vrši prenos vrednosti prilikom pozivanja funkcije. U praksi se često javljaju funkcije sa više definicijama izraza, kao što je funkcija:

f(X)=3*X+7 za X < 8
f(X)=2*X-5 za X >= 8

Problem možemo da rešimo upotrebom operatera AND:

DEF FN f(X)=(3*X+7) AND X<8+(2*X-5) AND X>=8

Postupak je primenjiv sve dok su svi definiciji izrazi (u ovom slučaju: oba) definisani na skupu argumentata koji nam interesuju. Problem nastaje kada izračunavanje nekog izraza treba preskočiti, kao u slučaju:

f(X)=LN ABS X za X <> 0
f(X)=10 za X=0

Pokušajmo da navedenu funkciju definisamo na jedan od sledećih načina:

DEF FN f(X)=(LN ABS X AND X<>0)+(10 AND X=0) ili
DEF FN f(X)=(LN ABS X AND X)+(10 AND NOT X)

Formalno je sve u redu i vrednost izraza zavisi samo od toga da li promenljiva X ima vrednost nula ili je različit od nule. Ipak, za vrednost X=0 nećemo dobiti rezultat, već poruku „Invalid argument“, jer se uvek računa vrednost oba izraza, a funkcija LN nije definisana za X=0. Problem rešava sledeća definicija:

DEF FN f(X)=(VAL("LN ABS X" AND X)+(10 AND NOT X))

Ova definicija logički formira izraz, a zatim izračunava njegovu vrednost dok je prethodna prvo izračunava vrednost svih izraza, pa tek onda logički postavlja dejstvo svih izraza sem onog izraza koji vazni za navedenu vrednost argumenta X. Moguće je definisati čak i rekurzivne, na primer za funkciju faktorijel sa sledećom rekurzivnom definicijom:

f(X)=1 za X=0
f(X)=X*(f(X-1)) za X<>0

gde postupak izračunavanja za X=4 izgleda ovako: 4!=4*(3!)=4*(3*(2!))=4*(3*(2*(1!)))=4*(3*(2*(1*(0!))))=4*3*2*1=24. Definicijom izraza u bejziku bi:

DEF FN f(X)=VAL("1" AND NOT X)+4*(X*FN f(X-1) AND X)

Ovaj metod možemo koristiti i za popravljavanje nekih manje korisnih i specifičnosti „spektrumovog“ bejzika. Funkcija BIN, na primer, i nije prava funkcija, jer i za nje ne može da stoji promenljiva ili izraz, pa je možemo zameniti sa:

DEF FN b(X)=VAL("BIN"+X\$)

Sada je moguće upotrebiti je u obliku:

LET QS="1101": PRINT FN b(Q\$)

Moguće je definisati i rekurzivnu funkciju koja dekadno broj prebacuje u binarni zapis:

DEF FN b(X)=VAL("STR\$(X" AND X<2)+("STR\$(X-2)INT(X/2))+FN b\$INT(X/2) AND X>1))

Treba naglasiti da se sve ovo odnosi na „spektrumov“ bejzik i ne postoji nikakva mogućnost da se ovako nešto direktno prenese na neki drugi računar. Ovaj prilog ima zadatak i da nas uvede u problematiku interpretacije aritmetičkih i logičkih izraza na drugim računarcima kod kojih funkcija VAL i VAL\$ nisu ovako fleksibilne.

Željko Jurić

Branko Marović



Uređuje: Žarko Berberski

„Amstrad CPC 464“

Preklapanje ekranskih modova

Miloš Pavlović ima problema sa „amstradom CPC 464“ i promenom grafičkog moda. Pri svakoj promeni moda briše mu se ceo ekran, što je vrlo neugodno za programe čiji se vizuelni efekti baziraju baš na izmeni modova.

Problem koji vas muči rešite jednostavno programom koji sledi i baziran je na tehnici takozvanog „preklapanja modova“. Vlasnicima CPC 128 ne preporučujem da koriste ovaj program, jer on koristi specifičnosti „starijeg brata“, pa na CPC 128 mogu napraviti samo štetu.

Dejan Predić

```
10 MODE 2
20 X = 100 : GOSUB 80
30 X = 200 : GOSUB 70
40 X = 300 : GOSUB 80
50 END
60 POKE &B1C8, 0 : POKE &B1CF, 255 : POKE &B1D0, 15 : PRINT „Mode 0“ : GOTO 90
70 POKE &B1C8, 1 : POKE &B1CF, 192 : POKE &B1D0, 48 : POKE &B1D1, 12 : POKE
&B1D2, 3 : PRINT „Mode 1“ : GOTO 90
80 POKE &B1C8, 2 : POKE &B1CF, 128 : POKE &B1D0, 64 : POKE &B1D1, 32 : POKE
&B1D2, 15 : POKE &B1D3, 8 : POKE &B1D4, 4 : POKE &B1D5, 2 : POKE &B1D6, 1 : PRINT
„Mode 2“
90 FOR I = X TO X + 100 STEP 10
100 PLOT I, 10 : DRAW I - 100, I
110 NEXT I
120 RETURN
```

„Spektrum“

Mikrodrajz sa MONS-om i GENS-om

Vladimir Janković ima problema pri korišćenju mikrodrajza sa DEVPAC-om.

Uplis i šifiranje datoteka ši mogući samo iz GENS-a (ako zanemarimo mogućnost da to uradimo iz bajzika). Sintaksa je sledeća: P n, m, i:ime_pr — spremanje izvornog programa na mikrodrajz i G, „i:ime_pr“ — uplis izvornog programa sa mikrodrajza i O, „i:ime_pr“ — spremanje prevedenog koda na mikrodrajz 1.

Parametri n i m označavaju prvu i poslednju liniju programa koji treba snimiti. Ime programa ne treba stavljati u navodnike. Postupke prebacivanja teksta iz MONS-a u GENS je sledeći: Udi u GENS i naredbom X pročitati adresu početka teksta i sa B se vratiti u bajzik. Startovati MONS i naredbom T pokrenuti izvršni program na pitanja „First:“ i „Last:“ do početnoj i krajnjoj adresi mašinskog koda koji treba disasembirati. Odgovori heksadekadno. Zatim pitanje „Printer:“ na koje treba odgovoriti sa „Y“ ili „N“. Ključno je pitanje „Text:“ na koje treba odgovoriti adresom koja je dobijena od GENS-a pomoću komande X. Naredna pitanja „First:“ i „Last:“ se odnose na delove mašinskog koda koji ne sadrže program, nego podatke. Pitanja se ponavljaju dok na oba ne odgovorite sa ENTER. Nakon toga sledi disasembiranje koje se završava izveštajem „End of text nnnnn“. Šesnaestobitni broj nnnnn zatim treba uneti u GENS, pomoću MONS-a ili bajzika i to na adresu n+54 i n+55 (niži bajt, viši bajt), gde je n adresa na koju je učitani GENS. Nadamo se da je dovoljno jasno.

Voja Gašić

Komodor 128 Televizor kao monitor

Damir Bronzić i Srđan Čekerevac postavili su pitanja vezana za korišćenje televizora koji ima EUROCONNECTOR kao monitor.

Televizori sa EUROCONNECTOR-om mogu se priključiti kao monitori na sve računare koji imaju kompozitni ili RGBI izlaz, uz uslov da frekvencija kojom računar generiše sliku nije previše visoka. To znači da se ni PC ni „atari“ ne mogu priključiti kad rade u režimima koji daju crno-belu sliku visoke rezolucije.

Računar C-128 daje kompozitni signal samo u C-64 modu. Ako je vaš računar predviđen za NTSC-standard, onda samo visokoj toleranciji televizora treba da zahvalite što ste uspešli da dobijete i crno-belu sliku. Za bilo šta više morate izvršiti zamenu ili prepravku celog video-stepena. Čeda Andrijević (čije odgovore čitate povremenu u ovoj rubrici) će vam rado pomoći ako mu se javite na telefon 011/162-434.

Kad je C-128 u CP/M modu, daje signal samo na RGBI izlazu. Tada spojite nožice 15, 11 i 7 EUROCONNECTOR-a na nožice 3, 4 i 5 RGBI konektora. Ne zaboravite da spojite i sve masne. Nožicu 16 EUROCONNECTOR-a treba spojiti na nožicu 8 RGBI konektora i to bi trebalo da reši vaše probleme.

„Galaksija“

Štampano kolo za „galaksiju“

Grahovac Zdranko želi da nabavi štampanu pločicu, tastaturu i konektorsku pločicu za „Galaksiju“.

Štampana pločica i tastatura za „Galaksiju“ se već odavno nigde ne prodaju. Možete, naravno, naručiti od nekog privatnika da vam napravi pločicu, ali to neće biti nimalo jeftino. Ako već želite sami da napravite svoj računar razmislite o sklapanju PC-kompatibilca, za koji u Nemačkoj postoje sve komponente po vrlo pristupačnim cenil, ili se strplje još mesec-dva. Redakcija priprema čitaocima jedno veće iznenađenje.

„Komodor 64“

Problemi sa GEOS-om na C-64

Dimitrije Golovović je nabavio GEOS za svoj C-64 i sada ima velikih problema sa štampanjem teksta iz tekst-procссора. Računar mu se blokira i odbija da štampa više od jedne stranice.

Dimitrije nije jedini koji ima problema sa GEOS-om. Vrlo je teško naći kopiju ovog programa bez bagova, koji korisnicima zadaju velike brige. Problem sa vašim GEOS-om je dvojak. Imate staru verziju ovog paketa (odnosno, jedan njegov deo) i imate loše piratovani verziju. Rešenje ipak postoji. Obratite se Bojanu Žanoškuru, koordinatorskom C-64 sekcije u Klubu Programera Elektrotehničkog fakulteta, u Beogradu, (tel.: 585-833).

Šarp 1500

Program za „Šarp 1500“

Milan Radičić traži adresu neke firme u Nemačkoj kod koje se mogu nabaviti kvalitetni programi za „Šarp 1500“.

Nemačka firma Günter Holtkötter GmbH, Albert Schweitzer ring 9, 2000 Hamburg, BRD, prodaje nekoliko izuzetno kvalitetnih programa za ŠARP-1500A ili običan ŠARP 1500 sa memoriziranim proširenjem. Najznačajniji program je PC-BASIC 84 koji predstavlja dvadesetak novih naredbi, među kojima je i proširenje seta numeričkih funkcija. Sa ovim programom ŠARP-1500A dobija kompletno strukturalan bajzik koji, osim toga, poznaje procedure sa lokalnim varijablama i definisanje funkcija koje mogu biti i vilišinski-PC-Macro je izvrstan makro i kondicionalni assembler čiji je izvršni obzično naginju „humanim assemblerima“ i omogućavaju lak i brz razvoj mašinskih programa. PC-Work predstavlja kolekciju programa koje možete koristiti kao potprogram u vlastitom programerskom radu. Među njima vredni izdvojni program koji višestruko ubrzava komunikaciju sa kasetofonom. PC-FORTH je potpuna implementacija ovog programskog jezika u svega 6 K. Ako znamo da je ovaj jezik po svojoj brzini blizak mašinskom kodu i da je i od mnogo lošijih računera činio čudo, onda je jasno zašto ga je ova firma uvrstila u sam vrh svoje ponude. **Nenad Veres**

Dejanović
pitalice

Sanja voli Madonu

Volite li lake zadatke? U toku januara primili smo nekih 260 tačnih i petnaestak pogrešnih rešenja sedamnaeste pitalice — nije baš apsolutni rekord, ali u poredenju sa dva decembarna odgovora... lako do rešenja nije bilo teško doći, redovni i veoma pažljivi čitaoci „Računara“ su bili u prilici da uštede dosta posla zahvaljujući jednoj „skrivenoj poruci“!

Podsetimo se, pre svega, priče — nastavka 14. pitalice koja se, sećamo se, odnosila na pet mladih parova zainteresovanih za kompjutere i (pomalo) alkohol. Juca, Sanja, Maca, Ceca i Mira i njihovi mladići Laza, Steva, Mika, Pera i Žika su

zajedno dočekivali Novu 1988. godinu. U kasne jutarnje sate društvo se razišlo, pa su Mira i Žika pristigli u svoju zelenu kuću na samom kraju ulice. Žika je, razume se, popio malo više viskija nego što je trebalo, pa se odmah svalio na foteľju, zaspaio i usnuo čudnu stvar: sanjao je da se društvo

razišlo ali da je svih pet devojaka pošlo sa njim u diskot Sve su, prirodno, želele da igraju sa njim, a on je ustao, prišao jednoj, pružio ruku i... probudio se. Pomalo ljut na Miru koja ga je drmusala, Žika joj je ispričao svoj san. Mira se narogušila i pitala „Pa, koju si izabrao?“ Suvviše mamuran da

```
#include <stdio.h>
#define JUCA 0
#define SANJA 1
#define MACA 2
#define CECA 3

main()
{
    register int i;
    char oci[4], kosa[4], visina[4], vm[4], b1, b2, b3, b4, san;
    static char ime[4][6] = { "Juca", "Sanja", "Maca", "Ceca" },
    dn[2][3] = { "ne", "da" };

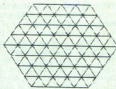
    for ( oci[0]=0; oci[0]<=1; oci[0]++ )
    for ( kosa[0]=0; kosa[0]<=1; kosa[0]++ )
    for ( visina[0]=0; visina[0]<=1; visina[0]++ )
    for ( vm[0]=0; vm[0]<=1; vm[0]++ )
    for ( oci[1]=0; oci[1]<=1; oci[1]++ )
    for ( kosa[1]=0; kosa[1]<=1; kosa[1]++ )
    for ( visina[1]=0; visina[1]<=1; visina[1]++ )
    for ( vm[1]=0; vm[1]<=1; vm[1]++ )
    for ( oci[2]=0; oci[2]<=1; oci[2]++ )
    for ( kosa[2]=0; kosa[2]<=1; kosa[2]++ )
    for ( visina[2]=0; visina[2]<=1; visina[2]++ )
    for ( vm[2]=0; vm[2]<=1; vm[2]++ )
    for ( oci[3]=0; oci[3]<=1; oci[3]++ )
    for ( kosa[3]=0; kosa[3]<=1; kosa[3]++ )
    for ( visina[3]=0; visina[3]<=1; visina[3]++ )
    for ( vm[3]=0; vm[3]<=1; vm[3]++ )

    /* prva 4 uslova? */
    if ( oci[SANJA] == oci[MACA] && kosa[JUCA] == kosa[MACA] &&
        visina[SANJA] == visina[CECA] && vm[JUCA] == vm[CECA] )

    {
        /* druga 4 uslova? */
        b1=b2=b3=b4=0;
        for (i=JUCA; i<=CECA; ++i)
        {
            if ( oci[i] && kosa[i] ) ++b1; san=i;
            if ( visina[i] && vm[i] ) ++b2;
            if ( visina[i] && kosa[i] ) ++b3;
            if ( oci[i] && vm[i] ) ++b4;
        }
        if ( b1==1 && b2==2 && b3==2 && b4==3 )

        {
            /* pronasli smo je! */
            printf ( "\n%ks je devojka iz snova.\n", ime[san] );
            printf ( "\n%29s%15s%15s%15s", "zelene oci", "plava kosa",
                "visoka", "voli Madonu" );
            for (i=JUCA; i<=CECA; ++i)
                printf ( "\n%10s%15s%15s%15s", ime[i], dn[oci[i]][0],
                    dn[kosa[i]][0], dn[visina[i]][0], dn[vm[i]][0] );
            printf ( "\n\n"); exit (0); /* nazad u OS */
        }
    }
}
```

29: Zbir brojeva na svakoj od dijagonala je _____



Ime i prezime _____

Adresa _____

Mesto _____

Računar _____

Vreme _____

Identifikacioni broj: ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
 iz broja tel. god.rodj. po izboru

```
10 REM
20 REM   Resenje 28. Pitalice
30 REM
40 REM   "Računari 36"
50 REM
60 REM
70 TX=TIME
80 DIM NIZ(16)
90 MX=0
100 FOR I=1 TO 16
110 NIZ(I)=0
120 NEXT I
130 IF NIZ(1)=NIZ(2) THEN 250
140 IF NIZ(6)=NIZ(7) THEN 250
150 IF NIZ(12)=NIZ(13) THEN 250
160 IF NIZ(15)=NIZ(16) THEN 250
170 IF FNab(1,2)=1 THEN 250
180 IF FNab(3,4)=2 THEN 250
190 IF FNab(5,6)=2 THEN 250
200 IF FNab(1,4)=2 THEN 250
210 FOR I=1 TO 16
220 PRINT NIZ(I);
230 NEXT I
240 PRINT
250 GOSUB 350
260 GOTO 130
270 DEFNAB(I,J)=
280 LOCAL I
290 ZB=0
300 FOR I=1 TO 4
310 IF NZ(FNab(1,I))+NIZ(FNab(2,I))+2 THEN ZB=ZB+1
320 NEXT I
330 ZB=ZB
340 DEFNAB(I,J)=*(I-1)+
350 REM SLIDE1
360 I=1
370 IF NZ(I)=1 THEN 410
380 NIZE=1
390 IF I=MX THEN MX=I
400 RETURN
410 NIZE=0
420 I=I+1
430 IF I=16 THEN 370
440 PRINT "Nema vise resenja"
450 PRINT
460 PRINT "Vreme: %TIME-TY/100" a;"
470 END

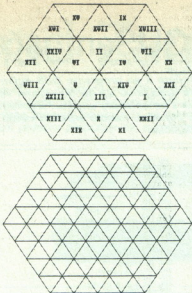
REM
1110100111011110
1110100110111110
Nema vise resenja!
```

Tajna smederevske tvrđave

Priča se da je u smederevsku tvrđavu (građena je negde 1430. godine) uzidavan kamen nastao raspokapanjem nekog prastarog rimskog naselja. Tom prilikom pronađen je kamen u koji je uklesan pravilan šestougao izdvojen na 24 jednakostranična trougla; u svaki trougao bio je upisan po jedan od brojeva 1—24, tako da je zbir u svakom redu paralelnom sa glavnom dijagonalom jednak i iznosi 75. Učeni smederevci toga doba zaključili su da je šestougao veoma značajno delo i pripisali mu određene magične osobine — obzirom da je smederevska tvrđava građena u obliku trougla, pretpostavljeno je da ona predstavlja samo jedno polje magičnog šestougla i da će svi važni događaji na neki način biti povezani za upisanim brojevima. Zato je odlučeno da se kamen ugradi na neko vidno mesto. Ali, želje su jedno a život drugo: neoprezni radnici su ispustili kamen i on je završio na dnu Dunava!

Slika 4 prikazuje mogući izgled smederevskog kamena — obzirom na rimsko poreklo, pretpostavili smo da su korišćene rimske cifre 29. pitalica zahteva da u skladu sa sličnim pravilima popunite magični šestougao sa slike 5 — u njega treba upisati brojeve između 1 i 96 (zaboravite na rimske cifre!), tako da zbir po svim dijagonalama paralelnim sa nekom od glavnih dijagonala bude jednak.

Rešenja, zajedno sa programima koje ste koristili da ih dobijete, šaljite na adresu: "Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd



tako da pristignu pre 25. marta 1988. Sva pisma sa korektnim rešenjima konkurisu za novčane nagrade od 60,000, 40,000 i 20,000 dinara dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete poslednje tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj.

Q zadatak

RQ4: Velika kazaljka Big Bena teži 1200, a mala 700 funti. Koliko funti teže kazaljke Big Bena? Naravno, ne 1900 nego 7600 — Big Ben ima četiri lica!
Q5: Anagramirajte ČAJ NERED tako da se bolje jedne reč.

je zelene oči i plavu kosu i obožavala Madonine pesme. Ispuviše svesna da ne ispunjava te uslove, Mira je pokušala da zaključni na koju končinicu treba da bude ljubomorna. Pristisla se da Sanja i Maca imaju istu boju oči, Maca i Juca istu boju kose, da su Sanja i Ceca jednako visoke i da Juca i Ceca nikada ne idu zajedno na koncerte. Potom je zaključila da samo jedna devojka ima plavu kosu i zelene oči, da samo dve visoke devojke vole Madonine pesme, da samo dve visoke devojke imaju plavu kosu i da čak tri zelenooke slušaju Madonu. Ko je suparnica?

Devojka iz snova

U konkurenciji su četiri devojke (Ceca, Juca, Maca i Sanja) i četiri osobine (visoka, zelenooka, plavokosa i „obožava Madonu“)

— svakoj osobini može se pridružiti po jedna Bulova promenjiva (*true* ili *false*), pošto se implicitno pretpostavlja da je, ako boja kose nije plava, sasvim svesjedno da li je ona crna ili ruda. Kombinacija, sve u svemu, ima 2¹⁶ = 65536; dovoljno malo da se do rešenja dođe primenom programa sa slike 1 ili 2 i do „zavota programera“ — BBC B je izvršio bezijk program za 8 minuta, dok je IBM PC za svega 2 sekunde izvršio sličnu rutinu pisanu na C-u (ko je rekao da C ne može da se koristi i za rešavanje logičkih problema?).

Programi, sa slikama 1 i 2, analiziraju dešenja zadatak: zadati uslovi nisu dovoljni da se odredi da li je Juca viša od Mace ili nije. U ova slučaja, međutim, „devojka iz snova“ je Sanja, što znači da je odgovor na postavljeno pitanje jednoznačan.

Do rešenja može da se dođe i bez primene računara, na primer metodom eliminacije. Pretpostavimo da je devojka iz snova Maca. Tada Juca ima plavu kosu i ne-zelene oči, a Sanja zelene oči i ne-plavu kosu. Pošto tri zelenooke devojke slušaju Madonu, a Jucine oči nisu zelene, ljubiteljke Madone su Ceca, Maca i Sanja. Sanja i Ceca su po pretpostavki jednako visoke, što dolazi u kontradikciju sa tvrdnjom da „samo dve visoke devojke slušaju Madonu“ — početna pretpostavka se pokazala pogrešnom. Sličnim rezonovanjem mogu se eliminisati Juca i Ceca.

Osim programa koji se svode na sliku 1 i rešenja dobijenih bez primene računara, primili smo mnogo brzih kompjuterskih rešenja — odgovor se na „spektrumu“ dobija posle svega par sekundi! U čemu je tajna? Uslovi zadatak se mogu manuelno transformisati, tako da se kompjutersko pretraživanje svode na daleko manji broj kombinacija, na primer njih 256. Ovakva rešenja su, na žalost, predugečka da bismo ih ovde publikovali.

Mnogi rešavači su primetili da pitanju „ko je devojka iz snova“ najviše odgovara ime Sanja. Malobrojni pažljivi čitaoci našeg časopisa dosetili su se da potraže „Računare 21“ i pročitali da je... Sanja Stevina devojka i zato Žika nije viski... Simpatija, dakle, nije baš skorašnjak.

Brz program i popularno objašnjenje doneli su prvu nagradu (60.000 dinara) Petru Mitliču iz Beograda. Preostale dve nagrade smo izvukli — Fortuna se nasme, šala Alanu Kureševiću iz Zagreba (40.000 dinara) i Željku Moštaku iz Suseda (20.000 dinara). Pohvale zaslužuju Željko Barbić, Saša Đukić, Dragan Sretenović, Ljupčo Taseki i Slavoljub Vukoljčić.

Završavajući diskusiju 27. pitalice, pozivamo vas na nastavak priče u sledećim novogodišnjim „Računarima“. Zanimljiv razvoj događaja je garantovan!

Još malo prostih brojeva

Još jednom se pokazalo da čitaoci „Računara“ prate svetske trendove — naš mali apel iz „Računara 34“ prapraćen tajanstvenom formulom koja generiše proste brojeve izazvao je nekoliko reakcija. Od Dorda Čubrčića iz Beograda smo, pre svega, dobili kopiju rada *Diophantine Representation of the Set of Prime Numbers* koji su James Jones, Daihachiro Sato, Hideo Wada i Douglas Wiens objavili u časopisu *American Mathematical Monthly* 83, 449—464 — ispostavlja se da polinom koji smo prepisali

bi pružio jedini spasonosan odgovor, Žika je, pre nego što je nanovo zasopao, promislao nešto poput Ah, bila je visoka, imala

```

10 REM
20 REM Generisanje N-tog prostog
30 REM broja za zadato N
40 REM
50 REM (Demonstracija)
60 REM
70 REM "Računari 36"
80 REM
90 REM
100 INPUT "Unesi redni broj: " N
110 IF N=0 THEN GOTO 270
120 S=1
130 FOR I=1 TO N*N
140 P=0
150 FOR J=1 TO I
160 P+=I
170 FOR K=1 TO J-1
180 P+=K*K
190 P+=MOD J
200 NEXT K
210 P=P+P
220 NEXT J
230 IF P=N THEN S=S+1
240 NEXT I
250 PRINT N"; prost broj: "S
260 GOTO 100
270 END

```

```

REM
Unesi redni broj: 1
1. prost broj: 2
Unesi redni broj: 2
2. prost broj: 3
Unesi redni broj: 3
3. prost broj: 5
Unesi redni broj: 5
5. prost broj: 11
Unesi redni broj: 7
7. prost broj: 17

```

GUŠan gaće ga, ga, ga

Sećate li se programa GUS kojim smo u „Računarima 32“ pokušali da dokažemo da se takozvana veštačka inteligencija može razvijati čak i na običnom bejkiku? Zahvaljujući doprinosu Saše Spasića, GUS je konačno naučio srpskohrvatski jezik i počeo da „izgovara“ svoje zaključke!

u „Računare 34“ zaista generiše sve proste brojeve: treba nam izabrati proizvoljne negativne cele brojeve a-z (zanimljivo je da je engleska azbuka tačno dovoljna za obeležavanje koeficijenta ovog polinoma) i izračunati vrednost polinoma. Ako je vrednost negativna, treba je ignorisati; ako je pozitivna, dobili smo još jedan prost broj. Džons i njegovi koautori su dokazali da je skup prostih brojeva zapravo identičan sa skupom negativnih vrednosti pomenutog polinoma.

Godišnje takmičenje

Najbolji u prošlog godini Milan Grbić iz Sremčice kod Beograda

U prvom godišnjem takmičenju rešavača Pitalica učestvovala su 1132 čitaoca „Računara“ — primili smo, jasno, i dosta rešenja čiji autori nisu pretendovali na godišnje nagrade, ali njihov broj, nismo evidentirali.

Prvo mesto osvojio je Milan Grbić iz Sremčice (id. broj 4356611) koji je sakupio 15 poena — njemu pripada naša godišnja nagrada od 100.000 dinara. Drugo mesto zauzeo je Milovan Kovačević iz Šida (id. broj 5856309), dok je treće mesto podelilo čak šest takmičara sa po 9 poena — Željko Barbić iz Zabočka (0185263), Dragica Dimitrijević iz Niša (4656053), Gorazd Divjak iz Brestanice (5166011), Primož Gabrijelečić iz Ljubljane (2936900), Predrag Miletić iz Foče (0707070) i Siniša Stamatović iz Sremske Mitrovice (0616319). Slede 4 rešavača sa po šest poena (id. brojevi 2805347, 9536207, 1488060 i 4066253), 14 rešavača sa po pet poena, 15 rešavača sa po 4 poena, 46 rešavača sa po tri poena, 89 rešavača sa po 2 poena i najzad 836 rešavača sa po jednim poenom. Razgovor sa pobednicima donosimo u okviru posebnog napisa.

Problem koji je vekovima mučio matematičare je, dakle, rešen — nije potrebno konstruisati formulu koja za zadato N generiše N-ti prost broj. U radu *Formula for the nth prime number* (Canadian Math. Bull. 18 no. 3) James P. Jones predlaže oblik:

$$P(n) = \sum_{i=0}^{n^2} \left(1 - \prod_{j=0}^i r(j-i)^2 \cdot j \right) \cdot n$$

gde je r(a,b) zapravo a MOD b uz dodatak r(a,0)=a. Slično tome, minus ne označava obično oduzimanje: a-b je definisano kao MAX(a-b,0). Ilustracije radi, formulu smo pretvorili u bejkik program sa slike 3 — unesite N i pročitajte N-ti prost broj! Nevolja sa programom je što je osetno sporiji od par hiljada godina starog Eratostenskog sita — značaj formule je uglavnom teorijski. Obzirom da detaljnije diskusije teorije brojeva izlaze izvan okvira Pitalice, pažljivo ćemo proučiti američki rad i vredne priloge drugih čitalaca i pripremiti napis o ovoj veoma zanimljivoj temi za neki od sledećih brojeva „Računara“.

```

3000 REM
3010 REM
3020 REM
3030 REM
3040 REM
3050 REM
3060 REM
3070 REM
3080 REM
3090
3100 DIM say$(100)
3110 INPUT a$
3120 PRINT FNsay(a$)
3130 GOTO 3110
3140
3150 DEFN say(i)
3160 LOCAL ad,sp,kuln$
3170 rem=0
3180 FOR sp=20 TO 0 STEP -1
3190 rem=5*10^22*sp^i
3200 say$(sp)=sp^i*prethodno elementa *
3210 NEXT sp
3220 sp=21
3230 say$(sp)=prethodno elementa *
3240 IF LEN(say)=0 THEN sp=0
3250 FOR plo=1 TO LEN(say)
3260 ad=MEM(say,plo,1)
3270 IF ad=sp THEN 3320
3280 sp=sp+1:IF ad=sp THEN ad=sp^i
3290 say$(sp)=prethodno elementa * ad+sp
3300 NEXT plo
3310 "Misliti da je sledeći element vrednost say$(sp)+1"
3320 IF ad=sp THEN 3480
3330 IF ad=sp^i THEN 3480
3340 IF ad=sp^i THEN 3520
3350 IF ad=sp^i THEN 3610
3360 IF ad=sp^i THEN 3580
3370 IF ad=sp^i THEN 3550
3380 IF ad=sp^i THEN 3780
3390 IF ad=sp^i THEN 3730
3400 IF ad=sp^i THEN 3760
3410 IF ad=sp^i THEN 3670
3420 IF ad=sp^i THEN 3640
3430 IF ad=sp^i THEN 3700
3440 rem=0
3450 rem=0
3460 REM p
3470 say$(sp)=sita say$(sp)+1
say$(sp)=1:sp=sp-1
3480 GOTO 3300
3490 REM m
3500 say$(sp)=sita say$(sp)+1
say$(sp)=1:sp=sp-1
    
```

```

3510 GOTO 3300
3520 REM t
3530 say$(sp)=1:FNsita say$(sp)+1
say$(sp)=1:sp=sp-1
3540 GOTO 3300
3550 REM d
3560 say$(sp)=1:FNsita say$(sp)+1
say$(sp)=1:sp=sp-1
3570 GOTO 3300
3580 REM r
3590 say$(sp)=1:FNsita pri odnosa say$(sp)+1
3600 GOTO 3300
3610 REM e
3620 say$(sp)=FNsita sita prethodno gde
sita ima vrednost say$(sp)+1
a=1:FNsita say$(sp)+1
3630 GOTO 3300
3640 REM W
3650 sita=say$(sp)+say$(sp)+say$(sp)+1
say$(sp)=1:sp=sp-1
3660 GOTO 3300
3670 REM C
3680 sp=sp+1:say$(sp)=say$(sp)+1
3690 GOTO 3300
3700 REM O
3710 sp=sp-1
3720 GOTO 3300
3730 REM p
3740 say$(sp)=prethodno broj čiji je medij broj
jednak vrednost say$(sp)
3750 GOTO 3300
3760 REM S
3770 say$(sp)=FNsita sija i say$(sp)
3780 GOTO 3300
3790 REM u
3800 say$(sp)=say$(sp)+prethodno elementa *
3810 GOTO 3300
    
```

```

REM
FN$
Misliti da je sledeći element vrednost prethodno broja 2
i sita prethodno elementa i 2. prethodno elementa.
FNm
Misliti da je sledeći element vrednost razlike prethodno
broja 10 i prethodno elementa i pozicije elementa.
FNp
Misliti da je sledeći element vrednost prethodno prethodno
broja čiji je medij broj jednaka vrednost prethodno
elementa i prethodno elementa.
FNd
Misliti da je sledeći element vrednost prethodno broja 2
i razlike prethodno elementa i pozicije elementa.
    
```

GUS je, da se podsetimo, program koji, na osnovu nekoliko zadatih elemenata, generiše ostatak sekvence: sledeći element sekvence 1, 1, 4, 10, 28, na primer, treba da bude 76, jer je $a(n) = 2^2(a(n-1) + a(n-2))$. Program GUS se zasniva na specijalnom jeziku (GUSL) koji je uveliko sličan fortu — formula koja generiše prethodnu sekvencu iskazana terminološki GUSL-a glasi p2t, pri čemu su naredbe GUSL-a opisane u „Računarima 25“. U „Računarima 25“ smo, međutim, najavili da će GUS pronaći relacije ispisivati rešenja, a ne samo formulu, ali proceduru SAY smo ostavili za neku drugu priliku. Tu su, na sreću, našli čitaoci: primili smo nezavisno razvijene procedure SAY koje su sastavili Saša Spasić iz Paraćina i Gorazd Tošar iz Ljubljane. Procenivši da se prva procedura ponaša „inteligentnije“, odlučili smo da je predstavimo čitaocima „Računara“ Slika 1 prikazuje funkciju SAY zajedno sa pratećim programom za testiranje — primetimo da SAY radi na sličnom principu kao i GUS, tj. zasniva se na steku say\$(

i njegovom pokazivaču sp. Na istoj slici prikazali smo i nekoliko rečenica koje je SAY „izgovorio“ opisujući relacije iz „Računara 25“ i „Računara 32“. Neke rečenice su, istini za volju, pomalo rogotabate, ali verujemo da su dovoljno jasne — od bejkik programa dugog stotinak redova teško se može očekivati da imitira stili live Andrića!

285 DIM say\$(100)

```

2000 IF FNrun(a$)=0 THEN
PRINT FNsay(a$):PRINT
PRINT "Sledeći element je "a$(sp)!"
    
```

Slika 2 prikazuje da izmene koje treba uneti u GUS („Računari 32“) da bi inicijalizovao i u pravom trenutku izvršio proceduru SAY; posle izmena program može prenumerisati.

„Računari“ su zainteresovani za priloge čitalaca koji bi se bavili unapređenjem GUSL-a i generatora hipoteza.

Dejan Ristanović

Dva se vola u programu u bola

U prošlom broju su date opšte ideje o konstrukciji simulatora borbe dva programa u memoriji, tzv. MARS-a. Za ovaj broj smo napisali simulator na turbo paskalu, a za sledeće pripremamo dva ljuta borca.

Simulator je testiran na IBM PC AT uz pomoć TURBO PASCAL-a V3.0. Međutim, prilikom testiranja pojavio se jedan značajan problem. Verzija 3.0 može da adresira svega 64 K podataka u memoriji, što je bilo

isuviše malo za naših 8000 memorijskih lokacija. Postojalo je nekoliko rešenja: ili smanjiti broj memorijskih lokacija, ili predefinisati strukture, ili, pak, pronaći verziju koja može da adresira datih 8000 memorijskih

skih lokacija. Najjednostavnije i, u isto vreme, najkorektnije rešenje je predefinisati strukturu memorijske lokacije tako što će, umesto da pamti i kod i mmemonik instrukcije, simulator pamti samo kod, na osnovu

```

PROGRAM MARS;
TYPE
  argument = RECORD
    adr_type : char;
    value   : integer;
  END;
  mem_loc = RECORD
    code   : char;
    arg_A,arg_B : argument;
  END;
  str = STRING [3];

VAR
  fi,fi2 : FILE OF mem_loc;
  pc      : ARRAY [0..999] OF mem_loc;
  mem     : ARRAY [1..2] OF integer;
  end_of_war : boolean;
  index     : 1..2;
  instruction : char;
  file_1,file_2 : STRING [12];

FUNCTION memloc(i:integer) : str;
BEGIN
  CASE mem[i].code OF
    '0' : memloc := '00';
    '1' : memloc := '00V';
    '2' : memloc := '00V';
    '3' : memloc := '00V';
    '4' : memloc := '00V';
    '5' : memloc := '00V';
    '6' : memloc := '00V';
    '7' : memloc := '00V';
    '8' : memloc := '00V';
  END
END;

FUNCTION adr_A : integer;
BEGIN
  adr_A := (pc[index]).mem[pc[index]].arg_A.value MOD 8000;
  IF mem[pc[index]].arg_A.adr_type='0'
  THEN adr_A := (adr_base+adr_A).arg_B.value MOD 8000
END;

FUNCTION adr_B : integer;
BEGIN
  adr_B := (pc[index]).mem[pc[index]].arg_B.value MOD 8000;
  IF mem[pc[index]].arg_B.adr_type='0'
  THEN adr_B := (adr_base+adr_B).arg_B.value MOD 8000
END;

PROCEDURE error;
BEGIN
  writeln('Error in',pc[index]);
  err := TRUE;
  end_of_war := TRUE
END;

FUNCTION ty_er_A : boolean;
BEGIN
  ty_er_A := (mem[pc[index]].arg_A.adr_type='1') OR
(mem[pc[index]].arg_A.adr_type='0')
END;

FUNCTION ty_er_B : boolean;
BEGIN
  ty_er_B := (mem[pc[index]].arg_B.adr_type='1') OR
(mem[pc[index]].arg_B.adr_type='0')
END;

PROCEDURE data;
BEGIN
  end_of_war := TRUE
END;

PROCEDURE move;
BEGIN
  IF ty_er_B
  THEN error
  ELSE
  BEGIN
    IF mem[pc[index]].arg_A.adr_type='1'
    THEN WITH mem[adr_B] DO
      BEGIN
        code := '0';
        arg_A.adr_type := '1';
        arg_A.value := mem[pc[index]].arg_A.value;
        arg_B.adr_type := '1';
        arg_B.value := 0
      END
    ELSE mem[adr_B] := mem[adr_A];
    pc[index] := (pc[index]+1) MOD 8000
  END
END;

```

```

END;

PROCEDURE addition;
VAR a : integer;
BEGIN
  IF ty_er_B
  THEN error
  ELSE
  BEGIN
    IF mem[pc[index]].arg_A.adr_type='1'
    THEN a := mem[pc[index]].arg_A.value
    ELSE a := mem[adr_A].arg_B.value;
    mem[adr_B].arg_B.value := (mem[adr_B].arg_B.value+a) MOD 8000;
    pc[index] := (pc[index]+1) MOD 8000
  END
END;

PROCEDURE subtraction;
VAR a : integer;
BEGIN
  IF ty_er_B
  THEN error
  ELSE
  BEGIN
    IF mem[pc[index]].arg_A.adr_type='1'
    THEN a := mem[pc[index]].arg_A.value
    ELSE a := mem[adr_A].arg_B.value;
    mem[adr_B].arg_B.value := (mem[adr_B].arg_B.value-a) MOD 8000;
    pc[index] := (pc[index]+1) MOD 8000
  END
END;

PROCEDURE jump;
BEGIN
  IF ty_er_A
  THEN error
  ELSE pc[index] := adr_A
END;

PROCEDURE jump_seri;
BEGIN
  IF ty_er_A OR ty_er_B
  THEN error
  ELSE IF mem[adr_B].arg_B.value=0
  THEN pc[index] := adr_A
  ELSE pc[index] := (pc[index]+1) MOD 8000
END;

PROCEDURE jump_grat;
BEGIN
  IF ty_er_A OR ty_er_B
  THEN error
  ELSE IF mem[adr_B].arg_B.value=0
  THEN pc[index] := adr_A
  ELSE pc[index] := (pc[index]+1) MOD 8000
END;

PROCEDURE dec_jump_seri;
BEGIN
  IF ty_er_A OR ty_er_B
  THEN error
  ELSE
  BEGIN
    mem[adr_B].arg_B.value := (mem[adr_B].arg_B.value-1) MOD 8000;
    IF mem[adr_B].arg_B.value=0
    THEN pc[index] := adr_A
    ELSE pc[index] := (pc[index]+1) MOD 8000
  END
END;

PROCEDURE compare;
BEGIN
  IF ty_er_A OR ty_er_B
  THEN error
  ELSE
  BEGIN
    IF (mem[adr_A].code=mem[adr_B].code) OR
(mem[adr_A].arg_A.adr_type=mem[adr_B].arg_A.adr_type) OR
(mem[adr_A].arg_B.value=mem[adr_B].arg_B.value) OR
(mem[adr_A].arg_B.adr_type=mem[adr_B].arg_B.adr_type) OR
(mem[adr_A].arg_B.value=mem[adr_B].arg_B.value)
    THEN pc[index] := (pc[index]+1) MOD 8000;
    pc[index] := (pc[index]+1) MOD 8000
  END
END;

PROCEDURE run;
BEGIN
  err := FALSE;
  end_of_war := FALSE;
  index := 1;
  REPEAT
    writeln('PC',index,' : ',pc[index]);
    CASE mem[pc[index]].code OF

```



```

'0' : data;
'1' : move;
'2' : addition;
'3' : subtraction;
'4' : jump;
'5' : jump_zero;
'6' : jump_greater;
'7' : dec_jump_zero;
'8' : compare;

END;
index := 3-index;
UNTIL end_of_war;
writeln;
IF not FALSE
THEN writeln('Program ',index,' is winner!')
END;

PROCEDURE pc_1;
BEGIN
writeln;
writeln('PC 1 := ',pc[1]);
END;

PROCEDURE pc_2;
BEGIN
writeln;
writeln('PC 2 := ',pc[2]);
END;

PROCEDURE quit;
BEGIN
writeln;
writeln;
writeln('Good bye,');
END;

PROCEDURE clear;
VAR i : 0..9999;
BEGIN
FOR i:=0 TO 9999 DO WITH mem[i] DO
BEGIN
code := '0';
arg_aadr_type := '1';
arg_a_value := 0;
arg_aadr_type := '1';
arg_b_value := 0;
END;
END;

PROCEDURE mem_scan;
VAR i,j,top,bottom : 0..9999;
BEGIN
writeln;
write('top:');
readln(top);
write('bottom:');
readln(bottom);
writeln;
IF top=bottom
THEN bottom := bottom+8000;
FOR i:=top TO bottom DO
BEGIN
j := i MOD 2000;
writeln(mem[j].code);
write(' ');
writeln(mem[j].arg_aadr_type);
writeln(mem[j].arg_a_value);
writeln(mem[j].arg_b_aadr_type);
writeln(mem[j].arg_b_value);
writeln;
END;
END;

FUNCTION code_error(i:integer) : boolean;
BEGIN
code_error := (ord(mem[i].code) < 48) OR (ord(mem[i].code) > 56)
END;

FUNCTION type_error(i:integer) : boolean;
END;

```

```

type_error := (mem[i].arg_aadr_type = '1') OR
(mem[i].arg_aadr_type = '0') OR
(mem[i].arg_aadr_type = '8') AND
(mem[i].arg_b_aadr_type = '1') OR
(mem[i].arg_b_aadr_type = '0') OR
(mem[i].arg_b_aadr_type = '8') OR
END;

PROCEDURE input_files;
VAR i,p : integer;
ty_err : boolean;
BEGIN
writeln;
write('File of program 1 : ');
readln(file1);
assign(f1,file1);
p := random(2000);
pc[1] := p;
pc[2] := p;
reset(f1);
WHILE NOT eof(f1) DO
BEGIN
read(f1,mem(p));
IF code_error(p)
THEN writeln('Syntax error in ',p);
IF type_error(p)
THEN writeln('Type error in ',p);
p := (p+1) MOD 2000;
END;

writeln;
write('File of program 2 : ');
readln(file2);
assign(f2,file2);
p := random(2000);
pc[2] := p;
pc[1] := p;
reset(f2);
WHILE NOT eof(f2) DO
BEGIN
read(f2,mem(p));
IF code_error(p)
THEN writeln('Syntax error in ',p);
IF type_error(p)
THEN writeln('Type error in ',p);
p := (p+1) MOD 8000;
END;
END;

PROCEDURE menu;
BEGIN
writeln;
writeln('Options:');
writeln;
writeln('e - scan memory');
writeln('C - clear memory');
writeln('i - input files');
writeln('R - run');
writeln('Q - quit');
writeln('1 - PC 1');
writeln('2 - PC 2');
writeln;
END;

clear;
pc[1] := 0;
pc[2] := 0;
REPEAT
menu;
readln(kbd,instruction);
CASE instruction OF
'M' : mem_scan;
'C' : clear;
'i' : input_files;
'R' : run;
'Q' : quit;
'1' : pc_1;
'2' : pc_2;
END
UNTIL instruction='Q'
END;

```

koga kasnije generiše i mnemonik. Pri tome, kod instrukcije je tip *chr* kako bi se uštedelo još jedan bajt po instrukciji. To donekle otežava unos programa u fajlove, ali se to može rešiti jednim malim programom za učitavanje fajlova. Uostalom, ukoliko neko može adresirati 80kB podataka, uz malu izmenu programa može koristiti originalnu strukturu memorijske lokacije, jer ona ne utiče bitnije na konceptiju simulatora.

Simulator je sastavljen od nekoliko glavnih procedura, od kojih je jedna zadužena za interpretaciju programa, dok ostale uglavnom služe za njenu pripremu.

Glavni deo programa je, praktično, jedan meni iz kojeg se pozivaju željene opcije. To su skeniranje memorije, njeno resetovanje, unos programa, njihova interpretacija i očitavanje PC registra.

Predviđeno je da se svi programi-borci nalaze u obliku dotoketa na spoljnoj memoriji, odakle bi, po potrebi, bili pozivani procedurom *inputfiles*. Time je omogućeno formiranje biblioteke i izbegnuta je potreba

za prethodnim storičanjem programa u bafere, već se oni direktno unose u memoriju. Nakon završenog unosa, automatski se vrši očitavanje sadržaja PC registra. Ukoliko su vrednosti koda instrukcije ili tipova adresiranja prilikom unosa pogrešne, simulator ispisuje adresu memorijske lokacije u kojoj je greška, ali *nastavlja* sa unosom.

Očitavanje PC registra se vrši procedurom *pc-1* i *pc-2*, tako da je u svakom trenutku moguće precizno odrediti područje aktivnosti programa-boraca.

Skeniranje memorije se vrši procedurom *mem-scan*, tako što se za dati njen interval varijablama *top* i *bottom*. Na taj način se može analizirati aktivnost programa tokom interpretacije.

Da bi se izbegao uticaj parazitskih podataka koji se zadržavaju u memoriji nakon završene interpretacije, neophodno je resetovati memoriju pre nego što unesemo programe u nju. Pozivanjem procedure *clear*, u svaku memorijsku lokaciju se postavlja O DAT O O.

Prava uloga MARS-a se vidi u interpretaciji programa za koju je zadužena procedura *run*, koja se izvršava preko specijalizovanih procedura, gde svaka odgovara određenoj instrukciji. Za vreme interpretacije neprekidno se očitava sadržaj PC registra.

Program se zaustavlja procedurom *quit*. Simulator je urađen tako da korektno izvršava instrukcije čak i ako je vrednost argumenta izvan intervala [0,9999], jer je uvek računao po nedeo 8000. Međutim, ukoliko instrukciji nije pridružen odgovarajući tip adresiranja, simulator prijavljuje grešku sa adresom memorijske lokacije u kojoj se ona nalazi. Pri tome se sadržaj PC registra ne menja, pa je nakon izvršene korekcije moguće nastaviti sa interpretacijom programa.

Time je problem simulatora rešen. Ostalo je još samo pisanje programa — boraca kao zanimljiviji deo. To smo, međutim, ostavili za sledeći broj.

Dragan Stanišić

DA LI ZNATE



koliko
jugoslovenskih
proizvođača može
da vam predstavi svoje dugoročne
vizije, da vam pokaže svoje razvojne tehnologije,
prototipove i planove, sopstvene proizvodne tehnologije,
da se pohvali celom paletom proizvoda koji imaju
sopstvenu istoriju u svojim prethodnicima i jasno
trasiran razvojni put?

Iskra Delta

proizvodnja računalniških sistemov in inženiring, p. o., 61000 Ljubljana, Parmova 41,
telefon: (061) 312-988

Buket grafičkih naredbi

Konstruktori „komodora 64“ nisu, kao što je poznato, svoj posao završili do kraja — izbacili su na tržište nedorađeni operativni sistem i bezjzik čije su mogućnosti znatno ispod potencijala inače prilično dobrog hardvera. Iako veoma podsticajne, ove okolnosti nikada nisu pokrenule programere u onoj mjeri u kojoj su to uspevale neke druge mašine. Jedan od najupornijih programera na „komodoru“ Viktor Cеровski pripremio je komplet veoma brzih grafičkih rutina. U prošlom broju objavili smo asemblerski listing za brzu grafiku, u ovom broju objavljujemo komplet novih naredbi za njenu komfornu upotrebu.

2010 :	
2050 :	„MODE m naredba
2070 nmode	: izračunaj vrednost argumenta mode naredbe
2080 :	jmp mode
2085 :	: i aktiviraj odgovarajući grafički mod
2090 :	„PLOT x, y naredba
2110 nplot	: uzmi prvi argument (x-koordin.) i smesti ga u xki i xkh
2120 :	jsr argum
	: uzmi drugi argument naredbe (y-koordinata tačke)
2130 :	sta yk
2140 :	jsr offrom
	: i smesti ga u promenljivu yk
	: isključi kernal radi pristupanja video memoriji
2150 :	jsr plot
2160 :	jmp onrom
2170 :	
2180 :	„SETCOL c1[, c2[, c3]]
2200 nsetcol	: uzmi prvi argument naredbe (kod prve boje)
2210 :	pha
2220 :	jsr argalt
2230 :	ora #80
	: postavi sedmi bit akumulatora na jedinicu (\$80 = %10000000)
2240 :	bcs push
	: ako ne postoji drugi argument, sadržaj akumulatora je u redu
2250 :	txa
	: inače vrati pravu vrednost drugog argumenta naredbe
2260 push	: kod treće boje sačuvaj za kasnije
2270 :	jsr argalt
2280 :	ora #80
2290 :	bcs okay
2300 :	txa
	: inače vrati stvarnu vrednost trećeg argumenta naredbe
2310 okay	: kod treće boje u y-registar
2320 :	pla
2330 :	tax
2340 :	pla
2350 :	jmp setcol
2360 :	
2370 :	„PEN pencil[, atr[, flag]]
2390 npen	: uzmi prvi argument naredbe, to je tip olovke
2400 :	sta pen
	: i postavi ga u odgovarajući sistemsku promenljivu
2410 :	jsr argalt
2420 :	bcs reti
2430 :	tsr a
2440 :	lda pen
2450 :	bcc noccol
	: ako je drugi argument naredbe bio nula, skoži
2460 :	and #87
	: inače resetuj bit 7 akumulatora (omogućiti postavljanje atributa)
2470 :	bpl oko2
2480 noccol	: obavezno skoži na oko2
	: u slučaju da je collg=0 zabrani postavljanje atributa
2482 ok2	: postavi novu vrednos pen-a
2484 reti	: kraj posla, povratak u interpreter
2490 :	
1500 :	„COLOUR bkgnd[, border]
2520 ncolour	: uzmi prvi argument naredbe (u akumulator i x-registar)
2530 :	sta \$d021
	: i postavi registar video kontrolera za pozadinu
2540 :	jsr argalt
2550 :	bcs reti
2560 :	sta \$d020
	: ako ne postoji drugi argument završi, inače postavi registar video kontrolera za okvir ekrana
2570 :	rts

2580 :	
2990 :	Pomoćni potprogrami koji koriste nove naredbe
3000 argumhi	: poziv tri potprograma interpretera za
3010 argumil	: izračunavanje jednog argumenta tekuće bezjzik naredbe i
	: smeštanje njegove vrednosti (0.6535) u lokaciji 20 i 21
3020 :	jmp 47095
3025 :	
3030 :	
3040 offrom	: argumhi treba zvati za uzimanje prvog argumenta nove naredbe, a za ostale argumente treba zvati argumhi
	: isključi prekide (time se neznatno dobija u brzini)
3045 :	lda 1
	: resetovanjem drugog bita reg. 1 se postiče isključivanje
3050 :	and #8c
	: rom-a sa o.s.-om i rom-a sa bezjzik interpreterom, radi
3060 :	sta 1
	: pristupanje ram memoriji \$e000.\$ffff, koja je video memorija
3070 :	rts
3080 :	
3090 onrom	: isključivanje
3091 :	lda 1
3092 :	ora #2
3093 :	sta 1
3094 :	cli
3097 :	rts
3100 :	
3370 argum	: poziv dva potprograma interpretera za
3380 argumj	: preuzimanje argumenta (0.255) iz bezjzika i njegovo
3390 :	txa
	: smeštanje u x-registar, zato se prebacuje i u akumulator
3395 :	clc
	: resetuj „carry“ — to je signal da je argument uzet
3400 :	rts
3410 :	
3412 :	Uzima argument, ako ovaj postoji, i smešta ga u ac i xr.
3414 :	lzax — c=1 nema argumenta, — c=0 argument u ac (i xr)
3420 argalt	: proverda da li je tekući bajt iz bezjzika programa
3430 :	lda (tempbyt), y
3440 :	cmp #"
3450 :	beq argum
3460 :	sec
	: ako jeste, ima još argumenta
3470 :	rts
	: inače postavi „carry“, to je signal da nema argumenta
3480 :	
3500 :	Kontrola sintakse. Prepoznata naredba biva odmah i izvršena.
3520 igone	: vektor za upućivanje na izvršavanje naredbi
3525 tempbyt	: pokazivač na tekući bajt bezjzik programa
3530 usrcmd	: neiskorišćena dva bajta
3535 lerror	: vektor za upućivanje na ispisivanje poruka u greškama
3540 getchr	: vraća u ac tekući bajt bezjzik programa
3545 :	
3550 init	: prepili niže bajt vektora igone u usrcmd
3560 :	sta usrcmd
3570 :	lda igone+1
	: a zatim i više bajt.
	: tako je stara vrednost ovog vektora sačuvana u usrcmd
3580 :	sta usrcmd+1
3590 :	lda #<check
3600 :	sta igone
3610 :	lda #<check
3620 :	sta igone+1
3621 :	
3622 :	lda #<errsub
3623 :	sta lerror
	: upiši adresu rutine
	: errsub u vektor lerror...

```

3624 : lda #>errsub
3625 : sta ierror+1
3630 : rts ; ... i inicijalizacija je završena
3635 :
3640 : Kontrola pojavljivanja novih naredbi
3650 check jr getchr ; uzmi tekući bajt iz bezjick programa
3660 : asi a ; prebacni bit 7 ovog bajta u „carry“, i provjeri da li
3670 : bcc maybe ; je setovan, ako nije radi se o običnom karakteru,
3673 back lda tempbyt ; inače je kod neke bezjick naredbe
3674 : bne dcr ; pa urnajni sadržaj tempbyt za jedan ...
3675 : dec tempbyt+1 ; (time se vraća njegova
3676 dcr tempbyt ; stara vrednost)
3680 : jmp (usrcmd) ; ... i nastavi rad u drugoj interpreterskoj rutini
3680 :
3690 :
3700 maybe ldx #0 ; postavi brojač ispitanih
3710 : stx temp ; novih naredbi na nulu
3720 nxt ldy #0 ; počni od prvog znaka tekućeg bezjick teksta
3740 comp lda (tempbyt, y ; uporedi karakter iz bezjick teksta sa
3750 : cmp namecmd, x ; karakterom jedne od novih naredbi
3760 : bne ntxcmd ; ako nije jednak, pređi na sledeću novu naredbu
3770 : lny ; inače pređi na sledeći znak bezjick teksta
3780 : lnx ; i na sledeći znak nove naredbe
3790 : lda namecmd,x ; ako je taj znak
3800 : cmp #", " ; zarez, naredba je prepoznata
3810 : bne comp ; inače nastavi sa upoređivanjem
3820 :
3830 : tya ; kako je naredba prepoznata,
3840 : clic ; prebacni y-reg u ac i pripremi „carry“ za sabiranje
3850 : adc tempbyt ; pa dodaj ac na trenutnu vrednost tempbyt-a
3860 : sta tempbyt ; i time pomeri tempbyt da pokazuje
3870 : bcc jump ; na prvi karakter iza nove naredbe
3880 : inc tempbyt+1 ; <modifikacija višeg bajta tempbyt ako postoji prenos
3890 :
3900 jump lda temp ; uzmi u akumulator redni broj prepoznate instrukcije — 1
3910 : asi a ; a zatim i viši bajt, tako da je sada adresa u jar
3920 : tax ; rezultat prebacni u x-registar
3930 : lda adrcmd,x ; i iz tabele adresa uzmi niži bajt adrese nove naredbe
3940 : sta mdf +1 ; modifikuj niži bajt jar instrukcije
3950 : lda adrcmd+1,x ; a zatim i viši bajt, tako da je sada adresa u jar
3960 : sta mdf +2 ; instrukciji jednaka izvršnoj adresi nove naredbe

```

```

3970 mdf jr 0000 ; izvrši novu naredbu
3975 : jmp $7aa ; naredba izvršena, skok u interpreter
3980 :
3980 :
3990 ntxcmd cmp #", " ; naredba nije prepoznata, traži u tabeli
4000 : beq ok ; zarez koji označava kraj naziva te naredbe
4010 : lnx ok ; pomeri pokazivač na sledeći znak u tabeli novih naredbi
4020 : lda namecmd,x ; uzmi sledeći znak
4030 : bne ntxcmd ; i nastavi da traži kraj imena naredbe
4040 :
4050 ok lnx ; pomeri pokazivač na početak imena sledeće naredbe
4055 : inc temp ; uvećaj brojač ispitanih naredbi
4060 : lda temp ; uzmi njegovu vrednost i provjeri da li
4070 : cmp #naredbi ; su sve naredbe upoređene sa bezjick tekstem
4080 : bne ntx ; ako nisu, nastavi upoređivanje
4090 : jmp back ; inače nije prepoznata nova naredba
4100 :
4110 : Rutine koja se izvršava pre ispisa poruke o grešci
4125 errsub stx temp ; kod greške u x-registru sačuvaj
4127 : ldx #0 ; inicijalizuj mode 0, tj.
4128 : jr mode ; standardnu tekst grafiku
4130 : ldx temp ; vrati kod greške u x-registar
4131 : jmp $8251 ; i skoči u standardnu bezjick rutinu za ispis greške
4140 :
4200 naredbi=5
4210 namecmd,asc „plot, mode, setcol,“
4220 : „asc „pen, colour,“
4230 :
4300 adrcmd „wor nplot, nmode, nsetcol
4310 : „wor npen, ncolour
4320 :
4330 :
4331 : Primer korišćenja novih naredbi
10000 colour 5,3
10020 setcol 0,5;mode 1;pen 1,1
10030 : X=160-rnd(0);y=100-rnd(0)
10040 : setcol 0;if y<100-x then: setcol 1
10100 : plot x, y ; plot 320-c-y
10110 : plot x, 200-y; plot 320-x, 200-y
10120 : get k$
10130 if k$="" then 10030
10140 end
ready.

```

Potprogram SETCOL je namenjen postavljanju boja kojima se crta. Na ulazu A, X i Y registri treba da sadrže respektivno boju 1, 2 i 3 u multi-kolor modu, odnosno INK i PAPER boju u visokoj rezoluciji (sadržaj Y reg. tada nema značaja). Istomerna bezjick naredba ima sledeću sintaksu:

SETCOL C1 [C2][C3]

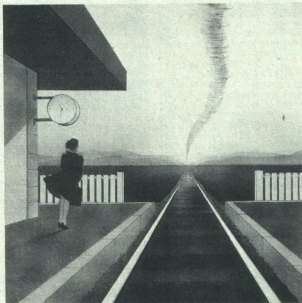
Boju broj 0 (koja je zajednička za sve tačke) u multi-kolor modu, odnosno boju pozadine u tekst modu određujemo direktnom promenom hardverskog registra sa adrese \$DO21 (53281). U bezjicku se ova boja obično menja POKÉ naredbom, a sad je to moguće postići korišćenjem COLOUR naredbe, kojom se može postaviti i boja okvira ekrana. Sintaksa ove naredbe je:

COLOUR C0 [border]

„Olovku“ kojom se vrši crtanje postavljamo iz mašinskog jezika direktnom promenom sistemске promenljive PEN, dok se iz bezjicka to radi naredbom PEN sa sintaksom:

PEN pencil[,cf]

Prvi argument određuje tip „olovke“, a drugi da li će se prilikom crtanja postavljati i atributi za svaku nacrtanu tačku (1-da, 0-ne).



Naredba PLOT x,y, naravno, vrši crtanje tačke, a naredba MODE m aktivira razne grafičke modove.

Pošto je osnovni cilj ovog teksta da predstavi mašinske grafičke programe, potprogram koji prepoz-

naje nove bezjick naredbe urađen je tako da bude što kraći, pa se ne vrši tokenizacija (što bi ubrzalo izvođenje samih naredbi), nego se svaki put po nalasku na neki deo teksta bezjicka koji već nije tokenizovan vrši pretraživanje čitave tabele sa nazivima svih novih naredbi. Pored ove rutine, bilo je potrebno i napisati posebnu rutinu koja će se izvršavati svaki put pre ispisa poruke interpretera o nekoj grešci u bezjick programu i njen jedini cilj je da inicijalizuje grafiku niske rezolucije, kako bi korisnik mogao da pročita samu poruku (naime, u višim grafičkim modovima operativni sistem i dalje ispisiše tekst kao da je aktivna najniža grafička rezolucija).

Asemblerski listing iz prošlog i ovog broja treba otkucati i assemblirati, a zatim izvršiti SAMO JEDNOU inicijalizaciju programa sa SYS 49152. Posle toga bezjick će prepoznati i sve ove nove naredbe (pritiskom na STOP+RESTORE najbolje je ne koristiti posle inicijalizacije). Iza asemblerskog listinga dat je primer kako koristiti pojedine potprograme (obratite pažnju na neodgovorne dve tačke iz THEN pre nove naredbe u 10040 liniji).

Viktor Cerovski

Najbrža elipsa u Jugi

Čitaoci „Računara“ i dalje se takmiče u crtanju krugova. Na redu je ponovo „spektrum“, a bojno polje je sada prošireno i na elipse. Radi se o veoma brzom programu, po algoritmu sličnom onom iz „Računara 34“. Tajna brzine je, između ostalog, i u posebnoj rutini za izračunavanje adresa u video memoriji. Trka sa vremenom se nastavlja ...

Elipsa čija je velika poluosa a , a mala b , može se, u pravougloj koordinatnoj sistemu, predstaviti jednačinom:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Pri čemu je koordinatni početak u centru elipse.

Možemo istu stvar zapisati i ovako:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$$

a funkciju sa leve strane znaka jednakosti ćemo kratko označiti sa $f(x,y)$. Za svako fiksirano x između $-a$ i a , tačka može ležati na elipsi, unutar nje, ili van nje, u zavisnosti od vrednosti y . Pri tome je:

$f(x,y) = 0$, za tačke na elipsi
 $f(x,y) < 0$, za tačke unutar elipse
 $f(x,y) > 0$, za tačke van elipse

Kako radj algoritam

Zbog simetrije u odnosu na ose koordinatnog sistema, nije potrebno generisati sve tačke elipse, već samo vrednosti u jednom kvadrantu. Ako, recimo, izračunamo tačku (x,y) , onda automatski možemo crtati i tačke: $(-x,y)$, $(x,-y)$ i $(-x,-y)$

Svaku tačku elipse u prvom kvadrantu računamo na osnovu koordinata prethodne tačke na sledeći način:

- jedan korak iznad prethodne tačke, ako je $f(x,y) < 0$
- jedan korak levo od prethodne tačke, ako je $f(x,y) \geq 0$

Pri tome, prva tačke elipse će imati koordinate $(a,0)$. Program se završava kada x dođe do nule.

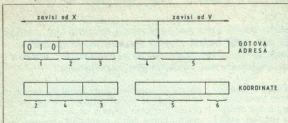
Korišćenjem samo dva smera kretanja prilično se gubi, ionako teško ostvariva glatkoća krive. Međutim, nije nikakav problem uvesti i dijagonalan korak „gore levo“, na sledeći način:

Prvo se ispita da li je za prethodnu tačku $f(x,y) < 0$. Ako jeste, pomeranje se vrši korak navise i odmah se ispituje da li je sada možda $f(x,y) \geq 0$. Ako jeste, obavlja se i korak levo.

ORG 4000	AND 7	LD (DY),HL	LD H,A	ADD HL,DE
ELIPSE LD A,E	LD C,128	LD (DY+2),A	JR H,KRAJ	NUL: RR H
LD (Y1),A	AND A	LD A,(Y1)	INC H	RR L
LD (Y2),A	JR Z,DALJE	DEC A	KRAJ: LD (X2),HL	RRA
LD A,D	RRR C	LD (Y1),A	LD HL,(X1)	DJNZ MNO
SUB B	DEC A	LD A,(Y2)	LD A,H	RET
LD E,D	JR LOOP	INC A	AND 7	DD_HL: LD B,B
LD D,A	JR LD,B,C	LD (Y2),A	JR NZ,UCAI	LD E,D
CALL RAC	LD HL,0	RLC C	LD A,H	XOR A
LD (X2),HL	LD (F1+1),HL	RRR B	LD H,A	MNO: RR E
LD A,E	XOR A	LD A,(F1+2)	LD A,L	JR NC,NUL1
LD D,A	LD (F1),A	AND 128	SUB 32	ADD A,D
CALL RAC	LD HL,(X1)	JR NZ,PLOT	LD L,A	NUL: RRA L
LD (X1),HL	LD A,(Y1)	LD A,(Z1)	JR NC,KRAJII	DJNZ MNO1
LD A,B	SRL A	DEC A	LD A,H	LD H,A
LD (Z1),A	SRL A	LD (Z1),A	SUB 8	RET
PUSH BC	OR L	LD HL,(F1)	JR H,KRAJIII	RAC: LD A,D
LD D,B	LD L,A	EX DE,HL	LD H,A	RRA
CALL DD_HL	LD A,(HL)	ADD HL,DX	JR KRAJIII	RCA
LD (DY),HL	OR C	LD HL,(F1+2)	KRAJIE: DEC H	RCA
XOR A	LD (H1),A	LD HL,(F1+2)	KRAJIE: LD (X1),HL	AND 24
LD (DY+2),A	LD A,L	LD A,(DX+2)	LD HL,(X1)	OR 64
ADD HL,HL	AND 31	ADC A,L	LD A,(Y2)	LD H,A
LD (DDY),HL	OR L	LD (F1+2),A	SRL A	LD A,D
LD D,C	LD L,A	LD HL,(DDX)	SRL A	AND 7
CALL DD_HL	LD A,(HL)	EX DE,HL	OR L	OR H
ADD HL,HL	OR C	LD HL,(DX)	LD L,A	LD H,A
LD (DDX),HL	LD (H1),A	ADD HL,DE	LD A,(HL)	LD A,D
SRL H	LD A,(F1+2)	LD A,(DX+2)	OR B	RLCA
RR L	AND 128	ADC A,0	LD (H1),A	RLCA
POP BC	JR Z,LEVO	LD (DX),HL	LD A,L	AND 224
LD A,B	LD HL,(F1)	LD (DX+2),A	AND 31	LD L,A
ADD A,A	EX DE,HL	LD HL,(X2)	LD HL,(X2)	RET
DEC A	LD HL,(DY)	A,H	OR L	X1: EQU 3000
EX DE,HL	LD HL,(DL)	CPL	LD L,A	X2: EQU 3002
LD C,A	ADD HL,DE	AND 7	LD A,(HL)	Y1: EQU 3004
CALL CDE	LD (F1),HL	JR NZ,UCA	OR B	Y2: EQU 3005
LD DE,0	LD A,(DY+2)	LD A,H	LD (H1),A	DDY: EQU 3006
EX DE,HL	ADC A,L	AND 248	LD A,(Z1)	DDX: EQU 3008
CPL	LD (F1+2),A	LD H,A	XOR 255	Z1: EQU 3010
INC A	LD HL,(DDY)	LD A,L	JR NZ,E_MAIN	DY: EQU 3011
CF	EX DE,HL	ADD A,32	RET	DX: EQU 3014
SBC HL,DE	LD HL,(DY)	LD L,A	CDE: LD HL,0	FI: EQU 3017
LD (DX+1),HL	ADD HL,DE	JR NC,KRAJ	LD B,B	
LD (DX),A	LD A,(DY+2)	LD A,H	MNO: RR C	
LD A,(Y1)	ADC A,0	ADD A,8	JR NC,NUL	

O video memoriji

Ako u 16-bitnom registru imamo adresu nekog bajta u video memoriji, a u nekome drugom (takođe 16-bitnom) registru koordinate tačke (8 bita X i 8 bita Y koordinata), onda pojedini bitovi predstavljaju:



1) osnova za video memoriju (16384)

- 2) oznaka za trećinu: 00 — prva
01 — druga
10 — treća
11 — prekoračenje

- 3) rastojanje od početka karaktera u kome se red nalazi,
4) broj karaktera u kome se red nalazi (nulti je na početku trećine),
5) rastojanje tačke od početka reda (u bajtovima)
6) položaj bita (koji odgovara tački) u bajtu video memorije

Umesto da svaki put „sklapamo“ od koordinata konačnu adresu, možemo na osnovu prethodne izračunati sledeću (ako je razlika jedan red). Zbog specifičnosti ekrana, razlikujemo sledeće slučajeve: 1) oba reda su u istom krakteru (najčešći slučaj), 2) jedan je na kraju jednog, a drugi na početku sledećeg karaktera, ali su oba karaktera u istoj trećini (8 puta ređi slučaj), 3) jedan je na kraju jedne, a drugi na početku sledeće trećine (takođe nisu u istom karakteru) (64 puta ređi slučaj).

U prvom slučaju „rastojanje od početka karaktera“ prethodnog reda je manje od 7 i adresa sledećeg reda se dobija samo povećanjem ove vrednosti za 1 (INC H). U drugom i trećem slučaju se red nalazi na dnu karaktera (rastojanje od početka karaktera je 7), pa se sledeći red nalazi na početku sledećeg karaktera. Zato treba postaviti „rastojanje od početka karaktera“ na 0 bez remećenja ostalih bitova, što se postiže instrukcijom AND 248. Pored toga, treba povećati „broj karaktera u kome se red nalazi“. S obzirom da su za to odgovorna najviša 3 bita nižeg bajta adrese, ovo povećanje ćemo postići instrukcijom ADD L,32. Takođe, činjenica da povećamo najviše bitove u registru nam omogućava da ne vodimo računa da li je prethodni karakter bio poslednji u trećini (najviša 3 bita su 1), jer ako jeste, prenos prilikom sabiranja jedinice odlazi u CF (ne remeti ostale bitove), a najviše 3 bita se postavljaju na 0 (automatski). Sada, ako je CF=1, moramo povećati broj trećina za 1 (ADD H,8), a ako CF=0, ne vršimo korekcije i završavamo rad. Na ovaj način je uštedeno malo memorije, ali dosta vremena.

Kako ubrzati račun

Vrednost funkcije $f(x,y)$ za svaku novu tačku elipse mogli bismo računati po formuli, ali bi to bilo čisto rasipanje vremena. Mnogo je bolje koristiti priraštaje funkcije, koje nije teško naći:

$$\text{korak levo: } f(x-1,y) = \frac{(x-1)^2}{a^2} + \frac{y^2}{b^2} - 1 = f(x,y) - \frac{2x-1}{a^2}$$

$$\text{korak gore: } f(x,y+1) = \frac{x^2}{a^2} + \frac{(y+1)^2}{b^2} - 1 = f(x,y) + \frac{2y+1}{b^2}$$

Dakle, svaka nova vrednost funkcije $f(x,y)$ dobija se iz prethodne, dodavanjem odgovarajućeg priraštaja.

Međutim, ni priraštaji, a ni sama funkcija nisu celii brojevi, što baš nije pogodno za mašinsko programiranje. Zato ćemo mi koristiti malo drugačiju funkciju $f(x,y)$, polazeći od jednačine elipse u obliku:

$$x^2b^2 + y^2a^2 - a^2b^2 = 0$$

Funkcija na levoj strani je sada celobrojna i označićemo je sa $f(x,y)$. I odgovarajući priraštaji će biti celobrojni:

$$f(x-1,y) = f(x,y) - (2x-1)b^2 = f(x,y) + DFX$$

$$f(x,y+1) = f(x,y) + (2y+1)a^2 = f(x,y) + DFY$$

Označili smo sa DFX priraštaj funkcije $f(x,y)$ kada se x smanji za jedinicu, a sa DFY priraštaj kada se y poveća za jedinicu. Očigledno, DFX zavisi od x , a DFY od y i moraju se računati za svaku tačku elipse. Međutim, račun se ne mora izvoditi po formuli. Nije teško pokazati da pri smanjenju x za jedinicu, DFX priraste za $2b^2$:

$$DF(x-1) = -(2(x-1)-1)b^2 = -(2x-1)b^2 + 2b^2 = DFX + 2b^2$$

Slično je:

$$DF(y+1) = (2(y+1)+1)a^2 = DFY + 2a^2$$

Prema tome, kompletan algoritam bi bio sledeći:

Vrednost $f(x,y)$ za svaku novu tačku elipse računamo na osnovu prethodne, dodavanjem odgovarajućeg priraštaja. Sam priraštaj se, opet, računa preko prethodnog, dodavanjem konstante. Početne vrednosti su:

$$f(x,y) = f(a,0) = 0$$

$$DFX = -(2a-1)b^2$$

$$DFY = a^2$$

Algoritam

```
x=a,y=0
DY=a2,DX=-(b2(2a-1))
F=0
repeat
  PLOT (x,y): PLOT (-x,y)
  IF f<0 THEN v=y+1; F=F+DY, DY=DY+2a2
  IF F>0 THEN x=x-1; F=F+DX, DX=DX+2b2
  PLOT (x,-y): PLOT (-x,-y)
until x<0
```

Kako radi program

Program se sastoji iz dva dela (inicijalni i glavni) i tri potprograma koje koristi inicijalni deo (RAC, DD_HL, CDE). Potprogram RAC računa adresu početka rada koji je zadat na ulazu preko registra D, potprogram DD_HL računa D² i rezultat smešta u HL, dok potprogram CDE množi C-D i rezultat smešta u HL.

Program se pridržava algoritma, ali ima i nekoliko napomena:

1) s obzirom na „spektrumov“ ekran, uveden je drugi koordinatni sistem: X je rastojanje od gornje ivice ekrana i raste što je tačka niže, dok je Y rastojanje od početka reda. Zato je „a“ u intervalu (0,96), a „b“ u (0,128). Što se korisnika tiče, ako ni odgovara ovakvo obeležavanje, može samo zameniti nazive „a“ i „b“, „X“ i „Y“ na ulazima, čime se dobija utisak „normalnog“ koordinatnog sistema.

2) Umesto da za svako novo (X,Y) računamo ponovo koordinate i priprema masku, program u promenljivima X1 i X2 čuva adrese početka redova (početne vrednosti računata pomoću potprograma RAC), u kojima su po 2 tačke, i na osnovu njih računata adresu sledećeg (ili prethodnog) reda, dok u promenljivima Y1 i Y2 čuva rastojanje tačkaka od početka reda. Kako se registar BC ne koristi, zgodno je poslužiti za čuvanje masaka, koje su iste za po dve tačke (sa istim X koordinatama). Konačna adresa se dobija dodavanjem na X1 (ili X2) vrednosti Y1/8 (ili Y2/8) (jer je to rastojanje od početka reda u bajtovima). Sada se vrednost sa konačne adrese OR-uje sa maskom u B (ili C) i nova vrednost smešta nazad u video memoriju. Tako, umesto (kao u algoritmu) da izvršava $X=X-1$, program samo izračunava adresu sledećeg reda od X2 i prethodnog od X1 i nove vrednosti ponovo čuva, a umesto da vrši $Y=Y+1$, samo povećava Y1 i smanjuje Y2 (za 1) i rotira maske u registrima B i C udesno, odnosno ulivo. Zbog toga deo programa označen kao PLOT samo dovršava računane adrese i jednom OR instrukcijom osvetljava odgovarajuću tačku, čime je uštedeno veoma mnogo vremena.

3) Poseban problem je izvođenje aritmetičkih operacija: Nije teško videti da su $f(x,y)$, DFX i DFY brojevi koji mogu preći 16 bita. Zbog veličine „spektrumovog“ ekrana, oni su manji od 24 bita. Zato svako sabiranje mora da se vrši na 3 bajta. U programu se vidi zašto je veoma zgodno sve računati na osnovu prethodne tačke (samo se u inicijalnom delu množi i kvadrira, dok glavni program samo sabira pripremljene vrednosti na DFX i DFY, a ove opet dodaje na $f(x,y)$)

Igor Čadež

Okružio pa na čoš

Poslednjih nekoliko meseci, uz skromnu crtno-ješenju pauzu, bavili smo se osnovama računarske grafike na malim računarima. Naš mali crni računar i njegov još crniji bežik doveli su nas na kraj ove edukativno-demonstracione serije o osnovama računarske grafike. Ovoj temi ćemo se, međutim, vraćati i ubuduće, ali sa moćnijim hardverom i softverom i, samim tim, ubedljivijim rezultatima.

Ko radi taj i greši, a ko greši — mora da se pospe pepelom, što ćemo i ovom prilikom učiniti. Umesto operatora stepenovanja, u programu stoji ponosito slovo „C“, za šta je kriv tvorac YUSCII standarda, koji je jedan od fundamentalnih operatora bežika zamenio našim folklornim znakom. Drugo i važnije je da su komentari u REM linijama sasvim obrnuti i da ih tako morate i čitati. Ako piše da se naredne formule koriste za preračunavanje sfernih u cilindrične koordinate, možete biti sigurni da se preračunavaju cilindrične u sferne. Kriv je, naravno, tekst procesor koji nije prepoznao i ispravio takvu grešku. Program je, inače, u redu i daje predviđene rezultate.

Sa dva na tri

Pokušajmo da utvrdimo neke sličnosti i razlike između ravnanskih i prostorne grafike na osnovu prethodnih programa i tekstova. Prva razlika sastoji se u tome što je ekran televizora ili monitora u obliku ravni (plohe), pa nisu potrebne nikakve prepravke dvodimenzionalne slike da bi bila prikazana. Dvojnogi je samo naznačiti koji deo XOY ravni vidimo na ekranu (ili popularnije —

otvoriti prozor). Treća dimenzija stvara teškoće, pa je potrebno primeniti transformacione izraze koji svode sliku na dve dimenzije. Pri tome što više treba sačuvati informaciju o trodimenzionalnosti.

Druga razlika je drastično povećanje broja podataka koje treba saopštiti računaru da bi se dobila slika. Za svaku tačku koju treba nacrtati potrebna je još jedna koordinata. Povećava se i broj karakterističnih tačaka trodimenzionalnih objekata u odnosu na dvodimenzionalne, a raste i raznovrsnost položaja i odnosa objekata u prostoru.

Kada sve to imamo u vidu, postaje besmislen zahtev da se za neke složene slike navode koordinate svih karakterističnih tačaka, postojanje i karakter njihovih veza. Moramo da tražimo načine za skratičavanje ovog postupka.

Moguća rešenja su slična kao i kod ravnanske grafike. Prvo je ukupnjavanje osnovnih objekata. Već smo primetili da takva nije najprezentativniji prostorni objekat, pa možemo definisati i neke druge standardne trodimenzionalne objekte. Njih mora biti dovoljno malo da ne stvaraju zbrku i dovoljno mnogo da je moguće

njihovom kombinacijom dobiti što raznovrsnije složene oblike. Crtanje svakog od tih oblika možemo poveriti posebnom potprogramu koze ćemo preneti neophodne podatke. Ti podaci prenose se u obliku vektora.

Kada smo u ravnanskoj grafici crtali pravougao, navodili smo četiri podatka. Prva dva podatka čine vektor položaja pravougaonika i pokazuju koliko je udaljen od koordinatnog početka mereno duž koordinatnih osa. Tačnije je reći da je to udaljenost njegove referentne tačke (to je ona tačka prema kojoj se ravnaju sve ostale tačke pravougaonika). Druga dva parametra čine vektor uvećanja objekta i govore o tome koliko je pravougao „rastegnut“ po X ili Y osi. Ako su sve komponente ovog vektora jednake, dobićemo takozvani normalizovani objekat.

Posmatrajmo sada kvadar kao trodimenzionalnog rodaka pravougaonika. Potreban je trodimenzionalni vektor položaja koji pokazuje udaljenje referentne tačke kvadra od ishodišta koordinatnog početka, mereno duž sve tri ose i vektor uvećanja objekta po X, Y, Z osi koji nam, u stvari, daje veličinu stranica kvadra. Tako možemo iz osnovnog

1	REM *****				
	* RACUNARI 36 *	110		LET y=y+dy	
	* EKRANSKI PRIKAZ *	120	REM *** POKROJ ***	NEXT j	NEXT t
	* z= f(x,y) *		*** PLOVINA ***	LET t=t+dx	LET t=t+0
	* V. Gasic 1988 *		*** LOPTE ***	END PROC	FOR t=1 TO m
					LET c=cos(t),s=sin(t)
9	10 ON ERROR	130	LET k=0,=10	9180	DEF PROC crt1 REF b()
	RETURN		LET as="50R (1-x*x/25-y*y/25)"	9185	DEF PROC crt2 REF b()
20	30 REM *** KOCKICA ***		CLS		CLS
	*** NA GRADELI ***		LOCAL i,j,k		LOCAL i,j,k
40	LET u=10,q=21,h=21		FOR i=1 TO yig-1		FOR i=1 TO xig-1
	LET as="5*(h-2)*c(x*(y-2)*(y-2))"	140	FOR j=1 TO xig-1		FOR j=1 TO xig-1
	gr13d as,u,u,u,u,g,h	150	REM *** HIPERBOLICNI ***		PLOT b(x=ixig,j),b(t=iyig,2)
	INIT 100,10,100		*** PARABOLICNI ***		DRAW TO b(x=ixig+1),b(t=iyig+2),
	per3d a(i),b(i)	160	LET k=0,=50		DRAW TO b(x=i+1),b(t=i+1),
	fnctd b(i)		LET as="h*x/5-y*y/5"		END PROC
	LINEF		CLS		
	cr1 b(i)		CLS		
50	60 REM *** POZNATA ***		DRAW TO b(x=ixig+1),b(t=iyig+2),		
	*** "NE TALASAJ" ***		b(t=iyig+2)		
	*** FUNKCIJA ***		DRAW TO b(x=i+1),b(t=i+1),		
70	LET u=15,q=25,h=25	999	STOP		
	LET as="20*COSE(SQR(x*x+y*y))*EXP(-SQR(x*x+y*y))*25"				
	gr13d as,u,u,u,u,g,h	9100	REM *** FUNKCIJE ***		
	INIT 100,10,100		*** DVA ARGUMENTA ***		
	per3d a(i),b(i)	9101	DEF PROC gr13d as,u,g,x1,y0,		
	fnctd b(i)		v1,m,n		
	LINEF		DEFAULT y0=0,y1=x1,=10,		
	cr1 b(i)		n=1		
80	90 REM *** PLOVINA ***		LOCAL i,j,dx,dy,s		
	*** LOPTE ***		LET m=(n-2)+10*(m(2),n		
100	LET g=11,h=11		m=(n-2)+10*(m(2),n		
	LET as="50R (1-x*x/25-y*y/25)"		DIM a(m,n,3)		
	gr13d as,-5,-5,-5,g,h		LET dx=(x1-0)/n+1,dy=(y1-y0)/(m-1)		
	INIT 10,30,100		LET x=0,xig=m,yig=n		
	per3d a(i),b(i)		FOR i=0 TO xig-1		
	fnctd b(i)		LET y=0		
	LINEF		FOR j=1 TO m		
	cr1 b(i)		LET as="5*(m-2)*c(x*(y-2)*(y-2))"		
			gr13d as,u,u,u,u,g,h		
			INIT 100,10,100		
			per3d a(i),b(i)		
			fnctd b(i)		
			LINEF		
			cr1 b(i)		

objekta (normalizovane kockice čija je stranica 1) dobiti različite razmere kvadra (san- duke, stubove, grede ...).

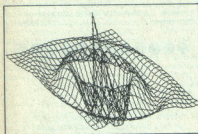
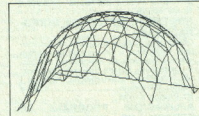
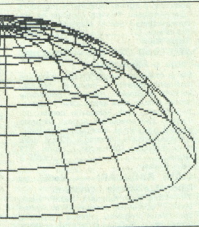
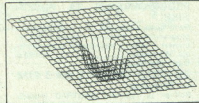
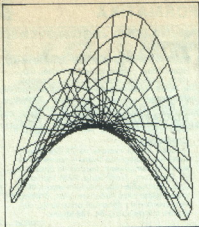
Pored ova dva osnovna vektora, možemo dodati i druge, kojima ćemo bliže opisati potrebni objekat.

Alkoholizam i nadrealizam

Bilo bi dobro da objekat može i da se „nagne“, a ne samo da stoji uspravno, pa možemo uvesti i vektor rotacije. Nismo ga uvodili pri crtanju pravougaonika, ali je njegovo dejstvo pokazano pri crtanju elipse u ravni. To je ugao nagiba elipse. I taj vektor bi mogao da se sveđe na jednu dimenziju manje (rotacija oko jedne od osa uvek može da se sveđe na dve rotacije oko druge dve ose), ali je u ovom obliku najpo- desniji.

Moguće je dodati i različite vektore deformacije u zavisnosti od potreba i predviđenih oblasti u kojima će se program primenjivati.

Standardne objekte je najlakše definisati kao žičane modele. Ali kao što je linija omeđena karakterističnim tačkama, a površina — karakterističnim linijama, tako je i zapremina ograničena karakterističnim površinama, pa se javlja potreba da se te površi naglase, jer žičani model ne daje uvek najjasniju predstavu o traženom predmetu. To se može učiniti tako što se površina kod



monohromne grafike ispuni nekim uzor- kom (teksturom), ili bojom određene nijan- se i osvetljenja da bi se dobila što vernija slika. Pri tome se fiktivni izvor svetla postavlja na proizvoljno mesto u koordinatnom sistemu. Zatim se za svaku elementarnu površinu računa intenzitet svetlosti koja se odbija prema posmatraču.

Elementarnu površinu u računarskom

smislu ne treba shvatiti kao beskonačno malu površinu, nego jednostavno kao malu ravnu površinu (što je moguće manju), razumno veliku da ne čekamo dugo na željeni rezultat (izgled objekta), a da taj izgled ne odstupa bitnije od stvarnog rezul- tata. Drugim rečima, da ne bude previše čouškast.

Često se elementarne površine koje čine neku komplikovaniju zakrivljenu površinu ostave srazmerno velike, pa se utisak zakri-

vljenosti pojačava metodama omešavanja (anti-aliasing).

Za intuitivne tipove

Drugi način da se dođe do trodimenzionalne slike je da se ona generiše pomoću nekog poznatog principa. Jedan od primera je postavljanje više preseka koje zatim sam računar međusobno spaja, ili zadavanje jednačine linije (generatriše) koja obrta- nom oko neke prave (direktrise) tvori prostorni objekat. Konačno, moguće je citavu sliku zadati u obliku matematički definisane krive zatvorene površi ili preseka više površi.

Mi ćemo se pozabaviti ovim poslednjim načinom generisanja trodimenzionalnih oblika. Ograničimo se na funkcije dva argumenta oblika $z=f(X, Y)$. Različitih oblicima mreže crtaćemo površi u blizini tačke koja nas interesuje. To će biti i naša poslednja praktična vežba iz ove oblasti. Pominjanje intuicije u ovom kontekstu se odnosi na veštinu iznalaženja interesantnih funkcija i tačaka u kojima se funkcija ponaša zanimljivo i pravi lepe slike. Naravno, i ova intuicija mora da ima podlogu u znanju matematike, jer se na rezultate prilično dugo čeka. Korisno je napomenuti da možete da menjate gustinu mreže, pa se manjom podelom možete napraviti probu pre glavnog iscrtavanja. Pored programa, u ovom broju morate prekućiti ili učitati delove programa iz prethodnog broja koji su neophodni za izvršavanje novih procedura.

Rezanci i kriške

Procedura **gr13d** crta u pravougaonu mrežu vrednosti funkcije. Vrednosti argumen- tana se menjaju kontinuirano i duž jedne linije je uvek jedan od argumenata konstan- tan. Potrebno je naznačiti izraz u obliku stringa, najmanju i najveću vrednost oba argumenta, a i gustinu mreže (koja nije obavezna).

Gustina mreže se navodi ko broj „reza- naca“ na koje će biti podeljena površina. Odgovarajuća procedura za iscrtavanje tako izdvojene površine je **cr1**, a prethodno treba pozvati i procedure za inicijalizaciju i transformaciju koordinata.

Druga procedura **Cl13d** preseca površinu koncentričnim kružnicama i radijalnim linijama (radijalna mreža) oko centralne tačke koja nas interesuje. Prednost ovakve mreže je što broj ispitanih tačaka raste sa približavanjem središnjoj tački. Potrebno je navesti izraz, koordinate središnje tačke, poluprečnik najveće kružnice i broj radijal- nih zraka. Nadam se da će vam dati primeri predstavljati dovoljan putokaz.

I gotovo?

Naš miljenik iz ranih osamdesetih ispuni je svoj zadatak i pokazao nam od računarske grafike tek toliko da shvatimo o čemu se radi. Nadamo se da su vam tekstovi i programi bili razumljivi i da su vas u nekoliko (problematičnih) slika uspešno uveli u ovu interesantnu oblast. Na novu stepenicu ćemo zakačiti kroz nekol- iko meseci, kad shvatite da računar koji sada prezirete košta tačno koliko i prvi „spektrumi“.

Voja Gašić

računar 36 • mart 1988. 57



RAČUNARI I INFORMATIKA

Tehnička knjiga

TEHNIČKA KNJIGA JE NAJVEĆI JUGOSLOVENSKI IZDAVAČ KNJIGA IZ OBLASTI RAČUNARA I INFORMATIKE.

Predstavljamo Vam deo ove popularne biblioteke.

1. **Grupa autora**
ŠTA MOŽE COMMODORE 64
Tastatura, raz su kasetofonom, programiranje palice za igru, BASIC I SIMON'S BASIC kroz primere, primena u raznim oblastima, programiranje funkcionalnih tastera, korisni dodaci. 196 strana 7.350 d
2. **Ian Stewart i Robin Jones**
COMMODORE 64 — Programiranje na lak način
Tastatura, promenišnje, pronalaznje i otklanjanje grešaka, PEEK I POKE, PET — grafika, sprajtovi, grafika u visokoj rezoluciji, datoteke. 236 strana 13.000 d
3. **Veljko Spasić i Dušan Veljković**
BASIC ZA MIKRORAČUNARE — COMMODORE 64
BASIC, korišćenje periferijskih uređaja, grafika i zvuk. 168 strana 3.700 d
4. **Andrew Bennett**
MAŠINSKE RUTINE ZA VAŠ COMMODORE 64
Proširenje BASIC-a za C-64 najznačajnijim mašinskim rutinama. 128 strana 9.700 d
5. **Dejan Ristanović**
MAŠINSKI PROGRAMIRANJE NA MIKROPROCESORIMA Z80 I 6502
Ovi osnovni mikroprocesori upotrebljavaju se u ZX 81, ZX Spectrum, Spectrum plus, Commodore 64, Commodore 128, Amstrad, Galaksiju, Oric itd. Iskoristite u potpunosti mogućnosti Vašeg računara prelaskom sa BASIC-a na mašinski jezik. 256 strana 16.000 d
6. **Mr Veselin Petrović i Zoran Mošorinski**
COMMODORE 128
C-128 i periferijski uređaji, tastatura, osnovne BASIC jezika, komande za rad sa disketnom jedinicom, sistemske naredbe (komande), grafika, sprajtovi, muzika, monitor, korisni dodaci. 190 strana 13.000 d
7. **Bob Steele i Jerry Wellington**
RAČUNARI I KOMUNIKACIJE
Knjiga je pisana na osnovu plana i programa za početni kurs informatike u Velikoj Britaniji tokom 1984/85. godine i uspešno se može koristiti za nastavu informatike u VIII razredu osnovne i I ili II razredu srednje škole. Tekst je izložen postupno i vrlo pregledno, a nakon svakog poglavlja dat je test za proveru uspešnosti usvajanja gradiva. 224 strana 14.050 d
8. **Grupa autora**
KUĆNI KOMPJUTERI — Algoritmi i programi za Spectrum i Commodore
Struktura algoritma i BASIC sa specifičnostima Spectrum-a i Commodore-a, primene računara u raznim matematičkim disciplinama, matematičko modeliranje i igre na računaru. 244 strana 2.700 d
9. **Dragan Majkić**
KOMPJUTERSKA GRAFIKA
Osnovo o računarskoj grafici, primena kompjuterske grafike u procesu projekovanja, organizacija crtanja i način učenja programa. 250 strana 16.000 d
10. **Clive Gifford**
AVANTURE ZA VAŠ ZX SPECTRUM — Listinzi Igar
Listinzi najinteresantnijih Igar, uputstva za igranje, osnove za pravilno spopetnih Igar slinog tipa. 116 strana 1.250 d
11. **Dr Mirčeta Danilović**
VIDEO-KOMPJUTERSKE IGRE
Podela i vrsta video-kompjuterskih Igar, sistemi video-kompjuterskih uređaja za njihovu realizaciju, programiranje video-
12. **Dr Dejan Stajić i Dragošlav Jovanović**
ODRŽAVANJE I OPRAVKA KUĆNIH RAČUNARA — Spectrum i Commodore
Najčešći uzroci kvarova na ZX Spectrumu, detekcija i dijagnoza kvarova, zamena neispravnih komponenti, hardverski dodaci za ZX Spectrum, proširivanja RAM memorije, opis i laptivanje rada mikroprocesora 6510, memorijaska mapa računara C-64, o kvarama memorije i njihovom otklanjanju, centroniks interfejs za C-64. 149 strana 3.350 d
13. **Phillp Crookall**
PROGRAMIRANJE ZA POČETNIKE
Namenjeno svima koji čine prve korake u programiranju. Osnovne naredbe BASIC-a, primeri i testovi, samostalno programiranje. 167 strana 10.000 d
14. **Garry Marshall**
AMSTRAD CPC 464 & 664 & 6128 — Priručnik
Softver za računare AMSTRAD (obrada teksta pomoću programa Amsword, baza podataka, programi za tabelarne proračune i East-Amscalc). Primene bazirane na hardveru (kasete i diskovi, štampači i ploteri). 120 strana 5.100 d
15. **Steve Webb**
AMSTRAD CPC 464 — Programiranje u Asembleru
Šta je mašinsko programiranje, upisivanje mašinskih naredbi u memoriju nekoliko korisnih mašinskih rutina — pomeranje (scroll) jednog reda teksta ulivo i udesno, zvuk lasera, zvuk eksplozije bombe itd. Dodaci: Z80 operacioni kodovi, ekranski modovi, program za dizajniranje karaktera, o nekim korisnim rutinama iz ROM-a, neke nove mašinske instrukcije i rutine. 112 strana 5.000 d
16. **John Graham**
LIČNI RAČUNARI — Vodič za izbor, korišćenje i primenu
Arhitektura ličnih računara, ulazni i izlazni uređaji, operativni sistem CP/M, razvoj aplikativnih programa, primena u poslovne svrhe, mreže, trendovi u ON LINE aplikacijama, izbor sistema. 270 strana 3.900 d
17. **Grupa autora**
NUMERIČKI METODI ZA MIKRO-RAČUNARE
Objašnjenja i listinzi programa za Commodore i Spectrum iz savremenih numeričkih metoda. Interpolacija, nelinearne i algebarske jednačine, sistemi linearnih i nelinearnih jednačina, karakteristične vrednosti i vektori. 188 strana 2.300 d
18. **Mr Dragan Pantić**
APLIKACIONI PROGRAMI ZA PERSONALNE RAČUNARE — IBM PC/AT/XT I APPLE II C
Tastatura IBM PC I APPLE II C, Procesiranje reči pomoću WORDSTAR-a i APPLEWORKS-a. Organizacija LOTUS 1-2-3 i njegove osnovne mogućnosti. Radne tabele, formiranje podataka, kopiranje i pomeranje, status tabele, upravljanje fajlovima, poziv za pomoć (HELP). 276 strana 9.700 d
19. **Mr Vojislav Milić**
IBM PC/AT/XT U 25 LEKCIJA
PC hardver, DOS — operativni sistem, rad
20. **Mr Veselin Petrović i Adem Jakupović**
LINIJSKI EDITOR ZA SISTEME EI — HONEYWELL
Knjiga detaljno obrađuje jedan od osnovnih softverskih paketa operativnog sistema računara H6 (ili DPS6) — linijski editor. Detaljno su opisana pravila za pisanje direktiva, korišćenje specijalnih simbola, mogućnosti adresiranja, postupci rada kao i sintaksa direktiva, uz brojne originalne primere koji ilustruju mogućnosti pomenutog paketa. 207 strana 6.150 d
21. **Dejan Ristanović**
OBRAĐA TEKSTA NA RAČUNARU
Obrada teksta na Word Star-u i Word Perfect-u za IBM PC, VISA Write, AMS-WORD za Amstrad 464/6128 itd. Upotreba Epson kompatibilnih štampača, priprema indeksa pojnova, povezivanje računara sa laserskim printerom, YU-set slova. 210 strana 14.000 d
22. **John Cunliffe**
LOGO — Programski jezik
Prvi put na našem jeziku — LOGO za Commodore, Atari, Spectrum. Listinzi programa, boja, muzika. 128 strana 2.250 d
23. **Boško Damjanović**
BASIC U NASTAVI MATEMATIKE
Knjiga je namenjena prvenstveno nastavnicima i profesorima osnovnih i srednjih škola, kao i učenicima — pre svega zbog svog metodičkog karaktera i velikog broja interesantnih zadataka iz raznih matematičkih disciplina. Svaki primer praćen je tekstuinalnim objašnjenjima, algoritmom i programom. 114 strana 5.400 d
24. **Boško Damjanović**
ZBIRKA ZADATAKA U BASIC-U (sa rešenjima)
Algoritmi i listinzi programa, elementarni zadaci, problematski zadaci, primene u raznim oblastima. 223 strana 5.600 d
25. **Dr Dušan Tošić i dr Vojislav Stokjović**
PROGRAMSKI JEZIK PASCAL — Zbirka rešenih zadataka iz programiranja
Uvod u metodološki prikazan proces kreiranja programa na principima strukturovanog programiranja. Zbirka sadrži veliki broj primera, od elementarnih do vrlo složenih. 252 strana 10.250 d



Uplate znak X preko rednog broja knjige koju poručujete. Porudbinu pošaljite na adresu: NIRO TEHNIČKA KNJIGA, Beograd, 7. julia 26. Isporuka odmah. Plaćanje pozuscem. Ove knjige možete nabaviti i u svim većim knjižarama.

1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25

Ime i prezime
Ulica i broj
Broj pošte Mesto

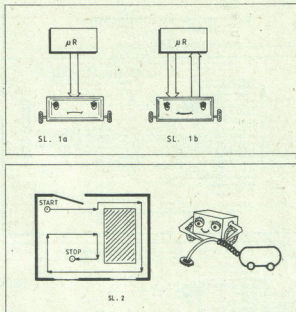
Put za periferiju

Robot, sam za sebe, predstavlja najobičniju gomilu gvožđa koja nije u stanju da uradi ništa suvislo. Računar je taj koji koordinira rad i pokreće delove robota, dajući tim radnjama neki smisao. Signali koje razmenjuju računar i robot prolaze kroz posebno dizajniran međusloj (interfejs), od čije strukture umnogome zavisi uspeh čitavog projekta.

Prvi nastavak serije o malim kućnim robotima (MKR) predstavlja teorijski uvod u komunikaciju kućnog računara sa periferijskim uređajima. To je i uvod u svojevrsnu „hardversku golgotu“, koja obsekuje svakog antuzijastu na tom polju rada. Zauzvrat, svaki pokret našeg budućeg robota priličnice nam ogromno zadovoljstvo.

Hardverska realizacija MKR podrazumeva kako poštovanje teorijskih principa projektovanja, tako i poznavanje strukture i namene standardnih integrisanih kola koja se ugrađuju u interfejs. Projektovanje hardvera mora se posvetiti velika pažnja, jer svaka greška može stvoriti nepremostive teškoće pri upravljanju robotom. Greške se, na žalost, ne mogu sasvim izbeći, ali ih makar možemo svesti na minimum.

Veoma važna stvar u projektovanju malog kućnog robota jeste dobro poznavanje tehnike interfejsa, dekodiranja adresa i, uopšte, komunikacije računara sa periferijom. Konstruktor u potpunosti mora shvatiti principe razmene informacija između računara i robota, jer jedino u tom slučaju je moguće iskoristiti sve potencijale čitavog sklopa.



ski analizirati karakter rada (zadataka) koje MKR treba da obavlja.

Prva koncepcija (jednosmerni prenos) predstavlja *programsko komandovanje* i podrazumeva da informacije (komande) potiču isključivo od računara i namenjene su periferiji, koja samo izvršava zadata naređenja. Povratna komunikacija od MKR ka računaru se ne podrazumeva (sl. 1a).

Uvijek koncept upravljanja pretpostavlja da su sve moguće situacije i radnje koje MKR treba da obavi poznate i definisane po jedinstvenom redosledu. To je, naprosto, niz aktivnosti koje slede jedna za drugom.

Druge koncepcija upravljanja podrazumeva protok informacija između računara i periferije u oba smera, prema utvrđenom protokolu (sl. 1b).

U ovom slučaju, upravljanje nije samo programirano komandovanje,

već podrazumeva i detektovanje informacija sa periferije i odlučivanje o daljnjim aktivnostima na osnovu primljenih informacija.

Informacije sa periferije daju računaru mogućnost detektovanja situacije (tzv. „uslovno analaženje“). Algoritam upravljanja dobija karakter strategije i predstavlja programsko odlučivanje o aktivnostima koje treba preduzeti na osnovu primljenih informacija sa periferije. Pošto situacija na terenu nije predviđena, na raspolaganju je više programskih izlaza (akcija). Za razliku od prvih koncepcija gde je izlaz uvek samo jedna, strogo definisana, sledeća akcija.

Suštinske razlike upravljanja dve navedene koncepcije ilustruje sledeći praktičan primer: Zamislimo MKR kao „pametni usisivač za prašinu“ (zamenja za „klasičnu“ čistačicu). Zadatak je usisati prašinu u nekoj prostoriji.

U prvoj koncepciji, program piše za tačno određenu prostoriju. Startovanje robota je uvek sa iste predviđene početne tačke. (sl. 2). Svi predmeti u prostoriji moraju biti programski predviđeni u zadatoj geometriji. MKR — usisivač uvek prelazi isti put, na isti način. Ako izmenimo bilo šta u prostoriji, „program se neće snadi“ i posao neće biti korektno obavljen.

Druge koncepcija podrazumeva senzore prepreka (npr. „paličice“, tj. najobičnije prekidače sa povratnom vezom ka računaru). Sada program MKR ne mora da pozna veličinu prostorije i raspored nameštaja. Kada naiđe na prepreku, robot programski menja pravac i nastavlja posao. MKR „pamti“ prethodni put i pozicije prepreka, da bi obišao i površine koje je propustio. Savim mu je svedjeno kakva je geometrija sobe i kakav je raspored nameštaja. Posle određenog vremena, čitav pod će biti usisan.

Bilo za koju koncepciju upravljanja da se opredelimo, izuzetnu pažnju moramo posvetiti projektovanju interfejsa.

Veza sa računarem

Šarolikost malih kućnih računara, različiti standardi i tehnologije, dikтираju opšti pristup tehnici interfejsa.

Nije nikakva tajna da su za potrebe MKR najopodgniji računari otvorene arhitekture, tipa „BBC“ ili „epi“. Konstruktori malih robota najčešće koriste „komodor 64“, „spektrum“, „ZX 81“, ili prave minimalne računarske konfiguracije oko nekog standardnog osmo-bitnog procesora. Pored navedenih računara, preporučujemo, naravno, i domaću „galaksiju“.

Male kućne robote baziramo na standardnim osmo-bitnim mikroprocesorima: 6502, 6080, 6802, 280. Svi oni su dobro opisani u literaturi, sa mnogo konkretnih primera, i omogućuju amaterima jednostavan izbor tipskog hardvera. Pri tom kućni računar najčešće koristimo samo kao „razvojni sistem“ do momenta kad ga zamenjujemo identičnom minimalnom konfiguracijom procesora sa epromom.

Ako ste primorani da koristite IC kola različitih proizvođača, konsultujte kataloge. Proizvođači rado daju alternacije i krosreferencije, nastojeći da olakšaju upotrebu svog proizvoda i ujedno istisnu konkurenta. U nedostatku kataloga, možete potražiti savet od iskusnijih kolega, ili od redakcije „Računara“.

Interfejs

Računar i mali kućni robot kao njegova periferija predstavljaju celinu sa jasno definisanim zadatkom koji obavljaju. Kroz namenu MKR sagledavamo sve hardverske zahteve i probleme vezane za upravljanje tako formiranim sistemom.

Odnos računara i periferije definiše se konstrukcijom interfejsa (međusklopa za komunikaciju). Primenjeno rešenje neposredno definiše upravljanje, odnosno određuje radnu sposobnost MKR. Sam interfejs formalno predstavlja način komunikiranja unutar sistema računara-periferija.

Profesionalni hardverari imaju ružan običaj da u prvi plan stavljaju samu mašinu, kao da ona određuje sve mogućnosti sopstvenog korišćenja. U robotici, međutim, to definitivno nije tako. Hardverski projekat potpuno je podređen nameni robota. Prvo se utvrđuju potrebe i namena MKR, a tek potom i na osnovu toga koncepcija hardvera. Formiranje interfejsa računara-periferija je prvi korak na tom putu.

Problem koji se odmah postavlja pri konstrukciji interfejsa je izbor jednosmernog ili dvosmernog prenosa informacija između računara i periferije. Da bismo izveli pravilan izbor, neophodno je teorijski

3007 LOG. KOLA U IC-u	NI	NIII	3007 ULAZI ZA 2 FUNK. LOG. KOLE	
4	*2	74LS00	74LS02	2
3	*3	74LS10	74LS27	3
2	*4	74LS20	nepotrebni	4
1	*8	74LS30	nepotrebni	8

Sve linije sa nivoom logičke nule sa konektora treba dovesti na ulaze ILI kola ako na izlazu želimo logičku nulu.

Ako nemamo odgovarajuće IC kolo, upotrebimo koje imamo, prema priloženoj tabeli 2, ali gornja pravila moramo poštovati.

Pravila važe i dalje, kod spreznja signala. Uvek treba pisati oznake na izlazima logičkih kola. Tada je šema preglednija, a mogućnost greške manje. Spreznja treba vratiti kao na šemi 6, sve dok se ne dobije jedan izlazni signal.

U tabeli 2 dati su tipovi logičkih kola, sa brojem nezavisnih kapija koje sadrže i brojem ulaza za svaku kapiju. Na priloženoj mapi dat je i raspored nožica na odgovarajućem IC kolu.

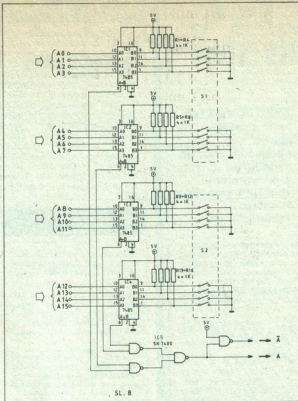
Neupotrebijene ulaze IC kola treba vezati na isti signal koji je vezan na ostale ulaze istog logičkog kola, što prikazuje detalj na slici 7.

Pozmatrajuci šemu dekodera na slici 6, vidimo da uvek postoje kola sledeći NILI i obrnuto. Kada je na ulazu jedinica, imamo NI-NILI-NILI u odnosu logičkih kola. A ako je na ulazu nula, imamo NILI-NI-NI-NI.

Osmoadresni dekodler

Opštenamenski projekti, poput malog kućnog robota, zahtevaju više od jedne adrese. Osam adresa bila bi nekakva „razumna odmere-na“ količina.

Procedura dizajna dekodera je ista. Adresu bloka od osam lokacija



SL 8

duju korektno napajanje motora, a upravljaju se signalom male snage iz računara.

Kada bismo na konektor „galakaje“ direktno vezali jednosmerni motor male snage, motor ne bi mogao da radi, jer signal upravljanja nema snage da ga pokrene.

U mikroprocesorskom hardveru, korektnost signala na magistrali obezbeđuje IC kolo poznato kao BAFAER (sl. 10). Standardni TTL bafer snabdeva magistralu potrebnom snagom za pogon do 20 pratećih IC kola.

Bafer je tzv. trostatičko kolo. Koristimo ga i kao programibilni prekidač, upravljajući ENABLE signalom (ENABLE možemo prevesti kao „dozvola“). Za ENABLE=0 prekidač je otvoren (nema prenosa signala sa magistrale). Za ENABLE=1 prekidač je zatvoren i signal se sa magistrale prenosi neinvertovan.

Nema opasnosti za računara ako dodamo jedno ili dva kola na postojeću magistralu. Standardni ugrađeni baferi mogu da izdrže mala dodatna opterećenja. Ali, jedan mali kućni robot ima sigurno više od dva IC-a (za jedan upravljački signal motoru preventivno treba računati potrošnju kao za pet TTL kola). Zato i odvajamo računara od periferije baferom.

Mi ćemo za napajanje periferije koristiti uglavnom poseban izvor napajanja, fizički odvojen od računara. Sam bafer za magistralu i dalje možemo napajati sa računara, ali bilo bi kritično napajati tako i štavu periferiju. Projekat jednog takvog zasebnog izvora dajemo u sledećem nastavku serije.

DOZVOLA (INHIBIT)

ULAZ

SL 10 TRO-STATIČKO KOLO — SKLOPKA

biramo opet pri vrhu korisničke memorije. Sada je broj identifikacionih žica tri. Svaka od osam odabranih adresa postavlja zaseban upravljački signal, i selektuje po jednu periferiju.

Novost na šemi osmoadresnog dekodera je IC kolo sa oznakom 74LS138. To je dekodler adresa sa tri na osam omogućavaju izbor osam različitih nezavisnih upravljačkih linija, koje se nikad ne aktiviraju istovremeno. Za određenu kombinaciju nula i jedinica na tri ulaza u dekodler aktiviraće se samo jedan od osam izlaza iz dekodera. Pri tome, izlaz je LOW ACTIVE, što znači da je aktivan kada se na njemu javi logička nula. Ako želimo aktiviranje jedinicom, moramo dodati inverter na izlaznu liniju 74LS138.

Na šemi 7 donja tri bita adrese A0-A2 (identifikacione žice) određuju kolo će od osam mogućih adresa biti aktivna. Ostalih 13 linija (definicione žice) određuju početnu

adresu bloka od osam korišćenih lokacija za adresiranje.

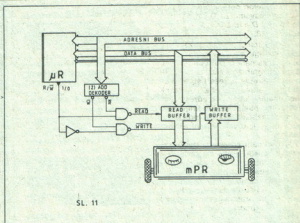
Programibilni dekodler

Za eksperimentalne potrebe, najpodesnije rešenje je upotreba četvorobitnog komparatora IC 7485. On omogućava ručno postavljanje bita koje adrese unutar opsega od 64K. Pri tome, hardverska konstrukcija sklopa nije zavisan od specifične adrese.

Šema sklopa na slici 5 predstavlja modul opšte namene. Realizaciju sklopa dajemo za 16-bitni dekodler adresa.

Dekoder, kako je ovde koncipiran, sa četiri 7485 i jednim 7400, dekodira jednu 16-bitnu adresu, blok od 16 adresa kada se izostavi IC-4, ili blok od 256 adresa kada se izostave IC-3 i IC-4.

Selektivno adresu postavljamo ručno, nizom mikroprekidača S1 i S2. U momentu kada se na ulazu kola sa adrese magistrale



SL 11

pojavi adresa postavljena na prekidačima, izlaz kola 7485 postavlja se na logičku jedinicu.

Baferi

Kod standardnih kućnih računara, procesor gotovo nikad nije direktno vezan za magistrale. Razlog za to je činjenica da procesor nema dovoljno snage da održava logičke nivoe signala na magistrali, jer kola vezana na magistralu doslovoce troše struju.

Sličan problem imamo kod upravljanja motorima. Oni pogotovo gutaju struju. Zato koristimo specijalne sklopove, koji obezbe-

Baferi imaju više namena. Na slici 11 jedan adresni dekodler upravlja baferom, koji može da opsluži kompletan mali kućni robot. Svaki put kad se na magistrali pojavi adresa periferne jedinice, bafer prosleđuje signale ka malom kućnom robotu.

Bafer nam je potreban i na magistrali podataka, ali mu je to namena nešto drugačija. Potrebno je, naime, sprečiti da više perifernih jedinica istovremeno upišuju nešto na magistralu podataka, jer bi tada nastao haos. Primenom bafera, računara daje dozvolu pristupa magistrali samo jednom uređaju u jednom trenutku.

Bratislav Bingulac
računari 36 • mart 1988. 61

AERO ionizator

AERO
ionizator

AERO JONIZATOR je elektronski aparat koji obogaćuje vazduh negativnim jonima, a istovremeno značajno smanjuje prisustvo bakterija i virusa.

U AERO JONIZATOR ugrađene su tri jonske igle sa elektronskim tajmerom koji reguliše njegovo uključivanje-isključivanje. Svaka igla AERO JONIZATORA emituje 1500 jona po 1 cm³ vazduha AERO JONIZATOR je mali potrošač (oko 1W) i radi nečujno.

AEROJONIZACIJA JE PREDMET PROUČAVANJA U RAZLIČITIM OBLASTIMA NAUKE, PRED SVEGA U MEDICINI I BIOLOGIJI.

ISPITIVANJA na VMA u Beogradu (dr Petar Paunović i saradnici) pokazala su da povećan broj jona (posebno negativnih) dovodi do značajnog smanjenja broja bakterija u vazduhu.

UPOTREBA AERO JONIZATORA posebno se preporučuje u prostorijama u kojima se puši, koje se teško provetravaju, koje imaju plinski ili centralno grejanje, koje su izložene uticaju industrijskog aerozagadenja, ili koje se nalaze u blizini frekventnih saobraćajnica.

AERO JONIZATOR PREPORUČUJE se osobama sa obolelim disajnim organima (astma, bronhitis, polenska alergija...). U slučaju nesanicne, rastrojstva, depresije, hroničnog zamora, ili loše koncentracije koriste AERO JONIZATOR.

ISTRAZIVANJA u Francuskoj, USA, SR Nemačkoj i SSSR-u pokazala su da jonska koncentracija treba da se kreće u intervalu od 1500 do 5000 jona u 1 cm³ vazduha.

EFEKTI rada AERO JONIZATORA nisu trenutni. Za prilagodavanje disajnog aparata, znavno od oboljenja, potrebno je 3—5 dana. Prvi efekti prilagodavanja su kvalitetniji — čvršći san i lakše buđenje.

NAPOMENA: AERO JONIZATOR postavite na visinu od 50—75 cm, 30 cm od bočnih zidova. Jonske otvore usmerite u pravcu dijagonalne prostorije, prema prozoru. Poželjno je povremeno čišćenje (vlažnim sunderom) jonskih otvora. Prilikom čišćenja isključite AERO JONIZATOR iz struje. Nataložena mikro-nečistoća oko jonskih otvora najbolja je provera efikasnosti rada AERO JONIZATORA.

ATEST: Institut za zaštitu na radu, zaštitu od požara i zaštitu čovekove sredine SR Srbije

TEHNIČKI PODACI:

Kapacitet do 75 m³. Izlazni napon 7500V/2mA

Dimenzije 200x120x40 mm

GARANČNI ROK 5 GODINA. SERVIS OBEZBEĐEN NARUČITE DANAS — PLATIĆETE ZA 2—3 NEDELJE (POUZEĆEM) cena: 47.100.-din.

NARUĐZBENICA: RAČUNARI 36

Neopovratno naručujem... komada AERO JONIZATORA po pojedinačnoj ceni od 47.100. din. Navedeni iznosi uveličani za pojedinačne troškove preložen pri preuzimanju poštom. Uveliči te narudžbenice važe do 15. IV 1988. godine.

Ime i prezime

Mesto i pošt. br.

Ulica i br.

Zanimanje

br. telefona

Br. lične karte

redovni u

Narudžbenice popunite štampanim slovima i pošaljite na adresu: 8102-Agencija DUGA, 11000 Beograd, Bulevar v. Mišića br. 17/11, sa obaveznom naznakom za JONIZATOR. Nećete narudžbenice ni realizuje se.

Mali oglasi

Cena običnog malog oglasa do deset reči je 3000 dinara. Svaka naredna reč košta još 300 dinara.

Cena uokvirenog malog oglasa je 4000 dinara i po visinskom centimetru u stupcu Mirne 9,5 cetera ako oglaš nije viši od pet centimetara i 6000 dinara po visinskom centimetru ako je mali oglaš visok između pet i deset centimetara.

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

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

Spektrum

Veoma povoljno prodajem programe za „spektrum“. Katalog besplatno. Obavezno nazovite telefon 011/814-870

Spektrumovci! Nudimo vam najnovije hitove, kvalitetno snimljene Program 130 dinara. Katalog besplatno. Željko Prutki, Bosanska 2, 54000 Osijek, tel. 054/54-355

KOMPLET 45: California Games (6 pr.), Halfpipe, Surfing, Skating, Footbag, BMX, Flying Disc, A — maze, Lucy and John, Fountain 2 (3 pr.), Gunship (3 pr.), Salamander. Snimak je izvrstan. Cangaroo soft, J. Gregorića 60, 44000 Sisaak, tel. 044/30-496

Spektrumovci! Izrada LOTO sistema i provera dobika. 14 najboljih, najboljih programa + kasete + uputstva = 6000 din. Momčilo Antić, Somborska 47/13, 18000 Niš

Prodajem nagradne i još jako poboljšane programe: Tekući račun (PC, spektrum), 3D Grafika (spektrum). Tel. 054/24-461

Sve vrste programa za najbolji računar „spektrum“ dobijate na adresi: Matjaž Ambróz, Molniške čete 19, 61110 Ljubjana, tel. 061/453-613

SILVERSOFI vam nudu najbolje od najboljih. Pristupačne cijene. Besplatno katalog. Antun Šoštarik, Kustožjaki 3, odvojak 4, 41000 Zagreb, tel. 041/576-844

Software Cangaroo

KOM Y: Jack the Nipper 2, Tal Pan, Ski Simulator, Dizzy, Indiana Jones, Bobbie, Joe Blade, Ghost Hunters. KOM 43: Jackal, Thunder Cats, Ace 2, Ninja Hamster (2 pr.), How to be a comp. Buster, Hercules, Return of gob. Pruncky, Tranator, Defilè 5k. Cena kompleta 1000 din. Kasete 2000 (Scotch) ili 1400 (Basf traka). Snimak je izvrstan (Technics deck). Cangaroo soft, I. Gregorića 60, 44000 Sisaak, tel. 044/30-496

Prodajem centronics interfejs za spektrum i kabl RS 232 za spektrum 128 i QL. Tel. 711-806

Spektrumovci! Velika rasprodaja!!! Komplet samo 1300 din + kasete (1700). Rok isporuke 1 dan. KOM-PILET 210: Play for your life, Football Fantasy, Micro ball, Implosion, Oriental Hero, Still flip 2, Dama, Indiana Jones (2 programa), Side wize, Slaine the King, Excalibur, Soft cuddly, KOMPLET 29: Rescue, Ski simulator, Super sprint, Psycho, Park patrol, Smash out, Mayhem, War cars, Penguin, Triaxox, Final matrix, Angle ball, Moon strike.

KOMPLET 28: Duet, Centurions, Mercenary, The tube, Last mission, Hybrid, Bubble bobble, Rapid fire, Motos, Renegade, Leviathan (3 programa).

KOMPLET 27: Jack the Nipper 2, Tai-pan (2 programa), Ghost hunters, Joe Blade, Solomon's key, Alien evolution, Prohibition, Atv simulator, Batty, Exorcism, G-man, Wizzball.

Vujujić Tamara, Lenjinova 8/11, 11080 Zemun, tel. 210-334

Spektrumovci! Komplet 1300 din + kasete (1700 din). Sva četiri kompleta samo 4000 din + cena kasete. Rok isporuke 1 dan. NABJOLJE IGRÉ 3: Rambo, impossible mission, Boulder dash 2, Strip park, Yie ar kung fu, Bonty Bob Strikes back, Marsport, international karate, Beach head 2, Fourth protocol, Dynamite Dan, Back to School, Macadam bumper.

NABJOLJE IGRÉ 4: Dynamite Dan 2, Ninja Master, Dan Dare, Knight rider, Paper Boy, Tilt Racer, Nightmare-rally, Ace, Tennis, Superman, Kamikaze, Kung fu master, Phantomas. NABJOLJE IGRÉ 5: Cobra-Stealon, Yie ar kung fu-2, Nosteratu, Scooby Doo, Galivan, 1942, SF Cobra, Speed King 2, Druid Asterix, Uridium, Great escape.

NABJOLJE IGRÉ 6: Top gun, Space harrier, Super soccer, Super cycle, Donkey kong, Motto cross, Golf-Imagine, Agent X, Legend of Kage, Archeologist, Goonies, Rogue Trooper, Vujujić Tamara, Lenjinova 8/11, 11080 Zemun, tel. 011/210-334

DECOY SOFTWARE CLUB vam nudu najbolje i najkvalitetnije programe koji su se pojavili na YU tržištu i šire. Nudimo: Gun Ship, Combat School, Salamander, Robo Cop, Out Run. Cene: pojedinačno — 200 din, komplet — 1800. Katalog dobijate prilikom prve narudžbenice. Mogućnost preplate. Ivan Rajković, Pančevačka 4, 11000 Beograd, tel. 011/186-509

GENERATOR NEGATIVNIH JONA AERO JONIZATOR

Software Cangaroo

KOM Y: Jack the Nipper 2, Tai Pan, Siki simulator, Dizzy, Indiana Jones, Bobble, Joe Blade, Ghost Hunters, Bride of Frankenstein, Super Sprint, KOM 43: Jackal, Thunder Cats, Ace 2, Ninja Hamster, How to be a comp. Buster, Hercules, Return of Gobots, Flunky, Trantor 5, Quadrant. Cena kompleta 1000 din. Kazeta 2000 (Scotch), ili 1400 (East trak). Snimak je izvrsan (Technica deck). Cangaroo soft. I. Gregorića 60, 44000 Sisaak, tel. 044/30-496

COMPUTER SERVICE

VIII Vrbiak 33a/6
41000 Zagreb
tel. 041 539-277 od 10 do 12 sati i od 15 do 17
— Spectrum, Commodore, Atari, Amstrad
— brzi i kvalitetni popravci
— prodaja josty's-a, interface-a, mrežnih ispravljača, kablova, memorijskih proširenja, rezervnih dijelova

C-64/128: Izaberite iz drugih opasla 40 igara + 10 igara poklon + kasete + PTT = 3500 din. Mogućnost preplate, igre Vloeki, Jane Sandanski 32/6, 96000 Ohrid, tel. 096/23-214

COMODO

NEVEROVATNO!! I PROGRAM = 45 DIN. Stari i najnoviji programi na kaseti i disketi. Za katalog pošliti marke. DISKETA: The Newsroom, Printshop, Print Master, Matematika, Grafika, Guro, GEOS, Platine, Giga CAD 1-2, Doodie, Koolpaint...

KASETE:

- K1: Yogi, Captain America, Bugu Boy, ...
 - K2: Trantor, Athena, Ramparts, Diablo, ...
 - K3: Int. Karate 2, Action Forza, Shanghai, ...
 - K4: Indiana Jones, Hysteria, Gun Smok, ...
 - K5: Waterpolo, Hockey, Matchday 2, ...
 - K6: Grand Prix sim., Death Valley, ...
 - K7: Sidewize, Gary Lineker's super Soccer, ...
- Miodrag Stakić, Radnička 64, 15314 Krujanj

Dada soft

C-64: HIT igre za januar i februar
K5: 35 programa + kasete = 4000
K6: 40 programa + kasete = 4500
PC-128: NAJNOVIJI programi
K2: 15 programa + kasete = 3000
K3: 20 programa + kasete = 5000
Telefon 030/21-050 od 19—21 čas

BLITZ SOFT COMPANY!!! VIC 20 (VC 20), C-64! Programi. VIC 20: programi od 3.5 K i za proširenje stigili i baše ih još! C-64: jedan program pojedinačno samo 800 din!!! Uslužni, igre, ... S.B.C. — Denis Isaković, Muhameda Džutića 33/2, 71000 Sarajevo, tel. 071/212-366, od 8 do 12 i od 16 do 22 h.

BLITZ SOFT COMPANY!!! Za vaš C-64 izvjerilo ih ne, stereo zvuk na obič. K kaseti preko kasetofona! Jedini stereo snimak možete dobiti samo kod B.S.C. — a. Proviertelj jedini u svijetu za C-64! B.S.C. — Denis Isaković, Muhameda Džutića 33/2, 71000 Sarajevo od 8 do 12 i od 16 do 22 h.



NAJBOLJI EPROM MODULI U VU ZA VAŠ C64,128

1. UNIMIKS 001: DUKLATOR, COPY 200, TURBO 200 SL, SPART LONDER, POG. BLAVE VAS, SPEZIFIKACIJE, BROT TRILONGIC, ...
2. UNIMIKS 002: ...
3. UNIMIKS 003: ...
5. TRAKAHIX 01: ...
6. TRAKAHIX 02: ...
7. TRAKAHIX 03: ...
8. TRAKAHIX 04: ...
9. DISKIKS 01: ...
10. BIRON'S BASIC VU 1.4: ...
11. EASY SCRIPT VU 1.4: ...
12. BROT: ...
17. BOLDER DASH SUPER 1.4: ...
18. COLLORAGE CHESS VU 1.4: ...

VAŠI MODULI IMA GARANCIJU FREEBET TABER! JEDI REZERVEŠE ŠE PROGRAME I: CENA MODULA 500, 50 I 14 IZ 800, DIN, MODULA BROJ 1, 2, 3, 7, 9, 10, 11, 12, 13 I 15 IZ 19,000, DIN, I MODULA BROJ 14 IZ 22,000, DIN. BUKVA DA NE MOŽEŠ DOBITI BETA 3011 I 3012, ZA 29,000, DIN. GARANCIJA ZA SVI MODULI JE JEDIN GODINA, JER VAN NEKODI VIKRIMKI KONTIET 111

SLBODAN SEKELIĆ, BULEVAR 23, OKTOBAR 87, 21000 NOVI SAD, TEL. 021/597-773

BLITZ SOFT COMPANY!!! Za C-64 50 uslužnih programa za 2500 din +3 igre po želji! PTT kasete uručavati u cijenu B.S.C. — Denis Isaković, Muhameda Džutića 33/2, 71000 Sarajevo, tel. 071/212-366, od 8 do 12 i od 16 do 22 h.

PRINTFOX YU — najbolji tekst procesor sa grafičkim editorom za C-64. Mešanje teksta i grafike. Ulo slova na ekranu i printeru. 25 kompleta slova, ... kreiranje sopstvenih znakova. Disketa u upizuv+ +poštarna=8000 din. U VIZWRITE + disketa + upizuv+ +poštarna = 4000 din. IGRE: Indiana Jones, Demons, Zolix, Stop the Express, Druids II, Gee Bee, Super Soccer I—II, Side Arms, B. Busters, Zig Zag, F. Frency, Morspicie, Survivors, Dark Side, Action Force, Time Race, Mobsters, Thundercats I—II, Fredy H. I—II, 7207, Sky Run, Flying Shark, Task Four, Sand war, Buzz of Fleet Feet, Zone II, Zodiac, Squalo, Balis, Matchboxes, Table Soccer, Rygar, Gnome, Trantor, Aardwerk, Side Wize, 40 prog.+kasete+PTT=3500 din. Ištvan Pakoci, M. Stanivukovića 92, 23000 Zrenjanin, tel. 023/63-920

Joysoft — provedite raspust igrajući najbolje igre na C-64. Komplet od 50 najboljih igara (uopšte objavljenih) za C-64 sadrži: Dig-Dug, Flipper Mag (najbolji flipper za C-64), Burger Time, Donkey Kong, Popay (automatska verzija), Paper Boy, Pac-man (atari), Green Beret... Uz komplet dobijate upisno i kraći opis igara, buklet za besplatnost, spisak igara sa projekcijama i poklon (disketa 64). Kvalitet snimka 100% za garanzovan, a rok isporuke je 24h. Cena kompleta 4900 din (sa kasetom). Takođe imamo najviše IGRE (80, 40 din.) Joysoft, V. Vlahovića 3/2—10, 34000 Kragujevac, tel. 034/68-07

Šah! šah! šah! za Commodore 64. Jedinstvena prilika za ljubitelje drvne igre. Trodimenzionalne figure, sjajna grafika, detaljna uputstva. Komplet S/88+nova kasete+PTT+uputstvo za svaki šah=4000 din. Usluga brza, tačna i kvalitetna. Nikolić Vladimir i Miroslav, Živka Jodića 9/13, 71000 Sarajevo, telefon 071/648-755.

C 16/1616+4: Prodajem sve najnovije igre po povoljnim cijenama. Tel. 059/29-297, Krešo Rujica

BLITZ SOFT COMPANY!!! C-64: specijalni programi za pravljenje demo programa na kaseti+PTT+5 igre po želji=2000 din. B.S.C. — Denis Isaković, Muhameda Džutića 33/2, 71000 Sarajevo, tel. 071/212-366 od 8 do 12 i od 16 do 22 h.

BLITZ SOFT COMPANY!!! VIC 20: komplet od dvadeset igara za +8K+20 programa na kaseti+PTT+45 igara po želji=G.S.C. (podzavr) Denis Isaković, Muhameda Džutića 33/2, 71000 Sarajevo, tel. 071/212-366 od 8 do 12 i od 16 do 22 h.

COMMODORE 64/128!! IZUZETNO PRAKLIKA! Kasetni komplet najnovijih hitova: K1: Aliens, Rambo II, Hysteria, Renegade, ... 30 hitova=2200 dinaral! K2: Ace, Fighter Pilot, Spiffins, ... 20 simulačenja letenja=2000 dinaral! K3: Waterpolo, UFO 2, Silent Service 1—3, ... 45 hitova+2 kasete=2400 dinara. Programi za izradu INTRO i DEMO programa: Real Writer, Demo Creator, Dream Writer, 1800 dinaral!! NAJBOJLI KORISNIČKI PROGRAMI 1: Vizarite, Megatape, Graph 64, Easy Script ... 50 programa+2 kasete=3000 dinaral! NAJBOJLI KORISNIČKI PROGRAMI 2: Oxford Pascal, Logos, Plassco ... 40 programa+2 kasete=2500 dinara! Ako naručite 3 kompleta POKLONI: Simon's Basic 2+hičajuđu pokovu! POUPIŠT 6 kompleta=10000 dinaral! Garantovanim kupcima besplatan spisak! GREATSOFT, Grobjanika 21, 11306 Grotka, tel. 011/893-618 Dejan.

Magnus soft za Commodore 64 nudi svojim cijenjnim kupcima komplet najnovijih i najboljih mrtovskih igara. Komplet 3/88: Red Ostober, Point Ice, Garfield, Gauslet 2, Card Sharks, Guss Sisters, Thunder Force 2, Basketball Master, Police Academy 2, Shootem Wild, Loco Switch, Magnetron 1+5, Code Hunter, Wix — Wax, Ast Cricket, Atv Simulator, Brave Star, Stratton, Apyss, Konjington, Sokoban 1+3, Fortes, Terminal Siti, +5 hitova. 35 igara + nova kasete+PTT + poklon programi = 4000 dinara. Starim kupcima popust, isporuka odmah. Nikolić Vladimir i Miroslav, Živka Jodića 9/13, 71000 Sarajevo, telefon 071/648-755.

Sport! sport! sport! za Commodore 64. Game Set and Match 1—10+Winter Olympic Game 1—10. Najbolje urađeni letnji i zimski sportovi, igrajte u udžetlje. Tenis, fudbal, košarka, atletika, boks, itd. Skijaški skokovi, slalom, bob, velesalom, biatlon, itd. Komplet sport 88+nova kasete+PTT=4000, din. Isporuca odmah. Usluga kvalitetna i tačna. Nikolić Vladimir i Miroslav, Živka Jodića 9/13, 71000 Sarajevo, telefon 071/648-755

COMPUTER SERVICE
VIII Vrbiak 33a/6
41000 Zagreb
tel. 041 539-277 od 10 do 12 sati i od 15 do 17
— Spectrum, Commodore, Atari, Amstrad
— brzi i kvalitetni popravci
— prodaja josty's-a, interface-a, mrežnih ispravljača, kablova, memo-rijskih proširenja, rezervnih dijelova

KOMODOR

Prodajem za C-64/128: Reset-turbo, turbo ostaje nakon restiranja većine programa (4000 din). Epron-modul+Resu+više turbo programa u modulu+podešavanje (15000 din), Simon's basic, Extended basic, Monitor, Easy script ... T-priključak za dva kasetofona, prenaminažavanje i zaštićenih programa (8000), Navlika — zaštita od prašine za kompjuter, disk, plati (1000 din/kom), kasetofon (800 din), palica (600 din), svjetlosno pero, programi, ... +poštarna. Zdenko Šimunić, Kolaraeva 58, 41410 V. Gorica, tel. 041/714-688

AMSTRAD

KING SOFT vam nudi sve što poželite i sve što nudi drugi, ali daleko jeftinije. Kvalitet vam garantuje jednodnevna tradicija. Pojedinačni programi su 200 din, a kompleti od 8 programa su 1000 din. Takođe nudimo i sve programe na disketama. Zato ne gubite vreme i odmah nas zovite na telefon 023/34-938 ili pišite na adresu: Danijel Spajers, Njegoševa 11, 23000 Zrenjanin.

WODDO SOFT nudi idealnu kupovinu: 20 najnovijih programa (koje vi odaberete) + kasete = 4700. Po želji programe prodajemo i pojedinačno. WODDO SOFT, Save Kovačevića 27a, 11000 Beograd, tel. 011/456-422.

AMSTRAD CLUB 007 vam nudi softvare za vaš 6128. CP/M programi: Turbo Pascal, Fortran, Pascal MT+, M.BASIC, Cobol, Wordstar, Spellstar, Dbase ii, Dr Graph, Dr Draw, Mica CAD, Multiplan, Supercalc 2, SDI, C-Basic, Copyfile, AMSDOS programi: Tasword 6128, Tasspell, Art Sudio, Profi Painter, Oddjob, Masterfile III, Format 213+updatva. Besplatan spisak. Svetozar, Sime Miroševića 35, 35000 Beograd, 011/432-026



AMISOFT YU

AMSOFT YU CP/M Software predstavlja najnovije CP/M programe: Quasar 2, Schwinter, Locoscript — 2 — Joyce, M.G.X (Mathematics' Graphic Extensions) — Jouce, Desktop Publisher — Jouce, Checker Designer — Jouce, Pagemaker, 1, 2, 3, 4, E — Basic, PL/O Compiler, Print, Supercopy, Ramdisc 64, Library (Subdirectory), SQUEEZE, Unsqueeze, micro Cobol, Forth 63, Small-C (Floating point), NEWCPM, Turbo Pascal Roe 33, CBasic 80, Exbasic, Dr Draw, Dr Graph, CP/M igre: Batman, Megan3, Almazar, monopoly, Baccarat, Adventure, 3D Clock Chess, Mogućnost isporuke svih programa sa YU setom. Amsoft YU, Trg Republike 4, 41000 Zagreb, tel. 041/270-777

Kompleti CP/M i utility programi: Komplet LANGUAGES: Fortran, Small C, JRT Pascal, Fortr. Komplet Text: Wordstar, Mailmerge, Prospell, Rotat. Komplet STATISTIKA: Amstat 1—4, Komplet 3.0: dBASE II, Supercalc 2, Wordstar 3.34, Zip, Sdi. Komplet 2.2: Microscript, Micropen, Microspread, Cambase Database. Novi AMSDOS programi: Masterfile, 6128 YU, Tasword 6128 YU, Tasspell, Minioffice 2, Profi Painter. Hardvare: Proširenje 464 na 6128 (CP/M 3.0), Silicon disc 256 k, Lightpen, EPROM-programator, EPROMi sa YU setom za printer. Amsoft YU, Trg Republike 4, 41000 Zagreb, tel. 041/270-777

AMSTRAD PCW: razmena programa, veštiki izobra. D. Stojilović, Put partizanskih baza, 21000 Novi Sad, tel. 021/397-743

AMSTRADOVCI — Eagle Soft Vam je za Vašeg ljubimca pripremio meke komplet od 40-ak megahit programa (iskada pojavljenih na kaseti: 720-simulacija vožnje akrobata, BASIL-The great mouse detective, Jack The Nipper 2, The 3D Duct, Kentillio, Lords Of Midnight, Sidewalk, Phantom Club, Captain America, Rygar, Spy Vs Spy 3, Nusferatu the Wampire 1—2, 2, Super Hang On, Trantor, War, Centurions, Jackal, Black Magic 1—2, Triacos, Evening Star) + 40 kvalitetne nove kasete + PTT = 9000 din. Izvanredan snimak uz ekspresnu isporuku pouzdem. Pojedinačno 250 din/prog. Sa-bajak Ilica, 7. Vojv. Brigade br. 62, 21208 Sr. Kamenica.

ATARI

ATARI MASTER CLUB XL/XE. Budite u vesti sa svatom preko našeg kluba. Za vaš Atari XL/XE: programi (oko 1000) na kaseti i disketi, najveći izbor literature u YU, kursevi za učenje Basic-a na kasetama, izuzetne šeme za hardverske dodatke, magazini: Atari User i Monitor, specijalni popusti, prodaja i razmena. Za katalog pošliti 500 din. Slobodan Jovanov, Prvomajka 2-A, 23000 Zrenjanin.

ST: Kad već imate najbolji hardware, imajte i najbolji softvare! Popusti i telefon u besplatnom katalogu Nebojša Arsić, Makedonska 7, 35250 Paraćin

ATARI XL/XE. Prodajem igre po ceni 200 din. Katalog besplatan. Vatroslav Jučić, Spiničić 140, 51215 Kastav, tel. 051/741-813



ATARI XL, XE, ST:
Prodaja, razmena svih vrsta programa. Besplatan novi katalog. Robert Barić, Danilaska 7, 59000 Srebrenik, tel. 059/356-17

Atari XL — super igre, besplatna uputstva, puno mapa. Katalog 300 din. Ivan Vučurović, JNA 65, 26210 Kovačica

Atari XL, XE: Speedking, Trailblazer, Mercedes... V.C.S., Miroslav Mikić, Đura Salaja 41, 42000 Varaždin, tel. 042/53-977

ATARI XL/XE. Prodajem programe, katalog besplatan. Ivan Mitić, B. Parovića 8/16, 18000 Niš, tel. 018/328-781

ATARI — Bahovec Ignj. Strelko. Novi programi (monokromatski PC — ditro, prvi Sigurni 2, Magamax 2, Standard Base, ST Pascal plus 2), nova literatura. Izaberite između 500 najboljih programa. Besplatan katalog. Pijadejeva 31, Ljubljana, tel. 061/312-046

GUNNERUS SOFT for Atari XL, XE — ako želite kvalitetno enimjene programe obratite se na G. S. G. S. ma i stare programe po povoljnijim cijenama. Katalog 300 din. Damjan Bačani, Kvedrova 36, 61000 Ljubljana

ATARI ST programi i diskete 3—5, katalog besplatan. Boris Durdin, Turinina 10, 41000 Zagreb, tel. 041/676-228 i 436-002

ATARI XL/XE — DON'T PANIC SOFTWARE. Noviteti! Niske cene Brzina, efikasnost: Besplatan katalog Vane Ulan-ski, Borko Tanev 42, 92000 Štip, tel. 092/22-182 (Vane) ili 35843 (Zoki)

ATARI ST: vrhunski programi po najpovoljnijim ceni. Spisak besplatan. Prodajem disk SF 354. Tel. 035/551-496

Povoljno prodajem 15 disketa marke Verbatim — Data Life. Diskete su 3.5 inča 25/2D. Slavko Parezanin, M. Marojević 19, 50000 Dubrovnik, tel. 050/24-229

ATARI ST — programi za sve vaše potrebe. Pojedinačno ili sačinete sami svoj komplet (do 50% jeftinije). Diskete po najnižim cenama — snimljena disketa — jeftinije na prazna na drugom mestu. Delux Vujčevića 79, 11070 Novi Beograd

ATARI ST — kvalitetni programi i igre. Kompleti. Specijalni popusti. Literatura. Katalog 200 dinara. ST — CLUB, Kadijska 113, 11126 Beograd, tel. 011/508-035

ATARI ST — programi i literatura. Sa IBM-a (preko PC emulatora): DBASE III+0, LOTUS, DR-HALO, itd. Sa MACINTOSH-a (preko ALADINA): MAC-PROJECT, ALDUS PAGE MAKER, MS-WORD, itd. Šeme, servisa uputstva, ovišćeni servis. Milar Nečaković, Baranjska 45, 23000 Zrenjanin, tel. 023/43-571

POWER WITHOUT THE PRICE

Preko 1000 programa na disketama za vaš Atari XL/XE u našem novom katalogu. NAJVEĆI izbor NAJNOVIJE IGR Software-a. Pouzdana i brza usluga. Zvonko Atilija, Zagrebčaka 21, 51000, Rijeka, tel. 051/37-723

SOFTWARE WITHOUT THE PRICE

— Programi za početnike i napredne na kaseti i disketi, u kompletu i pojedinačno. Katalog sa izborom programa je besplatan. Popusti, pakon programi, sve za vaš XL/XE. Saša Cvečević, M. Pijade 16, 44000 Sisak, 042/21-016

IBM

TERMIČKI PRORAČUN (prema najnovijem JUS-u). Linjski i specifični gubici, difuzija, toplotna stabilnost, u oplojni otpor vazduha. Interaktivni rad, bez upotrebe tablica i prethodne pripreme podataka. BB Budisa-vević, tel. 011/642-163

Za građevinarstvo, mašinstvo: program SAP 4 za IBM PC/XT/AT i kompatibilne, u delovima i celini. Izbrana i izvorna verzija 2D/3D, statika, dinamska, svi tipovi i oblici konstrukcije, pre/post procesor. Tel. 011/668-940



IBM PC SOFTWARE

NAJVEĆI IBOR SOFTWARE-A ZA IBM PC U JUGOSLAVIJU PO NAJNIŽIJI CENI. UENAMA. Word Wizard, Enable, UniCom v2.10, Anizade Layout, Formtool v. 33, Ventura 1.1, PC Storyboard... i još preko 212000 Kb vrhunskih programa najpoznatijih svjetskih proizvođača. Literaturni Pokloni! Ekstra popusti! Katalog besplatan! EE SOFTWARE, Marčevića 31, 78000 Banja Luka, tel. 078/40-940

IBM PC programi po popularnim cijenama. Za katalog — Ivan Duseparić, Hasana Kičića 1, 71000 Sarajevo

NAJBOLJE IGRE: software za PC. Niske cijene. Besplatan katalog. Dejan Rengo, 69204 Šalovci 3

Prodajem diskete DS/DD 5.25 inča 200 i 2500 din. Tel. 071/214-319

POBOLJŠAJTE VAŠ IBM PC/XT/AT. Hard disk 30 Mb i EGA kartica. USA proizvod. Kvalitet, garancija, tel. 434-480

SOFTWARE za IBM PC/XT/AT prodajem i mijenjam. Izbor 500 uzdužnih programa (FreeLance v.00, EE Designer v1.5) i 100 igara (šah Chessmaster 2000). Programi na disketama 5.25 i 3.5 inča. Antun Baksa, Ivana Milutinovića 34, 41000 Zagreb, tel. 041/254-581. Šajem katalog

Vrhunski građevinski programi za PC/XT/AT i kompatibilni: okvirne rešetke, rotirani, interaktivni i automizirani upis podataka. Za radne organizacije i pojedince. Gino Gracini, 51000 Rijeka, Kozala 17, tel. 051/517-291



Charlie Soft

IBM PC Programi i literatura. Diskete 5.25". Površino za 5000. Otes B-35 ul.Šof 71210 Irtzda

HARDVER

Prodajem ATARI ST 520+ sa 1 Mb memorije, mišem, RF modulatorom i 10 disketa, ali bez disketne jedinice. Tele-fon 025/775-623. Vlado

Prodajem povoljno „spektrum 48“, printer Seikošica GP-100A, Centronics prima-rena, Dubravko Rajnović, Maršala Tita 62, 34220 Lapovo

Spektrum 48K, tastatura TREND 2, Centroniksov interfejs za 2 dtpostaja, kasetofon, programi, literatura i dodatna ostala oprema. Dubravko Rajnović, tel. 056/741-282

Atari 1040 STF, monitor i miš prodajem. Pitati subotom i nedeljom na telefon 042/817-245

Kupujem disc drive i diskete za CPC 464, Joksimović, tel. 013/517-565 od 17,30 do 18,30, sati.

Prodajom Commodore 64, kasetofon, džojstik, T-raždelnik, oko 10 TDK kasetna sa programima. Gotovo novo — pod garancijom. Cena 700 DM. Srđan Banović, Novo Naselje bb, 85345 Denovik, tel. 062/882-734

Komodor MCS 801, kolor štampač, tel. 021/54-150

Schneider JOYCE jeftino prodaje (monitor, printer, dva diskodrajva), tel. 041/412-168

RAZNO

Jeftino prodajem diskete 3". Morčilo Karić, Krivice livade, 5/1, 18000 Niš, tel. 018/323-202

Prodajem sve spektrumsve dijelove, kasetofon, "Datarecorder", "Interface" za palice „DKronics", dvije palice, programe, časopise. Tel. 056/12-898

APPLE SOFTWARE AND HARDWARE
— RAZMJENA I PRODAJA Sistemski, dijagnostički, aplikativni, EPROM-programatorski softwara. Baza podataka, tabelarni kalkulatori, tekst procesori, grafika i igre pod operativnim sistemima CP/M, DOS 3.3, PRODOS, UCSD. Američki integrirani školski paketi. Sve vrste interfejsa-a, priprema, rezervnih dijelova i literature. Tel. 041/263-363 stan i 041/252-015 na poslu. Charlie.

Za APPLE lic računare: nudimo kompletne prevode originalnih uputstava za APPLE lic i to:
1. Apple Iic Owner's manual
2. Appleworks Tutorial
3. Appleworks Reference Manual
4. Apple Iic Scribe User's Manual
Informacije na adresu: UAD, pp 144, 71000 Sarajevo ili na telefon 071/516-885

P.N.P. ELECTRONIC

100 JERETOVA 12 BR 10261 86009-987
RADNO VREMENE: 8-12 i 16-19 subotom 8-12
Izdaci, knjige, papiri, manje dijelovi, potrošni materijal, delovi, kasete, programi, softver, servisi, kopiranje kaseti.

SPECTRIUM

COMMODORE

APPLE

IBM

MS-DOS

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG

PROLOG



LITERATURA
Za sve koji se obiljno bave računarske naučnike je i dobara knjiga.

COMETsoftwara Vas nudi originalnu LITERATURU SA ENGLESKOM JOŠTU — knjige sa vas računari
EX. SPECTRUM, COMMODORE, BBC, APPLE
- Mikroprocesori
- Programski jezici
- Operativni sistemi
- Obrada teksta
- Servisni stručnjaci

GARANCIJA ZA SVU BRZU USLUGU.
NELOVANOVIĆ Ljubisa
Petra Lovkovića 37/1112 BEOGRAD
tel: 011/59007 pošte 17h

BBC B. Master — razmena literature, hardware-a, veliki izbor programa. BBC-ovci javite se. Vlatko Stefanović, Joakima Osgovskog 5A, 91000 Skopje, tel. 091/251-847

Programa za titlovanje na MSX kompjuterima izrađujući. Kvalitetno obavljamo ribice za sve vrste filmova. Filistin, Vodarićna 7, 41000 Zagreb, telefon 041/341-485

45 igara+kazeta+PTT=4000 din.
KOMPLET 14: Iballi, Two on Two,
KOMPLET 15: Studio Art 1-2, Golf 2,
Isporuka u 24h. Anta Marulić, Oko Kmana 73, Split, tel. 058/510-736

Prodajem PC/XT računar, te disketnu jedinicu 074/21-531

KTS — C64: Prodajem beeper i najnovije hitove u kompletima i pojedinačno. Katalog, potpis, nagrade. Dragana Prokić, Bačvarčna 10, 34000 Kragujevac, tel. 034/48-324

BENZIN JE SVE SKUPLJI

BENZ SP

35.300. din.

ELEKTRONSKO PALJENJE (TRISTORSKO-TRANZISTORSKO) ZA SVE VRSTE BENZINSKIH MOTORA

ZAŠTO BENZ SP — sa ugrađenim BENZ SP-om smanjuje se potrošnja goriva za 8—14,5% zavisno od tipa i starosti motora. Kod vozila ruske proizvodnje uštede goriva kreću se i do 18%.

SA 10.000 PREDNIH kilometara i smanjenom potrošnjom goriva dvostruko se isplaćuje ugradnja BENZ SP.

ELEKTRONSKO PALJENJE BENZ SP omogućava startovanje motora u najlošijim vremenskim uslovima: na -30°C ili na + 85°C. I sa polupraznim akumulatorom u zimskom periodu motor će normalno startovati. Zaboravite „verglanje“, sa BENZ SP-om uvek imate 40.000 volti za startovanje motora.

PREDNOSTI BENZ PALJENJA: proizvoda vek akumulatora za 30—40%, a novim akumulatorima duplira vek eksploatacije. Dvostruko proizvoda vek svećicama — nije potrebno dodatno podešavanje zazoru (const. zazor 1,2 mm). Eksploatacioni vek platina višestruko je produžen (oko 100.000 km), zazor se ne podešava jer je struja jačine 10 mA. Proizvada se vek CILINDRIMA, KLIPOVIMA, IZDUVNIM VENTILIMA i kompletno izduvnoj gradnji. Radio blokada više nije potrebna — varničenje je otklonjeno.

UGRADNJA: Isključeno je svako oštećenje (bušenje) karoserije, jer se BENZ SP jednostavno fiksira za bobinu. Kontrolna lampica se pali pri davanju kontakta i svetli sve vreme rada motora signalizirajući da je BENZ SP pravilno spojen (prema shemi). Za ugradnju potrebno je 5—10 minuta, a za prelazak na klasično paljenje potrebno je desetak sekundi

GARANTNI ROK za BENZ SP je 3 GOD.

ROK ISPORUKE 15—20 dana od prijema narudžbenice

NARUČITE VEĆ DANAS — PLATIĆETE KADA DOBJETE BENZ SP USLOVI IZ NARUĐZBENICE VAŽE DO 15. MARTA 1988. GODINE

NARUĐZBENICA: RAČUNARI 36

NEOPOZIVNO naručujem... komada elektronskog paljenja BENZ SP po ceni od 35.300 din. Navedeni iznos uvećan za poštanske troškove platit ću poštom pri preuzimanju pošiljke. Uslovi iz narudžbenice važe do 15. IV 1988. god.

Ime i prezime

Pošt. br. i mesto

Ulica i br.

Br.ik. izdata u

Zanimanje telefon br.

Narudžbenicu popunite štampanim slovima i pošaljite je na adresu: BIGZ — Agencija DUGA, 11000 Beograd, Bulevar voj. Mišića 17/III sa obaveznom naznakom za „BENZ SP“

Pet plus

Uređuje: Nevenka Spalević

Algoritmi

Formalna definicija algoritma

U saradnji sa Образovnom redakcijom Radio-Beograda

Prema nastavnom programu bloka „Informatika i računarstvo“ algoritima su eksplicitno posećene metodске jedinice Algoritmi. Intuitivna i formalna definicija algoritma, Osobine algoritma, Načini zadanja algoritama (dijagrami toka) i Algoritamske strukture (linearna, razgranata i ciklična). Kažemo eksplicitno, jer sve ono što treba uraditi za 48 časova vezbi podrazumeva dalje produbljivanje pojmova koji se tekstu na ovim časovima. Zavod za unapređivanje vaspitanja i obrazovanja grada Beograda uputio je školama orijentacioni raspored nastavnih tema, prema kome su za izvođenje ovih sadržaja predviđena tri časa. Naša metodска uputstva kako realizovati ove časove samo su orijentaciona shema, jer je metodika nastave računarstva još uvek u povoju. Nastavnicima se ostavlja da polazeći od realnih uslova u razredu dopune predloženu metodсku shemu konkretnim sadržajima, formom i nastavnim sredstvima.

U prethodnim napisima serije o algoritima podrazumevali smo da je intuitivno jasno šta znači ovaj pojam. Sve do početka ovog veća reč algoritam se i koristila jedino u tom smislu, kada bi se navodili konkretni postupci za rešavanje zadataka određenog tipa. Međutim, pitanje formalne definicije algoritma moralo je doći u središte pažnje matematičara, jer bez takve definicije nije moglo biti dokazano da za neke klase problema ne postoji rešenje. Kao rezultat napora da se postavi formalna definicija algoritma koja bi obuhvatila i njegov intuitivni sadržaj, tridesetih godina ovog veka nastao je čitav niz teorija. Za kraj naše mini serije o algoritimima evo nekoliko reči o formalnoj definiciji algoritma zanimljivoj sa stanovišta računarstva, jer pojam algoritma otkriva razmatranjem postupaka ostvarenih na računaru.

Tjuringova mašina

Engleski matematičar Alan Turing predložio je 1936. godine idealizovan model računarske mašine za koju je tvrdio da može realizovati svaki algoritam. Ova teorija koja je pojavom elektronskih računara dobila punu potkvu, zasnovana je u vreme kada takvih računara upotrebe nije ni bilo. Što je još značajnije, uskoro se pokazalo da su i teorije drugih matematičara koji su u težnji da daju formalnu definiciju algoritma pošli od različitih tehničkih i logičkih razmatranja i došli do različitih definicija — međusobno ekvivalentnih.

Tjuring je pri formalizaciji pojma algoritma počeo samo od opšte ideje oponašanja rada čoveka koji računa u skladu sa nekim strogim uputstvom. Videli smo već da se korišćenje svih algoritama, od najjednostavnijih do najsloženijih, vrši

uvek po precizno opisanim elementarnim uputstvima — algoritamskim koracima, sasvim „mehanički“. Da bi neka mašina izvršila dati algoritam, dovoljno je da ima (kako je još Bebić zamislao):

• uređaj za pamćenje podataka (ulaznih, izlaznih i međurezultata) — MEMORIJU, koju u analogiji sa čovekom koji računa odgovara listu papira;

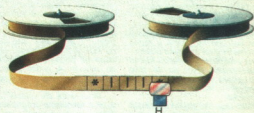
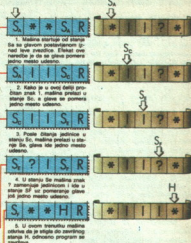
• uređaj za vršenje osnovnih operacija — ARITMETIČKI ORGAN, prema analogiji — čovekov mozak ili neko prilično računsko sredstvo i

• uređaj za upravljanje čitavim procesom — UPRAVLJAČKI ORGAN, koji na osnovu rezultata prethodnog koraka upućuje na sledeći. Pri tome Tjuring nije imao u vidu neku konkretnu mašinu sa neminovnim vremanskim i prostornim ograničenjima, već idealizovan

TJURINGOVA MAŠINA

(PREMA: The Home Computer Course 1978/81.)
(Izvor: The Home Computer Course 1978/81.)

S	*	*	S	R
S	0	0	S	R
S	1	1	S	R
S	0	0	S	R
S	1	1	S	R
S	0	0	S	R
S	1	1	S	R
S	?	0	S	R
S	?	1	S	R
S	*	*	H	R



model koji se od današnjih računara razlikuje po tome što ima neograničenu memoriju i što je razgranat računskog postupka na elementarne korake dovedeno do granicne mogućnosti.

... i kako radi

Tjuringova mašina (Sl. 1) sastoji se iz tri uređaja: za pamćenje, za

izvršavanje naznačenih operacija i za upravljanje. Uređaj za pamćenje je traka izdeljena na jednake delove — ćelije koje se mogu nalaziti u jednom od konačno mnogo stanja označenih slovima A, C, A, ..., Prn koje čine tzv. spoljašnju azbuku mašine. Jedno od tih stanja, recimo AA, nazivamo praznim stanjem. Idealizacija ovog uređaja ogleda se u tome što je ostavljena mogućnost neograničenog dodavanja ćelija u prazno stanje s leve i desne strane trake, čime mašina dobija memoriju neograničenoj kapaciteta.

Izvršavanje operacija odvija se u logičkom bloku L, koji se takođe može naći u jednom od konačno mnogo stanja — unutrašnjih stanja mašine koja možemo označiti sa Sa, Sb, ..., S i opet, radi određene, izabrati Sh (ili kraće H (halt)) za označavanje završnog stanja kojim se regulisane prestanak rada i Sa kao oznaku za početak rada mašine. Operacije koje mašina da izvrši Tjuringova mašina vrši su jednostavne — pomeranje glave za upis i čitanje utove ili udesno za jednu ćeliju, upis jednog znaka date azbuke u jednu ćeliju ili brisanje sadržaja ćelije nad kojom se nalazi glava. Međutim, i najsloženije računске postupke možemo razložiti na ove jednostavne operacije. Na osnovu sadržaja ćelije trake koja se

Programerska lektira

Za pripremu ovih napisa korišćeni su sledeći izvori:

1. N. Parezanović: Osnovi tehnike i proizvodnje;
2. N. Parezanović: Računari i programiranje za 3. razred usmerenog obrazovanja;
3. Izučenje osnov informatike i vidilite[ne] tehniki — Komplet udbenika i metodskih uputstava kojima se u 9, 10. razredu svojetskih škola uvodi računarska pismenost;
4. Fuori, Aufero: Computers and Information Processing — odgovarajuć američki udbenik;
5. D. Knuth: The art of computer programming (1. deo);
6. N. Krinčić — Algoritmi vokrug nas;
7. B. A. Trahtenbot: Što su algoritmi;
8. Predavanja N. Božović i Ž. Mijajlovića, Profesora PMF-a u Beogradu iz predmeta Matematičke osnove računarstva;
9. T. Katanjić: Zbirka rešenih zadataka iz uvoda u programiranje (3. deo).

Svima vama koji želite da dobro savladate programiranje preporučujemo da neku od ovih knjiga ili neku drugu koja se takođe bavi algoritima potražite u biblioteci i pažljivo je pročitate. Na kraju, naša osnovna poruka: najvažnije je da upotrebe rešite problem, a ako to već možete, potrudite se da vaše rešenje bude što je moguće bolje i da uvek važi.

Arhimedesov kutak

Komisija u sastavu Milan Čabarkapa, Gorica Gergelj i Nevenka Spalević, svi profesori programiranja Matematičke gimnazije u Beogradu, s ovom prilikom pregledala je i ocenila rešenja konkursnih zadataka, prisepala u redakciju do zaključenja broja.

Prema tim ocenama sastavljena je januarska rang lista najboljih jugoslovenskih mladih programera.

Najbolji rešavači u januaru 1988.

1. Miloš Prvulović (1), Beograd — 8
2. Ivan Stanisavljević (7), Šid — 7
3. Igor Ikodinić (1), Beograd — 6
4. Davor Obradović (8), Split — 6
5. Aleksandar Trifunović (2), Beograd — 6

Dobra rešenja dobili smo i od Gajić Zlatka (4) iz Beograda i Pejčinov Zorana (3) iz Skopja, ali kako jesu oni učesnici starijih razreda, pregledali smo ih van konkurencije. Ona će, međutim naći svoje mesto u zbirci rešenja svih drugih konkursnih zadataka.

I ovom Prilikom Miloš Prvulović od „Arhimedesa“ dobija knjigu na dar, a biblioteke Matematičke gimnazije „Veljko Vlahović“ iz Beograda dobija ukorčen komplet „Računara“ za 1987. godinu.

Rekli smo već da će posle petog kola najbolji programeri biti pozvani na takmičenje iz programiranja. Sada bi lista najboljih mladih programera izgledala ovako.

Generalni plasman posle 2. kola

1. Miloš Prvulović (1), Beograd — 17
2. Ivan Stanisavljević (7), Šid — 15
3. Igor Ikodinić (1), Beograd — 11
4. Davor Obradović (8), Split — 10
5. Ivo Beroš (2), Jelša — 9
6. Aleksandar Trifunović (2), Beograd — 9
7. Branko Kovačević (2), Crvenka — 8
8. Bojan Šćepanović (2), Beograd — 8
9. Fredrag Mišković (2), Gor — 7
10. Nikola Paljetak (2), Zadar — 7
11. Dragan Smiljanić (2), Gospić — 7

1. Konkursni zadaci — mart '88

1. Sa ulaza se zadaju koordinate centra i poluprečnik kruga $k(O(a,b),r)$ i koordinate tačke $T(x,y)$. Sastaviti algoritam i program koji će proveriti i štampati da li tačka T pripada krugu.

2. Sa ulaza se zadaju koordinate (x_1,y_1) ; (x_2,y_2) i (x_3,y_3) temena trougla i koordinate (a,b) tačke M. Sastaviti algoritam i program koji će proveriti i štampati da li tačka T pripada trouglu.

Rešenja zadataka sa imenom rešavača slati na zasebnim papirima. Uz rešenja obavezno priložiti evidencijsni listić ulijevu fotografiju, što nam je neophodno zbog ukupne evidencije o takmičarima i školama iz kojih se javljaju.

Rešenje svakog zadatka treba da ima algoritam, listing i kratko objašnjenje programa, a poželjno je da ima i osvrt na moguća poboljšanja programa.

Rešenja treba da stignu u redakciju do 20. tekućeg meseca.

Evidencijsni listić

Prezime i ime _____
 Razred i odeljenje _____
 Škola i mesto _____
 Broj poena (ispunjava komisija) _____

Napomena: Zbog velikog interesovanja učenika, u sledećem broju objavićemo i pet dopunskih zadataka čija rešenja mogu zamisliti zadatke koje u dosadašnjim kolima niste rešili, ili najlabilije ocenjena rešenja. Prema tome, nije kasno da se uključite u naše takmičenje. Posedimo da je ovo takmičenje za učenike zaključno sa 2. razredom srednje škole, a usmerenjaci i stariji mogu konkurisati jedino za najbolja rešenja.

Kako ocenjujemo

Rešenje svakog konkursnog zadatka može doneti učeniku od 1 do 5 bodova. Samo najbolje dotad poznato rešenje može doneti 5 bodova. Pod dobrim rešenjem podrazumeva se da je rešenje tačno, pregledno i korektno obrazloženo, da odgovarajući program radi što je moguće brže i zauzima što je manje moguće memorijskog prostora. Ako je rešenje dato tako da se može vršiti uoštavanje, ima šansu da bude trenutno najbolje rešenje. Elegancija rešenja, mislimo pod tim na njegovu jednostavnost, takođe se uzima u obzir prilikom ocenjivanja. Dobra rešenja u zavisnosti od toga koliko su dobra mogu dobiti tri ili četiri boda, a tačno rešenje koje ne zadovoljava prethodne zahteve donosi 2 boda. Ako se iz iste škole dobije više istovetnih dobrih ili najboljih rešenja, svi rešavači dobijaju po dva boda, jer nismo u mogućnosti da utvrdimo ko je u stvari autor takvog rešenja.

Zadaci drugog kola su, izgleda, bili dosta teški za naše mlade programere, posebno drugi. Nijedno rešenje nismo ocenili sa 5 poena, a objavljujemo, uz minimalne izmene, rešenje prvog zadatka Davora Obradovića, učenika osmog razreda iz Splita i rešenje drugog zadatka Miloša Prvulovića, učenika 1. razreda gimnazije iz Beograda. Oba rešenja ocenjena su sa po 4 poena. Nekoliko rešenja drugog zadatka ocenjeno je sa 4 poena, a izabrali smo kao najbolje Milošević zato što je njegov program nešto brži jer ne troši vreme na konverziju brojeva u azbučne promenljive i ne uvodi matrice kojima bi trošio memorijski prostor za čuvanje međurezultata. Međutim, u originalu rešenja bilo je malih grešaka (koje su ispravljene u priloženom listingu), pa zadatku nije ocenjen sa 5 poena. Rešenja prvog zadatka koja se zasnivaju na deljenju ocenili smo sa 2 poena (sem kod Bojana koji je ponudio još jedno preko logaritmovanja, ali sa osvrtom na tačnost), jer je smisao zadatka da netačno stepenovanje i množenje svede na tačnije sabiranje.

Zbog zahteva da se zadaci koncizno postave, mnoge stvari koje se podrazumevaju za starije programere nismo navedene, što očigledno dovodi u zabunu naše mlade kolege. Stoga ćemo, uz izvinjenje zbog prethodnih nedorečenosti i postavlji, od ovog kola preciznije definisati zahteve zadatka.

```
1 REM 1. JANUAR — GAJIĆ
10 CLS : T=0 : R=0
20 INPUT "BROJ":X
30 IF X<>INT(X) THEN GOT0 20
40 IF X=0 GOT0 180
50 IF X<0 THEN T=1 : X=ABS(X)
60 INPUT "STEPEN": N
70 IF N<>INT(N) THEN GOT0 60
80 IF N=1 THEN R=X : GOT0 180
90 IF N<1 THEN GOT0 60
110 Z=X
120 FOR A=1 TO N-1
130 FOR B=1 TO X
140 R=R+Z
150 NEXT B
160 IF A<>(N-1) THEN Z=R : R=0 : NEXT A
170 IF T=1 THEN GOSUB 190
180 PRINT R : END
190 IF N/2<>INT(N/2) THEN R=R-
200 RETURN
READY.
```

```
1 REM 2. JANUAR — PRVULOVIĆ
10 INPUT "BROJ CIFARA":N
20 DIM A(2*N), B(N), R(2*N)
30 PRINT "UNESI PRVI BROJ, CIFRU PO CIFRU"
40 FOR I=N TO 1 STEP -1 : INPUT A(I) : NEXT I
50 PRINT "A SADA DRUGI BROJ:"
60 FOR I=N TO 1 STEP -1 : INPUT B(I) : NEXT I
70 P=0
80 FOR I=1 TO 2*N
90 S=0
100 IF I<=N THEN X=1 : Y=I ELSE X=I-N+1 : Y=N
110 FOR J=X TO Y
120 S=S+A(I)*B(I-J+1)
130 NEXT J
140 S=S+P
150 P=INT(S/10)
160 S=S-S-10*P
170 NEXT I
180 IF R(2*N)=0 THEN U=1 ELSE U=0
190 FOR I=2*N-U TO 1 STEP -1
200 PRINT R(I) : NEXT I
210 END
READY.
```

Kako to rade u Trsteniku

Malo para, malo računara

Nakon objavljivanja preporuke Prosvetnog saveta Srbije kojom se određuje tip računara za upotrebu u obrazovanju, usledile su različite reakcije. Neki su bez mnogo dvomljenja postupili po pravilu, „za što manje para što više računara“, drugi još razmišljaju, gledajući šta komšije rade i ne mogu da se odluče, treći prikupljaju sredstva... Evo jednog nešto drugačijeg primera.

Prva asocijacija vezana za Trstenik, mali grad u centralnoj Srbiji, obično je fabrika „Prva petoletka“. Za to ime vezani su mnogi proizvodni uspehi, svetski ugled, veliki izvoz, saradnja sa najvećim svetskim firmama... U takvoj sredini razvila se tradicija ulaganja u nove tehnologije i mnogo razumevanja za obrazovanje i probleme sa kojima se ono suočava. Jos 1984. godine za potrebe obrazovanja samoinicijativno je nabavljeno 20 računara „spektrum“, a prošle godine je započeta akcija opremanja osnovnih i srednjih škola.

Budimir Atanasković: „Ovo što sada učinimo ne može se tek tako izbrisati!“



U prvom koraku, svih sedam osnovnih škola na teritoriji opštine opremljeno je IBM PC kompatibilnim računarima. Samoupravna interesna zajednica osnovnog obrazovanja opštine Trstenik pri tome je pokazala veliko razumevanje i pružila najveću moguću pomoć. Razgovarali smo sa profesorom **Budimirom Atanaskovićem**, sekretarom ove zajednice.

— Našavši se pred obavezom da naše osnovne škole opremimo računarima, bili smo svesni da se korak koji učinimo ne može tek tako izbrisati. Na opremi koju mi nabavimo obučavaće se desetak narednih generacija deka. Sve njih mi vidimo kao buduće stručnjake koji će ovaj kraj još više uzdizati i unaprediti. Ovo su razlozi zbog kojih smo od stručnjaka iz udruženja rada i obrazovanja formirali posebnu komisiju za nabavku opreme i podržali njenu preporuku da se ne kupuje mnogo jeftiniji, već manji broj kvalitetnijih računara. Preračunali smo se, odvojili najviše što smo mogli i krenuli, u stvari, u višegodnje, postepeno ulaganje na ovom polju. Prvi korak je zato bio i najveći i predstavlja osnovu za dalji rad. U narednom periodu postepeno će se dokupljivati nova oprema, ali to je sada u drugom planu. Niska glavna aktivnost biće usmerena ka osposobljavanju nastavnika i učenika za korišćenje ovih računara kroz kvalitetnu nastavu, organizaciju tečajeva, nabavku literature i programa. Pored toga, radi se na pripremi prostora u koji će računari biti smješteni, opremanju kabinata i slično.

Dragoslav Dobrić: „Kupovina osamobitnih računara predstavlja lošu investiciju“

Komisija osnovana sa zadatkom da izabere odgovarajući računar shvatila je ovaj zadatak ozbiljno. Nekoliko meseci je s tim **Dragoslav Dobrić**, **Zoran Prica** i **Dragiša Nedeljković** analizirao različite ponude, obilazio sajmove i proizvođače. Na kraju je doneta odluka da se od slovenačkog „Mikrohit“ kupi oprema za računarske učionice u 7 osnovnih škola na teritoriji opštine na, svo šteta o prvim iskustvima u radu je i kompletna lokalna mreža sa 10 računara. **Dragoslav Dobrić** je ukratko opisao šta je sve uključeno u donošenje konačne odluke.

Prilog:

Oprema za računarsku laboratoriju

- 1 centralni računar, IBM PC/XT kompatibilan
- 640 kB unutrašnje memorije
- 30 mB hard-disk
- 1 floppy-disk 360 kB
- HERCULES grafička kartica i monitor
- miš
- štampač 132 kolone EPSON FX-105
- priključak na lokalnu mrežu
- 5—9 radnih stanica, IBM PC/XT kompatibilnih računara
- 512 kB unutrašnje memorije
- 1 ili 2 floppy-diska 360kB
- HERCULES grafička kartica i monitor
- priključak na lokalnu mrežu

— Komisija je na početku rada kao jedini realan oislonac imala preporuku Prosvetnog saveta, pa se zato najpre pristupilo analizi opreme koja se u njoj navodi kao pogodna za upotrebu u obrazovanju. Posle vrlo detaljne analize, zaključili smo da kupovina malih osamobitnih računara predstavlja lošu investiciju, jer su mogućnosti takvih sistema nedovoljne za loše ozbiljniji rad (nemaju operativni sistem, imaju samo bežik...)

Kako se radi o većem broju računara, investicija u opremu nije mala, pa se zato ne bi smelo dozvoliti da se koriste samo lokalno, kao u slučaju u okviru nekog predmeta, već njihovom primenom treba obuhvatiti što veće područje. Ako se zahtevi tako definišu, namade se zaključak da treba nabaviti PC kompatibilne računare. Takvi računari su relativno skupi, ali se smanjenje ukupnih troškova može postići izborom minimalne konfiguracije računara, ali sa mogućnošću uključivanja u lokalnu mrežu. Time se dobija maksimalna fleksibilnost i iskorišćenost svakog od elemenata sistema.

Kao minimalnu konfiguraciju videli smo računar sa jednom disketnom jedinicom, jer je to najmanja celina sposobna da radi van mreže, što dozvoljava da se dinamika ulaganja prilagodi mogućnostima. Pristalo je još da se iz mnoštva različitih proizvođača izabere onaj kod koga je odnos kvaliteta/cena najpovoljniji. Proizvođač koji danas u Jugoslaviji vode PC računare ima mnogo, ali su njihove ponude vrlo različite. Pri opredeljivanju se, međutim, kao presudni element pojavio zahtev za lokalnom mrežom. Od svih ponuda koje smo dobili, jedino je ljubljanski „Mikrohit“ bio u stanju da na zadovoljavajući način demonstrira svoju mrežu, a da pored toga ponudi povoljniju cenu i kratak rok za isporuku.

Dragiša Nedeljković: „Naš cilj je da sveka laboratorija ima višestruku namenu“

Pokazalo se da je ova odluka bila ispravna, jer je čitav posao obavljen za vrlo kratko vreme i na visokom poslovnom nivou. Sada, kada je oprema dobijena i instalirana, svo šteta o prvim iskustvima u radu kaže **Dragiša Nedeljković**, nastavnik u OŠ „Mirograd Čajetina Čajka“ i aktivni učesnik svih protokih zbivanja;



— Kupovina računara i ostale opreme predstavlja tek mali korak na putu do stvarnog uvođenja računara u obrazovanje. Naš cilj je da svaka laboratorija ima višestruku namenu. To je, pre svega, mesto na kome će se sticati praktična znanja iz računarstva, kako na nastavi, tako i van nje. Dakle, to je kabinet za nastavu informatike i računarstva, mesto na kome će se sticati osnovna znanja o računarima i programiranju, ali i mesto na kome će se okupljati talentovani i posebno zainteresovani učenici, u okviru sekcija i drugih nastavnih delatnosti. Primena u nastavi vidi se kao vrlo široko i otvoreno polje primene, ali koliko će se i kako to iskoristiti zavisi i od broja kvalitetnih obrazovnih programa koji će se moći nabaviti.

U predstojećem delu posla nametnu se dva osnovna problema: problem kadrova i problem programa i literature. Prvi problem pokušavamo da rešimo pojačanom propagandom među nastavnim kadrom, posebno nastavnicima matematike, fizike OTP-a. Smatramo, takođe, da školski psiholozi i pedagozi treba da dobiju potrebno obrazovanje iz korišćenja računara, pa su zato i oni uključeni u početne tečajeve korišćenje i programiranja računara. Do sada su održana dva takva tečaja, za osnove korišćenja DOS operativnog sistema i programiranje u bežikcu, a za januar mesec se predviđa još jedan, specijalizovan tečaj iz obrade teksta na računaru. Planiramo da sličnim tečajevima obuhvatimo sve značajnije oblasti primene računara i na taj način omogućimo što većem broju zainteresovanih da koriste računar u svom radu.

Što se tiče programске podrške, nameravamo da deo poslova vezanih za školu kao radnu organizaciju prenesemo na računar. Tu spada priprema različitih pisanih materijala, školska statistika, pa i vođenje biblioteke, raspored časova i obračun ličnih dohoda. Priliv obrazovnih programa oduševljuje kroz saradnju i razmenu sa drugim školama, a na tom polju nam je obećana pomoć „Mikrohit“. Nadamo se da će u našem obrazovanju ovaj problem uskoro biti detaljno razmatran i da će se organizovanom akcijom doći do prvih obrazovnih programa za potrebe nastave.

U prvom polovini 1988. godine računari treba da uđu i u srednje škole u Trsteniku. Čini se da „ulaganje za budućnost“ tako kreću pravim tokom.

Programi iz školjke

Beogradski srednjoškolski će od proleća ove godine početi da štite prva iskustva na računaru TIM-011. Najnoviji proizvod instituta „Mihailo Pupin“ iz Beograda zapažen je po svojim kvalitetima i pre nego što je stigao u škole. Na prošlogodišnjem Međunarodnom sajmu „Učila 87“ stručni žiri mu je dodao Zlatnu plaketu. Serijom članaka iz pera Milana Tadića, jednog od saradnika na ovom projektu (konstruktor Nenad Dunjić, saradnici Milan Tadić i Ljubiša Gavrilović) nastojali smo da što temeljitije prikažemo novu Yu mašinu.

Imenovani katalozi

Mehanizam koji je ostao od CP/M-a-2.2, u radu sa diskom razlikuje 32 korisničkih područja (user area) koja prostro obeležava brojevnima od 0 do 31 neposredno iz jednoslovnog imena diska. Taj mehanizam poboljšan je na dva načina: prvo, prompt (poruka koju ZCPR3 ispisuje na ekranu kada očekuje komandu) sadrži pored imena diska i broj korisničkog područja. To jeste poboljšanje, ali je ipak neprijatno pamtići da se bezijk programi nalaze na A7 a tekstove na A13. Zato je uveden sledeći modul koji se zove „named directory“ (imenovani katalozi). U njemu se nalaze informacije o imenima dodeljenim diskovima i/ili korisničkim područjima.

Potrebno je jasno razgraničiti ovakvu organizaciju od hijerarhijskog stabla kataloga koji su zapisani na disku. Ovdje su informacije logičke prirode. Svi direktorijumi (sinonimi: katalozi, korisnička područja) su na istom nivou i fizički se nalaze ravnomerno na disku. To znači: a) da svi katalozi ne zauzimaju dodatni prostor na disku i b) da je pristup svim katalozima jednako brz, za razliku od hijerarhijske strukture gde svaki direktorijum predstavlja novu datoteku, a pristup tek u petom nivou kataloga jednak je po brzini kao pristupanje do šest teka. Oba mehanizma imaju i svoje prednosti i svoje mane, ali sama ideja razdvajanja diska na više logičkih celina omogućava uvođenje malo neophodnog reda u nepregledne nizove teka. Pored prednosti, ovaj sistem donosi i neprilike: poziv nekog programa ili komande zahteva ili da se nalazimo u istom korisničkom području kao i program ili da poznamo i eksplicitno navodimo ime diska i područja u kome se program nalazi, što je veoma zamorno i podložno greškama.

Zato je u strukturu ZCPR3-a uvedena mogućnost definisanja search path-a (tabela kataloga za pretraživanje). Kada pokuša da izvrši komandu, ZCPR3 kreće od svojih ugrađenih komandi; ako nijedna od njih ne odgovara ide dalje pretražuje tabelu modula različitih komandi i modula komandi kontrole teka; ako je nema ni tu, prozvaće disk i pokušava da je pronađe u aktivnom korisničkom području. Neuspех će dovesti do odustajanja i prijavljivanja greške samo u slučaju kada nije definisana tabela područja za pretraživanje. U slučaju kada jeste, ZCPR3 će nastaviti da traži sve dok ne pronađe traženu komandu ili dok ne dođe do kraja tabele. Ako sada područje

A0 nazovemo DOS, u njega smestimo sve komande i uvedemo putokaz (path) do DOS-a, možemo biti sigurni da će svaka komanda biti pronađena i izvršena bez obzira na trenutno aktivno područje.

Kontrola teka

Verovatno ste bili u prilici da radite neki posao koji zahteva često ponavljanje istih istih komandi. Gora varijanta istog problema je kada se komande izvršavaju relativno dugo, pa morate da sedite pored računara i čekate da se završi jedna, kako bi otkucali sledeću komandu. Naravno, u međuvremenu ne možete da radite ništa drugo. ZCPR3 nudi rešenje: možete otkucati sve komande, odvojeno tačkazarezom, a računar će ih izvršiti jednu po jednu. Vas omiljeni strip ponučiće vreme dok računar ne završi svoje brdo posla. Tada ćete ponovo otkucati komandu od 214 znakova i pri tome pogrešiti... Počmo stiče u obliku programa Alias: otkučate komandu liniju jedne komande, dodeliti joj neko (kratko) ime i smisliti je na disk. Sledeći put otkučate samo ime komande, a Alias će je za Vas smestiti u komandni bifer.

Lepota cele ove šeme je u tome da neka od komandi koje ste stavili u Alias može da bude poziv neke druge alias-komande. Praktično, možete da generišete vrlo složene komande, u stvari prave programe na nivou operativnog sistema... osim što moraju da imaju čisto linijku strukturu. Jasno je da se bez uslovnog grananja na može daleko stići, pa je u ZCPR3 uključio još jedan modul: „flow control package“ (modul za kontrolu teka). On proširuje ZCPR3 za još četiri komande: IF, ELSE, FI i XIF. Prvi dve su jasne, treća označava kraj IF — ELSE sekvence, a XIF bezuslovno izlazi iz svih ugrađenih if-ovca (exit all ifa).

Redirekcijom nazivamo proces kada ispis našeg programa „dotevujemo“ na ekranu, pa na kraju kačemo "e, sad isto to, samo na štampaču"; ili kad spisak grešaka koje je javio kompajler ne može da stane na ekran, pa ga snimimo na disk da bi mu se kasnije divili „natatena“. Drugim rečima, otkucijava se potreba da informacije koje program šalje nekome resursu preusmerimo ka nekom drugom, ili da ih snabdemo iz nekog drugog, kada ih program traži. Stari dobri CP/M mogao je samo da premešta iz šupljeg u prazno, četiri logičke jedinice — konzolu, čitač, bušać i štampač — mogao je međusobno da izmešta i ništa više. Programi za upravljanje fizičkim uređajima bili

su fiksirani i stalno u memoriji, trebali ih mi ili ne. Redirekcija sa diskom nije bila moguća, a veza sa nekim drugim računarom uglavnom je ostavljana za velike sisteme.

ZCPR3 uvodi „input-output package“ (modul za upravljanje ulazom-izlazom), koji omogućava pisanje proizvoljnog upravljačkog programa za proizvoljno ulazno-izlaznu jedinicu. Module je moguće menjati po želji, a tokom rada se može uključivati i isključivati redirekcija određenih u/i jedinica.

Programi iz školjke

Među mnogim lepim osobinama operativnog sistema TIM-a 011 nalaze se još dve koje treba spomenuti. Prva je nastala kada je (jednom davno) nekome hakeru (koji se tada još zvaao programer) palo na pamet da ga računar poseđa na luk. Mislim, na pravi crni luk, od koga suze nezadrživo teku. Ono se zvalo: sasvim razumem, ali je poenta da je hakeru na pameti bilo nešto sasvim drugo: slojevitost luka. Processor i hardver bili bi centar, BIOS bi bio prvi sloj, BDOS drugi, a ZCPR3 treći. Korisnik i njegovi programi bi došli kao četvrti i tu se slika završavala, ali je ideja ostala: zašto je neproduktivno da se po završetku jedne komande ili programa vraćamo na nivou operativnog sistema da bi startovali neki drugi program.

Sigurno ste se hiljadu puta nervirali kada kompajler javi grešku: učitalj editor, pa učitalj program, taj kompajler... i tako u nedogled. Zar ne bi bilo lepše da direktno iz editora pokrenete kompajler i da se iz njega direktno vratite u editor, a da pri tom ne pridete ni blizu operativnom sistemu? Na žalost, haker o kome je reč nije radio u Digital Researchu već u Bell Laboratories, tako da je umesto CP/M-a „shell system“ (sistem školjki, odnosno slojeva) dobio Unix. Školjke su vrlo praktična stvar, ali zahtevaju puno memorije, snažan procesor, pa čak i multitasking operativni sistem.

Iz ove priče nastuče se da je ZCPR3 i ovdje dao svoj pečat: moguće je pisati programe u obliku školjki. Konkretno, to znači da možemo prekinuti program-školjku i startovati neki drugi. Sistem će pre startovanja sledećeg da zapamti sve relevantne parametre prekinutog programa na stogu školjki (shell stack). Po završetku novog programa, stari nastavlja onde gde je stao. Jedinica nevolja ovog mehanizma proizlazi iz konstrukcije samog CP/M-a: svaki program koji se učitava sa diska (tj. nije reziden-

tan), učitava se na lokaciju 100H i pripremljen je (prilikom kompajliranja ili assembliranja) za izvršavanje na toj adresi. Zbog toga, učitanje i startovanje novog programa povlači nestanak starog iz memorije, a to znači da će povratak u prekinuti program biti, u stvari, njegovo ponovno učitanje sa diska. Pri tome postoji značajna razlika između ovog i inicijalnog učitanja. Prvo učitanje zahteva da korisnik otkuca komandu, a izvršavanje programa počne tzv. hladnim startom (cold start), odnosno prvobitnom inicijalizacijom. Učitanje prilikom povratka iz pozvanog programa izvršće se automatski, a mehanizam i informacije zapisane u stogu školjki obezbeđuje izvršavanje programa od mesta na kome je bio prekinut.

Ovaj mehanizam je za nas značajan iz dva razloga. Ugrađenom u samu koncepciju sistema obezbeđeno je njegovo dosledno i pouzdano funkcionisanje, a sama upotreba postala je vrlo jednostavna. Drugi razlog se sam otkriva, samo ako se setimo osnovne namene ovog računara, a to je obrazovanje i to kako mladih tako i starijih, kako iz oblasti informatike tako i iz skoro svih drugih oblasti. To znači da će ovaj računar koristiti veliki broj ljudi koji nemaju dovoljno znanja za pisanje obrazovnih programa na nekom standardnom jeziku kao što je paskal ili fortran; ljudi koji ne umeju da koriste računar iz prostog razloga što to nije deo njihove struke; oni su biolozi, istoričari, hemičari, ekonomisti... Za njih je potrebno izraditi školjke koje će im omogućiti da sa računarom komuniciraju na nivou njihovih znanjaj!

Stalno govoremo da za razvijanje novih zemljama kasinimo i više od deset godina. Međutim, jedna Engleska je tek pre nekoliko godina, pojavom računara BBC B, počela masovnije da uvodi računare u osnovno i srednje obrazovanje. Poznavaoi računara moraju da priznaju da je TIM-011, iako u sličnoj klasi, mnogo snažniji od BBC-ja, naročito ako se uzmu u obzir mogućnosti koje on pruža izradi programerske podrške za obrazovne primene. Danas je u svetu privlačno mišljenje da je hardver računara samo alatka i da pravu snagu obezbeđuje softver. U TIM-u sada imamo dobru alatku i samo je stvar organizovanog, masovnog i istrajnog rada na izradi obrazovnih programa koliko ćemo smanjiti jaz prema razvijenima. Pri tome je značajno što se radi baš o području opšteg i stručnog obrazovanja koje je glavni preduslov uspeha i na svim drugim poljima.

Milan Tadić



uređuje: Slobodan Perović

U saradnji sa emajlom Radio-Beograda
„Čip i sedam jarica“

Da rešimo zajedno

Od ovog broja uvodimo novu rubriku čije ime vas sve govori. Zašto smo se na to odlučili — ima izuzetnih igara, ali istovremeno i toliko teških da nam se čini da samo zajednički napor može da da rezultat, i zadovoljstvo potom.

Za početak odabrali smo igru „Jack the nipper“.

Zahvaljujući **Vladanu Stakiću**, daćemo vam niz elemenata do kojih smo došli, a vi potom nastavite. Kao prvo, pretpostavljamo da treba da napravite osam nestajućih da bi se Ilijana ne „nevaljatometru“ ispetala do kraja.

Četiri znamo. Prvi deo Ilijane pojavio se kada Tarzana pogodi te konzervu koja se nalazi na stepeničama ivo do starta. Da biste ga potražili, morate tačno da stanete na kraj granje koje gleda na šumskog junaka.

Drugi deo Ilijane zaradite kada slobodno preplivate uz pomoć miša koji se nalazi otkrpnite u sredini lavirinta.

Treći deo dobijate kada dualnog pogodiće košulicu. Četvrti je kada inače nauništivog poglavicu naramenite da krene za vama i padne u provaljuju ili reku.

Napredamo, iskodite, da je jedan deo zadržite da noseći štiti upadnete u hram (drugacije se ne može proći pored stražara) i vašem divovniku dobacite konopac. Budite pritom brzi jer štiti se troši pa vam se može desiti da više ne možete izdati napojke, a onda je jedino rešenje da upasite računar.

Nadamo se da vam je ovo dovoljno za početak. Najbolja pojedinačna rešenja mi ćemo nagraditi, a širom su vam vrata otvorena da i sami predložite igre za ovu rubriku.

Jedan prikaz — tri honorara

U rubrici „Razbarušeni sprajtovi“ objavljujemo prikaze koje, pored stalnih saradnika, pišu i mladi saradnici — štetao.

• Za objavljeni prikaz ili poukove saradniku pripada uobičajeni autorski honorar koji se, u zavisnosti od težine igre, kreće između 5000 i 15000 dinara po prilogu.

• Četiri najbolja priloga objavljuju se i na radju, u emajlu „Čip i sedam jarica“, što njihovim autorima donosi honorar od po još 8000 dinara.

• Najbolji prikaz meseca, po oceni urednika ove rubrike, od sledećeg broja dobija nagradu 10000 dinara.

Pozivamo mlade saradnike da nam zajedno sa prikazima pošalju i svoj žiro-račun ili žiro-račun roditelja ako ga sami nemaju. To će značajno pojednostaviti i ubrzati isplatu honorara.

Usijani džojstik

Sudeći po pismu Damira Milevića iz Titovog Vrbasa, Društvo su naveli u modi. I to ne samo ovaj nov, nego i onaj stari.

Problem se sastoji u tome kako premetati strelicu sa oružja na oružje i kako izdati i pože pratorije u drugom nivou.

Za prvo potraži rešenje u gornjem desnom uglu tastature, a za drugo upotredi ključ koji u prvom nivou možeš naći na jednom jedinom mestu.

Najliži se Dragiša Đorđić iz Zagona.

Želio je zna zašto smo u broju 34 izostavili njegovo ime prilikom objavljivanja rubrike „Poukova mreža“.

Odgovor je jednostavan: ko radi taj i greši.

Nadamo se da naše izvinjenje prihvataš.

Sada citiramo deo pisma koja su nam uputili **Mihajlo Stojanovski** i **Lazar Škarić** iz Skoplja:

„Nedavno smo vam poslali članak o igri „Torbruk“, ali Vlade nas je razočarao. Potpuno smo sigurni da nismo pobrkali lončine, naime — tekat nam je ispravim. Mi upotrebili želimo da pobeđe Nemoi, ali na samom početku računar vas automat-ki stavlja u ulogu Romela. Ako želite da pobeđe saveznici, onda morate izabrati opciju za dva igrača i vešto odigrati“.

E pa lupo.

Izaberite opciju za dva igrača, zamislite čuara kao Montegomija koji posle El Alamejne bitke Romela. Dobate da pri tom leti parje na sve strane, i šaljite prikaz.

Onda bez problema pobeđujete!

Evo i jedne poruke:

Hasan Boljić iz Zagreba poručuje **Milenu Zankoviću** iz Šibenika da njegova šifra za POLICE ACADEMY ne radi i da bi mu bilo zahvalan da nastavi sa traženjem, jer je i sam uočio sate i traženju se proklete šifre — i tako je nabavio i DEBÜGER — ali uzalud.

Proklete šifro — ili Sezame, otvori se!

Ko zna — neka pomogne.

Pa piše ovako:

„Ne mogu, a da ne budem ogorđen nakon onako odglednog potkivanja „spektruma“ u zapečak upravo na stranama rezervisanim za igre. Prvo, za nas ste objavili prikaze svega dva, a za „komodor“ čak pet igara. Drugo (još gore) tu je ona stranica lista sa 29 — poukova za C-64. Kada ste objavili toliko poukova za „spektrum“, molim vas lepo? Ako već pišete o igrama, pišite lako da i mi

„spektrumovci“ budemo zastupljeni Mi smo u vedri.“

Ovo je potpisao **Ivan Čulum** iz Beograda. Ibane, stvari su veoma jednostavne — vi pišete a mi objavujemo. Onda neka se tvoja vedrina iskaza.

Nešto malo i o rekordima.

Boris Redić iz Sarajeva ponosno nam javlja da je oborio rekorde **Mateja Freca**. (Računari 33) U igri „KRAJCOU“ nakupio je ravno 481.020 poena.

Ko voli neka žovi.

Upit: **Boris** protrešuje što nismo objavili njegov pouk za igru „Ghostbusters“.

Kao prvo, taj pouk je više puta objavljen. Kao drugo, upravo je pouk samo vam dati omogućava vam da igru zavrtite elegantnije i lepše. Probaj ako ne verujete.

Još nešto za **Borisa**. Prikaz igre „Vaterpolo“ je nekompletan. Ima tu još mnogo opcija, neme smisla da stvari radimo potpovino.

Sudeći po prilozima koja nam šaljete za rubriku „Poukova mreža“, neke stvari moramo da razis-imo.

Pouk je pouk, a pouk je pouk.

Poukova mreža nije leti što i poukova. A poukine je mnogo u vas — dvoje na izigrane igre kao „Commando“, „Fire ant“, „Bruce Lee“ i tako redom.

Budite novci i nepopovijivi, a časni uzor neka vam bude Bata Spraj.

Znate već da rubriku „Razbarušeni sprajtovi“ realizujemo u saradnji sa emajlom „Čip i sedam jarica“ koju svake subote počev od 14.02. možete slušati na Prvom programu Radio Beograda.

Znate i to da za sve vrste imamo zajednički telefon — 011/339-070.

Javite se utorkom od 12.00 — 14.00 da se dogovorimo oko prikaza igara, da vam se ne desi ono što se desilo **Bojani Majuru** i **Igoru Pinteru** — pišu nam o onome što je objavljeno.

Ili pišite, kao što je to uradio **Kemal Stakić** iz Sarajeva, kome poručujemo da odekujemo prikaz igre FUTURE KNIGHT.

U na kraju još jedan poziv pomoću.

Upisite za program „PSYDELIA“ (verzija za „komodor 16“, 116, +4) šaljite na adresu **Kemal Jakić** Livanjska 40, 71000 Sarajevo.

Nadamo da za uzrast sledi deset čepave u lepini nasred Baš čardija.

Kao prvo, novac koji vam se nalazi na putu jednostavno kupujete, i, kao drugo, u raznim parkovima učestvujete na takmičenjima koja donose medalje, ali i zadržu.

Postoje četiri parka.

RAMP PARK — Ivođite razne besne gliste (skrobacije i ukoliko ni jedan jedini put ne padnete dobijate zlatnu medalju) i sto dolara.

JUMP PARK — skakote sa platforme na platformu, naravno uz ograničeno vreme.

SLALOM PARK — proizlite između zastavica i to što brže to bolje.

DOWNHILL PARK — prvi pravcati spust niz padnju.

Na prvom nivou redosled takmičenja je proizvoljan, dok je na narednim nivoima strogo određen.

Ako vam se u toku igre na ekranu pojavji natpis SCATE OR DIE, morate brzo potražiti najbliži park ili ode život. Opremu kupujete kad skupite dovoljno novca — i sve radnje kao i prvotki levo se vide na mapi koje se pojavljuju na vašem putu.

Inače, u igri se još pojavljuju i razna smetnja (biljdi, protrozici, itd). U sudaru s njima trasađete utoliko što padate sa „scate“-a, što može biti opasno ukoliko se pojavi poruka SCATE OR DIE.

Zelim vam lepu zabavu.

Ivan Rajković



STARI 720 CONING SOON FROM US GOLD
DOWNHILL PARK SV. ČIP I SEDAM JARICA

720°

Jedan dečak, „scate board“, sto dolara u džepu, i paprenu kupku oprema (kacige, patike, itd.) — otkrpnite to su elementi sa kojima ulazite u ovu igru.

Razume se, skupujete novac, da bi uvek bili novi i neopovijivi.

To se radi na dva načina:

GOTOVO JE, GOTOVO



EVERGREEN/NEVERGREEN

Igora Pintara

Evergreen

SENTINEL

Ako igre ocenjujemo ocjenama 1-10, *Sentinel* zaslužuje najmanje ocjenu 11. Fenomenalna 3-dimenzionalna grafika sa velikom brzinom iscrtaavanja, kao i maksimalno i funkcionalno iskoristljive „spektrumske“ zvučne mogućnosti, oduljevit će svakog tko se do tada davio u gomili prosječnih igara (na sreću bilo je 20—30 i iznadprosječnih igara). A tek scenario — pobjediti nekog lukavca u brzini u pravom ambijentu daje igri pravu draž. Ukratko rečeno — savršeno.

Borisa Rabića

Evergreen

KANE

Jedna od retkih kaubojskih simulacija za C-64. Grafika odlična (nikad bolju video), zvuk takođe. U prvom dijelu lukom i strijelom gađate divlje patke. Zatim jašete u gradici Kane. Tu vas čeka gomila bendita i obračun. Na kraju jašete i dostižete voz. Sve u svemu — igra koja vas danima drži uz računar.

Nevergreen

WONDER BOY

Sto puta prežvakana tema pojavila se, eto, i sto prvi put. Iako grafiku u cjelini možemo nazvati prosječnom, animacija to nije ni u kojem slučaju. Škrolovanje je tako loše napravljeno da o njemu ne vrijedi raspravljati. Da bi se otvorila mogućnost za čuši skok, traže nađ i stisnuti dvije tipke, od kojih jedna nema nikakvu drugu funkciju. Stajem takođe nije za pohvalu, posebno odštavanje nivoa. Probajte igrati se dva igrača, pa ćete vidjeti jednu veliku glupost. Sudući prema reklamama, malo previše precijenjena igra.

Nevergreen

MOTOCROSS

Nikad nisam vidio gore igre. Vozite motorcikl koji nemoralno ubrzava, a staza uopšte nije označena. Nema nikakvih prepreka, pa vam igra čudno za pet minuta. Grafika je veoma loša, a zvuk otovo da nema. Loše da ne može biti lošije!

more skupiti jedan po jedan i odlagati u sanduk (ili u mora, ili na sto, ili na sto, ili na sto). Stvar se dopunski komplicuje time što joj je vreme ograničeno . . .

... a da sve bude još crnje i gore, ometaju je službenici (dibraši). U istu tikvu s njima duvaju i svakojaki kancelarijski predmeti, kivi na Micu i gladni njenih života.

Ako je već dobjit *Kempston* ili *A.G.F.*, nakrivite kapu. Ako nije, preostaje vam tastatura (dirke sa strelicama), što će vam unekoliko otežati posao.

Ali ko je rekao da je čistakli život lak?

THE DRINKER / PIJAČ

Autor: Saša Pušica
Izdavač: Suzy, Zagreb

I mi konja za triku imamo! I mi umemo da lik iz stripa pretvorimo u sprajt.

Dotični se lik zove Superhik (nemote ni problem, naravno, piće.

Tačnije: nedostatak pića. A pošto u piću leži njegova snaga, on mora lći kud ga ođi vode i noge nose, ne bi li nakupilo dovoljno životvornih fitaša.

Njih, kao što ste i pretpostavljali, nije jako nadi: em dočra sakrivena, em je vreme ograničeno.

Da i ne govorimo o ametalima: nakrivo nasadenim kalnerima, konkurentski nastrojenim pilancima, i ostalo velja. Teško bi Superhik izliao s njima na kraj de nije povremeni krigli piva, tj. izvora energije.

Njegov (vaš) protivnik može biti kompjuter, a može i neko iz kruga vaših prijatelja. To je, kao i nivo težine, stvar vašeg izbora.

Ako se na kraju igre čvrsto opredelite za antialkoholizam, neću se nimalo zabuditi.

PEČINSKI HEROJ

Autori: Tomislav Petrović i Mario Griman

Izdavač: Suzy, Zagreb

Hajdemo malo u prošlost ljudskog roda. Budimo sat-dva preistorijski čovek.

Budimo proizvođač točkova (upravo smo taklesali četvrti). Budimo i vitez: upravo smo čuli devojčaki poziv u pomoć.

U redu, bićemo. Točkovi nas već nose ka leglu čudovišnog pauka koji je zarobio lepu kromanjonku. Naizazimo na rupe (koje preskačemo), niske grane (ispod kojih se saginjemo), močvare (koje prelazimo skačući s jedne kornjače na drugu) . . . i u prvo vreme imamo ozbiljan utisak da smo ušli u neku drugu igru, recimo *BC's Quest for Tyres*. Kasnije taj utisak bledi, dok se nosimo sa usledilcom (koji u nama vidi poslednju banusu), i potom, sa paukom koji u nama vidi otičavač svog ručka). Patak je posebno gadan, što valja i očekivati: on je poslednja prepreka.

Pod uslovom da do njega stignemo, jer svaki pad znači jedan točak manje, a ako ostanemo i bez četvrtog, možemo da se silikamo (ukoliko ovaj poslednji glagol nije donekle preuranjen za preistorijska vremena).

Komodor 64

FLOWERMAN /

CVEČAR

Autor: Damir Ambrus

Izdavač: Suzy, Zagreb

Igra jednostavna, sačinjena u klasičnom maniru. Nema mnogo mudrovanja, nema grizevanja noktiju od nervoze, nema lutanja po stotinama ekrana.

Ima sadenja cveča i broje protiv štetočina. Vi sadite, tri spodobite čupaju, vi ih jurite, a one jure vas. Dok bežite kroz vrt — za koji bi se pre reklo da je lavirint — valja usput da sadite novo cveče.

Cilj je: posaditi cveče svuda gde je predviđeno, i

Domaće Igre

Spektrum

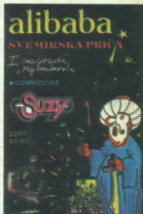
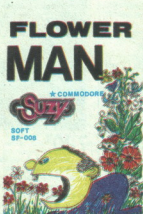
MICA SPREMAČICA

Autor: Aleksandar Radovanović

Izdavač: Suzy, Zagreb

Te žene daci nazivaju duševnom rečju „tetkice“. Odrašili, zaposleni ljudi zovu ih suvo „distačice“, „spremačice“, ili čak „higijenicarke“. Njihov delegal u našem kompjuteru, Mica, deli s nama svoje muke.

Njen je posao da dovede u red čitavu zgradu svoje firme, od tavana do ulaznog hola. Otpaci, međutim, ne stoje mirno; vrmaju se u raznim pravcima, a ona ih



GOTOVO JE, GOTOVO

Izvući živu glavu, jer je dodir sa spodobama smrtonosan. Na kraju ćete, zna se, izgubiti i posljednji život ne dovlađivši posao, jer ćete biti otkonodani, ali važno je navršati što više poena kod ste žila i aktivni.

Nastav igre — koji je, ko zna zašto, sročjen na engleskom — tvrdi da ste ovdje, ali kao što ste već primetili, u stvari ste bašтовan. (Naravoučenije: ne poseži za stranim jezikom ako imaš svoj maternji).

ALI BABA

Autori: Mario Mandić i Zoran Hrustić
Izdavač: Suzy, Zagreb

Razbojnici su se napilički blage, a Ali Baba — nadasnut svetlim primerom Robina Huda — ukrao ga je od njih i hoće, valjda, da ga vrati alirotnji. Naravno, razbojnicima to nije pravu. Kneću u alkoju, smislivši vrlo jednostavan plan: oni će grabiti vreće sa blagom, a njihov će harambaša juriti Ali Babu (tj. vas).

Vaše je da sprečite razbojnika, i da sve blago (ukupno 10 vreća) sklonite na sigurno mesto. Pregačanje sa razbojnicima isprva i nije komplikovano, iako se bore u gomili, ali harambaša će vam prilično zgorčati život.

Međutim: a) ako vam harambaša zađe iz jeda, možete između sebe i njega začas podići zid; b) možete ugubiti ključ razbojničke magaze, tj. šansu da iz njne izvadite jednu vreću; i c) kad se dohvatite poslednje vreće, brzina vam se povećava pa možete lakše uteći. Ako uspete, prelazite u teži nivo, i tako dok ne izmognete. Plaho zorli igra.

Vlada Stojiljković

Miris iz kuhinje

Najzamornije mirise, reklo bi se, impossible Mission II. Dugo joj je i trebalo da nastane, ako pomislimo na brzinu kojom se proizvođač nastavlja fudomsko stabiliji igra. Onaj sjajno animirani tričad i skakad našao se u bitno survojoj situaciji: mora obiti pet oblakodera i u svakom nadi brojanju šifru da bi se uopšte probio do glavnog negativca. Razume se, i ovdje ga čekaju visokonaponski roboti; liftove i platforme boje je da i ne pominjemo. Bideć upavo.

Možda će biti za nijansu lakše ući u kosmički brod i stići do Meseca. Apollo 18, kako izgleda, prilično je verna simulacija pravog leta na Mesec — a simulacija nam jedino i ostaje, stravično su tanke šanse da postanemo kosmonauti. Igra traži, nadasve, obavljanje pravih radnji u pravom trenutku, od automatskog uzletanja sa Zemlje do ručno kontrolisanog spuštanja pored Kopernikovog kratera, ili već tako negde. Uspet da bito i malo leonovani, koliko da nam život bude dramatičniji.

Sa izveštajem zaključenoj godina 1984. — ali ona Onelove — ulazi u C-64 i „amigu“. Umesto sa Velikim Bratom imadešno posla sa bezdušnim centralnim kompjuterom koji sa elektronskih cematračnica dobija podatke o nama, i samo gleda kako će nam smršati konce. Naslov: Paranoia — a i šta bi drugo?

Stripovi se i dalje preobraćaju u igre. Lenji i filozofski nastrojani mačor Garfield već se šunja kroz Big Fat Nasty Owl, a Kamerko, Kramerko, Dino i ostatak Hana-Barberine pratorijajske ekipe spremaju se za život u novoj epizodi. Još malo, pa više neće preostati nijedan strip koji nije adaptiran za kompjuter, sem možda Mirka i Slavka.

Sa mirisima iz kuhinje mešaju se zvuci pajbe iz laserskog oružja: spremaju se nove varijante pucadičkih igara. Sve je prilika da će biti teško znaoći Beudiam (u slobodnom prevodu „Kovin“ ili „Guberevac“), pole ćete imati da nadletite šesnaest neprijateljskih kosmičkih stanica i povremeno se sučavate sa osamnaest neprijatelja na jednom te istom ekranu. Cilj je: doći do šesnaestog nivoa pre i to isto vam se dugme FIRE pohaba dokraja.

V. St.

HYDROFOOL

Svevolje kao jurak igara poznat nam je i od ranije. Ovoga puta nahotice je upao u nekakav svet koji je, opet ko zna iz kojih razloga, neki anonimus ništa manje nego poplavo.

Svevolje mora da izvuče četiri čepa koje drže vodu, inače sutrašnjica mu nije baš ružičasta.

Svet u kome se nalazi sastoji se od šest spratova koji su međusobno povezani mehurima i vrtlozima (vidi mapu).

Na drugi sprat otpremate četiri svetiljke, na četvrti par čepela, svetiljku i konzervu, a na peti možete staviti ono što noćite i na prvi.

Razume se — predmete uzimate pritiskom na dugme za pucanje.

Pored toga na raspolaganju su vam i sledeća oružja: — dvozubac za eliminisanje ribe-vuka, — harpun za napucavanje bebe-kita, — kašika koja će prekratižiti žuti mezuđi, — potkovica od koje moraki konji beže kao davo od motike.

HYDROFOOL

MAP BY: BOJAN MAJER

Dabome da posao oko čepova koji se nalaze u određenim prostorijama ne bi bio posebno težak, ali da bi vam to pošlo za rukom, morate prethodno pronaći određene predmete razbacane povsuda i dovodi ih u pomenute odaje.

Kako se to radi? U sobu na najnižem — prvom spratu donesite po dve kristalne kugle i amajlice koje se nalaze u kovčezima koji se moraju otključati. A da biste ih otključali, morate smaknuti stvorenja koja stoje ispred njih.

— kijućevi čija je namena jasna. Patuljci koje možete kupiti nemaju nikakvog značaja za završetak igre, osim što donose bodove.

Pored već pomenutih neprijatelja probleme će vam stvarati kamernje, razna vegetacija i crvi koji izviru iz zemlje. A da vas ne bi nagrizla rad, ukucajte i programe objavljene u „Računarima 33“. Mapa će vam pomoći da završite ovo izvanredno igru.

Bojan Majer

Poukova mreža

Komodor 16,116,+4

Kemal Jakić

Poukovi se unose na sledeći način: nakon što učitajte program, uđite u MONITOR, pritiskom na tastere RUN/STOP i RESET. Zatim ukucajte poukovu liniju (bez strane adrese) koju navodim ispod imena programa, i pritisnite RETURN. Nakon što to učitajte, resetujte računar i ukucajte startnu adresu, u zagradu pored GHOSTS/N/OGBLINS 1

- 15F: A9 99 LDA #830 (SYS 4123)
- GHOSTS/N/OGBLINS 2 (SYS 4123)
- 102B A9 99 LDA #990 INTERCEPTOR II (SYS 4064)
- 1500 A9 99 LDA #990 ASTEROIDS (SYS 4165)
- 1045 A9 99 LDA #990 KING OF KING 2 (SYS 8128)
- 10C5 A9 99 LDA #990 CYBORG (SYS 11103)
- 2B90 A9 99 LDA #990 BOMBS HUNT (SYS 4112)
- 294A A9 99 LDA #990 GUN LAW (SYS 4120)
- 111C A9 99 LDA #990 SKY 1941 (SYS 8192)
- 25CD A9 99 LDA #990

Poukova mreža c-64

1. SKATE ROCK
POKE 5105, 165 (vreme)
POKE 9999, 165 (besmrtnost)

2. THE WIZ
POKE 3411, 173
3. FLASH GORDON 3
POKE 37576, 173; SYS 12271
Milan Zaninović

GOTOVO JE, GOTOVO

HYSTERIA

spectrum

Kao ideja već videno. Sakupljanje raznih predmeta, borba sa neprijateljima, i na kraju svakog od tri nivoa, obračun sa velikim zmajem.

Ali, dobra i teška igra koju je bez pouka skoro nemoguće proći.

Naravno da se bez oružja ne može.

Prvo koje od starta automatski poseduje je munja. Baza je iz očiju i efikasnija je samo u bliskoj borbi. Ostala stupaš i to tako što pritiskaš dole + pucanje, i to u trenutku kada se iznad slike oružja pojavljuje strelica.

Čarobni štapić služi za gađanje neprijatelja na daljinu, ali je neefikasan kad ti neprijatelj dolazi s leđa, jer moraš da se okreneš.

Treće oružje je metak koji leti oko tebe. Dakle, bezbedan si i odostrag.

Klizni rano je izuzetno korisna stvar jer letadi izbegavaš hrpu neprijatelja koji dolaze odozdo. Narobičito pomaže na kraju svakog nivoa.

Peta stvar je najubitačnija. To su tri metka koje velikom brzinom lete oko tebe i česte put bez problema.

Prilikom upotrebe oružja bitno je da obratiš pažnju na kretanje strelica koja ti pokazuje čime u čemu trenutku možeš da se boriš. Tu ima nekoliko caka koje ćeš i sam lako otkriti.



I onda da predemo na igru.

Prvi nivo se odigrava u pratoriji, drugi u srednjem veku, treći u dalekoj budućnosti. Neprijatelji su različiti, ali shema za prolaz je u principu ista.

Bitno je da pratiti šta se dešava na tri prozora u gornjem delu ekrana.

U prvom se nalazi tvoj lik koji se gubi onim tempom kako i ti gubiš energiju, ali uz pouk to nećemo mnogo brinut.

U drugom prozoru nalazi se život. Naime, u trenutku kada je gomila neprijatelja nemoguće pokidati vatrom, tvoj ratnik ima sposobnost da oko sebe stvori zaštitno polje i odmori se tako neko vreme. Štit se tako troši i kad nestane, onda više ni odbranbenog pojta nema.

Treći prozor je najvažniji. Kad god pogodiš neprijatelja iznad sebe, on ostavlja neki predmet koji ti pokupiš i tako se silika generiše do po deo. Kad je gotov, dočao si do kraja nivoa i tu ti predstoji borba sa ogromnim dvoživlim zmajem.

Možeš da nastušaš da do te potrajati — treba ga pogoditi više puta i to pravo u stomak. Ali, kad zmaj nestane — eto te u sledećem nivou.

Čak nekoliko upozorenja.

Odmah skupljaj sve predmete. U protivnom zmaj se neće pojaviti, a ti ćeš trčati beskonačno ekranom koji se stroluje.

Na trećem nivou neprijatelj koga treba pogoditi je suviše visoko, i zato obavezno koristiš oružje broj tri.

Postoji i predmet koji izlazi iz poda. Ukoliko kada preižasi preko njega pritisneš na dole, on će pomeriti strelicu kojom koristiš oružje za jedno mesto udesno.

Za besmrtnost igra RANDOMIZE USR ... treba ubaciti POKE 44588, 201.

I to bi bilo ev.

Vladimir Janković

Poukova mreža

Pokice čine čuda

Spectrumovci, pažnje! Evo i za vas dugo odevičanih poukova za nove, novije i najnovije igre. Naravno, uz poukove vam dajemo i način na koji se ubacuju u igru.

Road Runner

Program je snimljen „Spec-Mac“ sistemom, a POKE za besmrtnost glasi POKE 40806,0

Freddy Hardest



Igra se sastoji iz dva programa, koji se učitavaju „Spec-Mac“ sistemom. POKE za prvi deo glasi 64011,167, a za drugi deo POKE 61607,167. Oba pouka služe za dobijanje besmrtnosti u igru. Takođe, da biste bili u stanju da igrate drugi deo igre, morate prvo da završite prvi deo. Tako moraju da rade samo poštani igrači. Za one koji više vole da varaju dajemo POKE uz deo da pomoć moći da igraju i drugi deo: POKE 51987,0.

Hysteria

Za ovu igru obezbedili smo vam neuništivost, tj. neuništivost oklopa. POKE 44623,167 unese na način koji je standardan za igre koje se učitavaju „Spec-Mac“ sistemom.

Bride of Frankenstein

Po Jugoslaviji se „leta“ samo jedna verzija ove igre, koju je pustio „Rud“! On je, takođe, u igru uneo i POKE da vam se neprijatelji ne pojavljuju. Ukoliko želite i neesite, unesite i POKE 40478,201.

Slap Flight

Zamenite 20 liniju bejzika sledećom i postaćete besmrtni.

20 CLEAR 24999:POKE 23800,195:RANDOMIZE USR 23760:POKE 46873,0:RANDOMIZE USR 23803

Batty

I to ste dočekali. POKE za besmrtnost za najbolju verziju Arkanoida glasi POKE 48437,167:POKE 48446,167.

Action Force

Igra tek što je stigla, a već dobijate i besmrtnost, beskonačno municije, a takođe i goriva. Ovo se postize ako unesete sledeće poukove u bejzik ispred RANDOMIZE USR ... instrukcije. POKE 48617,167:POKE 48662,0:POKE 50013,0:POKE 51456,0:POKE 51641,0:POKE 53253,0.

Triaxos

Za ovu igru imamo POKE za beskonačno vreme. On glasi 31724,0, a ubacuje se na standardni način.

Jack the Nipper II

Ukoliko otkucate i startujete ovaj kratki bejzik program, a zatim učitate i celu igru, po njenom startovanju imaćete beskonačno života.

10 CLEAR 24575:LOAD "" CODE:LOAD "" CODE 20 POKE 43251,0:PRINT USR 34240

Ukoliko želite da imate odredeni broj života, unesite POKE 43251,0 unesite POKE 34866,0 (0<x<256).

Ball Breaker

U bejzik ispred instrukcije RANDOMIZE USR ... unesite sledeće poukove: POKE 35840,0:POKE 35904,0:POKE 39844,0.

Mystery of the Nile

Program se učitava „Spec-Mac“ sistemom, a POKE se unosi na gore opisani način. POKE za besmrtnost je 55471,17.

Thundercats

Da biste postigli besmrtnost, u bejzik unesite ispred RANDOMIZE USR ... sledeću naredbu: POKE 556-122-156,36.

Exolon

Sistem učitavanja je „Spec-Mac“, a POKE za besmrtnost glasi 38120,201. Nadamo se da ćete za uz

pomoć mape koju amo već objavili, moći da završite ovu izuzetno dobro urađenu igru.

Ghost Hunters

Besmrtnost se postize zamenom postojećeg bejzika datim:

10 LOAD "" CODE 23296:POKE 23354,201:RANDOMIZE USR 23296:POKE 214+256-216,0:RANDOMIZE USR (69+256-178)

Shadow Skimer

Igra je izuzetno dobro urađena, a takođe je i teška. Zbog toga vam dajemo POKE za 255 života (POKE 4788,255).

Star Runner

Unesite sledeći POKE i imaćete besmrtnost u igri. POKE 49560, 167.

Indiana Jones



Da, istina je, iako je, kako se mnogo puta pokazalo, besmrtni, ipak vam dajemo POKE za dobijanje beskonačno života i u igri. POKE glasi 33948,0, a unosi se na način već ubičajen za igre snimljene „Spec-Mac“ sistemom.

Wizball

Ovu izuzetno lepu grafički urađenu Oceanovu igru verovatno nećete moći da završite ukoliko ne posedujete besmrtnost. POKE 37052,0 ubacuje se na sledeći način. Prvo učitate originalni bejzik program u MULTICOPY, a onda mu skinite autostart opcijom Abort. Zatim učitate tako „abortion“ program i zamenite mu 20 liniju sa:

20 CLEAR 24999:POKE 23800,195:RANDOMIZE USR 23760:POKE 37052,0:RANDOMIZE USR 23803

Startujte program i pustite Wizball da se učitava od slike. Imaćete besmrtnost.

Renegade



Spec-MAC sistem učitavanja. POKE 41048,195. Besmrtnost.

Through the Trapdoor

Postojeći bejzik zamenite sa sledećim. Startuje ga i imaćete besmrtnost.

10 CLEAR 26998:LOAD "" CODE 63000:POKE 63041,201

20 RANDOMIZE USR 63004:POKE 47492,0:RANDOMIZE USR 34200

ATV Simulator

Ponovite isti postupak kao i u prethodnoj igri i postaćete besmrtni.

10 CLEAR 25400:LOAD "" CODE:POKE 65531,201

20 RANDOMIZE USR 65501:POKE 60250,0:POKE 57318,201:RANDOMIZE USR 54960

Bata Sprajt

GOTOVO JE, GOTOVO

Kako ubaciti POKE POKE, POKE, pokica i gotova igrice

U većini opisa igara i našoj rubrici „Paukova mreža“ često se pominje jedna za mlade igrače magična reč **POKE**. Šta je to **POKE** i kako ga ubaciti?

POKE je jedna od standardnih naredbi programskog jezika bezik koja na određenu memorijsku lokaciju smešta određenu vrednost. Sintaksa ove naredbe glasi:

POKE adresa, vrednost

Zbog čega je ovo važno. Većina igara za kućne računare pisane je u mašinskom kodu. U toj šumi bajtova negde je smeštena i rutina za puštanje života. Broj života je u najjednostavnijem slučaju, obično zapisan u nekoj memorijskoj lokaciji. Kako se ove lokacije otkrivaju sasvim je drugo pitanje. Za igrača je važno da se on može jednostavno promeniti naredbom **POKE** iz bezika.

Da ne ostane sve samo na teoriji dajemo vam program iz koji ćete pomoći moći iako da ubacite **POKE** u sve igre animirane **Spec-Mac** sistemom.

Spec-Mac

„Spec-Mac“ sistem se može lako prepoznati, jer se kod njegove igre učitava iz .pef, odnosno šest delova ukoliko ima dva skrinna. Prvi deo predstavlja bezik, zatim sledi skrin, koji je dugačak 6916 bajtova. Nakon toga dolaze dva dela dužine 20000 i 20536, a nakon toga jedan kraci deo dužine 1704 bajtova. Ukoliko ima šest delova, onda je prvi skrin dužine 6912, a ispred ovog zadnjeg, relativno kratkog dela, dolazi još jedan skrin dužine 6916 bajtova. Sada, kada smo naučili da prepoznamo „Spec-Mac“, biće nam jednostavno i da u njega unesemo pokove.

Pokove se obično unose u memoriju između 25000 i 65535. Ukoliko se unose u deo memorije između 25000 i 44999, potrebno je ubaciti **POKE** na prvi način, a ukoliko se unosi u memoriju između 45000 i 65535, **POKE** se unosi na drugi način.

Prvi način

Otkucavate ovaj kratki bezik program, a zatim ga startujete. Učitajte deo igre dužine 20000, unesite pokove, a zatim snimite **POKE**-ovu verziju igre.

10 PAPER 0: BORDER C: INK 7: CLEAR 24999

30 FOR A=23292 TO 23308

30 READ X: POKE A,X: NEXT A: RANDOMIZE USR 23296

40 DATA 221, 33, 168, 97, 17, 32, 78, 62, 255, 55, 195, 86, 5

Kada se učita deo igre dužine 20000, iz bezika unesite pokove, a zatim snimite ovaj deo na sledeći način. Otkucavajte **SAVE** '111' CODE 25000, 20000. Prilistite **ENTER**, i kada vam se pojavi poruka Start type... pritisnite neki taster. Sačekajte da prvi put prestane linije, pa tek onda pritisnite animiranje na kasetofonu. Na ovaj način ste snimili ovaj deo igre.

Drugi način

Drugi način je analogan prvom. Jedina razlika je što sada treba uneti drugi bezik program, a pri animiranju pokovane verzije koristite sledeću komandu: **SAVE** '221' CODE 45000, 20536.

10 PAPER 0: BORDER C: INK 7: CLEAR 24999

20 FOR A=23296 TO 23308

30 READ X: POKE A,X: NEXT A: RANDOMIZE USR 23296

40 DATA 221, 33, 200, 175, 17, 56, 80, 62, 255, 55, 195, 86, 5

I šta sada?

Dobro pitanje zahteva i dobar odgovor. Ukoliko ste **POKE** uneli na prvi način, učitajte originalnu igru do dela pošte skrinna, zatim učitajte ovaj pokovani deo igre, a nakon toga i originalnu igru od drugog dugačkog dela, tj. preskočite prvi dugački deo koji sledi nakon slike. Ako ste **POKE** ubacili na drugi način, uradite isto, ali ovaj put preskočite drugi deo.

Bata Sprajt

DEFENDER OF CROWN

O ovaj igri je mnogo pisano, ali ne i dovoljno da se postigne sličnost rešenja.

Podrazumevano, pri tom, da igrače pošteno i startuje samo sa 20 vojnika, dakle isključena je upotreba piratske verzije „Mega defender“ koja osnovni igri oduzima svaku draž.

Ovaj prikaz važi za ispravnu verziju za C-64 koja postoji u zemlji (doduše i ona ima jedan mali bag, ali o tome dočiruje).

Cilj igre je jednostavan — treba osvojiti tri južna normanska zemlja.

Bitno je da uveštate tri opcije bez kojih rešenja nema.

GO RAIDING se svodi na mačevanje koje nije teško — blokadom potiskujete protivnika, a onda u povoljnu zadržite udarac.

TURNAMENT zahteva nadasve oštro oko i preciznu ruku — kada protivnički vitez iza na vas, morate ga kopljem tačno pogoditi u visini znaka „x“ koji se nalazi na njegovim grudima.

SIEGE — morate veoma precizno gađati iz katapulte, jer svaki pogodak znajomno umanjuje snagu branilaca zemlja.

I još jedno upozorenje — pre nego što krenete, dobro proučite mapu i zapamtite ko su vaši saksonski saveznici, a ko normanski neprijatelji.

I onda da krenemo: **PRVI POTEZ**: prebacite jednog vojnika u armiju i zaposednite teritoriju koja se graniči sa sva tri moguća strana mesta.

DRUGI POTEZ: zaposednite teritoriju koja se nalazi južno od ove.

TREĆI, ČETVRTI I PETI POTEZ — opcijom **GO RAIDING** objačinate vam tri normanska zemlja, prvo srednju, a potom pratile da li je neprijatelj zaposedao nove teritorije. Ako jeste, napadajte, ako nije — nemojte, jer je kupio armiju i njegov trezor je prazan. U svakom slučaju, jedan od njih zero je za napad.

Već između četvrtog i petog poteza postoji mogućnost da kompjuter reaguje, pa vas pozove na turnir ili vam ponudi mogućnost da spasete saksonsku princozu i potom se obratite njome.

U prvom slučaju, pod uslovom da ste savladali opšti **TURNAMENT**, borite se za zemlju (iako ćete procenti koje teritorije su vam nužne), ili da zauzimate normanski napad, ili zaštitite svoje saksonski prijatelje. Imate priliku da se borite tri puta, dakle da osvojite tri zemlje i pošle toga ste šampion.

U drugom slučaju spasavate princozu i ženite se (postoje tri princoze — Katarina, Rozalinda i Ana). Poslednju ne dirajte, jer je to oraj mali bag i program se blokira. Žanidva vam donosi saksonske teritorije i 20 vojnika u mašinskom zamku vašeg tasta.

SESTI POTEZ je veoma važan. Kupite armiju, i to najmanje pet vitezova, dva katapulta, a za ostatak vojnika. Sada ste spremni za prvi odlučni napad.

SEDMI POTEZ — postoje dve mogućnosti. Ako se jedan normanski osvajač kretao i zaposeo novu teritoriju (dakle, vojnika mu je vam mašinskoj zamki) sa čitavom vojskom krenite u opasdu, pod uslovom da možete da ga preko savršenijskih teritorija direktno napadnete. Obavezno pozovite pri tom i Robinu Huda u pomoć.

Ukoliko nije moguć direktan napad, zaposednite teritoriju koja vam to omogućava, ali pazite da neprijatelj ne ugrozi vaš mašinski zamak, koji pri tom ostaje bez odbrane.

OSMI POTEZ — ukoliko niste uspeeli u sedmom, pod istim uslovima napadajte u osmom potezu. Najbolje je ako to bude srednji normanski zamak, ali nije strašno ako zaposednete i desni.

Ukoliko pobedite, a trebalo bi, ovim je kritična faza igre prošla, i u 90% slučajeva pobeđa je vaša, s tim da pažljivo i promućurno igrate.

Ne taktizirajte previše. Pokušajte da sredite i drugog neprijatelja. Treći tada postaje previše slabšan da bi vam se odupre.

Naravno, u svakoj igri postoji niz varijacija. Kompjuter ponekad malo odstupa od sheme, ali se u suštini sve svodi na isto. Prema tome, smatramo da smo vam otvorili vrata da časno ponese titulu kralja Engleske.

Dorde Perović

ARCADE CLASSIC



U ovom „Firebirdovom“ ostvarenju ima četiri igre: **SPACE INVADERS** — svoj svemirski brod pomerate levo-desno i uništavate neprijatelje koji vam se sve više približavaju. Kao zaštitu od napadačkih metaka koriste vam četiri zaštitna krila. Međutim, pucajući metak može ih uništiti (zato morate biti oprezni). Glavno obilježje ove igre jest je da ona u potpunosti prebacena s automata. **ASTERIODS** — svemirsku letelicu pomerate levo-desno i gađate asteroide. Da biste ih uništili, morate ih tri puta potkočiti. Naime, nakon prvog pogotka raspadne se na dva dela, a nakon toga i to dva će se podeliti na još manje, da bi tek u trećem potkočenju bili uništeni.

SNAKES — ovde mogu igrati dva igrača. Cilj je da se protivnički zmiji stane na rep, ali to treba ostvariti u određeno vreme. Poverljivo se na ekranu pojavljuje vođa koje pokušate, jer se time vaša zmija produžuje.

SPACE WAR — igra također napravljena za dva igrača. Na početku birate opcije koje su za vas i vašeg protivnika najpovoljnija, a zatim možete početi s igrom. Cilj vam je da uništite protivnikovo vozilo. Međutim, ometa vas neka vrsta magnetna koji se nalazi na sredini ekrana i privlači oba svemirske letelice.

To je sve. Prijatna vam zabava!

Milan Zaninović

MATCH DAY II

Sportske igre, naročito fudbal, česta su tema kompjuterskih igara. Uveć jedne prijetne novosti za ljubitelje takvih programa: legendarni Match Day doživio je svoj zvanični nastavak Match Day II.

I odmah da kažem: **ODLUČNO!** Nove mogućnosti koje ova igra nudi prikolave su me za ekran „duga“.

Match Day II ima čak tri menija sa mnoštvom opcija. Prvi, glavni meni omogućava:

- igru prvog ili drugog igrača,
- igru dvojice igrača protiv kompjutera,
- kup taktičenje,
- lišisko taktičenje.

U drugom meniju možete promeniti — imena timova (domaći pravi zvani „Rudy“ učinio je, pored različitih programa, i nešto loše (7) na tabeli se nalaze sve sami domaći prvotilaci).

— kontrole za ova igrača (ve vrste palice, tipke po želji),

- boje,
- taktiku (ofanzivna igra ili „bunker“).

Treći meni, ujedno i najbolji opcijama nudi:

- zvuk,
- vreme igre,
- šutvi (različiti tipovi, različite vrste, najbolje je odabrao svi).

— pomoć računara pri igri,
- nivo igre kompjuterskog suparnika,
- upravljanje golmanom (vrši ga računari ili vi lično).

Tokom igre pazite na sledeće: U gornjem delu ekrana nalaze se časovnik i merač jačine udarca (ovo zavisi od treće opcije trećeg menija). Ako dugne za pucajući pritisnete u trenutku kada merač pokazuje 3, šut će biti najjači, a ako u tom trenutku pokazuje — 1, izvešćete udarac petom, iako će vam se učiniti da je suviše spora, igra će sigurno zavesti vaše društvo.

Bojan Majer

GOTOVO JE, GOTOVO

TAI PAN

Trgovac, švercer ili gusar — pitanje je sad?

A opet ove trileme se svodi na isto — plovidbi dalekim morima, vaš je cilj da steknete imetak od ravno 8.000.000 dolara i postanete tai pan ili najbolji od najboljih.

Najbitnije je, pri tom, da logički razmišljanje čini prvi kvalitet ove privlačne igre koja se odvija na gornjoj polovini ekrana.

Dole su ikone (kupovina, prodaja, podizanje-spuštanje predmeta, snimanje i učitavanje pozicije sa trake ili diska, zavali od verzije koju imate), zatim kalendar, podaci o gotovini koju posedujete i o ukupnom novčanom stanju, kao i o posedovanju tojgase.

Tokom igre nalazite se na razne ličnosti (pljanke, policajce i švercere) i objekte:

ORUŽNICA (Armory) — ako odlučite da gusarite, ili pak da dobio te nagradu za uhvaćenog piratskog vođu, ovde možete nabaviti topovsku dulad ili municiju za muskete.

BANKA (Bank) — Ne, ovde nemate otvoreni račun. Bankar se bavi veoma unosnim poslom: prodajom brodova. Postoje tri vrste koje se razlikuju u ceni i kvalitetu: **lorcha** (mali, brzi brodic, pogodan za prevoz manjih količina tereta i robe kupijene od švercera) od 150.000 dolara, **clipper** (veći ali sporiji brod, službi za trgovačke poduhvate) od 250.000 dolara i **fregate** (snažni, brzi brodovi, služe u ratne svrhe) od 400.000 dolara.

ZATVOR (Gao) — ako kojim slučajem uderite predstavnika zakona (morka sa sabljom), bićete zatvoreni, oduzete mesec dana, a onda ćete opet biti na slobodi. Ukoliko vas u istom gradu zatvore četiri puta, postajete rob i igra se automatski završava.

KRČMA (Inn) — veselo društvo za šankom možete uzeti za posadu, ili im se možete pridružiti. Inače, posadu (besplatnu) možete nabaviti i na drugi način: pronađite motku i batinajte pijane prolaznike.

JAVNA KUĆA (Ladle's house) — ulazak ovde je čisto bacanje novca.

RESTORAN (Restaurant) — ovde svratite kada ste gladni, ili kada želite dobiti nešto novca igrajući jednostavnu igru na sreću: uložite pare na jednu pločicu i sačekajte. Ako vaša pločica prva „stigne“ do cilja, dobijete nekoliko puta više novca nego što ste uložili.

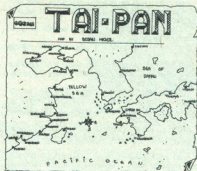
SKLADIŠTE (Supplies) — bez mape, teleskopa i hrane ne putujte nigde. Pazariate ih ovde. Više od dve kutije kiocpe ne kupujte nikada, jer pri ulasku u luku gubite sve.

BAKALNICA (Warehouse) — pored kockanja i šverca, kao i gusarenja, novac možete zaraditi i mirnom trgovinom. Robu kojom ćete trgovati kupite ovde. Na raspolaganju su vam: čaj, svila i žad.

Na početku idite u restoran. Na postavljeno pitanje odgovorite odlično. I naći ćete se u pomoćnom sobičku gde će vam Jin Gao pozajmiti 300.000 dolara. Ako opet uđinite isto, moći ćete i igrati već opisanu kockarsku igru. Čestim snimanjem i učitavanjem podataka na ovaj način možete doći do veće sume novca. Puť vas dalje vodi u banku. Kupite brod, skupite posadu, opremite se i se startne pozicije sliđite dolje. Učitajte dio igre koji se bavi plovidbom.

Sada se srećete sa novim ikonama:

- 1. MAPA** — Pokazuje sve gradove, kao i uco rtane kurseve kojima valja ići.
- 2. KOMANDE** — ako ikona pokazuje kormilo, svaki pritisak na pucanje znači ispaljivanje saive na obližnji brod. Ako ikona pokazuje top, pritisak na pucanje neće značiti baš ništa. Inače, ikone dozivate tipkom SPACE.
- 3. JEDRA** — ikonom određujete visinu jedara. Ako



GUANGZHOU W L L S I R + + + + + + A G + + + + + + + + P	MACAO + + + + + + A G W + + + + + + + + P	TOKIO B L I A G W + S + + + + + + + + + + P	OKINAWA R + + + + A G W + S + + + + + + + + + + P
FOSHAN B + L I + A + G W + S + + + + + + + + + + P	SHENZHEN A + G B B W I + R + + + + + + + + + + P	YINGKOU B W L L S I R + + + + + + + + + + P	WEI HAI B A G + I + W + + + + + + + + + + + + + P
XIAMEN R L L I S W + B + + + + + + + + + + P	FUZHOU B W L L S I R + + + + + + + + + + P	LU DA A + G R S W I + B + + + + + + + + + + P	YILONG R L L I S W + B + + + + + + + + + + P
HANGZHOU B W L L S I R + + + + + + + + + + P	SHANGHAI B L L S I R + + + + + + + + + + P	CHAJU DO A + G R S W I + B + + + + + + + + + + P	NAGASAKI W L L S I R + + + + + + + + + + P
WUXI A + G W L I + B + + + + + + + + + + P	SUZHOU R W L L S I R + + + + + + + + + + P	GUINGRAG W L L S I R + + + + + + + + + + P	SASEBO R L L S I R + + + + + + + + + + P
ZHENJIANG B W L L S I R + + + + + + + + + + P	YANTAI B L L S I R + + + + + + + + + + P	HAIKOU A + G B S W I + R + + + + + + + + + + P	KITAKYLSO B L L I A G W + S + + + + + + + + + + P
NINGBO B A G W L I + B + + + + + + + + + + P	TIANJIN R L L S I R + + + + + + + + + + P	TANGSHAN B W L L S I R + + + + + + + + + + P	LUSHUN B A A G W L I + B + + + + + + + + + + P
KANAZANA W L L S I R + + + + + + + + + + P	KUANGSU R W L L S I R + + + + + + + + + + P		

- W - WAREHOUSE
I - INN
A - ARMOURY
L - LADIOS
P - PORT
S - SUPPLY
G - GAOL
B - BANK
R - RESTAURANT

MAP BY: LAW & IVAN

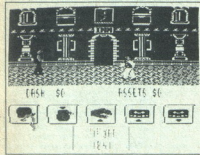
GOTOVO JE, GOTOVO

stoje, treba biti najmanja, ako maneirite biće na sredini, a punim jedrima plovite na otvorenom moru.

4. **NETRANA** — kada se na komandnoj tabli pojavi poruka da je posada gladna, odmah se prebacite ovdje. Ako ih ne budete dobro hranili, mornari će pomrnuti ili će se pobuniti. Da biste ovo sprečili, još pre bilo kakve poruke dajte im obroke, tako da će važiti ono „Judit i ovice na broju“.

5. **TELESKOP** — pokazuje vrstu ili pripadnost broda koji je pred vama.

6. **NETRANAK** — stralica pokazuje pravac veta. VAŽNO: ako izadete van okvira mape, vetar vam neće biti nakičovan.



Priklom plovitve treba voditi računa i o tajfunima, koji duvaju kako im defini, a najviše ih ima u jugozapadnim delovima mora.

Vaš dalji put zavisi i od novčanog stanja. Ako niste mnogo potrošili, krenite u Nagasaki (na severnoj obali Kine?) ili Sasebo. U tamnijim prodavnicama žad stoji 65.000 dolara, a u jugozapadnim lukama — 66.000 dolara. Tu svilu možete kupiti za 41.000 dolara, dok ćete je u Japanu naći za samo 35.000. Švercer iz Foshana, Macaoa i Guangzhoua svoju robu nude za 54.000 dolara, a Japancima ćete to isto uvaliti za 60.000.

Ako se naručujete, pazite na sledeće:

Trovački brodovi najčešće se kreću žutim morem i kursem između Shenzhena i Japanskog luku. Japanske obale vrede od pirata. Kada puca na neki brod (vidi ikonu „komande“), ako se menate: vidite top i okean pred vama. Ojdednom, tuđa počne da plove brodi, i vi pokušavate da ga izrešete! — ovo je vrlo silčno „Beach headu“, prastarijskoj igri. Zašim neprijatelji skaku na vaš brod, a vi ih toplo dočekujete musketom i sabljom. Ako pobedite — tovar i mornari vama. Ako izgubite celu posadu — ništa, mrtvi ste i sve ponovo počnje.

Siguran sam da će vas ova izvršna igra držati uz kompjuter bar mesec dana.

Srećno!

Bojan Majer

Poukova mreža

„Spektrum“

RENEGADE

Poke 410646 — neranljivo
Poke 343720 — vreme
Poke 403990 — životi

Vladimir Janković

SCARY MONSTERS

„Scary Monsters“ je igra sa originalnom temom i fantastičnom muzikom.

Cilj je stedeći: uništiti šest čudovišta koja lutaju izvešnim ostrvima.

Uz napomenu da možete igrati sa tastaturom, pod uslovom da definišete komande, život vam zagarovaju dve stvari:

- profesionalna smetala kojih ima kao pečuraka pose kiše,
- činjenica da svako čudovište unistavate posebnim oružjem.

Igra se odvija u gornjem delu ekrana, a u donjem se nalazi broj vaših života (u početku ih imate tri), broj bratova (pet) i slike svih šestoro čudovišta. Kako kome smrtni konce njegova slika nestane, a pojavi se simbol vešnih lovišta u koja ste ga uputili.

Obratite pažnju i na donji levi deo ekrana — tu je predstavljeno oružje koje trenutno nosite.

Za vešću koristite otrov, za munju krak, vukodlaka ubija pištolj sa srebrnim mecima (nešto mi poznato zvuci), vampira glogov kolac i čekić (a tako ovo), Frankštajnska baklja, a čudovišta u donjem desnom uglu ekrana.

Oružja i čudovišta su raspoređena u osam zgrada (tamkovi, crkve, piramide) i kada god uđete u bilo koje od ovih zgrada jurite vas razni vampiri, vedice, duhovi, zombiji, pa čak i luke (!!!). Eliminirajte ih vatrenim loptama (pritanite FIRE), a ukoliko pristanete nadole, iskoristite jednu od barjola koja sve pokretne neprijatelje pretvara u obične (uff!).

I pazite da se ne izgubite u lavirintu! Sve u svemu, dosta laka igra, zar ne?

Vladan Đukanović

FREDDY THE HARDEST

Ovoga puta je Tvrđi Fredi upao u zaista veliku nevolju. On treba da kao specijalni agent pobjegne letelicom kojom se srušilo na nepoznati i iznad svega negoteljivju planetu. Na planeti postoje tri vrste neprijatelja: sa onima koji lebde u vazduhu i idu na tebe ne treba da se boriti, već je dovoljan jedan hitac iz pištolja. Sa onima koji idu ka tebi i zakleste se kada puneš na njih možeš na dva načina: ili da ih zaobišćeš ne obrađujući pažnju na njih, ili da se bježiš t: strana-pucanje je udarac kursum, samo pucanje je udarac nogom. Treća vrsta neprijatelja su kugle koje lete ka tebi u visini plave. Njih sruđeš gore pomenutim udarcima.

Najvažnije su sobe sa slovom „N“ na patosu, jer na mesto ostavljaju trake koje skupljaš po bazi. Na ostalim kompjuterima dobišće neke korisne informacije bez kojih ne možeš da završiš igru. To su kodovi Rad (crvenog), Green (zelenog), White (belog) i Blue (plavog) kapetana. Kompjuter se aktivira pritiskom kursuma. Mnogi kompjuteri će pri pokušaju da ih aktiviraš odgovoriti „OUT OF ORDER“ i od njih diži ruke do daljnjeg. Kada ostaviš traku na mesto označeno sa „N“ i aktiviraš kompjuter, dobišće poruku: „BOJA“ ship full, što znači da je brod napunjen i spreman za poletanje. Kombinuj tako da boja broda i boja kapetana budu iste.

Kada dođeš na neku platformu za pokretanje, njome upravljaš tako što pritiskom nagore ili dole određuješ

DEATH WISH III

U gradu frka na veliko.

Toror nad mirnim gradovima. Na ulicama sve sama glog, bitanga do bitanga, pakjani koji tu i tamo na javitve daje tek ponekoga sklepaču, dame sumnjivo morala, i poneka slučajno zalutala zlehdna bakica.

Šta vam drugo preostaje nego da sredite vode svih gangova i u vaš mali gradit vratite red i mir?

Možemo vam reći da ima ukupno pet bosova, i svakoga redom morate da sredite namrtno ravno peli puta. Sređom, kad jednome dokašate, nastaje mala pauza dok se novi ne pojavi, pa možete i da predahnete od silne tabačine.

A kako igrati? **Levo-desno** — krećete se normalno. **Gore-dole** — okretanje za 90 stepeni.

Taster N — na mapi u donjem delu ekrana vidite položaj municije, a i mesta gde se nalaze befovi bandi.

Taster C — služi za promenu oružja. Na raspolaganju su vam pištolj, puška, puškomitraljez i bazuka.

ENTER — služi za ulazak u zgradu, naravno kada ste pred vratima.

Taster W — daje vam mogućnost da otvorite vatra iz zgrade, ali kada ste kraj prozora.

Korisno je još znati da se u donjem delu ekrana, pored mape i oružja, nalaze i kompas, tabla sa obavestjenjima i crtetav vešćeg pancira koji neprijatelji neumogu „bude“.

A kada u gradu zavladate red i mir, očekuju vas čestitke i zaslužan odmor na Bahamima.

Bojan Majer

pravac kretanja. Takođe postoje prolazi u zidovima u koje ulaziš pritiskom nagore. Onda ulaziš u mračnu prostoriju u kojoj vidiš samo svoje oči i oči nekog drugog zalutalog putnika kako trepu u crkni. Veoma je teško pratiti kretanje kroz prolaze, ali su oni na mapi obeleženi pa neće biti problema. U početku će ti predstavljati problem i hvatajke koje se nalaze na platformi, a koristiti ih kada nedostaje deo poda.

Problem je onda kako se boriti sa neprijateljima kada su nam ruke zauzete? Odgovor je: pritiskom nagore — nogama. Kada hoćeš da pucaš, pristanite dole i pucanje. Municija se veoma brzo troši, ti imaš samo nekoliko metaka za jedan život, a stanje municije prikazuje slika pištolja koja se natizi u malom donjem prozoru.

Kada posle svih peripetija skupiš sve trake, ubaciš ih u kompjuter i napuniš sve brodove, ostaje ti još da odeš, jedinom od prolaza u zidu, do mesta gde stoje brodovi. Kada dođeš gore, stani na sredinu ekrana i priteni tipku gore. Na ekranu će se ispisati: Nuclear cell (not ready/ready) i Hyper space (not ready/ready). Ako na obe stavke bude pisalo „ready“, znači da je brod spreman za poletanje, ali mu još nedostaje jedna silnica. Treba ispisati kod određenog komandanta, a to saznajemo po boji broda, ako smo prethodno zapisali kodove svih komandanta. Posle toga ostaje ti da posmatraš manevrisanje svog broda, koji te posle uspešno izvedene akcije vraća kući. Autori ove igre obavešavaju i nastavak. Ako si voleo Dana Smolera, ni Tvrđi Fredi te neće ostaviti ravnodušnim. Umalo da zaboravimo: nemoj se kretati izvan okvira mape jer ćeš se izgubiti.

Vladimir Janković



GOTOVO JE, GOTOVO

Poukova mreža

„amstrad“

Mladen Anđić

Možda želite da započnete svoj život na području za računalne aplikacije? Može biti korisno da znate nešto o „amstrad“ programima, jer vam oni omogućuju da razvijate i testirate svoje programe. Ovo je popis najpopularnijih i najkorisnijih programa koji su dostupni na našem tržištu. Možete ih pronaći u svakom dobrom trgovcu računarskim materijalom.

30 FIGHT
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

AMAGROTE
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

ARRASNO
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

ASUP WIEDERHORN MONTY
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

BACK TO REALITY
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

BALLBREAKER
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

BOBBY SEARING
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

BOULDER DASH 3
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

BRIAN BLOODAGE
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

CAMELOT WARRIORS
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

30 LCD 2870/10 MHz
30 FLOPPY 2870/10 MHz
30 HARD 2870/10 MHz
30 KEY 2870/10 MHz
30 MOUSE 2870/10 MHz
30 DATA 2870/10 MHz

CHRONOS
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

CLASSIC AXIENS
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

CLASSIC INVADERS
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

COMAL STALONE
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

COPOUT
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

COSMIC SHOCK ABSORBER
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

CRYSTAL CASTLES
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

DAN DARE
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

DEATH WISH 3
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

DONKEY KONG
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

DRAGON'S LAIR
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

DYNAMITE DAN II
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

ELEVATOR ACTION
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

ENDURO RACER
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

EXOLON
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

FLASH
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

FROST BITE
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

FUTURE KNIGHT
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

GALAXIAN
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

GHOST HUNTERS
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

GHOSTS 'N' Goblins
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

GUNNER RIDER
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

GORBAF EL VIRINGO
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

HALLS OF GOLD
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

HERBERT'S DUMMY RUN
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

HIGH RISE
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

HYDROFOOL
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

INTO THE EAGLES NEST
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

KAT-TRAP
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

KNIGHT SHADE
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

KRAKOUT
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

MAG-MAK
10 MODE 1
20 CPU 2870/10 MHz
20 LCD 2870/10 MHz
20 FLOPPY 2870/10 MHz
20 HARD 2870/10 MHz
20 KEY 2870/10 MHz
20 MOUSE 2870/10 MHz
20 DATA 2870/10 MHz

ON THE TILES

Konačno, igra u kojoj djetinjstvo možete odmoriti od napanucavanja veveričara u „komando“ stilu. Ovdje ste u ulozi mačke koju s jednog kraja grada morate dovesti u drugi. Sakupljajući riblje kosti (ni mački život nije lak), morate proći ukupno šest nivoa (toliko ukupno ima ova igra).

Na prvom nivou potrebno je sakupiti četiri ribe, na drugome pet, dok u ostala četiri između šest i deset ribi.

Dok prolazite gradskim četvrtima, vrebaju vas različiti neprijatelji:

SOVE su najopasnije jer se pojavljuju onda kada se najmanje nadate, a ako ih na vreme ne sreditte izgubite čete život.

JEZEVE treba prekositi, ili pljunuti (SPACE), a moguće je i ubiti.

KABE I MUHE nisu opasne po život, ali odzimaju mnogo energije.

Na kraju uloge stanuje STARI GOSPODIN. On ne podnosi mačke, i ako se popadne na njegovu kuću, pogodi će va vizom, i ode još jedan život.

Na ulicama se pojavljuju i MIŠEVI. Nastoje ih uhvatiti jer donose poene.

Na početku igre (kako i mački doljuke) imate 9 života, dok potrebna energija iznosi 9999 energetskih jedinica.

Možda će vam problem predstavljati mači neobične komande, ali vremenom ćete se navikati. Mačka je ajajno animirana, grafika je izvrsna.

Milan Zaninović

BUBBLE BOBBLE

U ovoj igri ste u ulozi brontosauka kojeg treba provesti kroz „samo“ 100 nivoa. Međutim, bejajati počine kad uvidite da je on blijedoj, pa morate preporučiti prevaranti u vode, jer mu jedino to omogućava da pređe na sledeći nivo. Pored toga, izgleda da je poeo pun sanduk sapuna, pa stoga ispušta balone koji su mu jedina odbrana od napadača. Da biste sredili naplate, oprta, napunjeni balon morate probušiti. Završite, oprta

može izdržati samo oko 10 sekundi (zato morate biti brzi).

Naizlazi čete na razne predmete koji donose različita pobojanja i nagradne bodove:

- kilobrani vas prebacuju tri nivoa napred
- patike vam omogućavaju brže kretanje
- munje uništavaju neprijatelje
- sprej vam omogućava da steknete bonus od 100.000 poena.

Ako na 6. nivou odmah uništite sve spodobе, pojavit će se veliki dijamant koji donosi ogroman broj bodova (ako ga pokupite). Morate paziti na vrijeme koje je ograničeno. Naime, ako puno odugovlačite, pojavit će se natpis „HURRY UP!“, a potom i sluga strašnog gospodara tame, od kojeg vam nema pomoći. Sakupljanje balona i nastojanje da sastavite riječ EXTEND, jer vam to donosi nagradni život. Kad nakon mnogo peripetija stignete do 100 nivoa, ugledat ćete gospodara tame, kojeg morate mnogo puta pogoditi da bi ste ga uništili i završili igru.

Prvi nagradni život dobivate na 30.000, a kasnije na svakih 70.000 poena.

Milan Zaninović

FLUNKY

Ovoga puta nalazite se u ulazu slugu na jednom engleskom dvorcu. Zamak je podeljen u tri glavne prostorije, od kojih svaka ima desetak soba, sem treće koju ču pomenuti na kraju. U ove prostorije morać izvesti dve misije da bi dobio autograme koji su potrebni za zarizanje lopte. Ako uđete u jednu odaju, nećete moći da izađete iz nje sve dok ne dobijete dva autograma, tj. ne izvesti dve zadatke. Svaki zadatak je ograničen vremenom, koje je veoma kratko, pa će ti trebati mnogo vežbe za kompletiranje igre. Najviše će ti smetati stražari koji će te juriti čim se neki od stanaara zamka požali na tebe. Kada stražar bude u stoji ravnaj sa tobom, pukaće i ode jedan od šest žvota. Životi su veoma lepo prikazani u obliku silike sa šest mađijskih "Takođe, ne smeš naleteti na stražara, jer da li ti Flunkya koštati života.

Na početku igra pomeranjem ručice na pruzi biraj jedan od pet evropskih jezika. Posle toga priteni dugmećna sa desne strane i kreni u igru. Pošto prvo u drugu prostoriju i kreni desno. Tamo nalaziš na veoma smešnan prizor: Čičica jea dnevno koga i zamijaš da je na Polo terenu. Njegova naredba je: "Flunky, get my polo balls" (Flunki donesi moje polo-lopote). Za ovo misliju ti je potrebno dosta veštine, koju ćeš steti praksom. Uzmi federe koje se nalaze u toj sobi i kreni u sobu desno, gde ćeš primetiti lopticu kako skaka. Postavi feder tako da ona padne na njega i odbije se. Feder, naravno, držiš u rukama. Juri lopticu i odbije je sve dok ona ne ode preko levog zida. Na početku će se puno puta dešavati da te loptica pogodi u glavu. Ito dovodi do gubitka jednog života, ali posle će ti izaći.

Kada prebačiš prvu lopticu preko zida, idi u sobu desno pa tu ponovi, sve dok ti kod dedice ne ostaviš tri loptice. Onda spusti federe, uzmi jednu lopticu i prinesi mi je. On će reći: "Place the ball for me", a ti mu postavi lopticu tako da može da je udari štapom. Ali, na tvoju nesreću, loptica koju on baci pogodi stražara u drugoj sobi koju naravno, počinje tebe da juri. Beži skoro desno dok se stražari koraci ne udalje, što znači da je zaboravio na tebe. Kada starić ispuca sve tri lopte, stavi slovo "A" ispred njega (sa kojim si počeo igru) i priteni lopticu. Ostaće ti samo da uživaš gledajući izvanredno programersko rešenje. Posle toga imaćeš ponovo šest žvota, vreme će biti na ulazu, a u mesto njega na broju će biti ispisani tvoji poeni. Sada možeš preći na drugu misiju.

Kreni sašvim levo koje te posmatra sve vreme sa karnina. Kada ga pogodiš, ono će poleteti gore. Sada te čeka veoma složen zadatak. Treba da pucajući ispred, iza, ili u njega naterati krljato stvorenje da dođe u sobu uvekivane dame. Čekaj da "perika" dođe tačno iznad prinovezine glave, a onda ispalite hitac i — gde žudal Perika se poslužno vraća na svoje staro mesto a prinovezina ti daje autogram od svag erca. Pošto si završio sa ovom odnom, pređi u ovu prvu.

Čim uđeš u ovu odaju idi desno, a onda udi u kupatilo gde se brčka nesudni mornar koji kaže: "You, get me a boat to play with... (Donesi mi brod da se igram). Ali, to nije onaj brod na koji ti mislite i koji stoji na stolu. On će do kraja igre ostati tu. Udi te u sobu gde se ogleđa jedna dama i uzmi predmet koji stoji pored nje. To je daljnji upravljač. Sada idite u sobu sa kaminom i isikame brodnice. Pritenite pucanje i aktivirajte čete helikoptera sa silke. Nijne skinite prvu veliki brod sa silke, pa onda mali. Kada helikopter uz tresak padne na zemlju, pomeće da te juri stražar koji misli da si uvek za sve ti krv (a i jesi), a ti beži u sobu desno sa daljinskim upravljačem u rukama.

Tamo gde je do sada bio zid otvorioće se tajni prolaz i tebi će biti omogućen prolaz u podzemne prostorije zamka. Udi u prolaz i sačekaj da stražar pokaže, pa uzmi mali brod i priteni drugo pucanje. On će nestati ali kada opet priteniš pucanje pojavioće se. Ovim postupkom možeš nositi dva predmeta odjednom. Uzmi i daljnji upravljač, pa kreni da istražuješ tamne proto-

rje zamka. Kada dođeš do prozora, pomeri prečagu tačno pet puta. Time si podigao rešetke sa donjih vrata. Udi u tu prostoriju (sa obelanim kostumom), a vrata će se za tobom zatvoriti. Spusti daljnji i uzmi brod koji je sakriven u vodi. Stavi ga na neko visljivo mesto. Mali brod stavi na sredinu tako da može da prođe kroz šupljinu u zidu. Ispred njega stavi veliki brod i priteni pucanje. Oba broda će otpliviti i proći kroz šupljinu u zidu. A kako ti da izađeš? Prosto, u tome će ti pomoći onaj kostur. Dodji do njega i puvci ga na dale točno trinaest puta, a onda pohlitaj ka izlazu što brže, jer se vrata spusti čim ti mirneš.

Kad naiđeš na prvi RED i i borbu njega, dobijaš tri pečurku kojim pucati na nepristajale. Lutku koja ti donosi neranjivost, ili "scate board" kojim ideš znatno brže. Para ne uzimaj jer uposravaju igru, a na vliam nizovima možeš naći i odskočnu dasku koja ti omogućava da preskočiš tri vatre koje su poredane jedna uz drugu.

WONDER BOY

Priča je jednostavna — dežurni negativac oteo ti je dragu, a ti je spasavaš.

Igra je podeljena u četiri nivoa, a svaki nivo u četiri sektora. Postoje razna smetala (najgora su žabe i oktopodi), ali i pomoćna sredstva.

Kad naiđeš na jele i predes preko njega, dobijaš li pečurku kojom pucati na nepristajale. Lutku koja ti donosi neranjivost, ili "scate board" kojim ideš znatno brže. Para ne uzimaj jer uposravaju igru, a na vliam nizovima možeš naći i odskočnu dasku koja ti omogućava da preskočiš tri vatre koje su poredane jedna uz drugu.

Svaki krug, odnosno predana četiri sektora, završava se borbom sa demonom ili čuvarom tog nivoa, a onda pocije sve ispočetka, samo malo teže.

Da vidimo kako izgledaju ta četiri nivoa:

1 — nalaziš se u šumi, u prvom jektu je pečurka, u drugom "scate board". Lutka se nalazi između dve vatre u indijanskom selu. Na kraju četvrtog sektora moraćeš malo skakati, a pazi i na vatru. Kad stigneš do putokaza sa natpisom "goal", završilo si nivo.

2 — nalaziš se na morskoj obali, u prvom jektu je lutka u drugom pečurka, u trećem, "scate board"; čuval se hobitnica koje lekaču iz vode. Na početku trećeg sektora potreban je veliki skok, pa drži pritenuto: gore+vatra+strana.

3 — na ulazu si u zamak, prate te leteće spodobite i kameñe koje se kotrlja. U trećem sektoru sa tavanice padaju zaleđene gromade, a na početku četvrtog sektora pali na skok.

4 — komañski si u zamku. Na kraju četvrtog sektora susrećeš se sa ogromnom kreaturom sa kojom sam u početku imao mnogo problema. Shema je sledeća: lagano je prati levo-desno. Kad podigne ruku da baci vatrenu kuglu — stani. Kad kugla promaš — skoči visoko (pucanje+skok) i gadjaj je ravno u glavu. Tu ponavljaj sve dok mu glava ne padne sa ramena i razbije se.

Sada on ostavlja određeni predmet. Naravno, u svakom krugu je nešto drugo.

Prvo je trelnja, drugo jabuka, treće opigalo, četvrtu plijeni medved, peto kilobaron, šesto odelo, sedmo praten, a za u, pake koji ćeš naći na kraju ovog teksta sam otkrij isti se dešava na kraju.

Ali, prer toga još dva saveta: Prvo, pazi se petog kruga, četvrtog dela, drugog sektora — veoma ćeš teško izvesti skok, ali kad ti to pade za rukom, odmah skaci dalje na gornji karmen koji se pomena.

Drugo, u sedmi krug, prvi nivo, prvi sektor ni si živu glavu ne ulazi bez pečurke (munjicije), jer ga sigurno nećeš preći, a onda slobodno možeš da iskužiš računar.

Poke za verziju "Ryde" glasi: 10 CLEAR 24576 LOAD " " CODE: POKE 65427195 POKE 65429, 124 20 RANDOMIZE USR 65404: POKE 34361, CODE "vj"—vj" 30 LET A=USR 32768

Vladimir Janković

EXOLON

Tema u principu otrcana. Rešitiš sile koji treba da se duboko probije u nepristajali sistem odbrane koji se sastoji od 125 zona (svaka zona — jedan skrin) i da ga unili (o)ji).

Opremniji ste mećima i granatama, skabaće, vozilo se teleportira, usput snabdevate hranom da ne bi bilo zio i naapako, a smetala kolikogod vam duša želi.

Po vama prašti — biju vas miraljezi, topovi, rakete, gljive, ožinovane čaura, mahanice koje, rudarska kolica, škorpije, pčete, i tako redom.

Poseban problem je i laser u koji treba nemilosrdno pucati ne bi li vam se put otvorio.

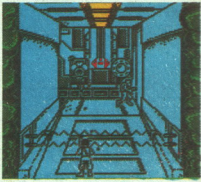
Posle svakih 25 skrinova pojavljuje se bonus ekran — treba u pravom času priteniti dugme za pucanje, tako da pokretna strelica pokaže maksimalni broj bodova.

Jasno je da vam je ružan pouk — u neke od verzija je i ugrađen, i onda ima smisla da pređete ovu igru. A kad se to desi — na ekranu se ukaže tekst i onda — hajde juo nanovo.

Dabome — ko volli

Rusmir Arslanagić

Sinclair GRYZOR GRAJZOR



Vatrenjamači opet izvode neke burgije. Sad su se tih i diskretno spustili na Zemlju, zavukli u džunglu i ograde pogon za obradu atmosfere. Pogon će menjati klimu kako njima odgovara. I tako im obezbediti vlast nad svima nama. Na vama je da se postarate da taj pogon postane promašena investicija.

Uproćeno rečeno, vaša se drama igra u tri čina. U prvom treba da stignete do tudinskog industrijskog kompleksa. Tu morate da silistite džež puk stražara, a u mltirajekim gradinama (ako ih dignete u vazduh) nalazeće ubojnice mašinka — ili tačnije, nađine pušice; normalno, brzo, kružno i trostruko (tj. u tri pravca istovremeno).

U drugom činu udi ste u kompleks, tj. u lavirint tunela. Tu na vas pucaju i gadjaju sa eksplozivnom braući. U trećem, pritenjivate se komandnoj prostoriji, putujući vertikalno i skadujući s jedne platforme na drugu. I to morate pucati iz sve snage, a uspeh će vam zavistiti od toga kakvu ste mašinku uzeli u prvom činu.

Ako razviate tudinski pogon, spašete svih pet milijardi Zemljana — a oni o tome neće imati pojma.

LAZERWHEEL UKRUG SA LASEROM

Mali odmor od igara sa 99 nivoa i 999 ekrana. Ovde su stvari veoma proste.

Usred ekrana je veliki krug. Usred kruga je vaš mali vaskosni brod. Po obodu kruga tudinske letelice koje vaju unistavati. Vaš brod se obrće oko svoje ose, a na njegove kante na moćete uticati: Ako promašite brod, odštelite krug. Kad šteta dostigne 100%, ode žvot. Ima ukupno osam nivoa, i morate skakati iz jednog u drugi zaviseo od toga koji je najugroženiji.

Prosto? Da, ali ne i nezanimljivo. Izazov je, naime, u kombinovanju: morate voditi računsa o svom kretanju,

GOTOVO JE, GOTOVO

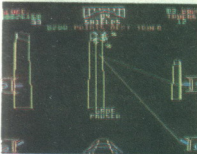
brzini putanji tudaških brodova, i o osobinama svakog tipa broda. Neki dobiću još veću brzinu ili promene pravac kad ih prvi put pogodite, neki se smanjuju ili razvijaju na delove; neke vam biti lako da sve to držite u glavi.

LAZERWHEEL



Zasad zapamtite samo ovo: ako pogodite pravac koji se obrće, popravite ga; ako pogodite te polovine kad se budu okretale, popravite svu štetu koju ste učinili krugu, i dobiti 100 poena pride. Nije lako.

Komodor STAR WARS ZVEZDANI RATOVI



Nagledali smo se tog filma i njegovih nastavaka... I desilo se ono što se desiti moralo: ovo ga u kompjuteru.

Vaš je zadatak veoma human, veoma lako opisiv i veoma leško izvodljiv: morate spasiti svemir. Ništa manje.

Ne sreću, stvor se obavije na samo tri nivoa: put do Zvezde smrti, površine dotične zvezde, i njena unutrašnjost. U prvom, kao što i možete pretpostaviti, pucate kao manjaci — a imate i u šta, jer neprijateljski brodovi ima kao sakavaca u Africi. U drugom, letite iznad površine i uništavate dušmanske instalacije (iz kojih se naravno, puca na vas). U trećem, probijate se kroz utrobu Zvezde smrti — preplunuta smetala — i nastojite da stignete do neke srednje rupe; u nju morate pucati da biste zvezdu razneli u pamparduz i tako smršali konoc zlikovcima.

Pređesete li da će potom doći na red Zvezdani ratovi II, III, IV itd.? I da se nikad neće pojaviti igra nazvana Zvezdani mir?

ZIG ZAG

CIK-CAK

Veoma je važan upadni ugao. Zbog? Evo zašto: vaša letelica ima mnogo šteta, ali ne ima da skrene — a upravo to morate raditi svaki čas, jer futurističke ulice iznad kojih bristete u veoma niskom letu, svaki čas idu u cik-cak; da se ne biste razbili o zid, morate u prizmu (smeštenu tu negde) udariti pod odgovarajućim upadnim uglom; ona će vas tada odbaciti ustranu, i to vam je jedini način da skrenete.

A sada dobra vest: to je jedini komplikovan deo ove igre. Čitav je vaš posao da letite kroz gradsku ulicu, uništavate smetala i skupljate kristale, kojih je ukupno osam. Jednom reči, ušli ste u pucačko-skupljačku trimenzionalnu igru, obogaćenu mogućnošću da poene prevarate u novac za koji kupujete korisne dodatke. Dodatka ima koliko volite: od dopunskog oružja i dopunskih života do infracrvenih uređaja za

82 računari 36 • mart 1988.



zamaćene delove grada i sistema za privremeno ukli-
njenje svih zgrada.

Ako vam je to nedovoljno, možete odabrati varijantu u kojoj uzva sve ovo imate i gomilu logičkih problema. Ako ni time niste zadovoljni, morate učitati neku drugu igru.

SKATE OR DIE VOZI SKEJTBORD ILI UMRI



Ne stigne čovek ni da se okrene, a ono već počnu varijacije na temu. Do juče je postojala samo jedna igra sa skejtbordom — Skate Rock — a sad ih je ima kao kush pasa.

Ova s kojom imamo posla sastavljena je od pet delova. Prva je disciplina „presečena ova“, tj. betonska pista u obliku slova U; na njoj treba izvoditi besne piste. Druga je preste kao pasu; od vas se traži da na takvoj pisti skidate što više. Kad to obavite, čeka vas srijanje nizbrdo, po neravnoj stazi punoj prepreka (a još vam je nametnuto i vremensko ograničenje). Štedi trika sa jednim od tri lika iz kompjutera, po vašem izboru (ili sa živim prijateljem, ako vam je pri ruci); put vas vodi kroz zabačene ulice, gde povrh suparnika imate posta i sa šahovima, zidovima, žičanim mrežama i silničnim radostima. Peti i poslednji čin: našli ste se na drug (prazno) bazena lice u lice sa protivnikom, izabranim kao i malopre. Jedino oružje jeste mašica koja poste svakog petog udara menja vlasnika; dakle, namerično bivate napadati i branilac, a cilj vam je da protivnika oborite triput. Ako uspete, dobijate orden: zlatni skejtbord sa oštećenim zglobovima.

RAMPAGE DIVLJANJE

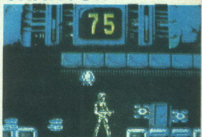


Crveno čudovište Dlord, zeleno čudovište Lzl i plavo čudovište Raf zabavljaju se rušeći gradove, kuću po kuću. Ako unište pedeset gradova, obavili su posao...

... ali stanovnicima gradova to se ne uklapa u generalni urbanistički plan, pa su angažovali oružane snage da smaknu čudovište. Što propuste tenkovi, to sačekaju helikopteri, što propuste helikopteri, sačekaju snajperi; ukratko, trika je velika. Protiv čudovišta radi i njihova sopstvena glad: ako ne da se smanje i nestanu, moraju jesti često i obilato. Hrane ima, a oni nisu izbirivi, i reklo bi se da je sve u redu... da nije jedne male smetlice: tosete i televizore ne smeju jesti, jer će gubiti energiju umesto da je dobijaju. No kad su siti i puni energije, njihovom jedinom oružju, pesnicama, ne može se odupreti niko i ništa.

Možete igrati sami, udvoje, pa i utroje (treći igrač koristi tastaturu; ako su vam promakli MOVIE MONSTERS, ovo će biti vrlo dobra zamena.

TRANTOR



Ne, ovo nije igra o Tarzanovom slonu. Trantor je, ovide, ime čoveka koji brani svoju planetu od osvajača iz neke komećke zabit. On je upao u neprijateljski glavni štab i traži centralni kompjuter da ga razbucua. Uput skuplja sve na šta naiđe jer zna da će mu svaki predmet zatrebati ranije ili kasnije.

Najbolje je da se osvojim nivou nađe terminal (u roku od 90 sekundi). Sastavljanje slova koje tamo nađe, sklopiti lozinku pomoću koje će moći da se uključi u kompjuter i nahrani ga bagovima. Naravno, ne ide svaka lozinka u svaki kompjuter: treba naći pravi.

Neprijatelji samo vrve, i pošto ih je više nego Trantorove municije, korisno je katkad sklanjati im se s puta.

Ali nije sve tako crno. Na kompjuter nivou postaje zalih goriva, energije itd., uključujući i veoma zgodan (ali kratkotrajan) energetski štik. Uključujući, na želost i temperne bombe, koje je lak aktivirati ali teško deaktivirati. Ubratko, čuvajte se vanzemaljaca i kad darove nose.

FLYING SHARK LETEĆA AJKULA



Naslov kao da je izišao iz kakvog srednjovekovnog bestijarnuma, i po tome odmah vidimo da se sredinim vokalom nema nikakve veze. „Leteća ajkula“ jeste stari dobar klasi avion iz, po svemu sudeći, II svetskog rata.

Sedeći za njegovim komandama, vi se borite s avionima nrvskog neprijatelja... ali primedujete da opasnost dolazi i odozdo: sa mora i kopna. Tenkovi, patrolni čamci i bojni brodovi imaju, izgleda, samo jedan cilj: vaš avion.

Vi imate mitraljez i tri „parmetne bombe“. Nije mnogo, ali zalihu bombi možete obnavljati, a vatrene moć vam se povećava ako uništite čitavu eskadrilu crvenih aviona. (Eskadrila žuti, uzgred budi rečeno, donosi vam bonus od 1.000 poena.)

Vaša zlati još i ovo: neprijateljski meci nisu baš najvidljiviji (pozdnuju mahom čine ličike ili talasi; pa nema dobrog kontrasta)... ali šta ćete: rat je rat.



NOVKABEL

KREATIVNOST U OBRAZOVANJU SOFTVERSKI PAKETI U NASTAVI

Kako se menjaju tradicionalni sistemi učenja?

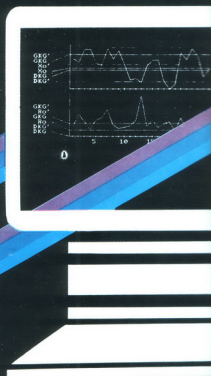
Promene su spore, zavise od razvoja tehnologije, inovacija i filozofije, istražuju se načini kako učenici najlakše i najbolje uče. Na koji način zaključuju i kako donose odluke. Važno je naći najpovoljniji način razmišljanja, uspostaviti uzročnu posledične veze u tokovima razmišljanja. Važno je istraživati svet oko nas i sam život.

Savremene učionice sa kompjuterskim učenjem sastavljaju se sa više stotina kurseva sklopljenih tako da obezbeđuju kompatibilnost načina učenja i iskustva.

U razvoju ovog sistema kompjuter pita, a učenik ima odgovornost odgovora. Ovakav način učenja povećava individualne sposobnosti učenika čak i za dramatičnih 150%. U školama su pedagozi slobodni i svoje radne zadatke mogu da obavljaju kroz neformalne kontakte. Eksperiment humanitarnosti ovde dolazi do izražaja i pospešuje razvoj multidisciplinarnih veština.

Škole se danas takmiče po boljim programima i savršenijim tehničkim pomagalima.

VODI VAS **ET-188**



• UNIVERZALNI PROGRAMSKI
PAKETI U NASTAVI — OSO
SOFTVERSKI PAKETI ZA:

- administraciju
- radionice
- knjigovodstvo i
- biblioteke

RZ KOMERCIJALNI POSLOVI

Poslovnica prodaje

Tel. 021/337-255

21000 Novi Sad

Put Novosadskog partizanskog odreda br. 4

NOVKABEL — RAČUNARI U
ŠKOLAMA



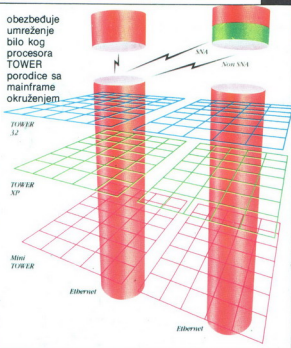
NCR TOWER 32

PROCESOR	32 Bit MC 68020
MULTI BUS PROCESORI	7
GLAVNA MEMORIJA	1 — 16 MB
CACHE	8 KB standard
VRJEME CIKLOSA	16,7 MHz, 60 ns
KAPACITET DISKETE	1,2 MB
WINCHESTER DISK (26—140 MB)	2
DODATNI WINCH. DISK (340 MB)	16
STREAMER TRAKA	45 MB
KOMUNIKACIJE—PROTOKOLI	TTY, HDLC, SDLC
	3270 BSC, 2780/3780 BSC
	SNA, X.25, ETHERNET
	10° — 40° C
	20 — 80% _o
	UNIX Sistem V
	„C“, COBOL, BASIC,
	FORTRAN, PASCAL
	do 32/48

- MREŽE
- RADNI USLOVI — temp.
- Rel. vlažnost
- OPERATIVNI SISTEM
- JEZIK

TERMINAL PRIKLJUČAK

TOWERNET



TOWER 32

- Nudi različita rešenja za:
- biro komunikacije
- SNA — I X. 25 — implementaciju modularnu dogradnju
- izgradnju lokalne mreže LAN
- primjenu u raznim oblastima
- priključenje svih, na tržištu raspoloživih perifernih sistema
- obradu teksta



UNIS — Elektronika Telekomunikacije Informatika

Iz proizvodnog programa UNIS-ETI predstavljamo TOWER-32, trenutno najnašniji sistem TOWER porodice računara. Za sve informacije stojimo vam na raspolaganju.

OUR MARKETING
58000 Mostar, M. Tita 237
(Tel. 088/35-077)

Poslovne jedinice:
Beograd, M. Pijade 117/3
(Tel. 011/338-659)

SEKTOR PLASMANA OPREME ZA AOP
71000 Sarajevo, Tršćanska br. 7
(Tel. 071/215-522/lok. 2455, 2456, 2475)

Zagreb, Optička 27/1
(Tel. 041/435-746)

Rijeka, Đure Šporera bb
(Tel. 051/37-693)

